

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія виявлення озброєних людей у

відеопотоці з використанням згорткових нейромереж»

Виконав: студент 2-го курсу, групи ЗКН-22м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Янковський Є.В.

(прізвище та ініціали)

Керівник: к.т.н., професор каф. КН

Колесницький О.К.

(прізвище та ініціали)

« 07 » 12 2023 р.

Опонент: к.т.н., професор каф. КСУ

Биков М.М.

(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« 08 » 12 2023 р.

Вінниця ВНТУ - 2023 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та
автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

**Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.**

29.08. 2023 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Янковський Євгеній Вячеславович

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж

керівник роботи к.т.н., професор кафедри КН Колесницький О. К.

затверджені наказом вищого навчального закладу від "18" 09 2023 року № 247

2. Строк подання студентом роботи 13.11. 2023 року

3. Вихідні дані до роботи:

вихідні дані – відео у форматі mp4; кількість класів розпізнавання – не менше 4; кількість навчальних зображень – не менше 1000; швидкість розпізнавання – не більше 1 секунди; точність розпізнавання – не менше 95%; використання об'єктно-орієнтованої мови програмування; операційна система Windows.

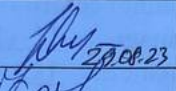
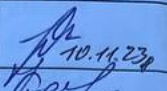
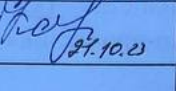
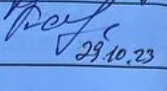
4. Зміст текстової частини:

Вступ, аналіз предметної області розпізнавання озброєних людей, розробка інформаційної технології виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж, програмна реалізація інформаційної технології виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж, тестування та аналіз результатів роботи інформаційної технології виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж, економічна частина, висновки, перелік використаних джерел, додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Схема алгоритму роботи інформаційної технології виявлення озброєних людей у відеопотоці; діаграма діяльності; структура згорткової нейронної мережі; робочі вікна програми розпізнавання озброєних людей; результати роботи програми розпізнавання озброєних людей та порівняння з аналогом.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Колесницький О.К., д.т.н., проф. каф. КН	 29.08.23	 10.11.23
5	Кавецький В.В., к. е. н., доц. каф. ЕПВМ	 27.10.23	 29.10.23

7. Дата видачі завдання 29.08. 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	При- мітка
1	Аналіз предметної області розпізнавання озброєних людей	01.09.23 - 07.09.23	
2	Розробка інформаційної технології виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж	08.09.23 - 24.09.23	
3	Програмна реалізація інформаційної технології виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж	25.09.23 - 15.10.23	
4	Тестування та аналіз результатів роботи інформаційної технології виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж	16.10.23 - 20.10.23	
5	Підготовка економічної частини	21.10.23 - 29.10.23	
6	Апробація та/або впровадження результатів дослідження	30.10.23 - 05.11.23	
7	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23 - 10.11.23	

Студент

(підпис)

Янковський Є. В.

Керівник роботи

(підпис)

Колесницький О. К.

АНОТАЦІЯ

УДК 004.8

Янковський Є. В. Інформаційна технологія виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж. Магістерська кваліфікаційна робота зі спеціальності 122 – «Комп'ютерні науки», освітня програма – «Системи штучного інтелекту». Вінниця: ВНТУ, 2023. – 105 с.

На укр. мові. Бібліогр.: 34 назв; рис.: 27; табл. 15.

Дана магістерська кваліфікаційна робота присвячена розробці програмного забезпечення для виявлення озброєних людей у відеопотоці. Були розглянуті та проаналізовані існуючі методи класифікації зображень, найбільш перспективним вибрано метод класифікації за допомогою нейронних мереж. Було досліджено різні типи згорткових нейронних мереж призначених для роботи в реальному часі, в результаті досліджень обрано мережу типу YOLO як найбільш оптимальний тип мережі. Розроблена структура нейронної мережі містить 295 шарів, вхідний шар мережі приймає зображення розміром 640x640 пікселів. Для розробки програми було використано мову програмування Python, також було використано наступні бібліотеки: PyTorch – для навчання нейронної мережі, OpenCV – для роботи з зображеннями та машинним зором, PyQt5 – для розробки графічного інтерфейсу. Достовірність виявлення озброєних людей в розробленої програми вища на 4,81% ніж в аналога.

Графічна частина складається з 5 плакатів.

У економічному розділі розраховано суму витрат на розробку та виготовлення нового технічного рішення, яка складає 490465,28 гривень, спрогнозовано орієнтовану величину витрат по кожній з статей витрат, розраховано чистий прибуток, термін окупності витрат для виробника 1,08 роки та економічний ефект для споживача при використанні даної розробки.

Ключові слова: виявлення, відеопотік, зброя, згорткова нейронна мережа.

ABSTRACT

Yankovsky E. V. Information technology for detecting armed people in a video stream using convolutional neural networks. Master's qualification thesis on specialty 122 – «Artificial intelligence systems», educational program - computer science. Vinnytsia: VNTU, 2023. – 105 p.

In Ukrainian speech Bibliography: 34 titles; Fig.: 27; table 15.

This master's thesis is devoted to the development of software for detecting armed people in a video stream. The existing methods of image classification were considered and analyzed, the classification method using neural networks was chosen as the most promising. Different types of convolutional neural networks designed for real-time operation were investigated, as a result of the research, the YOLO type network was chosen as the most optimal type of network. The developed structure of the neural network contains 295 layers, the input layer of the network accepts images with a size of 640x640 pixels. The Python programming language was used to develop the program, and the following libraries were also used: PyTorch - for training a neural network, OpenCV - for working with images and machine vision, PyQt5 - for developing a graphical interface. The reliability of the detection of armed people in the developed program is 4.81% higher than in the analogue.

The graphic part consists of 5 posters.

In the economic section, the amount of costs for the development and production of a new technical solution is calculated, which is 490465.28 hryvnias, the estimated amount of costs for each of the cost items is predicted, the net profit is calculated, the payback period for the manufacturer is 1.08 years, and the economic effect for the consumer when using this developments.

Keywords: detection, video stream, weapon, convolutional neural network.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗПІЗНАВАННЯ ОЗБРОЄНИХ ЛЮДЕЙ..	7
1.1 Постановка задачі.....	7
1.2 Класифікація зображень у відеопотоках за допомогою згорткових нейронних мереж	8
1.2.1 Логістична регресія	8
1.2.2 Дерево рішень.....	10
1.2.3 Метод опорних векторів	12
1.2.4 Класифікатор Баєса	13
1.2.5 Нейронні мережі.....	14
1.3 Вибір аналогу розпізнавання озброєних людей.....	15
1.4 Висновок до розділу 1	17
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ОЗБРОЄНИХ ЛЮДЕЙ У ВІДЕОПОТОЦІ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ	18
2.1 Структура процесів обробки інформаційної технології виявлення озброєних людей у відеопотоці.....	18
2.2 Вибір архітектури згорткової нейронної мережі.....	19
2.2.1 Faster R-CNN	23
2.2.2 RetinaNet.....	28
2.2.3 EfficientDet	31
2.2.4 CenterNet.....	33
2.2.5 YOLO	35
2.3 Розробка структури згорткової нейронної мережі	40
2.4 Розробка алгоритму роботи інформаційної технології виявлення озброєних людей у відеопотоці.....	41
2.5 Висновок до розділу 2	44

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВІЯВЛЕННЯ ОЗБРОЄНИХ ЛЮДЕЙ У ВІДЕОПОТОЦІ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ	45
3.1 Обґрунтування вибору набору даних для навчання і тестування.....	45
3.2 Обґрунтування вибору мови програмування і середовища розробки.....	47
3.3 Обґрунтування вибору фреймворків та бібліотек	49
3.4 Навчання мережі	52
3.5 Висновок до розділу 3	56
4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВІЯВЛЕННЯ ОЗБРОЄНИХ ЛЮДЕЙ У ВІДЕОПОТОЦІ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ.....	57
4.1 Тестування інформаційної технології виявлення озброєних людей у відеопотоці.....	57
4.2 Аналіз результатів тестування інформаційної технології виявлення озброєних людей у відеопотоці.....	61
4.3 Висновок до розділу 4	65
5 ЕКОНОМІЧНА ЧАСТИНА	66
5.1 Оцінювання науково-технічного рівня та комерційного потенціалу розробки	66
5.2 Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи	70
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	78
5.4 Висновок до розділу 5	84
ВИСНОВКИ.....	85
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	91
Додаток Б (обов'язковий) Лістинг програми	91
Додаток В (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА.....	96
Додаток Г (довідниковий) Інструкція користувача	102

ВСТУП

Актуальність теми. Магістерська робота присвячена розв'язанню актуальної наукової задачі реалізації інформаційної технології виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж.

У зв'язку з ситуацією, що склалась на території України, а також поширення вогнепальної зброї серед цивільного населення, виникає необхідність в посиленні контролю за обігом зброї. Найпоширенішим засобом забезпечення контролю є системи відеоспостереження, однак для таких систем актуальним є наявність модулів побудованих на основі нейронних мереж для підвищення якості контролю.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету спеціальності 122 «Комп'ютерні науки» та плану наукової та навчально-методичної роботи кафедри.

Метою магістерської роботи є підвищення достовірності та розширення функціональних можливостей розпізнавання озброєних людей за рахунок використання згорткових нейронних мереж.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз проблеми виявлення озброєних людей у відеопотоках;
- розглянути існуючі методи виявлення озброєних людей у відеопотоках/зображеннях;
- провести аналіз вибору та розробку моделі нейромережі;
- розробити структуру та алгоритм інформаційної технології;
- виконати програмну реалізацію запропонованого інформаційної технології;
- провести тестування та аналіз отриманих результатів.

Об'єктом дослідження є процес комп'ютеризованого виявлення озброєних людей у відеопотоці.

Предметом дослідження є інформаційна технологія та програмні засоби нейромережевого виявлення озброєних людей у відеопотоці та достовірність їх роботи.

Методи дослідження. У роботі використано такі методи наукових досліджень: метод системного аналізу для аналізу структури інформаційної системи; теорія нейронних мереж для реалізації інформаційної технології розпізнавання озброєних людей; методи розпізнавання в відеопотоках для розробки процесу обробки відеопотоків в інформаційній системі.

Наукова новизна отриманих результатів полягає в наступному:

- 1) Удосконалено інформаційну технологію виявлення озброєних людей у відеопотоці за рахунок використання модифікованої структури згорткової нейронної мережі YOLO, що дозволило підвищити достовірність розпізнавання озброєних людей та розширити кількість класів зброї для розпізнавання.

Практичне значення одержаних результатів полягає в наступному:

1. Розроблено новий спосіб виявлення озброєних людей у відеопотоці;
2. Розроблено алгоритм навчання згорткової нейронної мережі для виявлення озброєних людей;
3. Розроблено алгоритм функціонування згорткової нейронної мережі для виявлення озброєних людей;
4. Розроблено програмний засіб виявлення озброєних людей у відеопотоці з використанням згорткових нейронних мереж.

Розроблені алгоритми можуть бути впроваджені в початковий процес як лекції на тему «Нейромережевий метод виявлення озброєних людей у відеопотоці за допомогою згорткових нейронних мереж» дисципліни «Нейромережеві методи штучного інтелекту».

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та

збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У роботах, опублікованих у співавторстві, автору належать такі результати: - дослідження застосування згорткових нейронних мереж для розпізнавання озброєних людей; - експериментальні дослідження нейромережевих методів виявлення зброї; - розробка алгоритму виявлення озброєних людей за допомогою згорткових нейронних мереж; - розробка системи виявлення озброєних людей відповідно до розробленого алгоритму; - описані результати проведеного дослідження та тестування.

Апробація результатів роботи. Основні результати досліджень апробовано на LI Науково-технічна конференція підрозділів Вінницького національного технічного університету (31.05.2023 року, м. Вінниця, ©)

Публікації. За результатами досліджень опубліковано одні тези доповіді на науково-технічній конференції [1] , подано статтю до журналу ОЕІЕТ та подано заявку на отримання а.с. на твір (програму).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗПІЗНАВАННЯ ОЗБРОЄНИХ ЛЮДЕЙ

1.1 Постановка задачі

Задачею магістерської кваліфікаційної роботи є підвищення достовірності та розширення функціональних можливостей розпізнавання озброєних людей за зображеннями отриманими з засобів відеоспостереження з використанням ЗНМ для вирішення поставленої задачі.

Для вирішення поставленої задачі необхідно виконати наступну послідовність задач:

- визначити завдання, яке виконуватиме нейронна мережа;
- визначити обмеження розв'язуваної задачі (швидкість, точність, відповіді);
- визначити вхідні (тип, розмір, формат) та вихідні дані (кількість класів);
- визначити оптимальну архітектуру нейронної мережі необхідну для вирішення поставленої задачі;
- розробити алгоритм розбиття відеопотоку на масиви зображень для їх подальшої обробки;
- провести порівняння результатів розробленої програми з результатами аналогів.

Задачею розв'язання нейронною мережею є класифікація зображень, а саме виявлення у вхідному відеопотоці наявності зброї. Накладені обмеження на мережу – швидкість розпізнавання – не менше 30 кадрів/сек при використанні графічного процесора, точність розпізнавання – не менше 90%. Вхідні дані представляють зображення в форматі .jpg й мають розмір 640x640 пікселів. На виході мають бути чотири класи, що визначатимуть виявлена зброя на представленому відео чи ні.

1.2 Класифікація зображень у відеопотоках за допомогою згорткових нейронних мереж

Задача класифікації зображень у відеопотоках – це процес визначення категорії або категорій, до яких належать об’єкти або явища, що зображені на відео. Ця область відноситься до області машинного навчання і використовується в багатьох сферах, таких як нагляд за безпекою, аналіз спортивних ігор, моніторинг виробничих процесів та ін.

Основні методи класифікації:

- логістична регресія;
- дерево рішень;
- метод опорних векторів;
- класифікатор Баєса;
- нейронні мережі.

1.2.1 Логістична регресія

Логістична регресія – один з найпопулярніших алгоритмів машинного навчання, який підпадає під техніку наглядного навчання. Він використовується для прогнозування категоріальної залежної змінної за допомогою заданого набору незалежних змінних [2].

Логістична регресія передбачає вихід категоріальної залежної змінної. Тому результат повинен мати категоричне або дискретне значення. Це може бути або так, або ні, 0 або 1, істина або хибність тощо, але замість того, щоб дати точне значення як 0 та 1, воно дає імовірнісні значення, які лежать від 0 до 1.

Логістична регресія багато в чому схожа з лінійною регресією, за винятком того, як вони використовуються. Лінійна регресія використовується для вирішення задач регресії, тоді як логістична регресія використовується для вирішення задач класифікації.

У логістичній регресії, замість лінії регресії, ми підходимо до логічної функції у формі «S», яка передбачає два максимальні значення (0 або 1).

Крива з логістичної функції вказує на ймовірність того, що клітини ракові чи ні, миша страждає ожирінням чи ні, залежно від ваги тощо.

Логістична регресія є важливим алгоритмом машинного навчання, оскільки вона має можливість забезпечувати ймовірності та класифікувати нові дані, використовуючи безперервні та дискретні набори даних.

Логістична регресія може бути використана для класифікації спостережень з використанням різних типів даних і може легко визначити найбільш ефективні змінні, які використовуються для класифікації.

Діаграму логічної регресії зображено на рисунку 1.1.

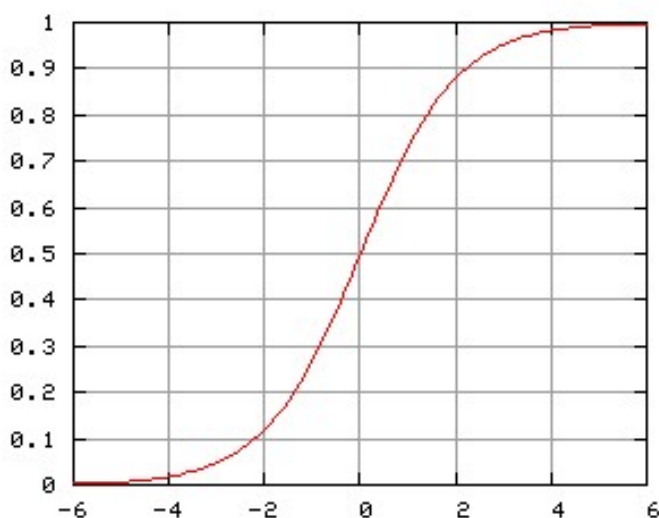


Рисунок 1.1 – Діаграма логічної регресії

До переваг логічної регресії можна віднести отримання кількісної оцінки ймовірності в у явному вигляді, гарна точність для простих наборів даних, а також робота з обмеженими обчислювальними ресурсами.

До недоліків логічної регресії можна віднести неточність при описі змінних, неможливість застосування коли кількість ознак більша за кількість спостережень, а також неможливість вирішення нелінійних проблем.

1.2.2 Дерево рішень

Дерево рішень – це наглядова техніка навчання, яка може бути використана як для задач класифікації, так і для регресії, але переважно вона є кращою для вирішення проблем класифікації. Це деревно-структурований класифікатор, де внутрішні вузли представляють особливості набору даних, гілки представляють правила прийняття рішень, а кожен листовий вузол представляє результат[3].

Процес роботи з деревом рішень включає наступні кроки:

1. Вибір атрибуту;
2. Вибір критерію зупинки навчання;
3. Вибір критерію відсікання гілок;
4. Оцінка точності та надійності побудованого дерева.

Необхідно обрати атрибут, відповідно за яким буде формуватись правило для розбиття у вузлі дерева. Існує загальне правило: завдяки атрибуту потрібно розбити множину спостережень у вузлі таким чином, щоб результуючі підмножини містили приклади з однаковими мітками класу (або є близькими до даного твердження). Інакше кажучи, кількість об'єктів інших класів у кожній множині повинна бути мінімальною. Для виконання цього твердження застосовуються різноманітні критерії, серед них найпопулярнішими є теоретико-інформаційний та статистичний.

Згідно з теорією процес навчання дерева рішень триває до тих пір, поки не буде отримано підмножин, що містять екземпляри одного класу. Цілком ймовірно, що дерево, яке було отримано внаслідок застосування подібного алгоритму, буде створено лист для кожного прикладу. Розглянутий підхід є неприпустимим, оскільки отримане дерево не несе жодної користі та є неефективним. Існує проблема перенавчання, яка характеризується точним розпізнаванням прикладів, що містяться у навчальному наборі, і повна некомпетентність у роботі з новими тестовими даними.

Примусова зупинка подальшої побудови дерева є рішенням проблеми перенавчання. Існує кілька варіантів реалізації цього підходу [4]:

1. Рання зупинка – припинення подальшої побудови дерева після виконання певного критерію, наприклад, достатній відсоток правильно розпізнаних екземплярів.
2. Обмеження глибини дерева.
3. Фіксована мінімальна кількість прикладів у одному вузлі. Це дозволить уникнути створення малозначущих правил.

Буде побудовано складне дерево з великою кількістю та листків, з яким важко працювати у подальшому та інтерпретувати результат, у випадку якщо ріст дерева не обмежити. Очевидно, що простіше організувати роботу з деревом з меншою кількістю вузлів, яким відповідає більша кількість прикладів. Пропонується альтернативний варіант, який при адекватній глибині дерева надає прийнятний рівень помилок розпізнавання, тобто існує баланс між складністю та точністю отриманої моделі.

Реалізувати розглянутий варіант можливо наступним чином [4]:

1. Побудувати дерево.
2. Визначити показники відносної точності моделі (відношення правильно розпізнаних прикладів до загального числа прикладів) та абсолютну помилку (кількість неправильно розпізнаних прикладів).
3. Видалити вузли та листки, відсікання яких не призведе до значного зменшення точності моделі.

До переваг дерева рішень можна віднести просту реалізацію та ефективне навчання, хороша точність для простих наборів даних, а також низьке споживання ресурсів.

До недоліків дерева рішень можна віднести велика кількість шарів при великій кількості рішень, неможливість розв'язання нелінійних проблем, застосовується лише для дискретних функцій.

1.2.3 Метод опорних векторів

Використання векторної машини або SVM є одним з найпопулярніших алгоритмів вирішення проблем класифікації в машинному навчанні [5].

Метою алгоритму SVM є створення найкращої лінії або межі рішення, яка може розділити n -вимірний простір на класи, щоб ми могли легко поставити нову точку даних у правильну категорію в майбутньому. Ця межа найкращого рішення називається гіперплощиною.

