

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА СИСТЕМ ІНФОРМАЦІЙНОГО ТА КІБЕРНЕТИЧНОГО ЗАХИСТУ

«На правах рукопису»
УДК 681.3.06

До захисту допущено
Завідуючий кафедрою СІКЗ
_____ к.т.н. Г.В. Шуклін
« ____ » _____ 2021 р.

МАГІСТЕРСЬКА АТЕСТАЦІЙНА РОБОТА

зі спеціальності 125 “Кібербезпека”

на тему: **МЕТОД ПОШУКУ І СУПРОВОДУ ОБ’ЄКТІВ У
ВІДЕОІНФОРМАЦІЙНИХ ПОТОКАХ**

Виконав: студент б курсу, групи СЗДМ-61
Спеціальності 125 Кібербезпека
Освітньо-професійної програми
«Технічні системи інформаційного та
кібернетичного захисту»

(шифр і назва спеціальності)

Бортник О.С.

(прізвище та ініціали)

Керівник Крючкова Л.П.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтролер Гребенніков А.Б.

ЗАТВЕРДЖУЮ

Завідувач кафедри СІКЗ

_____ к.т.н. Шуклін Г.В

“ ___ ” _____ 2020р.

ЗАВДАННЯ

на магістерську кваліфікаційну роботу

студенту Бортнику Олексію Сергійовичу

1. Тема роботи: Метод пошуку і супроводу об'єктів у відеоінформаційних потоках, керівник Крючкова Лариса Петрівна, д.т.н., затверджені наказом вищого навчального закладу від 14 листопада 2020 року № 518.

2. Термін здачі студентом оформленої роботи 21 грудня 2020 р.

3. Об'єкт дослідження: процес пошуку і супроводу об'єктів у відеоінформаційних потоках.

4. Предмет дослідження: методи і алгоритми пошуку і супроводу об'єктів у відеоінформаційних потоках.

5. Мета роботи: синтез удосконаленого методу пошуку і супроводу об'єктів у відеоінформаційних потоках.

6. Перелік питань, які мають бути розроблені:

1. Виконати аналітичний огляд існуючих методів пошуку і супроводу об'єктів у відеоінформаційних потоках.
2. Розглянути алгоритми відстеження руху об'єктів у відеоінформаційних потоках.
3. Здійснити синтез удосконаленого методу пошуку і супроводу об'єктів у відеоінформаційних потоках.
4. Здійснити оцінку ефективності запропонованого методу.

7. Перелік публікацій:

8. Перелік ілюстративного матеріалу:

1. Презентація виконана на ___ слайдах для подання за допомогою оверхедів (світлопроекторів) та комп'ютерних засобів.

9. Дата видачі завдання 14 вересня 2020 р.

Керівник: Крючкова Лариса Петрівна _____

Завдання прийняв до виконання: Бортник Олексій Сергійович _____

КАЛЕНДАРНИЙ ПЛАН

№ З\П	Назва етапів магістерської роботи	Строк виконання етапів	Примітка
1	2	3	4
1	Уточнення постановки завдання	до 16.09.20р.	Виконано
2	Аналіз літератури	до 11.10.20р.	Виконано
3	Обґрунтування вибору рішення	до 25.10.20р.	Виконано
4	Збір даних	до 01.11.20р.	Виконано
5	Написання першого розділу роботи	до 08.11.20р.	Виконано
6	Написання другого розділу роботи	до 15.11.20р.	Виконано
7	Написання третього розділу роботи	до 29.11.20р.	Виконано
8	Написання четвертого розділу роботи	до 13.12.20р.	Виконано
9	Підготовка ілюстративного матеріалу	до 06.01.20р.	Виконано
10	Отримання рецензій	до 10.01.20р.	Виконано
11	Захист в ДЕК	14.01.20р.	

Студент _____ О.С. Бортник

Науковий керівник _____ Л.П. Крючкова

РЕФЕРАТ

Текстова частина бакалаврської роботи: 94 сторінки, 47 рисунків, 20 джерел.

Актуальність: В останні часи одним з найпопулярніших засобів нагляду за об'єктами є відеомоніторинг. Існує маса різноманітних онлайн відео сервісів, та відеообладнань які транслюють зображення в прямому ефірі або ж безпосередньо записують відеопотік на накопичувач. Відеомоніторинг та відеообладнання удосконалюється з роками, якість відеозапису зростає, як і технології методу пошуку об'єктів та їх супроводу.

Основа відеомоніторингу – нагляд за важливими об'єктами, рухомими або нерухомими. Моніторинг подій та ситуацій в певному периметрі.

Було розглянуто та проаналізовано методи класифікації рухомого об'єкта на відео. Запропоновано модифіковані методи класифікації та ведення рухомої цілі на основі гістограмних оцінок і метод ідентифікації рухомих точок на відео на основі міжкадрової різниці. Отримано алгоритми, які використовуються для розроблення інформаційної технології оброблення відео

Мета роботи: запропонувати розробку алгоритму пошуку та супроводу об'єктів в відеопотоках, яка буде аналізувати контент потокового відео і за допомогою технологій розпізнавання та фіксації даних взаємодіяти з об'єктом.

Об'єкт дослідження: контент відеопотоку.

Предмет дослідження: методи аналізу контенту потокового відео.

Наукова новизна отриманих результатів полягає у:

- новому алгоритмічному підході до аналізу контенту відеопотоку у реальному часі;

Практичне значення одержаних результатів полягає у:

- створенні інформаційної системи, яка здатна проаналізувати біометричні дані та виявляти максимально схожий об'єкт за цими параметрами і запропонувати його дані глядачу.

Галузь використання – інформаційна безпека.

Ключові слова: ЗОБРАЖЕННЯ, ТРАЄКТОРІЯ РУХУ, БІНАРНА МАСКА, ЕРОЗІЯ, ВІДЕО ПОСЛІДОВНІСТЬ, ВІДЕОРЯД, МАТЕМАТИЧНА МОРФОЛОГІЯ, СИСТЕМИ ВІДЕОСПОСТЕРЕЖЕННЯ.

ABSTRACT

Text part of the bachelor's thesis: 94 pages, 47 drawings, 20 sources.

Relevance: Recently, one of the most popular means of monitoring objects is video monitoring. There are a lot of different online video services and video equipment that broadcast images live or directly record the video stream to the drive. Video monitoring and video equipment are improving over the years, the quality of video recording is increasing, as well as the technology of the method of searching for objects and their tracking.

The basis of video monitoring is the surveillance of important objects, movable or immovable. Monitoring of events and situations in a certain perimeter.

Methods of classifying a moving object on video were considered and analyzed. Modified methods of classification and guidance of a moving target based on histogram estimates and a method of identification of moving points on video based on interframe difference are proposed. The algorithms used for the development of information technology for video processing are obtained

Purpose: to propose the development of an algorithm for searching and tracking objects in video streams, which will analyze the content of streaming video and use technologies to recognize and capture data with the object.

Object of research: video stream content.

Subject of research: methods of content analysis of streaming video.

The scientific novelty of the obtained results is:

a new algorithmic approach to real-time video stream content analysis;

The practical significance of the obtained results is:

Creating an information system that is able to analyze biometric data and detect the most similar object in these parameters and offer its data to the viewer.

Field of use - information security.

Key words: IMAGE, MOTION TRAJECTORY, BINARY MASK, EROSION, VIDEO SEQUENCE, VIDEO SEQUENCE, MATHEMATICAL MORPHOLOGY, VIDEO SURVEILLANCE SYSTEMS.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ВИЯВЛЕННЯ ОБ'ЄКТІВ: НЕСКІНЧЕННА ТЕОРЕТИЧНА ПЕРСПЕКТИВА.....	11
1.1 Проблеми пошуку і супроводу об'єктів в відеофайлах.....	11
1.2 Способи зображення об'єктів.....	12
1.3 Метод вичитування фону.....	14
1.4 Статичні методи.....	15
1.5 Різниця в часових проміжках.....	15
1.6 Аналіз та виділення певних рис для пошуку та супроводу.....	17
1.7 Поняття оптичного потоку.....	18
РОЗДІЛ 2. ВІДСТЕЖЕННЯ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ.....	25
2.1 Алгоритми відстеження об'єктів.....	25
2.1 Особливості і проблеми згорткової нейронної мережі.....	34
2.2 Трекінг відеооб'єктів.....	38
2.3 Методи після обробки.....	39
2.4 Багатокадрові методи.....	41
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ.....	48
2.5 Інструментарій розробки.....	48
3.2 Аналіз бібліотеки OpenCV.....	50
3.3 Морфологічні зміни зображення.....	53
3.4 Модифікація алгоритму розпізнавання об'єктів.....	57
3.5 Підвищення якості відеопотоку за рахунок використання багатопоточності.....	60
3.6 Тестування розробленої підсистеми.....	68

РОЗДІЛ 4. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ.....	74
4.1 Опис ідеї проекту.....	74
4.2 Технологічний аудит ідеї проекту.....	77
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	77
4.4 Аналіз ринкової стратегії проекту.....	85
4.5 Розроблення маркетингової програми стартап-проекту.....	88
ВИСНОВКИ.....	94
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	95

ВСТУП

З розвитком інформаційних технологій для систем відеоспостереження визначаються всі нові завдання, які полягають у розробці алгоритмів обробки відеоданих для автоматизованих режимів роботи систем. Це зумовлено двома потребами:

- зменшенням людського фактора в процесах, що забезпечують відеоспостереження та безпеку;
- визначенням найбільшої кількості інформаційних характеристик для забезпечення безпеки або усунення наслідків несанкціонованих порушень безпеки.

Численні вчені зробили значний внесок у обробку, якість передачі та захист відеопотоків в інфокомунікаційних системах.

Сучасні технології обробки відеоданих дозволяють розглядати сформульоване завдання, розділяючи її на дві підзадачі: окремо щодо забезпечення якості та забезпечення конфіденційності. Такий підхід дозволяє створювати захист відеопотоку незалежно від операцій стиснення.

Однією з ключових складових сучасних систем відеоспостереження є систематизація обробки кінцевих даних. Якість їх роботи безпосередньо впливає на роботу систем безпеки. Інтелектуалізація - одна з головних вимог сучасних систем відеоспостереження. Його суть полягає в тому, що система повинна не тільки захоплювати відеопотік, але і виконувати задану цільову обробку. Технічними вимогами цієї обробки є її автоматична (або автоматизована) робота в режимі реального часу. Вплив оператора повинен бути мінімальним. Очевидно, що розвиток таких систем вимагає розробки нових методів групових методів. Контекст групового методу стосується процедури декомпозиції проблеми системи відеоспостереження. Залежно від мети основне завдання системи поділяється на підзавдання, кожне з яких вирішується своїми методами, як правило, штучним

інтелектом або DataMinig. Визначення та вирішення найбільш розумних підзадач є головним ключем до успішного розвитку систем відеоспостереження.

Аналіз та обслуговування рухомих об'єктів є одним із ключових завдань сучасних систем відеоспостереження. Оскільки ці системи постійно вдосконалюються, важливо створювати нові алгоритми та модернізувати існуючі алгоритми, призначені для підтримки рухомих об'єктів у полі зору камер відеоспостереження, щоб підвищити рівень їх інтелектуалізації та адекватність. У цій публікації реалізований алгоритм обслуговування, заснований на побудові морфологічної моделі. Потік даних у телекомунікаціях та програмуванні - це послідовність закодованих когерентних сигналів (пакетів даних) у цифровій формі, яка використовується для передачі або прийому інформації при передачі.

В електронній та комп'ютерній архітектурі потік даних визначає тривалість часу, коли елемент даних може входити або виходити з системи. Потік даних часто розглядається як альтернатива потоку команд, оскільки машина фон Неймана розглядається як потік команд - тоді як її альтернатива управляється потоком даних.

РОЗДІЛ 1. ВИЯВЛЕННЯ ОБ'ЄКТІВ: НЕСКІНЧЕННА ТЕОРЕТИЧНА ПЕРСПЕКТИВА

1.1 Проблеми пошуку і супроводу об'єктів в відеофайлах

При виявленні і відстеженні об'єктів в відеофайли виникають наступні сім проблем:

- Зміна яскравості. Яскравість зображення може змінюватися в часі і просторі; деякі частини об'єкта можуть бути яскравішими, і яскравість об'єктів на поточному зображенні може стати гіршою під час повторного виявлення.
- Можливість розширювати або стискати об'єкт. Об'єкт може наблизитися або виходити з рамки об'єктиву камери нагляду
- Обертання об'єкта. Об'єкт може обертатися щодо камери у всіх осях тривимірного простору.
- Скринінг об'єктів. Об'єкт може бути перекритий іншими об'єктами.
- На відео можуть бути присутні спотворені об'єкти або шум. Шум на зображенні може виникати з природничих і штучно створених причин. Ці всі похибки можна віднести до некоректної роботи фото (відео) камери і всі природні умови, такі як снігопад чи дощ. Штучно створені причини включати всі дії людини, спрямовані на зміну якості зображення, такі як зміна колірного простору, стиснення зображення з втратою якості, розмитістю зображення та ін.
- Зміна геометричної форми об'єкту.

Спостережуваний об'єкт може деформуватися, оскільки він може повернутися до камери частиною, яка відрізняється від форми, що спостерігається раніше.

Об'єкт може складатися з невеликих взаємопов'язаних об'єктів, які можуть переміщатися в просторі по різних траєкторіях. Наприклад, бажаний об'єкт - це людина, руки якого можуть рухатися в різні боки. Змінилася колірна гістограма об'єкта. У об'єкта можуть бути різні колірні гістограми на різних боках його тіла. Масштабованість і повороти відносяться до афінних перетворень. Проектне перетворення включає в себе афінне перетворення і враховується спотворення, пов'язане з нахилом камери.

Контури та обриси інших об'єктів можуть помилково розташовуватися на задньому фоні. На рис. 1.1 ми бачимо відтворення похибки в визначенні об'єкту.



Рисунок 1.1 – Похибка в визначенні об'єкта.

1.2 Способи зображення об'єктів

Як ми всі знаємо, відео - це набір окремих зображень, які постійно відображаються для створення ефекту руху. Таким чином, виявлення об'єктів в

відеофайлах також засноване на ідеї виявлення об'єктів у файлі зображення. Нам потрібно виконати етап поділу зображення, щоб визначити, де площа об'єктів ϵ , а де площа фону. Сегментація зображень – дуже важливий крок в обробці зображення, і цей етап аналізує зображення на компоненти однакового характеру, заснованих на кордонах або взаємопов'язаних регіонах. Образ - це деталь, певний об'єкт в панорамі. Область зображення описує властивості поверхні зображення; ця область оточена кордоном і точками з відносно однорідним рівнем сірого. Для фізичних властивостей області зображення ми визначаємо ряд методів поділу.

- Основні методи поділу включають:
- Класифікаційний або пороговий метод.
- Структурний метод.
- Граничний метод.

Виявлення об'єктів в відеофайлах має ґрунтуватися на історії руху об'єкта над дискретним зображенням з плином часу. Даний підхід, заснований на історії рухомих зображень, - це погляд, заснований на методі виявлення закономірностей в часі. Це простий, але ефективний спосіб відображення руху; він використовується в ряді досліджень, пов'язаних з розпізнаванням руху та аналізом руху.

Огляд виявлення, класифікації та розмітки відео в реальному часі показують, що ця система може реалізовувати такі операції:

- розрізняти перехідні процеси і стаціонарні об'єкти а також статичні фонові об'єкти в сцені;
- виявляти і розрізняти об'єкти, які рухаються і зникають;
- класифікувати об'єкти, виявлені в різних групи, такі як люди, групи, автомобілі та інші;
- зазначити об'єкти і створити інформаційний потік про шляхи в відеозображенні.

Додатки, засновані на обробці відео, мають загальні характеристики. Зазвичай використовуються технології виявлення рухомих об'єктів - віднімання фону, статичні методи, часова різниця, оптичний потік і інші.

1.3 Метод вичитування фону

Віднімання фону - це частина технології відслідковування руху в статичних сценах. Даний метод намагається виявляти області руху шляхом віднімання (вичитування) пікселів з поточного зображення в фонове зображення, за період одного циклу ініціалізації.

Є кілька підходів, які можна продемонструвати.

Нехай $I_n(x, y)$ буде представленим значенням інтенсивності відтінків сірого в місці розташування пікселя (x, y) зі значенням $[0 \div 255]$ в n -му випадку відеопослідовності I . Нехай $B_n(x)$ буде відповідним значенням величини фону для пікселя в момент часу (x, y) , оцінене з відеозображення з I_0 по I_{n-1} . Піксель в позиції (x, y) в поточному зображенні належить домінуючому компоненту. Якщо

$$|I_t(x, y) - B_t(x, y)| > T_n(x, y) \quad (1.1)$$

виконується, де $T_n(x, y)$ - зумовлений поріг.

Фонове зображення B_t оновлюється за допомогою фільтра нескінченної імпульсної характеристики:

$$B_{t+1} = \alpha \cdot I_t + (1 - \alpha) \quad (1.2)$$

Цей фон оновлюється для всіх типів пікселів. У фонових методах віднімання, вихідне фонове зображення оновлюється лише для пікселів у фоновому режимі.

1.4 Статичні методи

Даний метод використовує статичний фоновий візерунок, де кожен піксель знову представлений за допомогою найменшого значення (M) і його максимальним значенням (N) з точки зору інтенсивності і максимуму різниці в інтенсивності (D) між будь-якими послідовними кадрами, які спостерігаються під час тренування циклів ініціалізації, в якому сцена не містить рухомих об'єктів. Поточне зображення класифікується як унікальне, якщо воно задовольняє:

$$|M(x, y) - I_t(x, y)| > D(x, y) \text{ чи } |N(x, y) - I_t(x, y)| > D(x, y) \quad (1.3)$$

Кожен піксель моделюється сумішшю методів Гаусса та оновлюється онлайн за допомогою вхідних даних зображення. Розподіл Гаусса оцінить, чи належить точка до домінуючої точкової обробки або базової обробки.

1.5 Різниця в часових проміжках

Різниця в часі намагається виявити області руху, використовуючи різницю між пікселями в послідовних кадрах (два або три) в послідовності відеокліпів. Цей метод легко адаптується до сцени; проте це часто викликає помилки при виявленні рухомих об'єктів. Одноколірна область іншої сторони змушує алгоритм різниці в часі отримувати всі пікселі під час руху об'єкта.

Теорія ідентифікації та подання зображень була запропонована А. Vobick, та J. Davis. Зокрема, ця теорія аналізує ідентифікацію на основі руху разом з описом просторового елементу і описує, як рухається об'єкт. Він являє собою сформоване «рухоме енергетичне зображення» (РЕЗ) або «область довічного руху» (ОДР). В даній теорії ми можемо бачити, що вона являє собою. Позиції передачі доступні в послідовності зображень. МЕІ є формою руху і просторового розділення руху. Потім створюється ЗІР (зображення історії руху). Інтенсивність кожного пікселя в ЗІР представляє інтервал кількості, яке може кодуватися кожним кадром, і цей метод ЗІР буде пов'язувати тимчасову частку людських жестів. Якщо скласти РЕЗ і ЗІР разом, ми побачимо, що ЗІР і РЕЗ можна розглядати як двокомпонентні версії зразка з плином часу.

Піксель - це ряд функцій, певні конкретні зразки, які будуть конкурувати з шаблонами зберігання відомих видів руху. Об'єднання моделей РЕЗ і ЗІР побудовано з використанням методу ЗІР з історичним рухом функції НТ (x, y, t). Обчислюється вона з різниці в зображенні і оптичному потоці.

На найпростішому рівні кожен об'єкт ідентифікується і представлений певною рамкою. Рис 1.2. При переміщенні об'єкта гранична рамка також переміщується. Різниця в зображенні створюється всіма зображеннями в області обмежувальної рамки. Поєднання з існуючими шаблонами кінцево нададуть тип об'єкта, який потрібно ідентифікувати.



Рис. 1.2. Ілюстрація об'єкта в обмежувальній рамці.

На більш детальному рівні кожен об'єкт ідентифікується областю зображення, яка відрізняється від фону (межа буде мати форму об'єкта). Рух об'єкта контролюється шляхом вирахування різниці в оптичному струмі.

Ілюстрація зроблена шляхом перетворення в двійкове зображення, як показано на рис. 1.3.

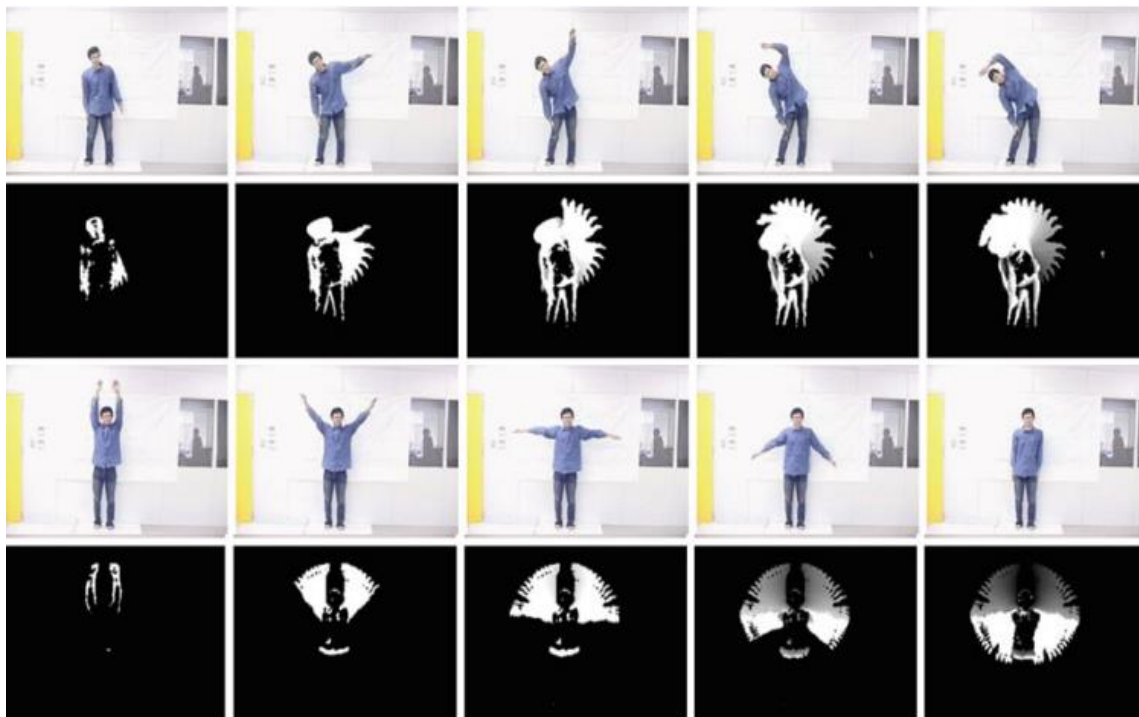


Рис. 1.3. Розробка образів МНІ для двох різних дій. Отримані зображення МНІ показані під діями послідовності.

