

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра комп'ютерних наук

**Пояснювальна записка**

до магістерської кваліфікаційної роботи

**на тему «Інформаційна технологія ідентифікації пішоходів у відеопотоці на  
основі згорткових нейромереж»**

Виконав: студент 2 курсу,  
групи 2КН-19 м  
спеціальності 122 «Комп'ютерні науки»  
**Радченко Д. Ю.**

Керівник: к.т.н., доц. Колесницький О.К.

Рецензент: д. т. н., проф. Ліщинська Л. Б.

Вінниця  
2020

ЗАТВЕРДЖУЮ  
Завідувач кафедри \_\_\_ КН \_\_\_  
д.т.н., проф.. Яровий А.А.

\_\_\_\_\_ (підпис)  
“ \_\_\_ ” \_\_\_\_\_ 2020 року

## ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра зі спеціальності: 122 – «Комп'ютерні науки»

---

08-22.МКР.022.19.000.ПЗ

Магістранта групи 2КН-19м Радченка Дениса Юрійовича

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія ідентифікації пішоходів у відеопотоці на основі згорткових нейромереж»

Вхідні дані: відеопотік роздільною здатністю не менше 300x300, вихідна інформація – виділення зображення пішохода прямокутним контуром на екрані, обсяг навчальної вибірки – не менше 200 зображень, обсяг тестової вибірки – не менше 15 зображень, середовище програмування – об'єктно-орієнтоване.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: Граф-схема алгоритму роботи програмного забезпечення ідентифікації пішоходів у відеопотоці, структура інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі, структура нейронної мережі; результати роботи програми ідентифікації пішоходів у відеопотоці.

2. Текстова (пояснювальна записка): вступ, аналіз предметної області ідентифікації пішоходів у відеопотоці, розробка інформаційної технології ідентифікації пішоходів у відеопотоці, програмна реалізація інформаційної технології ідентифікації пішоходів у відеопотоці, економічна частина, висновки, перелік використаних джерел, додатки.



## АНОТАЦІЯ

Дана магістерська кваліфікаційна робота присвячена розробці програмного забезпечення для ідентифікації пішоходів у відеопотоці за допомогою згорткової нейронної мережі. Були розглянуті основні методи ідентифікації пішоходів у відеопотоці, визначені їх переваги і недоліки, та на основі цього був зроблений вибір на користь згорткової нейронної мережі. Було досліджено структуру, математичну модель та алгоритм роботи обраної мережі, які слугували базисом для розробки алгоритму програмного забезпечення. Було визначено оптимальне для потенційного користувача програмне середовище, алгоритм роботи програми та в кінцевому підсумку на основі проведених досліджень був побудований програмний засіб ідентифікації пішоходів у відеопотоці. Створене програмне забезпечення написане на мові програмування Python та порівняно з аналогом має кращу на 16,1% достовірність ідентифікації пішоходів у відеопотоці.

## **ABSTRACT**

The master's degree work is devoted to the development of software for the identification of pedestrians in the video stream using a convolutional neural network. The main methods of pedestrian identification in the video stream were considered, their advantages and disadvantages were identified, and based on this, a choice was made in favor of a convolutional neural network. The structure, mathematical model and algorithm of the selected network were studied, which served as the basis for the development of the software algorithm. The optimal software environment for the potential user, the algorithm of the program operation were determined and finally on the basis of the conducted researches the software means of identification of pedestrians in the video stream was built. The created software is written in the Python programming language and has a better 16.1% accuracy of pedestrian identification in the video stream compared to its analogue..

## ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІДЕНТИФІКАЦІЇ ПІШОХОДІВ У ВІДЕОПОТОЦІ.....	12
1.1 Класифікація існуючих методів ідентифікації пішоходів у відеопотоці.....	12
1.1.1 Метод локальних бінарних ознак.....	12
1.1.2 Метод гістограми орієнтованих алгоритмів.....	14
1.2 Існуючі проблеми в задачі ідентифікації пішоходів і шляхи їх вирішення.....	15
1.3 Обґрунтування вибору аналогу до програми ідентифікації пішоходів у відеопотоці.....	16
1.4 Висновок.....	17
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІДЕНТИФІКАЦІЇ ПІШОХОДІВ У ВІДЕОПОТОЦІ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ.....	18
2.1 Обґрунтування вибору згорткової нейронної мережі для розв'язання задачі ідентифікації пішоходів у відеопотоці.....	18
2.2 Вибір архітектури згорткової нейронної мережі для ідентифікації пішоходів у відеопотоці.....	20
2.3 Аналіз математичної моделі згорткової нейронної мережі.....	31
2.4 Навчання згорткових нейронних мереж.....	35
2.5 Структура інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі.....	40
2.6 Висновок.....	42
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІДЕНТИФІКАЦІЇ ПІШОХОДІВ У ВІДЕОПОТОЦІ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ.....	43
3.1 Архітектура і алгоритм роботи SSD.....	43
3.2 Заходи з поліпшення моделі SSD.....	46
3.3 Обґрунтування вибору мови, середовища програмування та бібліотек...	51
3.4 Програмна реалізація інформаційної технології ідентифікації пішоходів у відеопотоці.....	55
3.5 Тестування та аналіз результатів роботи програми ідентифікації пішоходів у відеопотоці на основі згорткових нейромереж.....	57

3.6 Висновок .....	64
4 ЕКОНОМІЧНА ЧАСТИНА.....	65
4.1 Оцінювання комерційного потенціалу розробки.....	65
4.2 Прогнозування витрат на виконання наукової роботи та впровадження результатів.....	70
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.....	75
4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.....	76
4.5 Висновок .....	80
ВИСНОВКИ.....	81
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83
Додаток А Інструкція користувача.....	86
Додаток Б Лістинг програми.....	88
Додаток В Графічна частина.....	93

## ВСТУП

**Актуальність.** У наш час стрімко розвивається область досліджень у сфері комп'ютерного зору та штучного інтелекту. Комп'ютерний зір та штучний інтелект знаходять застосування в багатьох галузях промисловості і сферах діяльності. Зокрема, багато автомобільних компаній ведуть дослідження в області комп'ютерного зору, а саме - розробка систем автоматичної ідентифікації пішоходів з метою підвищення безпеки на дорогах загального користування. Без алгоритмів, що забезпечують високу точність ідентифікації пішоходів на реальному відео неможливо впровадження таких систем.

Ідентифікація пішоходів є однією з основних задач комп'ютерного зору. Основним застосуванням технології ідентифікації пішоходів є її використання в автомобілях для поліпшення системи безпеки. Аварії за участю пішоходів є другим джерелом травм і смертей у світі. Тому вдосконалені системи допомоги водієві і, зокрема, системи захисту пішоходів стали важливою областю досліджень для підвищення безпеки руху. Також розпізнавання пішоходів знаходить застосування в безпілотних автомобілях. В даний час багато компаній займається розробкою своїх продуктів для масового ринку, включаючи General Motors, Volkswagen, Audi, BMW, Volvo, Nissan, Google.

В даний час існує два основних напрямки роботи: одне засноване на зображеннях видимого спектру, а інше - на основі теплового інфрачервоного випромінювання для використання вночі. Перші набули більшого поширення через більшу доступність сенсорів, що працюють у видимому спектрі, їх меншу ціну, високе співвідношення сигнал/шум і високу роздільну здатність, а також тому, що більшість нещасних випадків відбувається в денний час. У даній роботі розглядаються зображення у видимому спектрі.

Завдання ідентифікації пішоходів у відеопотоці відноситься до завдань комп'ютерного зору. Серед завдань комп'ютерного зору можна виділити наступні:



- Класифікація. Класифікація найбільш відома проблема комп'ютерного зору. Вона полягає у віднесенні зображення до однієї з множини категорій.
- Локалізація. Локалізація визначає місце розташування одного об'єкта на зображенні. Вона може комбінуватися з класифікацією для визначення місця розташування об'єкта та класифікації його.
- Виявлення об'єкта. Виявлення об'єкта включає в себе завдання класифікації і локалізації декількох об'єктів на зображенні водночас.

Завдання ідентифікації пішоходів представляє собою завдання виявлення об'єкта, так як необхідно як виявити об'єкт, так і правильно його класифікувати як об'єкт, при тому, що шуканих об'єктів на зображенні може бути кілька.

Магістерська робота присвячена розпізнаванню образів, а саме ідентифікації пішоходів у відеопотоці. Розпізнавання образів вирішує задачу виділення істотних ознак та їх віднесення до певного класу, що характеризують даний образ, із загальної маси даних. У даній роботі нейронна мережа використовуються для побудови програмних засобів для ідентифікації пішоходів у відеопотоці.

#### **Зв'язок роботи з науковими програмами, планами, темами.**

Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

**Мета і завдання досліджень.** Метою магістерської кваліфікаційної роботи є підвищення достовірності ідентифікації пішоходів у відеопотоці програмними засобами за рахунок застосування згорткових нейронних мереж.

Для досягнення мети розробки необхідно виконати такі задачі:

- провести аналіз проблеми розв'язання задачі ідентифікації пішоходів у відеопотоці;

- розглянути існуючі методи вирішення задачі ідентифікації пішоходів у відеопотоці та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- розробити метод ідентифікації пішоходів у відеопотоці;
- сформулювати стадії інформаційної технології, розробити структуру та алгоритм роботи програмного засобу;
- виконати програмну реалізацію запропонованої інформаційної технології;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

**Об'єкт дослідження** – процес ідентифікації пішоходів у відеопотоці з використанням згорткових нейронних мереж.

**Предмет дослідження** – інформаційна технологія та програмні засоби ідентифікації пішоходів у відеопотоці з використанням згорткових нейронних мереж та достовірність їх роботи.

**Методи дослідження.** У роботі використані наступні методи наукових досліджень: системного аналізу, розпізнавання образів, теорії штучних нейронних мереж для реалізації інформаційної технології ідентифікації пішоходів у відеопотоці, методи математичної статистики для розробки процесу ідентифікації пішоходів у відеопотоці та обрахунків результатів експериментів із програмним засобом, об'єктно-орієнтованого програмування.

**Наукова новизна одержаних результатів.**

1. Набула подальшого розвитку інформаційна технологія ідентифікації пішоходів у відеопотоці, яка відрізняється використанням згорткової нейронної мережі, що дозволило підвищити достовірність ідентифікації пішоходів у відеопотоці.

2. Вдосконалено процес навчання згорткової нейронної мережі, який відрізняється введенням аугментації тренувальних даних та зміною розміру вікна передбачення об'єктів, що дозволило підвищити достовірність розпізнавання нейронної мережі.

**Практичне значення** одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення ідентифікації пішоходів у відеопотоці.

Запропонована інформаційна технологія сприяє підвищенню достовірності процесу ідентифікації пішоходів у відеопотоці, зокрема:

- розроблено алгоритм роботи програмного забезпечення ідентифікації пішоходів у відеопотоці на основі згорткової штучної нейронної мережі;
- розроблено програмні засоби для ідентифікації пішоходів у відеопотоці на основі згорткової штучної нейронної мережі;

**Достовірність теоретичних положень** магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології ідентифікації пішоходів у відеопотоці. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

**Особистий внесок здобувача.** Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз процесу ідентифікації пішоходів у відеопотоці на основі згорткової штучної нейронної мережі та методів підвищення достовірності [1].

**Апробація результатів роботи.** Результати досліджень апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи-2020», Вінниця, жовтень 2019 – травень 2020 року [1].

**Публікації.** За результатами досліджень опубліковано одні тези доповіді на науково-технічній конференції.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІДЕНТИФІКАЦІЇ ПІШОХОДІВ У ВІДЕОПОТОЦІ

## 1.1 Класифікація існуючих методів ідентифікації пішоходів у відеопотоці

Основна ідея в завданні розпізнаванні пішоходів - виділення ознак, відповідних шуканому об'єкту. Це дозволяє виявляти об'єкт не цілком, що викликає складнощі у зв'язку з високою мінливістю об'єкта, а виявляти об'єкт за характерними йому ознаками. У комп'ютерному зорі і обробці зображень, ознака - це певна структура в даних зображення, яка може бути представлена по-різному. Наприклад, колір конкретної області на зображенні може бути представлений у вигляді значення середнього кольору в області (три скаляра) або гістограми кольору (три функції).

У задачі розпізнавання об'єктів існують різні підходи до виділення ознак і їх подальше використання. Основні існуючі методи і алгоритми виявлення об'єктів можна класифікувати наступним чином [2]:

- Метод локальних бінарних шаблонів;
- Гістограма орієнтованих градієнтів;
- Згорткові нейронні мережі.

### 1.1.1 Метод локальних бінарних ознак.

Локальні бінарні шаблони вперше запропоновані в 1996 році для аналізу текстури напівтонових зображень [3]. Локальні бінарні шаблони - це певний вид ознаки, що представляє собою оператор. Локальний бінарний шаблон являє собою опис околиці пікселя зображення в двійковому поданні.

Базовий оператор локального бінарного шаблону, який застосовується до пікселя зображення, використовує вісім пікселів околиці, приймаючи значення інтенсивності центрального пікселя в якості порога. Пікселі зі значенням інтенсивності більшим чи рівним значенню інтенсивності центрального пікселя

приймають значення рівні «1», інші приймають значення рівні «0». Таким чином, результатом застосування базового оператора локального бінарного шаблону до пікселя зображення є восьмирозрядний бінарний код, який описує околицю цього пікселя. Базовий оператор зображений на рисунку 1.1.

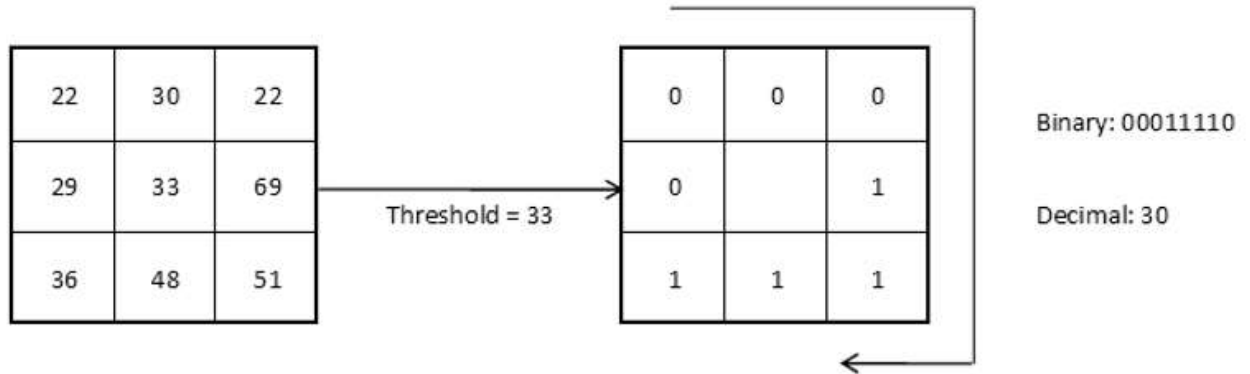


Рисунок 1.1 - Базовий оператор локального бінарного шаблону

Використання кругової околиці і білінійної інтерполяції значень інтенсивностей пікселів дозволяє побудувати локальний бінарний шаблон з будь-якою кількістю точок  $P$  і радіусом  $R$ , що зображено на рисунку 1.2.

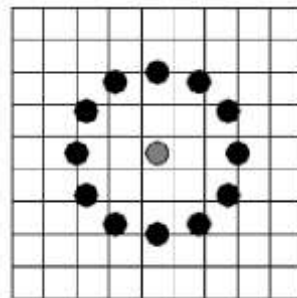


Рисунок 1.2 - Розширений оператор локального бінарного шаблону

Таким чином, бінарні коди несуть в собі інформацію про структуру зображення - кінці ліній, межі, кути, плями. Приклад ознак, що детектуються локальними бінарними шаблонами, зображений на рисунку 1.3

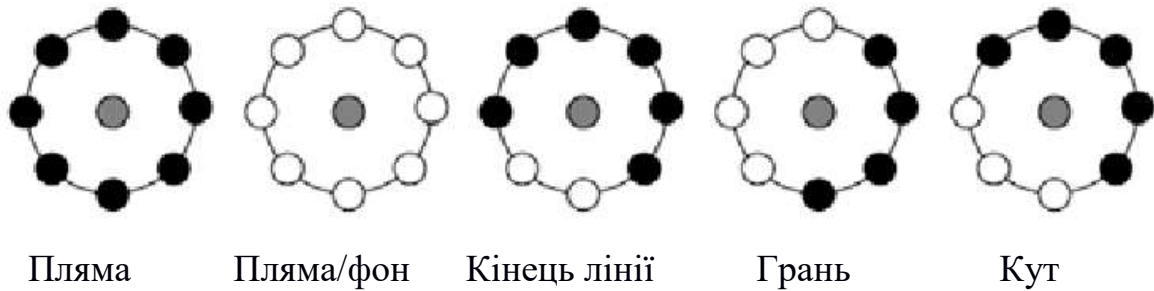


Рисунок 1.3 - Приклад локальних ознак, які детектуються локальними бінарними шаблонами

### 1.1.2 Метод гістограми орієнтованих градієнтів.

Подальший розвиток ідея вилучення ознак із зображення отримала в алгоритмі гістограми орієнтованих градієнтів (HOG) [5]. Основна ідея алгоритму гістограми орієнтованих градієнтів - допущення, що зовнішній вид і форма об'єкта на ділянці зображення можуть бути описані розподілом градієнтів інтенсивності або напрямком країв - дескрипторами. Приклад представлення зображення за допомогою дескрипторів зображень на рис. 1.4



Рисунок 1.4 а — вихідне зображення, б- зображення, представлене у вигляді гістограми орієнтованих градієнтів

Реалізація цих дескрипторів може бути проведена шляхом поділу зображення на маленькі пов'язані області (осередку), і розрахунком для кожного осередку гістограми напрямків градієнтів або напрямків країв для

пікселів, що знаходяться всередині осередку. Комбінація цих гістограм і є дескриптором. Для збільшення точності локальні гістограми піддаються нормалізації по контрасту.