SVM вибирає крайні точки/вектори, які допомагають у створенні гіперплощини. Ці крайні випадки називаються векторами підтримки, і тому алгоритм називається машиною підтримки вектора. Розглянемо діаграму наведену на рис. 1.2, на якій є дві різні категорії, які класифікуються за кордоном.

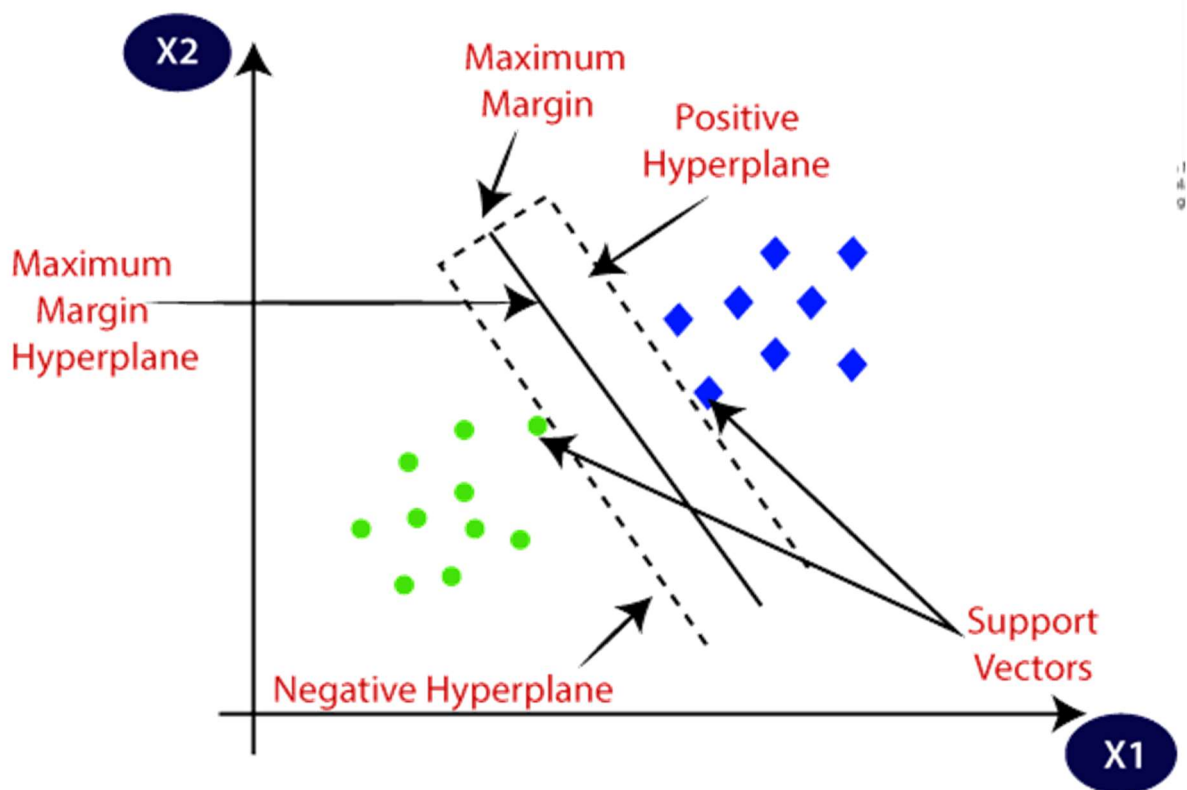


Рисунок 1.2 – Діаграма SVM

Гіперплощина: Може бути кілька ліній/кордонів рішень для відокремлення класів у n -вимірному просторі, але нам потрібно з'ясувати найкращу межу

прийняття рішень, яка допомагає класифікувати точки даних. Ця найкраща межа відома як гіперплощина SVM.

Розміри гіперплощини залежать від об'єктів, наявних у наборі даних, а це означає, що якщо є 2 об'єкти (як показано на зображенні), то гіперплощина буде прямою. А якщо є 3 ознаки, то гіперплощина буде двовимірною площиною.

Ми завжди створюємо гіперплощину з максимальним запасом, що означає максимальну відстань між точками даних. Точки даних або вектори, які є найближчими до гіперплощини і які впливають на положення гіперплощини, називаються опорними векторами. Оскільки ці вектори підтримують гіперплощину, тому їх називають вектором підтримки.

До переваг методу опорних векторів можна віднести хорошу ефективність за обмеженого набору даних, регуляризація параметра для уникнення перенавання, а також можливість хорошого визначення для нелінійних проблем.

До недоліків методу опорних векторів можна віднести складність обчислень для великих наборів даних, необхідність налаштування параметрів регуляції та ядра, що ускладнює оптимізацію, а також низька чутливість до шуму в даних, що зменшує ефективність SVM.

1.2.4 Класифікатор Баєса

Класифікатор Баєса використовує алгоритм заснований на теоремі Баєса і використовується для вирішення задач класифікації.

Алгоритм заснований на теоремі Баєса є одним з найпростіших та найефективніших алгоритмів класифікації, який допомагає у створенні моделей швидкого машинного навчання, які дозволяють швидко прогнозувати [6].

Це імовірнісний класифікатор, що означає, що він передбачає на основі ймовірності об'єкта. Деякі популярні приклади алгоритму наївного Байєса – фільтрація спаму, сентиментальний аналіз та класифікація статей.

Теорема Байєса також відома як Правило Байєса або закон Байєса, який використовується для визначення ймовірності гіпотези за наявності попередніх знань. Це залежить від умовної ймовірності.

Формула теореми Баєса зображена на у формулі 1.1.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \quad (1.1)$$

де A та B – події,

$P(A)$ та $P(B)$ – ймовірності A та B незалежно одна від одної,

$P(A|B)$ – ймовірність події A за умови істинності події B ,

$P(B|A)$ – ймовірність події B за умови істинності події A .

До переваг класифікатора Баєса можна віднести простоту розуміння та реалізацію класифікатора, можливість використання в режимі реального часу, так як класифікатор швидко навчається, а також багатокласове прогнозування.

До недоліків класифікатора Баєса можна віднести недосконалість роботи в ситуаціях межі категорій даних визначені нечітко.

1.2.5 Нейронні мережі

Нейронні мережі являють собою нелінійні системи, що дозволяють в багатьох випадках набагато краще класифікувати дані, ніж математичні та статистичні методи, що застосовуються на даний час. Відмінна особливість нейронних мереж полягає в тому, що вони не програмуються — не використовують ніяких правил виводу для прийняття рішення діагностики, а навчаються робити це на прикладах.

В основі сучасних глибоких нейронних мереж, як правило, лежать архітектури мереж згорткового типу. Їх ефективність і стрімкий розвиток обумовлено гібридним підходом до архітектурних рішень, розвитком методів навчання, додаткових методів захисту від перенавчання.

На відміну від багатосарових персептронів, згорткові нейронні мережі мають наступні ознаки:

1) Локальна повнозв'язність: відповідно до концепції рецептивних полів, ЗНМ використовують просторову локальність шляхом застосування схеми

42 локальної зв'язності між нейронами сусідніх шарів. Таким чином, ця архітектура забезпечує можливість навченим «фільтрам» (ядрам згортки) виробляти найсильніший відгук на просторово-локальний вхідний образ. Структура із багатьох таких шарів еквівалентна використанню нелінійного фільтру і є чутливою до більшої області піксельного простору. Таким чином, мережа спочатку створює представлення дрібних деталей входу, а потім з них збирає представлення більших областей [7];

2) Спільні ваги: у згорткових нейронних мережах кожен фільтр повторюється на всьому зоровому полі. Ці повторні вузли використовують спільну параметризацію (вектор ваг та порогів) та формують карту ознак. Це означає, що всі нейрони у заданому згортковому шарі реагують на одну й ту ж саму ознаку в межах свого рецептивного поля. Повторювання вузлів, таким чином, дозволяє виявити ознаки незалежно від їхнього положення в зоровому полі, забезпечуючи властивість інваріантності відносно зсуву [7].

Разом ці властивості дозволяють ЗНМ досягати кращого узагальнення на задачах розпізнавання зображень. Спільне використання ваг різко зменшує кількість вільних параметрів, завдяки яким навчається мережа, знижуючи вимоги до пам'яті для роботи мережі та уможливіючи тренування більших, потужніших мереж [8].

1.3 Вибір аналогу розпізнавання озброєних людей

Серед програмних реалізацій для виявлення озброєних людей можна виділити *Weapon detection*.

Програма реалізована на мові Python з використанням засобів обробки зображень бібліотеки *OpenCV*. Для розпізнавання озброєних людей використовується згорткова нейронна мережа *YOLOv5* з використанням засобів глибокого машинного навчання. Дана нейронна мережа складається з трьох частин: базова частина – представляє собою модифіковану магістральну мережу *CSPDarknet53*, шийна частина – представлена мережею агрегації шляхів (*PANet*) та головна частина – для генерації різних виводів карт функцій для досягнення

багатомасштабного прогнозування. Дана програма призначена для розпізнавання двох класів зброї «rifle» (рис. 1.3) та «pistol» (рис. 1.4).

Дана програма має хорошу швидкодію, але й такі суттєві недоліки як низька достовірність розпізнавання 93,7%, а також низькі функціональні можливості, такі як можливість розпізнавання зброї лише на зображеннях, що не дає можливості роботи з відеопотоками, а також можливість розпізнавання лише двох класів зброї.

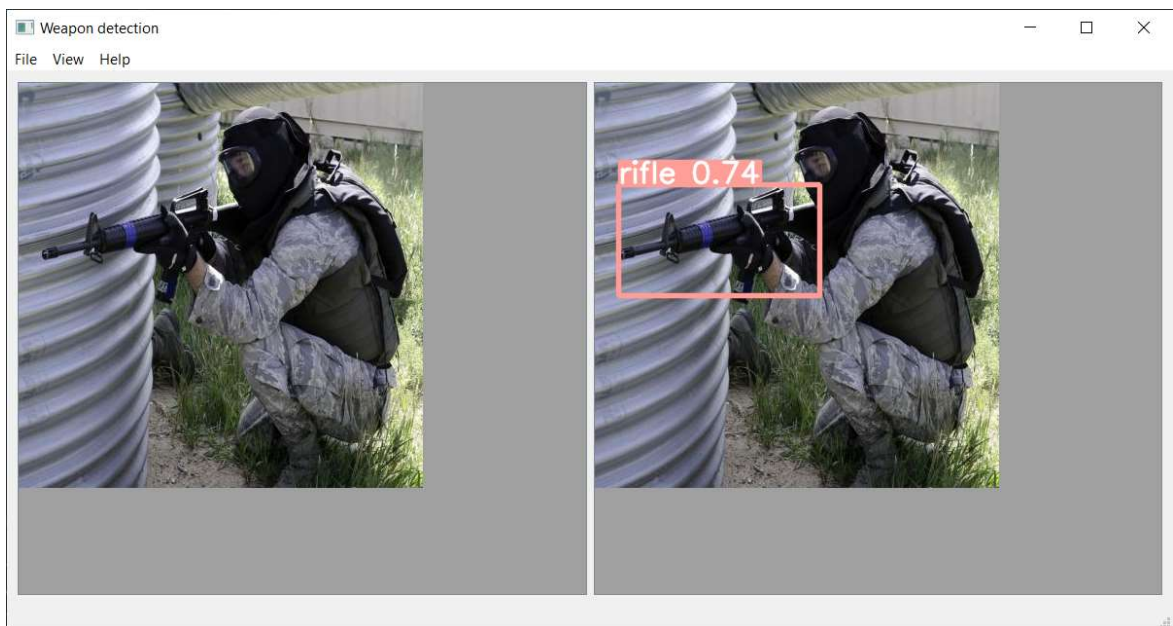


Рисунок 1.3 – Клас «rifle»



Рисунок 1.4 – Клас «pistol»

Інші програмні реалізації мають аналогічні недоліки, або вказані вище недоліки відсутні, але програмні реалізації важкодоступні через їх вартість. Тому виникає необхідність в розробці програмного засобу для виявлення озброєних людей для розширення функціональних можливостей, таких як розпізнавання в відеопотоках, збільшення кількості класів зброї для рзпізнавання, а також підвищення достовірності розпізнавання.

1.4 Висновок до розділу 1

В розділі було розглянуто постановку задачі розпізнавання озброєних людей у відеопотоці, обґрунтовано доцільність використання нейронних мереж для розпізнавання озброєних людей, зокрема згорткових нейронних мереж. Було проаналізовано існуючі програмні реалізації розпізнавання озброєних людей, обрано аналог і проаналізовано, що головним його недоліком є низькі функціональні можливості та невелика достовірність розпізнавання, що ставить мету дослідження – розширення функціональних можливостей та підвищення достовірності розпізнавання.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ОЗБРОЄНИХ ЛЮДЕЙ У ВІДЕОПОТОЦІ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ

2.1 Структура процесів обробки інформаційної технології виявлення озброєних людей у відеопотоці

Завдання роботи полягає в розробці інформаційної технології для визначення появи або відсутності у відеопотоці озброєних людей, отриманих з камер відеоспостереження або з завантажених відеофайлів. Для розробки згорткової нейронної мережі, що розпізнаватиме озброєних людей, а також її застосування на реальних даних необхідно зібрати набір даних, який буде містити зображення з різними видами зброї поділеними на класи. Кожен клас повинен містити не менше тисячі зображень для кожного класу зброї, оскільки чим більша кількість даних, на яких навчається мережа, тим краща точність її роботи. Для обробки набору даних зображень необхідно виконати наступну послідовність кроків (рис. 2.1):

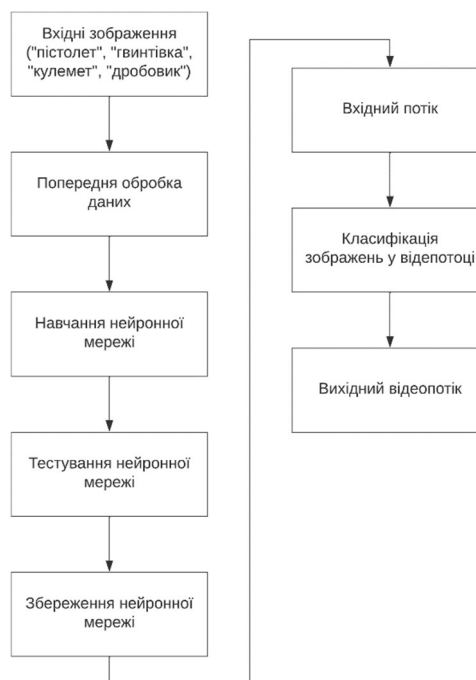


Рисунок 2.1 – Послідовність кроків обробки інформації виявлення озброєних людей

Всі вхідні зображення, що використовуються для навчання повинні пройти попередню обробку, а саме приведення їх до одного розміру і формату який сприймає нейронна мережа. Також необхідно задати оптимальну кількість епох навчання нейронної мережі, при якій виключається недостатнє навчання або перенавчання мережі, оскільки в таких випадках якість навчання падає. Навчена нейронна мережа тестується на відповідність поставленим вимогам, якщо вимоги виконані тоді мережа зберігається, інакше навчання мережі повторюється спочатку. Збережена мережа може бути використана класифікації відеопотоків з вилученням та виділенням вилучених класів.

2.2 Вибір архітектури згорткової нейронної мережі

Згорткові нейронні мережі (ЗНМ) – це підвид глибоких нейронних мереж, переважно застосовуваний до розпізнавання об’єктів та класифікації зображень. Згорткові нейронні мережі сприймають та обробляють дані у вигляді тензорів, що дозволяє працювати з даними зображень у природній формі. Кожне вхідне зображення має як параметри ширину та довжину, а також глибину, яка визначається кодуванням зображення. Найбільш поширеним із них є RGB-кодування, в якому колір кожного пікселя на зображенні кодується за допомогою значень для трьох кольорів – червоного, зеленого та синього. Такі категорії значень, що описують зображення, називаються каналами. Інший вимір тензорів утворюється в ході роботи мережі і містить карти виявлених ознак зображення. Таким чином, згорткові нейронні мережі розглядають кожне зображення як чотиривимірний масив даних. Зазвичай для кожної мережі визначається конкретна необхідна ширина і довжина зображень, хоча також існують мережі, здатні до масштабування.

Для задач розпізнавання в умовах використання великих об’ємів даних необхідно, щоб модель мала високу здатність до навчання та великий відсоток правильних припущень щодо ознак зображення. У порівнянні з традиційними нейронними мережами прямого поширення зі схожою кількістю шарів, згорткові нейронні мережі мають вищу здатність до навчання, оскільки містять набагато

менше параметрів та зв'язків. Для традиційних повнозв'язних нейронних мереж для їх правильного навчання в задачах розпізнавання образів необхідна набагато більша кількість даних, оскільки кожне вхідне зображення може мати розмірність у щонайменше кілька сотень тисяч, що вимагатиме відповідну кількість прикладів з різними значеннями для кожного виміру. [8]

Ще однією відмінністю згорткових нейронних мереж є те, що вони зазвичай не вимагають попередньої обробки даних, оскільки використовують дані зображень напряму, що дозволяє спрощувати структуру мережі, зважаючи на упорядкованість початкових даних. Водночас, незважаючи на численні переваги, їх застосування все ще вимагає значної витрати ресурсів, особливо для розв'язання задач із зображеннями з високою роздільною здатністю, тому виникає необхідність розробки згорткових нейронних мереж із високим рівнем оптимізації операцій.

Окрім застосування для задач розпізнавання та задач класифікації зображень, згорткові нейронні мережі завдяки модифікаціям широко застосовуються до задач обробки природної мови (ОПМ), зокрема, вони показали високу ефективність в задачах семантичного парсингу, а також моделюванні та класифікації речень. [8]

Структура згорткової нейронної мережі відповідає загальноприйнятій структурі нейронної мережі. Мережа складається з вхідного шару, певної кількості прихованих шарів та вихідного шару. Приховані шари зазвичай складаються зі згорткових шарів, шарів агрегування (субдискретизації), нормалізуючих та повнозв'язних шарів. Ці шари пов'язані між собою шарами з визначеними активаційними функціями. Головним елементом згорткової нейронної мережі є згорткові шари, де до даних з попереднього шару застосовується операція згортки, яка є математичною операцією, яка за допомогою дії ядра згортки k на вхідне зображення у вигляді тензора x отримує як результат карту ознак m . Кожний детектор ознак складається з деякої визначеної кількості фільтрів, кожний з яких в свою чергу є тензором параметрів з розмірністю, яка як правило дозволяє покривати лише невелику частину зображення в ширину та висоту, але при цьому обов'язково має відповідати

розмірності в глибину. Крім того, власне кількість фільтрів також може бути одним із гіперпараметрів нейронної мережі, оскільки збільшення кількості фільтрів веде до збільшення кількості потенційно виявлених ознак та закономірностей, але при цьому занадто велика їх кількість може призвести до повторень та знаходження ознак, які не є характеристичними, тобто до ускладнення та потенційного перенавчання моделі. Таким чином, кількість фільтрів відповідає кількості карт ознак, яка визначає глибину вихідних даних, а, отже, і глибину вхідних для наступного рівня мережі, що відповідно визначає кількість необхідних параметрів і на наступному кроці. В ході навчання фільтри набувають можливості визначати ознаки зображення, які й групуються в карти ознак. Кожен фільтр відповідає за створення однієї карти ознак. Карта ознак формується завдяки тому, що фільтр переміщується зображенням та визначає ознаки для сусідніх 18 рецептивних полів. Крок переміщення s може відрізнятися і є одним з параметрів моделі. Таким чином, хоча абсолютне положення кожної ознаки на зображення на карті і не зберігається, але це не має значення, оскільки відносне положення ознак одна відносно одної залишається сталим, і, таким чином, властивості зображення зберігаються. [9] Загальна структурна схема побудови згорткової нейронної мережі наведена на рис. 2.1.

