вступ

Актуальність роботи

Сучасні наукові дослідження активно зосереджуються на галузі комп'ютерного зору, яка останнім часом здобуває все більше уваги та практичного застосування. Зростання інтересу до аналізу руху людини зумовлене покращенням алгоритмів і моделей комп'ютерного зору, а також доступністю недорогого обладнання, такого як відеокамери, та появою нових програмних рішень, що здатні виконувати як ідентифікацію осіб, так і візуальний моніторинг.

Головною метою виявлення руху є визначення зміни положення об'єкта між двома послідовними зображеннями. Аналіз руху також сприяє підвищенню точності розпізнавання об'єктів, оскільки дозволяє виділити пікселі, що належать конкретному об'єкту. Водночас ефективність процесу детекції значною мірою залежить від зовнішніх факторів, серед яких:

- технічні характеристики камери та її стабільне положення відіграють ключову роль у точності виявлення руху;
- рівномірне освітлення без мерехтіння є критично важливим;
- висока частота кадрів та роздільна здатність камери значно підвищують ефективність детекції.

Аналіз руху людського тіла привертає увагу дослідників через його різноманітність і вплив на такі сфери, як фізична активність, оцінка продуктивності, медична діагностика та застосування у віртуальній реальності. На сьогоднішній день для виявлення рухомих об'єктів найчастіше використовують такі методи: кадрове віднімання, віднімання фону та оптичний потік.

Метод оптичного потоку передбачає обчислення векторного поля руху пікселів на зображенні та подальше кластерне групування на основі характеристик цього поля. Метод фонового віднімання полягає у визначенні різниці між поточним кадром та попередньо збереженим фоновим зображенням для виявлення руху об'єктів. Цей підхід, однак, дуже чутливий до змін зовнішнього середовища та недостатньо стійкий до перешкод.

Кадровий метод віднімання визначає наявність рухомих об’єктів шляхом обчислення різниці між двома послідовними кадрами. Будь-яка система, що базується на фоні, завжди буде чутлива до зовнішніх факторів, серед яких:

- наявність шумів на зображенні через низьку якість джерела відеосигналу;
- поступові зміни освітлення у сцені протягом часу;
- незначні рухи нестатичних об’єктів, таких як гілки дерев, кущі тощо.

Отже, для створення ефективної програмної системи відеоспостереження з функцією детекції рухомих об’єктів необхідно розробити алгоритм на основі методу ковзаючого вікна. При цьому слід визначити: граничну область, де спостереження ведеться постійно, та центральну область, яка контролюється лише у випадку виявлення руху. Таким чином, задача створення алгоритму детекції рухомих об’єктів із використанням ковзаючого вікна залишається актуальною.

Метою даної роботи є розробка алгоритму детекції рухомих об’єктів на основі ковзаючого вікна для комп’ютерних систем відеоспостереження.

Для досягнення цієї мети поставлено наступні завдання:

- здійснити аналіз і класифікацію основних завдань у галузі комп’ютерного зору;
- вивчити методи кодування та передачі відеосигналів;
- провести огляд наявних програмних рішень для виділення та класифікації об’єктів на зображеннях;
- дослідити існуючі підходи та методи для розпізнавання та класифікації об’єктів;
- розробити алгоритм детекції рухомих об’єктів на основі ковзаючого вікна;
- реалізувати програмну систему детекції об’єктів із подальшим тестуванням та порівнянням її ефективності з аналогічними програмами.

Об’єктом дослідження є процес обробки та аналізу цифрових зображень у системах відеоспостереження.

Предмет дослідження – методи та алгоритми розпізнавання об’єктів на цифрових зображеннях у відеопотоці.

- проведено всебічний аналіз і класифікацію алгоритмів виділення об’єктів на цифрових зображеннях, що дозволило виявити їхні сильні та слабкі сторони, а також розробити власний алгоритм детекції руху на основі ковзаючого вікна;
- розроблено алгоритм детекції рухомих об’єктів із використанням граничних полів області спостереження, що зменшує обчислювальну складність процесу і підвищує ефективність роботи системи.

Практична цінність отриманих результатів полягає у наступному:

- проведено теоретичне дослідження програмної системи відеоспостереження з елементами автоматичної детекції руху на основі ковзаючого вікна, що створило підґрунтя для її подальшої реалізації та перевірки розроблених алгоритмів;
- реалізовано програмне забезпечення для детекції рухомих об’єктів із використанням бібліотек OpenCV та ImageAI, а також алгоритмів комп’ютерного зору.

Публікації та апробація результатів. За матеріалами дослідження підготовлено тези доповідей на V Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі»:

- «Аналіз алгоритмів попередньої обробки зображень у системах автоматизованого моніторингу» (1 сторінка);
- «Аналіз алгоритмів сегментації для систем автоматизованого аналізу зображень» (1 сторінка).
- **1.1 Комп’ютерний зір: базові поняття та основні функції**
- На сучасному етапі розвитку штучного інтелекту алгоритми комп’ютерного зору (Computer Vision, CV) належать до найбільш інноваційних і потужних технологій. Системи комп’ютерного зору застосовуються у різних сферах, включаючи автономний транспорт, робототехніку, навігаційні системи, розпізнавання облич, контроль виробничих процесів і відеоспостереження.
- Щоб зрозуміти принципи роботи комп’ютерного зору, корисно провести аналогію з людським сприйняттям. Людина розпізнає об’єкти завдяки аналізу їхніх ключових характеристик: форми, кольору, глибини та текстури. Візуальні сигнали, що надходять від очей до мозку, спершу обробляються на рівні

виділення контурів і границь об’єкта. Потім ці елементарні ознаки об’єднуються в більш складні структури, що дозволяє сформувати цілісне сприйняття форми та просторового розташування об’єкта.

- Аналогічно, алгоритми комп’ютерного зору здійснюють багаторівневу обробку зображень. На початковому етапі визначаються контури та ключові ознаки об’єктів, після чого здійснюється класифікація та розпізнавання складніших структур. Такий підхід дозволяє системам CV ефективно ідентифікувати об’єкти, відстежувати їхнє положення та взаємодію у відеопотоці.

  Системи комп’ютерного зору функціонують подібно до людського зору: спочатку визначають контури об’єкта, а потім об’єднують їх для формування цілісного образу. Основна відмінність полягає в тому, що комп’ютери сприймають зображення у вигляді числових матриць, де кожен піксель має своє числове значення. Для аналізу системі необхідно оцінювати відмінності між сусідніми пікселями, щоб ідентифікувати краї об’єктів. Наприклад, у зображенні в градаціях сірого кожен піксель може приймати значення від 0 (чорний) до 255 (білий). Різка зміна значень пікселів у сусідніх точках сигналізує про наявність краю об’єкта.

  Цей базовий принцип порівняння пікселів застосовується й до кольорових зображень, коли система аналізує відмінності між каналами RGB. Таким чином, комп’ютерна система зору оцінює значення пікселів у різних спектральних компонентах, що дозволяє точно інтерпретувати вміст зображення. На прикладі згорткової нейронної мережі (CNN) можна наочно продемонструвати архітектуру таких систем.

  Згорткові нейронні мережі є ключовим типом штучного інтелекту для завдань комп’ютерного зору. Основний механізм — операції згортки, які дозволяють виявляти відмінності між пікселями на локальному рівні. Уявімо матрицю значень пікселів, на яку накладається менша матриця — ядро згортки. Це ядро переміщується по всьому зображенню, обробляючи невелику групу пікселів за один крок, що часто називають технікою «ковзного вікна». Значення, отримані під час такого аналізу, агрегуються мережею, що дозволяє зменшити складність зображення і полегшує виявлення характерних шаблонів.

Згорткові нейронні мережі (CNN) умовно поділяються на два типи шарів:

- **Згорткові шари** – слугують екстракторами ознак, їхнє завдання полягає в аналізі пікселів зображення та формуванні представлення даних, яке надалі використовується щільно пов’язаними шарами мережі для вивчення шаблонів. На початкових етапах згорткові шари виділяють базові ознаки зображення, такі як краї, а більш глибокі шари об’єднують ці елементи у складніші форми. В результаті мережа отримує комплексне уявлення про деталі зображення, яке передається повністю пов’язаним шарам.
- **Повністю пов’язані шари** – використовують інформацію, зібрану згортковими шарами, для класифікації або прийняття рішень на основі отриманих ознак.

Для ефективного використання згорткових нейронних мереж виділяють ключові етапи роботи, представлені на схемі **Рисунок 1.1**.

Згортка (Convolution). Згортка застосовується до зображення для виділення певних елементів і характеристик. Вона може виконувати такі функції, як розмивання, підвищення різкості, виявлення країв або зменшення шуму. Це дозволяє мережі ефективно розпізнавати конкретні ознаки та шаблони на зображенні, що є основою для подальшого навчання та класифікації.

Об’єднання (Pooling). Після згортки розмір зображення або карти ознак може бути занадто великим для подальшої обробки. Об’єднання використовується для зменшення розмірності карти ознак без втрати важливих функцій та шаблонів, що дозволяє прискорити обчислення та зменшити ризик перенавчання.

Згладжування (Flattening). Після етапу об’єднання двовимірна матриця ознак перетворюється на одномірний вектор. Цей вектор може бути поданий на вхід повністю пов’язаним шарам нейронної мережі або іншому класифікатору.

Повне підключення (Fully Connected Layers). На цьому етапі сплощене зображення або вектор ознак передається у повністю пов’язаний шар нейронної мережі. Тут здійснюється інтеграція та комбінування всіх ознак, що дозволяє мережі робити остаточні висновки або класифікувати об’єкти на зображенні.

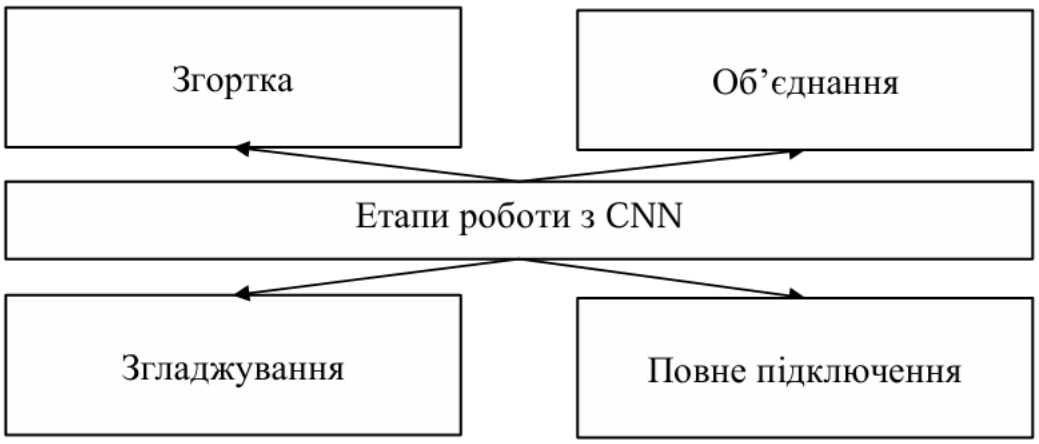


Рисунок 1.1 – Ключові етапи роботи згорткової нейронної мережі (CNN)

На рисунку показано послідовність основних етапів:

1. **Згортка (Convolution)** – виділення ознак зображення, таких як краї, текстури та деталі.
2. **Об’єднання (Pooling)** – зменшення розмірності карти ознак без втрати ключових характеристик.
3. **Згладжування (Flattening)** – перетворення двовимірної матриці ознак у одномірний вектор для подачі на класифікатор.
4. **Повне підключення (Fully Connected Layers)** – інтеграція всіх ознак для фінальної класифікації об’єктів на зображенні.

Хоча згорткова нейронна мережа здатна самостійно витягувати шаблони зображень, точність систем комп’ютерного зору значно підвищується при використанні анотацій. Анотація зображення – це процес додавання метаданих до пікселів або об’єктів на зображенні, що дозволяє класифікатору більш точно ідентифікувати важливі елементи сцени. Використання анотацій особливо критично у високоточних системах, таких як автономні транспортні засоби або робототехніка, де правильне розпізнавання об’єктів впливає на безпеку та ефективність роботи системи.

Існують різні підходи до анотації зображень, які покращують продуктивність класифікаторів у системах комп’ютерного зору. Найпоширенішим методом є використання обмежувальних рамок (bounding boxes), які оточують цільовий об’єкт та спрямовують увагу алгоритму всередину рамки.

Іншим підходом є семантична сегментація, яка передбачає присвоєння кожному пікселю на зображенні певного класу. Наприклад, пікселі, що відповідають «траві» або «деревам», позначаються як належні до відповідних класів. Цей метод забезпечує високу точність на рівні пікселів, проте його створення є більш складним та трудомістким порівняно з обмежувальними рамками.

Крім того, використовуються й інші методи анотації, такі як позначення об'єктів за допомогою ліній або точок, що дозволяє зменшити обчислювальні витрати та спростити підготовку навчальних даних.

Машинне навчання (ML) є однією з найбільш швидкозростаючих галузей сучасних технологій. Незважаючи на широке використання цього терміну, його сутність іноді складно зрозуміти. ML не стосується одного конкретного методу, а є загальним поняттям, що охоплює різні концепції та техніки аналізу даних.

У загальному сенсі, машинне навчання полягає у наданні комп'ютеру здатності виконувати завдання без явного програмування всіх кроків їх вирішення. Комп'ютер «навчається», виявляючи закономірності в даних і використовуючи їх для узагальнення на нові, раніше невідомі дані. Це дозволяє ефективно вирішувати складні завдання, для яких традиційне програмування було б непридатним.

Системи машинного навчання складаються з трьох основних компонентів:

1. **Входи (Inputs)** – дані, що подаються в систему для навчання. Вхідні дані поділяються на:
   - **Функції (Features)** – змінні, які аналізуються для виявлення закономірностей;
   - **Мітки (Labels)** – класи або описи, які асоціюються з конкретними прикладами даних.
2. **Алгоритми (Algorithms)** – методи обробки та аналізу вхідних даних для навчання моделі.
3. **Виходи (Outputs)** – результати роботи системи, що можуть включати прогноз, класифікацію або рекомендації на основі аналізованих даних.

Використання функцій і міток визначає тип машинного навчання:

- **Навчання з наглядом (Supervised Learning)** – система навчається на підготовлених даних з відомими мітками;
- **Навчання без нагляду (Unsupervised Learning)** – система знаходить закономірності в даних без попередньо заданих міток (рисунок 1.2).

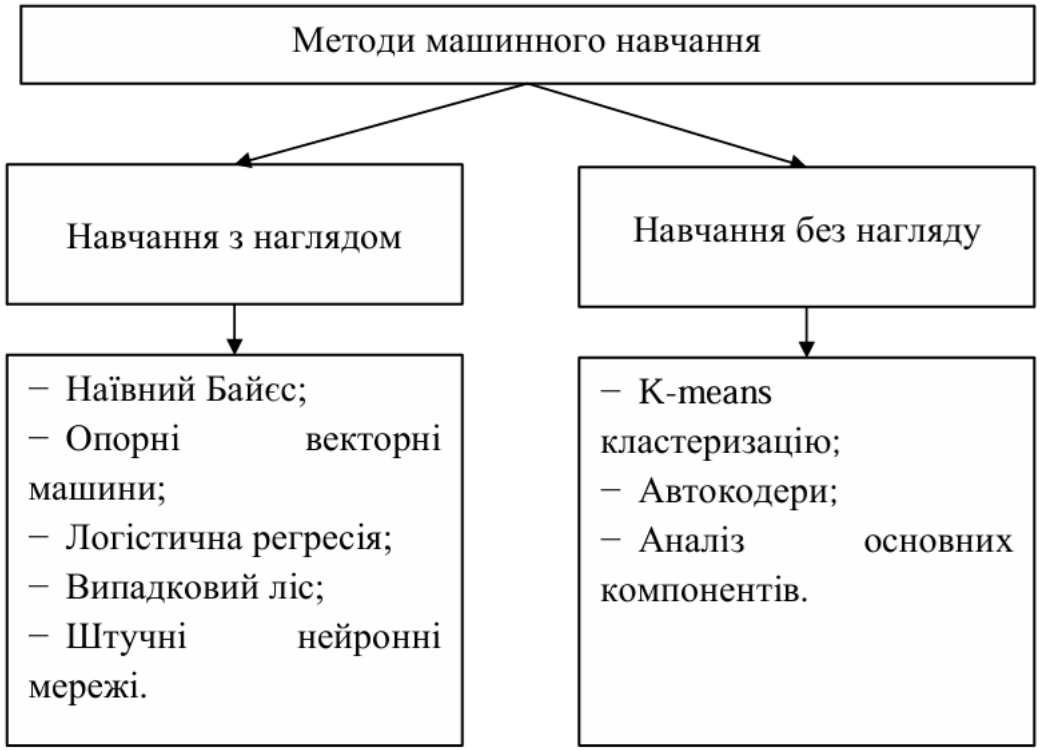


Рисунок 1.2 – Класифікація методів машинного навчання

Навчання з наглядом та навчання без нагляду

При навчанні з наглядом (supervised learning) вхідні дані супроводжуються правильними відповідями, тобто «істинними» мітками. Це дозволяє алгоритму зіставляти вхідні дані з очікуваними результатами. Такий підхід дає змогу фахівцям оцінювати ефективність моделі, порівнюючи передбачення алгоритму з відомими значеннями у тестовому наборі даних і визначати відсоток правильно класифікованих прикладів.

На відміну від цього, навчання без нагляду (unsupervised learning) працює з даними без попередніх міток. Алгоритм самостійно знаходить закономірності та структури у наборі даних, формуючи власні категорії або групи.