З цією метою обчислюється міра інтенсивності на великій фрагменті зображення, який називається блоком, і отримане значення використовується для нормалізації. Нормалізовані дескриптори мають кращу інваріантністю по відношенню до висвітлення.

Кінцевим кроком в розпізнаванні об'єктів з використанням гістограми орієнтованих градієнтів є класифікація дескрипторів за допомогою системи навчання з учителем

Переваги методу орієнтованих градієнтів:

- Оскільки HOG працює локально, метод підтримує інваріантність геометричних і фотометричних перетворень, за винятком орієнтації об'єкта. Подібні зміни з'являються тільки у великих фрагментах зображення.

- Розбиття простору, точне обчислення напрямків і сильна локальна фотометрична нормалізація дозволяють ігнорувати рух пішоходів, якщо вони підтримують вертикальне положення тіла.

## **1.2 Існуючі проблеми в задачі ідентифікації пішоходів і шляхи їх вирішення.**

Описані методи засновані на виділенні ознак. У цьому міститься основна проблема цих методів - структура ознак задається вручну різними способами. Це означає, що під кожну конкретну задачу необхідно сформувати власний набір ознак. Це може викликати складності в завданні ідентифікації пішоходів, у зв'язку з особливістю об'єктів, що виявляються:

- Об'єкти мають високу внутрішньокласову мінливість (одяг, освітлення, відстань, розмір і т.д.).

- Об'єкти можуть перекривати один одного.

Алгоритми, розглянуті раніше, можуть виділяти тільки дуже низькорівневі ознаки, такі, як межі, кути, плями і т.д., що не дозволяє оперувати більш високорівневими ознаками об'єктів. Дану проблему можна вирішити використанням згорткових нейронних мереж для виявлення пішоходів. Детальніше такі моделі будуть розглянуті далі.

### 1.3 Обґрунтування вибору аналогу до програми ідентифікації пішоходів у відеопотоці

Пошук програм для ідентифікації пішоходів у відеопотоці показав, що їх є дуже мало. Однією з програм-аналогів, яка вирішує поставлену задачу є програма під назвою RetinaNet [6]. Скріншот програми RetinaNet: зображено на рис. 1.5.



Рисунок 1.5 – Скріншот програми RetinaNet

Програма RetinaNet має кілька модифікацій, але побудована на використанні пірамідальної нейронної мережі, яка працює із попередньо отриманими ознаками зображення. Достовірність ідентифікації пішоходів у відеопотоці у RetinaNet становить в середньому 67,5%.



## 1.4 Висновок

В першому розділі було проведено аналітичний огляд відомих методів і засобів ідентифікації пішоходів у відеопотоці, виявлені їх можливості, переваги і недоліки. Обґрунтована перспективність застосування методів ідентифікації пішоходів у відеопотоці на основі згорткових нейронних мереж. Було наведено класифікацію методів ідентифікації пішоходів у відеопотоці, детально описано ідентифікатори пішоходів у відеопотоці на основі традиційних підходів. Було проаналізовано недоліки розглянутих ідентифікаторів пішоходів та запропоновано використовувати для побудови ідентифікаторів пішоходів згорткові нейронні мережі, оскільки вони мають переваги.

## 2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІДЕНТИФІКАЦІЇ ПІШОХОДІВ У ВІДЕОПОТОЦІ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ

### 2.1 Обґрунтування вибору згорткової нейронної мережі для розв'язання задачі ідентифікації пішоходів у відеопотоці

Моделі ідентифікації об'єктів, засновані на згорткових нейронних мережах, дозволяють виділяти більш високорівневі ознаки на зображенні. Ознаки виділяються в ієрархічному порядку. Від нижчих шарів до вищих верств нейронної мережі виділяються більш абстрактні ознаки.

Ідея даних моделей мотивована дослідженнями про зорову кору головного мозку. Зорова кора розташована в потиличній частині головного мозку. Вона ділиться на кілька зон - зорова зона V1 (visual area one), зорова зона V2 (visual area two), V3 ... V7. Зони відрізняються одна від одної фізіологією, архітектурою і положенням у корі. За нинішніми уявленнями, кожна зона зорової кори має наступні функції:

- в зоні V1 виділяються локальні ознаки невеликих ділянок зчитаного з сітківки зображення;
- V2 виділяє локальні ознаки, узагальнюючи їх і додаючи бінокулярний зір;
- V3 розпізнає колір, текстури об'єктів, з'являються перші результати їх сегментації і групування;
- V4 розпізнає геометричні фігури і обриси об'єктів нескладних форм. Крім того, тут найбільш сильна модуляція за допомогою уваги людини - активація нейронів в зоні V4 не рівномірна, а сильно залежить від того, на що свідомо чи несвідомо звернуто увагу;
- V5 розпізнає рух об'єктів, обриси яких виділені в зоні V4;
- V6 узагальнює дані про все зображенні, вона реагує на зміни по всьому полю зору;

- V7 розпізнає складні об'єкти, такі, як людські фігури;

Така функціональна спеціалізація добре відповідає тому, що зазвичай використовується в згорткових нейронних мережах - більш високі рівні потрібні для того, щоб виділяти найбільш загальні ознаки, відповідні абстрактним властивостям входу, а на нижніх рівнях ознаки більш конкретні.

Крім того, нейрони із зони V1 подаються на вхід не тільки в зону V2, але і безпосередньо в зону V5 - це також знайшло відображення в згорткових архітектурах. Також виходи нейронів із зони V1 і V2 обробляються відразу двома паралельними шляхами. Ця ідея використовується в згорткових нейронних мережах, коли вхід обробляється за допомогою відразу декількох розмірів фільтрів.

Зусієї зорової кори найбільш добре вивчена зона V1, первинна зорова кора: вона ближче всього до входів, досить просто влаштована і простіше зрозуміти, за що відповідають окремі нейрони і як вони між собою взаємодіють [7]. Нейрони в зоні V1 розташовуються не випадково: зона V1 містить повну карту полів зору обох очей, тобто близькі ділянки сітківки обробляються близькими нейронами в V1. При цьому локальна структура переноситься дуже точно, а на глобальному рівні присутні спотворення: по-перше, центральну ділянку поля зору сильно збільшено (половина всіх нейронів відповідає за 2% зору), по-друге, є геометричні спотворення - концентричні кола і радіальні лінії на зображенні перетворюються в вертикальні і горизонтальні лінії. Це дозволяє зберігати інваріантність зображення при зміні позиції і точки зору. Кожен нейрон в V1 працює з дуже маленьким ділянкою зображення - рецептивним полем, і між ними зберігаються просторові взаємозв'язки, подібні вихідній картинці.

Цю властивість також використано в згорткових нейронних мережах. Різні нейрони в зоні V1 розпізнають:

- Орієнтацію. Нейрон реагує на те, що освітленість уздовж діагоналі зображення висока, а по двох інших кутах низька, або на те, що освітленість

нижньої частини зображення вище, ніж верхньої. Таким чином, нейрони розпізнають межі зображення.

- Просторову частоту. Наскільки часто змінюється освітленість в межах рецептивного поля нейрона.

- Різницю між очима. В V1 у кожного нейрона два рецептивних поля, для кожного ока. Більшість нейронів відповідають за своє око, деякі розпізнають відмінності між тим, що бачать ліве і праве око.

- Колір. Нейрони розпізнають одне з трьох напрямків: червоний - зелений, синій - жовтий і чорний — білий.

Виходячи з вищесказаного видно, що обробка зображень в мозку влаштована як глибока нейронна мережа. Це наближення згорткових нейронних мереж до роботи зорової кори головного мозку дозволяє їм розпізнавати об'єкти на більш високому рівні абстракції з високою точністю. Далі будуть розглянуті ключові моделі ідентифікації об'єктів на основі згорткових нейронних мереж.

## 2.2 Вибір архітектури згорткової нейронної мережі для ідентифікації пішоходів у відеопотоці

Нейронна мережа, розроблена в 2012 році і відома як AlexNet, була першою моделлю згорткової нейронної мережі, що досягла в змаганні ILSVRC 2012 відсотка помилок 15,4% [8]. У моделі були запропоновані такі технології, як збільшення вхідних даних (аугментація) і дропаут - шар, який обнуляє виходи деяких нейронів, які обирають випадково і заново для кожного навчального прикладу. Ці методи показали перевагу згорткових нейронних мереж і використовуються по теперішній час. Архітектура мережі являє собою 5 згорткових шарів, шар субдискретизації, шар дропаутів і 3 повнозв'язних шари.

Стандартна модель виходу нейрона визначається як функція  $f$  від входу  $x$  як  $f(x)=\tanh(x)$  або  $f(x)=(1+e)^{-1}$ . У AlexNet запропонована функція  $ReLU$ . У

згорткових нейронних мережах функція *ReLU* збільшує продуктивність в кілька разів у порівнянні з функцією  $\tanh(x)$ . Чотиришарова згорткова нейронна мережа з *ReLU* досягає помилки 25% в шість разів швидше, ніж така ж мережа з функцією  $\tanh(x)$ .

Архітектура моделі зображена на рис. 2.1. Мережа містить 8 шарів з вагами; перші п'ять шарів згорткові, а останні три шари повнозв'язні. Останній шар міститься 1000 виходів, який визначає мітку класу. Фільтри другого, четвертого і п'ятого згорткового шару з'єднані з картами ознак тільки попереднього шару. Фільтр третього згорткового шару з'єднаний з усіма картами ознак другого шару. Нейрони повнозв'язних шарів пов'язані з усіма нейронами попередніх шарів. Шари нормалізації слідує за першим і другим згортковим шаром. Шари субдискретизації слідує за шарами нормалізації і п'ятим згортковим шаром. Функція активації *ReLU* застосовується в кожному згортковому і повнозв'язному шарі.

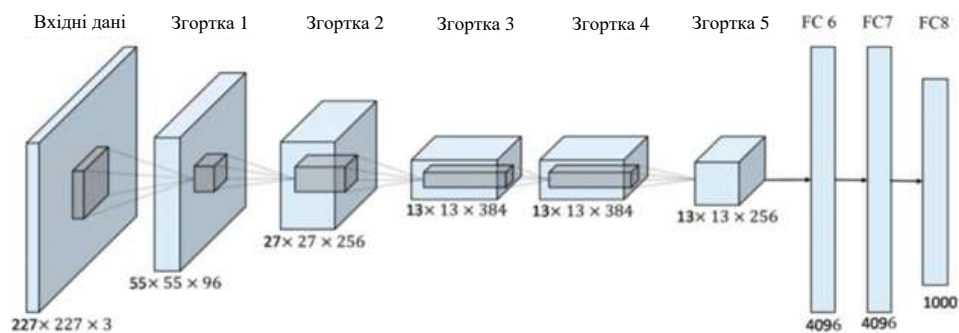


Рисунок 2.1 - Архітектура мережі AlexNet

Перший згортковий шар обробляє вхідне зображення  $227 \times 227 \times 3$  96-ма фільтрами розміром  $11 \times 11 \times 3$  з кроком 4 пікселі (це відстань між центрами областей сприйняття сусідніх нейронів на карті ознак). Другий згортковий шар приймає на вхід вихід першого шару і обробляє фільтрами розміром  $5 \times 5 \times 48$ . Третій, четвертий і п'ятий згорткові шари з'єднані з іншими без шарів субдискретизації і нормалізації. Третій згортковий шар має 384 фільтра розміром  $3 \times 3 \times 256$ , з'єднаних з нормалізованим і субдискретизованим виходом

другого шару. Четвертий шар має 384 фільтра розміром  $3 \times 3 \times 192$ , п'ятий шар має 256 фільтрів розміром  $3 \times 3 \times 192$ . Повнозв'язні шари мають 4096 нейронів кожен. Для зменшення перенавчання в AlexNet запропоновані наступні методи:

- Збільшення даних - найпростіший і часто використовуваний метод зменшення перенавчання. Метод полягає в додатковому перетворенні тренувальних зображень для збільшення кількості вхідних зображень. Використовується дзеркальне відображення зображення по горизонталі і зміни інтенсивностей кольору пікселів.

- Дропаут - установка в нуль значення кожного нейрона в прихованому шарі з ймовірністю 0,5. Нейрони, до яких застосовано дропаут, не беруть участі в зворотному поширенні помилки.

Тому, кожен раз, коли на вхід надходить зображення, нейронна мережа має різну архітектуру. Дропаут еквівалентний усередненню всіх моделей, які виходили на кожному кроці випадковим викиданням окремих нейронів.

Після успіху AlexNet в 2012 році, під час проведення ILSVRC 2012 Workshop широко обговорювалася можливість узагальнення результатів згорткових нейронних мереж для задачі розпізнавання об'єктів в PASCAL VOC Challenge. У моделі R-CNN показано, що згорткові нейронні мережі значно перевершують за якістю ідентифікації об'єктів в PASCAL VOC в порівнянні з методами, заснованих на гістограмах орієнтованих градієнтів [9].

Для досягнення цього результату було вирішено два завдання: локалізація об'єктів за допомогою глибокої мережі і тренування моделі тільки на невеликій кількості анотованих даних. Завдання ідентифікації вимагає локалізації об'єкта (або декількох об'єктів) на зображенні. Ранні моделі згорткових нейронних мереж вирішували це завдання методом ковзного вікна. Ці мережі мали, зазвичай, тільки 2 згорткових шари і шар субдискретизації. В R-CNN запропоновано використання замість методу ковзного вікна, «розпізнавання всередині області». Цей метод створює близько 2000 незалежних областей для вхідного зображення, витягує вектор ознак фіксованої довжини з кожної області, використовуючи згорткову нейронну мережу, і потім класифікує кожну

область за допомогою класифікатора SVM. На рис. 2.2 зображений алгоритм R-CNN. З вхідного зображення (1) витягується близько 2000 областей (2). У кожній області обчислюються вектори ознак фіксованої довжини за допомогою згорткової нейронної мережі (3). Отримані ознаки класифікуються за допомогою класифікатора SVM (4)

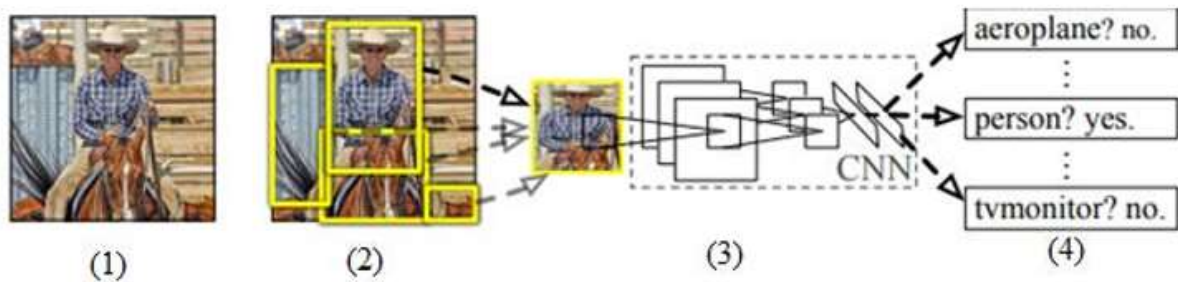


Рисунок 2.2 - Архітектура R-CNN

В роботі [9] показано, що R-CNN дає 30% приріст у порівнянні з найкращим результатом PASCAL VOC 2012. Недоліки моделі R-CNN:

- Навчання займає кілька стадій. R-CNN спочатку навчається за допомогою обчислення функції втрат. Потім навчається класифікатор ознак SVM. На третьому етапі навчання навчаються регресори прямокутників, що обрамляють, - для коригування області передбаченого об'єкта з істинної областю.

- Навчання вимагає багато часу і ресурсів. Для класифікатора SVM ознаки витягуються з кожної області зображення і записуються на жорсткий диск. Для дуже глибоких мереж, таких як VGG16, цей процес займає 2,5 дні на GPU для 5000 зображень. Ці ознаки вимагають сотні гігабайт для зберігання на диску.

- Низька швидкість ідентифікації об'єктів. Протягом обчислення, ознаки витягуються з кожної області для кожного об'єкта. Ідентифікації займає ~ 47 секунд на зображення з використанням GPU.

У моделі Fast R-CNN запропоновані методи поліпшення мережі R-CNN [10]. У Fast R-CNN запропоновано новий алгоритм навчання, який виправляє

недоліки R-CNN, підвищуючи його швидкість і точність. Алгоритм Fast R-CNN зображений на рис. 2.3. Вхідне зображення і область множинного інтересу (ROI) надходять на вхід згорткової мережі. Область множинного інтересу - це шар, що виконує субдискретизацію вхідного зображення і створює карту ознак фіксованого розміру  $7 \times 7$ . Основна мета цього шару - прискорення часу навчання і тестування.

Кожна область ROI об'єднується в карту ознак фіксованого розміру, а потім, в вектор ознак за допомогою повнозв'язних шарів (fully-connected layer, FC). Кожен вектор ознак надходить на послідовність повнозв'язних шарів, які в кінці поділяються на два вихідних шари.

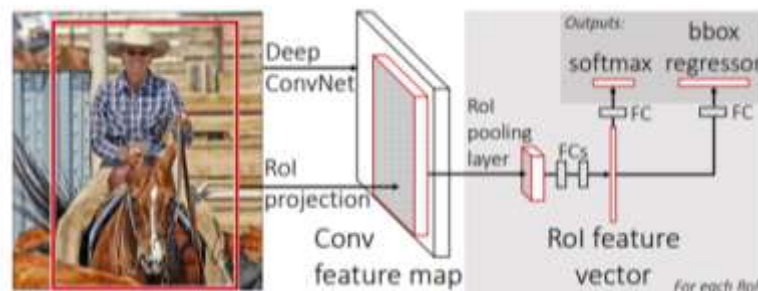


Рисунок 2.3 Алгоритм Fast R-CNN.

Fast R-CNN має такі переваги:

- Якість ідентифікації вище, ніж R-CNN.
- Навчання проходить в один етап.
- При навчанні оновлюються ваги всіх шарів мережі.
- Не потрібно простір жорсткого диска для зберігання ознак.

У попередніх моделях (R-CNN, Fast R-CNN) використовувалися методи пропозиції областей для первинного створення обмежувального прямокутника для об'єктів, потім ці області класифікувалися. Після класифікації, використовувалася постобробка обмежувальних прямокутників для виключення дублікатів ідентифікації. Такі методи повільні і важко оптимізуються, тому що кожен компонент необхідно було навчати незалежно.