1.6 Аналіз та виділення певних рис для пошуку та супроводу

Виділення правильних рис, та контурів об'єкту грає важливу роль у пошуку об'єктів. В загальному випадку, візуальні риси повинні мати унікальність, щоб об'єкт мав певну неповторність в порівнянні з іншими. Виділення таких рис дуже пов'язано з представленням об'єкта. Так границі об'єкта є характерними рисами для представлення контуром. Основні візуальні риси:

- видимий колір, представлений в діапазонах RGB чи HSV (Hue, Saturation, Value);

- краї та кордони об'єкта створюють певні зміни в інтенсивності та чіткості зображення. Розпізнавання границь використовуються для

знаходження цих змін. Досить важлива властивість границь – вони не такі чутливі до змін світла порівнюючи їх з рисами кольору. Одним з популярних детекторів є детектор Canny;

– оптичний потік – це уявне представлення сліду руху об'єктів, поверхонь, та країв сцени яку ми спостерігаємо, під час відносного руху між спостерігачем і сцени.

– текстура – це певна міра змінювання інтенсивності поверхні, яка має такі характеристики: *рівність та постійність*.

1.7 Поняття оптичного потоку

Оптичний потік постійно обробляється в нашій візуальній системі і може використовуватися для вирішення різних завдань. Такі завдання включають оцінку саморуку, сегментацію сцени на незалежно рухомі об'єкти і жорсткі частини або передній план і фон. Крім того, оптичний потік містить інформацію про час контакту з місцями в навколишньому середовищі, виведеними на зображенні. Оптичний потік також дозволяє оцінити відносну глибину всіх видимих і твердих об'єктів. Інформація, витягнута з оптичного потоку, може використовуватися в системах допомоги водієві для виявлення інших автомобілів і їх вектору напрямленості, пішоходів і руху автомобіля. Інше застосування - виявлення інших літаків для запобігання зіткнень в повітряному просторі. Оптичний потік використовується в відеокодеках для інтерполяції зображення між ключовими кадрами. Швидкі дисплеї з високою роздільною здатністю використовують оптичний потік для синтезу додаткових кадрів зображення в якості інтерполяції між існуючими. Часто більш висока частота кадрів неможлива через з'єднання з дисплеєм з обмеженою пропускнуою здатністю, і тому використовується цей метод інтерполяції. Оптичний потік може бути виявлений

за допомогою датчиків камери, які дешевше, ніж світлові радари, і пасивні в їх сприйнятті.

Оптичний потік визначається як зміна структурованого світла на зображенні, наприклад на сітківці або датчику камери через відносний рух між оком або камерою і сценою. Подальші визначення з літератури підкреслюють різні властивості оптичного потоку.

Гельмгольца: «Я теж вважаю, що одноокі люди можуть формувати правильні уявлення про матеріальні форми свого оточення в основному за рахунок варіацій зображення на сітківці ока через рухи тіла». (Гельмгольц, 1925, с. 297)

Гібсон: «З аналітичної точки зору це повне перетворення масиву, мабуть, означає, що елементи цієї текстури зміщені, причому елементи розглядаються як плями. З інтроспективної точки зору поле всюди буває рухом, коли рухається спостерігач ». (Гібсон, 1966, стор. 196f.)

Хорна: «рух яскравості, що спостерігається, коли камера рухається щодо об'єктів зображення, називається оптичним потоком». (Хорн, 1986, с. 278)

Гельмгольца в основному турбує сприйняття глибини, і він описує оптичний потік як «варіації зображення сітківки», які відбуваються через рух тіла і залежать від структури, а саме відстані, а також жорсткості навколишнього середовища. Гібсон описує зсув структури в оптичній матриці як перетворення, яке «оживає при русі, коли рухається спостерігач». Більш нове визначення оптичного потоку дано Хорном, який приписує «рух яскравості патернів» в зображенні відносним рухом між спостерігачем і об'єктами в навколишньому середовищі.

У біологічному контексті потік сітківки - це зміна структурованих світлових патернів на сітківці, які призводять до відчуття руху візуальних образів, що

проектуються на сітківку. На рис. 1.4(а). зображено створення оптичного потоку сітківки для усунення двох типових візуальних особливостей. З технічної точки зору і в контексті комп'ютерного зору зміни навколишнього середовища на зображенні представлені серією кадрів зображення. На малюнку 1.4.(б) показані три кадри послідовності зображень, які можуть бути отримані шляхом просторової і тимчасової вибірки падаючого світла. Оптичний потік фіксує зміну цих зображень через векторне поле. У дослідженнях робиться наголос на точну оцінку оптичного потоку по пікселям, що є складним обчислювальним завданням. В даний час оптичний потік можна оцінити майже в реальному часі для розумного дозволу зображення.

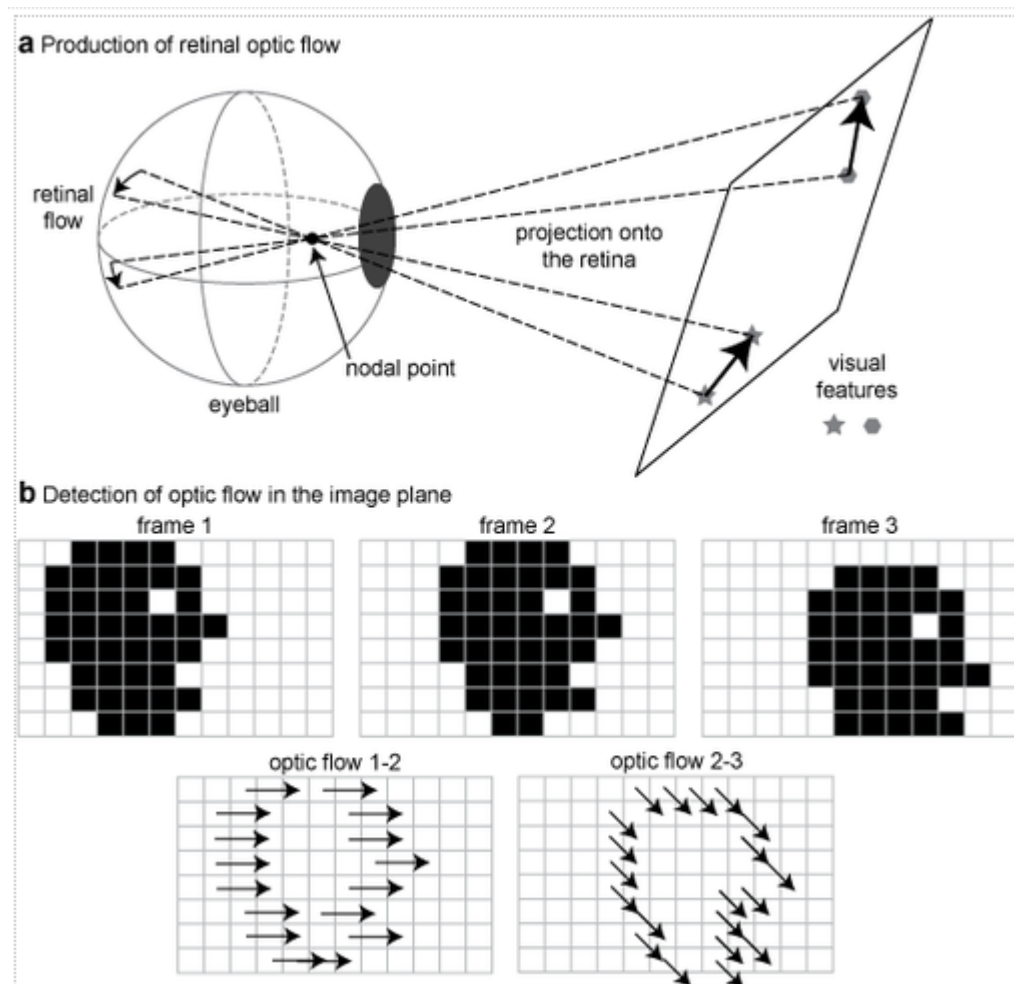


Рис1.4(а) оптичний потік створюється на сітківці за рахунок зміни структури світла. У прикладі показано зміщення двох візуальних елементів (зірки і шестикутника) на площині і їх кутові зміщення на поверхні очного яблука.

(б) якщо структурований світ дискретизується просторово і тимчасово, це призводить до послідовності зображень. У прикладі показані три кадри, які показують рух силуету голови. Оптичний потік зображується як відповідність контурних пікселів між кадрами 1 і 2, а також кадрами 2 і 3. Для методів оцінки потоку завдання полягає в тому, щоб знайти відповідність точок для кожного пікселя в зображенні, а не тільки контурних пікселів.

Алгоритм Лукаса-Канаді

Алгоритм оптичного потоку Лукаса-Канаді - це простий метод, який може дати оцінку руху місць послідовних зображень в сцені. Ми можемо зв'язати вектор руху (u, v) з кожним таким пікселем в сцені, отриманий шляхом порівняння двох послідовних зображень.

Алгоритм Лукаса-Канаді робить деякі неявні припущення:

- Два зображення розділені невеликим кроком часу Δt , в такому способі, яким об'єкти не змістилися (тобто алгоритм найкраще працює з повільно рухомими об'єктами).
- Зображення зображують природну сцену, яка містить текстуровані об'єкти, що мають плавно змінюватися відтінками сірого (різними рівними інтенсивностями).

Алгоритм не використовує інформацію про колір. Це не методика сканування другого зображення в пошуках відповідності заданому пікселю. Даний алгоритм працює намагаючись вгадати, в якому напрямку перемістився об'єкт, щоб локальні зміни за інтенсивністю можна було пояснити.

Також він не такий чутливий до шуму на зображеннях, в порівнянні з потоковими методами, але він являється суто локальним і не може визначати напрямки руху пікселів в областях.

Алгоритм обчислюється за допомогою диференціальних рівнянь та базується на обчисленнях похідних в горизонтальному і вертикальному напрямку.



Рисунок 1.5 – Демонстрація роботи алгоритму.

Оптичний потік Гуннара-Фарнбака

Оптичний потік відомий як зразок видимого руху об'єктів, тобто це рух об'єктів між кожними двома послідовними кадрами, яке викликане рухом об'єкта зйомки або камерою, захоплюючою його. Розглянемо об'єкт з інтенсивністю $I(x, y, t)$, з часом dt він переміститься на dx і dy , тепер нова інтенсивність буде:

$$I(x + dx, y + dy, t + dt) \quad (1.3)$$

Ми припускаємо, що інтенсивності пікселів постійні між двома кадрами, тобто:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (1.4)$$

У щільному оптичному потоці ми бачимо всі точки (на відміну від Лукаса Канаді, який працює тільки з кутовими точками, виявленими алгоритмом Ши-Томазі), при цьому виявляємо зміни інтенсивності пікселів між двома кадрами, в результаті чого після перетворення виходить зображення з виділеними пікселями в форматі hsv для чіткої видимості.

Даний алгоритм обчислює величину і напрямок оптичного потоку з масиву векторів потоку, тобто $(dx / dt, dy / dt)$. Пізніше він візуалізує кут (напрямок) потоку по відтінку і відстань (величину) потоку за значенням колірною уявлення HSV. Для забезпечення оптимальної видимості сила HSV встановлена на 255. OpenCV надає функцію `cv2.calcOpticalFlowFarneback` для перегляду в щільному оптичному потоці.

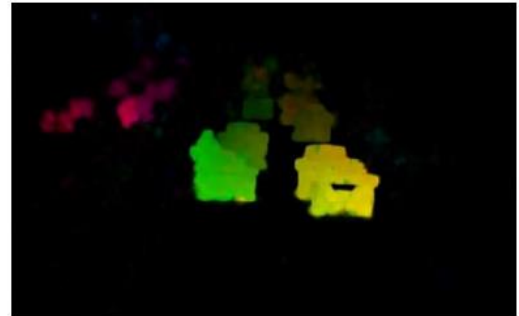


Рисунок 1.4 – Демонстрація роботи алгоритму Гуннара-Фарнбака.

Алгоритм Horn–Schunck

Пропозиція Хорна і Шунка складається в постановці завдання оцінки оптичного потоку в вигляді варіаційної задачі, в якій шукане векторне поле h визначається як мінімізатор деякої енергії функціоналу $J(h)$. У цьому функціоналі є два терміни: термін приєднання даних, що задається оптичним потоком, обмеження і член регулярності, заснований на градієнті потоку.

$$E = \int [(I_x u + I_y v + I_t)^2 + a^2 (|\nabla u|^2 + |\nabla v|^2)] dx dy \quad (1.5)$$

Сам алгоритм був запропонований в 1981 році. Пропонується використовувати multi-scaling. Але даний алгоритм складний в реалізації і трудомісткий, при чому з ємнісними зображеннями даний алгоритм використовувати не є вигідно.

Алгоритм Horn–Schunck має часову складність порядку $\Theta(SN)$, де S виступає кількістю пікселів у зображенні, N виступає кількістю ітерацій методу Гауса-Зейделя.

Оптичний потік і його проблеми вивчаються вже декілька десятиліть, удосконалюючи методи ідентифікації об'єктів, їх супровід та пошук. Проблема являється досить непростою в реалізації, адже від якості визначення зміщень в опрацьованні відеоданих залежить ефективність інших алгоритмів.



Рисунок 1.5 – Демонстрація роботи методу Horn–Schunck.

Висновки за розділом

В цьому розділі було розгорнуто проблеми та аналізи методів пошуку і супроводу об'єктів, їх ідентифікації. Представлено роботу алгоритмів які застосовуються для пошуку конкретних об'єктів. Були перераховані методи пошуку від самих дієвих до більш простих. Проведено аналіз алгоритмів, при цьому аналізі стало зрозуміло що самим дієвим та найпопулярнішим із алгоритмів ідентифікацій є метод Віоли-Джонса. Сам метод є найдієвішим і самим точним при виявленні об'єктів, так як використовується каскадний класифікатор. Зі списків алгоритмів можна виділити метод «YOLO». По своїм параметрам та способам він також заслуговує уваги, і може конкурувати з методом Віоли-Джонса.

РОЗДІЛ 2. ВІДСТЕЖЕННЯ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ

2.1 Алгоритми відстеження об'єктів

Метод виявлення об'єкта став свідком швидких революційних змін в області комп'ютерного зору. Його участь в поєднанні класифікації об'єктів і їх локалізації робить цю тему однією з найбільш складних тем в області комп'ютерного зору. Простіше кажучи, мета цього методу виявлення полягає в тому, щоб визначити, де знаходяться об'єкти на даному зображенні, зване «локалізацією об'єкта», і до якої категорії належить кожен об'єкт.

1) *Fast R-CNN*

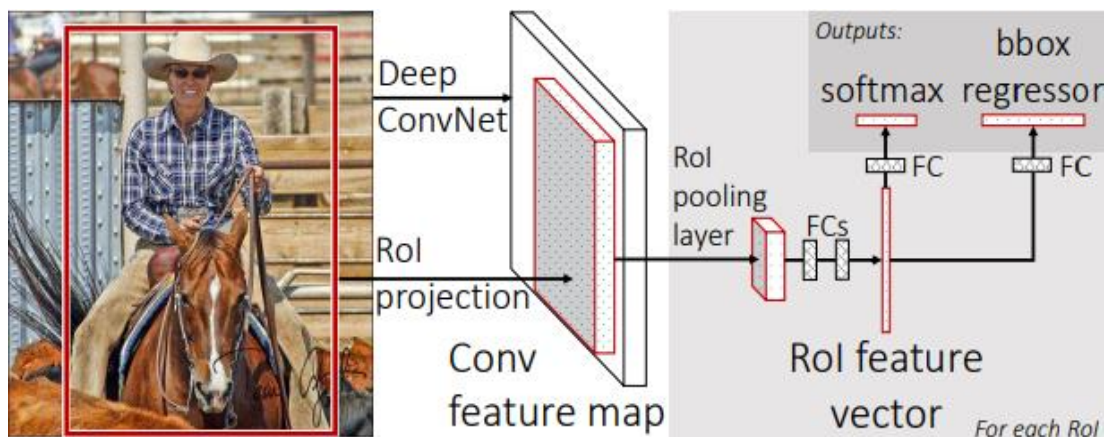


Рис 2.1 Приклад «Fast R-CNN» методу.

Переваги Fast R-CNN:

- більш висока якість виявлення (MAP), ніж R-CNN, SPPnet
- навчання одноетапне, з багатозадачним приміненням
- навчання може оновлювати всі мережеві рівні
- для кешування даних дискове сховище не потрібно

2) *Faster R-CNN*

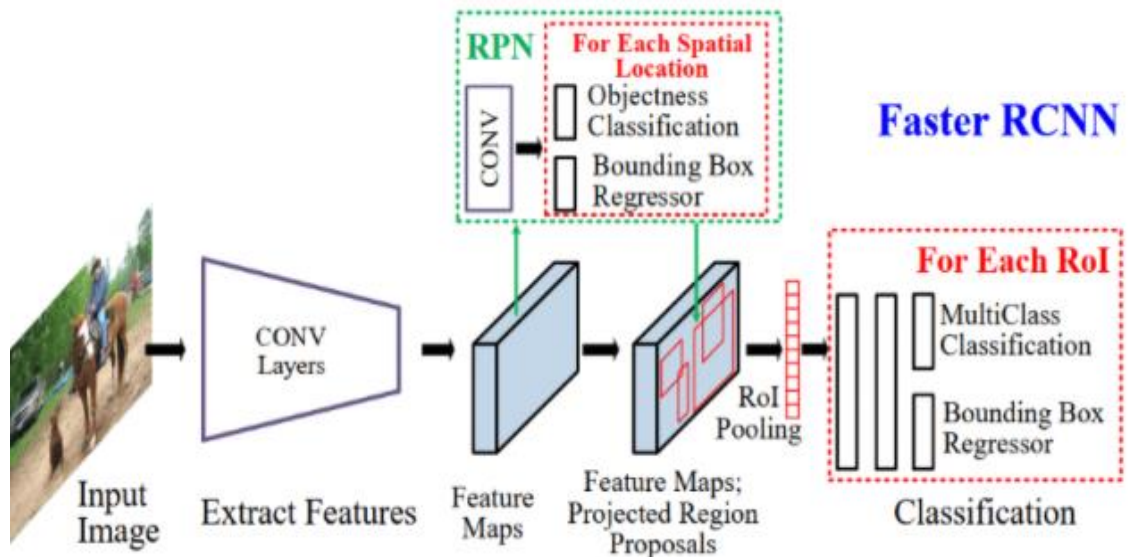


Рис 2.2 Приклад «Faster R-CNN» методу.

Faster R-CNN це алгоритм виявлення об'єктів, схожий на R-CNN. У цьому алгоритмі використовується мережа регіональних пропозицій (RPN), яка використовує згорткові функції повного зображення з мережею виявлення більш економічним способом, ніж R-CNN і Fast R-CNN. Мережа пропозицій регіону - це, по суті, повністю згорткова мережа, яка одночасно прогнозує межі об'єкта, а також оцінки об'єктивності в кожній позиції об'єкта і проходить наскрізне навчання для генерації високоякісних пропозицій регіонів, які потім використовуються Fast R -CNN для виявлення об'єктів.

3) Histogram of Oriented Gradients (HOG)

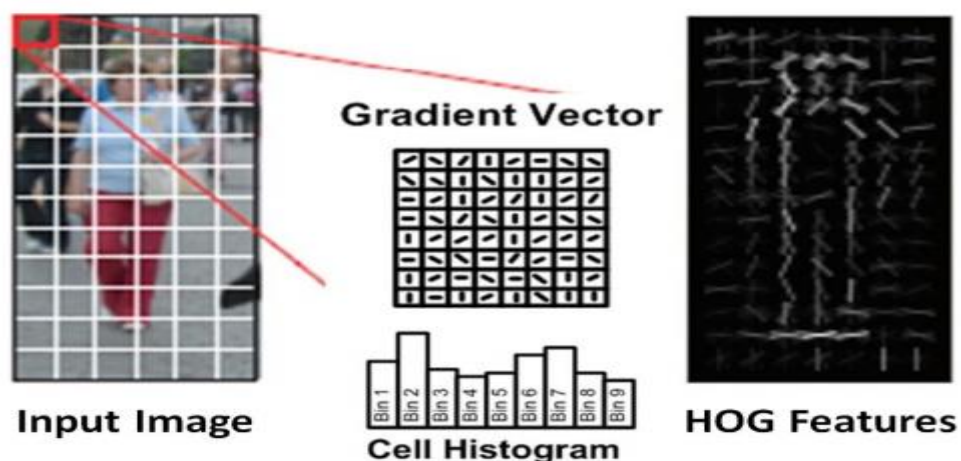


Рис 2.3 Приклад методу гістограми орієнтованих градієнтів (HOG).