Алгоритми навчання з наглядом найчастіше застосовуються для задач класифікації, де дані потрібно розподілити за заданими категоріями. Інший тип

завдань – регресія, де алгоритм прогнозує безперервні значення, а не конкретні класи.

Алгоритми навчання без нагляду зазвичай застосовуються для задач оцінки щільності даних, кластеризації та навчання репрезентацій. У цих випадках моделі повинні самостійно визначати структуру даних, оскільки заздалегідь визначені класи відсутні.

Серед найпоширеніших алгоритмів, що застосовуються у навчанні з наглядом, можна виділити:

- наївний Байєс;
- опорні векторні машини (SVM);
- логістичну регресію;
- випадковий ліс;
- штучні нейронні мережі.

Опорні векторні машини використовуються для поділу набору даних на різні класи. Алгоритм визначає лінію (або гіперплощину в багатовимірному просторі), яка відокремлює дані різних категорій. Точки, розташовані з одного боку лінії, відносяться до одного класу, а з іншого боку — до іншого. Мета алгоритму полягає у максимізації відстані між гіперплощиною та найближчими точками з обох сторін, що підвищує впевненість класифікатора у належності об’єкта до певного класу.

Логістична регресія застосовується для задач двійкової класифікації, коли необхідно визначити, до якого з двох класів належить об’єкт. Кожна точка даних отримує значення 0 або 1. Якщо розраховане значення менше або дорівнює 0,49, точка відноситься до класу 0, а якщо воно становить 0,5 або більше — до класу 1.

Алгоритми дерев рішень працюють шляхом послідовного поділу набору даних на все менші підгрупи. Критерії розподілу визначаються інженером з машинного навчання, а метою є досягнення чіткого поділу даних на окремі елементи, які потім можна ефективно класифікувати.

Random Forest представляє собою ансамбль численних дерев рішень, які спільно утворюють більш надійний та точний класифікатор.

Наївний байєсовий класифікатор оцінює ймовірність належності точки даних до певного класу на основі ймовірності попередніх подій. Цей підхід ґрунтується на теоремі Байєса і дозволяє розподіляти дані між класами залежно від обчислених ймовірностей. При застосуванні наївного байєсового методу передбачається, що всі ознаки мають однаковий вплив на результат класифікації.

Штучні нейронні мережі (ШНМ), зокрема багатошаровий персептрон, є алгоритмами машинного навчання, що надихаються структурою та функціями людського мозку. Мережа складається з численних нейронів, взаємопов'язаних між собою, які обробляють інформацію за допомогою математичних функцій. У ШНМ виділяють вхідний шар, приховані шари та вихідний шар.

Приховані шари відповідають за обробку та аналіз даних для виявлення шаблонів, тобто саме тут відбувається навчання алгоритму. Збільшення кількості нейронів і шарів дозволяє мережі виявляти більш складні закономірності та ефективніше опрацьовувати складні дані.

Типи навчання без нагляду

Серед алгоритмів неконтрольованого навчання виділяють такі:

- **Кластеризація K-середніх (K-means)**;
- **Автокодери (Autoencoders)**;
- **Аналіз головних компонент (PCA, Principal Component Analysis)**.

Кластеризація K-середніх — це методика, яка дозволяє групувати точки даних у кластери на основі схожості їхніх характеристик. Алгоритм аналізує ознаки об'єктів і формує групи так, щоб елементи всередині одного кластера були більш подібними один до одного, ніж до елементів інших кластерів.

Принцип роботи K-means полягає у виборі початкових центрів кластерів (центроїдів), після чого алгоритм ітеративно перепризначає положення центроїдів, щоб мінімізувати відстань між центроїдом і точками даних, які до нього належать. Кількість кластерів визначає дослідник відповідно до завдання.

Аналіз головних компонент (PCA) — це метод зменшення розмірності даних, який дозволяє скоротити кількість змінних або ознак до більш компактного набору. У процесі PCA визначаються основні компоненти, які зберігають

найбільшу частку варіації в даних, тоді як менш важливі ознаки стискаються в більш просте представлення. Такий підхід дозволяє зберегти структуру та взаємозв'язки між вихідними даними, водночас спрощуючи їхнє подальше аналізування та обробку.

Автокодери — це тип нейронних мереж, призначений для задач навчання без нагляду. Вони можуть обробляти дані без наявності міток і перетворювати їх у таку форму, яку мережа здатна використовувати для навчання, фактично генеруючи власні внутрішні «мітки». Основна мета автокодера полягає у відтворенні вхідних даних із максимальною точністю, що змушує мережу виявляти найбільш значущі ознаки та ефективно їх виділяти.

1.2 Аналіз технологій та протоколів передачі відео

Передача відеопотоків є ключовим елементом сучасних систем відеоспостереження та мультимедійних сервісів. Вона забезпечує доставку даних від джерела до кінцевого користувача або аналітичної системи в режимі реального часу, зберігаючи якість зображення та мінімізуючи затримки.

Основні технології передачі відео

1. **H.264/AVC** – одна з найпоширеніших технологій кодування відео, яка забезпечує високу компресію зображення при збереженні якості. Використовується в потокових сервісах, відеоконференціях та системах відеоспостереження.
2. **H.265/HEVC** – більш сучасний стандарт кодування, який дозволяє зменшити обсяг відеопотоку майже вдвічі порівняно з H.264 при схожій якості. Це робить його оптимальним для високоякісного відео 4K та 8K.
3. **VP8/VP9** – відкриті формати кодування, активно застосовуються в веб-технологіях, зокрема для потокового відео через браузери та сервіс YouTube.

Протоколи передачі відео

Для забезпечення ефективної доставки відеопотоків застосовуються різні мережеві протоколи:

- **RTSP (Real-Time Streaming Protocol)** – протокол, що дозволяє організовувати потокову передачу медіаданих у реальному часі, керувати відтворенням та синхронізувати клієнтські запити з сервером.

- **RTP/RTCP (Real-time Transport Protocol / Real-time Transport Control Protocol)** – протоколи, які працюють у парі для передачі аудіо- та відеопотоків із контролем якості доставки та часової синхронізації.
- **HTTP Live Streaming (HLS)** – технологія потокової передачі, що розбиває відео на сегменти та передає їх через HTTP. Підтримує адаптивну зміну якості потоку залежно від пропускної здатності мережі.

  Протокол потокового відео визначає стандартизований спосіб доставки відео, який передбачає розбиття його на сегменти, передачу цих сегментів приймачу та подальшу збірку для відтворення. Більшість цифрових відеофайлів розроблені переважно для зберігання та локального відтворення, що робить їх оптимальними за розміром і сумісними з різними пристроями, але не призначеними для потокової трансляції.

  Щоб забезпечити потокове відтворення, відео необхідно конвертувати у спеціальний формат, розбивши його на невеликі фрагменти. Ці фрагменти передаються послідовно та відтворюються по мірі надходження. У випадку трансляцій у реальному часі дані надходять безпосередньо з камери, тоді як для відео за запитом (VOD) потік формується з файлу.

  Сучасні протоколи потокової передачі часто включають функції адаптивного бітрейту, які дозволяють динамічно змінювати якість відео відповідно до поточної пропускної здатності мережі та можливостей пристрою користувача. Окремі протоколи оптимізовані для зменшення затримки між подією та її відтворенням на екрані, деякі працюють у конкретних екосистемах, а інші забезпечують контроль цифрових прав (DRM), захищаючи контент від несанкціонованого доступу або копіювання.

  Термін «кодек» позначає технологію, яка використовується для стиснення та декомпресії відео. Різні кодеки підходять для різних завдань: наприклад, Apple ProRes зазвичай застосовують для обробки та монтажу відео, тоді як H.264 є одним із найпоширеніших кодеків для потокового онлайн-відео завдяки його високій сумісності та ефективності стиснення.

Також варто розрізняти поняття «формат» у контексті відео. Часто під форматом мається на увазі контейнер відеофайлу, який об’єднує відео, аудіо та метадані в одному файлі. Найпопулярніші контейнери включають .mp4, .m4v та .avi. Контейнер виконує роль «коробки», у якій зберігаються всі компоненти, але він не є визначальним елементом для прямих трансляцій відео.

HTTP Live Streaming (HLS) — це протокол потокової передачі, розроблений компанією Apple. Він підтримується різноманітними платформами, включаючи медіаплеєри, веб-браузери, мобільні пристрої та медіа-сервери. HLS був представлений у 2009 році, щоб замінити Flash на iPhone, і з того часу став одним із найпоширеніших протоколів потокового відео. Цей протокол підтримується сучасними настільними браузерами, смарт-телевізорами, а також мобільними пристроями на базі Android та iOS. Крім того, відеоплеєри HTML5 можуть відтворювати відео через HLS, забезпечуючи сумісність з великою кількістю пристроїв.

Це дозволяє HLS досягати максимальної аудиторії, що робить його одним із найбільш надійних протоколів для масштабування прямих трансляцій. Стандарт HLS також підтримує адаптивний бітрейт, завдяки чому глядачі отримують найкращу можливу якість відео в реальному часі. Сучасні оновлення протоколу включають підтримку кодека H.265, який забезпечує подвоєну якість зображення при такому ж обсязі файлу, як H.264. Головним недоліком HLS залишається відносно висока затримка, проте існують методи та налаштування, які дозволяють її значно зменшити.

Переваги HLS:

- **Широка сумісність:** HLS підтримується відеопрогравачами HTML5, тому його можна використовувати для потокового відтворення на більшості пристроїв і операційних систем із доступом до Інтернету.
- **Безпека:** Протокол забезпечує надійну та захищену передачу відеопотоків.
- **Висока якість:** Використання адаптивного бітрейту дозволяє транслювати відео з високою роздільною здатністю та стабільною якістю.

Недоліки HLS:

- **Велика затримка:** HLS не може забезпечити настільки низьку затримку, як деякі інші протоколи потокової передачі.
- **Обмеженість для прийому:** HLS не завжди є оптимальним для приймальних пристроїв через обмежену доступність або високу вартість сумісних кодерів.

Протокол RTMP (Real-Time Messaging Protocol). Спочатку RTMP був розроблений компанією Macromedia на ранніх етапах розвитку потокового відео і досі залишається поширеним у певних сферах. Сьогодні цей протокол переважно використовується для надсилання прямих трансляцій із кодера, який підтримує RTMP. Водночас RTMP майже не застосовується для безпосереднього відтворення відео у глядачів, як це було раніше, через залежність від плагіна Flash, який нині застарів.

Переваги RTMP:

- **Низька затримка:** Протокол забезпечує мінімальну затримку відеопотоку, що дозволяє підтримувати стабільне з'єднання навіть при нестійкому інтернет-з'єднанні. Це означає менші паузи для глядачів і швидке відновлення трансляції після короткочасних перебоїв з мережею.

Переваги використання RTMP:

- **Адаптивність:** Завдяки адаптивному каналу глядачі можуть не лише переглядати прямий ефір, а й перемотувати його назад або приєднуватися до трансляції після її початку, якщо вміст зберігається на сервері RTMP.
- **Гнучкість:** Протокол дозволяє об'єднувати різні типи медіафайлів в один потік, комбінуючи аудіо, відео та субтитри. Також можливе використання кількох форматів, наприклад, аудіопотоки MP3 або AAC, відеопотоки MP4, FLV та F4V.

Недоліки RTMP:

- **Відсутність підтримки HTML5:** RTMP працює через Flash-плеєри, які застаріли. Для відтворення на сучасних HTML5-плеєрах потрібна конверсія у формат, наприклад HLS.
- **Чутливість до пропускної здатності:** RTMP-потоки можуть перериватися при низькій швидкості інтернету, що негативно впливає на якість перегляду.

- **Несумісність з HTTP:** RTMP не можна передавати безпосередньо через HTTP-з’єднання. Щоб показати потік на вебсайті, необхідно підключатися до спеціального сервера, наприклад Flash Media Server, і використовувати CDN.

Web Real-Time Communications (WebRTC) – це проект з відкритим вихідним кодом, призначений для трансляції відеопотоків із мінімальною затримкою. Спершу його створили для підтримки VoIP-технологій, а пізніше Google інтегрував його у свої інструменти для відеочату. Хоча WebRTC технічно є проектом потокової передачі, а не окремим протоколом, його часто використовують як частину потокових рішень через спільні функції.

Переваги WebRTC:

- **Відкритий код:** Проект можна налаштувати під конкретні потреби завдяки відкритому вихідному коду.
- **Мінімальна затримка:** Підтримка потокової передачі у режимі реального часу дозволяє передавати відео майже миттєво на пристрої глядачів.

Недоліки WebRTC:

- **Обмежена сумісність:** Технологія ще розвивається, і деякі системи потокової передачі не повністю сумісні або потребують додаткового налаштування.

2. МЕТОДИ ТА АЛГОРИТМИ ВИДІЛЕННЯ ТА КЛАСИФІКАЦІЇ ОБ'ЄКТІВ НА ЦИФРОВИХ ЗОБРАЖЕННЯХ

2.2Методи та алгоритми виділення об'єктів на зображенні

Методи детекції об'єктів і класифікації зображень посідають ключове місце серед підходів, що застосовуються у сфері комп'ютерного зору (рис. 2.1). Використання цих методів дає змогу автоматизованим системам інтерпретувати візуальну інформацію, розпізнавати об'єкти та аналізувати навколишнє середовище в режимі реального часу на основі цифрових зображень, які виступають вхідними даними.

Упродовж тривалого часу технології комп'ютерного зору активно впроваджуються в різних галузях, зокрема в медицині, промисловому виробництві, сфері роздрібної торгівлі та багатьох інших напрямках, де необхідна автоматична обробка візуальної інформації.

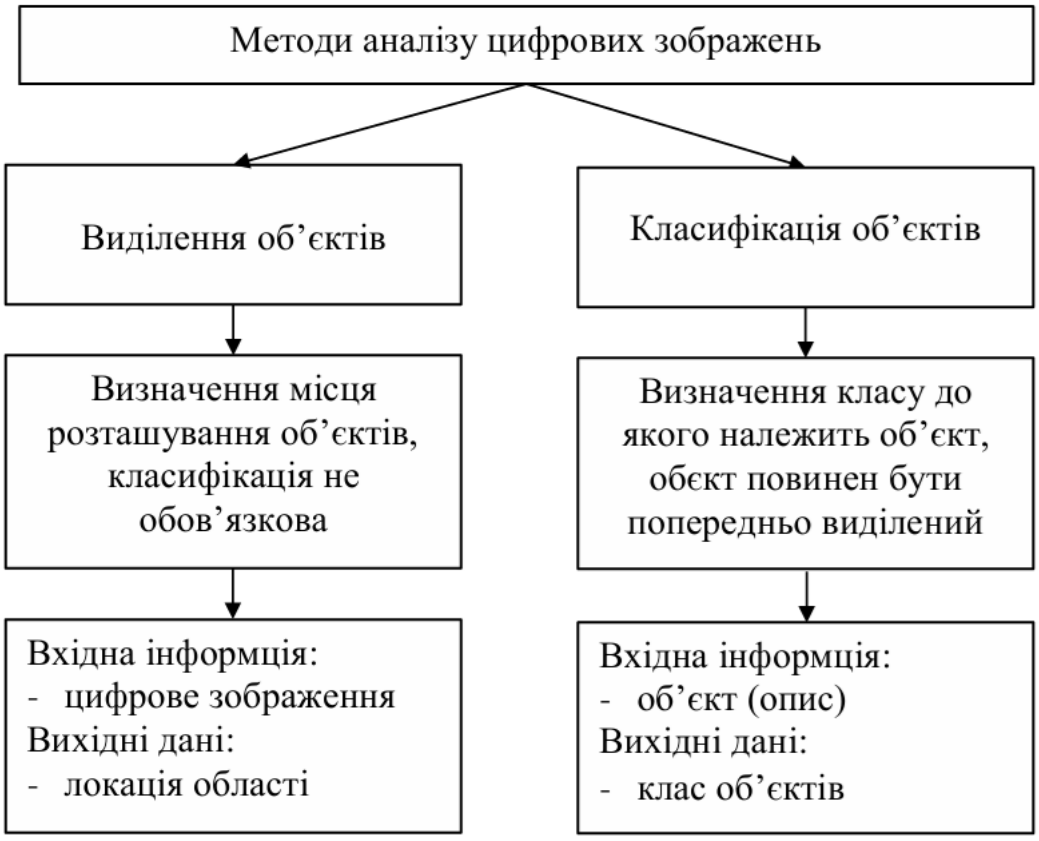


Рис. 2.1 Класифікація алгоритмів виділення (детекції) та класифікації об’єктів на зображенні

Оскільки такі підходи, як класифікація зображень і детекція об’єктів, спрямовані на ідентифікацію візуальних об’єктів у цифрових зображеннях, між ними досить часто виникає плутанина. Це зумовлює необхідність чіткого розмежування сутності кожного з методів та визначення їх ключових відмінностей.

Класифікація зображень являє собою метод обробки зорових даних, основним завданням якого є віднесення зображення або об’єкта на ньому до певного класу. Головною метою даного підходу є коректне визначення характерних ознак зображення та подальше прогнозування його класової належності.

Методи класифікації зображень зазвичай поділяються на параметричні та непараметричні, контрольовані та неконтрольовані, а також на жорсткі та м’які класифікатори. У випадку контрольованої класифікації рішення формується на основі навчальної вибірки з наперед заданими вхідними та вихідними даними, що дозволяє побудувати межу прийняття рішень. Натомість неконтрольована класифікація базується на самостійному аналізі структури вхідних даних без використання попередньо визначених міток або класів.