У моделі YOLO запропоновано визначити ідентифікації об'єктів як одне регресійне завдання - прямо від пікселів зображення до координат обмежувальних прямокутників і ймовірностей класів для кожного об'єкта [11]. Алгоритм моделі YOLO зображений на рис. 2.4

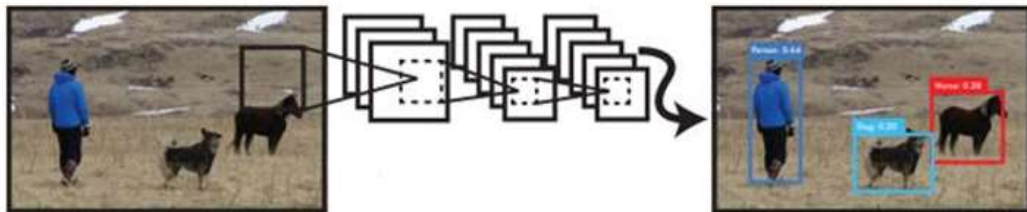


Рисунок 2.4 Алгоритм YOLO

Алгоритм детектування виглядає наступним чином:

1. Вхідне зображення перетворюється до розміру 448x448.
2. Запускається згорткова мережа для ідентифікації об'єктів.
3. Обчислюється результат ідентифікації як значення довіри.

Одна згорткова мережа одночасно передбачає безліч обмежувальних прямокутників і ймовірності класів для них. Алгоритм ідентифікації об'єктів в YOLO зображений на рисунку 2.5 і виглядає наступним чином:

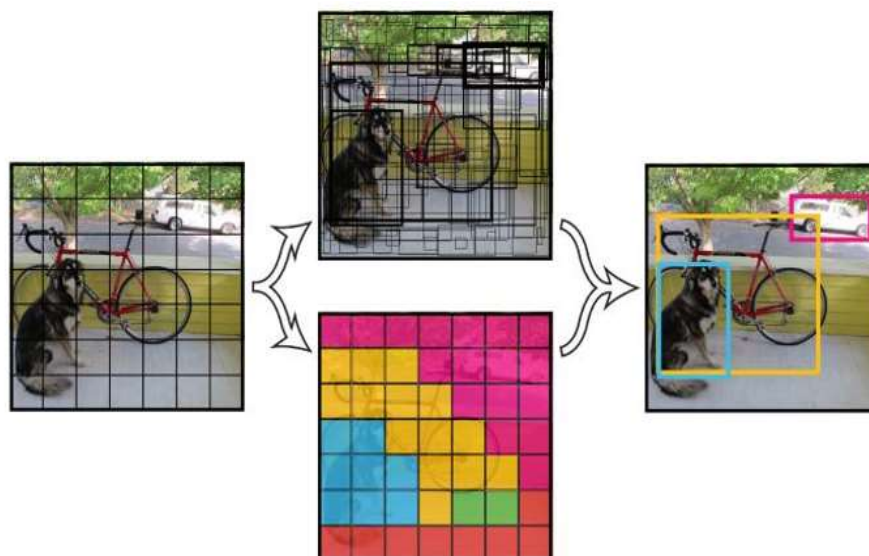


Рисунок 2.5 - Алгоритм ідентифікації об'єктів в YOLO

Модель ділить вхідне зображення сіткою  $S \times S$ . Якщо центр об'єкта потрапляє в осередок сітки, цей осередок бере участь у подальшому виявленні об'єкта. У кожній такій клітинці передбачається  $B$  обмежувальних прямокутників і обчислюється їх ймовірність. Ця ймовірність відображає, наскільки ймовірно, що цей обмежувальний прямокутник містить об'єкт. Далі визначається ймовірність передбачення прямокутника. Формально, ймовірність визначається як:

$$P = P(Obj) \cdot IOU(truth, pred)$$

де  $P(Obj)$  - ймовірність того, що обмежувальний прямокутник містить об'єкт, а  $IOU(truth, pred)$  - відношення області перекриття передбаченого і істинного обмежувальних прямокутників до області об'єднання цих прямокутників, що визначається як:

$$IOU(truth, pred) = \frac{truth \cap pred}{truth \cup pred}$$

Кожен обмежувальний прямокутник містить 5 передбачень: координат  $x$ ,  $y$ ,  $w$ ,  $h$  і ймовірність  $P$ . Координати  $(x, y)$  визначають центр області щодо кордонів осередку сітки. Координати  $(w, h)$  визначають розмір прямокутника щодо всього зображення. У кожному осередку сітки також передбачається умовна ймовірність класу  $C = P(Class_i | Obj)$ . Під час тестування множаться ймовірності класів і окремі області передбачення об'єктів:

$$C \cdot P = P(Class_i | Obj) P(Obj) \cdot IOU(truth, pred) = P(Class_i) \cdot IOU(truth, pred),$$

що дає оцінку ймовірності кожного класу для кожної області. Це передбачення кодується тензором розміру  $S \cdot S \cdot (B \cdot 5 + C)$ .

Архітектура згорткової мережі в моделі YOLO зображена на рисунку 2.6.

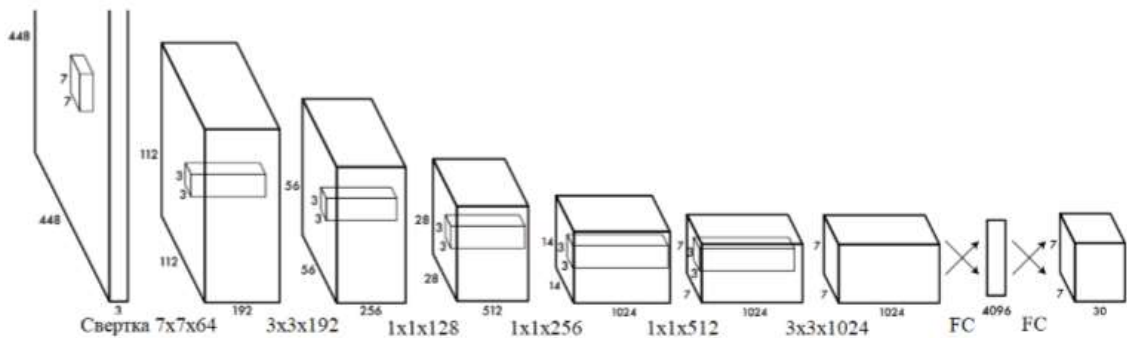


Рисунок 2.6 - Архітектура згорткової мережі в YOLO

Модель YOLO має такі переваги:

- Висока швидкість ідентифікації. Мережа має швидкодію 45 кадрів в секунду, а її швидка версія (Fast YOLO) більш ніж 150 кадрів в секунду.
- YOLO робить прогноз по всьому зображенню, на відміну від методів ковзного вікна і методів, заснованих на областях пропозиції. Це дозволяє моделі кодувати контекстуальну інформацію про класи. Завдяки цьому YOLO робить в два рази менше помилок на тлі зображення в порівнянні з R-CNN, так як модель R-CNN може робити помилки на тлі зображення, тому що не може побачити більший контекст.
- Завдяки тому, що YOLO вивчає узагальнені подання об'єктів, при навчанні на природних зображеннях і тестуванні на різних творах мистецтва, YOLO набагато перевершила найкращі методи ідентифікації (R-CNN, Fast R-CNN).

Подальший розвиток ідеї YOLO отримали в моделі SSD, що використовує такий же принцип ідентифікації об'єктів відразу по всьому зображенню, що відображено в її назві - Single Shot Detector [12]. Модель SSD - перша глибока мережа для ідентифікації об'єктів, що не використовує ознаки, які складаються з обмежувальних прямокутників. Це призводить до значного збільшення продуктивності і ідентифікації об'єктів з більш високою точністю. Основне поліпшення швидкості відбувається при відмові від використання

обмежувальних прямокутників. Також поліпшення точності відбувається шляхом використання невеликого згорткового фільтра для передбачення класу об'єкта і згорткового фільтра для корекції положення обмежувальних прямокутників з різним співвідношенням сторін і розміром. Ці згорткові фільтри застосовуються до кількох карт ознак, як на ранніх, так і на більш пізніх шарах мережі, що дозволяє виявляти об'єкти різного розміру.

На рис. 2.7 зображено порівняння архітектур YOLO і SSD. У моделі SSD додаються додаткові шари, що дозволяють виявляти об'єкти різного розміру.

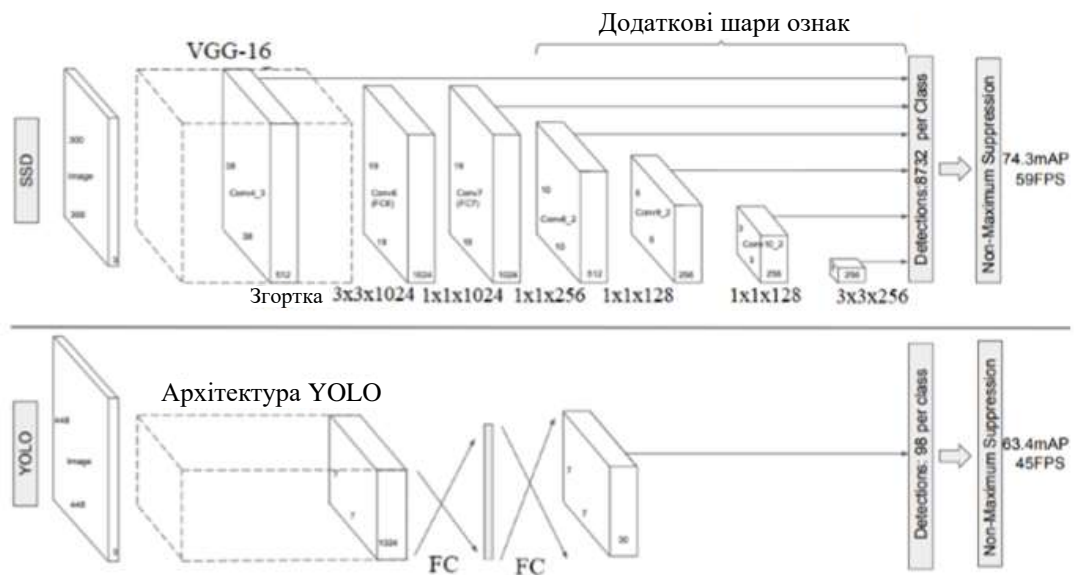


Рисунок 2.7 - Порівняння архітектури SSD (зверху) і YOLO (знизу)

При порівнянні моделей можна отримати однозначної відповіді, яка модель є кращою. Для застосування в реальному житті, необхідний баланс між точністю та швидкістю. Крім того, необхідно врахувати й інші фактори, які вносять вклад в якість моделі:

- Модель вилучення ознак (VGG, ResNet, Inception, MobileNet).
- Роздільна здатність вхідного зображення.
- Як пророкування виключається з розрахунку функції втрат.
- Число передбачень в моделі.
- Кодування обмежувальних прямокутників.

- Розширення (аугментація) наборів даних.
- Набір даних, що використовується для навчання.
- Використання багатомасштабних зображень в навчанні або тестуванні.
- Локалізація функції втрат.
- Робота з програмним забезпеченням для глибокого навчання.
- Параметри навчання, такі як розмір батча (кількість даних для навчання на одному кроці), крок навчання, падіння швидкості навчання.

Також на адекватність порівняння впливає те, що технології розвиваються швидко, і будь-яке порівняння швидко застаріває. В цьому розділі об'єднані результати з окремих робіт. Результати PASCAL VOC 2012 наведені в таблиці 2.1 [13]. Використовується середня точність по всіх класах ( $mAP$ ) як критерій якості, найкращі результати у моделей SSD512 ( $mAP = 74,9$ ) і SSD300 ( $mAP = 72,4$ ).

Таблиця 2.1 - Результати порівняння моделей по  $mAP$  і  $AP$  по класу «person»

Модель	$mAP$	person
Fast R-CNN	68,4	72,0
YOLO	67,9	63,5
SSD300	72,4	77,4
SSD512	74,9	78,3

На рис. 2.8 показаний порівняльний результат точності моделей, які навчалися на наборі PASCAL VOC 2007 і PASCAL VOC 2012 [13]. Значення вимірювалися на перевірочному наборі даних PASCAL VOC 2012. Для SSD, показані результати для вхідного зображення  $300 \times 300$  і  $512 \times 512$ . Для YOLO показані результати для вхідного зображення  $288 \times 288$ ,  $416 \times 416$  і  $514 \times 514$ . Більш високий дозвіл вхідного зображення для моделі забезпечує більш високе значення  $mAP$ , але меншу продуктивність.

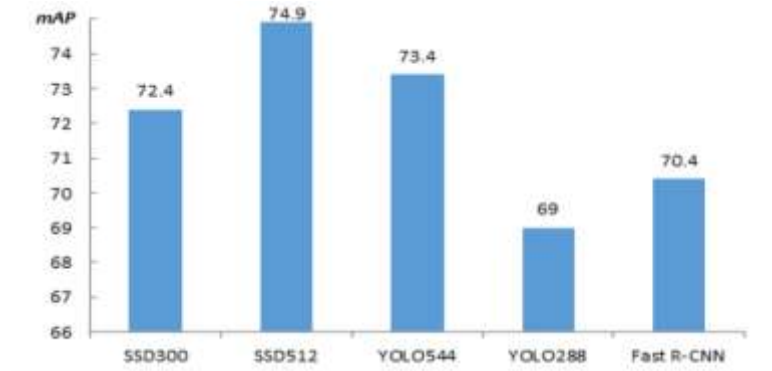


Рисунок 2.8 – Результати точності в PASCAL VOC 2007 і PASCAL VOC 2012

На рисунку 2.9 показаний порівняльний результат продуктивності, що вимірюється в FPS (кадрів в секунду) [13]. Навчальні та перевірочні набори даних аналогічні попереднім.

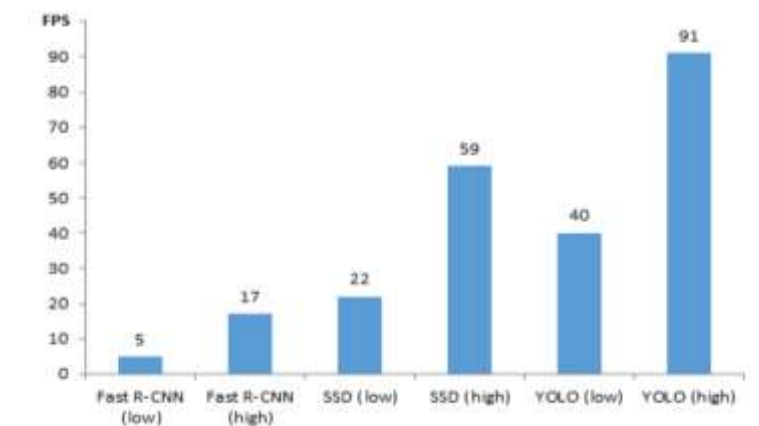


Рисунок 2.9 - Порівняння швидкодії моделей

За наведеними вище результатами видно, що найбільшу продуктивність забезпечує модель YOLO, але вона має точність нижче в порівнянні з іншими моделями. Високу точність має модель R-CNN і її модифікації, але вона має дуже низьку продуктивність. Модель SSD має оптимальне співвідношення точності і продуктивності, тому ця модель взята за основу для розробки моделі ідентифікації пішоходів у відеопотоці.

Для того щоб розробити власну модель на основі згорткових нейронних мереж, необхідно розглянути принцип роботи та навчання згорткових нейронних мереж.

### 2.3 Аналіз математичної моделі згорткової нейронної мережі

Згорткові нейронні мережі (convolutional neural networks) - широкий клас архітектур, основна ідея яких полягає в тому, щоб перевикористати одні і ті ж частини нейронної мережі для роботи з різними локальними ділянками входів. Основне застосування згорткових мереж - обробка зображень [15].

Основна ідея згорткової мережі полягає в тому, що обробка ділянки зображення часто повинна відбуватися незалежно від конкретного розташування цієї ділянки. Розпізнавання об'єкту на зображенні - локально і незалежно від конкретного положення ділянки з об'єктом всередині всього зображення. Для цього, вхід покривається невеликими вікнами (наприклад,  $5 \times 5$  пікселів) і виділяються ознаки в кожному такому вікні невеликою нейронною мережею, причому ознаки будуть виділятися одні й ті ж, тобто маленька нейронна мережа буде всього одна, у неї буде  $5 \times 5 = 25$  входів, а з кожного зображення буде багато виходів.

Потім результати цієї нейронної мережі знову можна буде подати у вигляді зображення, замінюючи вікна  $5 \times 5$  на їх же центральні пікселі, і на ньому можна застосовувати другий згортковий шар, з уже іншою нейронною мережею.

Зазвичай зображення, що подаються на вхід, представлені у вигляді декількох прямокутних матриць, кожна з яких задає рівень одного з колірних каналів в кожному пікселі зображення. Зображення розміром  $200 \times 200$  пікселів - це 120000 чисел, три матриці інтенсивностей розміром  $200 \times 200$  кожна. Якщо зображення чорно-біле, то така матриця одна, якщо зображення багатоспектральне, то таких матриць може бути багато. Передбачається, що в кожному пікселі вхідного зображення стоїть деякий тензор (вектор чисел), а його компоненти називаються каналами (channels).

Такі ж матриці будуть виходити і після згорткового шару, в них як і раніше буде просторова структура, відповідна вихідній картинці, але каналів

може стати більше. Значення ознак, виділених вікнами з вхідного зображення, утворюють матрицю - карту ознак (feature map).