Гістограма орієнтованих градієнтів (HOG) - це в основному дескриптор функції, який використовується для виявлення об'єктів при обробці зображень і інших методах комп'ютерного зору. Метод дескриптора гістограми орієнтованих градієнтів включає в себе входження орієнтації градієнта в локалізованих частинах зображення, таких як вікно виявлення, яка цікавить область (ROI), серед іншого. Одним з переваг HOG-подібних функцій є їх простота і більш легке розуміння інформації, яку вони несуть.

4) *Region-based Convolutional Neural Networks (R-CNN)*

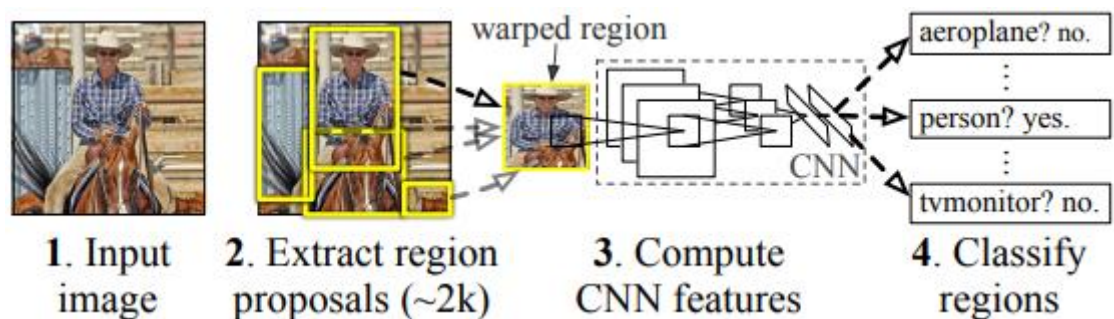


Рис 2.4 Приклад методу згорткової мережі на основі нейронних обчислень.

Метод згорткової мережі на основі регіонів (RCNN) являє собою комбінацію пропозицій регіонів зі згортковими нейронними мережами (CNN). R-CNN допомагає в локалізації об'єктів за допомогою глибокої мережі і навчанні високопродуктивної моделі тільки з невеликою кількістю анотованих даних виявлення. Він забезпечує чудову точність виявлення об'єктів за рахунок використання глибокої нейронної мережі для класифікації пропозицій по об'єктах. R-CNN може масштабуватися до тисячі класів об'єктів, не вдаючись до приблизних методів, включаючи хешування.

5) *Region-based Fully Convolutional Network (R-FCN)*

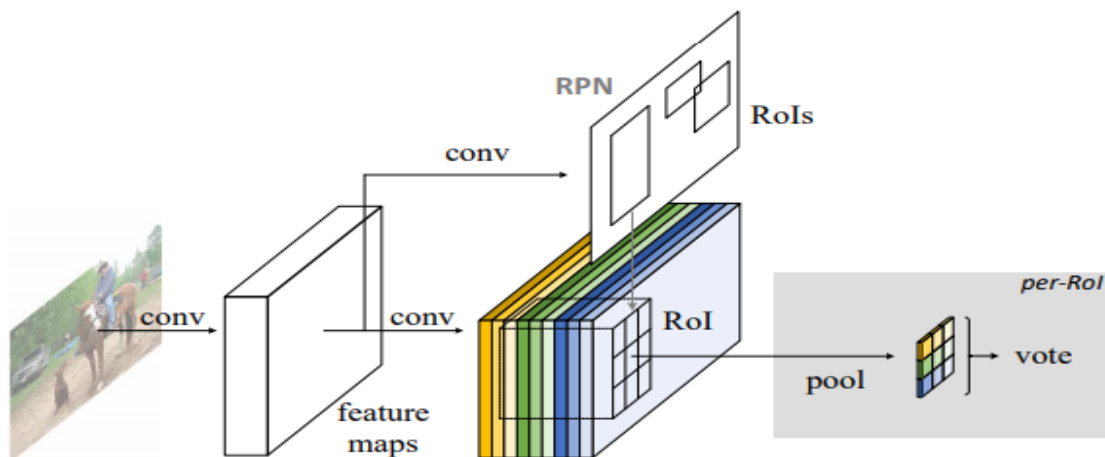


Рис 2.5 Приклад методу повністю згортковій мережі на основі області.

Метод повністю згортковій мережі на основі області або R-FCN - це детектор на основі області для виявлення об'єктів. На відміну від інших детекторів, які застосовують дорогу підмережу для кожної області, таку як Fast R-CNN або Faster R-CNN, цей детектор на основі області є повністю згортковим, і майже всі обчислення застосовуються для всього зображення.

R-FCN складається із загальних, повністю згортальних архітектур, як і в разі FCN, який, як відомо, дає кращий результат, ніж Faster R-CNN. У цьому алгоритмі всі вагові рівні є згортковими і призначені для класифікації областей інтересу за категоріями об'єктів та фонам.

6) Single Shot Detector (SSD)

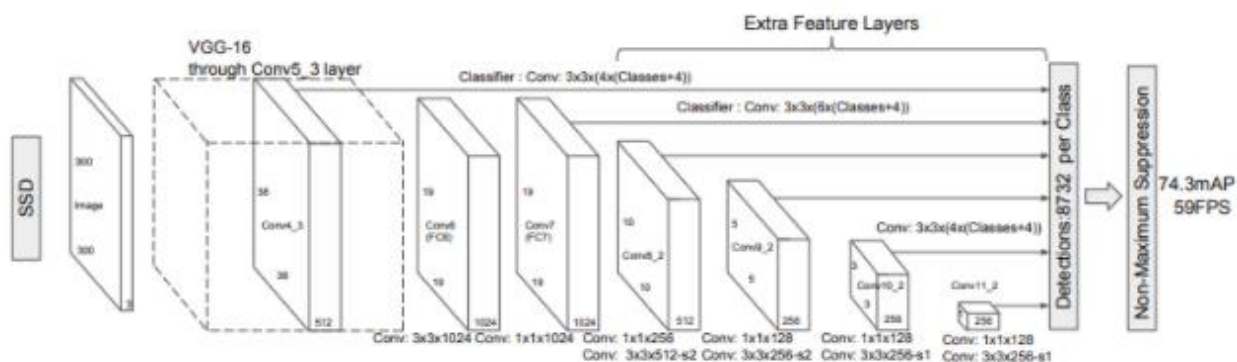


Рис 2.6 Приклад методу детектору.

Single Shot Detector (SSD) - це метод виявлення об'єктів на зображеннях з використанням однієї глибокої нейронної мережі. Підхід SSD дискретизує вихідний простір обмежувальних рамок на набір рамок за замовчуванням з різними співвідношеннями сторін. Після дискретизації метод масштабується відповідно до розташування на карті об'єктів. Мережа Single Shot Detector об'єднує прогнози на основі декількох карт функцій з різним дозволом для природної обробки об'єктів різних розмірів.

Переваги SSD:

- SSD повністю виключає створення пропозиції і наступні етапи повторної вибірки пікселів або функцій та інкапсулює всі обчислення в єдиній мережі;
- легко навчити і легко інтегрувати в системи, що вимагають компонент виявлення;
- SSD має конкурентоспроможну точність в порівнянні з методами, які використовують додатковий етап пропозиції об'єкта, і він набагато швидше, забезпечує єдину структуру як для навчання, так і для виведення.

7) *Spatial Pyramid Pooling (SPP-net)*.

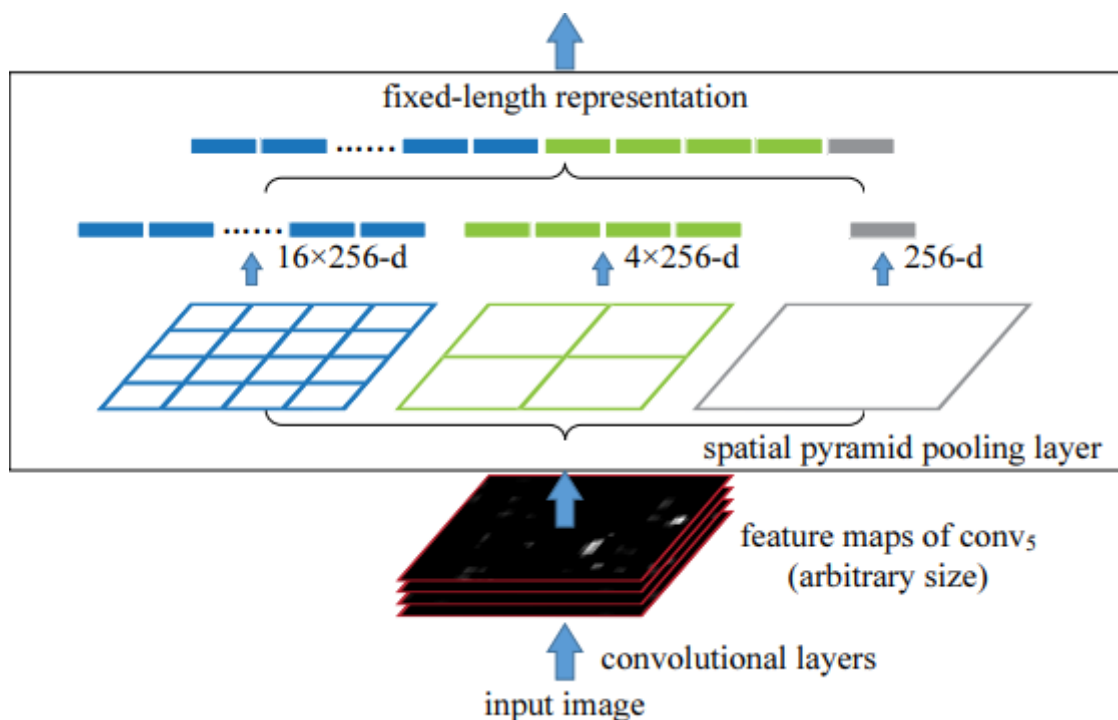


Рис 2.7 Метод Об'єднання просторових пірамід (SPP-net).

Об'єднання просторових пірамід (SPP-net) - це мережева структура, яка може генерувати уявлення фіксованої довжини незалежно від розміру або масштабу зображення. Вважається, що об'єднання пірамід стійке до деформацій об'єктів, а SPP-net покращує всі методи класифікації зображень на основі CNN. Використовуючи SPP-net, дослідники можуть обчислити карти характеристик з усього зображення тільки один раз, а потім об'єднати функції в довільних областях (суб-зображеннях) для створення уявлень фіксованої довжини для навчання детекторів. Цей метод дозволяє уникнути повторного обчислення згортальних функцій.

8) YOLO (You Only Look Once)

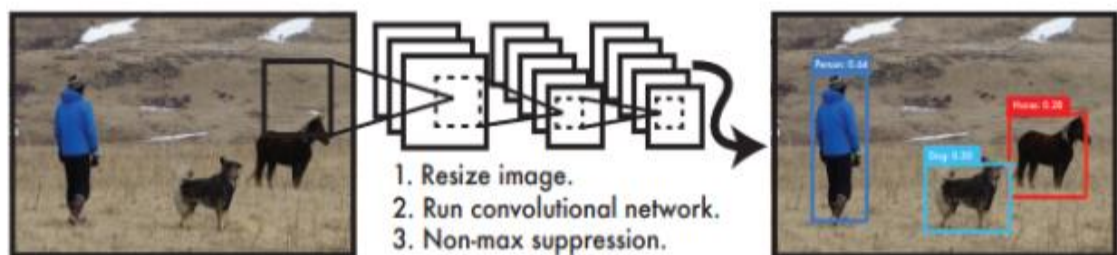


Рис 2.8 Метод YOLO

You Only Look Once або YOLO - один з найпопулярніших алгоритмів виявлення об'єктів, які використовуються дослідниками по всьому світу. За словами дослідників з Facebook AI Research, уніфікована архітектура YOLO працює дуже швидко. Базова модель YOLO обробляє зображення в режимі реального часу зі швидкістю 45 кадрів в секунду, в той час як молодша версія мережі Fast YOLO обробляє вражаючі 155 кадрів в секунду, при цьому досягаючи при цьому вдвічі більшого, ніж у інших детекторів реального часу, MAP. Цей алгоритм перевершує інші методи виявлення, включаючи DPM і R-CNN, при узагальненні від природних зображень на інші області, такі як твори мистецтва.

9) Метод Віоли-Джонса

Розроблена Полом Віолою і Майклом Джонсом ще в 2001 році, система виявлення об'єктів Віоли-Джонса може швидко і точно виявляти об'єкти на

зображеннях і особливо добре працює з людським обличчям (Viola & Jones, 2001). Незважаючи на свій вік, фреймворк як і раніше залишається провідним гравцем в області виявлення осіб поряд з багатьма аналогами CNN. Платформа виявлення об'єктів Віоли-Джонса поєднує в собі концепції Хаара-подібних функцій, інтегральних зображень, алгоритму AdaBoost і каскадного класифікатора для створення системи для швидкого і точного виявлення об'єктів. Таким чином, щоб зрозуміти структуру, ми спочатку повинні зрозуміти кожен з цих концепцій індивідуально, а потім з'ясувати, як вони з'єднуються один з одним, щоб сформувати структуру.

Алгоритм AdaBoost (Adaptive Boosting) - це алгоритм машинного навчання для вибору кращої підмножини функцій серед усіх доступних опцій. Результатом роботи алгоритму є класифікатор (він же функція прогнозування, функція гіпотез), званий «сильним класифікатором». Сильний класифікатор складається з лінійних комбінацій «слабких класифікаторів» (кращих характеристик). На високому рівні, щоб знайти ці слабкі класифікатори, алгоритм запускається для T ітерацій, де T - кількість слабких класифікаторів, які потрібно знайти, і воно встановлюється вами. На кожній ітерації алгоритм знаходить частоту помилок для всіх функцій, а потім вибере з найнижчою частотою помилок для цієї ітерації.

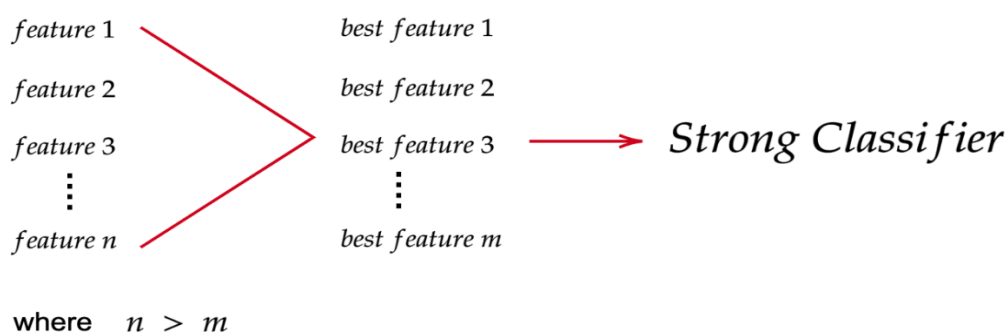


Рис 2.9 Мета використання алгоритму AdaBoost - дістати кращі функції з n функцій. Примітка: кращі функції також відомі як слабкі класифікатори.

Каскадний класифікатор - це багатоступінчастий класифікатор, який може виконувати ідентифікацію швидко і точно. Кожен етап складається з сильного

класифікатора, створеного алгоритмом AdaBoost. Від етапу до етапу кількість слабких класифікаторів в сильному класифікаторі збільшується. Вхідні дані оцінюються послідовно (поетапно). Якщо класифікатор для певного етапу дає негативний результат, введення негайно відкидається. Якщо вихід позитивний, вхід направляється на наступний етап. Згідно Віолі і Джонсу (2001), цей багатоетапний підхід дозволяє створювати більш прості класифікатори, які потім можна використовувати для швидкого відхилення більшості негативних (без особи) вхідних даних, приділяючи більше часу позитивним (лицьовим) вхідним даним.

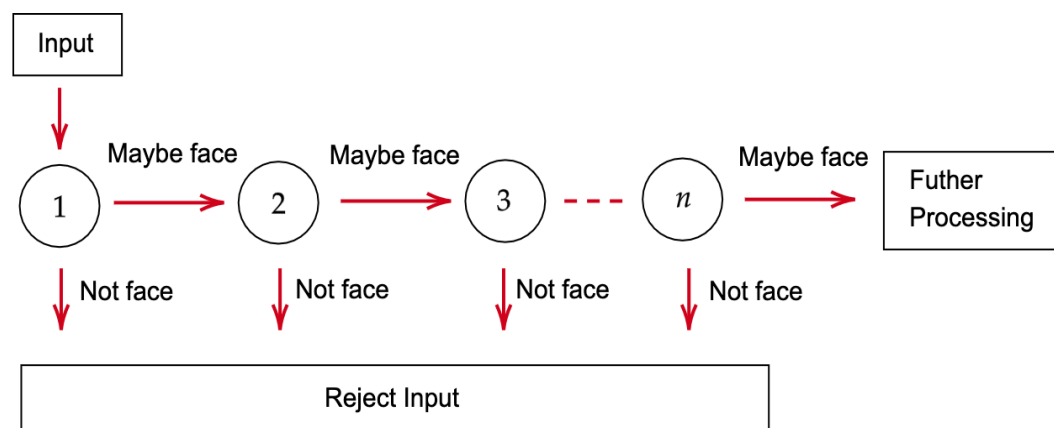


Рис 2.10 Каскадний класифікатор

Дізнавшись про основні концепції які використовуються в середі ідентифікацій об'єктів, дізнаємося як ці концепції працюють разом.

Машинне навчання

Мета цього етапу - створити каскадний класифікатор для особи, який зможе точно класифікувати особу і швидко відкинути об'єкт не маючий людських рис. Щоб домогтися цього, потрібно спочатку підготувати свої навчальні дані, а потім побудувати каскадний класифікатор, використовуючи модифікований алгоритм AdaBoost для цих навчальних даних.

Припускаючи, що у вас вже є навчальний набір, що складається з позитивних зразків (осіб) і негативних зразків (чи не осіб), першим кроком є отримання функцій з цих зразків зображень. Віола і Джонс (2001) рекомендують зображення розміром 24×24 . Оскільки кожен тип об'єктів типу Хаара може мати різні розміри і положення у вікні 24×24 , можна виділити понад 160 000 об'єктів, подібних Хаара. Проте, на цьому етапі необхідно розрахувати всі 160000+ характеристик Хаара. На щастя, впровадження інтегральних образів допомагає прискорити цей процес. На рисунку 2.9 показаний весь процес підготовки даних.

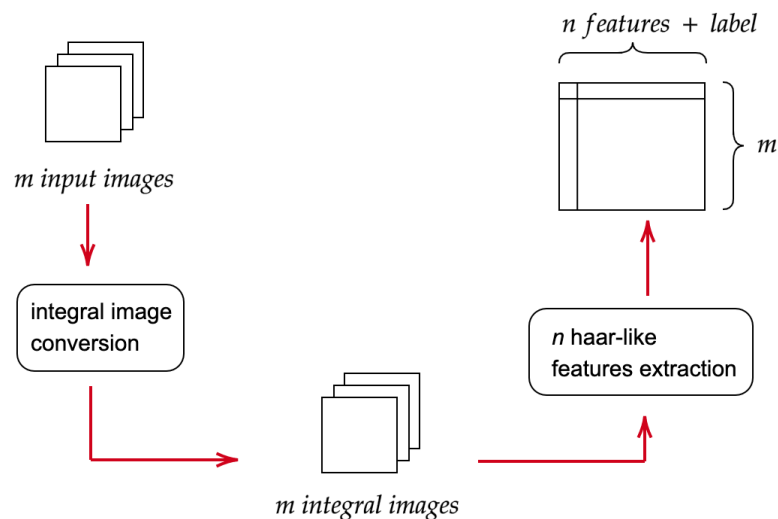


Рис 2.11 Процес підготовки даних

Побудова каскадного класифікатора з модифікованим алгоритмом AdaBoost.

Зрозуміло, що використання всіх 160000+ функцій безпосередньо з обчислювальної точки зору неефективно. Віола і Джонс (2001) запропонували два рішення, які можуть вирішити цю проблему. По-перше, за допомогою алгоритму AdaBoost скоротити кількість функцій до кількох корисних функцій. По-друге, розділити функції які залишилися на етапи та оцінити кожне введення поетапно (каскадно). Віола і Джонс (2001) розробили модифіковану версію алгоритму

AdaBoost, щоб мати можливість навчати каскадний класифікатор. На рисунку 2.10 показана спрощена версія алгоритму, представленого Ramsri.