Процес класифікації зображень включає низку послідовних етапів, серед яких: вибір системи класифікації, виділення інформативних ознак, формування навчальної вибірки, попередня обробка зображень, застосування відповідного класифікатора, постобробка результатів та оцінювання точності отриманих результатів. У таких системах вхідними даними зазвичай є зображення об’єкта, а вихідними — клас або категорія, до якої він належить. Найбільш поширеним інструментом для розв’язання задач класифікації зображень є згорткові нейронні мережі (Convolutional Neural Networks, CNN).

У повсякденному житті користувачі регулярно взаємодіють із технологіями, що реалізують алгоритми детекції об’єктів. Прикладами є системи розпізнавання облич для розблокування мобільних пристроїв або системи відеоспостереження в торговельних і складських приміщеннях, які автоматично виявляють підозрілу поведінку чи несанкціоновані дії.

Основні напрями застосування методів і алгоритмів виявлення об’єктів наведені на рис. 2.2.



Рис. 2.2. Сфери застосування алгоритмів виділення об’єктів

Одним із поширених практичних прикладів застосування методів детекції об’єктів є система розпізнавання номерних знаків транспортних засобів. У таких рішеннях поєднуються алгоритми виявлення об’єктів та оптичного розпізнавання символів (OCR), що дає змогу автоматично ідентифікувати буквено-цифрові послідовності на автомобілях. На першому етапі модель детекції визначає транспортний засіб і локалізує область номерного знака на зображенні, після чого OCR-модуль перетворює двовимірне зображення символів у текстове представлення, придатне для машинної обробки.

Ще одним важливим напрямом є виявлення та розпізнавання облич. Сучасні алгоритми комп’ютерного зору дозволяють автоматично визначати наявність людських облич у статичних зображеннях або відеопослідовностях. Завдяки розвитку методів машинного навчання, зокрема одноразового (one-shot) навчання,

стало можливим коректно розпізнавати особу навіть за мінімальної кількості навчальних зразків.

Значну роль у відеоаналітиці відіграє задача відстеження об’єктів. Наприклад, під час спортивних трансляцій м’яч переміщується з високою швидкістю та змінює напрям руху, що ускладнює його візуалізацію. Алгоритми трекінгу дозволяють безперервно фіксувати положення об’єкта, аналізувати траєкторію його руху та оцінювати пройдену відстань у реальному часі.

Особливої актуальності методи виявлення об’єктів набули у сфері автономного керування транспортними засобами. Для безпечного руху безпілотні автомобілі повинні в режимі реального часу розпізнавати пішоходів, транспортні засоби, дорожні знаки та інші об’єкти навколишнього середовища. Навчені на багатьох класах моделі детекції є критично важливими для забезпечення стабільної та надійної роботи таких систем.

У робототехніці алгоритми детекції застосовуються для автоматизації різноманітних операцій, зокрема задач захоплення, переміщення та розміщення об’єктів. Завдяки виявленню об’єктів роботи отримують можливість орієнтуватися в просторі, коректно взаємодіяти з навколишнім середовищем і виконувати складні дії в режимі реального часу.

Задача виявлення об’єктів полягає у визначенні просторового розташування об’єктів на зображенні (локалізація), а також у встановленні їхньої приналежності до певного класу (класифікація). Таким чином, детекція об’єктів поєднує в собі елементи класичної класифікації зображень та задачі просторового аналізу, забезпечуючи пошук і ідентифікацію об’єктів із заданого набору категорій у складних природних сценах.

Методи детекції широко застосовуються для пошуку визначених класів об’єктів, таких як люди, автомобілі, тварини чи елементи інфраструктури, і активно використовуються в сучасних системах обробки зображень та відео. Значний прогрес у цій галузі став можливим завдяки розвитку підходів глибокого навчання та появі потужних нейронних мереж.

Практичне впровадження алгоритмів виявлення об’єктів охоплює такі завдання, як розпізнавання облич, детекція пішоходів і транспортних засобів, ідентифікація дорожніх знаків, системи відеоспостереження та інтелектуальний моніторинг.

Традиційний конвеєр моделей детекції об’єктів зазвичай складається з трьох основних етапів: визначення області інтересу, виділення інформативних ознак та подальшої класифікації. З розвитком глибоких нейронних мереж з’явилася значна кількість ефективних архітектур, які використовуються як у наукових дослідженнях, так і в промислових системах. Серед найбільш поширених підходів можна виділити Fast R-CNN, Faster R-CNN, MobileNet, YOLO, Mask R-CNN, RetinaNet та методи на основі гістограм орієнтованих градієнтів (HOG).

Алгоритм Fast R-CNN, реалізований із використанням мов програмування Python і C++ (Caffe), є удосконаленим варіантом регіонно-орієнтованих згорткових мереж. Він усуває ключові обмеження попередніх підходів, зокрема R-CNN і SPP-net, та забезпечує підвищену швидкість обробки разом із покращеною точністю виявлення.

Подальшим розвитком цього підходу є Faster R-CNN, у якому використовується мережа пропозицій регіонів (Region Proposal Network, RPN). Дана мережа спільно використовує згорткові ознаки з основним детектором і дозволяє ефективно генерувати області-кандидати для виявлення об’єктів, що суттєво знижує обчислювальні витрати.

Метод HOG ґрунтується на аналізі орієнтацій градієнтів у локальних ділянках зображення та використовується як дескриптор ознак. Його перевагою є простота реалізації та наочність інформації, яку він кодує, що робить даний підхід зручним для пояснюваних моделей.

Метод R-CNN поєднує механізм пропозицій регіонів із глибокими згортковими нейронними мережами, забезпечуючи високу точність локалізації об’єктів навіть за обмеженої кількості анотованих даних. Цей підхід здатний масштабуватися на велику кількість класів без використання спрощених евристик.

Мережі типу R-FCN відрізняються повністю згортковою архітектурою, що дозволяє виконувати більшість обчислень для всього зображення одночасно. Такий підхід зменшує обчислювальну складність і підвищує швидкодію системи виявлення.

Алгоритм Single Shot Detector (SSD) реалізує детекцію об’єктів за допомогою єдиної нейронної мережі без етапу генерації пропозицій. Завдяки використанню кількох карт ознак з різною роздільною здатністю SSD ефективно виявляє об’єкти різного масштабу, поєднуючи високу швидкість роботи з прийнятною точністю.

Метод просторового пірамідального об’єднання (SPP-net) дозволяє формувати ознакові представлення фіксованої довжини незалежно від розміру зображення. Це забезпечує стійкість до масштабних змін і дозволяє уникнути багаторазового обчислення згорткових ознак.

Одним із найпопулярніших алгоритмів детекції об’єктів є YOLO, який реалізує уніфікований підхід до локалізації та класифікації об’єктів. Його ключовою перевагою є висока швидкість обробки, що дозволяє використовувати алгоритм у системах реального часу.

Незважаючи на значні досягнення у контрольованих умовах, задача виявлення об’єктів залишається складною у неконтрольованих середовищах, де присутні складні фони, перекриття об’єктів і довільні пози. Проте активний розвиток нейронних мереж, зростання рівня автоматизації та зниження вартості апаратних рішень сприяють широкому впровадженню таких технологій у різних галузях, що свідчить про перспективність подальших досліджень у цьому напрямку.

1.1 Алгоритми класифікації об’єктів на зображенні

Класифікація та розпізнавання зображень є ключовими складовими сучасних інтелектуальних систем і лежать в основі більшості значущих досягнень у сфері штучного інтелекту. Саме завдяки цим підходам стає можливим автоматичне

інтерпретування візуальної інформації, що використовується в широкому спектрі прикладних задач — від медичної діагностики до автономного керування технічними системами.

До методів контрольованої класифікації зображень належать такі класичні підходи, як метод паралелепіпеда, класифікатор мінімальної відстані, байєсівський класифікатор максимальної правдоподібності та інші алгоритми, що базуються на попередньо розмічених навчальних вибірках. Ці методи передбачають наявність еталонних даних, на основі яких формується правило прийняття рішень щодо належності пікселів або об’єктів до певних класів.

Загалом сучасні методи класифікації зображень можна систематизувати за кількома ознаками, зокрема:

класифікація зображень із використанням інформації, отриманої від різних типів сенсорів і датчиків;

класифікація, що базується на характеристиках та структурі навчальної вибірки, яка застосовується для побудови моделі;

класифікація зображень залежно від набору параметрів і характеристик вхідних даних;

класифікація з урахуванням природи піксельної інформації, яка міститься у цифровому зображенні;

класифікація на основі кількості та типу вихідних результатів, що формуються для кожного просторового елемента;

класифікація зображень із використанням просторових взаємозв’язків між пікселями та об’єктами.

Узагальнюючи наведене, методи класифікації зображень доцільно поділяти на дві основні групи, що відрізняються принципами обробки даних і підходами до формування рішень, що схематично подано на рис. 2.3.



Рис. 2.3 Групи алгоритмів класифікації об’єктів

Піксель є найменшим структурним елементом цифрового зображення, а аналіз сукупності пікселів становить основу більшості підходів до автоматичної класифікації зображень. При цьому алгоритми обробки візуальних даних можуть спиратися виключно на спектральні характеристики окремих пікселів або ж додатково враховувати просторові взаємозв’язки між сусідніми пікселями. У першому випадку класифікація ґрунтується лише на значеннях інтенсивності, тоді як у другому — поєднує спектральну інформацію з геометричним розташуванням елементів зображення.

Піксельно-орієнтовані методи класифікації використовують лише числові характеристики кожного окремого пікселя. До найпоширеніших алгоритмів цього класу належать метод мінімальної відстані до середнього, байєсівський класифікатор максимальної правдоподібності та метод мінімальної відстані Махаланобіса. Їх спільною особливістю є необхідність попереднього визначення статистичних параметрів класів, зокрема середніх значень і дисперсій, а процес прийняття рішення базується на оцінюванні відстані між характеристиками пікселя та відповідними класовими центрами.

Основним недоліком піксельної класифікації є ігнорування контекстної інформації, оскільки сусідні пікселі не залучаються до аналізу. На відміну від цього, об’єктно-орієнтовані методи працюють із групами суміжних пікселів, формуючи регіони або сегменти, що дозволяє одночасно використовувати як спектральні, так і просторові ознаки. Слід зауважити, що під поняттям «об’єкт» у

цьому контексті мається на увазі зв’язна область пікселів, а не обов’язково реальний фізичний об’єкт сцени.

Сучасні високоточні системи аналізу зображень переважно застосовують класифікацію на рівні об’єктів, що вимагає спеціальної підготовки вхідних даних. Перед безпосередньою класифікацією необхідно виконати виділення регіонів інтересу та здійснити їх попередню обробку. Це дозволяє перетворити вихідне зображення у форму, придатну для подальшого аналізу алгоритмами машинного навчання.

Оскільки цифрові зображення представлені у вигляді числових матриць, вони потребують інтерпретації та нормалізації перед подачею в модель. Одним із ключових етапів підготовки даних є виявлення об’єктів, яке забезпечує визначення інформативних ділянок зображення та значно підвищує ефективність навчання класифікатора.

Методи попередньої обробки залежать від кількості об’єктів, що підлягають аналізу. У випадку наявності одного об’єкта застосовується локалізація, під час якої формується обмежувальна рамка навколо цільової області. Така рамка дозволяє алгоритму зосередити обчислювальні ресурси лише на значущій частині зображення. Якщо ж у сцені присутні кілька об’єктів, використовується процедура детекції об’єктів, яка забезпечує побудову обмежувальних рамок для кожного з них.

Альтернативним підходом до попередньої обробки є сегментація зображень. Вона полягає у поділі зображення на однорідні області на основі подібності піксельних характеристик. У результаті формуються маски, що точно відображають форму та межі об’єктів. Порівняно з обмежувальними рамками, сегментація забезпечує більш детальну розмітку на рівні окремих пікселів, що особливо важливо для задач високоточної класифікації.

Після завершення етапів детекції або сегментації кожному виділеному регіону присвоюються відповідні мітки. Разом із числовими ознаками пікселів ці дані використовуються для навчання моделей машинного навчання, які виявляють характерні закономірності, притаманні кожному класу.

Після підготовки та анотації даних вони подаються до алгоритмів машинного навчання, що здійснюють безпосередню класифікацію зображень. Одним із найбільш відомих підходів є алгоритм K-Nearest Neighbours (KNN), який приймає рішення на основі аналізу найближчих до тестового зразка навчальних прикладів. У задачах класифікації зображень для цього алгоритму зберігаються вектори ознак навчальних зразків разом із їх мітками, а під час тестування обчислюється міра подібності між новим зображенням і вже відомими прикладами.

Алгоритм KNN відзначається простотою реалізації та водночас достатньо високою ефективністю, що зумовлює його широке застосування в практичних задачах машинного навчання. Для наочного пояснення принципу його роботи доцільно розглянути приклад розподілу точок у двовимірному просторі (рис. 2.4), де об’єкти формують компактні кластери. Основне припущення цього методу полягає в тому, що зразки, розташовані близько один до одного у просторі ознак, мають схожі властивості, тоді як віддалені точки, як правило, належать до різних класів. Виходячи з цього припущення, алгоритм виконує подальші обчислення та формує результат класифікації.

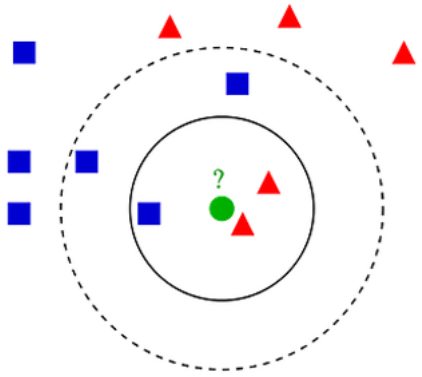


Рис. 2.4 Схема роботи алгоритму K-Nearest Neighbours

Алгоритм K-Nearest Neighbours визначає ступінь подібності між об’єктами шляхом оцінювання відстані між відповідними точками у просторі ознак. Чим

меншою є відстань між двома точками, тим більш схожими вважаються відповідні їм об’єкти. Для кількісної оцінки цієї подібності можуть застосовуватися різні метрики відстані, однак на практиці найчастіше використовується евклідова відстань, яка характеризує найкоротший шлях між двома точками у багатовимірному просторі.

Алгоритм KNN належить до методів навчання з учителем, що означає наявність заздалегідь відомих міток класів для всіх елементів навчальної вибірки. Водночас KNN має низку принципових особливостей. По-перше, він є непараметричним методом, тобто не ґрунтується на жодних апріорних припущеннях щодо статистичного розподілу даних. Уся модель формується безпосередньо на основі наявного набору прикладів. По-друге, на відміну від багатьох інших алгоритмів машинного навчання, KNN не виконує явного поділу даних на навчальну та тестову вибірки. Усі доступні приклади зберігаються та використовуються під час формування прогнозу для нового об’єкта.

Процес роботи алгоритму KNN можна представити у вигляді послідовності таких основних етапів:

Вибір параметра K

K, який визначає кількість найближчих сусідів, що враховуються при прийнятті рішення.

Обчислення відстаней між тестовим зразком та всіма елементами наявного набору даних.

Упорядкування отриманих значень відстаней у зростаючому порядку.

Вибір K

K об’єктів з мінімальними відстанями та визначення їх міток.

Формування остаточного прогнозу для тестового прикладу.

На першому етапі користувач задає значення параметра K

K, яке безпосередньо впливає на чутливість алгоритму до локальних змін у даних. Далі модель обчислює відстань між цільовим прикладом і кожним елементом набору даних, після чого результати зберігаються та сортуються. Із відсортованого списку відбираються K

K найближчих сусідів, мітки яких використовуються для прийняття рішення.

Характер формування результату залежить від типу задачі. У випадку регресії прогнозоване значення визначається як середнє арифметичне значень цільової змінної для вибраних K

K сусідів. Для задач класифікації застосовується принцип більшості, тобто тестовому зразку присвоюється той клас, який найчастіше зустрічається серед K

K найближчих елементів набору даних. Наочна ілюстрація цього підходу подана на рис. 2.5.

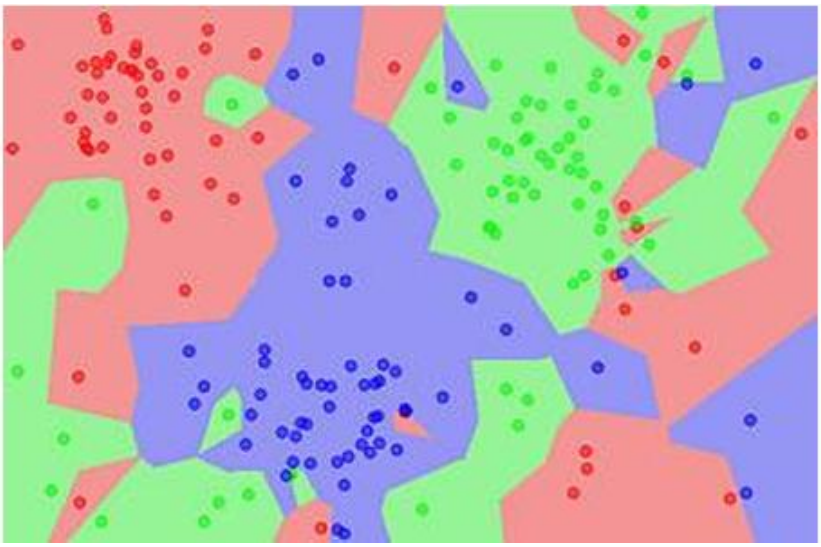


Рис. 2.5. Приклад результату роботи алгоритму K-Nearest Neighbours

Конкретні математичні обчислення, що застосовуються в алгоритмі KNN, визначаються обраною метрикою відстані між об’єктами у просторі ознак. Залежно від характеру даних та поставленої задачі можуть використовуватися різні функції відстані. Серед найбільш поширених слід відзначити евклідову метрику, манхеттенську відстань, а також узагальнену метрику Мінковського, кожна з яких по-різному оцінює міру близькості між векторами ознак.