Згортка - лінійне перетворення вхідних даних. Якщо  $x^l$  - карта ознак в шарі під номером  $L$ , то результат двовимірної згортки з ядром розміру  $2d+1$  і матрицею ваг  $W$  розміру  $(2d+1) \times (2d+1)$  на наступному шарі буде наступним

$$y_{i,j}^l = \sum_{-d \leq a, b \leq d} W_{a,b} x_{i+a, j+b}^l$$

де  $y_{i,j}^l$  - результат згортки на рівні  $l$ , а  $x_{i,j}^l$  її вхід, тобто вихід попереднього шару. Тобто, щоб отримати компоненту  $(i, j)$  наступного рівня, застосовується лінійне перетворення до квадратного вікна попереднього рівня, тобто скалярно множаться пікселі з вікна на вектор згортки, що зображено на рисунку 2.10. Застосовується згортка з матрицею ваг  $W$  розміру  $3 \times 3$  до матриці розміру  $5 \times 5$ . Приклад обчислення згортки зображений на рисунку 2.10.

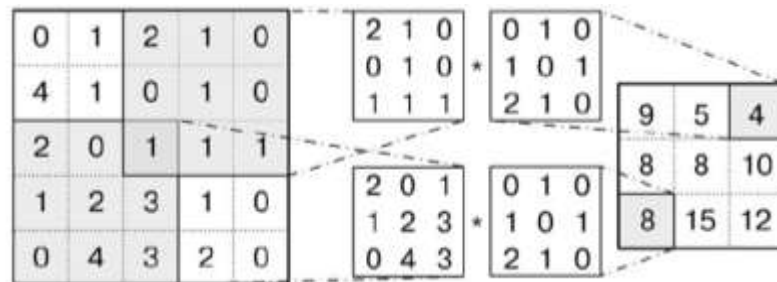


Рисунок 2.10 — Приклад обчислення результату згортки

Згортка має такі властивості:

- Згортка зберігає структуру входу (взаємне розташування пікселів), так як застосовується до кожної ділянки вхідних даних окремо.
- Операція згортки має властивість розрідженості, так як значення кожного нейрона чергового шару залежить тільки від невеликої частки вхідних нейронів.



• Згортка багаторазово перевикористовує одні і ті ж ваги, так як вони повторно застосовуються до різних ділянок входу.

У більшості випадків за згорткою слідує нелінійність, яка записується як

$$z_{i,j}^l = h(y_{i,j}^l)$$

В якості опції  $h$  зазвичай використовується функція *ReLU*.

У класичному шарі, крім лінійної згортки і наступної за нею нелінійності, є ще операція субдискретизації (pooling). Ідея субдискретизації полягає в тому, що наявність або відсутність ознаки важливіше, ніж її точні координати, тому можна узагальнити виділені ознаки, втративши частину інформації про їх місцезнаходження, але скоротивши розмірність. В якості операції субдискретизації застосовується операція взяття максимуму (max-pooling)

$$x_{i,j}^{l+1} = \max(z_{i+a,j+b}^l), a \in [-d, d], b \in [-d, d]$$

де  $d$  - розмір вікна субдискретизації, для  $d = 2$  приклад субдискретизації зображений на рисунку 2.11.

0	1	2	1
4	1	0	1
2	0	1	1
1	2	3	1

4	2	2
4	1	1
2	3	3

4	2
2	3

Рисунок 2.11 - Приклад субдискретизації з вікном розміру  $2 \times 2$ :  $a$  - вихідна матриця;  $b$  - матриця після субдискретизації з кроком 1,  $v$  - матриця після субдискретизації з кроком 2.

Стандартний шар згорткової нейронної мережі складається з трьох компонентів:

- Згортка у вигляді лінійного відображення, що виділяє локальні ознаки,
- Нелінійна функція, застосована покомпонентно до результатів згортки,
- Субдискретизація, що скорочує геометричний розмір вихідних тензорів.

На рис. 2.12 зображений такий шар. У порівнянні з вхідним зображенням, на виході розмірність тензора збільшилася - згорткову мережу зазвичай навчає кілька карт ознак на кожному шарі.

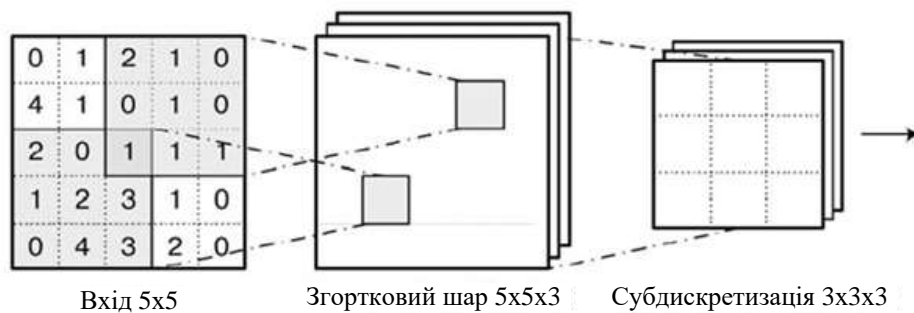


Рисунок 2.12 - Схема одного шару згорткової мережі

Для навчання карт ознак, необхідно здійснити поширення градієнта в зворотну сторону через згортковий шар. Оптимізується якась функція помилки  $E$ , і відомо її значення на виході. Через функцію взяття максимуму помилка проходить без зміни, так як шар субдискретизації нічого не навчає. Однак він робить проходячі по графу обчислення градієнти розрідженими, так як з усіх елементів вікна субдискретизації  $z_{i,j}^l$  приватна похідна  $\frac{\partial E}{\partial x_{i,j}^{l+1}}$  відноситься тільки до максимального, а решта отримують нульовий градієнт, і на даному тренувальному прикладі їх навчання можна вважати закінченим. Пропустивши градієнт через нелінійність, виходить

$$\frac{\partial E}{\partial y_{i,j}^l} = \frac{\partial E}{\partial z_{i,j}^l} \frac{\partial z_{i,j}^l}{\partial y_{i,j}^l} = \frac{\partial E}{\partial z_{i,j}^l} h'(y_{i,j}^l),$$

де  $\frac{\partial E}{\partial z_{i,j}^l}$  вже відомо, а  $h'(y_{i,j}^l)$  можна обчислити.

На згортковому рівні з'являються ваги, які необхідно навчити:

$$\frac{\partial E}{\partial w_{a,b}^l} = \sum_i \sum_j \frac{\partial E}{\partial y_{i,j}^l} \frac{\partial y_{i,j}^l}{\partial w_{a,b}^l} = \sum_i \sum_j \frac{\partial E}{\partial z_{i+a,j+b}^{l-1}},$$

де індекси  $i$  і  $j$  проходять всі елементи зображення на проміжному шарі  $y_{i,j}^l$ , тобто після згортки, але до субдискретизації.

Пропустивши градієнти на попередній шар, виходить:

$$\frac{\partial E}{\partial x_{i,j}^l} = \sum_a \sum_b \frac{\partial E}{\partial y_{i-a,j-b}^l} \frac{\partial y_{i-a,j-b}^l}{\partial x_{i,j}^l} = \sum_j \sum_i \frac{\partial E}{\partial y_{i-a,j-b}^l} w_{a,b}.$$

Таким чином, процедура зворотного поширення помилки адаптована для згорткового шару.

Глибока мережа будується шляхом подачі виходу згорткового шару на вхід наступного. Розмір шару за рахунок субдискретизації буде поступово скорочуватися, що дозволяє шарам, що знаходяться в кінці мережі, розпізнавати більш загальні ознаки на зображенні.

## 2.4 Навчання згорткових нейронних мереж

Основне завдання в навчанні нейронних мереж - мінімізація заданої функції помилки [14]. Зазвичай, оптимізується апостеріорна імовірність.

$$p(\theta|D) = p(\theta) \prod_{d \in D} p(d|\theta),$$

де функція правдоподібності  $p(\theta)$  - помилка на навчальній вибірці, апіорний розподіл  $p(d | \theta)$  — регуляризація.

Завдання оптимізації - по заданій функції знайти аргументи, в яких ця функція мінімізується. Функція помилки в нейронних мережах має багато локальних екстремумів. Для того, щоб знайти оптимальний екстремум, використовується евристичний метод оптимізації - градієнтний спуск.

Представивши поверхню функції помилки, завдання оптимізації зводиться до обчислення градієнта. Якщо функцію, що визначає поверхню, позначити через

$$E(\boldsymbol{\theta})=E(\theta_1, \theta_2, \dots, \theta_n),$$

де  $(\theta_1, \theta_2, \dots, \theta_n)$  – параметри функції, то її градієнт  $\nabla E$  – це вектор похідних функції декількох змінних по кожній з компонент

$$\nabla_{\boldsymbol{\theta}} E = \begin{pmatrix} \frac{\partial E}{\partial \theta_1} \\ \vdots \\ \frac{\partial E}{\partial \theta_n} \end{pmatrix}.$$

Градієнт визначає напрямок, в якому функція зростає швидше за все, отже, напрямок, в якому вона швидше за все спадає – це напрямок, зворотній градієнту:  $-\nabla_{\boldsymbol{\theta}} E$ . Позначаючи через  $\boldsymbol{\theta}_t$  вектор параметрів моделі на кроці  $t$ , а через  $E$  - мінімізуєму функцію, можна визначити вектор поновлення параметрів на етапі  $t$  як

$$\begin{aligned} \mathbf{u}_t &= -\eta \nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}_{t-1}), \\ \boldsymbol{\theta}_t &= \boldsymbol{\theta}_{t-1} + \mathbf{u}_t, \end{aligned}$$

де  $\eta$  – швидкість навчання. Швидкість навчання регулює крок поновлення параметрів.

У стандартному методі градієнтного спуску швидкість навчання задається вручну, що може привести до наступних проблем:

– Якщо крок занадто маленький, то навчання буде занадто довгим і підвищується ймовірність потрапити в невдалий локальний мінімум функції (див. Рисунок 2.13(а)),

– Якщо крок занадто великий, то можна пропустити необхідний локальний мінімум (див. рисунок 2.13(б)).

Для вирішення цих проблем використовується стохастичний градієнтний спуск - підрахунок помилки і оновлення ваг відбувається не після проходження всієї тренувальної множини, а після кожного прикладу:

$$\theta_t = \theta_{t-1} - \eta \nabla E(f(\mathbf{x}_t, \theta_{t-1}), y_t)$$



Рисунок 2.13 - Проблеми зі швидкістю градієнтного спуску:

а) — занадто маленькі кроки,

б) — занадто великі кроки.

Основна перевага стохастичного градієнтного спуску - висока швидкість обчислення помилки, так як оновлення ваг відбувається після кожного кроку, що збільшує швидкість навчання. Також стохастичний градієнтний спуск

працює більш випадково, що дозволяє знайти більш вдалий локальний мінімум функції помилки.

На практиці використовується стохастичний градієнтний спуск по міні-батчам, невеликим підмножинам тренувального набору. Це дозволяє зберегти плюси стохастичного градієнтного спуску, і при цьому користуватися процедурами матричної арифметики, які обчислюються швидко на пристроях, що підтримують паралельні обчислення.

При оптимізації великої нейронної мережі, функція помилки - складна композиція інших функцій. Представивши складну функцію як композицію простіших, можна обчислити її похідну, що потрібно для здійснення градієнтного спуску.

Функція представляється у вигляді графа обчислень. Граф обчислень - це граф, вузлами якого є функції, а ребра пов'язують функції зі своїми аргументами. На рис. 2.14 зображений приклад графа обчислень для функції

$$f(x, y) = x^2 + xy + (x + y)^2.$$

На рисунку 2.14а - з використанням функцій  $\times$ ,  $+$  і  $x^2$ , б - з використанням функцій  $\times$  і  $+$ , в - у вигляді дерева з використанням функцій  $\times$  і  $+$ .

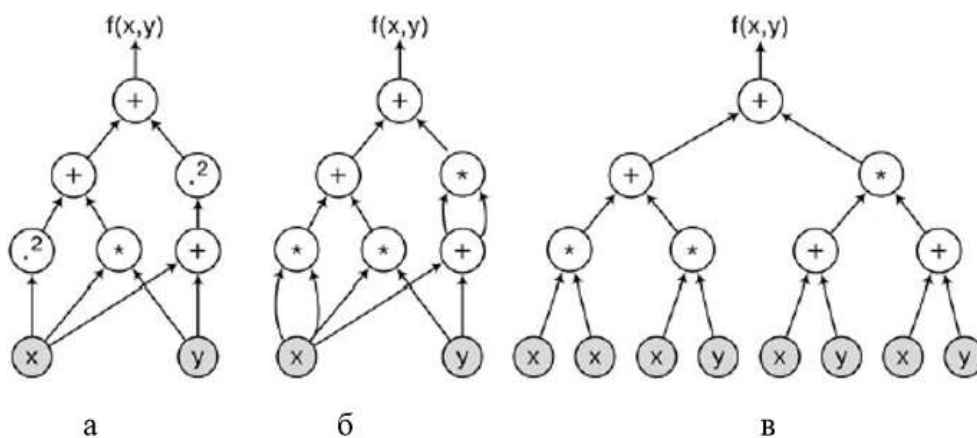


Рисунок 2.14 - Граф обчислень для функції  $f(x, y) = x^2 + xy + (x + y)^2$ .

У нейронних мережах в якості елементарних функцій використовуються функції, з яких отримуються нейрони. Наприклад, скалярного добутку векторів  $x \top y$  і логістичного сигмоїда  $\sigma$  достатньо для того, щоб побудувати будь-яку мережу, складену з нейронів з функцією активації  $\sigma$ .

Таким чином, представивши складну функцію в якості простих, можна обчислити похідну складної функції, обчисливши похідні її складових:

$$(f \circ g)'(x) = (f(g(x)))' = f'(g(x))g'(x)$$

Якщо  $x$  - вектор  $x=(x_1, \dots, x_n)$ , то замість часткової похідної обчислюється градієнт

$$\nabla_x f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}.$$

Застосовуючи метод для кожного компонента окремо, в результаті виходить

$$\nabla_x (f \cdot g) = \begin{pmatrix} \frac{\partial f \cdot g}{\partial x_1} \\ \vdots \\ \frac{\partial f \cdot g}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial g} \frac{\partial g}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial g} \frac{\partial g}{\partial x_n} \end{pmatrix} = \frac{\partial f}{\partial g} \nabla_x g.$$

Якщо  $g$ - вектор, то виходить

$$\nabla_x f = \frac{\partial f}{\partial g_1} \nabla_x g_1 + \dots + \frac{\partial f}{\partial g_k} \nabla_x g_k = \sum_{i=1}^k \frac{\partial f}{\partial g_i} \nabla_x g_i$$

В результаті, виходить формула матричного множення

$$\nabla_x f = \nabla_x g \nabla_g f,$$

$$\nabla_x g = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \dots & \frac{\partial g_k}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial x_n} & \dots & \frac{\partial g_k}{\partial x_n} \end{pmatrix}.$$

Таким чином, можна обчислити похідні і градієнти будь-якої композиції функцій, в тому числі і векторних.

Таким чином, можна застосовувати формулу диференціювання композиції на графі або від витоків до стоків, отримуючи часткові похідні кожного вузла по одній і тій же змінній  $\frac{dx}{dx}$ ,  $\frac{da}{dx}$ , ... ,  $\frac{df}{dx}$  або від стоків до витоків, отримуючи часткові похідні по всіх проміжних вузлах  $\frac{df}{df}$ ,  $\frac{df}{de}$ , ... ,  $\frac{df}{dx}$ . Дійшовши до витоків графа - вершин  $x_1, \dots, x_n$ , виходять часткові похідні  $\frac{df}{dx_1}, \dots, \frac{df}{dx_n}$ , тобто градієнт  $\nabla_x f$ .

Так як часткові похідні рахуються в напрямку, протилежному графу обчислень, такий алгоритм називається алгоритмом зворотного поширення (backpropagation).

## 2.5 Структура інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі

Структура інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі, детально описана у попередніх підпунктах, зображена на рис. 2.15.

В основі інформаційної технології лежить згорткова нейронна мережа. А оскільки кожна нейронна мережа має два режими: навчання та функціонування, то структура інформаційної технології має дві частини. Ліва частина структури на рис. 2.15 відображує навчання згорткової нейронної мережі, а права частина



структури на рис. 2.15 відображує функціонування. Для роботи інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі в режимі навчання вхідною є інформація у вигляді набору даних Caltech Pedestrian Dataset для навчання і тестування, а також набори даних PASCAL VOC і MS COCO для попередньої ініціалізації ваг мережі шляхом перенесення навчання. В цій частині спочатку відбувається процес створення (генерації) згорткової нейронної мережі згідно обраної архітектури SSD. Потім здійснюється процес початкової ініціалізація ваг мережі попереднім навчанням на великих наборах даних (PASCAL VOC і MS COCO) - transfer learning. А далі відбувається процес навчання нейронної мережі на основі набору даних Caltech Pedestrian Dataset. Після цього згорткова нейронна мережа готова до ідентифікації пішоходів у відеопотоці.