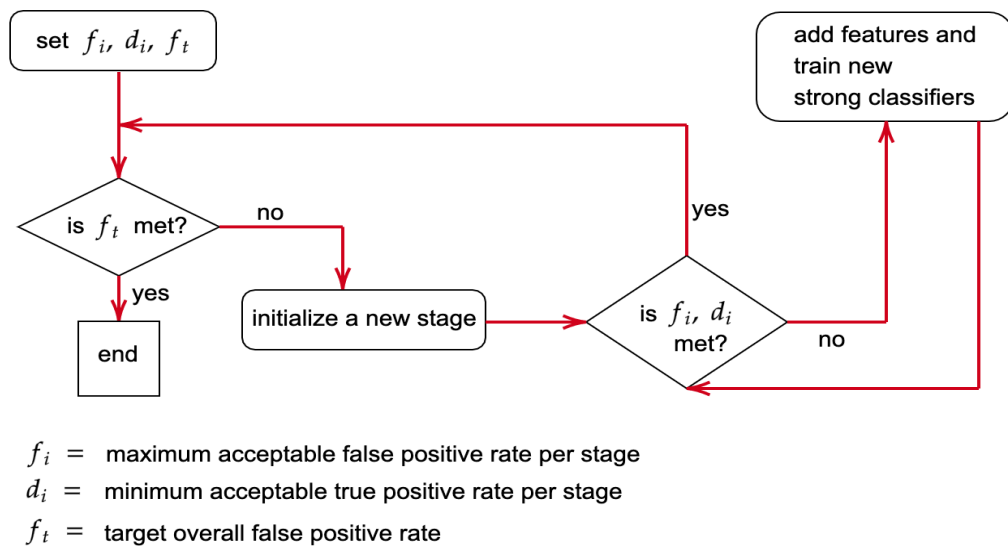


Рис 2.12 Процес побудови каскадного класифікатора

2.1 Особливості і проблеми згорткової нейронної мережі

Відстеження частинок - це потужний біофізичний інструмент, який вимагає перетворення досить ємкісних відеофайлів в тимчасові ряди положень, тобто сліди цікавих видів для аналізу даних. Сучасні методи спостереження, засновані на обмеженому наборі вхідних параметрів для ідентифікації яскравих об'єктів, погано пристосовані для обробки спектра просторово-часової неоднорідності і поганого відношення сигналу або шуму, зазвичай властивого субмікронних видам в складних біологічних середовищах. Для оптимізації та виконання методів відстеження часто необхідна активна участь користувачів, що не тільки неефективно, але й призводить до упередженості користувачів. Щоб розробити повністю автоматизований метод відстеження, було розроблено згорткову нейронну мережу для локалізації частинок на основі даних зображення, що містить більше 6000 параметрів, і взяли в використання методи машинного навчання для навчання мережі в різноманітному портфелі умов відео.

Відстеження нейронної мережі забезпечує безпрецедентну автоматизацію і точність, з виключно низьким рівнем хибно-позитивних і помилково негативних результатів як для 2D, так і для 3D моделювання відео і 2D експериментальних відео з важко відслідковуваними видами.

Основні труднощі при оптимізації методів відстеження для конкретних експериментальних умов - це доступ до «наземної істини», отримання якої може бути досить суб'єктивним і трудомістким процесом. Один з підходів до застосування методу відстеження до експериментальних відеороликів полягає в налаштуванні значень параметрів вручну при якісній оцінці помилок в ряді відеороликів. Процедура трудомістка і суб'єктивна. Кращим підходом з використанням кількісної оптимізації є створення змодельованих відеороликів для яких достовірно відомо які максимально відповідають спостережуваним експериментальним умовам. Потім даний метод відстеження, відповідний для цих умов, може бути застосований до змодельованих відео, і помилка може бути оцінена кількісно. Кількісно оцінюючи помилку відстеження, параметри методу відстеження можна систематично оптимізувати, щоб мінімізувати помилку відстеження для великої кількості відео. Нарешті, як тільки параметри були оптимізовані на змодельованих даних, ті ж параметри можна використовувати (після точної настройки параметрів і додавання або видалення слідів для забезпечення точності) для аналізу експериментальних відео.

Щоб навчити мережу в широкому діапазоні умов відео, ми розробили програмне забезпечення для моделювання відео, яке враховує широкий діапазон умов, виявлених в відео спостереження за частками (рис. 2.11). Основна перевага полягає в тому, щоб включити моделювання того, як частинки, що рухаються в трьох вимірах, з'являються в зрізі 2D зображення, знятого камерою.

Стандартна камера створює зображення, які зазвичай є одноканальними (шкала сірого), а дані зображення збираються в 4D (три просторових і один часовий вимір) масиви з 16-бітових цілих чисел. Роздільна здатність в площині (x, y) визначається камерою і може перебувати в діапазоні мегапікселів. Роздільна

здатність по координаті z набагато менше, оскільки для кожного зрізу осі z , що відображається камерою, потрібно п'єзоелектричний двигун для переміщення лінзи щодо зразка. Хороший п'єзоелектричний двигун здатний переміщатися між зрізами по осі Z за кілька мілісекунд, а це означає, що існує компроміс між великою кількістю зрізів по осі Z і загальною частотою кадрів. Для відстеження частинок типове відео включає від 10 до 50 z -зрізів на обсяг. Довжина відео відноситься до кількості тимчасових точок, тобто кількістю зібраних томів. Довжина відео часто обмежується фотознебарвленням, яке повільно знижує SNR в міру просування відео.

Щоб змоделювати відео відстеження частинок спочатку вказується, як частинки з'являються на зображенні. Ми називаємо інтенсивності пікселів, захоплених мікроскопом і камерою в результаті центрування частки в заданому положенні (x, y, z) , як спостережувану функцію розсіювання точки (PSF), що позначається $\psi_{ijk}(x, y, z)$, де i, j і k - індекси пікселів. PSF стає більш тьмяним і менш сфокусованим в міру віддалення частки від площини фокусу ($z = 0$). Далеко від площини фокусу PSF також формує дискові візерунки, викликані дифракцією, яка може погіршуватися через сферичну аберацію. Хоча деконволюція може пом'якшити дискові візерунки, що з'являються в PSF, точна форма PSF повинна бути відома, інакше в зображення можуть з'явитися непередбачені артефакти.

Форма PSF залежить від декількох параметрів, які варіюються в залежності від мікроскопа і камери, включаючи довжину хвилі випромінюваного світла, числову апертуру, розмір пікселя і відстань між зрізами по осі z . Хоча фізичні моделі, засновані на оптичній фізиці, які розкривають ці параметри, були розроблені для колоїдних сфер, вони непрактичні для цілей автоматичного відстеження частинок в складних біологічних середовищах. На практиці є багато додаткових чинників, які впливають на PSF, такі як показник заломлення предметного скла, масла для лінз і середовища, що містить частинки, зображення яких можна отримати. Остання являє найбільшу складність, оскільки біологічні зразки часто бувають неоднорідними, а їх оптичні властивості важко

передбачити. На PSF також може впливати швидкість частинок, в залежності від тривалості інтервалу експозиції, що використовується камерою. Це робить машинне навчання особливо привабливим, тому що ми можемо просто рандомізувати форму PSF для охоплення широкого діапазону умов, а отримана CNN здатна автоматично «деконволюціонувати» PSF без необхідності знань будь-яких з вищезазначених параметрів.

Низьке відношення сигналу до шуму - додаткова проблема для відстеження частинок субмікронного розміру. Високопродуктивні цифрові камери використовуються для запису зображень з досить високою частотою кадрів для визначення статистичних характеристик руху частинок. Частинки з гідродинамічним радіусом в діапазоні від 10 нм до 100 нм переміщуються швидко, що вимагає невеликого часу експозиції для мінімізації помилки динамічної локалізації (розмитості під час руху) (28). Більш дрібні частинки також випромінюють менше світла, який збирає камера. Щоб навчити нейронну мережу працювати в цих умовах, ми додаємо в навчальні відео дробовий шум Пуассона випадкової інтенсивності. Ми також додаємо повільно мінливі випадкові шпалери (рис. 2.10).

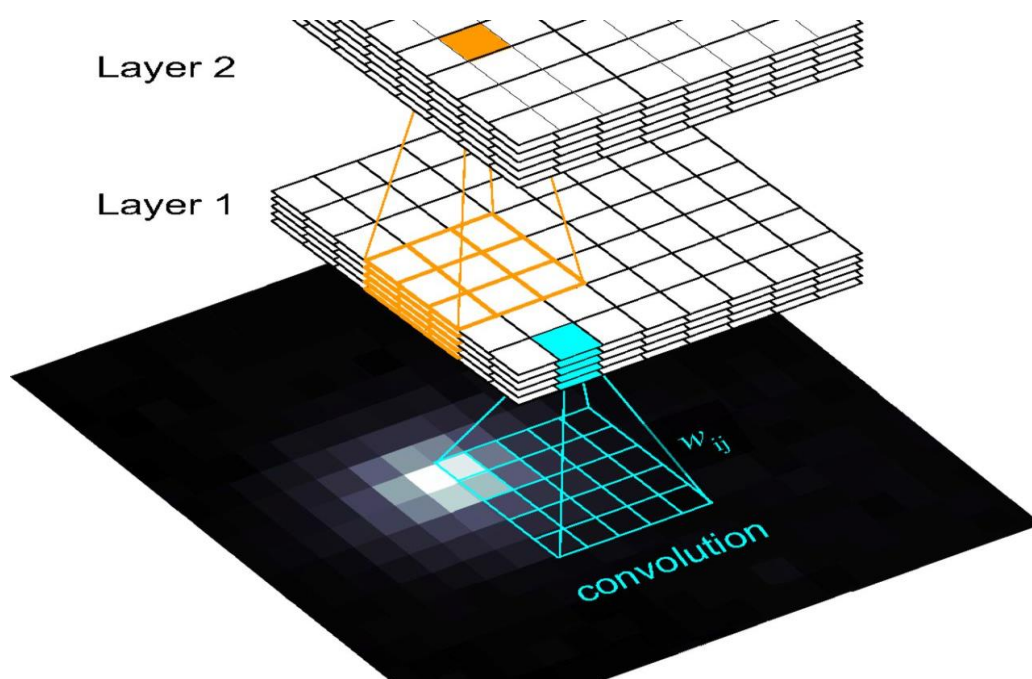


Рис 2.13 Схема багаторівневого зв'язку штучної нейронної мережі

2.2 Трекінг відеооб'єктів

Між відстеженням і виявленням існує тісний взаємозв'язок. Виявлення полягає у ідентифікації одного або декількох об'єктів на заданому зображенні, тоді як мета відстеження полягає в тому, щоб визначити місцезнаходження цих об'єктів по всьому відео, відстежуючи, який об'єкт знаходиться в кадрах відео. Щоб відстежувати об'єкт, спочатку необхідно надати оригінал зазначеного об'єкта в алгоритм відстеження, і це виконується або алгоритмом виявлення, або вручну.

Наївний спосіб відстежити - застосувати алгоритм виявлення до кожного кадру відео, але є кілька причин, за якими відстеження необхідно або корисно:

- відстеження дозволяє зберігати ідентичність об'єкта
- виявлення вимагає великих обчислювальних ресурсів
- трекери без виявлення дозволяють відстежувати об'єкти, для яких не навчений жоден детектор.
- відстеження може допомогти вирішити складні загальні проблеми, такі як зміна освітлення, розмиття руху, зміна масштабу, оклюзії (коли мета частково або повністю прихована іншим об'єктом протягом певного періоду часу на відео), низька якість зображення.

Два основних підходи до відстеження: *«відстеження одного об'єкта»* (SOT) і *«відстеження декількох об'єктів»* (MOT).

У SOT обмежувальна рамка цілі в першому кадрі передається трекеру. Тоді мета трекера - знайти ту саму мету у всіх інших кадрах. Трекери одного об'єкта відносяться до категорії трекерів без виявлення, тому що даний перший обмежувальний прямокутник. Вони повинні мати можливість відстежувати будь-який об'єкт без будь-якого тренування на об'єкті.

Сіамські мережеві трекери і трекери на основі *фільтра кореляції* є кращими виконавцями для короткострокових завдань (тобто без повної оклюзії). У MOT є кілька об'єктів для відстеження. Очікується, що алгоритм відстеження,

по-перше, визначить кількість об'єктів в кожному кадрі, а по-друге, буде відслідковувати їх особистості.

МОТ - більш складніша проблема, і важче продемонструвати явний клас алгоритмів, який перевершив би інші.

У задачах класифікації та виявлення об'єктів був досягнутий величезний прогрес, і алгоритми досягли продуктивності, порівнянної з людськими. Але сучасні трекери все ще далекі від цієї мети, яка вимагає фіксувати просторові і тимчасові відносини між цілями.

Щоб дати вам уявлення, у зведеній метриці МОТ, кращі алгоритми не виходять за межі 0,6 (найкраща можлива оцінка - 1,0). Щоб зрозуміти ці показники, потрібно вдаватися в подробиці, але це все одно означає, що є можливості для поліпшення в порівнянні з ідеальним випадком.

2.3 Методи після обробки

Перші методи, які з'явилися, були модифікаціями, застосованими до етапу постобробки конвеєра виявлення об'єктів. Це тому, що для цього потрібно менше інфраструктури і не потрібно змін в архітектурі моделі. Методи обробки поста і раніше будуть покадровим процесом виявлення та, отже, не матимуть підвищення продуктивності (обробка може зайняти трохи більше часу). Проте, він може значно підвищити точність.

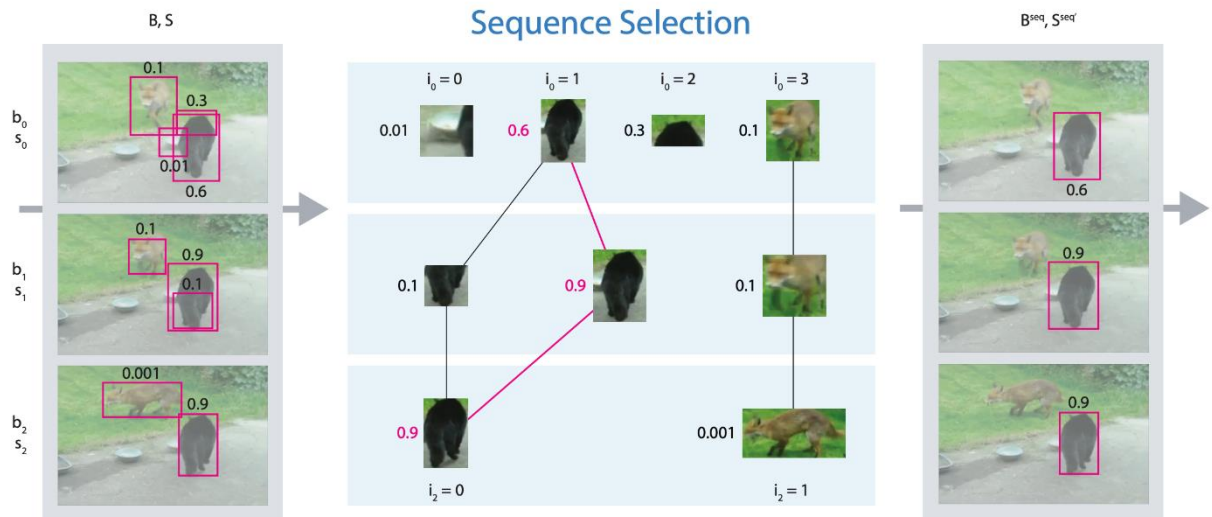


Рис 2.14 Ілюстрація вибору послідовності.

Відомим методом є Seq-NMS (Sequence Non-Maximal Suppression), який застосовує модифікацію до достовірності виявлення на основі інших виявлень на «треку» за допомогою динамічного програмування. Наприклад, більш слабкі прогнози позитивного об'єкта можуть бути викликані оклюзією, розмиванням руху або іншими дефектами, але оскільки він буде присутній в «доріжці» (критерій перекриття), витягнутої з попередніх кадрів, достовірність буде підвищена. Це ефективно мінімізує кількість помилкових виявлень між кадрами або виявлень випадкових переходів і стабілізує вихідний результат.

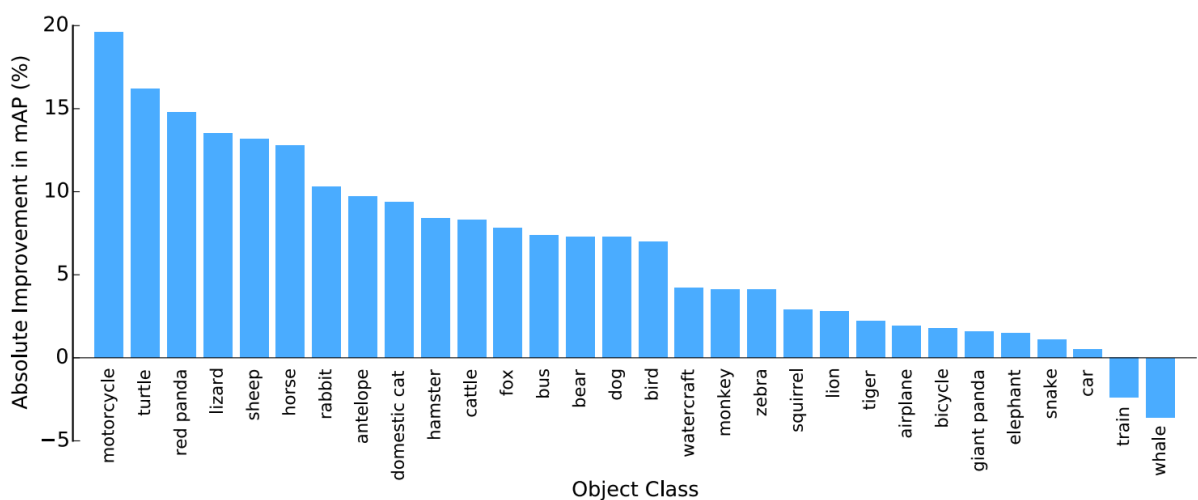


Рис 2.15 Абсолютні поліпшення mAP (%) з використанням Seq-NMS. Поліпшення відноситься до NMS з одним зображенням.

Як видно з наведеного вище графіка, точність була поліпшена на відповідну величину: абсолютні поліпшення в mAP (%) з використанням Seq-NMS в порівнянні з NMS з одним зображенням збільшилися більш ніж на 10%, для 7 класів є поліпшення більш ніж на 10% , в той час як тільки два класи показують знижену точність. Хоча ця робота була однією з перших робіт в напрямку кращого виявлення відео, вона не виявилася кращою як з точки зору точності, так і з точки зору продуктивності. Однак очевидною перевагою є те, що цей метод не вимагає навчання і діє скоріше як доповнення, яке можна підключити до будь-якого детектору об'єктів.

2.4 Багатокадрові методи

3D згортки.

Наприклад, перший природний інстинкт розробника, який має досвід класифікації зображень, - це подумати про якусь тривимірну згортку, засновану на двовимірній згортці, яка виконується для зображень. Імовірність такої архітектури цілком імовірна: повторення n кадрів в якості вхідних даних для моделі і висновок послідовних виявлень на послідовних кадрах. Це безперечно потенційний напрямок для виявлення, оскільки він може витягувати низькорівневі функції для просторово-часових даних, але згорткова нейронна мережа з тривимірними згортками в основному виявилася корисною і плідною, коли справа доходить до обробки тривимірних зображень, таких як 3D MNIST або MPT. Ось чому ці моделі є великим проривом в галузі медичної візуалізації та менш актуальні для відеовідслідковування.

Проте, одним із прикладів дослідницької роботи, присвяченій використанню тривимірної згортки при обробці відео, є наскрізна тривимірна згорткова нейронна мережа для виявлення дій і сегментації в відео. У дослідницькій роботі відео спочатку ділиться на кліпи рівної довжини, а потім

для кожного кліпу генерується набір пропозицій трубок на основі функцій 3D CNN. Потім пропозиції трубок різних кліпів зв'язуються разом, і за допомогою цих пов'язаних відео пропозицій визначатиме просторово-часового дії. Хоча в статті в основному йдеться про сегментацію і виявленні дій, похідні від архітектури можна навчити виявленню об'єктів.

Що стосується продуктивності, через великий обсяг обчислень з багатовимірними матрицями час обробки не може бути таким же швидким, як в реальному часі (30 кадрів в секунду або вище) в поточному стані. Подальші поліпшення та дослідження в цій області можуть змінити напрямок, але складність збільшення продуктивності тривимірної згортки - непросте завдання. Що стосується точності, я вважаю, що на неї точно можна позитивно вплинути. Стабільність, а також точність виявлення можуть бути поліпшені за допомогою тривимірної згортки, оскільки архітектура може ефективно використовувати часовий вимір в цілому (агрегування функцій між кадрами). Однак в даний час це всього лише припущення, засноване на інших сучасних згортальних тривимірних моделях. До сих пір немає дослідницької роботи, в якій детально розповідається про виявлення відео.

Рекурентні нейронні мережі.

Тим, у кого більше досвіду роботи з послідовними даними, можна подумати про використання повторювальної нейронної мережі, такої як LSTM. RNN - це особливі типи мереж, які були створені для обробки послідовних, в тому числі тимчасових, даних. Область, в якій ця архітектура отримала велику користь, - це обробка природної мови. Наприклад, показано, що AWD-LSTM працює на одному рівні з сучасною моделлю трансформатора BERT, але має набагато менше параметрів.

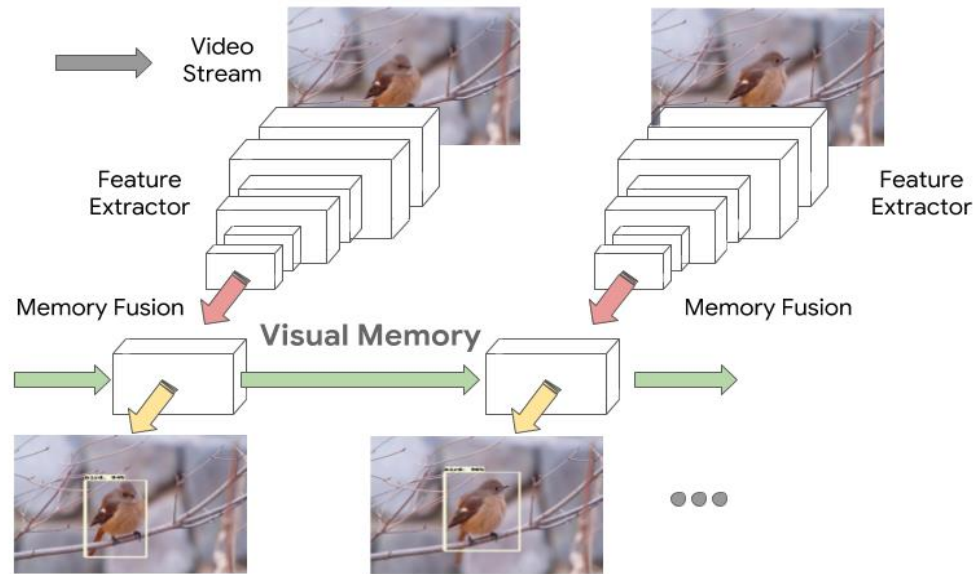


Рис 2.16 Шари LSTM для підвищення продуктивності.