Однією з ключових проблем при використанні алгоритму KNN є вибір параметра K

K, що визначає кількість найближчих сусідів, які беруть участь у формуванні рішення. Невдале значення цього параметра може призвести до значного зниження точності прогнозування. Тому під час застосування KNN особливу увагу приділяють пошуку такого значення K

K, яке забезпечує оптимальний баланс між здатністю моделі до узагальнення та мінімізацією кількості помилкових рішень на нових, раніше не бачених даних.

Малі значення параметра K

K роблять модель надто чутливою до локальних шумів у даних, що негативно впливає на стабільність результатів. Для ілюстрації цього ефекту можна розглянути ситуацію, коли навколо цільової точки розташовано сім сусідніх зразків. Якщо більшість із них належить до одного класу, наприклад Blue, але два найближчі елементи мають мітку Red, то при значенні K=2

K=2 алгоритм зробить висновок на користь класу Red. У такому випадку рішення не відображатиме загальної структури локального оточення, що знижує коректність класифікації.

Разом із тим, вибір надто великого значення K

K також є небажаним. Зі збільшенням кількості врахованих сусідів алгоритм починає охоплювати зразки, що належать до інших кластерів, розташованих далі від цільового об’єкта. Це призводить до розмивання меж між класами та підвищення ймовірності помилкової класифікації. Таким чином, під час практичного використання KNN зазвичай проводиться експериментальний підбір параметра K

K з метою досягнення найкращої продуктивності моделі.

До основних переваг алгоритму KNN можна віднести такі характеристики:

алгоритм придатний як для задач класифікації, так і для задач регресії, що робить його універсальним інструментом аналізу даних;

простота концепції та реалізації забезпечує легкість інтерпретації результатів і швидке впровадження;

відсутність припущень щодо розподілу даних дозволяє застосовувати KNN до широкого спектра практичних задач.

Водночас алгоритм має і суттєві недоліки:

необхідність зберігати весь або майже весь набір даних призводить до високих вимог до пам’яті та значних обчислювальних витрат, особливо для великих вибірок;

чутливість до масштабу ознак і наявності нерелевантних характеристик може негативно впливати на якість класифікації без попередньої нормалізації та відбору ознак.

Незважаючи на простоту побудови, K-Nearest Neighbours залишається ефективним алгоритмом машинного навчання, який у багатьох практичних випадках демонструє високу точність. Для досягнення найкращих результатів доцільно тестувати різні значення параметра K

K та обирати те, що забезпечує оптимальні показники якості.

Алгоритми класифікації на основі KNN легко масштабуються на задачі з кількома класами та не потребують складної попередньої підготовки моделі. Проте рівнозначне врахування всіх ознак може стати джерелом помилок у ситуаціях, коли лише частина характеристик є інформативною для класифікації зображень.

Машини опорних векторів (SVM) належать до методів класифікації, що реалізують розділення даних у просторі ознак за допомогою оптимальних гіперплощин. Кожен об’єкт відноситься до певного класу залежно від свого положення відносно розділової межі. Завдяки використанню ядерних функцій SVM здатні виконувати нелінійну класифікацію. Водночас суттєвим обмеженням цього підходу є зниження швидкодії та зростання обчислювальної складності при роботі з великими наборами даних.

Багатошарові персептрони, які є різновидом штучних нейронних мереж, побудовані за аналогією з біологічними нейронними структурами. Вони складаються з послідовно з’єднаних шарів нейронів, що дозволяє моделювати складні нелінійні залежності між вхідними даними та вихідними класами. У

процесі навчання параметри мережі поступово коригуються з метою зменшення похибки. Попри високу точність, такі моделі можуть мати проблеми, пов’язані з неопуклістю функцій втрат та складністю оптимізації.

На сьогодні провідне місце в задачах класифікації зображень займають згорткові нейронні мережі (CNN). Вони є спеціалізованим різновидом нейронних мереж, які поєднують багатошарову архітектуру з згортковими та підвибірковими шарами. Це дозволяє автоматично виділяти найбільш інформативні ознаки зображень та значно підвищувати точність класифікації, що наочно продемонстровано на рис. 2.6.

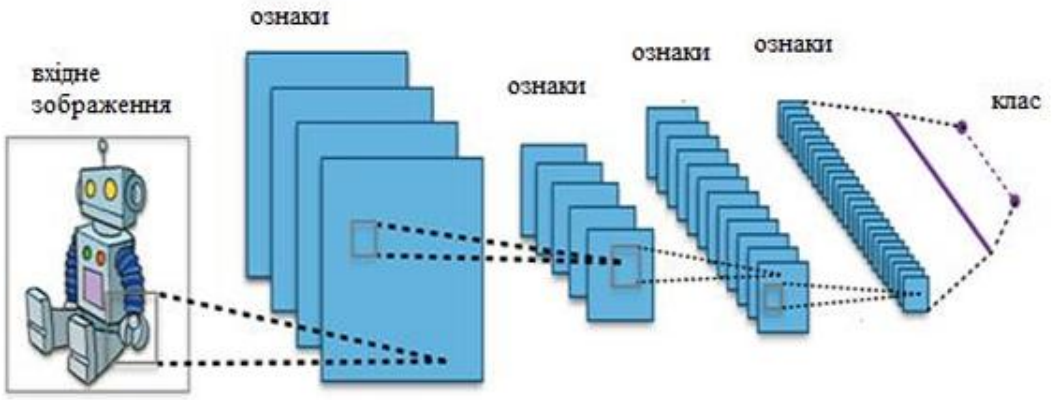


Рис. 2.6 Схема роботи згорткових нейронних мереж

Згорткові нейронні мережі (CNN) здатні автоматично виявляти та формувати ознаки зображень, а також навчатися на них без необхідності ручного маркування або сегментації даних перед застосуванням алгоритмів машинного навчання. Така особливість значно знижує трудомісткість підготовки даних та робить CNN ефективнішими порівняно з багатошаровими персептронами (MLP), особливо при роботі з неопуклими функціями втрат.

Назва «згорткова нейронна мережа» походить від операції згортки, яка є ключовим механізмом виділення ознак. Під час роботи CNN фільтр ковзає по зображенню, подібно до рухомого вікна, що аналізує локальні ділянки. Кожен фільтр містить числові коефіцієнти, які множаться на значення пікселів у відповідній ділянці зображення. Результатом цієї операції є нова матриця чисел,

яка відображає виявлені ознаки. Цей процес повторюється для кількох фільтрів, після чого отримані карти ознак об’єднуються в більш компактне та спрощене представлення зображення. Додатково застосовується операція об’єднання (pooling), яка дозволяє залишати лише найбільш значущі характеристики, сприяючи виділенню ключових елементів для подальшої класифікації.

Архітектура CNN складається з кількох типів шарів, що виконують різні функції. Згорткові шари відповідають за виділення ознак та перетворення їх у форму, придатну для обробки наступними шарами мережі. Ранні шари здатні розпізнавати базові елементи зображення, такі як прості лінії, контури та межі. Середні шари захоплюють більш складні структури, наприклад криві та кути, тоді як глибші шари витягують високорівневі характеристики, що формують абстрактні представлення об’єктів. Отримані ознаки передаються в повнозв’язні шари мережі, де відбувається навчання класифікатора та формування остаточного прогнозу.

1.2 Алгоритм виділення рухомих об’єків на основі ковзного вікна

Під час розробки алгоритму детекції окремих об’єктів у відеопотоці на основі методу ковзного вікна були враховані переваги та обмеження існуючих алгоритмів, що використовуються у сучасних системах відеоспостереження та цифрових бібліотеках функцій. Оскільки алгоритм спроектовано для роботи в режимі реального часу, особлива увага приділялася мінімізації кількості обчислювальних операцій для забезпечення швидкого отримання результатів.

Однією з ключових проблем, яку необхідно було вирішити, є варіативність освітлення при обробці відеопотоку, включаючи можливість появи або відсутності сторонніх джерел світла. Ця особливість може призводити до значних шумів на кадрах і впливати на точність детекції об’єктів.

Ще одним критичним фактором є роздільна здатність камер спостереження. При низькій роздільній здатності об’єкти займають меншу кількість пікселів, що, з одного боку, зменшує обчислювальні витрати, але з іншого може погіршувати

точність алгоритму. Висока роздільна здатність навпаки підвищує точність, проте значно збільшує обчислювальні та логічні витрати.

Для оптимізації роботи алгоритму було вирішено аналізувати не все зображення цілком, а лише його ключові ділянки, зокрема краї кадру, де найчастіше з’являються об’єкти інтересу. Такий підхід дозволяє зменшити обсяг обчислень та прискорити обробку відеопотоку.

Блок-схема роботи алгоритму детекції на основі ковзного вікна наведена на рис. 2.7.

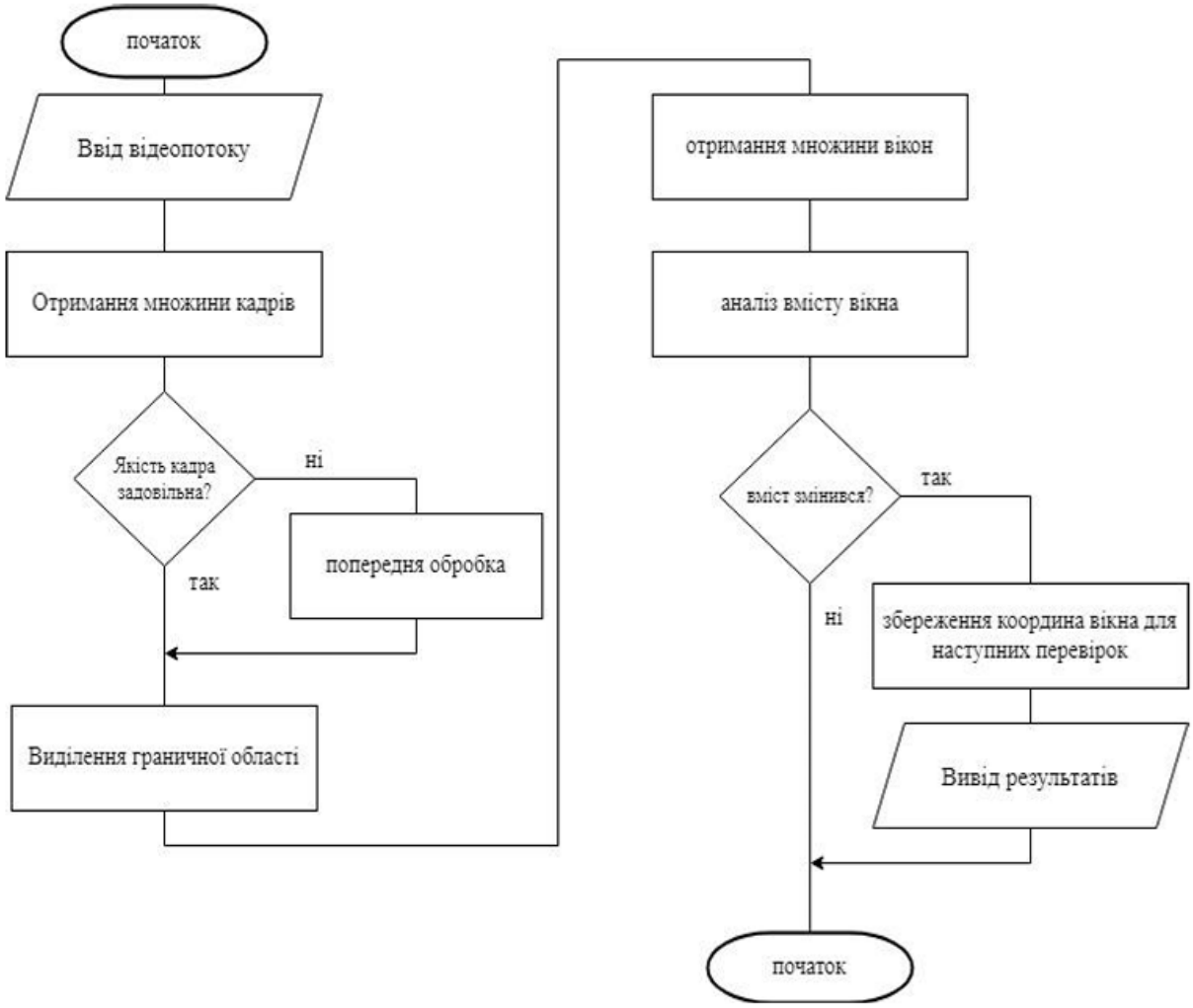


Рис. 2.7 Блок-схема алгоритму перетворення елементів на зображенні

Основна ідея запропонованого алгоритму полягає в тому, що при спостереженні певної області рухомий об’єкт повинен спочатку з’явитися на її

межах. Таке припущення дозволяє значно зменшити кількість перевірок, оскільки більша частина центральної області не аналізується. Центральна частина кадру перевіряється лише у випадку, якщо об’єкт було виявлено на периферії, після чого здійснюється його супровід у межах аналізованої області.

Алгоритм включає наступні кроки:

Активація програми та встановлення параметрів роботи, зокрема розмірів граничної області спостереження та розмірів ковзного вікна.

Отримання першого кадру відеопотоку.

Розбиття граничної області першого кадру на окремі вікна заданого розміру та формування масиву спостереження.

Отримання наступного кадру відеопотоку та розбиття його граничної області на окремі вікна.

Порівняння значень вікон поточного кадру зі значеннями вікон масиву спостереження, які мають відповідні координати.

Якщо вікно не змінилося, перехід до кроку 8.

Якщо зміни виявлено (понад 20% пікселів у вікні перевищують поріг T), вікно заноситься до масиву потенційних виявлень.

Перевірка вікон з масиву потенційних виявлень для мінімізації точок, що залишилися незмінними, та формування остаточного вікна спостереження за об’єктом.

Перевірка сусідніх областей з масиву потенційних виявлень із збільшенням розміру вікна на 25% для продовження супроводу руху об’єкта.

Фіксація інформації про виявлені об’єкти та напрям їх руху.

Оновлення масиву спостереження шляхом заміни значень вікон поточного кадру.

Якщо спостереження завершено – вихід з алгоритму; інакше повернення до кроку 4.

Використання принципу аналізу лише граничних областей зображення дозволяє зменшити кількість обчислень, особливо у випадках, коли рух на області спостереження відсутній або кількість рухомих об’єктів мінімальна.

До основних переваг алгоритму належать:

висока швидкість роботи за рахунок зменшення кількості перевірок;

низькі вимоги до апаратного забезпечення;

простота та зрозумілість реалізації;

можливість автономної роботи без втручання людини.

Недоліки алгоритму:

ефективність детекції залежить від початкових параметрів розміру вікна та частоти оновлення кадрів;

проблеми з детекцією при дуже високій або дуже низькій швидкості руху об’єкта.

Результати моделювання показали високу швидкість роботи алгоритму та точність детекції рухомих об’єктів. Проведено аналітичний огляд методів виділення та класифікації об’єктів на цифрових зображеннях, що дозволило розробити алгоритм детекції рухомих об’єктів у відеопотоці. Запропонований підхід, заснований на принципах ковзного вікна для граничних областей кадру, дозволив здійснити структурне проектування та реалізацію програмної системи детекції рухомих об’єктів на відеопотоках.

2 ПРОГРАМНА СИСТЕМА ДЕТЕКЦІЇ ОБ'ЄКТІВ НА ЦИФРОВИХ ЗОБРАЖЕННЯХ

2.1 Структура програмної детекції рухомих об'єктів

Наприклад, домашній робот легко може навчитися розпізнавати кавову машину на порожньому столі. Але завдання значно ускладнюється, якщо машина розташована на кухонній плиті, захаращеній посудом, електронікою та іншими предметами. У таких сценаріях процес пошуку або розпізнавання об'єкта стає надзвичайно складним, і на сьогодні ефективного універсального рішення для цієї проблеми ще не знайдено.

Більшість методів виявлення та розпізнавання об'єктів можна розділити на дві категорії залежно від типу ознаки, яку вони використовують:

Методи на основі країв.

Методи на основі міток.

Іноді дослідники застосовують комбінований підхід, використовуючи одночасно ознаки країв і міток. Експериментальні результати показують, що поєднання цих двох підходів забезпечує переваги обох типів ознак, що робить його перспективним для майбутніх систем.

Методи на основі країв працюють шляхом виділення карти країв зображення та ідентифікації особливостей об'єкта за межами його контурів. Використання країв як ознаки має низку переваг: вони здебільшого інваріантні до змін освітлення, кольору та текстури об'єкта, а також ефективно відображають його межі, представляючи дані у високорозмірному просторі зображень.

Спрощена структура програмної системи представлена на рис. 3.1.