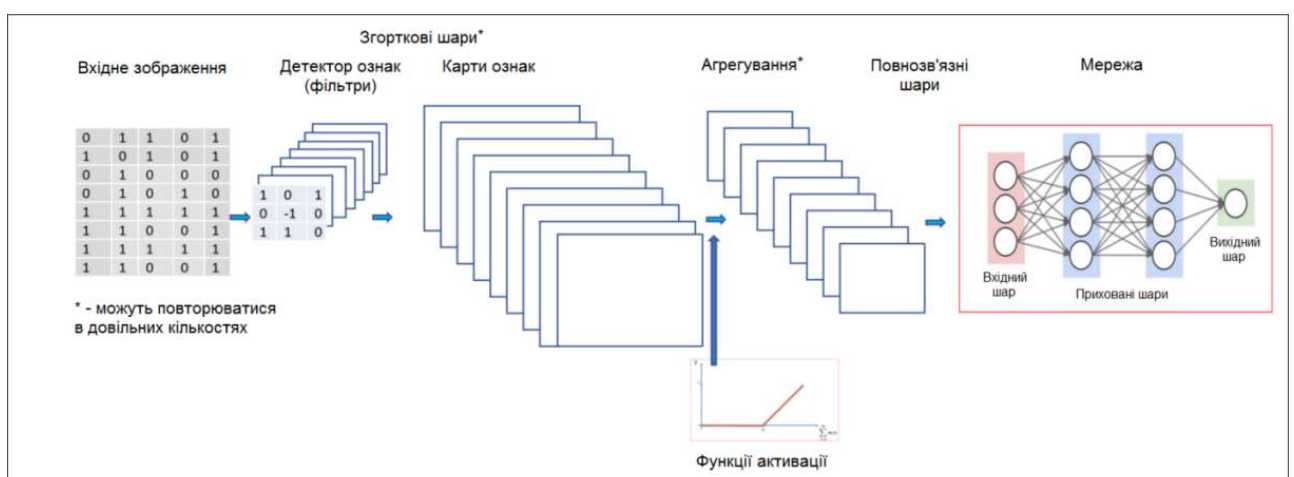


Рисунок 2.1 – Загальна структурна схема побудови згорткової нейронної мережі

Згорткові нейронні мережі здобули широку популярність в задачах розпізнавання зображень завдяки тому, що вони уникають або зменшують

головні проблеми, пов'язані з обробкою зображень в традиційних повнозв'язних мережах. Велика кількість зв'язків в таких мережах та велика кількість параметрів у кожному зображенні потребує великої кількості вхідних прикладів, а у випадку їх нестачі швидко виникає перенавчання 16 мережі, тобто, модель стає занадто складною та навчається на нехарактеристичних другорядних ознаках, знайдених у вхідних даних. Ще більшою проблемою є те, такі моделі не є стійкими до збурень або будь-яких змін у зображеннях, наприклад, до розташування об'єкта в іншій частині зображення або зміні кута погляду на нього. Крім того, такі мережі не здатні до врахування топології вхідних даних: зображення є структурованими даними, де пікселі, що розташовані близько один від одного, мають високу кореляцію, що робить необхідним знаходження локальних ознак та взаємозв'язків для ефективного визначення образів.

В згорткових нейронних мережах ці проблеми вирішуються завдяки використанню кількох ідей, найголовнішою з яких є локальна обробка значень на прихованих рівнях. Кожен з рівнів поділений на частини, кожна з яких сприймає лише дані попереднього рівня, розташовані в певній невеликій області. Такі області називаються локальними рецептивними полями. Використовуючи локальні рецептивні поля, мережа може визначати найпростіші елементи та ознаки зображення, як повороти або грані між ділянками зображення. Іншою ідеєю в основі згорткових нейронних мереж є спільні ваги, використання яких зменшує чутливість мережі до змін положення об'єктів на зображенні, незначних поворотів та його спотворення: так, детектор певної простої ознаки зображення, що використовується на певній його ділянці, може визначати аналогічні ознаки і на іншій його ділянці. Завдяки цьому детектори для локальних рецептивних полів на різних ділянках будуть мати однакові ваги, і, як наслідок, в різних частинах зображення для схожих ознак виконуються однакові перетворення.

Крім того, важливим елементом є операція субдискретизації, яка дозволяє зменшувати загальну розмірність даних, при цьому зменшуючи їх чутливість до значних збурень. Зазвичай субдискретизація в архітектурі мережі чергується із операціями згортки, таким чином поступово зменшуючи розмірність даних та збільшуючи кількість карт ознак.

Детектори на основі згорткових нейронних мереж поділяються на двоетапні та одноетапні. Двоетапні детектори використовують регіональний підхід, який спочатку ідентифікує підмножини регіонів зображення, які можуть містити об'єкт і виконує класифікацію об'єктів на цих регіонах. [10] Завдяки такому підходу двоетапні детектори забезпечують дуже точні результати виявлення об'єктів, однак дані детектори, за винятком Faster R-CNN, працюють дуже повільно і не можуть працювати в режимі реального часу. Одноетапні детектори виконують класифікацію об'єктів за допомогою обмежувальних прямокутників і потім прогнози декодуються для створення остаточних обмежувальних прямокутників для розпізнаних об'єктів. [11] Незважаючи на менші показники точності в порівнянні з двоетапними детекторами, одноетапні детектори працюють набагато швидше і можуть працювати в режимі реального часу.

Оскільки, ми з'ясували, що найкращим для вирішення поставленої задачі є застосування згорткової нейронної мережі, нам необхідно вибрати архітектуру згорткової нейромережі, що найбільше відповідає нашим вимогам, а саме забезпечувати хорошу точність виявлення об'єктів з можливістю працювати в режимі реального часу.

В даному розділі будуть розглянуті наступні моделі згорткових нейронних мереж:

- Faster R-CNN;
- RetinaNet;
- EfficientDet;
- CenterNet;
- YOLO.

2.2.1 Faster R-CNN

Faster-RCNN [12] – найсучасніша архітектура для виявлення об'єктів сімейства R-CNN. Дана мережа поєднує в собі переваги глибинного навчання,

згорткових нейронних мереж (CNN) і мереж регіональних пропозицій (RPN) у єдину мережу яка покращує швидкість та точність моделі.

Архітектура мережі Faster RCNN складається з двох компонентів (рис. 2.2):

- мережі регіональних пропозицій (RPN);
- детектор Fast R-CNN.

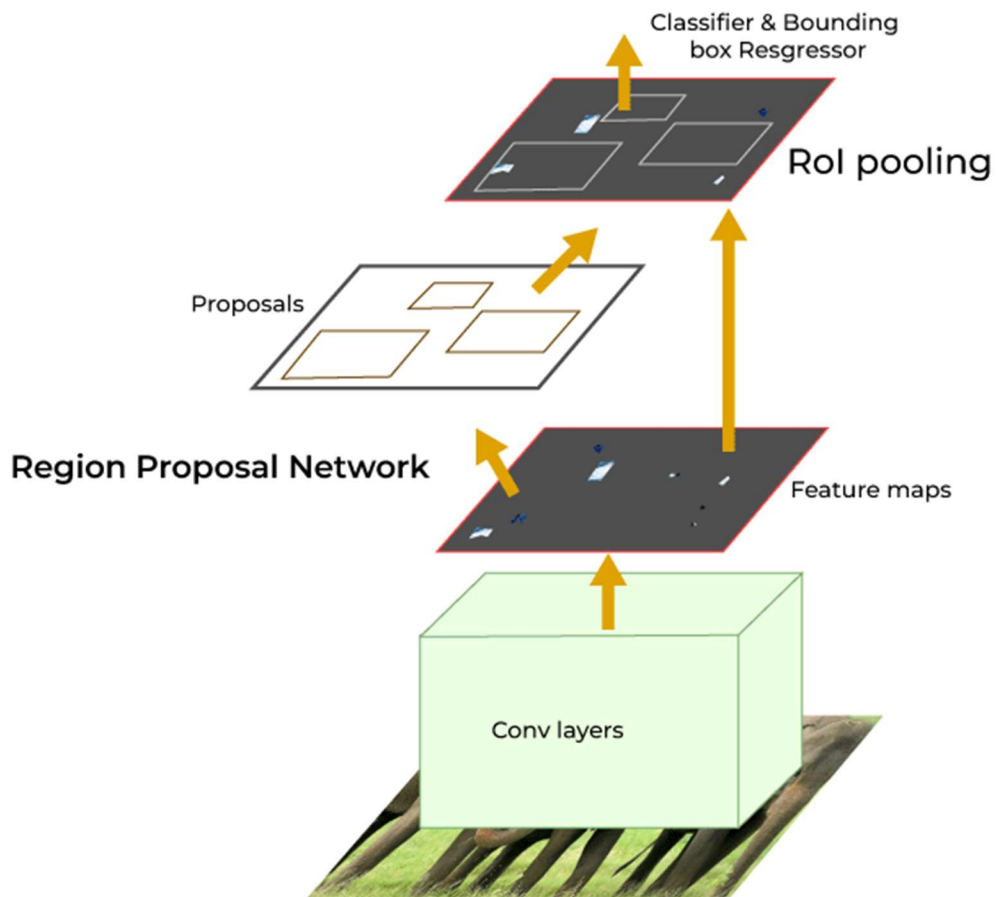


Рисунок 2.2 – Архітектура Faster RCNN

Мережа регіональних пропозицій (RPN) (рис. 2.3) є важливим компонентом Faster R-CNN. Він відповідає за створення можливих регіонів інтересу (пропозицій регіонів) на зображеннях, які можуть містити об'єкти. Він використовує концепцію механізму уваги в нейронних мережах, які інструктують подальший детектор Fast R-CNN, де шукати об'єкти на зображенні.

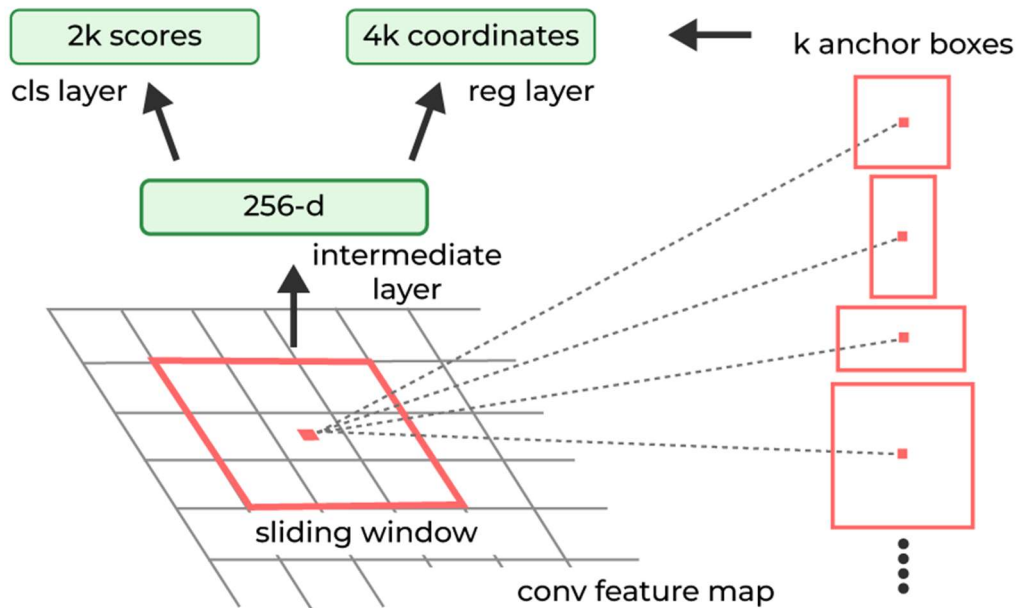


Рисунок 2.3 – Мережа регіональних пропозицій

Карти функцій, отримані з магістралі CNN, використовуються RPN. На цих картах об'єктів RPN використовує підхід ковзного вікна з прив'язними блоками різних масштабів і форм для визначення потенційних положень об'єктів. Мережа вдосконалює ці прив'язки під час навчання, щоб краще відповідати фактичним положенням і розмірам об'єктів. Для кожного якоря RPN передбачає два параметри: імовірність того, що прив'язка містить об'єкт («оцінка об'єкта»); коригування координат прив'язки відповідно до фактичної форми об'єкта.

Коли генерується велика кількість регіональних пропозицій, багато з них можуть збігатися і відповідати одному об'єкту. Тут Non-Maximum Suppression (NMS) використовується для ранжування блоків прив'язки на основі ймовірності їх об'єктності та вибору перших N блоків прив'язки з найвищими оцінками. NMS гарантує, що остаточно відібрані пропозиції є точними та не збігаються. Ці вибрані анкерні коробки розглядаються як можливі регіональні пропозиції.

Детектор Fast R-CNN є критично важливим компонентом архітектури Faster R-CNN, відповідальним за виявлення об'єктів у пропозиціях регіону, запропонованих RPN.

Робота Fast R-CNN включає в себе наступні кроки:

1. Об'єднання регіонів інтересів (RoI): першим кроком є використання пропозицій регіону, запропонованих RPN, і застосування об'єднання ROI. Об'єднання регіонів інтересів використовується для перетворення пропозицій регіонів змінного розміру RPN у карти функцій фіксованого розміру, які можуть подаватися на наступні рівні мережі. Об'єднання ROI розділяє кожну пропозицію регіону на сітку клітинок однакового розміру, а потім застосовує максимальне об'єднання в кожній клітинці. Ця процедура генерує карту функцій фіксованого розміру для кожної пропозиції регіону, яка може бути додатково оброблена мережею (рис. 2.4).

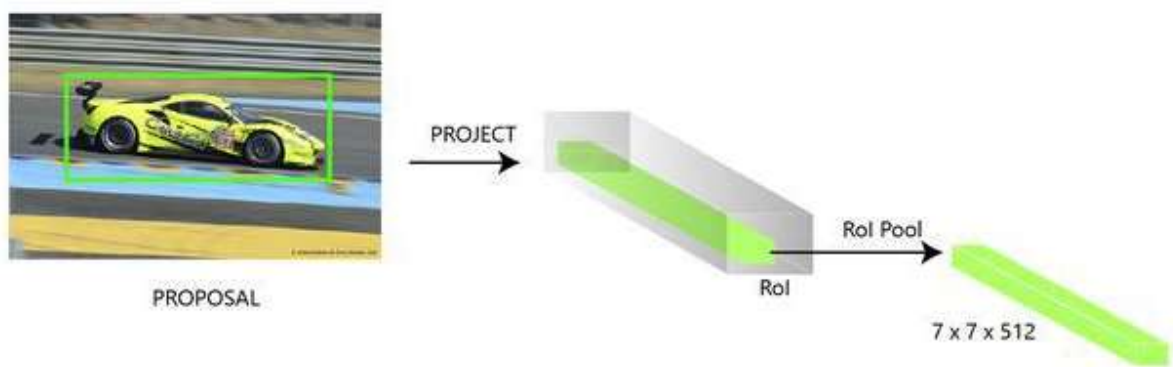


Рисунок 2.4 – Об'єднання регіонів інтересів

2. Вилучення функцій: об'єднані карти функцій RoI надходять у магістраль CNN (ту саму, що використовується в RPN для вилучення функцій), щоб витягти значущі функції, які фіксують інформацію про об'єкт. Він черпає ієрархічні особливості з пропозицій регіону. Ці функції зберігають просторову інформацію, абстрагуючись від низькорівневих деталей, дозволяючи мережі зрозуміти зміст запропонованих регіонів.

3. Повністю з'єднанні шари: регіони, об'єднані RoI та виділені функції, потім проходять через серію повністю з'єднаних шарів. Ці рівні відповідають за класифікацію об'єктів і завдання регресії обмежувальної рамки. Перший рівень – це рівень softmax із $N+1$ вихідними параметрами (N – кількість міток класу та фону), який передбачає об'єкти в пропозиції регіону. Другий шар – це шар

регресії обмежувальної рамки, який має $4 \cdot N$ вихідних параметрів. Цей шар регресує розташування обмежувальної рамки об'єкта на зображенні (рис. 2.5).

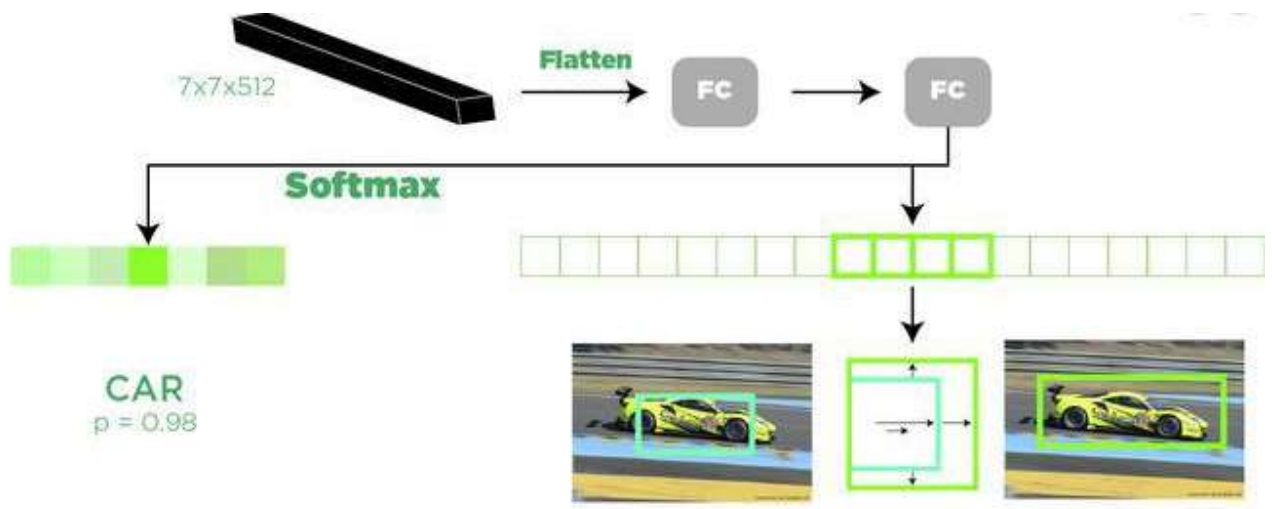


Рисунок 2.5 – Повністю з'єднані шари

4. Багатозадачна функція втрати: багатозадачна функція втрати, яка поєднує класифікацію та регресію, використовується детектором Fast R-CNN. Втрата класифікації обчислює різницю між очікуваною та справжньою ймовірностями класу. Втрата регресії обчислює різницю між очікуваними та фактичними налаштуваннями обмежувальної рамки.

$$L(p_i, t_i, v_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, v_i) \quad (2.1)$$

де N_{cls} – кількість RoI, що використовується для класифікації;

N_{reg} – кількість RoI, що використовується для регресії обмежувальної рамки;

p_i – прогнозована ймовірність класифікації i -го RoI;

p_i^* – індикатор істинності (1 або 0) для i -го RoI, який є об'єктом переднього плану або фону;

t_i – представляє параметри обмежувального прямокутника для i -го RoI;

v_i – представляє передбачувані коригування обмежувальної рамки для i -го RoI;

L_{cls} – функція класифікаційних втрат, яка часто обчислюється з використанням крос-ентропійних втрат;

L_{reg} – регресійна функція втрат, яка часто обчислюється з використанням плавних втрат L1;

λ – балансуєчий параметр, який контролює компроміс між двома компонентами збитку.

5. Постобробка: після того, як мережа прогнозує ймовірності класу та зміни обмежувальної рамки, остаточні результати виявлення уточнюються за допомогою процедури постобробки. На цьому етапі немаксимальне придушення (NMS) використовується для зменшення надлишкових виявлень, зберігаючи при цьому найбільш впевнені виявлення, що не перекриваються.

Серед переваг даної моделі є висока точність визначення об'єктів, а також єдність процесів класифікації об'єктів. Однак дана мережа є складною в реалізації, а процес її навчання займає велику кількість ресурсів.

2.2.2 RetinaNet

RetinaNet [13] – одна з найкращих одноетапних моделей виявлення об'єктів, яка добре працює з щільними та малими об'єктами. RetinaNet сформована удосконаленням одноетапних моделей виявлення об'єктів Feature Pyramid Networks (FPN) [14] і Focal Loss [15].

Традиційно в комп'ютерному зорі для виявлення об'єктів із різними масштабами на зображенні використовували піраміди зображень із представленими зображеннями. Піраміди представлених зображень – це піраміди функцій, побудовані на пірамідах зображень. Це означає, що можна взяти зображення та розділити його на зображення з нижчою роздільною здатністю та меншим розміром (таким чином, утворюючи піраміду). Потім з кожного шару піраміди витягуються вручну сконструйовані об'єкти для виявлення об'єктів [14]. Це робить піраміду незмінною за масштабом. Але цей процес потребує обчислень і пам'яті.

З появою глибокого навчання ці ручні функції були замінені CNN. Пізніше сама піраміда була похідною від властивої пірамідальної ієрархічної структури CNN. В архітектурі CNN вихідний розмір карт функцій зменшується після кожного наступного блоку згорткових операцій і утворює пірамідальну структуру.