У першому випадку в статті поєднуються швидке виявлення об'єкта з одним зображенням з шарами згорткової «довготривалої короткострокової пам'яті» (LSTM), званими вузьким місцем - LSTM, для створення переплетеної рекуррентно-згорткової архітектури. Шар LSTM знижує обчислювальні витрати, але при цьому уточнює і поширює карти функцій по кадрам. Цей документ призначений для роботи в режимі реального часу на мобільних і вбудованих пристроях з низьким енергоспоживанням з частотою 15 кадрів в секунду на мобільному пристрої.

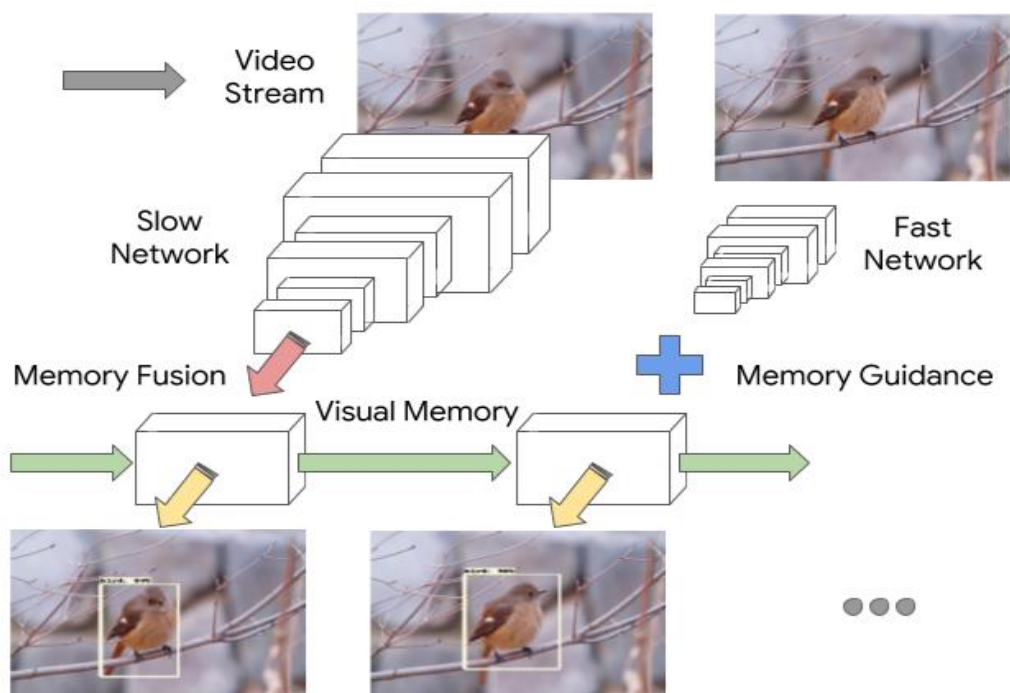


Рис 2.17 Великі і малі нейронні мережі, що використовують шари LSTM.

В останньому випадку дослідники пропонують використовувати «суть» (багате уявлення складного навколишнього середовища за короткий період часу) сцени, покладаючись на відповідні попередні знання, які засновані на тому, як люди можуть розпізнавати і виявляти об'єкти. Архітектура моделі заснована на чергуванні традиційних екстракторів ознак з легкими, яким необхідно тільки розпізнавати суть сцени (мінімальні обчислення). Це ефективно створює довгострокову пам'ять для архітектури з ключового кадру, який фіксує «суть», яка направляє невелику мережу туди, що слід виявляти. Документ також містить алгоритми навчання з підкріпленням для досягнення адаптивної політики виведення. У документі пропонуються багатообіцяючі результати, такі як 70 кадрів в секунду на мобільному пристрої, але при цьому досягаються найсучасніші результати для невеликих нейронних мереж на ImageNet VID.

Одним з ключових висновків є те, що архітектура є наскрізною, що означає, що вона бере зображення і виводить замасковані дані, і необхідно провести навчання для всієї архітектури. Оскільки ми маємо справу з відеоданими, модель необхідно навчити на великій кількості даних.

Розріджене поширення функцій для продуктивності

Архітектура функціонує з концепцією розрідженого ключового кадру. Оскільки мережа оптичного потоку може бути порівняно малою, час обробки та обчислювальна потужність, необхідні для таких мереж, менші, ніж об'єктні детектори. Отже, конвеєр функціонує як цикл з n кадрів. Перший кадр називається ключовим. Це кадр, який виявляє детектор об'єктів. Оскільки зараз детектори дають точне виявлення всіх предметів, виявлення будуть підпорядковуватися алгоритмам оптичного потоку. Після отримання векторів переміщення відомі виявлення наступних $n-1$ кадрів, і цикл повторюється.

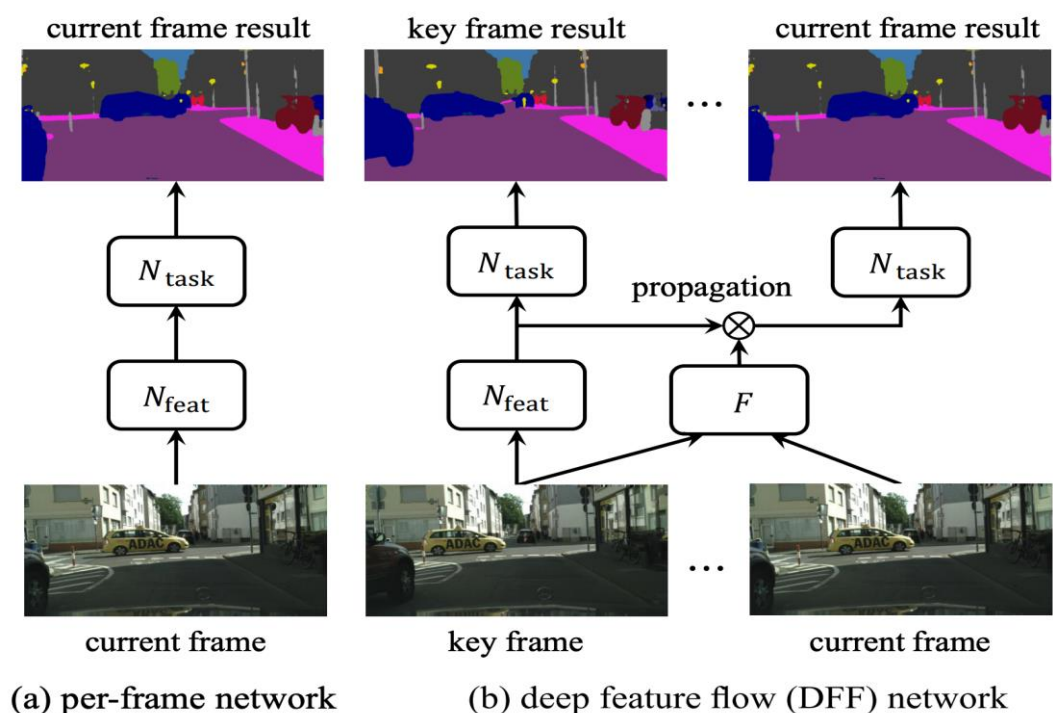


Рис 2.18 Поширення рідкісних ознак.

Багатокадрове об'єднання функцій для точності

Метод підвищення точності виявлення відео - це багатокадрова агрегація ознак. Існують різні способи його реалізації, але всі вони засновані на одній ідеї: детектування з щільним обчисленням по кадрам при деформації функцій від сусідніх кадрів до поточного кадру та його узагальнення. Таким чином, поточний кадр отримає користь від безпосередніх кадрів, а також деяких подальших кадрів

для кращого виявлення. Потім це може вирішити проблеми з рухом та обрізаними предметами з відеокадру.

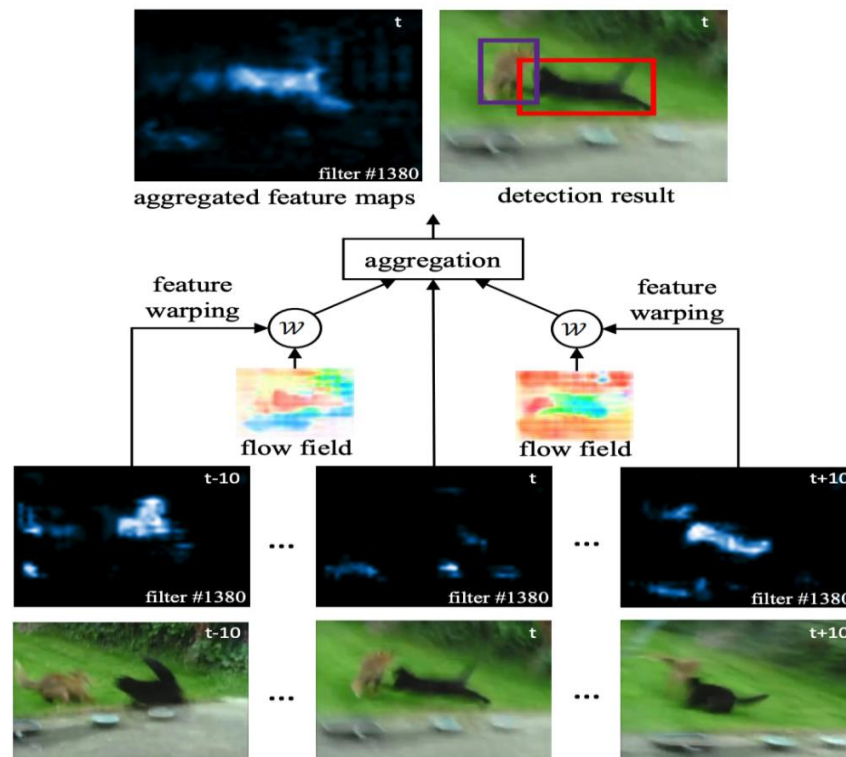


Рис 2.19 Багатокадрове об'єднання функцій

Одним з таких прикладів є агрегування функцій на основі потоку дослідних робіт (FGDA). Агрегація ознак на основі потоків об'єднує карти ознак з найближчих кадрів, які добре вирівняні по передбачуваному потоку. Архітектура - це наскрізна структура, яка використовує тимчасову узгодженість на рівні функцій.

Висновки за розділом

В даному розділі було досліджено та проаналізовано конкретну задачу пошуку об'єктів та їх ідентифікації в відеопотоці. Під даним поняттям знаходження або відслідковування мається на увазі ідентифікація одного конкретного статичного чи динамічного об'єкту в різних умовах. Припускається що вони можуть мати деяку схожість ознак або рис між собою. Саме ці ознаки використовуються згідно з алгоритмом. Під поняттям трекінгу розуміється

знаходження місця дислокації об'єкта який має певний напрямок руху, прямолінійний чи хаотичний за допомогою відеообладнання. Також було розглянуто відомі алгоритми які опрацьовують рух об'єкта в оптичному потоці та відслідковують його в полі зору відеообладнання. Було проаналізовано використання згорткових нейронних мереж, їх примінення в структурі ідентифікації. Також в даному розліді розглядалося поняття начання каскаду Хаара, який сприяє підвищенню точності розпізнавання при частковому перекритті об'єкта, та багатокамерний трекінг. Для цього потрібно точно відкалібрувати камеру між сценами охоплюваних зон. Даний метод також позбавляє такого недоліку як оклюзії.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ

2.5 Інструментарій розробки

Метою роботи є підвищення якості процесу розпізнавання об'єктів в відеопотоці шляхом який передбачає використання операції предобробки кадрів та збільшення значення fps для інтелектуального робота в реальному часі. Код з яким будемо працювати створений в середовищі розробки PyCharm, на мові програмування Python, с використанням бібліотеки OpenCV рис. 3.1.

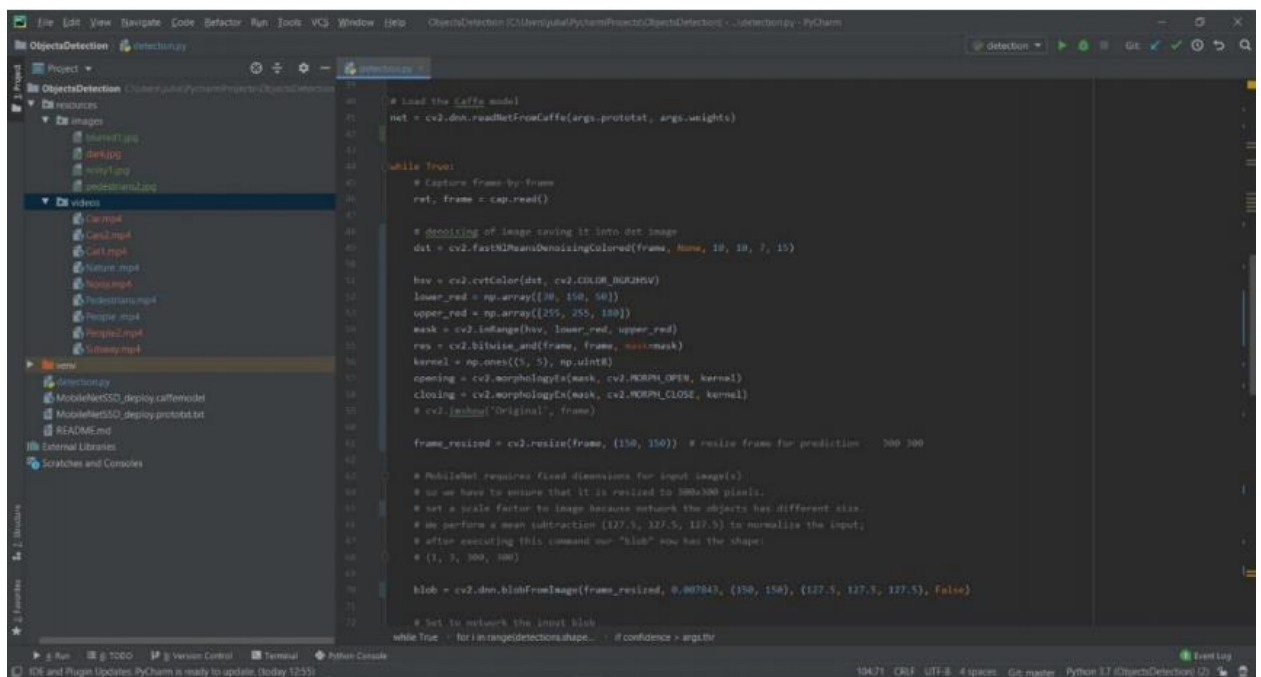


Рисунок 3.1 – Середовище розробки

Розглянемо програмні та операційні засоби, які будуть використовуватися при розробці програмного забезпечення:

Microsoft Windows, також звана Windows і ОС Windows, комп'ютерна операційна система (ОС), розроблена корпорацією Microsoft для роботи на персональних комп'ютерах (ПК). Оснащена першим графічним призначенням для користувача інтерфейсом (GUI) для IBM-сумісних ПК, ОС Windows незабаром стала домінувати на ринку ПК. Приблизно 90 відсотків комп'ютерів працюють під управлінням тієї чи іншої версії Windows.

Python - це об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Його високорівневі вбудовані структури даних в поєднанні з динамічною типізацією і динамічною прив'язкою роблять його дуже привабливим для швидкої розробки додатків, а також для використання в якості мови сценаріїв або сполучного мови для з'єднання існуючих компонентів. Простий і легкий в освоєнні синтаксис Python підкреслює легкість для читання і, отже, знижує вартість обслуговування програми. Python підтримує модулі та пакети, що сприяє модульності програми і повторного використання коду. Інтерпретатор Python і велика стандартна бібліотека доступні в вихідній або двійковій формі безкоштовно для всіх основних платформ і можуть вільно поширюватися.

Часто програмісти закохуються в Python через підвищену продуктивність, яку він забезпечує. Оскільки етап компіляції відсутній, цикл редагування-тестування-налагодження неймовірно швидкий. Налагодження програм Python проста: помилка або невірне введення ніколи не викличуть помилку сегментації. Замість цього, коли інтерпретатор виявляє помилку, він викликає виключення. Якщо виключення програми перехоплює виняток, інтерпретатор виводить трасування стека. Отладчик на рівні вихідного коду дозволяє перевіряти локальні і глобальні змінні, оцінювати довільні вирази, встановлювати точки зупинки, виконувати послідовне виконання коду по рядку за раз і т.д.. Отладчик написаний на самому Python, що свідчить про інтроспективну силу Python. З іншого боку, часто найшвидший спосіб налагодження програми - це додати кілька операторів друку до джерела: швидкий цикл редагування-тестування-налагодження робить цей простий підхід дуже ефективним.

NumPy – бібліотека обчислень для Python з відкритим вихідним кодом. Містить такі структури даних, як багатовимірні масиви та матриці. Може слугувати для виконання певних математичних операцій над масивами, на зразок тригонометричних, статистичних, алгебраїчних обчислень. Бібліотека містить багато математичних, алгебраїчних функцій та функцій перетворення.

3.2 Аналіз бібліотеки OpenCV

OpenCV - це кросплатформенна бібліотека, за допомогою якої ми можемо розробляти програми комп'ютерного зору в реальному часі. В основному він фокусується на обробці зображень, захоплення і аналізі відео, включаючи такі функції, як виявлення осіб і виявлення об'єктів.

Комп'ютерний зір можна визначити як дисципліну, яка пояснює, як реконструювати, перервати і зрозуміти тривимірну сцену на основі її двомірних зображень з точки зору властивостей структури, яка присутня в сцені. Він займається моделюванням і відтворенням людського зору за допомогою комп'ютерного програмного забезпечення і устаткування.

- Комп'ютерний зір перетинається з наступними полями:
- Обробка зображень - фокусується на маніпулюванні зображеннями.
- Розпізнавання образів - пояснює різні методи класифікації патернів.
- Фотограмметрія - пов'язана з отриманням точних вимірювань з зображень.

Комп'ютерний зір - це побудова явних, змістовних описів фізичних об'єктів на основі їх зображень. Результатом комп'ютерного зору є опис або інтерпретація структур в тривимірній сцені.

Модулі бібліотеки OpenCV

1) Основна функціональність

Цей модуль охоплює основні структури даних, такі як Scalar, Point, Range і т. Д., які використовуються для створення додатків OpenCV. На додаток до них, він також включає «багатовимірний масив» Mat, який використовується для зберігання зображень. У бібліотеку Java OpenCV цей модуль включений як пакет з ім'ям org.opencv.core.

2) Обробка зображення

Цей модуль охоплює різні операції обробки зображень, такі як фільтрація зображень, перетворення геометричних зображень, перетворення колірному простору, гістограми і т. Д. У бібліотеці Java OpenCV цей модуль включений у вигляді пакету з ім'ям `org.opencv.imgproc`.

3) відео

Цей модуль охоплює такі концепції відеоаналізу, як оцінка руху, віднімання фону і відстеження об'єктів. У бібліотеку Java OpenCV цей модуль включений як пакет з ім'ям `org.opencv.video`.

4) Відео введення або виведення

Цей модуль пояснює захоплення відео і відеокодеки з використанням бібліотеки OpenCV. У бібліотеку Java OpenCV цей модуль включений як пакет з ім'ям «`org.opencv.videoio`».

5) `calib3d`

Цей модуль включає в себе алгоритми, що стосуються основних алгоритмів геометрії із багатьма кутами, калібрування одиночної і стереокамери, оцінки стану об'єкта і елементів тривимірної реконструкції. У бібліотеку Java OpenCV цей модуль включений у вигляді пакету з ім'ям «`org.opencv.calib3d`».

6) `features2d`

Цей модуль включає в себе концепції виявлення і опису функцій. У бібліотеку Java OpenCV цей модуль включений у вигляді пакету з ім'ям «`org.opencv.features2d`».

7) `Objdetect`

Цей модуль включає виявлення об'єктів і примірників визначених класів, таких як особи, очі, гуртки, люди, автомобілі і т. Д В бібліотеці Java OpenCV цей модуль включений у вигляді пакету з ім'ям «`org.opencv.objdetect`».

8) Highgui

Це простий у використанні інтерфейс з простими можливостями для користувача інтерфейсу. У бібліотеці Java OpenCV функції цього модуля включені в два різних пакети, а саме «org.opencv.imgcodecs» і «org.opencv.videoio».