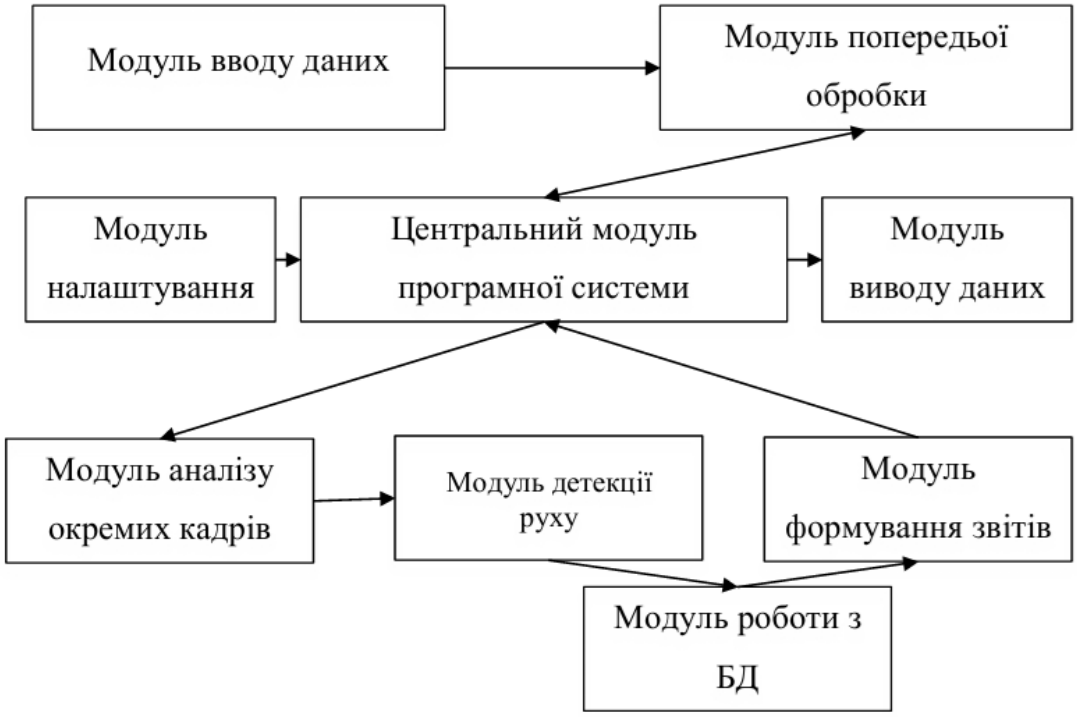


Рис. 3.1 Узагальнена структура програмного додатку детекції руху на відеопотоках

Серед головних структурних частин виділяються такі:

Центральний модуль програмної системи – це ядро додатку, яке відповідає за підтримку цілісності та стабільності роботи системи у різних умовах (рис. 3.2). У цьому модулі реалізовано такі функції:

перевірка цілісності програмного додатку;

контроль наявності актуальних оновлень модулів та їх версій;

відключення непотрібних або малоінформативних компонентів системи;

конвертація та перевірка коректності повідомлень як всередині системи, так і під час обміну даними із зовнішніми структурами.

Вхідні дані модуля формуються у вигляді вектора параметрів роботи програми, тоді як вихідні дані включають:

корекційні пакети обміну даними;

повідомлення для користувача щодо необхідності внесення змін у роботу програмного додатку;

сигнали для оптимізації або корекції роботи системи у цілому.

Цей підхід дозволяє забезпечити ефективну інтеграцію алгоритмів детекції руху у загальну архітектуру системи, зберігаючи при цьому стабільність та надійність функціонування програмного додатку.

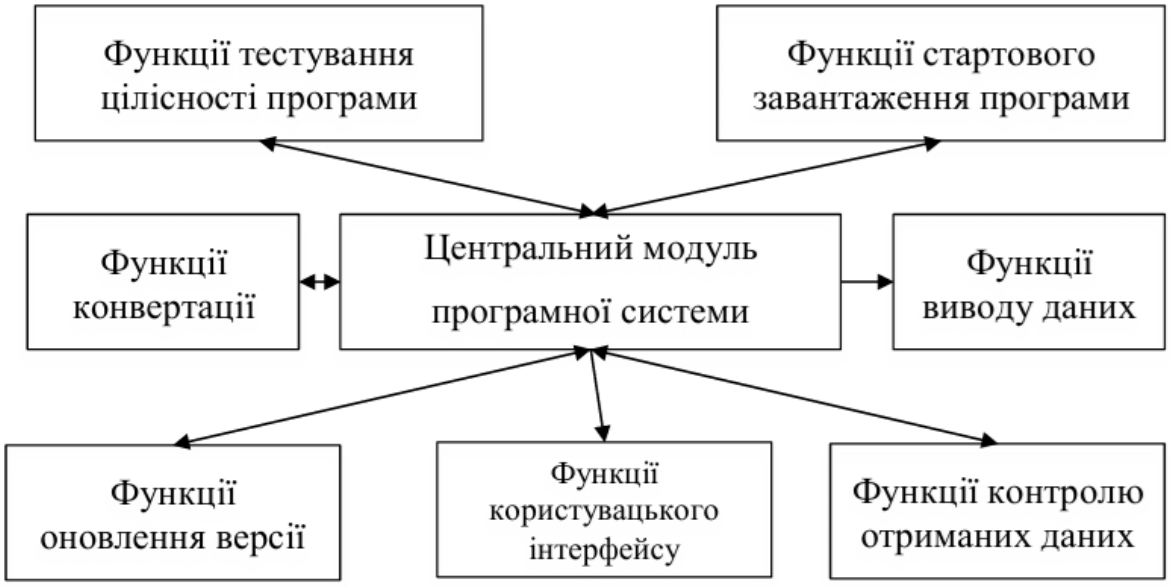


Рис. 3.2 Узагальнена структура центрального модуля програмної системи

Модулі програмного додатку

Розроблена програмна система побудована за модульним принципом, що забезпечує гнучкість, зручність у підтримці та можливість подальшого розширення функціоналу. Основні модулі системи представлені нижче.

1. Модуль завантаження відео

Модуль призначений для організації інтерфейсу між апаратними засобами, що генерують або зберігають відеопотік. Його пріоритет не є високим, оскільки у випадку відсутності зовнішніх джерел дані можуть опрацьовуватися з внутрішнього сховища системи.

Вхідні дані: набір параметрів для підключення до зовнішніх джерел відеопотоку.

Вихідні дані: відеофайл для подальшого аналізу.

2. Модуль попередньої обробки

Цей модуль є одним із центральних компонентів системи, оскільки результати первинної обробки відеоданих значною мірою визначають швидкість обробки файлів та точність кінцевих результатів. У модулі реалізовано:

підвищення якості зображень;

попередню оцінку та підготовку даних для наступних етапів аналізу;

перетворення зображень для спрощення обробки алгоритмами детекції.

Більшість функцій модулю реалізовані за допомогою зовнішніх цифрових бібліотек, оскільки розробка індивідуальних алгоритмів попередньої обробки не входила до завдань даного етапу.

вихідні дані: встановлення основних параметрів функціонування програмного додатку.

3. Модуль налаштування

Цей допоміжний модуль підвищує зручність роботи користувачів та дозволяє обирати стартові параметри роботи системи залежно від досвіду користувача та завдань. Можливі налаштування включають:

тип та розмір шрифтів інтерфейсу;

параметри модулів виводу звітів;

підключення та налаштування фото/відеокамер;

параметри роботи алгоритмів детекції руху (розмір ковзного вікна, частота оновлення кадрів).

4. Модуль детекції руху

Цей модуль є одним із головних у системі та реалізує алгоритми детекції руху об’єктів у відеопотоці.

Функції: аналіз кадрів відеопотоку, формування двох масивів даних:

масив спостереження (вікна з попередніх етапів аналізу кадрів);

масив рухомих цілей (вікна, у яких зафіксовано зміни).

Результат: повідомлення користувачу про виявлення руху на відео.

Коректна робота модуля залежить від взаємодії з іншими компонентами системи, однак модуль здатний виконувати автономну обробку даних, хоча це може збільшувати час обробки та кількість хибних спрацювань.

5. Модуль роботи з базою даних (БД)

Допоміжний модуль, що реалізує взаємодію з сервером бази даних для збереження робочих масивів вікон. Результати зберігаються тимчасово і видаляються після завершення роботи програми.

6. Модуль створення звітів

Додатковий модуль, який дозволяє користувачам обирати формат та спосіб виводу результатів роботи системи, а також відстежувати процес детекції руху. Відеопослідовності та результати обробки формуються на основі параметрів, встановлених користувачем, і можуть коригуватися під час виводу.

Переваги модульної структури:

Гнучкість та універсальність системи;

Можливість збереження працездатності навіть при пошкодженні окремих модулів;

Легкість інтеграції нових компонентів або модифікації наявних без порушення цілісності системи.

Моделювання структури системи

Перед реалізацією алгоритмів проведено моделювання системи та алгоритму на основі відомих програмних пакетів UML. Недоліки використання універсальної мови моделювання:

Часові витрати: створення та синхронізація діаграм займає значний час;

Можливі втрати інформації: неможливо відобразити всі умови та варіанти результатів;

Фінансові витрати: якісне програмне забезпечення UML потребує коштів та часу на освоєння.

Моделювання на основі діаграм прецедентів дозволяє оцінити взаємодію модулів та визначити важливість окремих функцій у загальній структурі проекту.

Діаграми також відображають можливості користувача при взаємодії з системою (рис. 3.3)

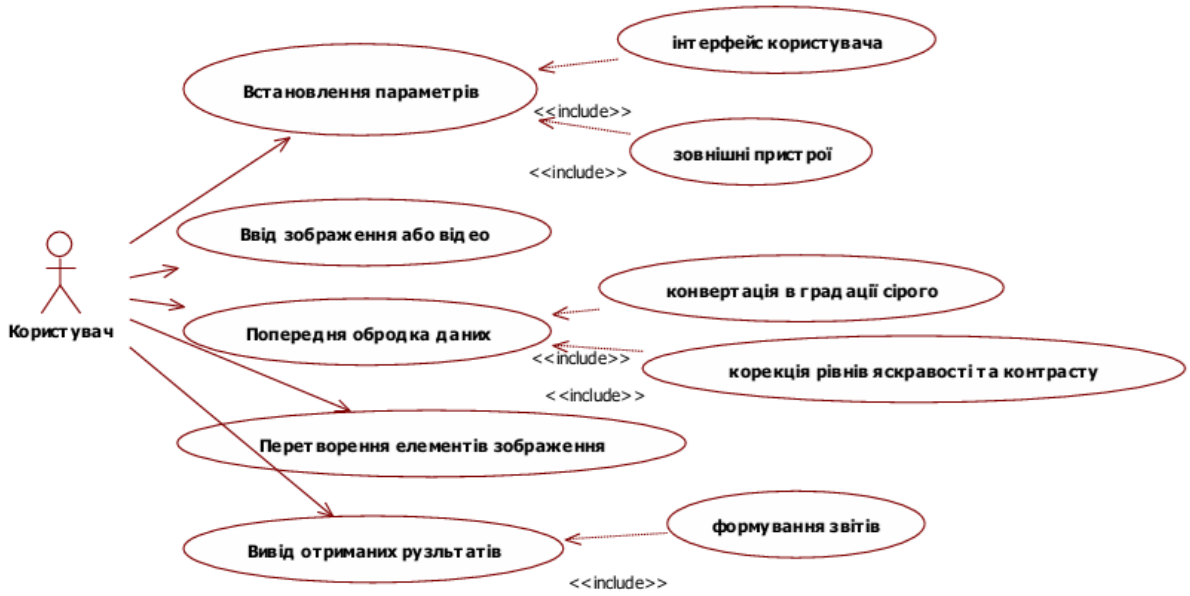


Рис. 3.3 Діаграма прецедентів програмної системи

Наведена діаграма ілюструє результати етапу аналізу рівня взаємодії окремих функцій між модулями програмного додатку, а також демонструє можливості користувача при роботі з системою. Дії користувача обмежені зовнішнім впливом на систему, водночас внутрішні механізми захищені від прямого доступу. Таке моделювання підтвердило правильність обраної структури та набору реалізованих функцій.

Високий рівень автоматизації роботи програмного додатку забезпечує можливість його використання навіть користувачами без досвіду роботи. Процес опанування програмою є коротким і не вимагає від нових користувачів глибоких попередніх знань у сфері комп’ютерної техніки.

На наступному етапі моделювання здійснювався аналіз коректності виконання кроків алгоритму для досягнення необхідних результатів. Було оцінено взаємодію процесів та об’єктів, що відбуваються під час роботи програми, а також визначено порядок виконання основних дій користувача та системи.

Результат цього етапу моделювання наведено на рис. 3.4, що відображає послідовність дій, взаємозв’язки між процесами та контрольні точки для забезпечення коректної роботи системи.

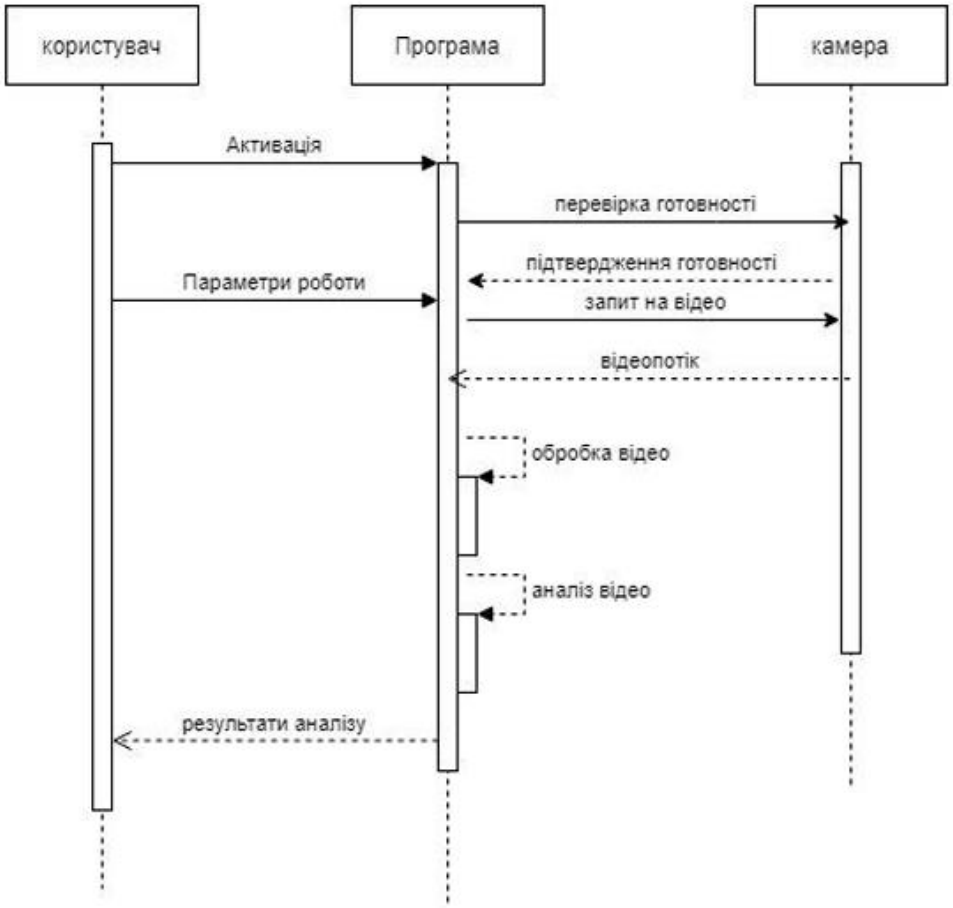


Рис. 3.4 Діаграма послідовності програмної системи

На етапі моделювання було проаналізовано основні фактори, які можуть впливати на працездатність системи, а також функціональні можливості, необхідні для користувача для досягнення результату.

Графічний інтерфейс користувача розроблено з урахуванням сучасних вимог: застосовано принцип мінімалізму, при якому користувач отримує мінімальну кількість активних елементів керування, що забезпечує простоту використання для початківців. Для досвідчених користувачів програмна система

надає можливості більш глибокого контролю процесів, при цьому відповідальність за коректну роботу системи покладається на користувача.

Для роботи з програмою користувачеві необхідно виконати просту та послідовну низку кроків. Цей набір кроків супроводжується підказками під час запуску програмного додатку та може бути відображений під час роботи програми у вигляді інтерактивних підказок. Послідовність кроків візуально представлено на рис. 3.5.

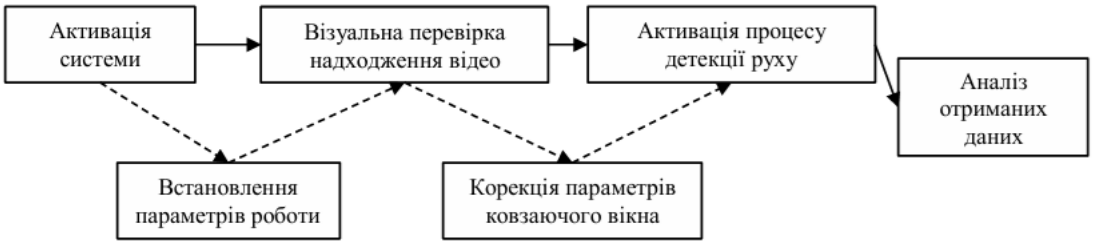


Рис. 3.5 Послідовність дій користувача при роботі з системою відеоспостереження

2.2 Програмні модулі системи перетворення елементів зображення

Інтернету, включаючи API для:

виявлення об’єктів;

визначення розташування об’єктів;

обробки відео;

відстеження об’єктів.

Бібліотека здатна виявляти та розпізнавати до 80 різних видів звичайних об’єктів, використовуючи попередньо навчені моделі, які можна легко налаштовувати.