Існують різні архітектури, які використовують структуру піраміди (рис. 2.6). (a) Піраміда представлених зображень, як ми вже обговорювали, вимагає інтенсивних обчислень. (b) Для швидшого виявлення використовувалися карти окремих (масштабних) функцій. Незважаючи на те, що вони надійні та швидкі, піраміди все одно потрібні для отримання найточніших результатів [14]. (c) Пірамідальна ієрархія об'єктів використовується в таких моделях, як Single Shot detector, але вона не використовує повторно багатомасштабні карти об'єктів з різних шарів. (d) Feature Pyramid Network (FPN) компенсує недоліки цих варіантів. FPN створює архітектуру з багатою семантикою на всіх рівнях, оскільки вона поєднує семантично сильні функції низької роздільної здатності з семантично слабкими функціями високої роздільної здатності [14].

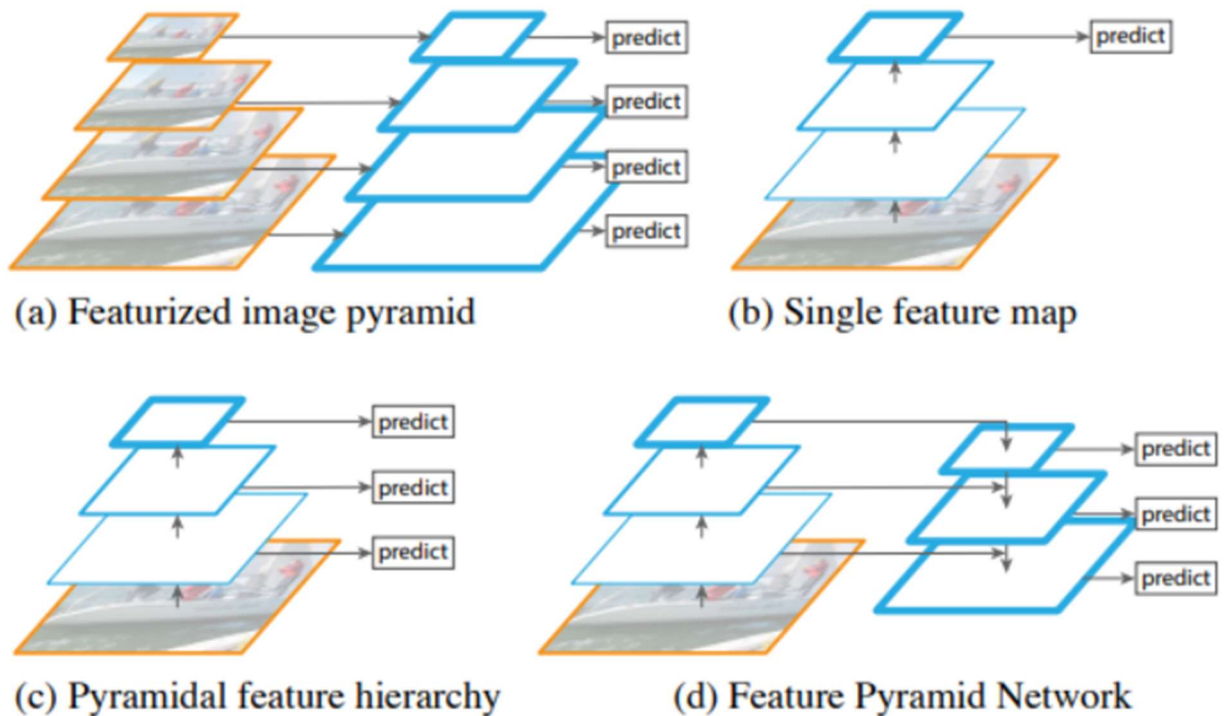


Рисунок 2.6 – Різні типи пірамідних архітектур

Архітектура мережі RetinaNet складається з чотирьох основних компонентів (рис. 2.7):

a) Шлях «знизу вгору» – магістральна мережа, яка обчислює карти функцій у різних масштабах, незалежно від розміру вхідного зображення чи магістралі;

b) Шлях «зверху вниз» і бічні зв'язки. Шлях «зверху вниз» підвищує дискретизацію просторово більш грубих карт характеристик із вищих рівнів піраміди, а бічні зв'язки об'єднують шари «згори вниз» і «знизу вгору» однакового просторового розміру;

c) Класифікаційна підмережа – вона передбачає ймовірність присутності об'єкта в кожному просторовому місці для кожного блоку прив'язки та класу об'єктів;

d) Підмережа регресії – регресує зсув для обмежувальних рамок від опорних рамок для кожного наземного об'єкта.

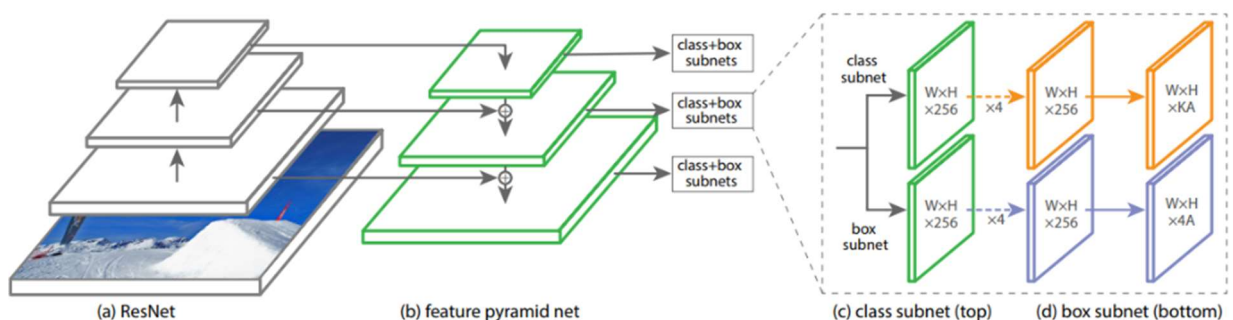


Рисунок 2.7 – Архітектура мережі RetinaNet

Фокальна втрата (Focal Loss) є вдосконаленням у порівнянні з втратою перехресної ентропії (Cross-Entropy Loss) і введена для вирішення проблеми дисбалансу класів у одноетапних моделях виявлення об'єктів. Одноетапні моделі страждають від надзвичайної проблеми дисбалансу класів переднього плану та заднього плану через щільну вибірку блоків прив'язки (можливих місць розташування об'єктів) [15]. У RetinaNet на кожному шарі піраміди можуть бути тисячі блоків прив'язки. Лише деякі з них будуть призначені для наземного об'єкта, тоді як переважна більшість буде фоновим класом. Хоча ці прості

приклади (виявлення з високою ймовірністю) призводять до малих значень втрат, разом можуть перевантажити модель. Focal Loss зменшує внесок втрат від простих прикладів і підвищує важливість виправлення неправильно класифікованих прикладів.

Серед переваг мережі є високі показники продуктивності та швидкості, а також вирішення проблеми незбалансованих даних, що часто виникає при вирішенні задач виявлення об'єктів, однак дана модель є складною в реалізації і може бути неоптимальною для вирішення деяких задач.

2.2.3 EfficientDet

EfficientDet [16] – тип моделі виявлення об'єктів, що використовує декілька оптимізацій та налаштувань основи, таких як ViFPN, а також метод складного масштабування, який рівномірно масштабує роздільну здатність, глибину та ширину для всіх основ, мереж ознак та мереж прогнозування блоків/класів одночасно.

Архітектура EfficientDet (рис. 2.8) відповідає парадигмі одноетапних детекторів. Вони використовують EfficientNet, навчену на основі ImageNet, як базову мережу, і запропонувану ViFPN, яка слугує мережею ознак, що бере ознаки 3-7 рівнів (P3, P4, P5, P6, P7) з базової мережі і багаторазово застосовує двонаправлене злиття ознак зверху-вниз і знизу-вгору. Ці об'єднані ознаки подаються на мережу класів та боксів для створення прогнозів класів об'єктів та обмежувальних боксів відповідно [17].

Магістральна мережа використовує ті ж самі коефіцієнти масштабування ширини/глибини EfficientNet-B0-B6, щоб можна було використовувати попередньо підготовлені контрольні точки ImageNet.

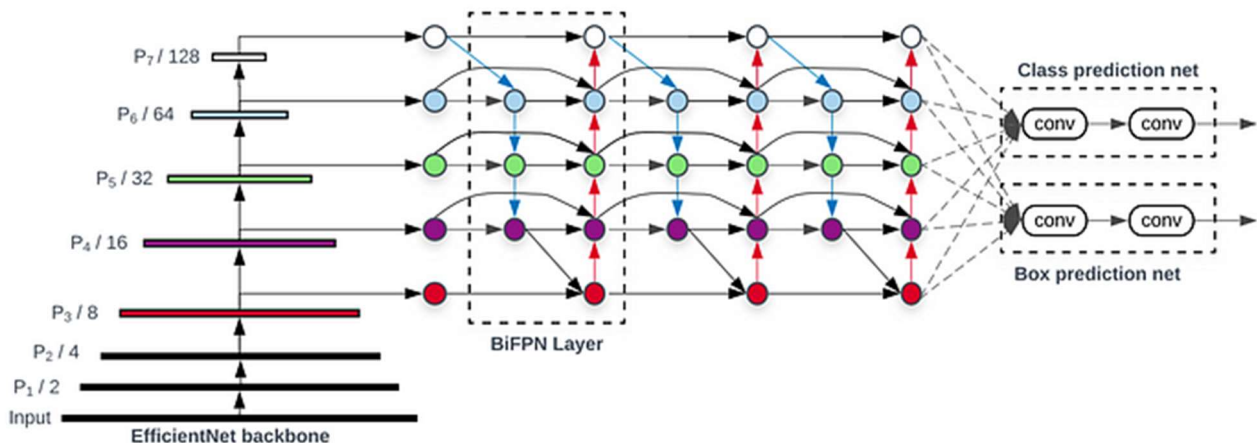


Рисунок 2.8 – Архітектура EfficientDet

Ширина BiFPN експоненціально збільшується (#channels), як це зроблено в EfficientNets, але лінійно збільшується глибину (#layers), оскільки глибину потрібно округляти до малих цілих чисел. Після пошуку по сітці було визначено, що найкращим масштабним коефіцієнтом для ширини є 1.35.

$$W_{bifpn} = 64 * (1.35^\varphi), \quad D_{bifpn} = 64 + \varphi \quad (2.2)$$

Ширина мережі прогнозування боксів/класів залишається такою ж, як і в BiFPN, але глибина (#layers) лінійно збільшується.

$$D_{box} = D_{class} = 3 + [\varphi/3] \quad (2.3)$$

Оскільки в BiFPN використовуються рівні ознак 3-7, вхідна роздільна здатність має ділитися на $2^7 = 128$, тому лінійно збільшується роздільна здатність.

$$R_{input} = 512 + \varphi * 128 \quad (2.4)$$

де φ – це складений коефіцієнт, що контролює всі інші масштабовані розміри; BiFPN, мережа прогнозування боксів/класів та вхідна роздільна здатність масштабуються за допомогою рівняння 1, 2, 3 відповідно.

Результати масштабування представлено в таблиці 2.1.

Таблиця 2.1 – Конфігурації масштабування для EfficientDet D0-D6

	R_{input}	Магістральна мережа	BiFPN		D_{class}
			W_{bifpn}	D_{bifpn}	
D0 ($\phi=0$)	512	B0	64	3	3
D1 ($\phi=1$)	640	B1	88	4	3
D2 ($\phi=2$)	768	B2	112	5	3
D3 ($\phi=3$)	896	B3	160	6	4
D4 ($\phi=4$)	1024	B4	224	7	4
D5 ($\phi=5$)	1280	B5	288	7	4
D6 ($\phi=6$)	1280	B6	384	8	5
D7 ($\phi=7$)	1536	B6	384	8	5
D7x	1536	B7	384	8	5

EfficientDet має такі переваги як, висока продуктивність мережі при обмеженій кількості параметрів, що дозволяє запускати мережу при обмежених ресурсах, мережа є компактною та масштабованою в порівнянні з іншими моделями, а також забезпечує баланс між швидкістю та продуктивністю. Однак недоліками даної мережі є складність архітектури мережі, складність її навчання і залежність попередньонавчених мереж.

2.2.4 CenterNet

CenterNet [18] – це одноетапний детектор об'єктів, який виявляє кожен об'єкт як триплет, а не пару ключових точок. Він використовує два спеціальні модулі, названі каскадним об'єднанням кутів і об'єднанням центрів, які збагачують інформацію, зібрану в лівому верхньому і правому нижньому кутах, і надають більш розпізнавану інформацію в центральних областях, відповідно. Інтуїція полягає в тому, що якщо прогнозована гранична область має високий IoU

з опорною областю, то ймовірність того, що центральна ключова точка в її центральній області буде віднесена до того ж класу, є високою, і навпаки. Таким чином, під час виведення, після того, як пропозиція згенерована у вигляді пари кутових ключових точок, ми визначаємо, чи дійсно пропозиція є об'єктом, перевіряючи, чи є центральна ключова точка того ж класу, що потрапляє в її центральну область. Архітектура мережі CenterNet представлена на рисунку 2.9.

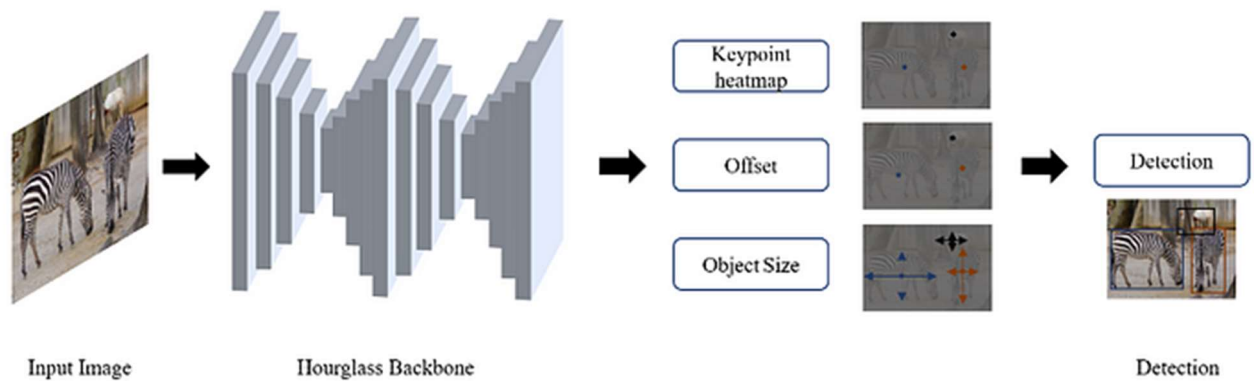


Рисунок 2.9 – Архітектура CenterNet

Як було описано вище, замість того щоб визначати обмежувальні рамки, що є складним при наявності великої кількості об'єктів, CenterNet визначає кожному об'єкту центральну точку. Це дозволяє виявляти об'єкти і підраховувати їх, використовуючи центральну точку в поєднанні з прогнозованими зміщеннями, щоб створити обмежувальну рамку, яка охоплює об'єкт [19].

Важливою особливістю CenterNet є використання Hourglass Backbone, що дозволяє об'єднувати різномасштабну інформацію та покращує здатність моделі фіксувати контекстну інформацію.

Після Hourglass Backbone, CenterNet виконує оцінку та виявлення ключових точок.

Даної модель має такі переваги як, простота реалізації в порівнянні з традиційними моделями, також CenterNet може забезпечувати більшу ефективність та точність в порівнянні з складнішими моделями. Недоліками даної моделі є проблеми з виявленням перекритих об'єктів і залежність точності

виявлення від розташування центра об'єкта, також дана модель вимагає великої кількості ресурсів для навчання і є менш гнучкою до адаптації для різних задач.

2.2.5 YOLO

YOLO [20] – одноетапна модель, що використовує наскрізну нейронну мережу для одночасного прогнозу обмежувальних рамок та ймовірностей класів. Це відрізняється від підходу, який використовували попередні алгоритми виявлення об'єктів, що перепрофілювали класифікатори для виконання виявлення.

Застосовуючи принципово інший підхід до виявлення об'єктів, YOLO досягла найсучасніших результатів, випередивши інші алгоритми виявлення об'єктів у реальному часі з великим відривом.

На відміну від алгоритмів таких як Faster RCNN, які спочатку виявляють можливі області інтересу за допомогою мережі регіональних пропозицій, а потім виконують розпізнавання цих пропозицій окремо, YOLO виконує всі свої прогнози за допомогою одного повністю зв'язаного шару.

Тобто якщо методи, що використовують Region Proposal Networks виконують кілька ітерацій для одного зображення, то YOLO виконує лише одну ітерацію.

Алгоритм YOLO отримує зображення на вході, а потім використовує просту нейронну мережу глибокої згортки для виявлення об'єктів на зображенні. Архітектура моделі CNN, яка є основою YOLO, показана на рисунку 2.10.

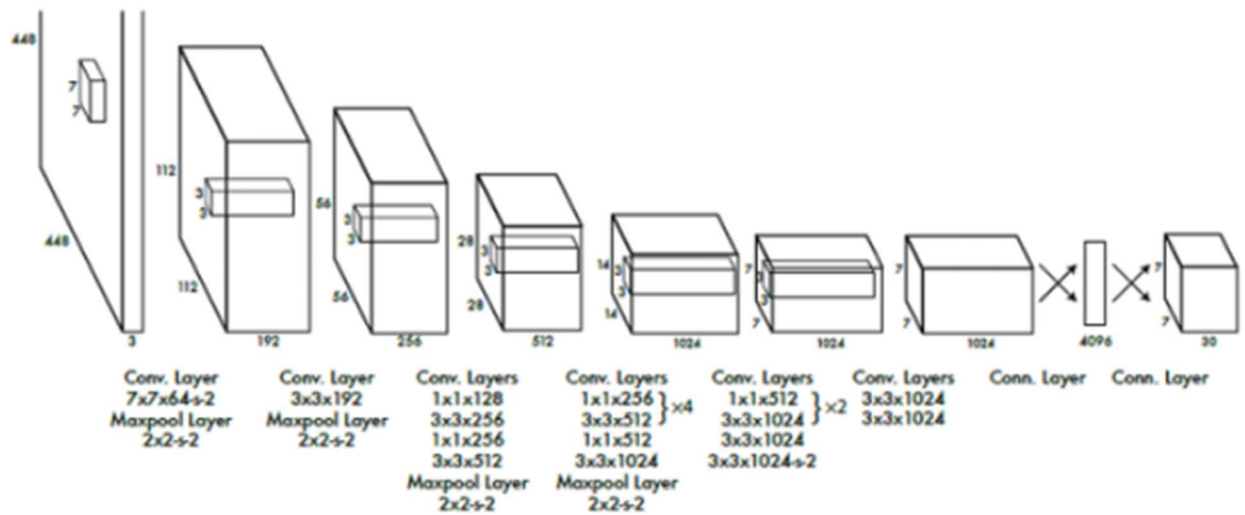


Рисунок 2.10 – Архітектура YOLO

Перші 20 шарів згортки моделі попередньо навчаються за допомогою ImageNet шляхом додавання тимчасового середнього об'єднання і повністю зв'язаного шару. Потім ця попередньо навчена модель конвертується для виконання виявлення, оскільки попередні дослідження показали, що додавання згортки та зв'язаних шарів до попередньо навченої мережі покращує продуктивність. Кінцевий повністю зв'язний шар YOLO передбачає як ймовірності класів, так і координати обмежувальних рамок.

Алгоритм YOLO працює за допомогою наступних трьох методів [21]:

- Залишкові блоки;
- Регресія обмежувальної коробки;
- Перетин через об'єднання (IOU).

Спочатку зображення розбивається на різні сітки. Кожна сітка має розмір $S \times S$. Приклад розбиття вхідного зображення на сітки показано на рисунку 2.11.



Рисунок 2.11 – Розбиття зображення на комірки

На зображенні вище є багато комірок сітки однакового розміру. Кожна комірка сітки визначатиме об'єкти, що з'являються в ній. Наприклад, якщо центр об'єкта з'являється у певній комірці сітки, то ця комірка відповідатиме за його виявлення.

Обмежувальна рамка – це контур, який виділяє об'єкт на зображенні. Кожна обмежувальна рамка на зображенні складається з наступних атрибутів:

- Ширина (bw);
- Висота (bh);
- Клас – позначається літерою c ;
- Центр обмежувальної рамки (b_x , b_y).

Приклад обмежувальної рамки показано на рисунку 2.12, обмежувальна рамка зображена жовтим кольором.