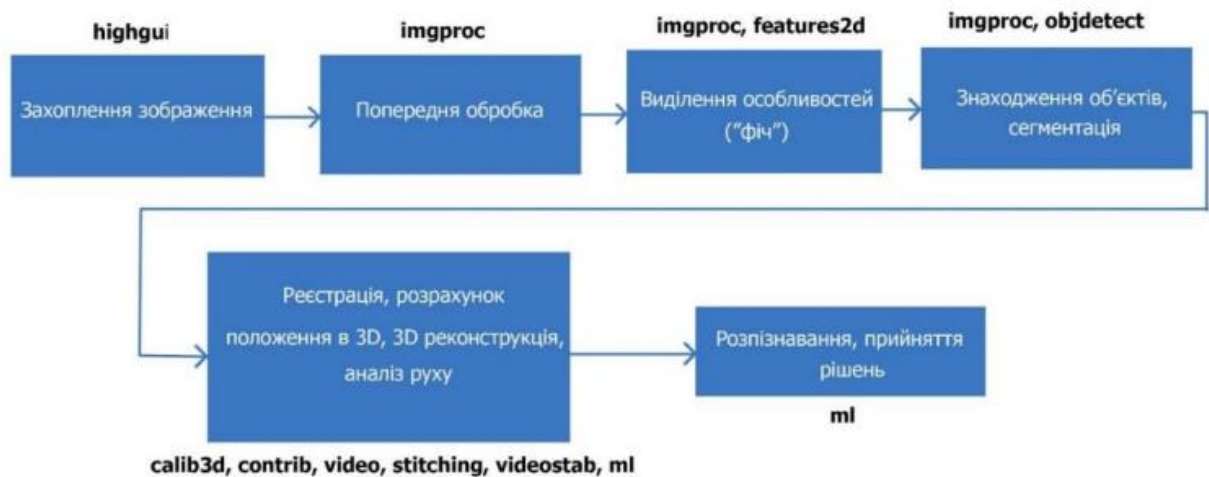


Рисунок 3.2 –Схема модулів в бібліотеці openCV

Дія розпочинається з захвату зображення (модуль highgui). Потім дане зображення зчитується з файлу чи з відео-поточної камери через мережевий протокол. На другому кроці має здійснюватися попередня обробка (модуль imgproc). При задіянні даного модулю усувається шум, вирівнюється яскравість, контраст, виділеність і видаленість відблисків, тіней. Представимо один і той же об'єкт при різних сценах освітлення. Він виглядатиме по-різному. При яскравому світлі червоний автомобіль, рух якого, необхідно відслідковувати, буде яскраво-оранжевою. У досить похмуру погоду той же автомобіль буде виглядати червоно-рожевим. У даному випадку на зображенні потрібно виконати вирівнювання кольору. Попередня обробка може бути простою, але може містити в собі складну технологію.

На другому етапі відбувається виділення особливостей (модулі imgproc, features2d). Припустимо в завданні супроводу об'єкту це може бути пошук спеціальних точок на об'єкті, за якими легко спостерігати; для завдання

детектування (тобто виявлення на зображенні) обличчя - обчислення опису кожного пікселя.

На третьому етапі відбувається розпізнавання цікавих об'єктів для нас, виділяються значущі частини, сегментації зображення (модулі `imgproc`, `objdetect`). Для прикладу візьмемо нерухому камеру, а саме зображення буде в нас рухомим. В даному випадку використовується алгоритм віднімання фону.

Після цього основне завдання - обчислити розташування 3D-об'єкта, реконструювати 3D-структуру, проаналізувати структуру, зберегти тощо. (`Calib3d`, `contrib`, `video`, `assembly`, `videostab`, `ml` модулі).

Наприклад, у проблемі вставки панорамних зображень - це порівняння частин різних кадрів, що визначає бажане перетворення. У задачі відеоспостереження входить відновлення траєкторій руху об'єктів тощо. На завершальному етапі приймаються конкретні рішення та їх визнання (модуль `ml`). Наприклад, у системі відеоспостереження: небажаний об'єкт є або не з'явився у кадрі. У завданні на виявлення тексту - визначити текст, що він з себе представляє.

3.3 Морфологічні зміни зображення

Для усунення недоліків шуму в зображенні відеопотоку використовуються морфологічні зміни в зображенні. Вони змінюють форми об'єктів на кінцевому зображенні.

Алгоритм реалізації морфологічних операцій полягає в наступному. Створюється масив такого самого типу як зображення, який має такі самі розміри як вихідне зображення. У нього записується результат морфологічної операції. До вибору «притягується» вікно, розміром з структурний елемент. Вікно переміщується попиксельно по всьому зображенні і для точки, яка

опиняється в центрі вікна обирається мінімальне чи максимальне (в залежності від виконаної операції, ерозії чи нарощування) значення із значень пікселей, які знаходяться в межах вікна і це значення записується в масив.

Візьмемо до розгляду такий спосіб як ерозія. Ерозія(звуження) зображення зазвичай використовується щоб позбутися випадкових вкраплень на зображенні. Ідея полягає в тому, що вкраплення при розмиванні усуваються, тоді як крупні і відповідно більш візуально-значущі регіони залишаються. Це операція зменшує границі форми. Операція для потовщення називається розширення. Ця операція збільшує границі для всіх форм на зображення, усуває шум на зображенні та сприяє об'єднанню областей, які були розділені шумом. Над наступним зашумленим зображенням виконаємо декілька морфологічних змін.



Рисунок 3.3 – Оригінальне зображення.

Для подавлення шумів на зображенні розробимо та впровадимо функцію `erosion()`, яка зображена на рис. 3.4.

```

erosion_size = 0
max_elem = 2
max_kernel_size = 21
title_trackbar_element_type = 'Element:\n 0: Rect \n 1: Cross \n 2: Ellipse'
title_trackbar_kernel_size = 'Kernel size:\n 2n +1'
title_erosion_window = 'Erosion Demo'
title_dilatation_window = 'Dilatation Demo'

def erosion(val):
    erosion_size = cv2.getTrackbarPos(title_trackbar_kernel_size, title_erosion_window)
    erosion_type = 0
    val_type = cv2.getTrackbarPos(title_trackbar_element_type, title_erosion_window)
    if val_type == 0:
        erosion_type = cv2.MORPH_RECT
    elif val_type == 1:
        erosion_type = cv2.MORPH_CROSS
    elif val_type == 2:
        erosion_type = cv2.MORPH_ELLIPSE
    element = cv2.getStructuringElement(erosion_type, (2 * erosion_size + 1, 2 * erosion_size + 1),
                                       (erosion_size, erosion_size))
    erosion_dst = cv2.erode(dst, element)
    cv2.imshow(title_erosion_window, erosion_dst)

```

Рисунок 3.4 – Функція для виконання ерозії.

Після впровадження функції erosion отримаємо зображення (рис. 3.5).



Рисунок 3.5 – Зображення після використання ерозії.

Операція розмикання допомагає видалити з зображення елементи менші за структурний елемент, але це зумовлює нарощення контуру. Щоб уникнути цього

застосовується операція замикання. Тому реалізуємо функцію `dilatation` одразу після виконання ерозії (рис. 3.6).

```
def dilatation(val):  
    dilatation_size = cv2.getTrackbarPos(title_trackbar_kernel_size, title_dilatation_window)  
    dilatation_type = 0  
    val_type = cv2.getTrackbarPos(title_trackbar_element_type, title_dilatation_window)  
    if val_type == 0:  
        dilatation_type = cv2.MORPH_RECT  
    elif val_type == 1:  
        dilatation_type = cv2.MORPH_CROSS  
    elif val_type == 2:  
        dilatation_type = cv2.MORPH_ELLIPSE  
    element = cv2.getStructuringElement(dilatation_type, (2 * dilatation_size + 1, 2 * dilatation_size + 1),  
                                       (dilatation_size, dilatation_size))  
    dilatation_dst = cv2.dilate(dst, element)  
    cv2.imshow(title_dilatation_window, dilatation_dst)
```

Рисунок 3.6 – Функція для виконання розширення

Після впровадження функції `dilatation`, отримаємо зображення (рис. 3.7).



Рисунок 3.7 – Зображення після використання розширення

3.4 Модифікація алгоритму розпізнавання об'єктів

Автоматичне виділення рухомих об'єктів сцени з вхідного відеопотоку – одна з найважливіших задач аналізу зображень. За останній час було запропоновано велику кількість методів для вирішення цього завдання за умови нерухомості відеокамери. У представленій роботі запропоновано метод націлений на покращення процесу розпізнавання, в його основі лежить виділення об'єктів на зображенні, ідентифікація кожного об'єкта відповідно до одного з 20 класів та визначення точності.

```
# Labels of network.
classNames = {0: 'background',
             1: 'aeroplane', 2: 'bicycle', 3: 'bird', 4: 'boat',
             5: 'bottle', 6: 'bus', 7: 'car', 8: 'cat', 9: 'chair',
             10: 'cow', 11: 'diningtable', 12: 'dog', 13: 'horse',
             14: 'motorbike', 15: 'person', 16: 'pottedplant',
             17: 'sheep', 18: 'sofa', 19: 'train', 20: 'tvmonitor'}
```

Рисунок 3.8 – Класи для ідентифікації об'єктів

Основні труднощі аналізу відеопотоку в реальному часі полягають в дуже великому обсязі даних при обмеженому часі для прийняття рішення. Щоб знайти в відеопотоці необхідний нам об'єкт, необхідно спочатку його виявити в режимі реального часу, а потім розпізнати. Ідентифікувати об'єкт будемо за допомогою алгоритма SSD (Single Shot MultiBox Detector).

Все починається з захвату зображення (модуль `highgui`). Зображення зчитується з файлу або відео з потокової камери через мережевий протокол. Далі здійснюється попередня обробка (модуль `imgproc`), така, як усунення шуму, вирівнювання яскравості, контрасту, виділення і видалення відблисків, тіней. Наприклад, один і той же об'єкт при різному освітленні виглядає по-різному.

Для фільтрації шуму будемо використовувати функцію `fastNLMMeansDenoisingColored()`.

```
# убираем шум и сохраняем изображение в dst
dst = cv2.fastNlMeansDenoisingColored(frame, None, 10, 10, 7, 21)
```

Рисунок 3.9 – Фрагмент коду для зменшення шуму зображення

Функція `fastNlMeansDenoisingColored()` працює для кольорових зображень.

Вона приймає такі аргументи:

- `h`: параметр, що визначає міцність фільтра. Більше значення `h` краще знімає шум, але також видаляє деталі зображення (рекомендовано 10);
- `hForColorComponents`: те саме, що `h`, але лише для кольорових зображень. (як правило, таке саме, як `h`);
- `templateWindowSize`: має бути непарним (рекомендується 7);
- `searchWindowSize`: має бути непарним (рекомендується 21).

В більшості програмних продуктах для обробки зображень використовують гама-корекцію. Якщо зображення необхідно освітлити чи затемнити, можна використовувати гама-корекцію. Наприклад, при освітленні можлива поява нових деталей в темних областях, які раніше не були помітні. Реалізуємо функцію для виконання гама-корекції для кожного кадру відеопослідовності:

```
def adjust_gamma(image, gamma=1.0):
    invGamma = 1.0 / gamma
    table = np.array([((i / 255.0) ** invGamma) * 255
                      for i in np.arange(0, 256)]).astype("uint8")
    return cv2.LUT(image, table)
```

Рисунок 3.10 – Фрагмент коду для гама-корекції кадру

Для зниження шуму зображення і зменшення неточних визначень об'єктів, наступним кроком виконуються морфологічні операції. Послідовне використання операцій розмикання та замикання. Операція розмикання допомагає видалити з зображення елементи менші за структурний елемент, але це зумовлює нарощення контуру. Щоб уникнути цього застосовується операція замикання. Вище зазначені

методи й операції дозволяють безпосередньо перейти до визначення об'єктів. Алгоритм розроблено за умови, що камера, яка передає відео послідовність нерухома. Алгоритм може оброблювати зображення з частотою кадрів камери. Розмір об'єктів, що визначаються не менше 10 пікселів висотою і шириною. Рух об'єкта поступальний по прямолінійній або криволінійній траєкторії, або періодичний. Швидкість руху не більше 100 пікселів на кадр.

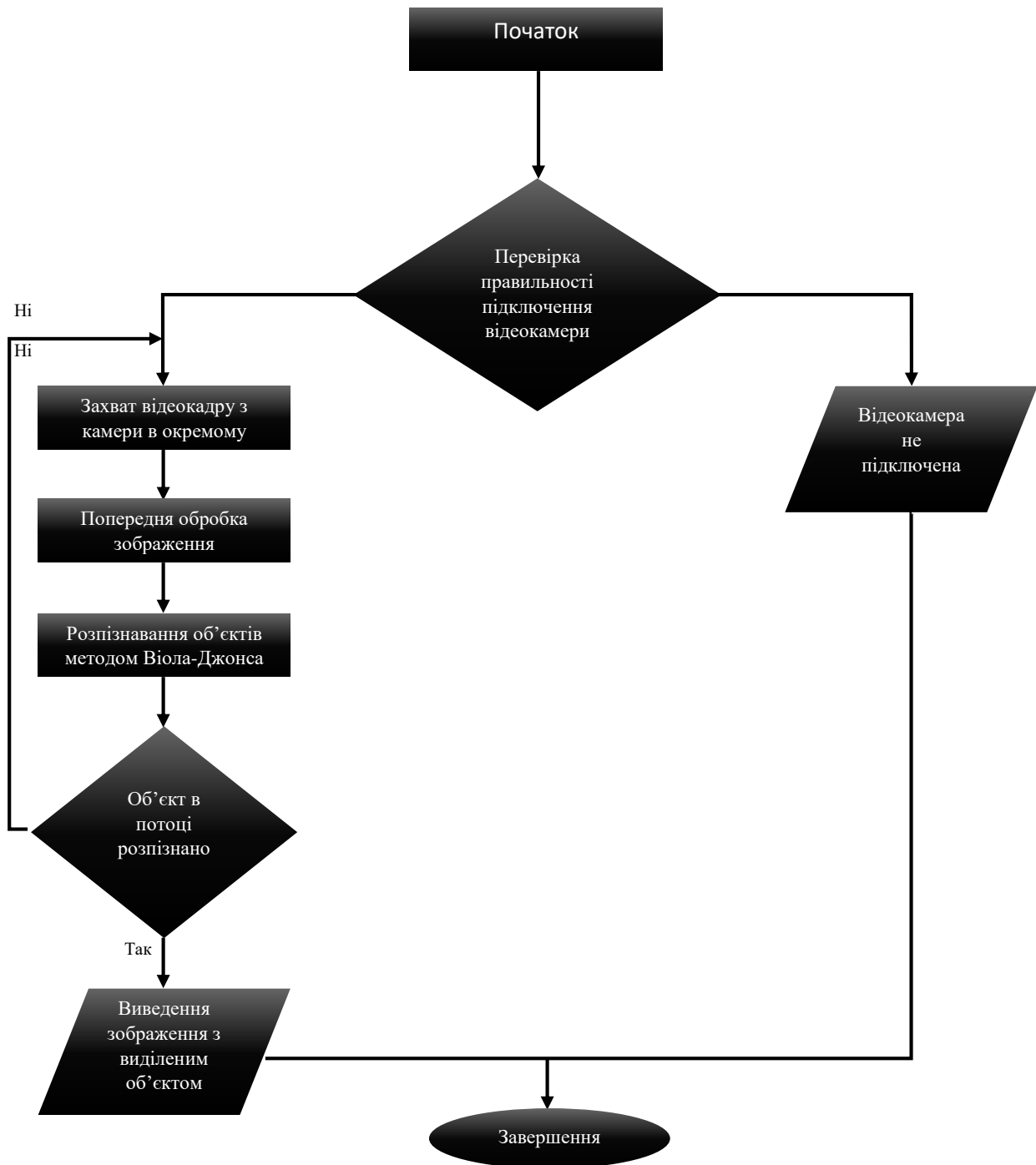


Рисунок 3.11 – Узагальнена схема роботи підсистеми.

3.5 Підвищення якості відеопотоку за рахунок використання багатопоточності

Обробка відео може являти собою обчислювально-ресурсномістке завдання, і, оскільки обчислювальна потужність часто невисока, чим більше ми зможемо прискорити конвеєр обробки відео, тим краще. Це особливо вірно для додатків, що вимагають обробки в реальному часі, наприклад обробки відеопотоку з веб-камери. Хоча важливо, щоб частина обробки зображень в конвеєрі обробки відео була оптимізована, операції введення-виведення (I / O) також мають тенденцію бути основним вузьким місцем. Один із способів пом'якшити це - розділити обчислювальну навантаження між декількома потоками. Важливо відзначити, що в Python існує так зване глобальне блокування інтерпретатора (GIL), яке забороняє доступ декількох потоків до одного і того ж об'єкту. Іншими словами, що блокують операції введення-виведення (які «блокують» виконання більш пізнього коду до тих пір, поки не буде виконано поточний код), такі як читання або відтворення відеокадрів.

Вимірювання змін до продуктивності

По-перше, ми повинні визначити «продуктивність» і те, як ми збираємося її оцінювати. У однопоточному додатку для обробки відео у нас може бути основний потік, що виконує наступні завдання в нескінченному циклі `while`: 1) отримання кадру з веб-камери або відеофайлу за допомогою `cv2.VideoCapture.read()`, 2) обробка `frame` як нам потрібно, і 3) вивести оброблений кадр на екран з викликом `cv2.imshow()`. Переносючи операції читання і відображення в інші потоки, кожна ітерація циклу `while` повинна займати менше часу для виконання. Отже, ми визначимо нашу метрику продуктивності як кількість ітерацій циклу `while` в основному потоці, виконуваних в секунду.

Щоб виміряти кількість ітерацій, які в основному виконуються циклі `while` в секунду, ми створимо клас і назвемо його `CountsPerSec`.

```

1  from datetime import datetime
2
3  class CountsPerSec:
4      """
5      Class that tracks the number of occurrences ("counts") of an
6      arbitrary event and returns the frequency in occurrences
7      (counts) per second. The caller must increment the count.
8      """
9
10     def __init__(self):
11         self._start_time = None
12         self._num_occurrences = 0
13
14     def start(self):
15         self._start_time = datetime.now()
16         return self
17
18     def increment(self):
19         self._num_occurrences += 1
20
21     def countsPerSec(self):
22         elapsed_time = (datetime.now() - self._start_time).total_seconds()
23         return self._num_occurrences / elapsed_time

```

Рис 3.12 Використання класу CountsPerSec.

Ми імпортуємо модуль `datetime` для відстеження минулого часу. В кінці кожної ітерації циклу `while` ми викликаємо `increment()`, щоб збільшити лічильник. Під час кожної ітерації ми будемо здобувати середню кількість ітерацій в секунду для відео за допомогою виклику методу `countsPerSec()`.

Продуктивність без багатопотоковості.

Перш ніж досліджувати вплив багатопотоковості, давайте подивимося на продуктивність без неї. Створимо файл з ім'ям `thread_demo.py`.

```

1  import argparse
2  import cv2
3  from CountsPerSec import CountsPerSec
4  from VideoGet import VideoGet
5  from VideoShow import VideoShow
6
7  def putIterationsPerSec(frame, iterations_per_sec):
8      """
9      Add iterations per second text to lower-left corner of a frame.
10     """
11
12     cv2.putText(frame, "{:.0f} iterations/sec".format(iterations_per_sec),
13                (10, 450), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (255, 255, 255))
14     return frame

```

Рис 3.13 Приклад коду

Почнемо з імпорту, включаючи клас `CountsPerSec`, який ми зробили вище. Ми ще не розглянули класи `VideoGet` і `VideoShow`, але вони будуть використовуватися для виконання завдань отримання відеокадрів і показу відеокадрів, відповідно, в їх власних потоках. Функція `putIterationsPerSec()` накладає текст, який вказує частоту основного циклу `while` в ітераціях в секунду, на кадр перед його відображенням. Він приймає в якості аргументів фрейм (масив `numpy`) і кількість ітерацій в секунду (`float`), накладає значення як текст через `cv2.putText()` і повертає змінений фрейм.

Потім ми визначимо функцію `noThreading()` для отримання кадрів, обчислення та накладення значення ітерацій в секунду в кожному кадрі і відображення кадру.

```

16 def noThreading(source=0):
17     """Grab and show video frames without multithreading."""
18
19     cap = cv2.VideoCapture(source)
20     cps = CountsPerSec().start()
21
22     while True:
23         (grabbed, frame) = cap.read()
24         if not grabbed or cv2.waitKey(1) == ord("q"):
25             break
26
27         frame = putIterationsPerSec(frame, cps.countsPerSec())
28         cv2.imshow("Video", frame)
29         cps.increment()

```

Рис 3.14 Приклад коду з приміненням команди noThreading ().

Функція приймає джерело відео в якості єдиного аргументу. Якщо задано ціле число, джерело вказує, що джерелом відео є веб-камера. 0 відноситься до першої веб-камери, 1 - до другої підключеної веб-камери і так далі. Якщо вказана рядок, вона інтерпретується як шлях до відеофайлу. У рядках 19-20 ми створюємо об'єкт OpenCV VideoCapture для захоплення і декодування кадрів з веб-камери або відеофайлу, а також об'єкт CountsPerSec для відстеження продуктивності основного циклу while. Рядок 22 починає основний цикл while. У рядку 23 ми використовуємо метод read () об'єкта VideoCapture для отримання і декодування наступного кадру відеопотоку; він повертає логічне значення grabbed, яке вказує, чи був кадр успішно захоплений і береться стверджувати, а також сам кадр у вигляді масиву numpy, frame. У рядку 24 ми перевіряємо, чи не вдалося захопити кадр або користувач натиснув клавішу «q», щоб вийти і вийти з програми. У будь-якому випадку ми зупиняємо виконання циклу while за допомогою break. Якщо виключити будь-яке з умов, ми продовжуємо, одночасно отримуючи і накладаючи поточну «швидкість» циклу (в ітераціях в секунду) в нижньому лівому кутку кадра в рядку 27. Нарешті, кадр відображається на екрані в рядку 28 з викликом cv2.imshow (), і лічильник ітерацій збільшується в рядку 29.