Серед нових можливостей ImageAI слід відзначити:

інтеграцію моделей SqueezeNet, ResNet50, InceptionV3 і DenseNet121 для користувацького навчання та прогнозування зображень;

додавання спеціальної моделі навчання та файлу JSON для імпорту і експорту користувацьких зображень;

попередній перегляд з визначенням відеооб’єкта та відстеженням користувацьких об’єктів;

підтримку всіх завдань прогнозування зображень і виявлення об’єктів, включаючи роботу з масивами NumPy;

регулювання частоти виявлення кадрів у відео (frame_detection_interval), що дозволяє оптимізувати продуктивність при обробці відеопотоку.

```
import cv2, os
import numpy as np
import matplotlib.pyplot as plt

OBJ_THRESH = .6
P_THRESH = .6
NMS_THRESH = .5

np.random.seed(999)
```

Рис. 3.6 Приклад встановлення початкових параметрів роботи програми

Для реалізації розробленої програмної системи на Python необхідно підключити низку бібліотек, що забезпечують коректну обробку зображень та відео, чисельні обчислення та візуалізацію даних.

OpenCV

Першим кроком є підключення бібліотеки OpenCV, яка містить широкий спектр функцій для обробки цифрових зображень та відео. Підключення відбувається командою:

import cv2

При інтеграції з бібліотекою NumPy, яка забезпечує оптимізовану роботу з багатовимірними масивами, можна об’єднувати обчислювальні функції обох бібліотек. OpenCV також включає каскадні файли Haar, що використовуються для розпізнавання об’єктів, доступ до яких можна отримати через cv2.data.haarcascades.

Модуль os

Модуль os у Python надає функції для взаємодії з операційною системою. Він забезпечує портативний доступ до функцій файлової системи та інших служб OS. Модулі os та os.path дозволяють працювати з робочим каталогом (CWD – Current Working Directory), що визначає папку, у якій Python виконує операції з файлами.

NumPy

Оскільки програмний додаток обробляє великі масиви числової інформації, підключається бібліотека NumPy (Numerical Python). Вона забезпечує роботу з багатовимірними масивами та надає широкий набір математичних і логічних операцій для обробки даних.

Matplotlib

Для візуалізації даних використовується бібліотека Matplotlib. Це кросплатформна бібліотека для створення двовимірних графіків з даних у масивах. Вона використовує NumPy для обчислень і надає об’єктно-орієнтований API для інтеграції графіків у GUI-додатки на Python (PyQt, WxPython, Tkinter), а також для роботи в Jupyter Notebook чи веб-серверах. Matplotlib має процедурний інтерфейс Pylab, який схожий на MATLAB і дозволяє швидко створювати графічні представлення даних.

Попередня обробка зображень

Для оптимізації швидкодії системи реалізовано методи попередньої обробки зображень. Оскільки алгоритм детекції руху повинен швидко оцінювати наявність змін у кадрі, проводиться перетворення кольорового зображення у монохромне. Це значно зменшує обсяг обчислень, скорочує час обробки відеопотоку та забезпечує більш швидке виконання алгоритму (рис. 3.7).

```python
import cv2

image = cv2.imread('C:\\test.jpg')
cv2.imshow('Original', image)
cv2.waitKey(0)

gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

cv2.imshow('Grayscale', gray_image)
cv2.waitKey(0)

cv2.destroyAllWindows()
```

Рис. 3.7. Програмний код перекодування зображення з кольорового в монохромне

Приклад результатів роботи фуфнкції переконвертації зображення з кольорового в монохромне наведено на рис. 3.8.



а) Кольорове зображення



б) Монохромне

Рис. 3.8. Програмний код перекодування зображення з кольорового в монохромне

```python
import cv2
import os

cam = cv2.VideoCapture("C:\\Test_video.mp4")
try:
    if not os.path.exists('data'):
        os.makedirs('data')
except OSError:
    print ('Error: Creating directory of data')
currentframe = 0
 while(True):
    ret,frame = cam.read()
    if ret:
        name = './data/frame' + str(currentframe) + '.jpg'
        cv2.imwrite(name, frame)
        currentframe += 1
    else:
        break
cam.release()
cv2.destroyAllWindows()
```

Рис. 3.9. Програмний код розбиття відео на окремі кадри

```
import cv2

cap = cv2.VideoCapture(0)
i = 0
while(cap.isOpened()):
    ret, frame = cap.read()
    if ret == False:
        break
    cv2.imwrite('Frame'+str(i)+'.jpg', frame)
    i += 1
cap.release()
cv2.destroyAllWindows()
```

Рис. 3.10. Отримання відео з камери в режимі онлайн

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('temp.avi', fourcc, 20.0, (640, 480))
while(True):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    out.write(gray)
    if cv2.waitKey(1) & 0xFF == ord('a'):
        break
cap.release()
cv2.destroyAllWindows()
```

Рис. 3.11. Отримання відео з камери в режимі онлайн, конвертацыя та збереження на жорсткий носій даних

Важливою складовою функціонування програмного застосунку є механізм порівняння двох вікон з однаковими просторовими координатами. Процедура зіставлення виконується на піксельному рівні, що дає змогу визначити відсотковий показник подібності між відповідними фрагментами зображень. Отримане значення використовується як критерій для подальшого виявлення руху у відеопотоці. При цьому допустимий рівень похибки може бути налаштований користувачем відповідно до вимог конкретного сценарію застосування.

Швидкодію алгоритму порівняння можливо підвищити шляхом аналізу не кожного пікселя, а вибіркових точок зображення з фіксованим кроком. З огляду на те, що рухомий об’єкт, як правило, займає площу, значно більшу за один піксель, повна перевірка всіх елементів вікна не завжди є необхідною. Водночас у випадку виконання додаткових процедур класифікації для запобігання втраті суттєвої інформації доцільно враховувати всі пікселі без винятку. Такий підхід збільшує обчислювальні витрати, проте забезпечує формування більшого набору інформативних ознак, які використовуються на наступних етапах обробки даних.

Приклад результатів порівняння двох вікон наведено на рисунку 3.12.

```
import cv2

new_image = cv2.imread("c:\\new_image.jpg")
test_image = cv2.imread("c:\\test_image.jpg")

    difference = cv2.subtract(original, duplicate)
    b, g, r = cv2.split(difference)
    if cv2.countNonZero(b) == 0 and cv2.countNonZero(g) == 0 and cv2.countNonZero(r) == 0:
        print("The color is equal")
    else:
        print('The color of image is different')
```

Рис. 3.12. Програмний код порівняння двох зображень

У загальному вигляді програмна реалізація розроблюваної системи передбачає логічний поділ коду на окремі структурні модулі, кожен з яких відповідає за виконання визначеної підзадачі. Такий підхід суттєво спрощує процес подальшого налаштування та модернізації програмного продукту, оскільки зміни можуть вноситися не до всієї системи загалом, а лише до конкретних компонентів. За необхідності окремі модулі можуть бути модифіковані або повністю замінені без порушення працездатності інших частин програмного забезпечення.

Першим умовним етапом програмної реалізації є модуль ініціалізації, який забезпечує запуск застосунку та встановлення основних параметрів його роботи. На цьому етапі також виконується вибір джерела відеоданих: або підключення до вебкамери для обробки відео в режимі реального часу, або завантаження

заздалегідь підготовленого відеофайлу для роботи в автономному режимі. На рисунку 3.13 наведено фрагмент програмного коду, що ілюструє процес задання параметрів функціонування системи та відкриття попередньо збереженого відеоматеріалу. Зокрема, встановлюється інтервал вибірки кадрів, який у наведеному прикладі становить один кадр із шістдесяти. Експериментальні дослідження показали, що за умов типової швидкості переміщення об’єктів така частота є достатньою для своєчасного виявлення руху та прийняття коректних рішень у процесі аналізу відеопотоку.

```
15 count = 0
16 cv2.namedWindow('Video window')
17 while(vid.isOpened()):
18 perform detection = count % 60 == 0
```

Рис. 3.13. Встановлення початкових параметрів роботи програми

Після задання початкових параметрів функціонування програмної системи здійснюється зчитування першого кадру відеопослідовності, який використовується для формування початкового набору зображень та подальшого аналізу. Із відеопотоку за допомогою функції blobFromImage() виконується виділення стартового кадру. Перший аргумент функції визначає вхідне зображення, другий задає коефіцієнт масштабування, третій відповідає за розміри перетвореного зображення, четвертий визначає середнє значення пікселів, п’ятий керує перестановкою колірних каналів, а шостий параметр вказує на виконання операції обрізання.

Отримані дані передаються на наступний етап обробки з метою формування інформативних ознак ковзного вікна. Застосування зазначеної функції дає змогу оперативно отримати необхідні характеристики без суттєвих витрат обчислювальних і часових ресурсів. Після формування початкового набору вікон виконується аналіз наступних кадрів відеоряду шляхом порівняння відповідних ділянок зображень, що дозволяє виявляти факти руху або появу нових об’єктів у кадрі. На рисунку 3.14 наведено приклад маркування відповідних вікон на

вихідному відео, який ілюструє процес детекції рухомих об’єктів.