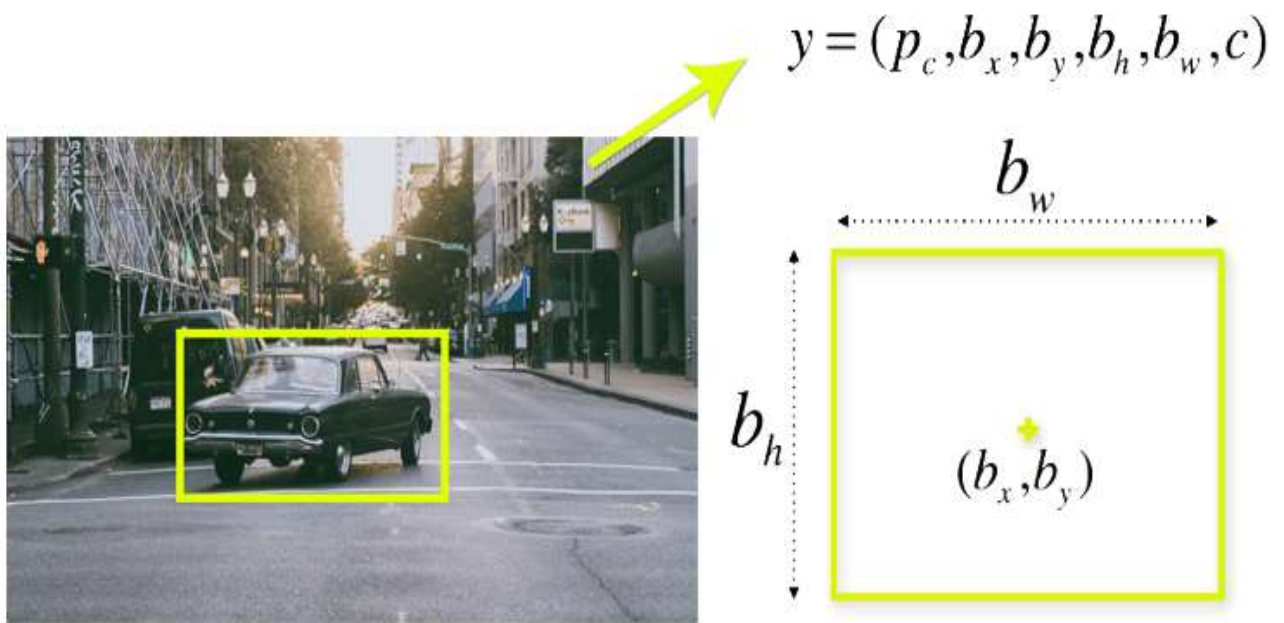


Рисунок 2.12 – Обмежувальна рамка

YOLO використовує єдину регресію обмежувальної рамки для прогнозування висоти, ширини, центру та класу об'єктів. На зображенні вище показано ймовірність того, що об'єкт з'явиться в обмежувальній рамці.

Перетин над об'єднанням (IOU) – це явище у виявленні об'єктів, яке описує, як блоки перетинаються. YOLO використовує IOU для створення вихідного вікна, яке ідеально оточує об'єкти.

Кожна комірка сітки відповідає за прогнозування обмежувальних рамок та їх достовірність. IOU дорівнює 1, якщо спрогнозована обмежувальна рамка збігається з реальною рамкою. Цей механізм усуває обмежувальні рамки, які не збігаються з реальними рамками.

Приклад роботи IOU показано на рисунку 2.13.

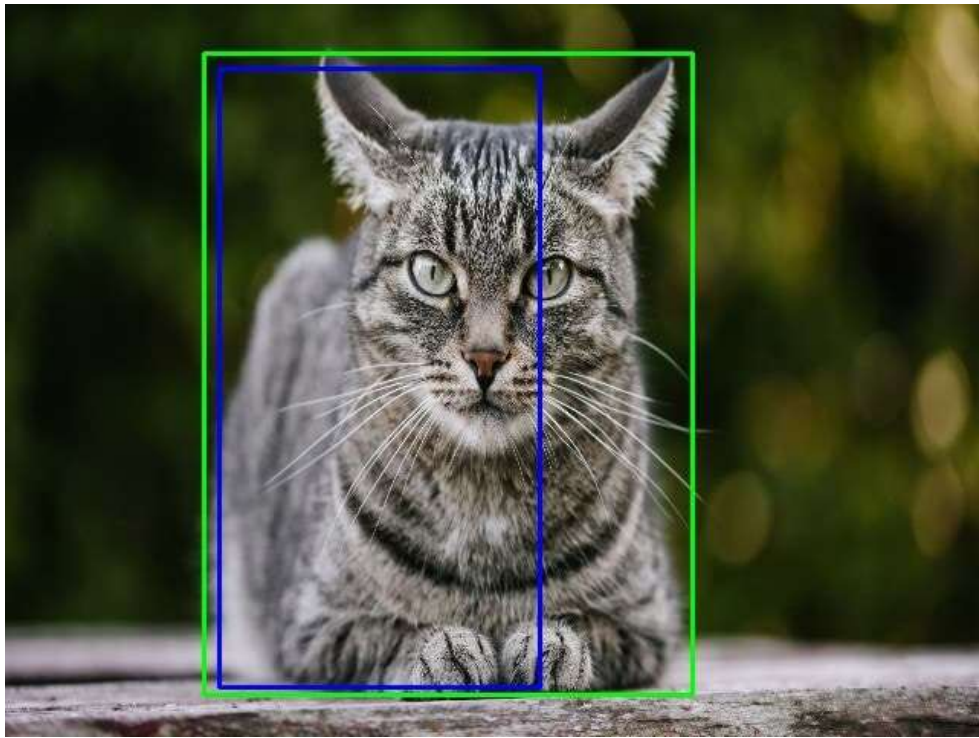


Рисунок 2.13 – Приклад роботи IOU

На зображенні вище є дві обмежувальні рамки, одна зеленого кольору, а інша – синього. Синя рамка – це прогнозована рамка, а зелена рамка – реальна рамка. YOLO гарантує, що ці два обмежувальні поля рівні.

Модель YOLO має такі переваги як, можливість роботи в реальному часі забезпечуючи хорошу точність на багатьох датасетах, дана модель є простою в розумінні і навчанні, може визначати мілкі об'єкти на зображенні. Недоліками даної моделі є труднощі виявлення об'єктів, що знаходяться близько одне до одного, а також менша точність локалізації в порівнянні з Faster-RCNN.

За наведеними результатами дослідження вирішено взяти модель YOLO, оскільки ця модель забезпечує оптимальне співвідношення між швидкістю та точністю класифікації зображень, також вона є простою в розумінні, що не створить труднощів в її створенні і навчанні.

2.3 Розробка структури згорткової нейронної мережі

Модель розробленої згорткової нейронної мережі складається з трьох головних частин: хребет (backbone), шийка (neck) і голова (head).

Хребет представляє собою модифіковану версію архітектури CSPDarknet53, вона складається з 53 згорткових шарів і використовує міжступеневі часткові з'єднання для покращення потоку інформації між різними шарами [22]. Замість модуля C3, що використовувався в мережі аналога, вирішено використати модуль C2f. Даний модуль інтегрує дві паралельні гілки градієнтного потоку за рахунок більшої кількості пропускних з'єднань і додаткових операцій розбиття, що забезпечує більш надійний потік градієнтної інформації [23]. Хребетна частина мережі закінчується модулем SPPF (Spatial Pyramid Pooling Fusion) для об'єднання вхідних карт ознак у карту фіксованого розміру для адаптивного виводу. Порівняно зі структурою просторового пірамідального об'єднання (SPP), SPPF зменшує обчислювальні зусилля і має меншу затримку завдяки послідовному з'єднанню трьох максимальних шарів об'єднання [24].

Шийна частина використовує мультимасштабний модуль об'єднання представлений комбінацією FPN (мережа піраміди ознак) і PAN (мережа агрегації шляхів). Завдяки двонаправленому об'єднанню низькорівневих ознак і високорівневих ознак, він посилює низькорівневі ознаки з меншими рецептивними полями і покращує здатність виявлення цілей різного масштабу. Рівень виявлення прогнозує положення цілі, категорії, оцінки впевненості та іншу інформацію.

Головна частина використовує без якірний підхід і замість зіставлення IOU або призначення односторонньої шкали, використовує призначувач завдань для зіставлення позитивних і негативних зразків. Зрештою, він виконує багатомасштабні прогнози, використовуючи 8-кратну, 16-кратну і 32-кратну дискретизацію, щоб досягти точних прогнозів для малих, середніх і великих цілей. Загальна структура згорткової нейронної мережі показана на рисунку 2.14.

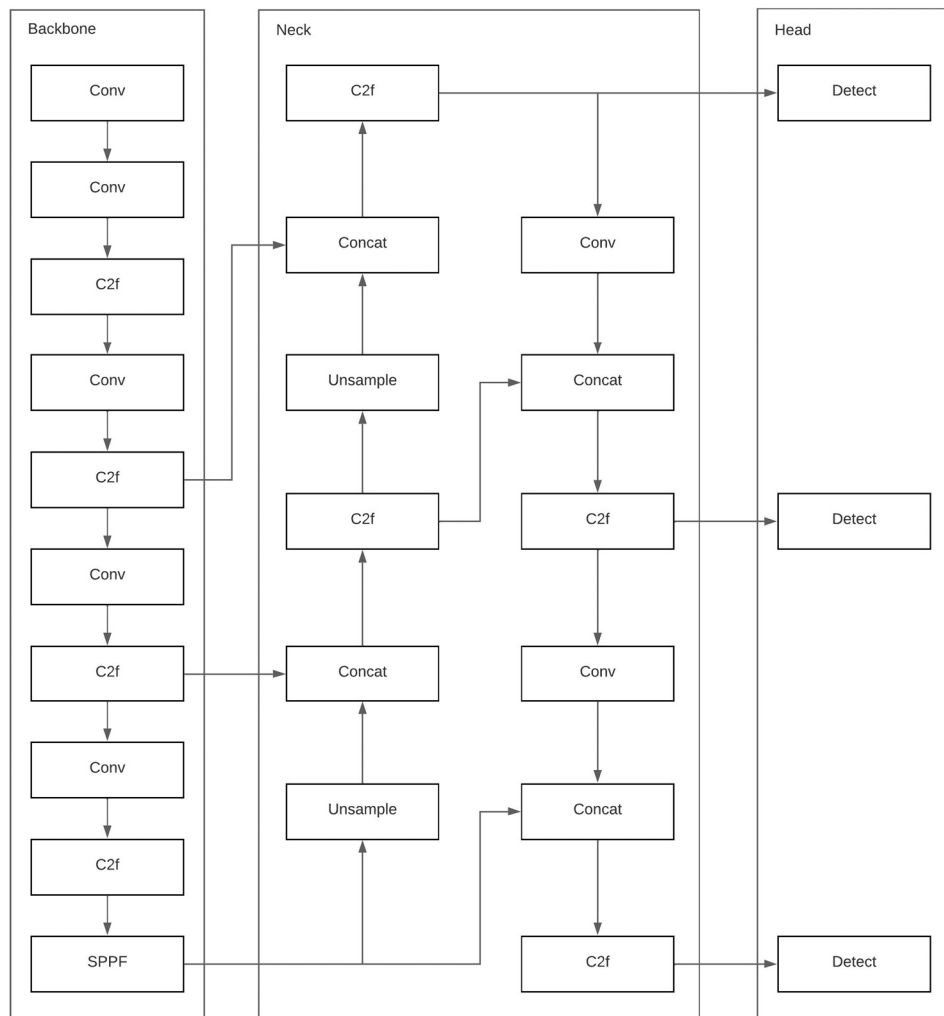


Рисунок 2.14 – Загальна структура згорткової нейронної мережі

Для підвищення ефективності та точності роботи нейронної мережі, вхідні зображення повинні бути квадратними і мати однаковий розмір. Вхідні зображення зберігаються в як масив NumPy і приводиться до розміру 640x640.

2.4 Розробка алгоритму роботи інформаційної технології виявлення озброєних людей у відеопотоці

Алгоритм роботи інформаційної технології виявлення озброєних людей у відеопотоці показано на рисунку 2.15.

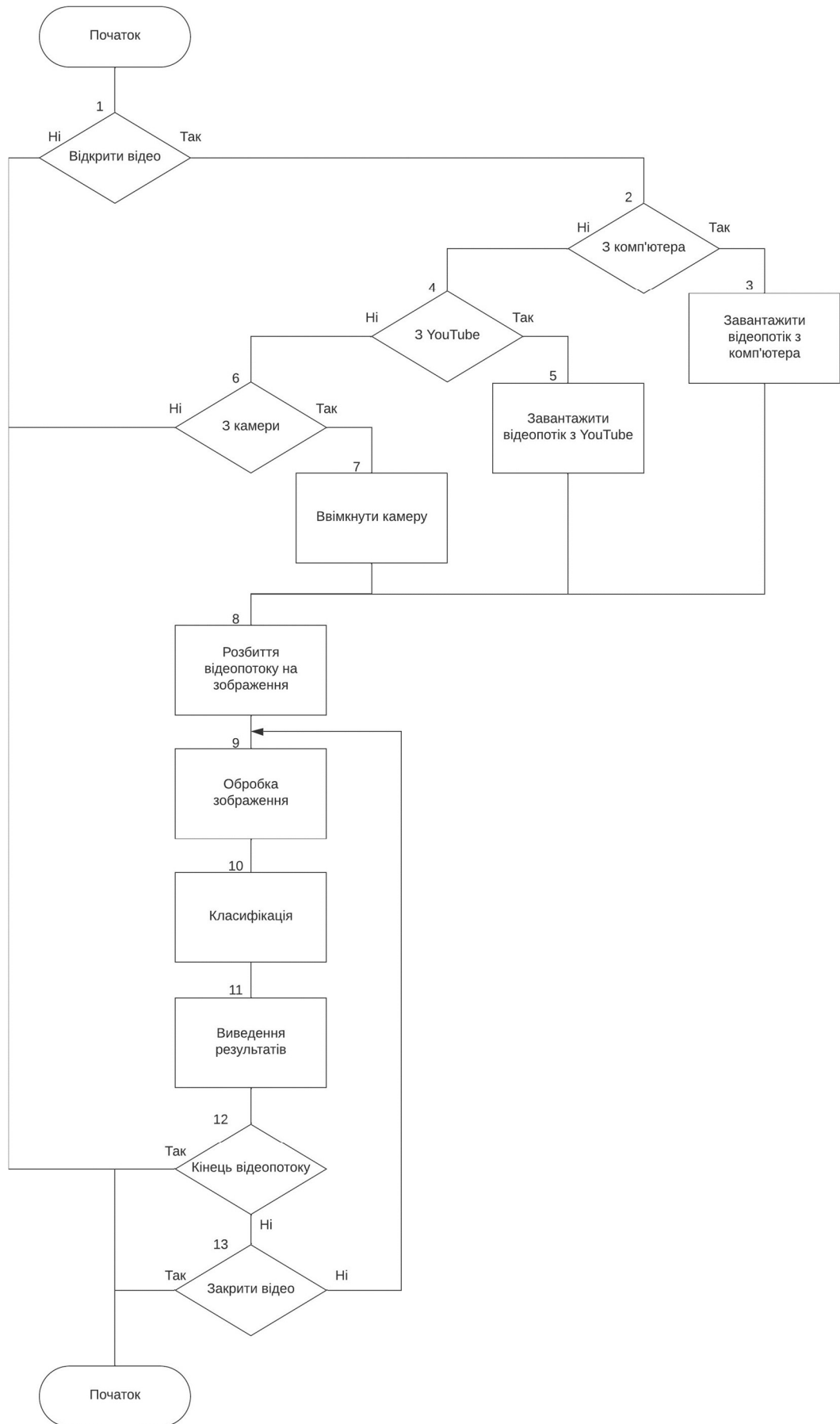


Рисунок 2.15 – Алгоритм роботи інформаційної технології виявлення озброєних людей у відеопотоці

Користувач вибирає джерело відеопотоку, з комп'ютера, з YouTube або з веб-камери, з вхідного потоку береться зображення, яке приводиться до необхідного розміру і передається на модуль виявлення для класифікації. Якщо на зображенні було виявлено об'єкт, що належить до одного з класів зброї («пістолет», «гвинтівка», «дробовик» або «кулемет»), тоді даний об'єкт позначається на зображенні з позначенням класу до якого він належить. Вихідне зображення з результатом класифікації виводиться на інтерфейс користувача і з відеопотоку береться наступне зображення для виявлення. Якщо відеопотік закінчився або користувач вибрав команду закрити відеопотік, тоді процес виявлення закінчується.

Діаграма діяльності інформаційної технології показана на рисунку 2.16.

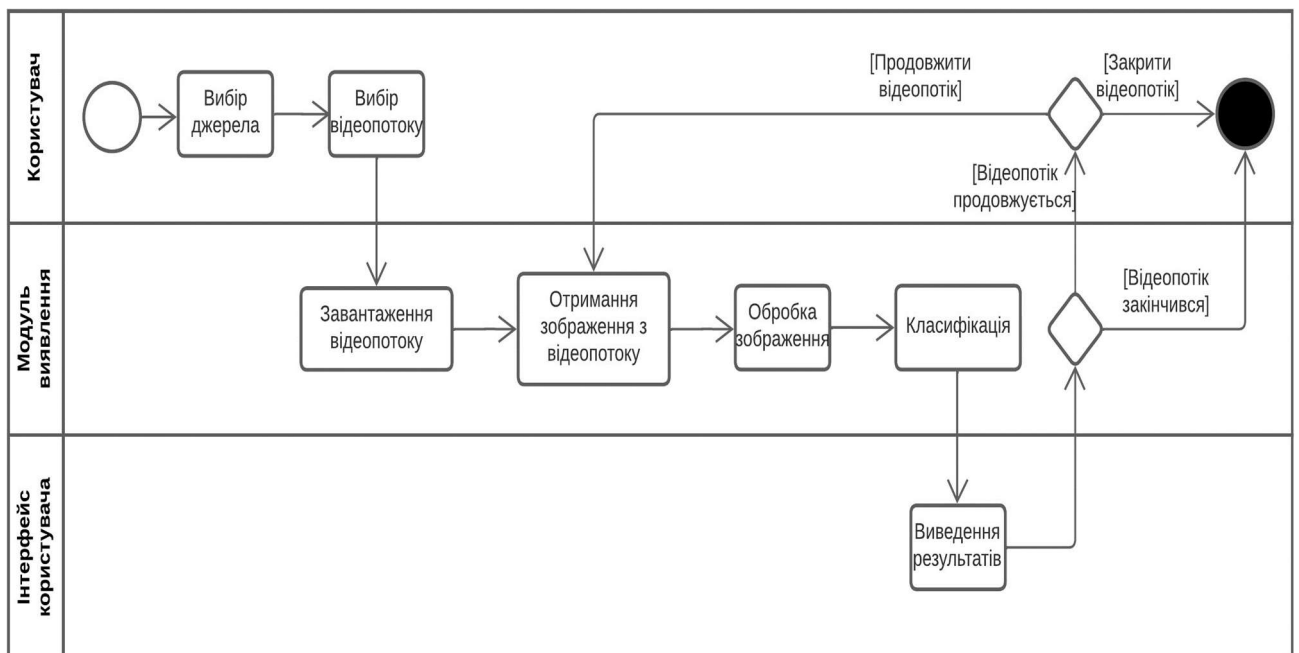


Рисунок 2.16 – Діаграма діяльності

Також в майбутньому можливо додати нові функціональні можливості для взаємодії користувача з інформаційною технологією, наприклад можливість ставити відеопотік на паузу або зберігати результати роботи модуля виявлення.

2.5 Висновок до розділу 2

В розділі було сформовано послідовність процесів обробки відеопотів для виявлення озброєних людей, а також визначено критерії яких необхідно дотримуватись в процесі навчання нейронної мережі.

Проведено порівняння архітектур згорткових нейронних мереж призначених для роботи в режимі реального часу, визначено їх особливості, принцип роботи, переваги й недоліки і на основі отриманих досліджень вибрано архітектуру, що підходить для розробки інформаційної технології.

Розглянуто принцип роботи вибраної архітектури нейронної мережі і розроблено її структури, а також розроблено алгоритм та діаграму діяльності роботи інформаційної технології виявлення озброєних людей.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ОЗБРОЄНИХ ЛЮДЕЙ У ВІДЕОПОТОЦІ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ

3.1 Обґрунтування вибору набору даних для навчання і тестування

Для якісного навчання нейронної мережі необхідно мати якісний набір даних, для цього набір даних повинен містити зображення з різними видами зброї з відповідними анотаціями. Також для забезпечення якості навчання навчальна вибірка даних повинна мати велику кількість прикладів для навчання.