Як виглядають результати як для веб-камери, так і для відеофайлу? Ось значення, які я отримав на своєму обладнанні:

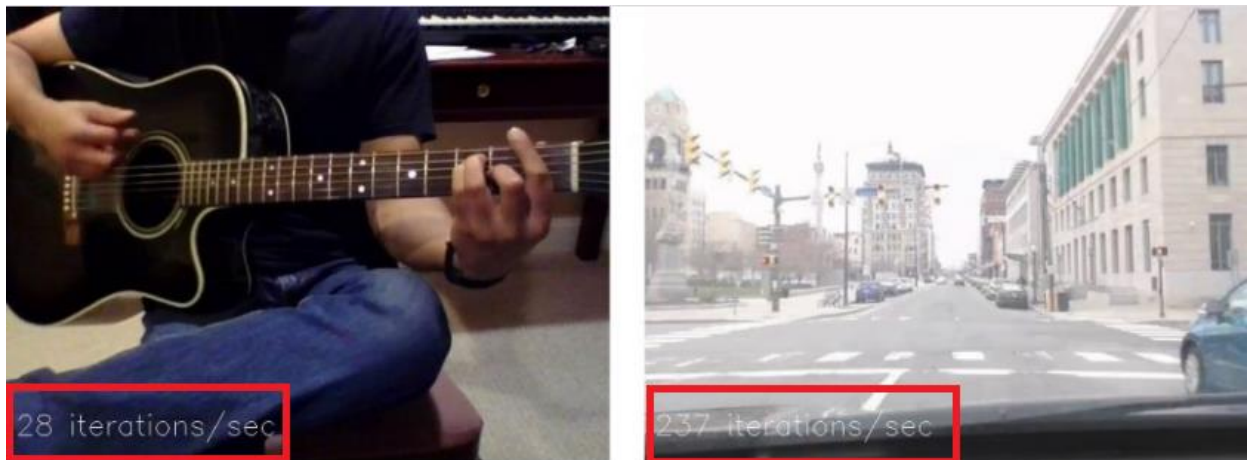


Рис 3.15 При читанні з веб-камери цикл while виконувався приблизно 28 ітерацій в секунду. Читання з файлу AVI, близько 240 ітерацій в секунду.

Це лише базові покращення.

Окрема гілка для показу відеокадрів.

Щоб перемістити завдання відображення відеокадрів в окремий потік, ми слідуємо процедурі, аналогічній попередньої теми, і визначаємо клас з ім'ям VideoShow в файлі з ім'ям VideoShow.py

```

13     def start(self):
14         Thread(target=self.show, args=()).start()
15         return self
16
17     def show(self):
18         while not self.stopped:
19             cv2.imshow("Video", self.frame)
20             if cv2.waitKey(1) == ord("q"):
21                 self.stopped = True
22
23     def stop(self):
24         self.stopped = True

```

Рис 3.16 Команда Show()

Зверніть увагу, що під час введення користувачем у рядку 20 досягається в окремому потоці, а не в основному потоці, оскільки функція OpenCV waitKey ()

не обов'язково добре працює в багатопотокових застосуваннях, і я виявив, що вона не працює належним чином при розміщенні в основному потоці. Ще раз, повертаючись до `thread_demo.py`, ми визначаємо функцію з ім'ям `threadVideoShow()`.

```

50 def threadVideoShow(source=0):
51     """
52     Dedicated thread for showing video frames with VideoShow object.
53     Main thread grabs video frames.
54     """
55
56     cap = cv2.VideoCapture(source)
57     (grabbed, frame) = cap.read()
58     video_shower = VideoShow(frame).start()
59     cps = CountsPerSec().start()
60
61     while True:
62         (grabbed, frame) = cap.read()
63         if not grabbed or video_shower.stopped:
64             video_shower.stop()
65             break
66
67         frame = putIterationsPerSec(frame, cps.countsPerSec())
68         video_shower.frame = frame
69         cps.increment()

```

Рис 3.17 Визначення функції з ім'ям `threadVideoShow()`

Як і раніше, ця функція схожа на функцію `noThreading()`, за винятком того, що ми ініціалізуємо об'єкт `VideoShow`, який я назвав `video_shower` і запустив новий потік в рядку 58. Рядок 63 побічно перевіряє, натиснув користувач «q» для виходу з програми, оскільки об'єкт `VideoShow` фактично перевіряє введення даних користувачем і встановлює для свого атрибута `Stop` значення `True` в тому випадку, якщо користувач натискає «q». Рядок 68 встановлює атрибут кадру об'єкта `VideoShow` для поточного кадру.

Тоді маємо результат:

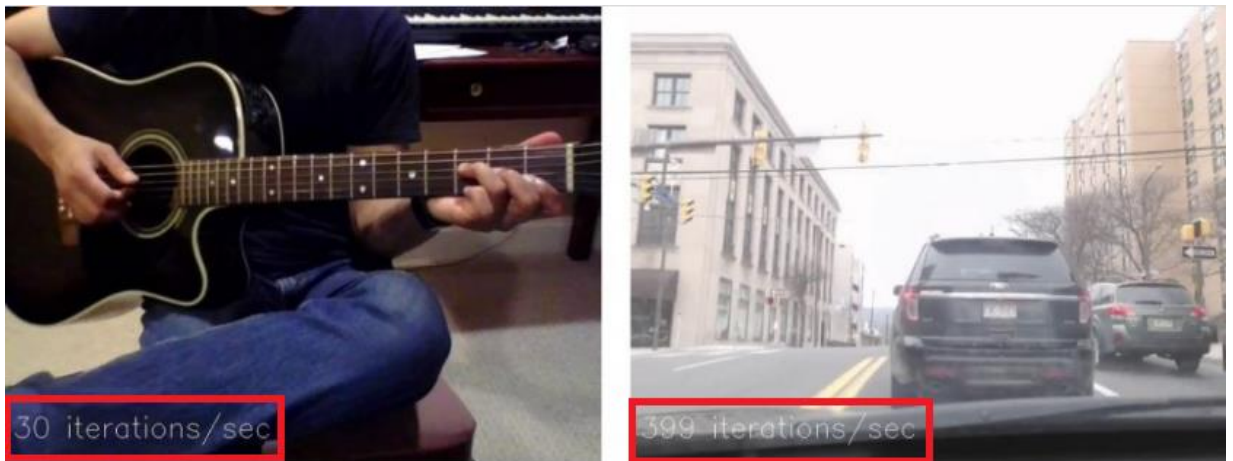


Рис 3.18 Результат пророблених маніпуляцій

Це цікаво. Веб-камера працює зі швидкістю 30 ітерацій в секунду, лише трохи швидше, ніж 28, отримані в разі одного потоку. Однак відеофайл працює зі швидкістю ~ 400 ітерацій в секунду - швидше, ніж його однопоточковий аналог (240 ітерацій в секунду), але повільніше, ніж відеофайл з читанням відео в окремому потоці (585 ітерацій в секунду). Це говорить про те, що існує фундаментальна різниця між читанням з потоку з камери і з файлу і що основним вузьким місцем для потоку з камери є читання і декодування відеокадрів.

Окремі потоки для отримання і відображення відеокадрів.

Нарешті, ми реалізуємо функцію з ім'ям `threadBoth ()` в `thread_demo.py`, яка створює потік для отримання відеокадрів через клас `VideoGet` і інший потік для відображення кадрів через клас `VideoShow`, при цьому основний потік існує для обробки і передачі кадрів між двома об'єктами.

```

71 def threadBoth(source=0):
72     """
73     Dedicated thread for grabbing video frames with VideoGet object.
74     Dedicated thread for showing video frames with VideoShow object.
75     Main thread serves only to pass frames between VideoGet and
76     VideoShow objects/threads.
77     """
78
79     video_getter = VideoGet(source).start()
80     video_shower = VideoShow(video_getter.frame).start()
81     cps = CountsPerSec().start()
82
83     while True:
84         if video_getter.stopped or video_shower.stopped:
85             video_shower.stop()
86             video_getter.stop()
87             break
88
89         frame = video_getter.frame
90         frame = putIterationsPerSec(frame, cps.countsPerSec())
91         video_shower.frame = frame
92         cps.increment()

```

Рис 3.19 Реалізація функції threadBoth ()

Ця функція являє собою суміш функцій threadVideoGet () і threadVideoShow (), яка дає дуже цікавий результат:

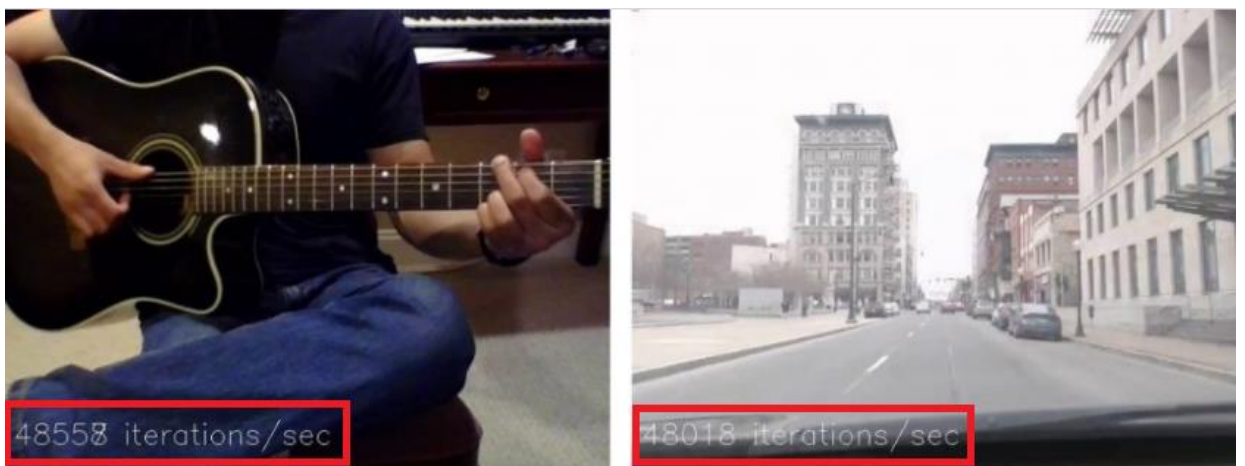


Рис 3.20 Результат застосування функції.

Це, схоже, розходиться з попереднім висновком, який припускав, що рамки читання були основним вузьким місцем для веб-камери. З якоїсь причини

комбінація приміщення як читання кадру, так і відображення кадру в виділені потоки збільшує продуктивність в обох випадках до колосальних ~ 48000 ітерацій в секунду. Я не так добре розбираюся в багатопотоковості, як хотілося б, тому не можу пояснити цей результат. Проте, здається досить очевидним, що використання багатопотоковості для введення-виведення відео може вивільнити значні ресурси для виконання інших завдань обробки зображень.

3.6 Тестування розробленої підсистеми

Розберемося наскільки точно дане програмне забезпечення відслідковує та розпізнає об'єкти. Дане ПЗ опрацьовує лише послідовні зображення і рухомі об'єкти які мають статичний фон. Єдиний недолік – об'єкти природи такі як дерева і т.д. розпізнаються як об'єкт переднього фону. Сама картинка повинна бути якісною, використовувалися відеокамери з хорошою роздільною здатністю та оптикою, так як неякісне зображення приводить до хибних вимірювань та ідентифікацій.

Емпіричним шляхом зробимо аналіз проробленої роботи.

Приклад № 1



Рис 3.21 – Приклад роботи ПЗ

ПЗ послідовно виводить зображення, з ідентифікованим об'єктом. ПЗ робить захват об'єкта в синю рамку і ідентифікує його під спеціальним кодовим іменем зі швидкістю близько десятки тисяч ітерацій в секунду.

Послідовність має 4 об'єкти які хаотично рухаються. При початковій ідентифікації ПЗ розпізнало 4 об'єкти з точностями 0,4012%, 0,6489%, 0,8771% та 0,8382%.

Таблиця 3.1 – Приклад точності розпізнавання об'єктів в певному індексі

Час, сек.	Точність розпізнавання, %			
	Об'єкт 1	Об'єкт 2	Об'єкт 3	Об'єкт 4
1	40,12	64,89	87,71	83,82
2	82,69	90,43	46,41	75,85
3	86,94	92,17	67,28	91,90
4	78,16	95,95	75,30	96,65

Результат досліджень співпадає з нормами. Об'єкти які відслідковувались правильно ідентифікувались. З приближенням об'єктів точність їх розпізнавання суттєво збільшується, що і очікувалось в результаті.

Приклад № 2

Розглянемо приклад ідентифікації з роздільною здатністю послідовних кадрів 640x360 пікселів. Беремо до уваги розглядання лінійного руху об'єктів. При даній послідовності спостерігається сильне зашумлення кадрів за рахунок більш інтенсивного руху об'єктів. Програмне забезпечення ідентифікувало на початку лише 2 об'єкти, при приміненні оклюзії більше половини об'єктів в кадрі. ПЗ на 12 секунді правильно ідентифікувала об'єкти, як «person» з особливим кодом ідентифікації для кожного навіть при сильному зашумленні.



Рисунок 3.22 – Результат роботи програми.

Таблиця 3.2 – Точність розпізнавання об'єктів через рівні проміжки

Час, сек.	Точність розпізнавання, %			
	Об'єкт 1	Об'єкт 2	Об'єкт 3	Об'єкт 4
1	74,34	97,82	Не розпізнано	Не розпізнано
4	90,02	86,09	30,92	Не розпізнано
8	58,96	94,01	Не розпізнано	54,24
12	58,96	94,01	Не розпізнано	54,24

Як і очікувалось, робота ПЗ повністю співпадає з вимогами, об'єкти ідентифіковані, але за рахунок зашумлення об'єкт 3 розпізнано лише в той момент коли він змінив напрямок руху. Сам об'єкт 3 знаходився на серйозній відстані в плотному потоці. Навіть з такими непростими умовами ПЗ на мить розпізнало його.

Ще одна причина того що об'єкт на першій, восьмій і дванадцятій секунді не розпізнаний через перекриття об'єктом №1.

Що стосується об'єкта 4, він не був ідентифікований на першій і четвертій секунді за рахунок сильного шуму.

Приклад №3

Розглянемо даний приклад з такою самою роздільною здатністю послідовності зображень як і на 2 прикладі, тільки в даному дослідженні візьмемо до уваги поступальних рух об'єктів. Дана послідовність зображень містить зміни освітлення в різних сценах, що надає певну складність в точному розпізнаванні.

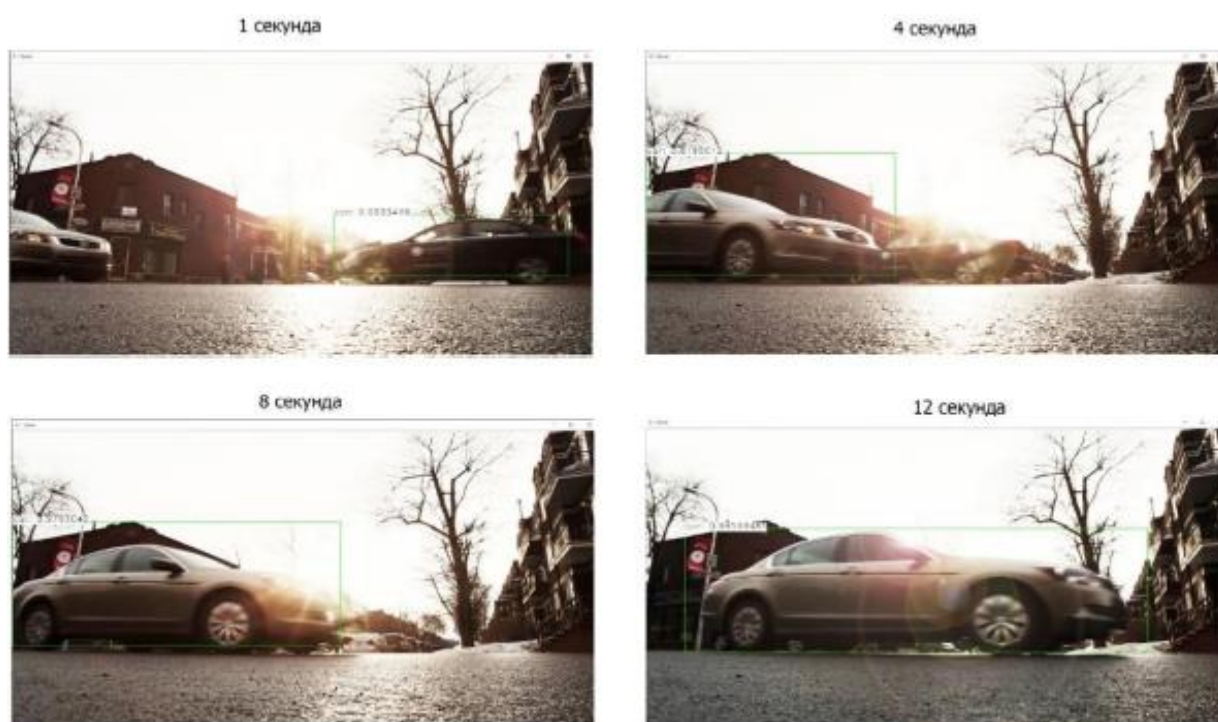


Рисунок 3.23 – Результат опрацювання ПЗ

Досліджувана послідовність має в собі 2 об'єкти, конкретно автомобілі які ідуть паралельно одне одному в різних напрямках, по прямій траєкторії. При початковому етапі відслідковування ПЗ слідкувало за обома об'єктами. Точність ідентифікації становила 0,6533%, при тому ідентифікований другий об'єкт знаходився в кадрі лише на половину.

Розробимо таблицю для порівняння точностей ідентифікацій об'єктів.

Таблиця 3.4 – Точність розпізнавання об'єктів через рівні проміжки часу

Час, сек.	Точність розпізнавання, %	
	Об'єкт 1	Об'єкт 2
1	65,33	Не розпізнаний
4	Не розпізнаний	81,95
8	Не розпізнаний	97,93
12	Не розпізнаний	99,51

В даному результаті бачимо що об'єкти були розпізнані в різні проміжки часу. Можемо спостерігати що об'єкт №1 не був розпізнаний в проміжках від четвертої до дванадцятої секунди за рахунок зміни інтенсивності освітлення.

Проведемо зрівняння кожного прикладу до існуючого методу ідентифікації та модифікованого методу, отримані результати будуть наведені в таблиці.

Проведемо розрахунок середнього відсотка значень відслідковуваних об'єктів методом який модифікований та методом без наших модифікацій. Візьмемо до уваги різні умови при яких проводилась ідентифікація об'єктів: часткове перекриття, шуми, зміна освітлення.

Отже зробимо висновок, що при використанні багатопотоковоті та комбінування різних методів фільтрацій, морфологічних модифікувань, застосування новітніх бібліотек із застосуванням сегментацій примінюючи метод Віоли-Джонса, похибка в відслідковуванні та сама точність ідентифікації зростає.

Таблиця 3.5 Дослідження різниці попереднього алгоритму розпізнавання з модифікованим

Умови тестування	Точність розпізнавання, %		
	Модифікований метод розпізнавання об'єктів	Вихідний метод розпізнавання об'єктів	Відсоток покращення
Ідеальні умови розпізнавання	92,14	92,14	Не змінився
Сильний шум	82,63	90,00	7,37
Зміна інтенсивності освітлення	55,12	63,45	8,33

Висновки за розділом

В даному розділі було запропоновано модифікування можливості розпізнавання та відслідковування об'єктів, проведений аналіз бібліотеки OpenCV, було примінено функції erosion (), та розроблено дослідження впливу багатопотоковості на швидкість ідентифікації об'єкта в ефірі та в послідовностях зображень. Єдиним вагомим мінусом алгоритму є брання до уваги зашумлення як об'єкта переднього плану. Для уникнення таких помлок будемо використовувати методи співставлення за шаблонами. Шаблонами можуть бути будь-які силуети. Застосування багатоканального трекінгу буде як раз доречно в нашій ситуації. Застосовуючи його ми співставляємо об'єкти на різних кадрах в одному часовому проміжку. Рекомендацією буде збільшення навантаження на локальну машину яка обробляє алгоритми виведення-введення зображення та їх ідентифікації. Може знизитись ефективність роботи трекінгу при цьому, але за те підвищиться якість відстеження.

РОЗДІЛ 4. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Розробка алгоритмів для систем відеоспостереження в даний час надзвичайно актуальна і широко використовується в житті людини. З появою цифрових відеокамер, які досить недорого розробити, стало можливим обробляти дані вимірювань за допомогою комп'ютера. Прикладами таких завдань є захист периметра та внутрішньої частини об'єкта, виявлення та розпізнавання людей, розпізнавання руху. Системи відеоспостереження широко використовуються в боротьбі з тероризмом та наглядом за дорожнім рухом. Великий інтерес до алгоритмів відстеження руху призвів до поширення потужних комп'ютерів, наявності високоякісних камер за низькою вартістю та зростаючої потреби в автоматизованому відеоаналізі. Основними ключовими етапами відеоаналізу є: розпізнавання рухомого об'єкта, відстеження об'єкта за кадром та аналіз об'єктів для визначення їх поведінки. Виходячи з вищесказаного, ідея стартап-проекту полягає у створенні підсистеми для поліпшення якості процесу розпізнавання об'єктів у відеопотоці за допомогою комбінованого методу ідентифікації. Для реалізації цього під-елемента було розглянуто та проаналізовано, а також узагальнено в таблицях:

- 1) зміст ідеї;
- 2) можливі сфери застосування;
- 3) переваги, які отримує користувач продуктів (для кожної галузі застосування);
- 4) основні відмінності порівняно з існуючими рішеннями.