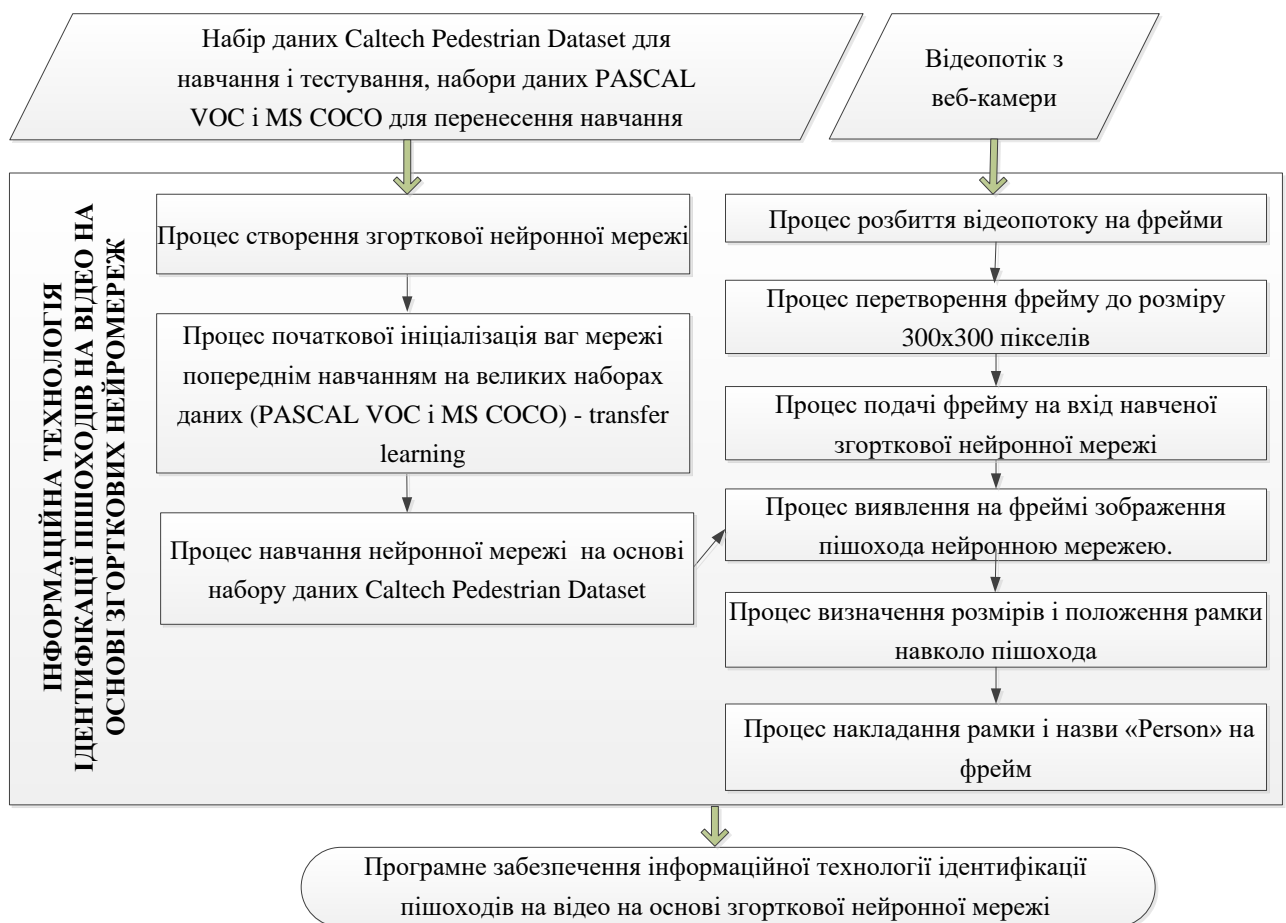


Рисунок 2.15 - Структура інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі

Для роботи інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі в режимі функціонування входною є інформація у вигляді відеопотоку з веб-камери. В цій частині спочатку відбувається процес розбиття відеопотоку на фрейми. Потім здійснюється процес перетворення фрейму до розміру 300x300 пікселів, оскільки згорткова нейронна мережа SSD на вхід сприймає зображення саме такого розміру. Далі відбувається процес подачі фрейму на вхід навченої згорткової нейронної мережі. Після цього здійснюється процес виявлення на фреймі зображення пішохода нейронною мережею. Далі визначаються розмірів і положення рамки навколо пішохода. І, нарешті, проходить процес накладання рамки і назви «Person» на фрейм.

Таким чином, розроблена структура інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі може бути використана для подальшої розробки програмних засобів.

## **2.6 Висновок**

В даному розділі було обгрунтовано вибір згорткової нейронної мережі для побудови інформаційної технології ідентифікації пішоходів у відеопотоці, обрано архітектуру згорткової нейронної мережі, проаналізовано математичну модель згорткової нейромережі та процес її навчання. За основу для розробки моделі ідентифікації пішоходів у відеопотоці взята модель згорткової нейромережі SSD, яка має оптимальне співвідношення точності і продуктивності. Для згорткових та агрегуювальних шарів буде використана функція активації – ReLU. Метод, за яким буде проходити навчання мережі – це метод зворотного поширення помилки. У підсумку було розроблено структуру інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІДЕНТИФІКАЦІЇ ПШОХОДІВ У ВІДЕОПОТОЦІ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ

### 3.1 Архітектура і алгоритм роботи SSD

Архітектура SSD представлена на рисунку 3.1. Вхідні шари мережі являють собою мережу стандартної архітектури, що використовується для класифікації зображення.

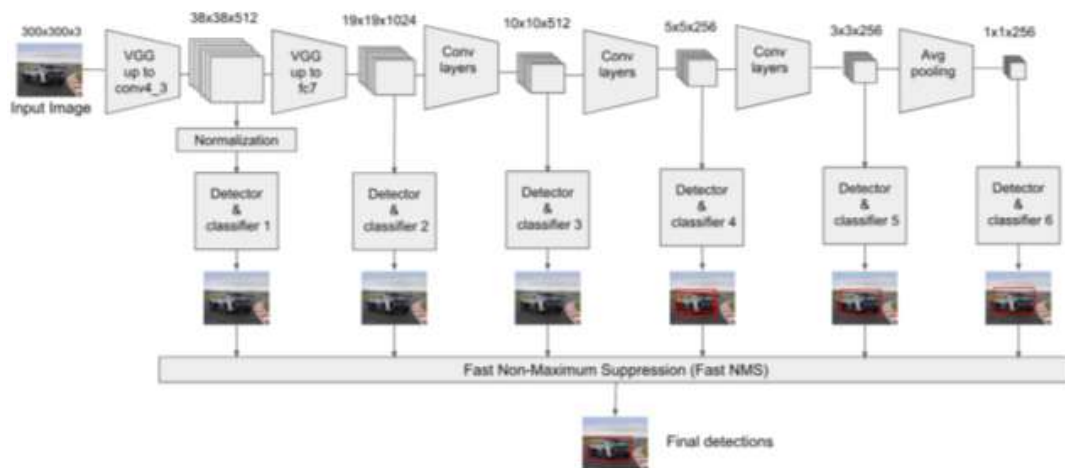


Рисунок 3.1 — Архітектура SSD

В нашій архітектурі застосовується мережа VGG. Основна ідея VGG - використання фільтрів  $3 \times 3$  з одиничним кроком згортки замість використовуваних раніше фільтрів  $7 \times 7$ . Переваги фільтрів розміру  $3 \times 3$ :

- Рецептивне поле трьох послідовних згорткових шарів розміром  $3 \times 3$  має розмір  $7 \times 7$ , але ваг у них буде 27 замість 49 в фільтрі  $7 \times 7$ . З цього випливає, що VGG може бути більш глибокою, тобто містити більше шарів, при цьому з меншою кількістю ваг;

- Наявність додаткової нелінійності між шарами дозволяє збільшити роздільну здатність в порівнянні мережі з одним шаром з більшою згорткою.

Архітектура мережі VGG зображена на рисунку 3.2, на якому показані особливості мережі VGG:

- Кілька згорток слідують одна за одною без шару дискретизації;
- Число карт ознак поступово зростає на більш глибоких рівнях мережі;
- В кінці мережі ознаки представлені у вигляді одновимірної ознаки, яку подають на вхід повнозв'язних шарів.

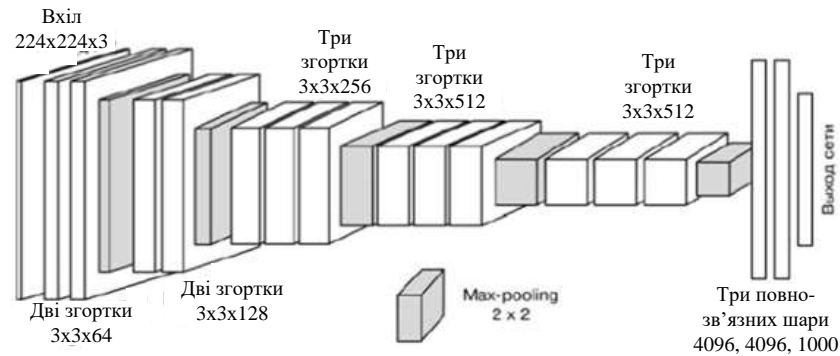


Рисунок 3.2 — Архітектура VGG-16

В архітектурі, яка розробляється, на вхід подається зображення  $300 \times 300$ , що збільшує здатність мережі розпізнавати невеликі об'єкти на зображенні. За набором згорткових шарів (які мають різну глибину в різних архітектурах) слідує три повнозв'язних шари: перші два мають 4096 каналів кожен, третій - 1000 каналів (за кількістю класів в наборі даних ILVRC). Останній шар - шар активації з функцією *softmax*.

Після VGG слідують блоки згорткових шарів, необхідних для отримання ознак. Шари на початку мережі витягають більш часткові ознаки, шари в кінці - більш загальні. Після кожного блоку вилучення ознак, карта ознак подається на блок ідентифікації і класифікації. На рисунку 3.3 зображений приклад алгоритму блоку ідентифікації і класифікації, що знаходиться після 4 згорткового шару.

На вхід блоку ідентифікації і класифікації подається проміжна карта ознак. На основі її обчислюються області для детектування, проводиться

корекція меж (localization) і обчислюється значення впевненості передбачення об'єкта в області детектування (confidence).

Далі відбувається корекція меж, класифікація і фільтрація, на виході блоку результат детектування об'єктів.

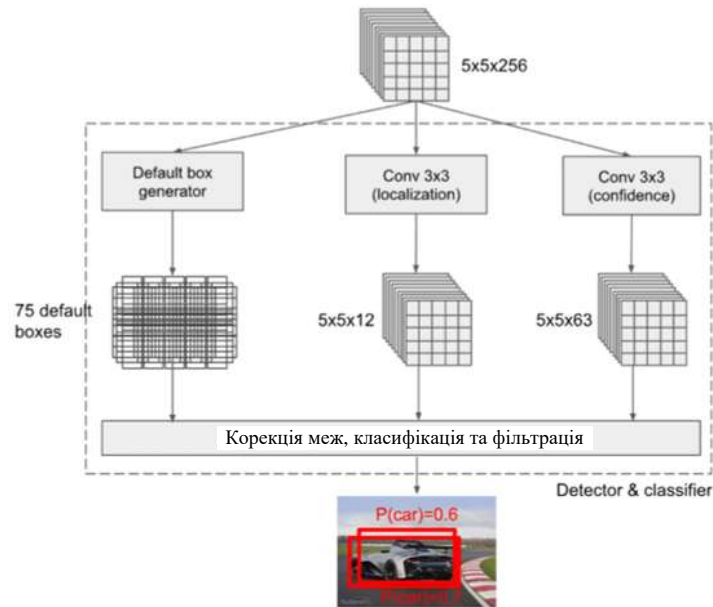


Рисунок 3.3 - Блок ідентифікації і класифікації об'єктів.

Розмір початкового зображення  $300 \times 300$ . Просторова розмірність карти ознак  $5 \times 5$ . У кожному осередку карти ознак будується 3 прямокутника з різними співвідношеннями сторін. На рисунку 3.4 точками показано розташування прямокутників.

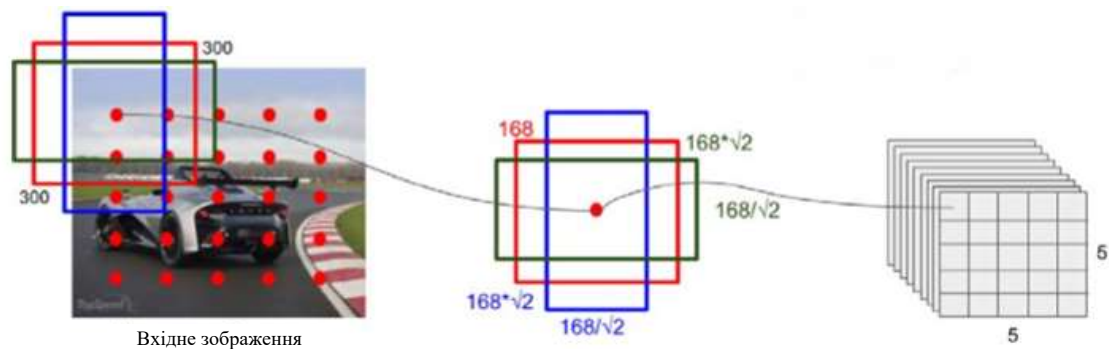


Рисунок 3.4 - Алгоритм обчислення областей детектування.

Прямокутник задається наступними величинами:

- $x_0$  - центр прямокутника по осі  $x$  ;
- $y_0$  - центр прямокутника по осі  $y$  ;
- $w$  - ширина прямокутника;
- $h$  - висота прямокутника.

Для кожного осередку карти ознак буде побудовано по 3 прямокутника, тому кількість прямокутників буде дорівнювати  $5 \times 5 \times 3 = 75$ .

Для кожної області детектування за допомогою згортки обчислюється вектор корекції ознак ( $dx, dy, dw, dh$ ). Також для кожної області за допомогою згортки визначаються впевненості для кожного класу, які подаються на блок немаксимального придушення (softmax), який обчислює ймовірність виявлення об'єкта в даній області.

### 3.2 Заходи з поліпшення моделі SSD

Завдання даної роботи - розпізнавання пішоходів у відеопотоці зображенні. Тому необхідно вибрати відповідний набір даних для навчання і тестування. Обрано набір даних Caltech Pedestrian Dataset [16]. Цей набір найбільш наближений до реального відеозображення, яке можна отримати з відеореєстратора автомобіля. Характеристики цього набору наведені в табл. 3.1.

Таблиця 3.1 — Характеристики набору даних Caltech

Число кадрів	~ 1000000
Розмічене число кадрів	~250000
Кадри, що містять об'єкт «пішохід»	~ 132000
Число обмежувальних прямокутників	~ 350000
Число обмежувальних прямокутників, що перекриваються	~ 126000
Число унікальних об'єктів «пішохід»	~ 2300
Середня кількість об'єктів на кадр	~ 1,4

Розмічено 250000 кадрів дозволом 640x480. Всього в наборі даних міститься 2300 пішоходів, виділених за допомогою 350000 обмежувальних прямокутників. Також для кожного кадру позначено, чи видно пішохід цілком або спостерігається оклюзія (перекриття обмежувальних прямокутників).

Розподіл пішоходів за розміром (висота в пікселях) показано на рисунку 3.5(a). Висота пішоходів розподілена нормально до медіанного значенням 48 пікселів. На рисунку 3.5(b) показано співвідношення висоти і ширини пішохода, який також розподілено нормально до медіанного значенням 0,41.

Розмір і форма областей детектування, що використовуються в розробленій моделі, буде підібрані далі відповідно до цього розподілу розмірів і форми об'єктів на наборі даних.

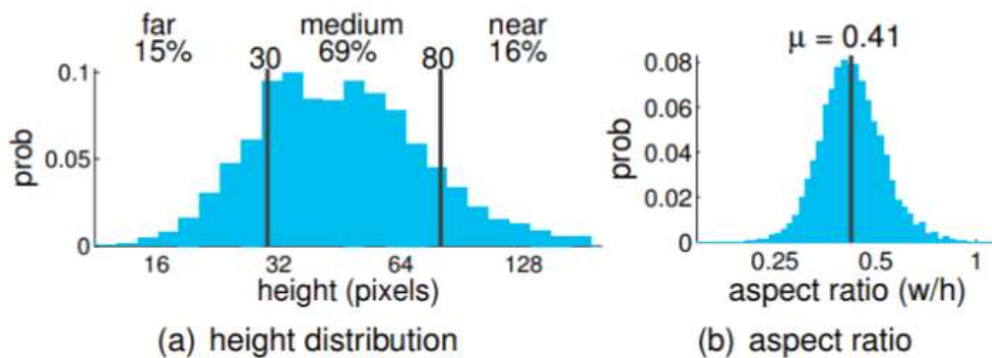


Рисунок 3.5 — Розподіл об'єктів в наборі даних за розміром (а) і за формою (б).

Під час навчання необхідно визначити, які із запропонованих областей відповідають істинному стану детектованого об'єкта, і навчити мережу відповідно до цього. Для кожного справжнього стану об'єкта вибирається область з відповідним розташуванням, співвідношенням сторін і масштабом. Вибирається область пропозиції на основі міри  $IoU$ , яка визначається як відношення перетину запропонованої області та істинної до об'єднання цих областей:

$$IOU(truth, pred) = \frac{truth \cap pred}{truth \cup pred}$$

Завдання навчання - мінімізація функції втрат. Загальна функція втрат є зваженою сумою втрати локалізації (*loc*) і втрати передбачення (*conf*):

$$L(x, c, l, g) = \frac{1}{N} \left( L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right),$$

де  $N$  – число запропонованих областей, що збіглися.

Втрати локалізації визначаються як функція  $smooth_{L1}$  між параметрами передбаченої ( $l$ ) і справжньої ( $g$ ) області:

$$L_{loc}(x, l, g) = \sum_{i \in Pos} \sum_{m \in \{x_0, y_0, w, h\}} x_{i,j}^k smooth_{L1}(l_i^m - \hat{g}_j^m),$$

Графік функції  $smooth_{L1}$  зображений на рисунку 3.6

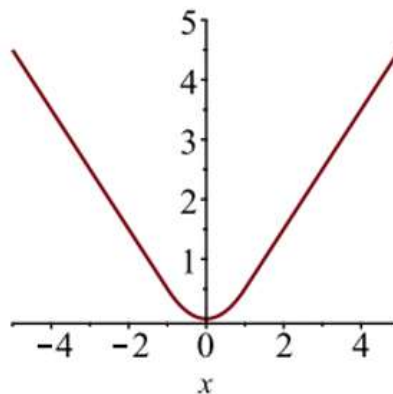


Рисунок 3.6 — Графік функції  $smooth_{L1}$

У розробленій моделі розміри областей пропозиції визначені як  $\{0.05, 0.08, 0.12, 0.16, 0.21, 0.27, 0.5\}$  і співвідношення сторін як  $\{1, 0.7, 0.5, 0.4, 0.3\}$  відповідно до розміру і формою об'єктів, які містить використовуваний набір даних.



Шляхом комбінації прогнозів для всіх запропонованих областей з різним розміром і різним співвідношенням сторін з усіх точок на карті ознак, виходить набір прогнозів, що покривають об'єкти виявлення різних форм і розмірів.