Для збирання навчальних зображень скористаємось відкритими джерелами в мережі Інтернет. Навчальний набір був сформований злиттям наборів даних Pistol Detection [25], Weapon Detection Dataset [26] та Open Images Dataset (OID). Додатково зображення для навчального набору було взято з відкритих Інтернет джерел iStockPhoto, Pixabay, Unsplash та Shutterstock. Сформований набір зображень розбитий на класи pistol, rifle, shotgun та machinegun, що дозволить визначити яким видом зброї людини людина озброєна.

Розмітка зображень є складним і дуже важливим при побудові навчального набору, оскільки від якості зробленої розмітки залежить якість навчання нейронної мережі. Найчастіше для готових наборів даних розмітка зображень вже виконана і входить в набір даних, однак така розмітка може не задовольнити наші вимоги і необхідно виконати розмітку самостійно. Для розмітки зображень найчастіше використовують програму LabelImg, що дозволяє робити розмітку зображень в форматах Pascal VOC та TXT. Замість цього вирішено використати онлайн платформу roboflow, дана що забезпечує наступні можливості:

- простота розмітка зображень та можливість її зміни;
- просте розбиття зображень на навчальний, валідаційний та тестовий набори;
- можливість виконання попередньої обробки та аугментації наборів зображень для наборів даних;

- можливість генерації різних версій набору даних;
- вивантаження наборів даних в різних форматах, наприклад TXT, JSON, COCO, CSV, як на локальний комп'ютер, так і через roboflow API через згенероване посилання.

Всього кінцевий набір даних містить 4902 зображення в форматі .jpg, з них 3340 – навчальні, 762 – валідаційні, 800 – тестові. Розбиття кінцевого набору даних на класи показано в таблиці 3.1

Таблиця 3.1 – Розбиття кінцевого набору даних на класи

Клас	Вибірка	Кількість зображень
Pistol	Навчальна	1105
	Валідаційна	243
	Тестова	200
Rifle	Навчальна	1152
	Валідаційна	233
	Тестова	200
Shotgun	Навчальна	569
	Валідаційна	167
	Тестова	200
Machinegun	Навчальна	620
	Валідаційна	144
	Тестова	200

Всі зображення в кінцевому наборі приведені до одного розміру – 640x640, також додатково проведено аугментацію навчальних зображень: перевернення, обертання, яскравість та експозиція. Виконані кроки збільшили кількість навчальних зображень до 10020 додавши доповнення до існуючих зображень, що покращить якість навчання нейронної мережі.

3.2 Обґрунтування вибору мови програмування і середовища розробки

Сьогодні існує велика кількість мов програмування, при виборі мови програмування для розробки інформаційної технології необхідно врахувати такі особливості як, постановка задачі інформаційної технології, складність розробки, швидкість виконання коду, наявність доступних бібліотек та фреймворків.

В даному розділі проведемо порівняльну характеристику таких мов програмування: Python, C++, Java та R.

Python [27] – це високорівнева, інтерпретована, об'єктно-орієнтована мова програмування, яка підкреслює ідеали читабельності та простоти коду. Простий синтаксис Python відповідає за його стрімке зростання популярності. Крім того, стислий синтаксис дозволяє витратити значно більше часу на розробку фундаментальної структури, що робить Python відмінним вибором для процедур машинного навчання.

Також популярність Python обумовлена великою колекцією доступних бібліотек та фреймворків. Наприклад, такі бібліотеки, як TensorFlow, PyTorch і Keras, дозволяють розробникам створювати складні моделі типу ChatGPT і LLaMA, scikit-learn та NLTK полегшують завдання обробки природної мови, а Keras дозволяє зручно будувати модулі нейромережі, просто додаючи функції чи класи [26].

Єдиним недоліком мови Python можна виділити невисоку швидкість виконання програми.

C++ [27] – це розширення мови комп'ютера, яке можна використовувати для створення нейронні мережі. Швидкість C++ є найважливішою перевагою, оскільки розробка AI вимагає складних обчислень, а ця мова може прискорити обчислення. Вона має низькорівневий контроль над пам'яттю і підтримує програми, що вимагають активів, програми, які мають критичну для продуктивності, тощо. C++ може використовуватися в програмуванні зі штучним інтелектом для пошукової оптимізації та рейтингування. Однією з причин цього є широка гнучкість мови, що робить її ідеальною для ресурсомістких програм.

Серед мінусів C++ зазвичай відзначають її складність та неінтуїтивність, які можуть стати серйозними перешкодами для новачків [28].

Java [27] – це високорівнева, об’єктно-орієнтована мова програмування загального призначення. Синтаксис Java можна порівняти з синтаксисом мов C і C++; однак Java призначена як автономна та має мінімальні залежності.

Однією зі значних переваг Java є її продуктивність. Це компільована мова, отже, перед виконанням код перетворюється на машинний код. Цей процес призводить до швидшого виконання та кращої продуктивності порівняно з мовами, що інтерпретуються (PHP, Python і JavaScript). Java надає ряд фреймворків для розробки AI, включаючи Weka, Java-ML, H2O, DeepLearning4j і MOA, OenNLP, Kubeflow, бібліотеку Deep Java, Neuroph [28].

Java має один недолік — це не дуже доброзичлива до новачків мова, яка може бути складною в освоєнні без попереднього досвіду в програмуванні.

R [28] – це мова програмування, спеціально розроблена для статистичних обчислень та аналізу даних, що робить її ідеальною для роботи зі штучним інтелектом.

R має велику колекцію статистичних бібліотек і фреймворків, таких як Gmodels, RODBC, OneR і Tm, забезпечують широкую підтримку операцій машинного навчання. Мова є об’єктно-орієнтованою, розширюваною та дозволяє маніпулювати об’єктами іншими мовами. Ефективність R в обробці та аналізі даних є однією з його найважливіших переваг. Він також має чудові навички створення графіків. R, з іншого боку, важко вивчити. Він повільний і має недоліки безпеки [27].

Проаналізувавши отримані дані, для розробки вирішено вибрати мову програмування Python, оскільки дана мова програмування забезпечує простоту та ефективність розробки, також дана мова має велику кількість бібліотек призначених для роботи зі штучним інтелектом.

Для розробки на мові Python вибрано середовище розробки PyCharm, дане середовище підтримує такі особливості як потужний і функціональний редактор коду з підсвічуванням синтаксису, проста і потужна навігація в коді, потужний

рефакторинг коду для виконання швидких глобальних змін в проекті, простота встановлення то контролю версій бібліотек Python.

3.3 Обґрунтування вибору фреймворків та бібліотек

Як було розглянуто вище, для розробки складних моделей на мові Python є бібліотеки TensorFlow, PyTorch і Keras.

Keras [29] – це ефективний високорівневий інтерфейс прикладного програмування нейронних мереж, написаний на мові Python. Ця бібліотека нейронних мереж з відкритим вихідним кодом призначена для швидкого експериментування з глибокими нейронними мережами і може працювати поверх CNTK, TensorFlow і Theano.

Keras фокусується на модульності, зручності та розширюваності. Він не обробляє низькорівневі обчислення; натомість, він передає їх іншій бібліотеці, яка називається Backend. Keras була прийнята та інтегрована в TensorFlow в середині 2017 року. Користувачі можуть отримати доступ до неї через модуль `tf.keras`. Проте, бібліотека Keras все ще може працювати окремо і незалежно.

Переваги: [30]

- відмінний високорівневий API;
- безшовна інтеграція з TensorFlow, Aesara/Theano та CNTK;
- простота в вивченні та побудові нових архітектур;
- швидке експериментування.

Недоліки:

- найкраще підходить для роботи з невеликими наборами даних;
- при взаємодії з іншими фреймворками (наприклад TensorFlow), може працювати повільніше ніж інші фреймворки.

PyTorch [29] – це відносно новий фреймворк для глибокого навчання на основі Torch. Розроблений дослідницькою групою Facebook зі штучного інтелекту та розміщений з відкритим кодом на GitHub у 2017 році, він використовується для додатків з обробки природної мови. PyTorch має репутацію

простого, легкого у використанні, гнучкого, ефективного використання пам'яті та динамічних обчислювальних графіків. Він також відчувається як рідний, що робить кодування більш керованим і збільшує швидкість обробки.

Переваги: [30]

- простота вивчення;
- динамічна графова логіка, що підтримує швидке виконання;
- хороша взаємодія з мовою Python;
- підтримка навчання як на GPU, так і на CPU;
- підтримка розподіленого навчання.

Недоліки:

- менша кількість бібліотек та функцій розробки в порівнянні з TensorFlow;
- обмеженість візуалізації процесу навчання.

TensorFlow [29] – це наскрізний фреймворк глибокого навчання з відкритим вихідним кодом, розроблений компанією Google і випущений у 2015 році. TensorFlow є бібліотекою символічної математики, яка використовується для нейронних мереж і найкраще підходить для програмування потоків даних у широкому спектрі завдань. Вона пропонує кілька рівнів абстракції для побудови та навчання моделей.

Перспективний і швидкозростаючий вхід у світ глибокого навчання, TensorFlow пропонує гнучку, всеосяжну екосистему ресурсів спільноти, бібліотек та інструментів, які полегшують створення та розгортання додатків для машинного навчання.

Переваги: [30]

- чудова підтримка обчислювальних графіків, як для обчислень, так і для візуалізації;
- управління бібліотеками за підтримки Google з частими оновленнями та випусками;
- високопаралельні конвеєри з чудовою масштабованістю;
- наявність TPU (тензорних процесорів Google).

Недоліки:

- складність використання;
- використання TPU дозволяє лише виконання моделі, а не навчання;
- підтримка лише графічних процесорів NVIDIA.

На основі проведених досліджень для навчання згорткової нейронної мережі вирішено обрати бібліотеку PyTorch, оскільки вона бібліотека є простою та ефективною в роботі, підходить для навчання на великих наборах даних, а також підтримує можливість навчання на GPU.

Для обробки зображень засобами машинного зору, було обрано бібліотеку OpenCV.

OpenCV [31] – бібліотека з відкритим кодом для комп'ютерного зору, машинного навчання та обробки зображень. З її допомогою можна обробляти зображення та відео для ідентифікації об'єктів, облич або навіть почерку людини. При інтеграції з різними бібліотеками, такими як NumPy, Python здатний обробляти структуру масиву OpenCV для аналізу. Для ідентифікації шаблону зображення та його різних характеристик OpenCV використовує векторний простір і виконує математичні операції над цими характеристиками.

Для розробки графічного інтерфейсу програмного модуля обрано бібліотеку PyQt5.

PyQt [32] – це бібліотека, яка дозволяє використовувати фреймворк графічного інтерфейсу Qt з Python. Сама Qt написана на C++. Використовуючи її з Python, ви можете створювати програми набагато швидше, не жертвуючи при цьому швидкістю C++.

PyQt5 – це повний набір зв'язок Python для Qt v5, який реалізовано у вигляді більш ніж 35 модулів розширення і дозволяє використовувати Python як альтернативну мову розробки додатків до C++ на всіх підтримуваних платформах, включаючи iOS та Android.

3.4 Навчання мережі

Навчання згорткових нейронної мережі, особливо при застосуванні великих наборів навчання, вимагає наявності великої кількості обчислювальних ресурсів. Для навчання нейронної мережі обрано сервіс Google Colab, даний сервіс представляє собою онлайн блокнот для мови Python і надає можливість використання апаратного прискорення графічних процесорів, що пришвидшує навчання нейронних мереж. Для навчання мережі використовувався графічний процесор Nvidia Tesla T4, даний графічний процесор побудований на архітектурі Turing з використанням апаратних тензорних ядер і містить 16Gb відеопам'яті, що робить його одним з найкращих графічних прискорювачів для машинного навчання.

Завантажимо набір інструментів `ultralitics` необхідний для навчання згорткової нейронної мережі і встановимо всі необхідні для роботи пакети:

```
!git clone https://github.com/ultralitics/ultralitics
%cd ultralitics
!pip install -r requirements.txt
```

Імпортуємо бібліотеку `ultralitics` і перевіримо чи правильно бібліотеку було встановлено:

```
%cd ..
!pip install ultralitics

import ultralitics
ultralitics.checks()
```

Завантажимо набір даних для навчання нейронної мережі з сайту `roboflow`. Для цього необхідно встановити бібліотеку `roboflow` для взаємодії з платформою, на сайті згенеруємо код необхідний для завантаження необхідної версії

створеного набору даних, додатково в кореневій папці створимо папку `datasets` в яку вивантажимо набір даних:

```
import os
HOME = os.getcwd()
!mkdir {HOME}/datasets
%cd {HOME}/datasets
!pip install roboflow -quiet

from roboflow import Roboflow
rf = Roboflow(api_key='Rnwfi3fZEz4j8L8IIB1T')
project = rf.workspace('evgeny-yankovsky').project('weapon-vtrsk')
dataset = project.version(4).download('yolov8')
```

Наступним кроком необхідно створити файл конфігурації `.yaml` відповідно до розробленої структури згорткової нейронної мережі. Спочатку необхідно задати параметри нейронної мережі:

```
nc: 4 # кількість класів для розпізнавання
scales: [0.67, 0.75, 768] # глибина, ширина та розділювальна
здатність мережі
```

Як було описано вище, розроблена мережі YOLO складається з трьох головних частин: `backbone`, `neck` і `head`.

Реалізувати компонент `backbone` відповідно до розробленої структури нейромережі:

```
backbone:
  # [from, repeats, module, args]
  - [-1, 1, Conv, [64, 3, 2]]
  - [-1, 1, Conv, [128, 3, 2]]
  - [-1, 3, C2f, [128, True]]
  - [-1, 1, Conv, [256, 3, 2]]
```

- [-1, 6, C2f, [256, True]]
- [-1, 1, Conv, [512, 3, 2]]
- [-1, 6, C2f, [512, True]]
- [-1, 1, Conv, [1024, 3, 2]]
- [-1, 3, C2f, [1024, True]]
- [-1, 1, SPPF, [1024, 5]]

Створимо компонент `head`, даний компонент включає в себе компонент `neck` і вихідний шар розмір якого створюється відповідно до значення параметра `nc`:

`head`:

- [-1, 1, nn.Upsample, [None, 2, 'nearest']]
- [[-1, 6], 1, Concat, [1]]
- [-1, 3, C2f, [512]]

- [-1, 1, nn.Upsample, [None, 2, 'nearest']]
- [[-1, 4], 1, Concat, [1]]
- [-1, 3, C2f, [256]]

- [-1, 1, Conv, [256, 3, 2]]
- [[-1, 12], 1, Concat, [1]]
- [-1, 3, C2f, [512]]

- [-1, 1, Conv, [512, 3, 2]]
- [[-1, 9], 1, Concat, [1]]
- [-1, 3, C2f, [1024]] # 21 (P5/32-large)

- [[15, 18, 21], 1, Detect, [nc]]

Створені вище параметри та компоненти нейронної мережі запишемо в файл `yolov8m.yaml`:

```
%cd /content
with open('yolov8m.yaml', 'w') as f:
    f.write(content)
```

Створимо нейронну мережу відповідно до конфігурації в створеному .yaml файлі з трансферними вагами:

```
from ultralytics import YOLO
model = YOLO('yolov8m.yaml').load('yolov8m.pt')
```

Внесемо зміни в файлі data.yaml вивантаженого набору даних, задамо повні шляхи до наборів даних щоб уникнути виникнення помилок при навчанні мережі:

```
content = «»»
names:
- Machinegun
- Pistol
- Rifle
- Shotgun
nc: 4
roboflow:
  license: CC BY 4.0
  project: weapon-vtrsk
  url: https://universe.roboflow.com/evgeny-yankovsky/weapon-
vtrsk/dataset/3
  version: 3
  workspace: evgeny-yankovsky
test: /content/datasets/Weapon-4/test/images
train: /content/datasets/Weapon-4/train/images
val: /content/datasets/Weapon-4/valid/images

«»»
%cd /content/datasets/Weapon-3
```

```
with open(«/content/datasets/Weapon-4/data.yaml», «w») as f:
    f.write(content)
```

Виконаємо навчання нейронної мережі за допомогою функції `train`, задавши наступні параметри:

- `data` – шлях до файлу набору даних `data.yaml`;
- `epochs` – кількість епох навчання;
- `imgsz` – розмір навчальних зображень.

```
Import os
HOME = os.getcwd()
%cd {HOME}
results = model.train(data='/content/datasets/Weapon-4/data.yaml',
epochs=50, imgsz=640)
```

Збережемо навчені ваги нейронної мережі на локальний комп'ютер:

```
from google.colab import files
files.download('/content/runs/detect/train/weights/best.pt')
```

3.5 Висновок до розділу 3

У розділі було обґрунтовано створення набору даних для навчання нейронної мережі і приведено структуру розробленого набору. Обґрунтовано вибір мови програмування Python та бібліотеки PyTorch для навчання нейронної мережі. Також для програмної реалізації було використано бібліотеки OpenCV – для реалізації обробки зображень та PyQt5 – для реалізації графічного інтерфейсу. Наведено та пояснено кроки створення та навчання згортової нейронної мережі для виявлення озброєних людей у відеопотоці.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ОЗБРОЄНИХ ЛЮДЕЙ У ВІДЕОПОТОЦІ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ

4.1 Тестування інформаційної технології виявлення озброєних людей у відеопотоці

Головне вікно інформаційної технології (рис. 4.1) містить випадаючий список для вибору джерела відеопотоку: З комп'ютера, З YouTube, З камери (список джерел можна в майбутньому можна розширюючи), для джерела відеопотоку YouTube додано поле для введення URL-адреси відеопотоку, кнопки відкриття та закриття відеопотоку, а також область перегляду відеопотоку.

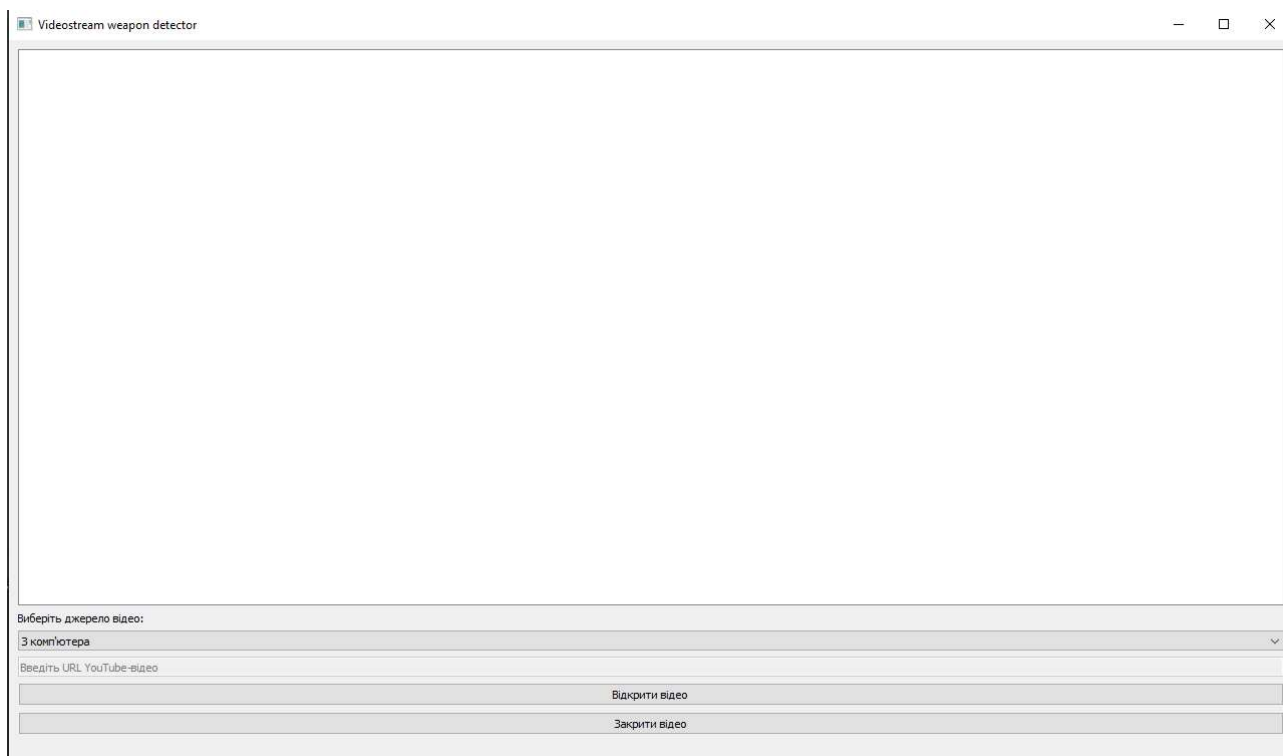


Рисунок 4.1 – Головне вікно програми

Після натискання на кнопку Відкрити відео відкривається відеопотік необхідний для розпізнавання озброєних людей, залежно від вибраного джерела відео, спосіб відкриття відеопотоку буде різний:

- З комп'ютера: відкривається вікно для вибору відеопотоку (рис. 4.2);

- З YouTube: в текстове поле вводиться посилання на відео (рис. 4.3);
- З камери: вмикається веб-камера комп'ютера.