Таблиця 4.1 – Ідея стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розроблену підсистему можна застосувати до багатьох систем відео-аналітики, а також в роботизованих системах	1. Автоматизоване спостереження, що виявляє підозрілу активність	Можливість використання програмного забезпечення з отриманням високої точності розпізнавання, навіть при неякісних вхідних даних в режимі реального часу
	2. Взаємодія людина-комп'ютер, наприклад, слідування за поглядом для вводу даних, розпізнавання жестів	
	3. Навігація машин, що пов'язана з планування маршруту та униканням перешкод.	

Проаналізуємо технічні та економічні переваги ідей. Даний аналіз забезпечує:

- визначення переліку техніко-економічних властивостей та характеристик ідеї;
- визначення попереднього кола конкурентів (конкуруючих проектів) або товарів-замінників або подібних товарів, що вже існують на ринку, та збирає інформацію про значення техніко-економічних показників для ідеї власного проекту та конкуруючих проектів відповідно до до списку вище;
- проводиться порівняльний аналіз показників: для правильної ідеї визначаються показники, що мають найгірші значення (W, слабкі), подібні (N, нейтральні), найкращі значення (S, сильні) (табл. 4.2).

Певний перелік слабких, сильних і нейтральних характеристик і властивостей ідеї потенційного товару є основою для формування його конкурентоспроможності.

Таблиця 4.2 – Аналіз сторін проектної ідеї

№	Техніко-економічні характеристики ідеї	Товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S(силына сторона)
		Мій проект	Програмне забезпечення для розпізнавання			
1	Зручний інтерфейс	Програмний додаток має головне вікно, в якому відображається відеопотік з розпізнаними об'єктами, можливість зберегти зображення кадру.	Має головне вікно, в якому відображається відеопотік з розпізнаними об'єктами.			+
2	Ціна	Має низьку вартість	Тільки платне програмне забезпечення			+
3	Кросплатформеність	Працює на ОС Windows	Працює на ОС Windows		+	
4	Вихідний код	Невеликий розмір вихідного коду	Суттєвий розмір вихідного коду			+
5	Наявність у відкритому доступі	Наявне	Відсутнє		+	

4.2 Технологічний аудит ідеї проекту

В цьому підрозділі було реалізовано аудит технології. З його допомогою буде реалізовано ідею проекту.

Щоб визначити доцільній проекту та ідеї потрібно проаналізувати наступні елементи:

- технологія доцільна для використання при виробництві програмного забезпечення до певної ідеї
- існують такі ідеї чи потребують певного вдосконалення або розробки
- доступність технології

Провівши аналіз таблиці можемо зробити висновок що бібліотека OpenCV, та мова програмування Python були обрані за рахунок своєї універсальності та доступності.

Таблиця 4.3 – технологічна доцільність проектної ідеї

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Розроблення підсистеми для розпізнавання об'єктів відеопотоку	Середовище розробки PyCharm	Наявна	Доступна
	Бібліотека OpenCV	Наявна	Доступна
	Мова програмування Python	Наявна	Доступна

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можуть бути використані під час реалізації на ринку проекту, та ринкових загроз, які можуть перешкодити проекту, дозволяє планувати розвиток проекту з урахуванням ринкового середовища, потенційні потреби споживачів та конкуруючі проектні пропозиції. Спочатку був проведений аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4)

Таблиця 4.4 – Попередній опис стартових проектів

Показники стану ринку (найменування)	Характеристика
Кількість головних гравців, од	2
Загальний обсяг продаж, грн/ум. Од	25000
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу (вказати характер обмежень)	-
Специфічні вимоги до стандартизації та сертифікації	-
Середня норма рентабельності у галузі (або по ринку), %	15

Середню прибутковість у галузі порівнювали із банківською процентною ставкою за інвестиціями. Останній менший, тому має сенс інвестувати в цей проект. Згідно з аналізом таблиці 4.4 зроблено висновок про те, що ринок привабливий для входу. Крім того, були визначені потенційні групи споживачів, їх характеристики та встановлено орієнтовний перелік вимог до продукції для кожної групи (таблиця 4.5).

Таблиця 4.5 – Характеристика потенційний споживачів

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Підсистема для розпізнавання об'єктів в відеопотоці в реальному часі	Компанії, приватні підприємства специфіка роботи яких пов'язана з використанням комп'ютерного зору	Відмінності у сферах діяльності компаній та приватних підприємств	Прийнятна цінова політика, зручний інтерфейс.

Проаналізувавши цільову аудиторію проведемо аналіз ринку. В таблицях наведено та проаналізовано впливові фактори загроз та фактори можливостей які сприяють реалізації проекту.

Таблиця 4.6 – Зміст загроз та їх фактори

Фактор	Зміст загрози	Можлива реакції компанії
Конкуренція	Вихід на ринок продуктів з кращими показниками	Вдосконалення використовуваних алгоритмів Популяризація продукту за рахунок рекламної кампанії
Зростання вимог користувачів	Користувачам необхідна підсистема, що буде забезпечувати захист від нових видів загроз	Розширення функціоналу продукту Популяризація продукту за рахунок рекламної кампанії

Було досліджено як відреагує компанія на конкуренцію, при появі нових способів які можливі реакції можна використати з врахуванням можливостей.

Таблиця 4.7 – Зміст можливостей та їх фактори

Фактор	Зміст можливості	Можлива реакції компанії
Конкуренція	Майже повна відсутність аналогів	
Поява нових способів	Нові методи для підвищення якості розпізнавання	Оптимізація існуючого рішення впровадженням нових алгоритмів та способів

Далі провели аналіз пропозицій, риси конкуренції на ринку та аналіз конкуренції.

Таблиця 4.8 – Аналіз та модель конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємництва (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції – монополія	На ринку присутні декілька компаній-конкурентів, але їх товар дещо відрізняється між собою.	Підтримка якості продукту та постійні вдосконалення
2. За рівнем конкурентної боротьби – міжнародний	Компанії-конкуренти з інших країн	Розробити універсальну архітектуру продукту, що дозволить легко адаптувати продукт до інших галузей
3. За галузевою ознакою – міжгалузєва	Продукт може бути використаний для інших галузей	Постійне вдосконалення та оновлення продукту, що сфокусовано на первісну ціль
4. Конкуренція за видами товарів – товарно-видова	Конкуренція між видами програмного продукту, їх якістю та оновленням	Створити програмний продукт, враховуючи недоліки конкурентів та актуальність функціоналу
5. За характером конкурентних переваг – нецінова	Вдосконалення технології створення продукту для низької собівартості	Покращення моделі функціонування продукту. Використання дешевих актуальних технологій, що дозволяють дотриматись вимог якості продукту
6. За інтенсивністю – не марочна	Бренд присутній, але його роль незначна	Проведення рекламних компаній

Згідно таблиці 4.9 зробимо висновок. Реалізувати проект можливо, враховуючи конкурентну ситуацію на ринку. Також повинні бути реалізовані ідеї які конкурентноспроможні на ринку. Проаналізувавши конкурентність ринку та врахувавши певні норми та характеристики проекту які наведені в таблиці 4.2, певні вимоги що до товару які наведені в таблиці 4.5 та характеристики

маркетингових факторів (таблиця 4.6 та 4.7) визначаються та мають обґрунтування.

Таблиця 4.9 – Аналітика конкуренції в сфері діяльності

Складові аналізу	Прямі конкуренти у галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
	Програмне забезпечення для розпізнавання	Наявність вже існуючих рішень	-	Контроль якості продукту	Авторитет конкурентів на ринку, більш широкий функціонал
Висновки	Досить інтенсивна боротьба з конкурентами на ринку гравцями	Є можливість виходу на ринок, але є конкуренти. Строки – 12 місяців	-	Споживачі диктують умови на ринку: надійний, швидкий та точний програмний продукт для захисту веб-орієнтованих систем	Необхідно випускати програмний продукт не гірше, ніж у конкурентів та розширяти функціонал

За визначеними факторами конкурентоспроможності (табл. 4.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.10 Аналіз факторів унікальності

Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для конкурентних проектів значущим)
Цінова політика	Оптимальна – товар доступний для значної кількості споживачів.
Висока якість	Програмне забезпечення розроблено згідно стандарту і є ліцензованим.
Актуальність	Реалізована підсистема 81 актуальна у використанні в різних галузях

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

Фактор конкурентоспроможності	Бали 1- 20	Рейтинг товарів-конкурентів у порівнянні						
		-3	-2	-1	0	1	2	3
Цінова політика	15				+			
Висока якість	20						+	
Актуальність	15		+					

Заключним кроком ринкового аналізу можливостей проекту є проведення SWOT-аналізу сильних та слабких сторін, беручи до факту матриці для аналізу можливостей та проблем (таблиця 4.12) на основі деяких ринкових загроз та можливостей, а також сильних та слабких сторін які слід підготувати (Таблиця 4.11) Перелік ринкових загроз та можливостей було встановлено на основі аналізу загроз та факторів маркетингового середовища.

Ринкові загрози та ринкові можливості - це наслідки (прогнозовані результати) факторів, що впливають, і, з іншого боку, ще не реалізовані на ринку і мають певну ймовірність реалізації. Наприклад: зниження доходу потенційних споживачів є фактором загрози, на основі якого можна робити прогнози щодо зростаючого значення цінового фактору у виборі товарів і, отже, щодо цінової конкуренції (що вже є ринковою загрозою).

Проведемо SWOT-аналіз проекту, визначимо певні ринкові альтернативи та визначимо скільки часу потрібно на їх реалізацію з урахуванням конкурентів. Альтернативи були проаналізовані та визначені в таблиці 4.13.

Таблиця 4.12 – SWOT-аналіз проектування

Сильні сторони:	Слабкі сторони:
Поширення реклами, яка буде ознайомлювати користувачів з новим товаром Застосування новітніх технологій при розробці Залучення професійних і кваліфікованих кадрів	Недостатній імідж, оскільки програмний продукт тільки починає роботу на ринку
Можливості:	Загрози:
Збільшення попиту на продукцію Залучення висококваліфікованого персоналу Більшість клієнтів зацікавлені у використанні програмного продукту	Нестабільна економічна і політична ситуація у країні Наявність прямих конкурентів

Таблиця 4.13 – Впровадження альтернатив

Альтернатива (орієнтований комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
Безкоштовне розповсюдження створеного продукту	70%	12 місяців
Створення продукту з подальшим розповсюдженням за певну оплату	75%	12 місяців

Проробивши аналіз оберемо альтернативу під другим номером.

4.4 Аналіз ринкової стратегії проекту

Ринкова стратегія відіграє важливу роль в постанові проекту. Було проведено аналіз стратегії охоплення ринку, також визначено цільові групи клієнтів (таблиця 4.14). Проаналізувавши потенційних споживачів оберемо цільові групи, які будуть розглядати товар до використання. Оберемо стратегію диференційованого маркетингу. Передбачає роботу з декількома сегментами.

Таблиця 4.14 – Цільові групи потенційних споживачів

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів прийняти продукт	Орієнтований попит в межах цільової групи (сегменти)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Компанії діяльність яких пов'язана з системами розпізнавання	Висока	Високий	Сильна	Складно
Приватні підприємства міського та міжнародного рівня, діяльність яких пов'язана з системами розпізнавання	Висока	Високий	Сильна	Складно
Приватні підприємства, обласного рівня	Висока	Високий	Помірна	Середня складність
Підприємства регіонального характеру	Висока	Середній	Слабка	Просто

Щоб закріпитися в ринкових сегментах сформуємо стратегію розвитку яка наведена в таблиці 4.15.

Таблиця 4.15 – Поверхневий аналіз стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Створення продукту з подальшим розповсюдженням за певну оплату	Визначити потреби кожної з груп, розробити відповідно до них стратегії приваблення клієнтів та маркетингової комунікації	Цінова політика, універсальність продукту (миттєве практичне застосування), якість та актуальність	Стратегія диференціації

Далі обрано фактори поведінки яка передбачає конкурентність.

Таблиця 4.16 – Стратегія конкурентності на ринку

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
«першопроходець»	Забирати існуючих	Ні	Стратегія зайняття конкурентної ніші

За результатами аналізу сегментів які будуть примінені до постачальника, та до розробленого продукту (таблиця 4.5), та певних залежностей які приписані стратегіям розвитку (таблиця 4.15), та розробленою конкурентною поведінкою (таблиця 4.16), буде реалізовано фактор позиціонування на ринку (таблиця 4.17). Даний фактор реалізований на ринковій позиції, за допомогою якого замовники ідентифікують проект (марку).

Таблиця 4.17 – Стратегія позиціонування на ринку

Вимоги до товару цільовою аудиторією	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувану комплексну позицію власного проекту (три ключових)
Просте використання, надійність, швидкість, наявність оновлень та наявність документації для програмного продукту	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Надійність Швидкість Просте використання

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії на ринку.

4.5 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, яку отримає споживач. Для цього у табл. 4.18 підсумовано результати попереднього аналізу конкурентоспроможності товару. Концепція товару – письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигідностей, які він обіцяє певній групі споживачів.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
Швидкість обробки даних	Швидка обробка вхідних даних	Алгоритми реалізовані найоптимальнішим способом та забезпечують найменшу затримку при обробці
Зручність застосування	Не потребує складного налаштування для впровадження у систему	Заощаджує час впровадження у систему за рахунок створеної архітектури
Точність обробки даних	Висока точність ідентифікації об'єктів	Висока вірогідність виявлення об'єктів за рахунок використання сучасних алгоритмів

Розроблена трирівнева модель збуту товару: конкретизується ідея товару та / або послуги, її фізичні компоненти, характеристики процесу її постачання яка наведена в таблиці 4.19. Рівень 1: Коли формується ідея товару, вирішується питання, яким способом вирішити потребу чи проблему, яка його головна перевага. Ця проблема безпосередньо пов'язана з формуванням технічного завдання при розробці проектно-конструкторської документації. 2-й рівень: являє собою рішення про те, як товар буде продаватися в реальному житті, включає в собі якість, властивості, дизайн, упаковку, ціну. 3-й рівень: товари з підкріпленням (підтримка) - додаткові послуги та переваги для споживача, створені на основі продукту за задумом, та товару в реальних робочих характеристиках (гарантії якості, доставка, умови оплати тощо).

Після формування моделі маркетингу товару слід зазначити, що проект буде захищений від копіювання за допомогою ноу-хау. Наступним кроком є

визначення цінових обмежень, якими слід керуватися при встановленні ціни на потенційний товар, що передбачає аналіз ціни на подібні або замінні товари, а також аналіз ціни. рівень доходу цільової групи споживачів (табл. 4.20). Аналіз проводився експертним методом.

Таблиця 4.19 – Характеристика 3-х рівнів моделей товару

Рівні товарів	Сутність та складові		
Товар за задумом	Швидке та точне розпізнавання об'єктів відеопотоку в реальному часі		
Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
Товар із підкріпленням	1.функція для предобробки вхідних даних 2. функція для зчитування кадрів в окремому потоці		
	Якість: точна ідентифікація об'єктів в реальному часі		
	Пакування: відсутнє		
	Марка: Detection		
Товар із підкріпленням	До продажу: відсутнє		
	Після продажу: персональна підтримка з можливістю розширення функціоналу під власні потреби		
Вихідний код закритий. На ідею зареєстровано патент.			

Таблиця 4.20 – Призначення меж для встановлення вартості

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
50000 грн	44000 грн	У всіх трьох груп достатній рівень доходів	Базова покупка 30000 грн Подальша персональна підтримка в обслуговуванні 3000 грн

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (табл. 4.21).

Таблиця 4.21 – Фактори збуту та їх характеристика

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які забезпечать точність та швидкість розпізнавання об'єктів. Вони цікавляться існуючими рішеннями та інноваціями у сфері комп'ютерного зору.	Встановлення контактів із споживачами і підтримання їх. Формування попиту і стимулювання збуту. Дослідницька робота зі збору маркетингової інформації. Забезпечення зворотного зв'язку зі споживачами.	Один (від виробника одразу споживачу)	Прямий канал збуту до споживача, мінімізувати збутові витрати розвиток маркетингового спілкування із споживачем

Заключною частиною маркетингової програми є розробка концепції маркетингової комунікації, заснованої на заздалегідь обраній основі позиціонування, визначених особливостей поведінки клієнтів (табл. 4.22). Результатом цього підрозділу стала ринкова (маркетингова) програма, що включає концепції товарів, продажі, просування та попередній аналіз цінових можливостей на основі цінностей та потреб потенційних споживачів, конкурентних переваг ідея, стан та динаміка ринкового середовища, в якому реалізовується проект, та відповідна альтернатива обраній поведінці ринку.

Таблиця 4.22 – Поняття маркетингової комунікації

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які забезпечать точність і швидкість ідентифікації об'єктів. Вони цікавляться існуючими рішеннями та інноваціями у сфері комп'ютерного зору	Конференції, інтернет конференції, семінари, огляд професійної літератури, інтернет, періодичні видання у різноманітних (профільних) галузях.	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Створення репутації фірми — виробнику чи посереднику; збільшення чистого прибутку та рентабельності фірми; збільшення потоків покупців та обсягів продажу; стабілізація обсягів продажу в період зменшення попиту та загального спаду ділової активності.	

Висновки за розділом.

Розібравши аспекти програмного продукту як проекту можемо зробити висновки що у даному проекті є можливість закріпитися та мати існування на ринку, так як сама тема досить актуальна та користується підвищеним попитом за рахунок невеликої конкуренції, а якісна розробка проекту надає нам перевагу.

Даний проект заключає в собі реалізацію швидкого розпізнавання та ідентифікації об'єкта в подальшому його супроводу за рахунок вдосконаленого алгоритму дій.

Потрібно також обрати альтернативу яка дозволить нам реалізувати розроблення програмного продукту з його розповсюдженням по інтернет мережі

ВИСНОВКИ

Підбиваючи підсумки хотілося б сказати що дана тема є новітньою в даний час і набирає великих обертів в галузі відеоспостереження. У даній магістерській роботі досліджено проблематику існуючих рішень пошуку та супроводу об'єктів в відеопотоці, а також розглянуті способи та наведені рекомендації що до покращення розпізнавання об'єктів. Проведені порівняння алгоритмів які відповідають за ідентифікацію об'єкта. Описані відомі способи відслідковування об'єктів від яких ми відштовхувались.

Дослідження дали змогу підбити наступні підсумки:

- провели аналіз методів розпізнавання та ідентифікації об'єктів
- провели удосконалення алгоритму пошуку об'єктів в відеопотоці, використовуючи фільтр шуму `erosion()`, провели дослідження щодо покращення гамма-корекції, застосували спосіб багатопоточності на практиці
- розроблене ПЗ для розпізнавання та супроводу об'єктів з використанням бібліотеки OpenCV за допомогою Python.

Отже за даними результатами можна зробити кінцевий висновок, що метод який ми модифікували дає можливість швидше і якісніше ідентифікувати та відслідковувати об'єкти, також при сильних шумах швидше взаємодіяти з серією кадрів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ИНТУИТ. URL: <http://www.intuit.ru/studies/courses/10621/1105/lecture/17985>
2. Розпізнавання об'єктів в реальному часі. URL: <https://proglib.io/p/real-timeobject-detection/> (дата звернення 20.10.2019).
3. OpenCV library. URL: <https://opencv.org/> (дата звернення 15.09.2019).
4. NumPy. URL: <http://www.numpy.org/> (дата звернення 07.10.2019).
5. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs / [L.-C. Chen, G. Papandreou, I. Kokkinos, et al.] URL: <https://arxiv.org/pdf/1606.00915.pdf> (дата звернення 12.10.2019).
6. Вьюгин В.В. Математические основы теории машинного обучения и прогнозирования / В.В. Вьюгин. – М.: МЦНМО, 2013. 387 с.
7. Лукьяница А. А. Обработка видеоизображений / А.А. Лукьяница, А.Г. Шишкин; Под ред. Поздняков С.А. — М.: Ай-Эс-Эс Пресс, 2009. 764 с.
8. Алпатов Б.А., Бабаян П.В., Балашов О.Е., Степашкин А.И. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление. М., Радиотехника, 2008. 176 с.
9. Гансалес Р., Вудс Р. Цифровая обработка изображений. М., Техносфера, 2005. 1012 с.
10. Сойфер В. А. Методы компьютерной обработки изображений. М., ФИЗМАТЛИТ, 2003. 784 с.
11. Shi J., Tomasi C. Good features to track. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Seattle, USA, 1994, pp. 593–600.

12. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs / [L.-C. Chen, G. Papandreou, I. Kokkinos, et al.]. URL: <https://arxiv.org/pdf/1606.00915.pdf>(дата звернення 10.09.2019).

13. Comaniciu D., Ramesh V., Meer P. Kernel-based object tracking. IEEE 91 Transaction on Pattern Analysis and Machine Intelligence, 2003, pp. 564–577.

14. Isard M., Blake A. Condensation — conditional density propagation for visual tracking. International Journal of Computer Vision, 1998, pp. 5–28.

15. Freeman W.T., Roth M. Orientation histograms for hand gesture recognition. Proceedings of the Workshop on Automatic Face and Gesture Recognition. Zurich, Switzerland, 1995, pp. 296–301.

16. Kwon J., Lee K.M. Visual Tracking Decomposition. Conference on Computer Vision and Pattern Recognition, 2010.

17. Babenko B., Yang M.-H., Belongie S. Visual Tracking with Online Multiple Instance Learning. Conference on Computer Vision and Pattern Recognition, 2009.

18. Lowe G. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, vol. 60, no. 2, pp. 91–110.

19. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features. Conference on Computer Vision and Pattern Recognition, 2001.

20. Апальков И.В., Хрящев В.В. Удаление шума из изображений на основе нелинейных алгоритмов с использованием ранговой статистики. Ярославский государственный университет имени П.Г. Демидова, 2007.