Для створення моделі, більш стійкої до змін розміру і форми вхідного зображення, кожне навчальне зображення випадково перетворюється одним з методів:

- Використовується вихідне зображення;
- Вирізається фрагмент зображення з мінімальним значенням  $IoU$  з об'єктом, що дорівнює 0,1, 0,3, 0,5, 0,7 або 0,9;
- Вирізається випадковий фрагмент зображення. Розмір кожного фрагмента 0.1 ... 1 від вихідного зображення, співвідношення сторін  $\frac{1}{2}$  ... 2;
- Зображення дзеркально відбивається по горизонталі;
- Зображення буде повернуто на випадковий кут.

Як було сказано в розділі 2.2, функція активації необхідна для внесення в нейронну мережу нелінійності. Класична функція активації - логістичний сигмоїд. В моделі, що розробляється, застосовується функція активації  $ReLU$

$$ReLU(x) = \begin{cases} 0, & \text{если } x < 0 \\ x, & \text{если } x \geq 0 \end{cases}$$

Графік функції зображено на рисунку 3.7

Модель, у якій багато вільних параметрів погано узагальнюється і в результаті недостатньо добре пророкує потрібні значення в нових точках. Сучасні моделі нейронних мереж містять мільйони ваг, тому їх необхідно регуляризувати - обмежити значення оптимізованих змінних. Це дозволяє краще узагальнювати мережу на різні приклади, тобто добре передбачати потрібні значення в нових точках.

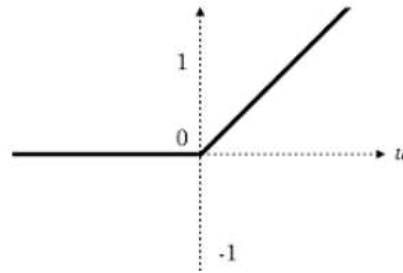


Рисунок 3.7 — Графік функції *ReLU*

Така регуляризація називається скороченням ваг (*weight decay*), так як вона призводить до зменшення їх абсолютних значень.

Також використовується додавання валідаційної множини. При навчанні на основній тренувальній множині також буде обчислюватися помилка і на валідаційній множині. Помилка на валідаційній множині буде добре оцінювати помилку і на нових точках (тестовій множині), тому що вона взята з даних тієї ж природи, але на яких не навчалася. Зупинити навчання можна тоді, коли почне погіршуватися помилка на валідаційній множині, цей підхід називається методом ранньої зупинки (*early stopping*). Набір даних розділений на навчальний, валідаційний і тестувальний в співвідношенні 50%, 25%, 25% відповідно.

Навчання мережі - оптимізація функції помилки методом градієнтного спуску, тобто пошук локального оптимуму. Тому, важливо визначити, де починати цей локальний пошук - в залежності від якості початкового наближення можна потрапити в різні локальні оптимуми. Хороша ініціалізація дозволяє навчити глибокі мережі швидше і краще.

В моделі, що розробляється початкова ініціалізація виконується попереднім навчанням потрібних мереж на великих наборах даних, таких як PASCAL VOC і MS COCO. Такий метод називається перенесенням навчання (*transfer learning*) [17].

### 3.3 Обґрунтування вибору мови, середовища програмування та бібліотек

Для вирішення поставленої задачі нам потрібно використати мову програмування, на якій можна працювати з бібліотеками комп'ютерного зору. Під дану вимогу потрапляють C, Python та Java. Розглянемо кожен з цих мов.

C (Cі) — мова програмування високого рівня з підтримкою декількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Одними з особливостей мови C++ є:

**Продуктивність.** Швидкість роботи програм на C++ практично не поступається програмам на C, хоча надає нові можливості і нові засоби.

**Універсальність.** На мові C++ розробляють програми для самих різних платформ і систем.

**Можливість роботи на низькому рівні з пам'яттю, адресами, портами.**  
**Можливість обчислення на етапі компіляції з використанням шаблонів і створення узагальнених алгоритмів для різних типів даних, їх спеціалізація.**

**Можливість перевантаження операторів.**

**Можливість перевантаження імен функцій.**

**Підтримка узагальненого програмування через шаблони.**

Java — мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

Один із ключових принципів розробки мови Java полягав у забезпеченні захисту від несанкціонованого доступу. Програми на Java не можуть викликати глобальні функції й одержувати доступ до довільних системних ресурсів, що забезпечує в Java рівень безпеки, недоступний для інших мов. Багатофункціональність середовища Web висуває надзвичайно високі вимоги до надійності програм. Як наслідок, при розробці Java пріоритет був відданий можливості створення стійких до помилок програм. Java звільняє програміста від хвилювань з приводу багатьох поширених причин, які викликають помилки програмування. Ще виконуюча система Java бере на себе «прибирання сміття», тобто автоматично звільняє пам'ять, яка була розподілена динамічно. Java підтримує об'єктно-орієнтовану обробку виключних ситуацій подібно до C++. Але на відміну від C++ в Java обробка виключних ситуацій є обов'язковою.

Java розроблялася з орієнтацією на вимоги до створення інтерактивних програм, які працюють з мережею. З цією метою Java підтримує багатозадачність програмування, яке дозволяє легко розробляти програми, що виконують багато процесів одночасно. Java є системою, яка легко розширюється за рахунок створення нових стандартних класів та бібліотек.

Python - високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує структурне, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети.

Еталонної реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ [18]. Він поширюється під вільною ліцензією Python Software Foundation License, що дозволяє

використовувати його без обмежень в будь-яких додатках, включаючи пропрієтарні.

Для реалізації програми потрібне середовище що буде задовольняти хоча б мінімальні умови, такі як підтримка коду Python, можливість виділення кольорами різних структурних елементів коду, можливість зручного підключення додаткових бібліотек. Більшість середовищ програмування вже мають усі данні опції. Тому візьмем Visual Studio Code через його простоту, розповсюдженість та схожість роботи на різних системах таких як Windows, Linux та інші.

При розробці програми була використана бібліотека для комп'ютерного зору OpenCV.

OpenCV (англ. Open Source Computer Vision Library, бібліотека комп'ютерного зору) — бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях [19].

Бібліотека містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору. Алгоритми OpenCV застосовують у таких сферах:

- Аналіз та обробка зображень,
- Системи з розпізнавання обличчя,
- Ідентифікації об'єктів,
- Розпізнавання жестів у відеопотоці,
- Відстежування переміщення камери,
- Побудова 3D моделей об'єктів,
- Створення 3D хмар точок зі стерео камер,

- Склеювання зображень між собою, для створення зображень всієї сцени з високою роздільною здатністю,
- Система взаємодії людини з комп'ютером,
- Пошук схожих зображень із бази даних,
- Усування ефекту червоних очей при фотозйомці зі спалахом,
- Стеження за рухом очей,
- Аналіз руху,
- Ідентифікація об'єктів,
- Сегментація зображення,
- Трекінг відео,
- Розпізнавання елементів сцени і додавання маркерів для створення доповненої реальності та інші.

Як програмне забезпечення була обрана бібліотека глибокого навчання Keras. Keras - головна міжнародна надбудова над бібліотекою TensorFlow, яка використовує її для обчислень [20]. У TensorFlow реалізовані тензорні обчислення, які дозволяють проводити векторні обчислення, що підвищує ефективність навчання нейронних мереж [21].

Для навчання моделі використаний сервіс Google Cloud з наступними характеристиками:

- Платформа ЦП - Intel Sandy Bridge;
- Графічні процесори - NVIDIA Tesla P100;
- 78 Гб RAM

Тепер коли ми визначились з бібліотекою, фреймворком та алгоритмом для його запуску та коректної роботи нам необхідні ваги для нейронної мережі. Для даної задачі є спеціальна відкрита бібліотека TensorFlow. TensorFlow - відкрита програмна бібліотека для машинного навчання, розроблена компанією Google для вирішення завдань побудови і тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття. Застосовується як для досліджень, так і для розробки

власних продуктів Google. Основний API для роботи з бібліотекою реалізований для Python, також існують реалізації для C ++, Haskell, Java, Go і Swift.

### 3.4 Програмна реалізація інформаційної технології ідентифікації пішоходів у відеопотоці

На вхід програма буде приймати відеопотік з веб-камери і розбивати цей потік на фрейми. Кожен фрейм буде аналізуватись за допомогою алгоритму з бібліотеки OpenCV. Узагальнений алгоритм роботи програми ідентифікації пішоходів у відеопотоці наведено на рис. 3.8.

Будемо використовувати наступну тренувальну конфігурацію:

```
{
  "lr": 0.0001,
  "epochs": 10,
  "batch_size": {
    "train": 8
  },
  "input_size": {
    "width": 300,
    "height": 300
  },
  "bn_momentum": 0.01,
  "gpu_devices": [0],
  "data_workers": {
    "train": 3
  },
  "dataset_tags": {
    "train": "train"
  },
  "subdivisions": {
    "train": 1
  },
  "print_every_iter": 10,
  "weights_init_type": "continue_training",
  "enable_augmentations": true
}
```

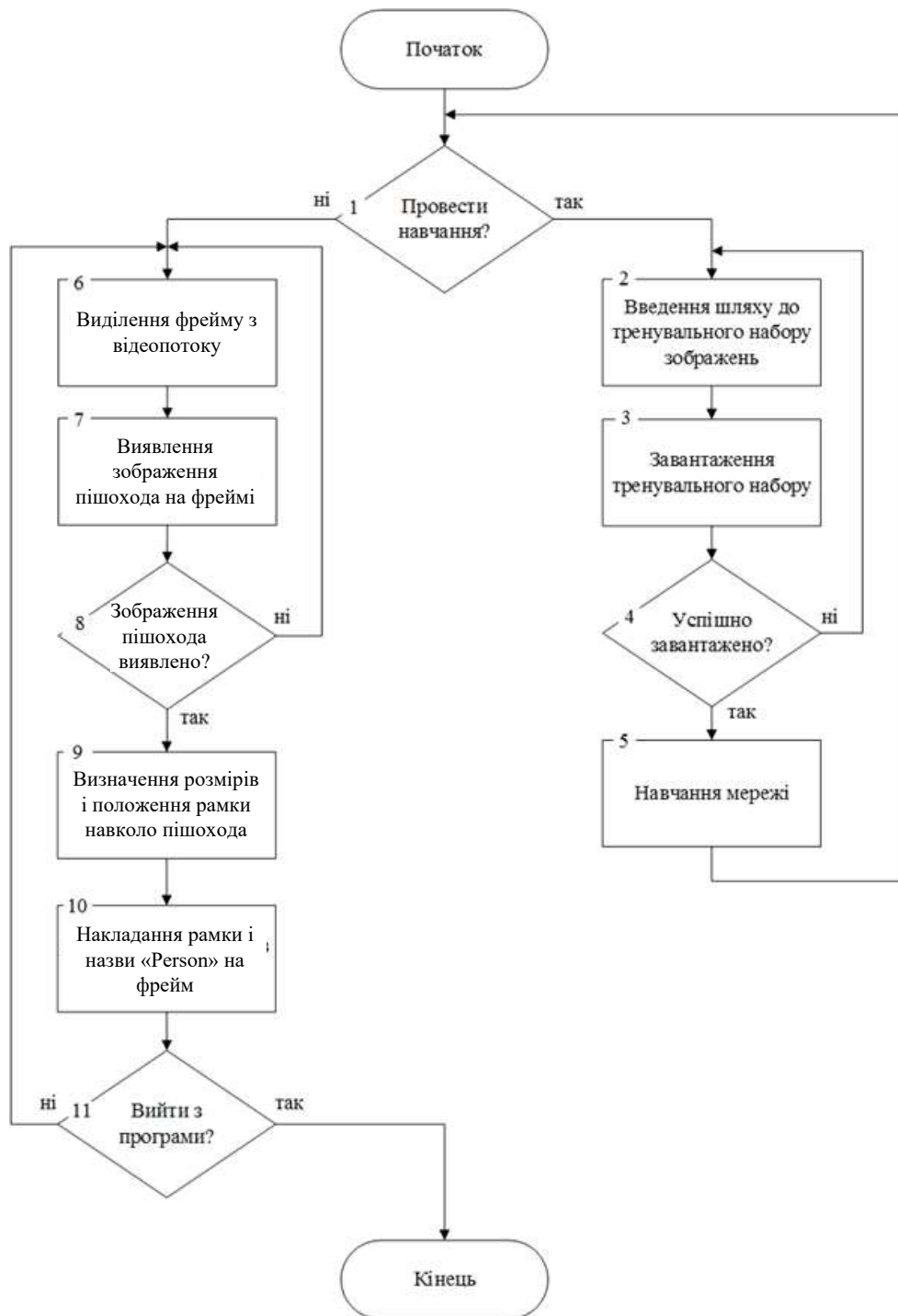


Рисунок 3.8 — Узагальнений алгоритм роботи програми ідентифікації пішоходів у відеопотоці

Модель налаштування виводу - група містить унікальні параметри для кожної моделі:

- `gpu_device` - пристрій для виведення. Зараз підтримується тільки один GPU.



- `confidence_tag_name` - ім'я мітки довіри для передбачених прив'язок.
- `mode` - група містить всі налаштування режиму:
- `name` - назва режиму визначає, як застосувати НМ до зображення (наприклад, `full_image` - застосувати НМ до повного зображення)
  - `model_classes` - які класи будуть використовуватися, напр. НМ виробляє 80 класів, і ми збираємось використовувати лише декілька і ігнорувати інші. У цьому випадку ми повинні встановити поле `save_classes` зі списком зацікавлених імен класів. Рядок `add_suffix` буде доданий до нового класу для запобігання подібним іменам класів з існуючими класами в проєкті. Якщо ми збираємось використовувати всі класи моделей, просто встановіть `"save_classes": "__all__"`.

Приклад конфігурації повного зображення:

```
{
  "model": {
    "gpu_device": 0,
    "confidence_tag_name": "confidence"
  },
  "mode": {
    "name": "full_image",
    "model_classes": {
      "save_classes": "__all__",
      "add_suffix": "_yolo"
    }
  }
}
```

Лістинг програмного коду наведено в додатку Б.

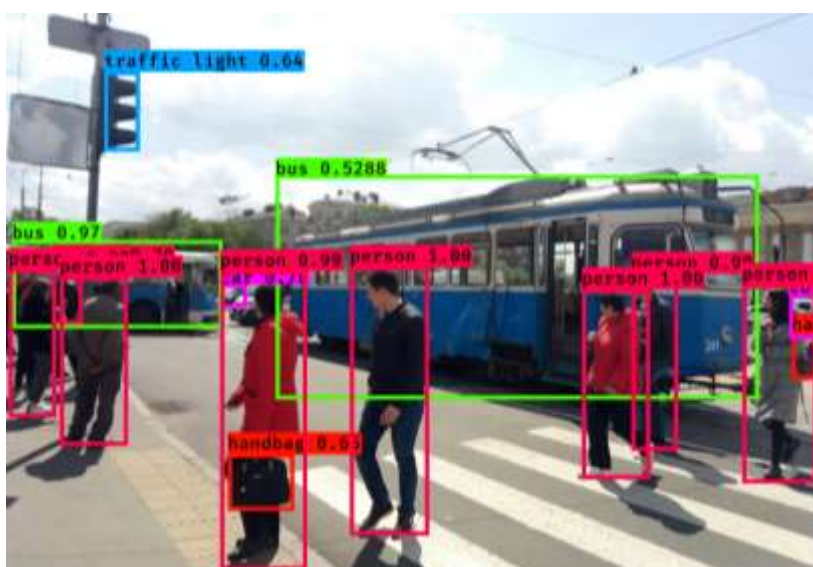
### **3.5 Тестування та аналіз результатів роботи програми ідентифікації пішоходів у відеопотоці на основі згорткових нейромереж**

Тестування показало надійну роботу програми. Інструкція користувача наведена у Додатку А. На рис. 3.9а наведено початкове зображення відеофрейма з видом дорожньої обстановки у м.Вінниця. А на рис. 3.9б

наведено зображення цього ж відеофрейма після обробки розробленою програмою.



а)



б)

Рисунок 3.9 — Результат роботи розробленої програми

Інші результати роботи розробленої програми наведено у Додатку В.

Результати вихідної моделі SSD на наборі даних PASCAL VOC зображені на рисунку 3.10. Вихідна модель забезпечує середню точність  $AP = 0,774$

Застосувавши вихідну модель без змін на використовуваний набір даних Caltech Pedestrian Dataset, виходять результати, зображені на рис. 3.11, з якого видно, що якість виявлення значно впало з  $AP = 0,796$  до  $AP = 0,470$ .

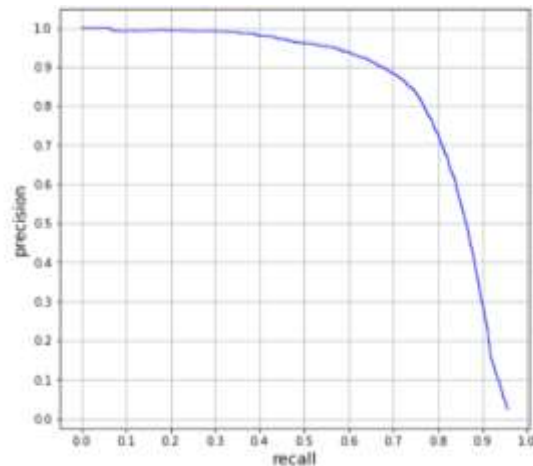


Рисунок 3.10 — Результати вихідної моделі SSD на наборі даних PASCAL VOC

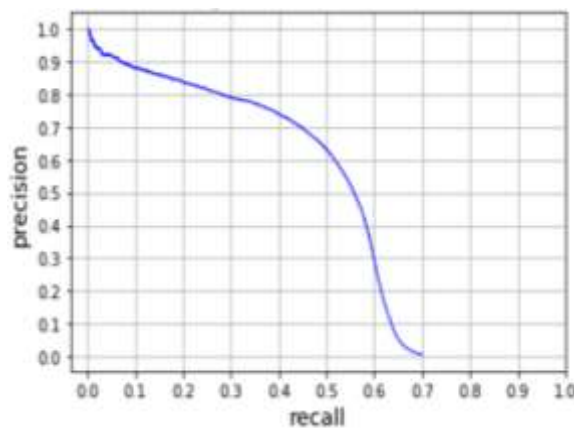


Рисунок 3.11 — Результат застосування вихідної моделі SSD на наборі даних Caltech Pedestrian Dataset.