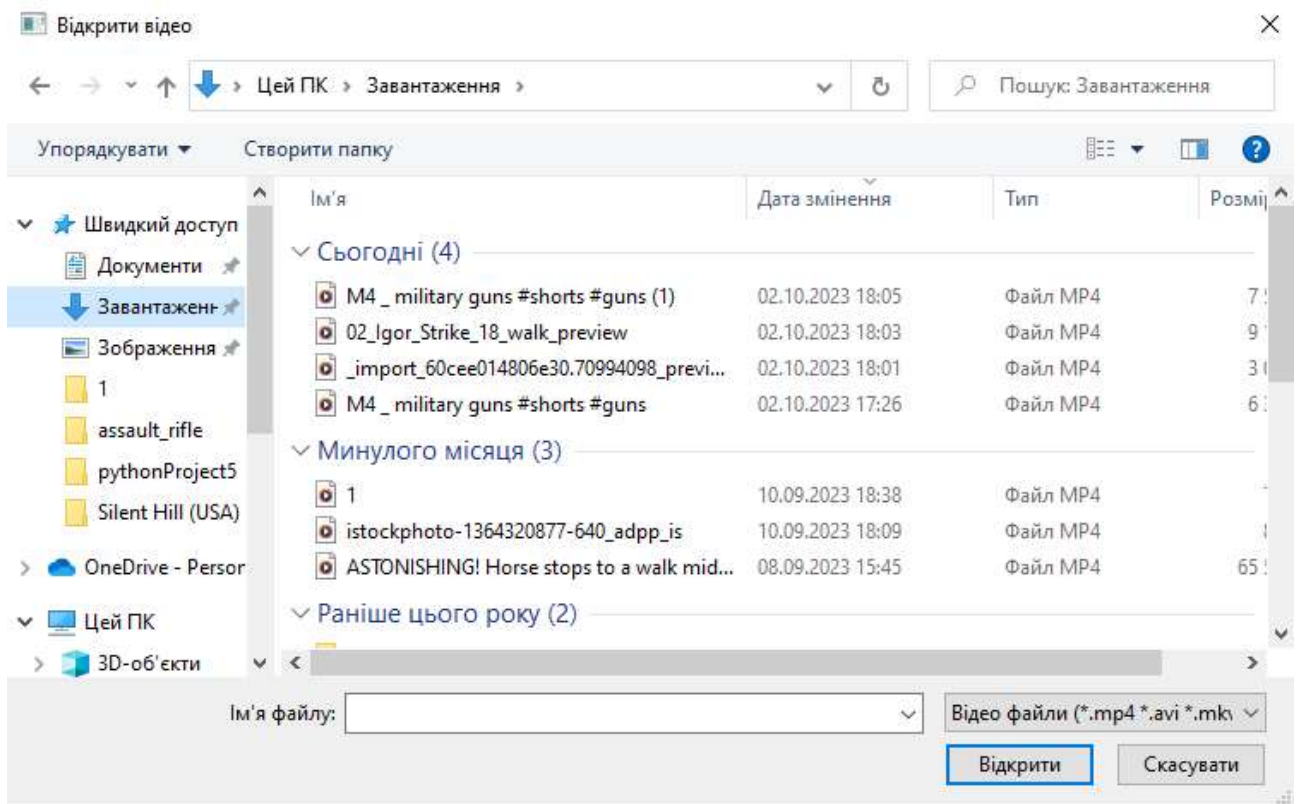


Рисунок 4.2 – Вікно вибору відеопотоку

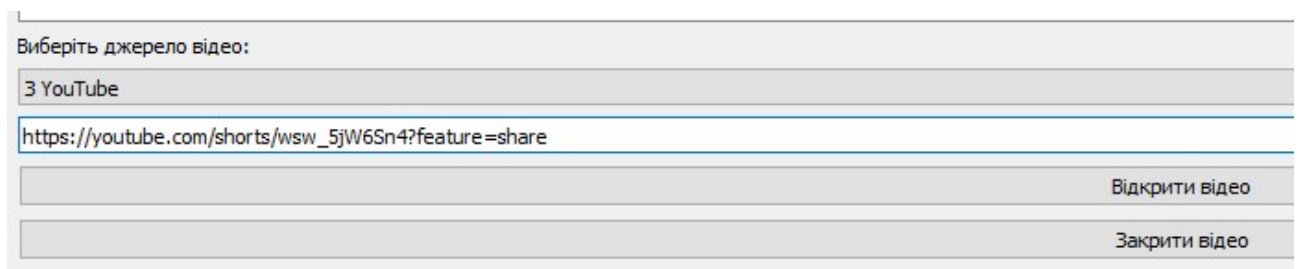


Рисунок 4.3 – Текстове поле для посилання на відеопотік

Вибраний відеопотік завантажується і розбивається зображення, що подаються на вхід нейронної мережі. Зображення проходять класифікацію і на них виділяються і розпізнані об'єкти з позначення класу до якого вони належать. Результати розпізнавання виводяться користувачу на головному вікні у вигляді відеопотоку який триває доки відеопотік не закіниться або користувач не натисне кнопку Закрити відео.

Якщо на зображенні взятому з відеопотоку розпізнано зброю, вона виділяється і підписується до якого класу вона належить («pistol», «rifle», «shotgun» або «machinegun»), як показано на рисунках 4.4 – 4.7.

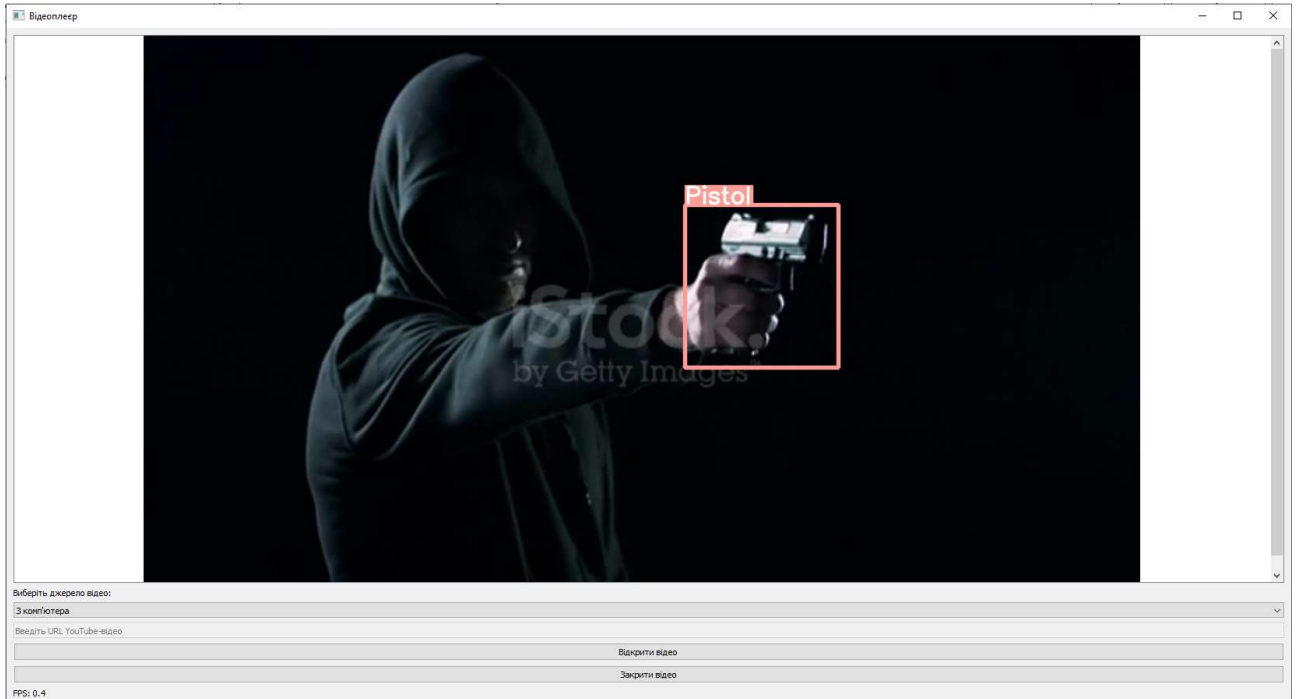


Рисунок 4.4 – Результат класифікації класу «pistol»

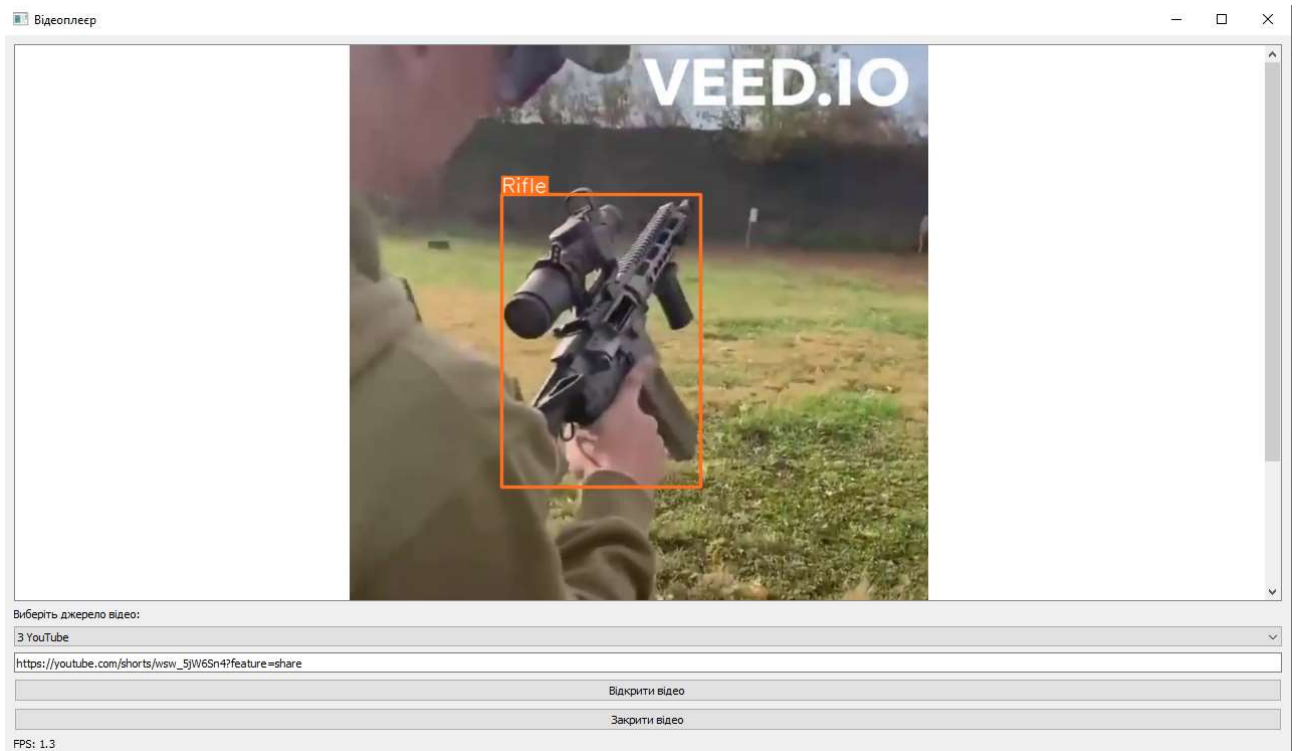


Рисунок 4.5 – Результат класифікації класу «rifle»

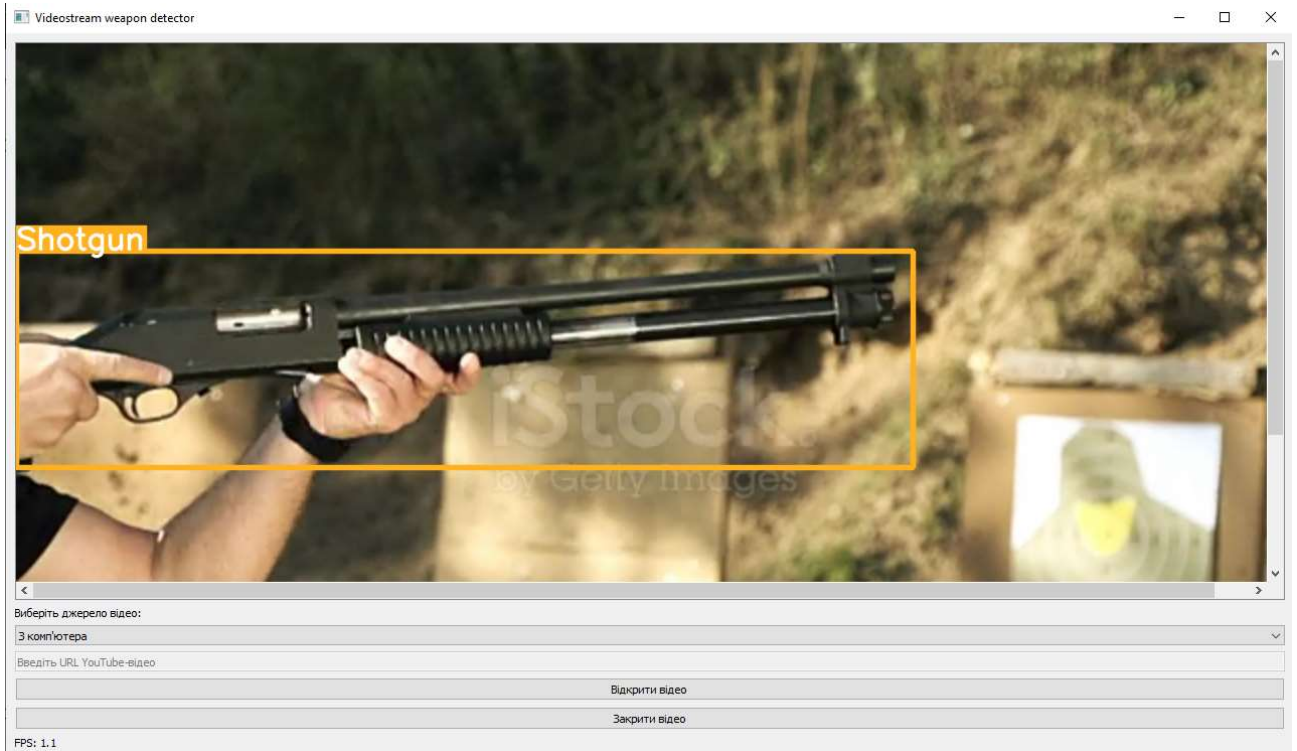


Рисунок 4.6 – Результат класифікації класу «shotgun»

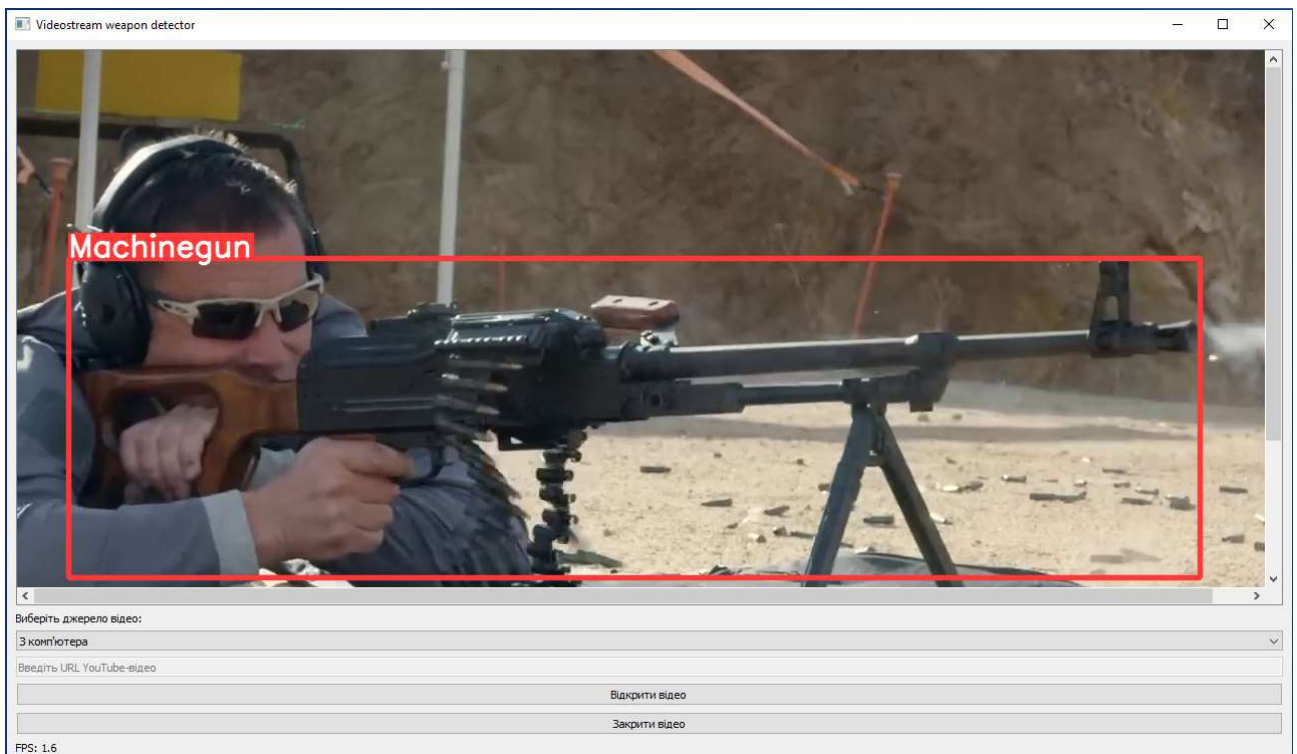


Рисунок 4.7 – Результат класифікації класу «machinegun»

При тестуванні достовірності розробленої інформаційної технології було використано 200 зображень для класу «пістолет», 200 зображень для класу

«гвинтівка», 200 зображень для класу «дробовик» та 200 зображень для класу «кулемет».

Результати тестування інформаційної технології наведено в таблиці 4.1.

Таблиця 4.1 – Результати тестування інформаційної технології

	pistol	rifle	shotgun	machinegun
pistol	196	0	2	5
rifle	2	195	3	3
shotgun	2	2	194	2
machinegun	0	3	1	192

4.2 Аналіз результатів тестування інформаційної технології виявлення озброєних людей у відеопотоці

Розрахуємо показники якості інформаційної технології за даними наведеними в таблиці 4.1. Для цього перетворимо таблицю результатів тестування на матрицю типу «один проти всіх» (бінарно-класова матриця) для обчислення метрик кожного класу, таких як достовірність, влучність, повнота.

Бінарно-класові матриці для класів «pistol», «rifle», «shotgun» та «machinegun» наведено в таблицях 4.2 – 4.5.

Таблиця 4.2 – Бінарно-класова матриця для класу «pistol»

	Pistol	Not pistol
Pistol	196	7
Not pistol	4	593

Достовірність (4.1):

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% = \frac{196+593}{800} \times 100\% = 98,63\% \quad (4.1)$$

Влучність (4.2):

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% = \frac{196}{196+4} \times 100\% = 98\% \quad (4.2)$$

Повнота (4.3):

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% = \frac{196}{196+7} \times 100\% = 96,6\% \quad (4.3)$$

Таблиця 4.3 – Бінарно-класова матриця для класу «rifle»

	Rifle	Not rifle
Rifle	195	8
Not rifle	5	592

Достовірність (4.4):

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% = \frac{195+592}{800} \times 100\% = 98,4\% \quad (4.4)$$

Влучність (4.5):

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% = \frac{195}{195+5} \times 100\% = 97,5\% \quad (4.5)$$

Повнота (4.6):

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% = \frac{195}{195+8} \times 100\% = 96,1\% \quad (4.6)$$

Таблиця 4.4 – Бінарно-класова матриця для класу «shotgun»

	Shotgun	Not shotgun
Shotgun	194	6
Not shotgun	6	594

Достовірність (4.7):

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+T} \times 100\% = \frac{194+594}{800} \times 100\% = 98,5\% \quad (4.7)$$

Влучність (4.8):

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% = \frac{194}{194+6} \times 100\% = 97\% \quad (4.8)$$

Повнота (4.9):

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% = \frac{194}{194+6} \times 100\% = 97\% \quad (4.9)$$

Таблиця 4.5 – Бінарно-класова матриця для класу «machinegun»

	Machinegun	Not machinegun
Machinegun	192	4
Not machinegun	8	596

Достовірність (4.10):

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+T} \times 100\% = \frac{192+596}{800} \times 100\% = 98,5\% \quad (4.10)$$

Влучність (4.11):

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% = \frac{192}{192} \times 100\% = 96\% \quad (4.11)$$

Повнота (4.12):

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% = \frac{192}{192} \times 100\% = 97,9\% \quad (4.12)$$

Знайдемо середнє арифметичне значення отриманих показників якості для класів і отримаємо значення показників якості інформаційної технології.