```
mtracker.add(cv2.TrackerMedianFlow_create(), frame, (x, y, w, h))
count += 1
else:
 cv2.putText(frame, 'Detection failed', (20, 80),
 cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,0,255), 2)
 else:
is_tracking, bboxes = mtracker.update(frame)
 if is_tracking:
for i, bbox in enumerate(bboxes):
 x, y, w, h = [int(val) for val in bbox]
 class_id = classes[class_ids[idxs[i][0]]]
col = [int(c) for c in colors[class_ids[idxs[i][0]], :]]
cv2.rectangle(frame, (x, y), (x+w, y+h), col, 2)
 cv2.putText(frame, class_id, (x, y - 15),
 cv2.FONT HERSHEY SIMPLEX, 0.5, col, 2)
```

Рис. 3.14. Приклад результатів детекції та нанесення міток на виділений об’єкт на зображенні

Поданий перелік програмних функцій у повному обсязі дає змогу проаналізувати результати реалізації окремих компонентів системи, а також наочно відображає ефективність роботи модулів і доцільність обраних програмних рішень. Використана мова програмування разом із набором сучасних цифрових бібліотек відповідає актуальним вимогам та широко застосовується при розробленні програмних засобів для обробки й аналізу цифрових зображень і відеоданих. Завдяки цьому створений програмний продукт використовує переваги бібліотек OpenCV та ImageAI, що забезпечує підвищення загальної ефективності та якості його функціонування.

2.3 Тестування та аналіз реалізованої системи

Для виконання експериментальних досліджень було використано графічну робочу станцію середнього класу, технічні можливості якої є достатніми для розв’язання завдань із проєктування простих об’єктів і двовимірних моделей. Така конфігурація може застосовуватися як універсальне робоче місце інженера-конструктора, дизайнера або адміністративного працівника. Апаратна платформа

має наступні характеристики: процесор Intel Core i3-7100; системна логіка Intel® B250; оперативна пам’ять обсягом 4 ГБ DDR4 з тактовою частотою 2400 МГц із можливістю розширення до 32 ГБ за рахунок двох слотів; жорсткий диск SATA ємністю 500 ГБ; дискретна відеокарта ASUS серії GeForce GT 730 з обсягом відеопам’яті 2 ГБ; мережевий інтерфейс зі швидкістю передачі даних до 1 Гбіт/с; блок живлення потужністю 450 Вт. Виробником материнської плати є компанія Gigabyte.

Для відображення результатів роботи програмної системи використовувався монітор DELL P2018H з діагоналлю 19,45 дюйма. Пристрій оснащений TN-матрицею з роздільною здатністю 1600×900 пікселів та співвідношенням сторін 16:9. Частота оновлення екрана становить 75 Гц, яскравість — 250 кд/м², тип підсвічування — WLED. Підключення забезпечується через інтерфейси HDMI 1.4, DisplayPort 1.2 та VGA. Габаритні розміри монітора складають 570×446×226 мм.

Зазначені технічні параметри апаратного забезпечення забезпечили можливість коректного виконання тестування розробленого програмного продукту та отримання достовірних результатів для подальшого аналізу правильності функціонування системи відеоаналізу.

Для проведення експериментальних досліджень і об’єктивної оцінки ефективності запропонованого алгоритму було сформовано набір відеофайлів, що відрізнялися за низкою характеристик. Тривалість усіх відеоматеріалів була приблизно однаковою та складала близько 30 секунд, що дозволило простежити поведінку програмної системи в динаміці. За рівнем освітленості було обрано відео без різких змін яскравості протягом коротких проміжків часу, зокрема без ситуацій миттєвого ввімкнення або вимкнення освітлення. Кількість рухомих об’єктів у кадрі варіювалася в широких межах: від одиничних цілей (1–5), середньої кількості (10–20) до значної кількості — понад 20 об’єктів одночасно. Додатково враховувалися середні геометричні розміри об’єктів у кадрі та параметри ковзного вікна, що використовувалося для детекції.

Кожен експеримент із відповідним відеофайлом повторювався десять разів, після чого результати усереднювалися. Такий підхід дозволив зменшити вплив випадкових похибок і підвищити достовірність отриманих даних. Важливим фактором, що впливає на ефективність роботи алгоритму, є правильний вибір розмірів ковзного вікна відносно габаритів рухомих об’єктів. Експериментально встановлено, що у випадку, коли розмір об’єкта перевищує розмір вікна більш ніж на 30 % хоча б за однією координатною віссю, такий об’єкт не детектується. З огляду на це параметри ковзного вікна необхідно підбирати емпіричним шляхом для забезпечення коректної роботи програмної системи.

Під час тестування відеоматеріалів із невеликою кількістю рухомих об’єктів (до п’яти) візуальних або функціональних проблем не було виявлено. Детекція виконувалася практично миттєво, супровід об’єктів здійснювався стабільно навіть у випадках часткового виходу об’єкта за межі поля зору камери. Підрахунок кількості цілей здійснювався без помилок. Водночас було зафіксовано некоректну роботу алгоритму у ситуаціях, коли два об’єкти перебували в безпосередньому контакті — у такому разі вони розпізнавалися як одна ціль. Після їх просторового розділення система знову поверталася до правильного визначення кількості об’єктів.

Результати тестування відео із середньою кількістю людей загалом відповідали попереднім експериментам. Однак зі збільшенням кількості об’єктів зростала ймовірність їхнього контакту між собою, що негативно позначалося на точності підрахунку. При цьому сам факт наявності руху визначався стабільно, без втрати чутливості алгоритму.

У випадку аналізу відеоматеріалів з великою кількістю рухомих об’єктів детекція руху виконувалася з високою точністю. Підрахунок цілей був коректним за умови наявності чіткої візуальної межі між окремими об’єктами. Якщо ж значна кількість об’єктів зливалася в одну суцільну область, її розміри могли перевищувати параметри ковзного вікна, що ускладнювало процес коректного підрахунку. Після розділення об’єктів алгоритм знову забезпечував правильне визначення їх кількості.

Опрацювання та узагальнення експериментальних даних показали, що створений програмний комплекс коректно функціонує в процесі обробки відеоматеріалів різної структури та якості. Під час тестування система підтвердила надійність і стійкість роботи, забезпечуючи безперервне виконання алгоритмів детекції без збоїв. Обробка відеопотоків здійснювалася повністю в автоматизованому режимі без втручання оператора, а отримані результати аналізу фіксувалися та накопичувалися у текстовому форматі для подальшої інтерпретації й використання.

На основі експериментальних даних було сформовано узагальнену підсумкову таблицю, у якій результати тестування наведено в систематизованому вигляді (таблиця 3.1). Отримані дані підтвердили гіпотези, висунуті на етапі моделювання, та засвідчили високу швидкодію реалізованого програмного коду. Алгоритм забезпечує ефективну обробку відеопотоків за різних вхідних параметрів при збереженні належної якості результатів.

Загалом результати тестування підтвердили обґрунтованість обраного підходу до розробки програмного застосунку. Створений програмний продукт успішно витримав експериментальні перевірки та показав стабільні й високі показники ефективності в різних умовах використання.

У ході виконання дослідження було здійснено теоретичне опрацювання та моделювання програмного засобу, призначеного для автоматизованого виявлення рухомих об’єктів у відеопотоці, на основі чого реалізовано функціональну систему відеоспостереження. Для оцінювання працездатності розробленого рішення проведено серію експериментальних перевірок програмного комплексу детекції руху, який базується на використанні алгоритму ковзного вікна, за різних сценаріїв подачі вхідних відеоданих. Аналіз результатів випробувань засвідчив правильність прийнятих програмно-алгоритмічних рішень, а також підтвердив доцільність і результативність запропонованого методу виявлення рухомих об’єктів.

Технологія автоматичного розпізнавання руху на основі алгоритму нечіткого кластеризації та VR-відеозображення

Швидкий розвиток технології віртуальної реальності (VR) підвищив інтерес до занурювальних відеододатків. Сучасні методи сегментації динамічного відео наголошують на ітеративній класифікації на рівні пікселів на основі нечітких рівнів членства. Традиційне розпізнавання руху, однак, має труднощі з варіаціями постави у складних ситуаціях, що призводить до неточностей. Щоб вирішити цю проблему, ми пропонуємо фреймворк на основі нечіткої кластеризації для розпізнавання руху відео у VR, підвищуючи точність завдяки

оптимізованій сегментації пікселів і попередній обробці. Було оцінено стабільність системи при різних частотах кадрів. Експерименти використовували CCD-захоплену базу даних із шести позами руху (сидячи, присідання, коліна тощо). Результати показали 96% точність для виявлення присідань, 6,7% хибнопозитивних результатів при поставах на колінах і діапазон похибки 2%–8% для сидячих поз — значно вищий, ніж у інших категоріях через мінливість постави. Це дослідження сприяє розвитку розпізнавання руху у VR, інтегруючи нечітке кластеризацію з адаптивним Обробка зображень, пропонуючи надійні рішення для занурювального навчання, моніторингу реабілітації та систем взаємодії людини з комп'ютером.

З постійним розвитком технологій віртуальної реальності застосування VR-відео стає дедалі ширшим. Однак автоматичне розпізнавання та аналіз рухомих об'єктів у VR-відео залишається складною проблемою [1]. Технологія автоматичного розпізнавання руху на основі алгоритму нечіткого кластеризації та VR-відеозображення може покращити інтерактивність VR-відео [2], [3], [4]. У VR-середовищі користувачі можуть взаємодіяти з віртуальними сценами в реальному часі [5], [6]. Якщо люди зможуть автоматично ідентифікувати та відстежувати рухомі об'єкти у VR-відео, вони зможуть досягти більш інтелектуального інтерактивного досвіду та надати користувачам більш насичений віртуальний досвід [7], [8]. Наразі монтаж VR-відео здебільшого базується на ручному режимі, що займає багато часу і є схильним до помилок [9]. Якщо люди зможуть автоматично розпізнавати рухомі об'єкти у VR-відео та використовувати їх як об'єкти монтажу, ефективність монтажу може значно покращитися.

Хоча алгоритм нечіткої кластеризації може певною мірою покращити автоматичне розпізнавання руху VR-відеозображень, все ж існують певні проблеми: якість зображення у VR-відео може залежати від різних факторів, таких як обладнання камери та умови освітлення [10], [11]. Ці фактори можуть призводити до таких проблем, як розмиття та шум у зображенні, що впливає на точність розпізнавання рухомих об'єктів [12]. Рухомі об'єкти у VR-відео

можуть мати ефект розмиття, особливо якщо вони рухаються на великій швидкості [13]. Таке розмиття руху призводить до нечіткого контуру об'єкта, що ускладнює алгоритму точне розпізнавання об'єкта. Фон у VR-відео зазвичай складний і може містити кілька рухомих об'єктів або фоновий об'єкт, який суттєво змінюється. Цей складний контекст ускладнює сегментацію та розпізнавання об'єктів.

У цій статті аналізується технологія розпізнавання руху на основі алгоритму нечіткого кластеризації та VR-відеозображення. Після обговорення відповідної літератури відповідних науковців аналізуються три вибірки, які будуть використані в цій статті, шляхом аналізу процесу обробки відео у VR та технології попередньої обробки зображень. У цій статті зразок попередньо обробляється, а процес оптимізації алгоритму нечіткого кластеризації аналізується для забезпечення теоретичної основи для наступних експериментів. Згідно зі статистикою бази даних про пози руху, точність розпізнавання пози присідань сягає 96%, а рівень хибнопозитивних результатів — до 6,7%.

Суміжна робота

Через вплив початкової якості відеозображення на пози руху, помилка розпізнавання автоматичного методу розпізнавання пози руху залежить від багатьох факторів, і ефективна класифікація типів рухів дозволяє точно ідентифікувати різні дії руху. Стеффі Л. Кольєр наративно розглядала еволюцію методів вилучення інформації про рух із зображень, спостерігаючи, як техніки еволюціонували від трудомістких ручних методів до систем на основі фотоелектричного маркування [14]. Севгі Зубейде Гурбуз проаналізував застосування радарів, що включають класифікацію цілей і візуалізацію на основі технологій глибокого навчання, і виявив інтерес до глибокого навчання у сфері внутрішнього моніторингу для класифікації щоденної діяльності людини, виявлення падінь і моніторингу аномалій ходи [15]. Пен Ван вивчав глибоке навчання як технологію, засновану на даних, для безперервного аналізу людських рухів і майбутнього прогнозування попиту HRC (Людина-робот співпраця), тим

самим покращуючи планування та контроль роботів під час виконання спільних завдань [16]. Чжіган Ту вважав, що хоча згорткові нейронні мережі були видатними у розпізнаванні зображень, вони не досягли таких вражаючих результатів у розпізнаванні відеодій [17]. Сун Дун вважав, що розпізнавання рухомих траєкторій у реальному часі, точне та без тегів, є актуальною точкою досліджень. На основі введення даних зображень комп'ютерного зору він систематично переглядав відповідні дослідження з машинного навчання або глибокого навчання розпізнавання конкретного руху або цільового руху [18]. Для підвищення частоти розпізнавання рухомих об'єктів Сюе Чжень запропонував глобальний адаптивний метод різниці кадрів і метод фонового віднімання з використанням моделі кодової книги для виявлення одного й того ж рухомого об'єкта після аналізу та комплексного порівняння переваг і недоліків різних алгоритмів виявлення рухомих об'єктів [19]. Наведені вище дослідження показують, що технологія автоматичного розпізнавання руху значно покращилася після поєднання машинного навчання та інших методів. Однак методи автоматичного розпізнавання зображень у умовах високошвидкісного руху зазвичай мають проблеми, такі як висока помилка розпізнавання, тривалий час роботи та слабкий ефект.

Алгоритм нечіткої кластеризації може використовуватися для кластеризації ознак руху. У відеозображеннях ознаки руху можуть включати положення цілі, швидкість, напрямок тощо [20]. Алгоритм нечіткої кластеризації може розділяти та групувати ознаки руху, враховуючи схожість і нечітке членство між вибірками, щоб реалізувати автоматичне розпізнавання різних типів руху. Намейракпам Дханачандра вважав, що сегментація зображень є важливим кроком у обробці зображень, а нечіткі C-середні — це простий і ефективний алгоритм кластеризації [21]. Однак алгоритм нечіткої кластеризації має недоліки: чутливий до початкових значень, легко потрапляє до локально оптимальних рішень і чутливий до шуму. Тао Лей вважав, що алгоритми кластеризації, які мінімізують цільову функцію, мають очевидний недолік — кількість кластерів потрібно встановлювати вручну. Хоча пікове кластеризація щільності може визначати

кількість кластерів, при використанні для сегментації зображень на неї впливає переповнення пам'яті [22]. Cong Wang запровадив залишкові нечіткі C-середні для сегментації зображень, що стало першим методом для точної оцінки залишків і дозволу безшумним зображенням брати участь у кластеризації. Інтегруючи залишкові регуляризаційні терміни кореляції, отримані з особливостей розподілу різних типів шуму, у алгоритм нечіткого кластеризації, він широко використовується для сегментації зображень у відтінках сірого та кольорових зображень. Однак більшість із них займають багато часу і не дають бажаних результатів сегментації кольорових зображень [23]. Сюй Цзіньдун запропонував метод сегментації зображення на основі C-середніх, який використовує фонову інформацію C-середніх для підвищення точності та антишумової здатності сегментації зображення. У цьому дослідженні також застосовується міра достовірності на основі просторового контексту сегментації зображення [24]. Хоча метод кластеризації з нечіткими C-засобами має певні переваги у розпізнаванні зображень, він забирає багато часу та простору для кластеризації. Наразі бракує спеціальної бази даних рухомих відеопослідовностей для дослідження доказів.

VR-відеозображення та попередня обробка

VR-відеозображення

У цій статті розглядається особливості VR-відеозображень, такі як панорамний кут огляду, відстеження голови, сприйняття глибини тощо, а також їхній вплив на технологію автоматичного розпізнавання руху. Віртуальна реальність на основі моделей головним чином призначена для моделювання певної сцени, а віртуальна реальність на основі відео — це використання камери для досягнення. VR-технологія може не лише створити віртуальне середовище, що відповідає правилам реального світу, а й створити повне гіпотетичне середовище, що суперечить реальності, щоб люди могли відчути занурення [25].

Виходячи з наведених вище кроків, ця стаття розширює свій зміст. По-перше, відеозахоплення вимагає використання спеціальних приладів, такі пристрої

зазвичай оснащені кількома камерами для захоплення 360-градусних панорамних або стереозображень. Зібрана інформація про зображення передавалася на наступний етап обробки. Щоб забезпечити цілісність і послідовність мозаїчного зображення, його потрібно виправити, вирівняти та злити для покращення якості зображення та покращення відчуття перегляду. Рендеринг зображень у віртуальній реальності — це процес перетворення обробленого зображення для відтворення на пристрої віртуальної реальності. Щоб відтворювати та взаємодіяти з пристроєм віртуальної реальності в реальному часі, зображення має бути сегментоване, стиснене та закодоване, а також необхідна інтерактивна обробка, коли відеозображення віртуальної реальності має бути інтерактивним із глядачем.

Вплив віртуальної реальності на технологію автоматичного розпізнавання є значним. У цій статті досліджується вплив VR-відеозображень на технологію автоматичного розпізнавання руху.

Технології віртуальної реальності можуть надати користувачам більш реалістичні та занурюючі сцени, краще збирати велику кількість даних, а також надавати більше навчальних і тестових даних для автоматичної ідентифікації користувачів [26]. Використання технологій віртуальної реальності для моделювання різних сценаріїв дозволяє ефективно контролювати необхідні динамічні дані, що надає користувачам більш реалістичне середовище симуляції, щоб вони могли навчати та тестувати технологію розпізнавання мовлення точніше та ефективніше. Технологія віртуальної реальності може покращити комунікацію між людьми та зробити технологію автоматичної ідентифікації зручнішою у математичному аналізі та керуванні. Оскільки віртуальний дисплей має властивості занурення, рух може інтуїтивно взаємодіяти з системою машинного зору, що дозволяє інформація та зворотний зв'язок бути точнішими.

Загалом, віртуальна реальність дарує людям позитивне відчуття, забезпечуючи якісне середовище для збору даних, навчання та тестування для розробки комп'ютерних обчислювальних систем. Вона також забезпечує ідеальне середовище для збору даних, навчання та тестування для розробки систем

автоматизованих обчислень. Це може забезпечити більш насичену інформацію про сприйняття середовища для розробки систем автоматизованих обчислень, а потім підвищити точність і ефективність алгоритму.

Технологія попередньої обробки зображень

У цій статті розглядаються технології попередньої обробки для VR-відеозображень, включно з видаленням шуму, вилученням ознак зображення, сегментацією зображень тощо. Для підвищення точності та стабільності автоматичного розпізнавання руху дослідницький об'єкт зібрав 3 зразки з онлайн-бази даних і провів дослідження наступним чином:

У цій статті встановлюється модель трансформації просторово-просторової подібності на основі централізації, концентрується піксельна та об'єктна координата багатосмугового зображення на одному зображенні, встановлюється приблизна модель параметрів реєстрації зображення шляхом розрахунку коефіцієнта масштабу кожного пікселя, і нарешті отримує ефект зняття шуму зображення, як показано на рис. 2:

Припустимо представляє координати простору об'єктів у цій точці; R — це матриця обертання між двома системами координат, а f — фокусна відстань камери. Цю фокусну відстань можна визначити як номінальне значення, що визначає систему координат простору об'єктів у цій точці . Потім використовується комбінація просторових рівнянь і моделей перетворень для усереднення координат точок у мультиспектральних зображеннях і об'єктах:

Впровадження алгоритму нечіткого кластеризації

У цій статті описані принципи поширених алгоритмів нечіткої кластеризації, таких як нечіткий алгоритм кластеризації C-середніх та нечіткий алгоритм кластеризації матриці тенденцій C-середніх.

Як багатовимірний статистичний метод, кластерний аналіз є важливим напрямком досліджень методів неконтрольованого розпізнавання образів і широко

використовується в багатьох сферах, таких як класифікація патернів, обробка зображень і обробка нечітких правил [27]. Вибірки без класифікаційних балів були поділені на кілька підгруп за певними критеріями. Жорстка кластеризація строго поділяє кожен об'єкт, який потрібно ідентифікувати, на клас і має характеристики «або/або». Нечітке кластеризація створює невизначений опис класу вибірки, який може більш об'єктивно відображати об'єктивний світ, тому він став основним у кластерному аналізі.