Таким чином, можна визначити, що вихідна модель недостатньо добре виявляє пішоходів на реальному зображенні. Набір даних Caltech Pedestrian Dataset містить реальне відеозображення, яке отримано з відеокамери автомобіля. На рис. 3.12 зображено приклад зображення з PASCAL VOC, який містить об'єкт «person», а на рис. 3.13 - приклад кадру з набору даних Caltech

Pedestrian Dataset, який містить об'єкт «pedestrian». Таким чином, можна побачити, що виявлення пішоходів на реальному зображенні є більш складним завданням, так як об'єкти можуть бути різного розміру, форми і накладатися один на одного. Також набір даних Caltech має дозвіл 640x480, а на вхід розробленої моделі необхідно подавати зображення з роздільною здатністю 300x300, що вносить додаткові спотворення в вихідне зображення.



Рисунок 3.12 — Приклад зображення з набору даних PASCAL VOC, що містить об'єкт класу «person»



Рисунок 3.13 — Приклад зображення з набору даних Caltech Pedestrian Dataset, що містить об'єкти класу «pedestrian»

З отриманих результатів випливає, що необхідно донавчання моделі на наявному наборі даних. Результат навчання представлений на рисунку 3.14, в якості метрики навчання використана функція втрат на навчальному та перевірконому наборах:

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g)).$$

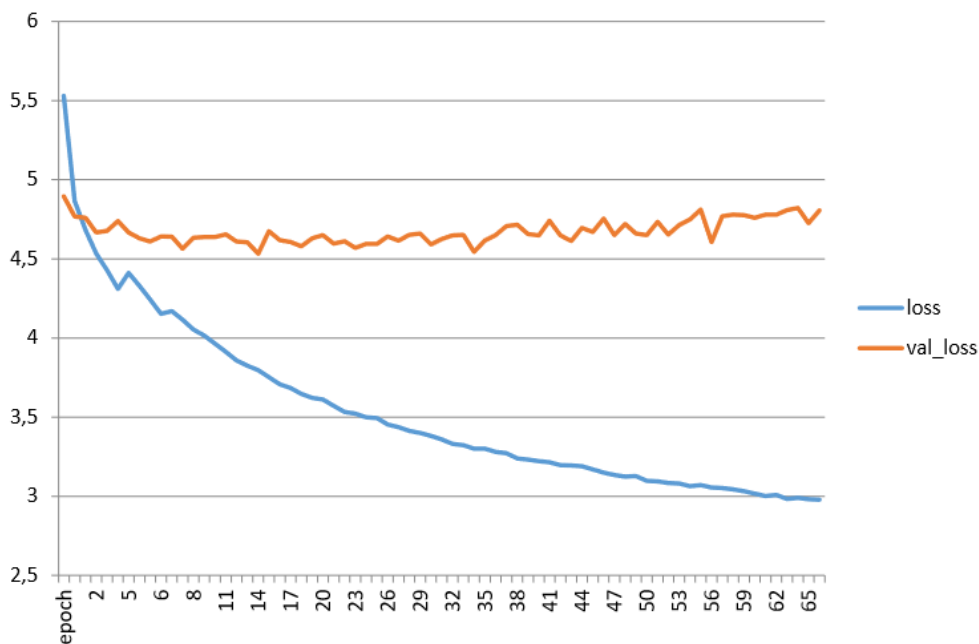


Рисунок 3.14 — Результат навчання моделі з отриманням перенавчання.

На графіку видно, що відбувається проблема перенавчання (overfitting) - модель навчається, функція втрат зменшується на тренувальному наборі даних, але на перевірконому наборі даних зменшення не відбувається. З цього випливає, що необхідні додаткові заходи для усунення проблеми перенавчання. В якості вирішення проблеми використано розширення тренувального набору даних (аугментація).

Також для зменшення перенавчання використана регуляризація L1 [10].

Після проведених модифікацій підвищилася стійкість моделі до перенавчання, що показано на рисунку 3.15. Мінімальне значення функції втрат

3,63 досягається на 44 епісі. Цей результат використовується для подальшого розрахунку точності моделі.

Результат тестування навченої моделі на тестувальному наборі даних зображень на рис. 3.16. Оптимізований алгоритм забезпечує середню точність  $AP = 0,838$ .

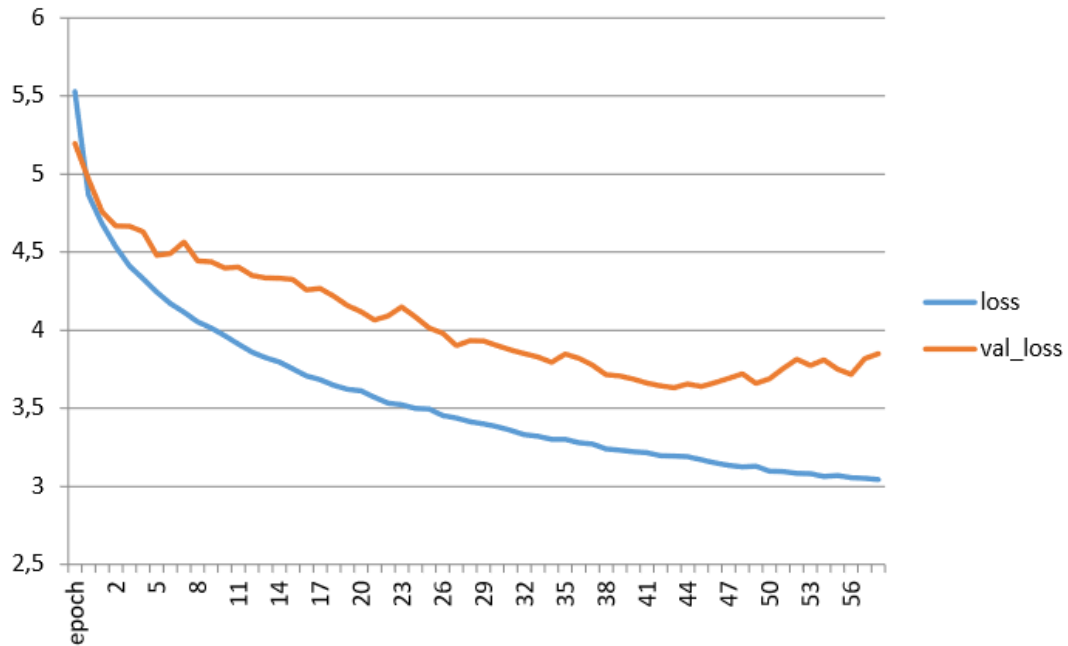


Рисунок 3.15 — Результат навчання моделі після підвищення стійкості до перенавчання.

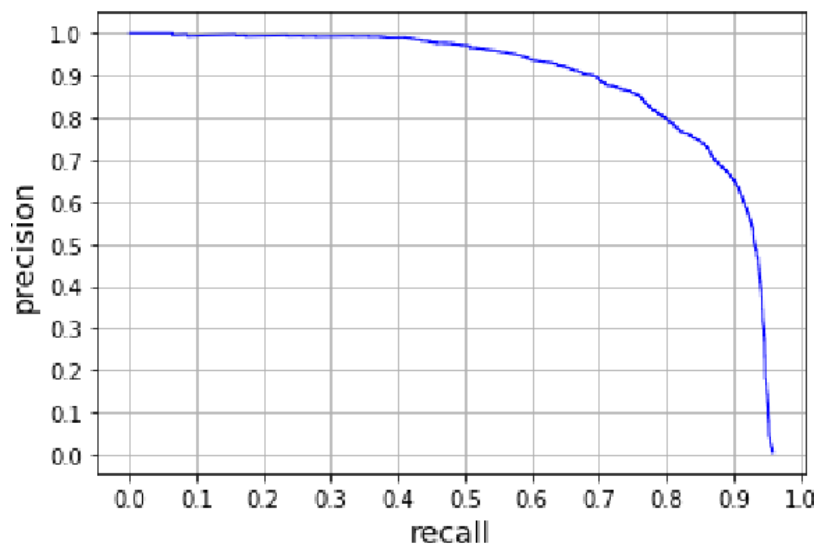


Рисунок 3.16 — Результат ідентифікації пішоходів удосконаленою моделлю.

В ході проведеного моделювання видно, що розроблена модель має точність ідентифікації пішоходів вище, ніж існуючі моделі на основі згорткових нейронних мереж. Вихідна модель SSD мала точність ідентифікації пішоходів  $AP = 0,774$ . Як оптимізації моделі використані:

- збільшення (аугментація) тренувальних даних;
- зміна параметрів вихідної мережі, таких, як розмір вікна передбачення об'єктів;
- використання додаткової регуляризації ваг моделі;
- додаткова обробка тренувальних і перевірочних даних.

В результаті проведеної оптимізації, отримана точність ідентифікації пішоходів на наборі даних Caltech Pedestrian Dataset  $AP = 0,838$ , що вище, ніж точність вихідної моделі на наборі даних PASCAL VOC.

Для доведення досягнення поставленої в роботі мети - підвищення достовірності роботи програми для ідентифікації пішоходів у відеопотоці – було протестовано роботу розробленої програми та програми-аналога RetinaNet [6] на 15 прикладах відеозображень із тестової вибірки. Результати тестування представлені у табл. 3.2.

Таблиця 3.2 – Результати тестування розробленої програми та програми-аналога RetinaNet

Програмний засіб	Кількість відеозображень у тестовій вибірці	Кількість пішоходів на відеозображеннях	Кількість правильно ідентифікованих пішоходів	Достовірність ідентифікації пішоходів, %
Програма RetinaNet	20	130	88	67,7
Розроблена програма	20	130	109	83,8

Із табл. 3.2 видно, що розроблена програма має вищу достовірність ідентифікації пішоходів (83,8%), ніж аналогічна програма (67,7%), а значить

достовірність ідентифікації пішоходів покращена на 16,1%, тобто мета роботи досягнута.

Інструкцію до користування програмою наведено в додатку А, деякі ілюстрації до програми (у т.ч. скріншоти) наведено в додатку В.

### **3.6 Висновок**

В ході практичної реалізації інформаційної технології ідентифікації пішоходів у відеопотоці у даному розділі було розглянуто архітектуру та алгоритм роботи згорткової нейронної мережі SSD. Також були запропоновані заходи з поліпшення моделі SSD. Було обгрунтовано вибір мови програмування Python, середовища програмування Visual Studio Code, бібліотеки комп'ютерного зору OpenCV, бібліотеки глибокого навчання Keras та бібліотеки векторно-матричних обчислень TensorFlow. Також було розроблено узагальнений алгоритм роботи програми, описано роботу основних частин програми та наведено відповідні фрагменти коду. В результаті було розроблено програмне забезпечення ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі, яке порівняно з аналогом має кращу на 16,1% достовірність ідентифікації пішоходів у відеопотоці. Таким чином, мета роботи досягнута – достовірність ідентифікації пішоходів у відеопотоці підвищена. Розроблена програма повністю відповідає завданню, що підтверджується її тестуванням.



## 4 ЕКОНОМІЧНА ЧАСТИНА

### 4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності [22].

Результатом виконання магістерської кваліфікаційної роботи є інформаційна технологія ідентифікації пішоходів на відео на основі згорткових нейромереж. Для проведення технологічного аудиту залучено трьох незалежних експертів: Колодний Володимир Володимирович (к.т.н., доцент каф. комп'ютерних наук ВНТУ) Колесницький Олег Костянтинович (к.т.н., доцент каф. комп'ютерних наук ВНТУ), та Озеранський Сергій Володимирович (к.т.н., доцент каф. комп'ютерних наук ВНТУ).

Оцінювання комерційного потенціалу буде здійснене за критеріями, що наведені в таблиці 4.1 [22].

Таблиця 4.1 - Критерії оцінювання комерційного потенціалу розробки  
бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
<b>Технічна здійсненність концепції:</b>					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
<b>Ринкові переваги (недоліки):</b>					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
<b>Ринкові перспективи</b>					
5	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
<b>Практична здійсненність</b>					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

## Продовження таблиці 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Колодний	2 – Колесницький	3 – Озеранський
	Бали, виставлені експертами:		
1	3	4	4
Ринкові переваги (недоліки):			
2	2	2	2
3	3	4	3
4	4	4	3
5	3	3	4
Ринкові перспективи			
5	3	3	2
7	2	3	3
Практична здійсненність			
8	3	3	3
9	2	2	3
10	4	4	4
11	3	3	3
12	2	3	2
Сума балів	СБ <sub>1</sub> =34	СБ <sub>2</sub> =38	СБ <sub>3</sub> =36
Середньоарифметична сума балів $\overline{СБ}$	36		

За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

З таблиці зрозуміло, що рівень комерційного потенціалу розробки становить 36 балів, така оцінка відповідає рівню «вище середнього».

Нова розробка являє собою інформаційну технологія ідентифікації пішоходів на відео на основі згорткових нейромереж, що дозволяє виявляти пішоходів на відео.

Розглянемо основну систему-конкурент, які теж дають змогу виявляти пішоходів на відео.

- RetinaNet - це одноступінчаста модель виявлення об'єктів, яка використовує функцію фокусної втрати для усунення дисбалансу класу під час навчання. Фокальна втрата застосовує модулюючий термін до перехресних ентропійних втрат, щоб зосередити навчання на жорстких негативних прикладах. RetinaNet - це єдина, уніфікована мережа, що складається з магістральної мережі та двох конкретних підмереж. Магістраль відповідає за обчислення карти згорткових характеристик по всьому вхідному зображенню і є незалежною згортковою мережею. Перша підмережа виконує класифікацію згорткових об'єктів на вихідних даних магістралі; друга підмережа виконує згорткову регресію обмежувальної рамки. Дві підмережі відрізняються простим дизайном, який автори пропонують спеціально для одноетапного щільного

виявлення. У таблиці 4.4 наведені основні технічні показники аналога і розроблюваного програмного продукту

Таблиця 4.4 - Основні технічні показники аналога і нового програмного продукту

Показники (приклад)	Аналог	Нова розробка	Відношення параметрів нової розробки до параметрів аналога
Функціональність	Висока	Висока	1
Надійність	Середня	Висока	1.5
Сумісність	Висока	Висока	1
Економія ресурсів і часу	Середня	Висока	1,5
Простота використання	Середня	Висока	1,5

Хоча за функціональністю нова розробка і програма аналог однакові і дозволяють детектувати пішоходів на відео, нова розробка дозволяє виявляти пішоходів на відео з більшою точністю ніж у програми-аналога.

За рахунок застосування згорткових нейромереж нова розробка дозволить виявляти пішоходів швидше та точніше ніж у програми-аналога, що побудований на основі пірамідальної нейронної мережі, яка працює із попередньо отриманими ознаками зображення.

Також, нова розробка простіша та зручніша у використанні ніж програму-аналог.

Даний продукт в основму спрямований на реалізацію на українському ринку у сфері автомобільних систем.

На даний продукт очікується високий попит, адже основним застосуванням технології розпізнавання пішоходів є її використання в автомобілях для поліпшення системи безпеки, що завжди є актуальною темою в зв'язку з розповсюдженістю автомобілів та ДТП. Також даний продукт може застосовуватись в безпілотних автомобілях, які вже з'явилися і будуть більш розповсюдженими в майбутньому.

Розробку планується реалізовувати шляхом ліцензійної угоди. Розробку планується продавати шляхом щомісячної підписки на продукт з trial версією на перший місяць.

Розробка потребує залучення незначних фінансових ресурсів, а більшість матеріалів для її реалізації відомі і давно використовуються. Існує достатня кількість фахівців на ринку праці, що не буде вповільнювати задачу пошуку розробників.

Потрібно провести пошук інвесторів та партнерів для комерціалізації розробленого продукту.

## **4.2 Прогнозування витрат на виконання наукової роботи та впровадження результатів**

Здійснимо прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи для розробки програмного забезпечення.

Виконаємо розрахунок витрат з урахуванням того, що розробкою займався один розробник програмного забезпечення.

Основна заробітна розробника-дослідника  $Z_o$ :

$$Z_o = \frac{M}{T_p} \cdot t \text{ [грн]}, \quad (4.1)$$

де  $M$  – місячний посадовий оклад – 13529 грн;

$T_p$  – число робочих днів в місяці;  $T_p = 22$  дні;

$t$  – число робочих днів роботи розробника-дослідника - 69.

Працівник	Оклад $M$ , днів	Оплата за робочий день, грн.	Число днів роботи, $t$	Витрати на оплату праці, грн.
Науковий керівник	6500	295.45	5	1477.27
Інженер-програміст	13529	614.95	69	42431.82
Всього:				43909,09

Додаткова заробітна плата  $Z_d$  розробника розраховується як 12% від основної заробітної плати:

$$Z_d = 0,12 \cdot 43909,09 = 5269,09 \text{ (грн).}$$

Нарахування на заробітну плату  $H_{зп}$  розробника становить:

$$H_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100} \text{ [грн]}, \quad (4.2)$$

де  $Z_o$  – основна заробітна плата розробника;

$Z_d$  – додаткова заробітна плата розробника;

$\beta$  – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування – 22%.

$$H_{зп} = (43909,09 + 5269,09) \cdot 0,22 = 10819,19 \text{ (грн).}$$

У спрощеному вигляді амортизаційні відрахування розраховуємо за формулою:

$$A = (Ц \cdot T) / (12 \cdot T_v) \text{ [грн]}, \quad (4.3)$$

де  $Ц$  – загальна балансова вартість обладнання, приміщення тощо, грн;

$T$  – фактична тривалість використання, міс;

$T_B$  – термін використання обладнання, приміщень тощо, роки.

Розробка програмного забезпечення проводилася протягом 3 місяців.

Зроблені розрахунки зведено до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування

Найменування	Балансова вартість, грн	Термін використання, роки	Фактична трив. використання, міс.	Величина амортизаційних відрахувань, грн
Офісне приміщення	110000	20	4	1833,33
Комп'ютер	21300	4	4	1775,00
Всього				3608,33

Матеріали, що використовуються при виготовленні продукту внесено до таблиці 4.6.