Відповідні дослідження показують, що алгоритм нечітких C-середніх є найпоширенішим і найуспішнішим на даний момент. Оптимізуючи цільову функцію, вона отримує ступінь належності між кожною точкою вибірки та різними центрами, а потім визначає категорію вибіркових точок для реалізації автоматичної класифікації даних вибірки [28]. Це забезпечує функцію членства для кожного кластера та класифікує вибірку відповідно до її складу. Основні параметри алгоритму нечітких C-середніх — це кількість кластерів, центроїд кластерів і кластер, де кожна точка даних належить найближчому центроїду.

Більш чіткий покроковий розподіл алгоритму FCM:

- 1.
  Ініціалізуйте центри кластерів і матрицю належності випадково.
- 2.
  Обчислюйте центри кластерів на основі зваженого середнього значення всіх точок даних.
- 3.
  Оновіть значення членства за допомогою функції нечіткого членства.
- 4.
  Ітерація до збіжності, тобто коли зміни значень членства та центрів кластерів мінімальні.

У цій статті центр кластеризації отримується за допомогою алгоритму нечітких C-середніх для мінімізації цільової функції. По суті, цільова функція — це евклідова відстань між однією точкою і одним класом. Процес кластеризації — це процес мінімізації цільової функції. Після повторних ітеративних операцій значення помилки цільової функції поступово зменшується, і нарешті можна отримати фінальний результат кластеризації, коли цільова функція збігається.(10)

У формулі, — кількість кластерів у кластерах; — кількість вибірок у кластерах, і — кількість центрів кластерів. Серед них, представляє характеристичну розмірність центр кластера; позначає співвідношення підпорядкування між центрами кластерів, і представляє співвідношення підпорядкування центрів кластерів. Серед методів, які описують схожість між даними, найпоширенішим є евклідова функція.(11)

Матриця членства представляє матриці довжини, до якої кожна точка вибірки належить кожному класу. За однією вибіркою загальний ступінь належності кожного кластера дорівнює 1, і категорія з найбільшим ступенем належності кожної точки вибірки класифікується до цієї категорії. Чим ближче до 1, тим сильнішим буде відчуття приналежності.

Поясніть, як працює рівняння оновлення членства:(12)

Детальніше поясніть, як виводиться рівняння оновлення центру кластера:(13)

Взаємозв'язок між та є взаємно інклюзивним, і проблема алгоритму FCM полягає в тому, що жоден з них не існує ані на початковому етапі. Цей алгоритм можна вважати кращим рішенням, якщо J є сталою. За цільовою функцією , та Сформуйте позитивний зворотний зв'язок, тобто ступінь членства та Cluster Center C можна ітеративно обчислити без урахування цільової функції для відповідності вимогам.

У цій статті досліджується метод кластеризації, заснований на нечіткій матриці C-середнього тенденцій, яка оптимізує метод кластеризації на основі традиційного методу кластеризації C-середнього. Використовується новий метод для реалізації

нечіткої частки вибіркових точок у кількох кластерах, щоб краще описати розподіл вибіркових точок у кластерах. Процес оптимізації його алгоритму аналізується наступним чином:

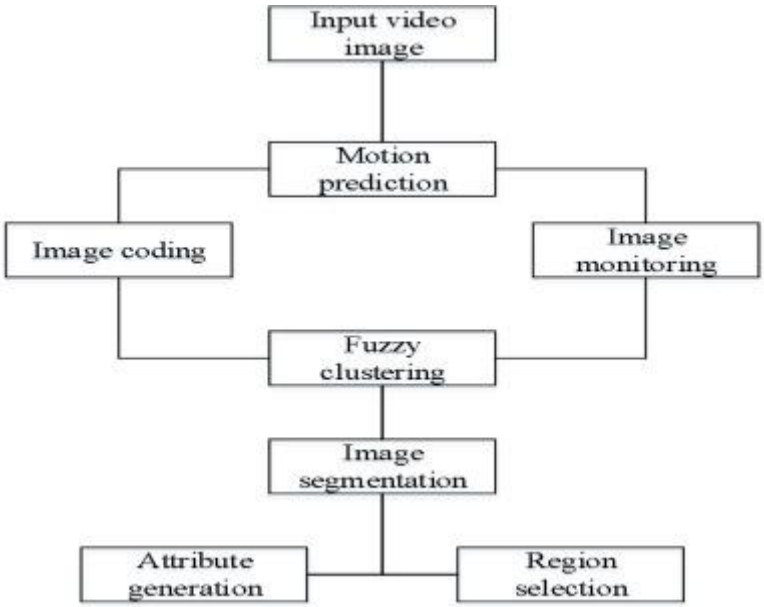
Визначаються кількість центрів кластерів K та умова завершення ітерації, після чого K кластерних центрів випадково ініціалізуються. Для кожної точки вибірки, щоб обчислити її схожість із центром кожного кластера, ймовірність того, що кожна точка вибірки буде призначена більш ніж одному центру кластера, може розглядатися як елемент матриці схильності. Елементи матриці схильності відображають ступінь належності кожної точки вибірки, що відповідає центру кластера, а нова позиція кожного центру кластера обчислюється відповідно до матриці схильності. На цій основі зважене середнє кожної точки вибірки використовується для оновлення кожного центру кластера, де вага — це ступінь зміщення кожної точки вибірки відносно центру кластера, а другий і третій кроки повторюються до завершення циклу.

Загалом, нечіткий алгоритм матриці тенденцій C-середніх використовує метод матриці тенденцій для розмиття ступеня належності вибіркових точок, щоб краще відобразити розподіл вибіркових точок. Система автоматичного розпізнавання руху

У цій статті розглядається новий алгоритм, який спрямований на велику випадковість, тенденцію нечітких змін і складність точного позиціонування рухомої позиції цільового об'єкта на зображенні відео. Відокремивши рухому позицію цільового об'єкта від непотрібної піксельної площі, досягається точне вилучення рухового стану цільового об'єкта. Рис. 7 показує систему автоматичного розпізнавання руху:

Як видно на рис. 7, рухома відеопослідовність отримується за умови відтворення рухомого відео, а програма визначення країв запускається синхронно. Він прогнозує та компенсує рух рухомих об'єктів на основі програми. Однаковий фон будується на кількох зображеннях з дуже малими відстанями, а FDTD (Finite Difference-Time Domain) зображення формується розміром кадру сусідніх

зображень. Властивість нечіткості рухомих об'єктів встановлюється і записується як функція членства. Процес нечіткого кластеризації використовується для сегментації контуру краю рухомого об'єкта та отримання передньої площі рухомого об'єкта, щоб реалізувати прогнозування руху та компенсацію рухомого об'єкта.



4.3. Реалізація та оптимізація алгоритму

Реалізована технологія автоматичного розпізнавання руху на основі алгоритму нечіткого кластеризації, стратегії оптимізації спрямовані на підвищення точності та продуктивності в реальному часі.

У цій статті концепція нечіткого кластеризації інтегрована в реалізацію алгоритму для ефективного пригнічення порушень у відеопотоках під час передачі. Алгоритм використовує адаптивні методи кластеризації для динамічного уточнення центрів кластерів, забезпечуючи надійне розпізнавання руху. Для відстеження локальних варіацій використовується підхід ковзного вікна, що дозволяє алгоритму ефективно порівнювати послідовні кадри та підвищувати чутливість.

На цій основі застосовується вдосконалений метод кластеризації, заснований на алгоритмі нечітких С-середніх (FCM) для приглушення шуму. Впроваджено новий підхід до цифрової обробки, що використовує адаптивний

нечіткий ітеративний алгоритм усереднення C-середніх. У цьому методі піксель, встановлений у маленькому вікні, розглядається як нечітка множина, де кожен елемент має змінний ступінь належності відносно центру концентрації. Алгоритм адаптивно коригує значення членства, ефективно мінімізуючи шум шляхом пригнічення точок даних, які суттєво відхиляються від центру кластера.

Крім того, оптимізація вибору ознак інтегрована в процес. Методи зменшення розмірності, такі як аналіз основних компонентів (PCA), використовуються для вилучення найважливіших ознак руху, що підвищує точність кластеризації та знижує обчислювальну складність. Крім того, використання паралельних методів обробки, таких як прискорення GPU, підвищує обчислювальну ефективність і продуктивність у реальному часі.

Припустимо, що відеопотік, захоплений одним кадром, обробляється як файл зображення, а ширина та висота зображення встановлені як та , відповідно. Стратегії оптимізації значно підвищують точність розпізнавання, скорочують час обробки та підвищують стійкість до шуму та оклюзій, роблячи алгоритм придатним для реальних застосувань, таких як занурювальне тренування та моніторинг реабілітації.

Для піксельне зображення, візьміть вікно з початкової точки, наприклад, формулу:(14)

представляє середнє та представляє значення пікселя; це взяти вікно парного розміру і знайти середнє значення його першого вікна. Тоді візьміть вікно з початкової точки зображення, як наступна формула:

Ітеративне середнє можна отримати, де ABS є функцією абсолютного значення. ε — це невелике додатне число. Видно, що чим ближче значення пікселя до середнього , чим більший був би ступінь членства і тим більший вплив на ітеративне середнє значення Можливо. Тут використано три згадані вище зразки зображень і аналізують їхні ітеративні середні значення, як показано в Таблиці 1:

Як показано в Таблиці 1, отриманий результат є середнім значенням після 6 ітерацій. Якщо виконана операція збігається з трьома операціями на кроці 2, можна отримати середнє значення кількох ітерацій. Однак чим більше ітерацій у людей, тим швидше програма буде суттєво змінюватися. Щоб моніторити в реальному часі та отримати кращі результати виявлення, зазвичай потрібно лише дві-три ітерації.

5. Експеримент і результат автоматичного розпізнавання руху

У цій статті аналізуються особливості руху відеозображень віртуальної реальності на основі нечіткої кластеризації, а відеозображення перетворюються на коди, які комп'ютери можуть читати та розпізнавати безпосередньо, а потім функцію ідентифікації методу проектування виконує комп'ютерна робота. На встановленій тестовій платформі була узагальнена динамічна відеотека, захоплена камерою CCD (Charge Coupled Device), щоб забезпечити узгодженість динамічного відео та відповідних відеопараметрів. Усі відеопослідовності зберігаються в базі даних послідовно за серійним номером, а формат ідентифікації конвертується [29].

Згідно з наведеними вище методами зберігання та збору даних, створюється база даних, що містить 100 рухомих відеопослідовностей 6 типів, і кожен тип відеопереміщення є одним і тим самим методом. Параметри відеозображень у базі даних наведені в Таблиці 2:

Як видно з Таблиці 2, існує шість типів форматів зображень для рухового відео: 1920 × 1080, 1680 × 1050, 1280 × 720 та 1280 × 960, а кількість кадрів коливається від 100 до 180. Серед них є 35 відеознімків стоянки, 15 відеознімків присідання та 10 відеознімків положення на колінах. Було загалом 8 відеознімків лежачого положення, 22 відеозображення сидячого положення та 10 відеозображень підтримки.

На основі налаштувань бази зображень з рухомим відео вище, тут аналізується правильна кількість розпізнавання 6 типів поз руху, як показано на рис. 8:

Легенда показує різні положення руху, включно з стоянням, присіданням, коліном, лежачим, сидячим положенням і підтримкою. Видно, що кількість зображень, правильно ідентифікованих першим тестом, становила 80, а кількість зображень, правильно ідентифікованих другим тестом — 79. Третій тест правильно ідентифікував 74 зображення наведеної вище рухомої пози; Четвертий тест правильно ідентифікував 76 зображень наведеної вище рухомої пози, а п'ятий — 81 зображення наведеної вище рухомої пози. У шостому тесті було правильно ідентифіковано 94 зображення наведеної вище пози руху; У 7-му тесті було правильно ідентифіковано 76 зображень наведеної вище пози руху; У 8-му тесті було правильно ідентифіковано 76 зображень наведеної вище пози руху; Під час 9-го тесту було правильно ідентифіковано 82 зображення. У десятому тесті кількість зображень, які правильно ідентифікували вищезазначену позу руху, становила 75 (див. Таблицю 3). На основі даних першого тесту у Таблиці 2 та Рис. 8, позитивні зразки стоячи, присідання, колінного положення, положення лежачи, сидячого положення та опори були встановлені як і негативні зразки , як показано на рис. 3. Потім розраховувалися рівень точності, відтворення та хибнопозитивних результатів, як показано в Таблиці 4:

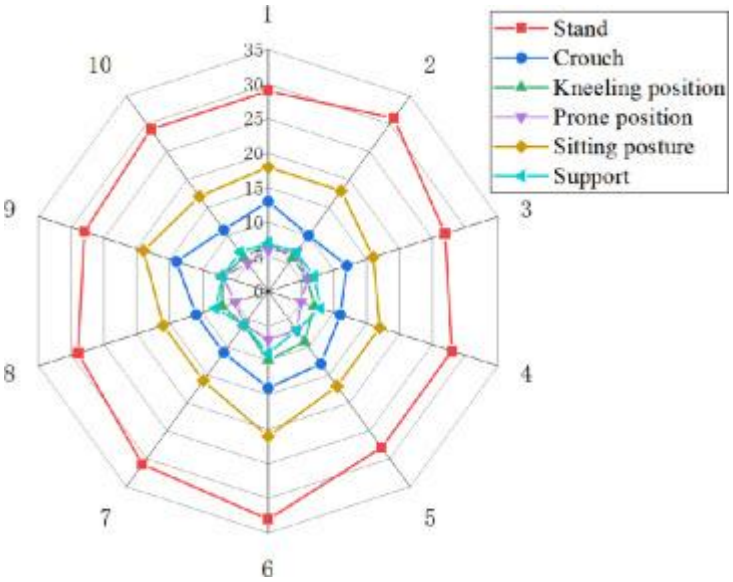


Рис. 8. Правильний кількісний аналіз розпізнавання зображень.

На основі правильного дослідження частоти 10 тестів на рис. 8 було проаналізовано кількість визнаних пози в базі даних. Потім було проаналізовано

та розраховано рівень точності, відпочивання та хибнопозитивний показник розпізнавання 6 спортивних поз у Таблиці 4, як показано на рис. 10:

Перспективи застосування та виклики

На основі наведених вище досліджень ця стаття вважає, що технологія автоматичного розпізнавання руху може фіксувати та ідентифікувати людські рухи в реальному часі, забезпечуючи більш реальний і занурюючий досвід віртуальної реальності. Базуючись на технології автоматичного розпізнавання руху, він реалізує сприйняття та реакцію рухів і дій користувачів у реальному часі, щоб користувачі могли взаємодіяти з рухами тіла, забезпечуючи більш інтуїтивний і занурюючий досвід. Що стосується ігор, технологія автоматичного розпізнавання руху може застосовуватися до ігор із датчиком руху, спортивних ігор тощо, щоб гравці могли використовувати рухи тіла для маніпуляції персонажами гри, що покращує задоволення та взаємодію ігор [30].

Зосереджені на задачах реального часу, точності та складності сцени у віртуальній реальності, ця стаття розглядає метод автоматичного розпізнавання дій на основі нечіткого кластеризації. У сферах віртуальної реальності та соматосенсорної взаємодії необхідно вивчати та розпізнавати людські рухи та жести з високою точністю та з низькою затримкою, що створює великі труднощі для автоматичного розпізнавання людських рухів. У віртуальній реальності, соматосенсорній взаємодії та інших сценаріях поведінка або постава користувача часто знаходяться на складному фоні багатоосібної взаємодії та оклюзії, що створює серйозні виклики для стабільності та надійності систем розпізнавання поведінки. Однак створення великомасштабних і різноманітних наборів даних розпізнавання дій є не лише трудомістким і трудомістким, а й складним для адаптації до різних сценаріїв застосування та груп користувачів. На основі алгоритму нечіткої кластеризації та технології автоматичного розпізнавання руху VR-відеозображень, ця стаття розпізнає та аналізує рухомі об'єкти. У розділі обговорення підсумовуються основні висновки дослідження, висвітлюються інноваційні аспекти дослідження та обговорюються обмеження.

Технологія автоматичного розпізнавання руху на основі алгоритму нечіткої кластеризації та VR-відеозображення досягла значних покращень порівняно з традиційними методами розпізнавання руху. На відміну від традиційних методів, які часто страждають від високої чутливості до розмиття руху, оклюзій і шуму навколишнього середовища, запропонована структура нечіткого кластеризації підвищує стійкість шляхом динамічного коригування значень членства та використання адаптивної сегментації. Це дозволяє точніше розпізнавати рух навіть у складних VR-відеосередовищах.

Однією з ключових інновацій цього дослідження є інтеграція нечіткого кластеризації з адаптивною обробкою зображень, що підвищує як точність розпізнавання, так і обчислювальну ефективність. Оптимізуючи сегментацію та попередню обробку пікселів, система ефективно зменшує хибнопозитивні результати та підвищує стабільність відстеження об'єктів. Крім того, запропонований підхід демонструє високу адаптивність у роботі з різними типами рухів, що робить його придатним для різноманітних застосувань, таких як занурювальне навчання, моніторинг реабілітації та інтерактивні системи людини-комп'ютер.

Потенціал застосування цієї технології є широким, включаючи, але не обмежуючись, відеоспостереженням, управлінням трафіком і взаємодією в реальному часі VR. На відміну від традиційних алгоритмів розпізнавання руху, які мають труднощі з динамічною природою VR-відеозображень, підхід на основі нечіткої кластеризації пропонує більш гнучку та точну систему розпізнавання, здатну адаптуватися до варіацій у патернах руху та складності фону.

Незважаючи на ці переваги, деякі обмеження залишаються. Обчислювальне навантаження на обробку великих VR-відеонаборів даних є викликом, що вимагає подальшої оптимізації для покращення продуктивності в реальному часі. Крім того, для дуже складних рухових патернів нечітке кластерування все ще може мати труднощі з точним розпізнаванням, особливо у випадках із значними оклюзією або накладанням рухів.

Для вирішення цих викликів пропонується кілька напрямків для майбутніх досліджень. По-перше, підвищення ефективності алгоритму шляхом інтеграції паралельних обчислень або апаратних технологій прискорення, таких як обробка GPU, може суттєво зменшити обчислювальні витрати. По-друге, вдосконалення процесу розпізнавання для підтримки ширшого спектра типів рухів покращить здатність алгоритму до узагальнення. Нарешті, впровадження гібридних моделей, які поєднують нечітке кластеризацію з глибоким навчанням або передовими методами обробки зображень, може додатково підвищити точність і надійність розпізнавання.

Підсумовуючи, ця стаття пропонує інноваційний підхід до розпізнавання руху на основі нечіткого кластеризації у VR-відеозображеннях, ефективно вирішуючи деякі ключові обмеження традиційних методів. Хоча проблеми з масштабованістю та точністю розпізнавання залишаються актуальними, запропонована методологія пропонує перспективний напрямок для майбутніх досягнень у технології розпізнавання руху.

Висновок

Аналізується технологія розпізнавання руху, заснована на алгоритмі нечіткої кластеризації та VR-відеозображення. У цій статті ітеративний метод використовується для сегментування цільової області пікселів, дослідження класифікації пікселів проводиться за різними нижчими рівнями, а також аналізуються процеси обробки та технологія попередньої обробки зображень VR-відео. Аналізуються 3 зразки, які будуть використані в цій статті, а також точність і стабільність розпізнавання зразків зображень під різними номерами кадрів. У цій статті вивчається технологія автоматичного розпізнавання руху на основі алгоритму нечіткої кластеризації, а також обговорюється процес створення фреймворку автоматичного розпізнавання руху. Він зібрав бібліотеку динамічних відео та вивчав різні вправні пози. Рівень точності, запам'ятовування та рівень помилок у базі даних досліджуються як орієнтир для дослідження технології автоматичного розпізнавання руху.

ВИСНОВКИ

На підставі детального дослідження сучасних програмних комплексів відеоспостереження, що використовують механізми автоматичної детекції руху, а також аналізу методів виділення об’єктів шляхом обробки послідовностей кадрів відеопотоку, можна сформулювати узагальнені висновки щодо стану та напрямів розвитку даної галузі.

У ході виконаного дослідження було здійснено ґрунтовний аналіз задач комп’ютерного зору та їх структурну класифікацію. Це дозволило впорядкувати основні типи задач, що виникають під час обробки цифрових зображень і відеоданих, а також визначити ключові функціональні напрями роботи відповідних алгоритмів. Зокрема, розглянуто задачі попередньої обробки зображень, виявлення та сегментації об’єктів, аналізу руху, відстеження цілей і прийняття рішень на основі візуальної інформації.

Отримана класифікація дала змогу систематизувати підходи до побудови алгоритмів комп’ютерного зору та оцінити їх придатність для використання в системах відеоспостереження з вимогами до реального часу, точності детекції та обчислювальної ефективності. Це створює підґрунтя для подальшого вдосконалення алгоритмів обробки відеопотоків і їх адаптації до конкретних прикладних завдань.

Дослідження методів кодування та передачі відеосигналів

У межах роботи було виконано аналіз сучасних підходів до кодування та передавання відеоданих у цифрових системах. Розглянуто особливості різних форматів стиснення та протоколів передачі відеопотоків, з урахуванням вимог до пропускної здатності каналу, затримки та якості зображення. Це дозволило обґрунтовано визначити найбільш доцільні алгоритмічні рішення для створення програмної системи, орієнтованої на ефективну детекцію руху в реальному часі.

Аналіз існуючих програмних засобів для виділення та класифікації об’єктів

Проведено огляд та порівняльний аналіз наявних програмних рішень, що реалізують функції виявлення та класифікації об’єктів у відеопотоці. У результаті дослідження було визначено типову архітектуру подібних систем, виокремлено основні функціональні модулі та встановлено принципи організації інформаційного обміну між ними. Це дало змогу сформувати вимоги до структури власної програмної реалізації.

Огляд алгоритмів виділення та класифікації об’єктів

Здійснено аналітичний огляд алгоритмів обробки цифрових зображень, які використовуються для виділення та подальшої класифікації об’єктів. Розглянуто як класичні методи, засновані на аналізі різниці кадрів та фонових моделей, так і більш сучасні підходи. Результати цього огляду стали теоретичною основою для розробки власного методу виділення рухомих об’єктів у відеопотоці.

Розробка алгоритму детекції рухомих об’єктів

У роботі запропоновано оригінальний алгоритм детекції рухомих об’єктів, який базується на використанні принципу ковзного вікна вздовж меж зображення. Застосування даного підходу дозволило виконати структурне проєктування програмної системи та реалізувати ефективний механізм виявлення об’єктів у відеопотоці з урахуванням обмежень обчислювальних ресурсів.

Тестування та підтвердження працездатності системи

У процесі виконання роботи було проведено комплексне тестування створеної програмної системи з використанням різних наборів вхідних даних та сценаріїв роботи. Перевірка здійснювалася як у штатних, так і у граничних режимах функціонування, що дало змогу оцінити стабільність та коректність роботи програмного забезпечення. Отримані результати підтвердили обґрунтованість

обраних архітектурних і алгоритмічних рішень, а також високу ефективність запропонованого алгоритму виявлення рухомих об’єктів у відеопотоці.