Таблиця 4.6 – Матеріали, що використовуються при виготовленні даного продукту

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено, шт.	Вартість витраченого матеріалу, грн
Папір (упаковка)	84,00	1	84,00
Канцтовари	9,00	2	18,00
Карта пам'яті	520	1	520,00
Всього			622,00

Витрати на електроенергію  $V_e$  розраховуються за допомогою формули:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi} \text{ [грн]}, \quad (4.4)$$

де  $V$  – вартість 1кВт електроенергії;



$P$  – установлена потужність обладнання, кВт;

$\Phi$  – фактична кількість годин роботи комп'ютера при створенні програмного продукту, годин;

$K_p$  – коефіцієнт використання потужності.

Отже, витрати на енергію становлять:

$$B_e = 1,88 \cdot 0,5 \cdot 704 \cdot 0,4 = 264,7 \text{ (грн)}.$$

Витрати за доступ до Інтернет можна розрахувати за формулою:

$$B_{дi} = C_{дi} \cdot T \text{ [грн]}, \quad (4.5)$$

де  $C_{дi}$  – це ціна доступу за місяць;

$T$  – кількість місяців використання доступу до мережі.

$$B_{дi} = 139 \cdot 4 = 556,00 \text{ (грн)}.$$

Інші витрати  $B_{ін}$  охоплюють: витрати на управління організацією, оплату службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Інші витрати  $I_b$  можна прийняти як 50% від суми основної заробітної плати розробника:

$$B_{ін} = 0,5 \cdot 43909,09 = 21954,54 \text{ (грн)}.$$

Сума всіх попередніх статей витрат зумовлює витрати на виконання даної частини (розділу, етапу) роботи –  $B$ .

$$B = Z_o + Z_d + H_{зп} + A + B_{мат} + B_e + B_{дi} + B_{ін} \text{ [грн]},$$

$$B = 43909,09 + 5269,09 + 10819,19 + 3608,33 + 622 + 264,7 + 556 + 21954,54 = 87002,94 \text{ (грн)}.$$

2-й етап. Проведемо розрахунок загальних витрат на виконання та впровадження результатів даної роботи. Загальна вартість всієї наукової роботи визначається за  $B_{заг}$  формулою:

$$V_{\text{заг}} = \frac{B}{\alpha} \text{ [грн]}, \quad (4.6)$$

де  $\alpha$  – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях.

Оскільки у роботі було задіяно лише одну людину, якою виконується вся робота, то  $\alpha$  становить 1. Підставивши дані у формулу, отримуємо:

$$V_{\text{заг}} = 87002,94 \text{ (грн)}.$$

3-й етап. Прогнозування загальних витрат на виконання та впровадження результатів виконаної роботи. Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної роботи здійснюється за формулою:

$$ЗВ = \frac{V_{\text{заг}}}{\beta} \text{ [грн]}, \quad (4.7)$$

де  $\beta$  – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то  $\beta \approx 0,1$ ;
- на стадії технічного проектування, то  $\beta \approx 0,2$ ;
- на стадії розробки конструкторської документації, то  $\beta \approx 0,3$ ;
- на стадії розробки технологій, то  $\beta \approx 0,4$ ;
- на стадії розробки дослідного зразка, то  $\beta \approx 0,5$ ;
- на стадії розробки промислового зразка,  $\beta \approx 0,7$ ;
- на стадії впровадження, то  $\beta \approx 0,9$ .

Отже, підставимо дані в формулу й отримаємо результат:

$$ЗВ = \frac{87002,94}{0,9} \approx 96669,93 \text{ (грн)}.$$

### 4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

В умовах ринку узагальнюючим позитивним результатом, що його отримує підприємство від впровадження результатів тієї чи іншої розробки, є збільшення чистого прибутку підприємства. Зростання чистого прибутку забезпечить підприємству надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Виконання даної наукової роботи та впровадження її результатів складає приблизно 1 рік. Позитивні результати від впровадження розробки очікуються вже в перші місяці після впровадження.

Проведемо детальніше прогнозування позитивних результатів та кількісне їх оцінювання по роках.

Обчислимо збільшення чистого прибутку підприємства  $\Delta\Pi_i$  для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \cdot \Delta N)_i \text{ [грн]}, \quad (4.8)$$

де  $\Delta\Pi_{\text{я}}$  – покращення основного якісного показника від впровадження результатів розробки у даному році;

$N$  – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$\Delta N$  – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$  – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

$n$  – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Припустимо, що внаслідок впровадження результатів наукової розробки чистий прибуток підприємства збільшиться на 150 грн., а кількість одиниць реалізованої послуги збільшиться: протягом першого року – на 350 од., протягом другого року – ще на 200 од., протягом третього року – ще на 150 од.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 1 шт., а прибуток, який отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки був 600 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового прибутку.

Збільшення чистого прибутку підприємства  $\Delta\Pi_1$  протягом першого року складе:  $\Delta\Pi_1=600\cdot 1+(600+150)\cdot 350=263100,00$  (грн).

Збільшення чистого прибутку підприємства  $\Delta\Pi_2$  протягом другого року:  $\Delta\Pi_2=600\cdot 1+(600+150)\cdot (350+200)=413100,00$  (грн)

Збільшення чистого прибутку підприємства  $\Delta\Pi_3$  протягом третього року становитиме:

$$\Delta\Pi_3=600\cdot 1+(600+150)\cdot (350+200+150)=525600,00 \text{ (грн).}$$

Отже, розрахунки показують, що комерційний ефект від впровадження розробки виражається у значному збільшенні чистого прибутку підприємства.

#### **4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності**

Щоб оцінити доцільність фінансування проекту, необхідно провести розрахунки ефективності вкладених інвестицій. Основними показниками є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

На першому етапі розрахуємо теперішню вартість інвестицій PV, що вкладаються в наукову розробку. Такою вартістю ми можемо вважати прогнозовану величину загальних витрат ЗВ на виконання та

впровадження результатів науково дослідної роботи. Тобто будемо вважати, що  $ZB = PV = 96669,93$  грн.

На другому етапі розраховуємо очікуване збільшення прибутку  $\Delta\Pi_i$ , яке отримає підприємець від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку було розраховане раніше. Результати вкладених у наукову розробку інвестицій можна побачити по тому, що у першому році підприємство отримає збільшення чистого прибутку на 263100,00 грн. відносно базового року, у другому році – збільшення чистого прибутку на 413100,00 грн (відносно базового року), у третьому році – збільшення чистого прибутку на 525600,00 грн (відносно базового року).

На третьому етапі, для спрощення подальших розрахунків, будемо вісь часу, на яку наносимо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Платежі показуються у ті терміни, коли вони здійснюються. Тоді рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 4.1.



Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

На четвертому етапі розраховуємо абсолютну ефективність вкладених інвестицій  $E_{abc}$  за формулою:

$$E_{abc} = (ПП - PV) [грн], \quad (4.9)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки, грн, розраховано за формулою 4.10;

PV – теперішня вартість інвестицій  $PV = ЗВ = 96669,93$  (грн).

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t} [грн], \quad (4.10)$$

де  $\Delta\Pi_i$  - збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

$t$  – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

$t$  – період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

Проведемо розрахунки.

$$ПП = \frac{263100,00}{(1 + 0,1)^1} + \frac{413100,00}{(1 + 0,1)^2} + \frac{525600,00}{(1 + 0,1)^3} = 975477,81 \text{ грн.}$$

Тоді ,

$$E_{abc} = 975477,81 - 96669,93 = 878807,88 \text{ (грн).}$$

Оскільки  $E_{abc} > 0$ , то результат від проведення наукових досліджень та їх впровадження принесе прибуток, але це також ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даного проекту.

На п'ятому етапі розрахуємо відносну (щорічну) ефективність вкладених у наукову розробку інвестицій  $E_e$ . Для цього використаємо формулу:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1 \quad (4.11)$$

де  $E_{абс}$  – абсолютна ефективність вкладених інвестицій, грн;

$PV$  теперішня вартість інвестицій  $PV = ЗВ$ , грн;

$T_{ж}$  – життєвий цикл наукової розробки, роки.

$$E_B = \sqrt[3]{1 + \frac{878807,88}{96669,93}} - 1 = 1,16, \text{ або } 116\%.$$

Порівняємо  $E_e$  з мінімальною (бар'єрною) ставкою дисконтування  $\tau_{мін}$ , яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування визначається за формулою:

$$\tau = d + f, \quad (4.12)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках;  $d = (0,14...0,2)$ ;

$f$  – показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = (0,05...0,1)$ .

$$\tau = 0,2 + 0,05 = 0,25$$

Оскільки  $E_B = 116\% > \tau_{мін} = 0,25 = 25\%$ , то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію проєкту інвестицій.

Термін окупності вкладених у реалізацію проєкту інвестицій  $T_{ок}$  можна розрахувати за формулою:

$$T_{ок} = 1 / E_e \text{ [роки]}, \quad (4.13)$$

$$T_{ок} = 1/1,16 = 0,86 \text{ (року)}.$$

Термін окупності вкладених у реалізацію проєкту інвестицій становить 0,86 року. Аскільки  $T_{ок} < 3...5$  років, що свідчить, що фінансування даної наукової розробки є доцільним.

Отже, розрахунок ефективності вкладених інвестицій та періоду їх окупності показав, що фінансування розробки є доцільним.

#### **4.5 Висновок**

В даному розділі було виконано оцінювання комерційного потенціалу розробки інформаційної технології ідентифікації пішоходів на відео на основі згорткових нейромереж. За участю трьох незалежних експертів було визначено рівень потенціалу комерційного розвитку. Аналіз показує, що рівень потенціалу комерційного розвитку вищий за середній. Аналіз комерційного потенціалу проєкту розвитку показує, що розроблена інформаційна система випереджає аналогічні інформаційні системи за показниками та має широкі перспективи. Товар має кращі функціональні показники, тому є досить конкурентоспроможним на ринку. Згідно з розрахунками кошторисів на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи, загальна вартість розробки становить 96669,93 грн. Було розраховано абсолютну ефективність вкладених інвестицій. Ефективність досягла 878 807,88 грн., що свідчить про отримання вигоди від залучення коштів у розробку. Річна ефективність інвестицій у наукову розробку становить 116%, що вище ніж найнижча бар'єрна ставка дисконтування в 25%. Термін окупності інвестицій у проєкт становить 0,86 років, що ще більше демонструє доцільність фінансування нових проєктів розвитку.



## ВИСНОВКИ

В результаті виконання магістерської кваліфікаційної роботи було розв'язано задачу розробки інтелектуальної інформаційної технології та програмного забезпечення класифікації тональності речень згортковою нейронною мережею.

У першому розділі магістерської роботи було проведено аналітичний огляд відомих методів і засобів ідентифікації пішоходів у відеопотоці, виявлені їх можливості, переваги і недоліки. Обґрунтована перспективність застосування методів ідентифікації пішоходів у відеопотоці на основі згорткових нейронних мереж. Було наведено класифікацію методів ідентифікації пішоходів у відеопотоці, детально описано ідентифікатори пішоходів у відеопотоці на основі традиційних підходів. Було проаналізовано недоліки розглянутих ідентифікаторів пішоходів та запропоновано використовувати для побудови ідентифікаторів пішоходів згорткові нейронні мережі, оскільки вони мають переваги.

У другому розділі магістерської кваліфікаційної роботи було обґрунтовано вибір згорткової нейронної мережі для побудови інформаційної технології ідентифікації пішоходів у відеопотоці, обрано архітектуру згорткової нейронної мережі, проаналізовано математичну модель згорткової нейромережі та процес її навчання. За основу для розробки моделі ідентифікації пішоходів у відеопотоці взята модель згорткової нейромережі SSD, яка має оптимальне співвідношення точності і продуктивності. Для згорткових та агрегувальних шарів буде використана функція активації – ReLU. Метод, за яким буде проходити навчання мережі – це метод зворотного поширення помилки. У підсумку було розроблено структуру інформаційної технології ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі..

У третьому розділі в ході практичної реалізації інформаційної технології ідентифікації пішоходів у відеопотоці було розглянуто архітектуру та алгоритм роботи згорткової нейронної мережі SSD. Також були запропоновані заходи з

поліпшення моделі SSD. Було обґрунтовано вибір мови програмування Python, середовища програмування Visual Studio Code, бібліотеки комп'ютерного зору OpenCV, бібліотеки глибокого навчання Keras та бібліотеки векторно-матричних обчислень TensorFlow. Також було розроблено узагальнений алгоритм роботи програми, описано роботу основних частин програми та наведено відповідні фрагменти коду. В результаті було розроблено програмне забезпечення ідентифікації пішоходів у відеопотоці на основі згорткової нейронної мережі, яке порівняно з аналогом має кращу на 16,1% достовірність ідентифікації пішоходів у відеопотоці. Таким чином, мета роботи досягнута – достовірність ідентифікації пішоходів у відеопотоці підвищена. Розроблена програма повністю відповідає завданню, що підтверджується її тестуванням.

У четвертому розділі було виконано оцінювання комерційного потенціалу розробки інформаційної технології ідентифікації пішоходів на відео на основі згорткових нейромереж. Було визначено, що рівень потенціалу комерційного розвитку вищий за середній, а значить розроблена інформаційна система випереджає аналогічні інформаційні системи за показниками та має широкі перспективи. Товар має кращі функціональні показники, тому є досить конкурентоспроможним на ринку. Загальна вартість розробки становить 96669,93 грн. Абсолютна ефективність вкладених інвестицій досягла 878 807,88 грн., що свідчить про отримання вигоди від залучення коштів у розробку. Річна ефективність інвестицій у наукову розробку становить 116%, що вище ніж найнижча бар'єрна ставка дисконтування в 25%. Термін окупності інвестицій у проект становить 0,86 років, що ще більше демонструє доцільність фінансування проекту.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Радченко Д. Ю. Колесницький О. К., «Використання нейромережевого підходу у задачах розпізнавання образів», в Матеріали конференції «XLIX Науково-технічна конференція підрозділів Вінницького національного технічного університету (2020)», Вінниця, 2020, с. 928-929. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/allvntu/index/pages/view/zbirn2020> Дата звернення: Черв. 2020.
2. Benenson R. Ten Years of Pedestrian Detection, What Have We Learned? [Електронний ресурс]. URL: [https://rodrigob.github.io/documents/2014\\_eccvw\\_ten\\_years\\_of\\_pedestrian\\_detection\\_with\\_supplementary\\_material.pdf](https://rodrigob.github.io/documents/2014_eccvw_ten_years_of_pedestrian_detection_with_supplementary_material.pdf) (дата обращения: 15.11.2020).
3. Маенраа Т. The Local Binary Pattern Approach to Texture Analysis — Extensions and Applications. Oulu University Press, 2003.
4. Viola P. Rapid Object Detection using a Boosted Cascade of Simple Features [Електронний ресурс]. URL: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf> (дата обращения: 15.11.2020).
5. Dalal N. Histograms of Oriented Gradients for Human Detection [Електронний ресурс]. URL: [https://lear.inrialpes.fr/pubs/2005/DT05/hog\\_cvpr2005.pdf](https://lear.inrialpes.fr/pubs/2005/DT05/hog_cvpr2005.pdf) (дата обращения: 15.11.2020).
6. Review: RetinaNet — Focal Loss (Object Detection) [Електронний ресурс]. URL: <https://towardsdatascience.com/review-retinanet-focal-loss-object-detection-38fba8afabe4>.
7. Carandini M. Area V1 [Електронний ресурс]. URL: [http://www.scholarpedia.org/article/Area\\_V](http://www.scholarpedia.org/article/Area_V) (дата обращения: 15.11.2020).
8. Krizhevsky A. ImageNet Classification with Deep Convolutional Neural Networks [Електронний ресурс]. URL: <https://papers.nips.cc/paper/4824imagenet>

classification-with-deep-convolutional-neural-networks.pdf (дата обращения: 15.11.2020).

9. Ren S. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [Электронный ресурс]. URL: <https://arxiv.org/pdf/1506.01497> (дата обращения: 15.11.2020).

10. Girshick R. Fast R-CNN [Электронный ресурс]. URL: [https://www.cvfoundation.org/openaccess/content\\_iccv\\_2015/papers/Girshick\\_Fast\\_RCNN\\_ICCV\\_2015\\_paper.pdf](https://www.cvfoundation.org/openaccess/content_iccv_2015/papers/Girshick_Fast_RCNN_ICCV_2015_paper.pdf) (дата обращения: 15.11.2020).

11. Redmon J. You Only Look Once: Unified, Real-Time Object Detection [Электронный ресурс]. URL: <https://pjreddie.com/media/files/papers/yolo.pdf> (дата обращения: 15.11.2020).

12. Liu W. SSD: Single Shot MultiBox Detector [Электронный ресурс]. URL: <https://www.cs.unc.edu/~wliu/papers/ssd.pdf> (дата обращения: 15.11.2020).

13. [Электронный ресурс]. URL: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/results/index.html> (дата обращения: 15.11.2020).

14. Николенко С. Глубокое обучение. СПб.: Питер, 2018. 480 с.: ил. (Серия «Библиотека программиста»).

15. Гудфеллоу Я. Глубокое обучение / пер. с англ. А. А. Слинкина. 2-е изд., испр. М.: ДМК Пресс, 2018. 652 с.: цв. ил.

16. Dollar P. Pedestrian Detection: An Evaluation of the State of the Art [Электронный ресурс]. URL: [http://www.vision.caltech.edu/Image\\_Datasets/CaltechPedestrians/files/PA112pedestrians.pdf](http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/files/PA112pedestrians.pdf) (дата обращения: 15.11.2020).

17. Yosinski J. How transferable are features in deep neural networks? [Электронный ресурс]. URL: <https://arxiv.org/pdf/1411.1792.pdf> (дата обращения: 15.11.2020).

18. Python [Электронный ресурс]. URL: <https://www.python.org/>

19. OpenCV [Электронный ресурс] – режим доступа: <https://uk.wikipedia.org/wiki/OpenCV>.

20. [Електронний ресурс]. URL:<https://keras.io/> (дата звернення: 15.11.2020).

21.[Электронный ресурс]. URL:<https://www.tensorflow.org/> (дата звернення: 15.11.2020).

22.Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.