Достовірність (4.13):

$$\text{Accuracy} = 98,63 + 98,4 + 98,5 + 98,5 = 98,51\% \quad (4.13)$$

Влучність (4.14):

$$\text{Precision} = 98 + 97,5 + 97 + 96 = 97,13\% \quad (4.14)$$

Повнота (4.15):

$$\text{Recall} = 96,6 + 96,1 + 97 + 97,9 = 96,9\% \quad (4.15)$$

Ми отримали такі показники якості розробленої інформаційної технології виявлення озброєних людей у відеопотоці: достовірність 98,51%, влучність 97,13%, повнота 96,9%.

Для доведення поставленої мети, порівнюємо показники якості розробленої інформаційної технології з аналогом. Результати порівняння наведено в таблиці 4.6.

Таблиця 4.6 – Порівняння показників якості запропонованої інформаційної технології та аналога

	Аналог(Weapon detection)	Запропонована інформаційна технологія
Достовірність	93,7%	98,51%
Влучність	94,1%	97,13%
Повнота	93,2%	96,9%

З таблиці 4.6 видно, що достовірність виявлення озброєних людей в розробленої інформаційної технології вища на 4,81% (98,51% проти 93,7%) ніж в аналога. Також в розробленої інформаційної технології в порівнянні з аналогом влучність вища на 3,03% (97,13% проти 94,1%), а повнота вища на 3,7% (96,9%

проти 93,2%). Тобто мета роботи досягнута – достовірність виявлення озброєних людей у відеопотоках підвищена на 4,81%.

4.3 Висновок до розділу 4

В даному розділі було проведено тестування інформаційної технології виявлення озброєних людей у відеопотоках з використанням згорткових нейромереж. Проведено аналіз тестування кожного класу і визначено показники якості розробленої інформаційної технології. Проведений аналіз показав, що достовірність виявлення озброєних людей в розробленій інформаційній технології вища на 4,81% (98,51% проти 93,7%) ніж в аналога. Також в розробленого інформаційної технології в порівнянні з аналогом влучність вища на 3,03% (97,13% проти 94,1%), а повнота вища на 3,7% (96,9% проти 93,2%). Це означає що поставлена мета задачі виконана.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Оцінювання науково-технічного рівня та комерційного потенціалу розробки

Для оцінювання науково-технічного рівня та комерційного потенціалу розробки, проведемо комерційний та технологічний аудит розробки створеної в результаті виконання магістерської кваліфікаційної роботи. [33]

Для проведення комерційного та технологічного аудиту було залучено 3-х незалежних експертів. Результати оцінювання науково-технічного рівня та комерційного потенціалу розробки представлено в таблиці 5.1.

Таблиця 5.1 – Результати оцінювання науково-технічного рівня та комерційного потенціалу розробки

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	3	2	3
2. Ринкові переваги (наявність аналогів)	3	3	2
3. Ринкові переваги (ціна продукту)	3	3	4
4. Ринкові переваги (технічні властивості)	2	3	2
5. Ринкові переваги (експлуатаційні витрати)	2	3	3
6. Ринкові перспективи (розмір ринку)	2	2	3
7. Ринкові перспективи (конкуренція)	3	2	3
8. Практична здійсненність (наявність фахівців)	3	4	4
9. Практична здійсненність (наявність фінансів)	3	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	3	4
11. Практична здійсненність (термін реалізації)	3	3	4

Продовження таблиці 5.1

12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	35	35	37
Середньоарифметична сума балів СБ _c	35,7		

Відповідно до отриманих результатів науково-технічний рівень та комерційний потенціал розробки є вище середнього, такий рівень був досягнутий шляхом покращення та/або розширенням функціональних можливостей нової розробки в порівнянні з аналогом, а також за рахунок збільшення показника точності виявлення озброєних людей.

Проаналізуємо рівень конкурентоспроможності розробки та аналога відповідно до технічних параметрів. Технічні параметри розробки та аналога представлені в таблиці 5.2

Таблиця 5.2 – Технічні параметри аналога та розробки

Параметр	Одиниця виміру	Аналог	Нова розробка	Індекс зміни значення параметра	Коефіцієнт вагомості
<i>Технічні</i>					1
Інтерфейс	-	5	8	1,6	0,2
Функціональні можливості	-	4	9	2,25	0,15
Кількість класів розпізнавання	-	2	4	2	0,25
Потужність графічного процесора	Вт	45	65	0,69	0,3
Швидкість роботи	кадрів/сек	25	30	1,2	0,1

Продовження таблиці 5.2

<i>Економічні</i>					1
Ціна товару	грн	1700	1500	0,88	0,3
Транспортування	грн	600	550	0,92	0,1
Встановлення	грн	1000	900	0,9	0,2
Експлуатаційні витрати	грн	1100	1200	1,09	0,4

Розрахуємо нормативні параметри конкурентоспроможності за нормативними параметрами за формулою:

$$I_{\text{нп}} = \prod_{i=1}^n q_i \quad (5.1)$$

де $I_{\text{нп}}$ – загальний показник конкурентоспроможності за нормативними параметрами, програмне забезпечення відповідає нормативам, тому $I_{\text{нп}} = 1$;

q_i – одиничний показник за нормативним параметром.

$$I_{\text{нп}} = 1 \quad (5.2)$$

Знайдемо значення групового параметричного індексу за технічними параметрами з урахуванням вагомості кожного параметра:

$$I_{\text{тп}} = \sum_{i=1}^n q_i * \alpha_i \quad (5.3)$$

де $I_{\text{тп}}$ – груповий параметричний індекс за технічними параметрами;

q_i – одиничний параметричний показник параметра;

α_i – вагомість параметричного показника.

$$I_{\text{тп}} = 1,6 * 0,2 + 2,25 * 0,15 + 2 * 0,25 + 0,69 * 0,3 + 1,2 * 0,1 = 1,48 \quad (5.4)$$

Знайдемо значення групового параметричного індексу за економічними параметрами:

$$I_{\text{еп}} = \sum_{i=1}^n q_i * \beta_i \quad (5.5)$$

де $I_{\text{еп}}$ – груповий параметричний індекс за технічними параметрами;

q_i – одиничний параметричний показник параметра;

α_i – частка економічного параметра.

$$I_{\text{еп}} = 0,88 * 0,3 + 0,92 * 0,1 + 0,9 * 0,2 + 1,09 * 0,4 = 0,972 \quad (5.6)$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розраховують інтегральний показник конкурентоспроможності за формулою:

$$K_{\text{інт}} = I_{\text{нп}} * \frac{I_{\text{тп}}}{I_{\text{еп}}} \quad (5.7)$$

де $K_{\text{інт}}$ – інтегральний показник конкурентоспроможності;

$I_{\text{нп}}$ – загальний показник конкурентоспроможності за нормативними параметрами;

$I_{\text{тп}}$ – груповий параметричний індекс за технічними параметрами;

$I_{\text{еп}}$ – груповий параметричний індекс за економічними параметрами.

$$K_{\text{інт}} = 1 * \frac{1,48}{0,972} = 1,52 \quad (5.8)$$

Отже, $K_{\text{інт}} > 1$, це означає що нова розробка перевищує аналог за конкурентоспроможністю і є перспективною при виведенні нової розробки на ринок.

5.2 Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи

Розрахуємо основну заробітну плату дослідників (Z_o) відповідно до посадових окладів працівників, за формулою: [33]

$$Z_o = \sum_{i=1}^k \frac{M_{ni} * t_i}{T_p} \quad (5.9)$$

де M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – кількість днів роботи конкретного робітника, дн;

T_p – середня кількість робочих днів в місяці, $T_p=22$ дні.

Відповідно до вище описаної формули, знайдемо основну заробітну плату керівника проекту з місячним посадовим окладом 20000 грн та 40 днями роботи.

$$Z_o = \frac{20000 * 40}{22} = 36363,64 \quad (5.10)$$

Отже, заробітна плата керівника проекту складає 36363,64 гривень, решту розрахунків зводимо до таблиці 5.3.

Таблиця 5.3 – Витрати на заробітні плати дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Кількість днів роботи	Витрати на заробітну плату, грн
Керівник проекту	20000	909,09	40	36363,64
Науковий співробітник	18000	818,18	38	31090,91
Всього				67454,55

Розрахуємо додаткову заробітну плату дослідників ($Z_{\text{дод}}$), за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) * \frac{N_{\text{дод}}}{100\%} \quad (5.11)$$

де Z_o – заробітна плата дослідників;

Z_p – заробітна плата робітників, $Z_p = 0$;

$N_{\text{дод}}$ – норма нарахування додаткової заробітної плати, $N_{\text{дод}} = 12\%$.

$$Z_{\text{дод}} = (67454,55 + 0) * 0,12 = 8094,55 \quad (5.12)$$

Отже, додаткова заробітна плата дослідників складає 8094,55 гривень.

Розрахуємо відрахування на загальнообов'язкове державне соціальне страхування та здійснення заходів соціального захисту населення.

Нарахування (Z_n) на заробітну плату працівників розраховується як відсоток від суми основної та додаткової заробітної плати за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) * \frac{N_{\text{зп}}}{100\%} \quad (5.13)$$

де Z_o – заробітна плата дослідників;

Z_p – заробітна плата робітників, $Z_p = 0$;

$Z_{\text{дод}}$ – додаткова заробітна плата;

$N_{\text{зп}}$ – норма нарахування на заробітну плату, $N_{\text{зп}} = 22\%$.

$$Z_{\text{дод}} = (67454,55 + 0 + 8094,55) * 0,22 = 16620,80 \quad (5.14)$$

Отже, відрахування на загальнообов'язкове державне соціальне страхування та здійснення заходів соціального захисту населення складає 16620,80 гривень.

Розрахуємо витрати на матеріали (М) придбані в сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за прямим призначенням згідно з нормами їх витрачання, за формулою:

$$M = \sum_{j=1}^n H_j * C_j * K_j - \sum_{j=1}^n V_j * C_{Vj} \quad (5.15)$$

де H_j – норма витрат на матеріалу, кг;

C_j – вартість матеріалу, грн/кг;

K_j – коефіцієнт транспортних витрат, $K_j = 1,13$;

V_j – маса відходів, кг;

C_{Vj} – ціна відходів, грн/кг.

$$M = 236,17 + 61,02 + 597,77 = 894,96 \quad (5.16)$$

Отже, витрати на матеріали складають 894,96 гривень, зведемо отримані результати до таблиці 5.4.

Таблиця 5.4 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн	Вартість витраченого матеріалу
Офісний папір, Navigator, A4, 500 80 г/м ²	209	1	0	0	236,17
Канцелярське приладдя, Flair, кулькова ручка, чорна	27	2	0	0	61,02

Продовження таблиці 5.4

Картридж, Canon, PG-46, Black	529	1	0	0	597,77
Всього					894,96

Витрати на комплектуючі відсутні.

Витрати на спецустаткування відсутні.

Розрахуємо витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення ($V_{\text{прг}}$) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення, за формулою:

$$V_{\text{прг}} = \sum_{i=1}^k \text{Ц}_i * \text{С}_i * \text{К}_i \quad (5.17)$$

де Ц_i – ціна одиниці відповідного програмного забезпечення, грн;

С_i – кількість одиниць відповідного програмного забезпечення, грн;

К_i – коефіцієнт, що враховує інсталяцію, налагоджування програмного забезпечення, $\text{К}_i = 1,11$.

$$V_{\text{прг}} = 394,68 + 956,02 = 1350,70 \quad (5.18)$$

Отже, витрати на програмне забезпечення складає 1350,70 гривень, зведемо отримані результати до таблиці 5.5. Витрати враховують абонентну плату за їх використання протягом періоду розробки (хмарний сервіс).

Таблиця 5.5 – Витрати на придбання програмного забезпечення

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Середовище програмування PyCharm	1	355,57	394,68
Хмарне середовище Google Caloboratory	2	430,64	956,02
Разом			1350,70

Розрахуємо амортизаційні відрахування ($A_{обл}$) по кожному виду обладнання, устаткування та інших приладів і пристроїв, а також програмного забезпечення для проведення науково-дослідної роботи, за його наявності в дослідній організації або на підприємстві.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} * \frac{t_{вик}}{12} \quad (5.19)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Відповідно до описаної вище формули знайдемо амортизаційні відрахування по приміщенню з балансовою вартістю 50000 гривень з терміном використання 2,5 місяці.

$$A_{обл} = \frac{50000}{20} * \frac{2,5}{12} = 5208,33 \quad (5.20)$$

Отже, амортизаційні відрахування по приміщенню складають 5208,33 гривень, решту розрахунків зводимо до таблиці 5.3.

Таблиця 5.6 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Приміщення	500000	20	2,5	5208,33
Комп'ютер та комп'ютерна периферія	40000	2	2,5	4166,67
Офісне обладнання	20000	4	2,5	1041,67
Всього				10416,67

Розрахуємо витрати на електроенергію (B_e), що витрачається з технологічною метою на проведення досліджень, за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} * t_i * C_e * K_{впі}}{\eta_i} \quad (5.21)$$

де W_{yi} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість обробки обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; 7,50

$K_{впі}$ – коефіцієнт, що враховує використання потужності, $K_{впі} = 0,875$

η_i – коефіцієнт корисної дії обладнання, $\eta_i = 0,95$

Відповідно до описано вище формули знайдемо витрати на електроенергію для комп'ютера потужністю 0,068 кВт, що пропрацював 266 годин.

$$B_e = \frac{0,068 * 266 * 7,50 * 0,875}{0,95} = 124,95 \quad (5.22)$$

Отже, витрати на електроенергію для комп'ютера на час проведення досліджень складає 124,95 гривень, решту розрахунків зводимо до таблиці 5.4.

Таблиця 5.7 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Комп'ютер	0,068	266	124,95
Обслуговування робочого місця	0,018	266	33,08
Всього			158,03

Витрати на службові відсутні.

Витрати на роботи, які виконують сторонні підприємства, установи і організації відсутні.

Розрахуємо статтю витрат «Інші витрати» (I_B), до неї входять витрати, які не ввійшли в зазначені статті витрат і не відносяться до собівартості досліджень за прямими ознаками. Дані витрати розраховуються як відсоток від основної заробітної плати дослідників та робітників за формулою:

$$I_B = (Z_o + Z_p) * \frac{N_{iB}}{100\%} \quad (5.23)$$

де Z_o – заробітна плата дослідників, грн;

Z_p – заробітна плата робітників, $Z_p = 0$ грн;

N_{iB} – норма нарахування за статтею «Інші витрати».

Відповідно до описаної вище формули знайдемо «Інші витрати» при заробітній платі дослідників 67454,55 і нормі нарахувань 75%.

$$I_B = (67454,55 + 0) * 0,75 = 50590,91 \quad (5.24)$$

Отже, витрати за статтею «Інші витрати» складають 50590,91 гривень.

Розрахуємо статтю витрат «Накладні (загальновиробничі) витрати» ($V_{\text{нзв}}$), до них належать: витрати пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати розраховуються як відсоток від основної заробітної плати дослідників та робітників за формулою:

$$V_{\text{нзв}} = (Z_o + Z_p) * \frac{N_{\text{нзв}}}{100\%} \quad (5.25)$$

де Z_o – заробітна плата дослідників, грн;

Z_p – заробітна плата робітників, $Z_p = 0$ грн;

$N_{\text{нзв}}$ – норма нарахування за статтею «Інші витрати».

Відповідно до описаної вище формули знайдемо «Накладні (загальновиробничі) витрати» при заробітній платі дослідників 67454,55 і нормі нарахувань 125%.

$$V_{\text{нзв}} = (67454,55 + 0) * 1,25 = 84318,19 \quad (5.26)$$

Отже, витрати за статтею «Накладні (загальновиробничі) витрати» складають 84318,19 гривень.

Знайдемо витрати на проведення науково-дослідної роботи ($V_{\text{заг}}$), для цього знайдемо суму всіх попередніх статей витрат:

$$V_{\text{заг}} = 67454,55 + 0 + 8094,55 + 16620,80 + 894,96 + 0 + 0 + 1350,70 + 10416,67 + 158,03 + 0 + 0 + 50590,91 + 84318,19 = 245232,64 \quad (5.27)$$

Отже, витрати на проведення науково-дослідної роботи складають 245232,64 гривень.

Знайдемо загальні витрати (ЗВ) на завершення науково-дослідної роботи та оформлення її результатів за формулою:

$$ЗВ = \frac{V_{\text{заг}}}{\eta} \quad (5.28)$$

де $V_{\text{заг}}$ – витрати на проведення науково-дослідної роботи, грн;

η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи, $\eta = 0,5$.

Відповідно до описаної вище формули знайдемо загальні витрати на завершення науково-дослідної роботи та оформлення її результатів:

$$ЗВ = \frac{245232,64}{0,5} = 490465,28 \quad (5.29)$$

Отже, загальні витрати на завершення науково-дослідної роботи та оформлення її результатів складають 490465,28 гривень.

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів

тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. [33] Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Розрахуємо прибутки інвестора протягом трьох років після впровадження науково-технічної розробки. Для цього розрахуємо можливе збільшення чистого прибутку $\Delta\Pi_i$, для кожного з років протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки за формулою:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 * N + \Pi_0 * \Delta N)_i * \lambda * \rho * (1 - \frac{\vartheta}{100}) \quad (5.30)$$

де i – рік, $i = 1,2,3$;

$\pm\Delta\Pi_0$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році, $\pm\Delta\Pi_0 = 0$;

N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки, $N = 0$;

Π_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році;

ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році;

λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість, $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги), $\rho = 0,3$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, $\vartheta = 18\%$.

Відповідно до описаної вище формули розрахуємо прибутки інвестора протягом трьох років при ціні реалізованої розробки 1500 і при впровадженні 18000 одиниць реалізованої розробки:

$$\Delta\Pi_1 = (0 * 0 + 1500 * 12000)_1 * 0,8333 * 0,3 * \left(1 - \frac{18}{100}\right) = 3689852,4 \quad (5.31)$$

$$\Delta\Pi_2 = (0 * 0 + 1500 * 9000)_2 * 0,8333 * 0,3 * \left(1 - \frac{18}{100}\right) = 2767389,3 \quad (5.32)$$

$$\Delta\Pi_3 = (0 * 0 + 1500 * 6000)_3 * 0,8333 * 0,3 * \left(1 - \frac{18}{100}\right) = 1844926,2 \quad (5.33)$$

Отже, прибутки інвестора протягом першого року складають 3689852,4 гривень, протягом другого року – 2767389,3 гривень, протягом третього року – 1844926,2 гривень.

Розрахуємо приведену вартість збільшення всіх чистих прибутків ПП, які отримає інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^t} \quad (5.34)$$

де $\Delta\Pi_i$ – збільшення прибутку для кожного з років протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,1$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = \frac{3689852,4}{1,1^1} + \frac{2767389,3}{1,1^2} + \frac{1844926,2}{1,1^3} = 7027630,23 \quad (5.35)$$

Отже, приведена вартість збільшення всіх чистих прибутків, які отримає інвестор від можливого впровадження та комерціалізації науково-технічної розробки складає 7027630,23 гривень.

Розрахуємо величину початкових інвестицій PV , які інвестор повинен вкласти для впровадження та комерціалізації науково-технічної розробки, за формулою:

$$PV = k_{\text{інв}} * ЗВ \quad (5.36)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізації, $k_{\text{інв}} = 2$;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Відповідно до описаної вище формули розрахуємо величину початкових інвестицій при загальних витратах на проведення науково-технічної розробки та оформлення її результатів 490465,28 гривень:

$$PV = 2 * 490465,28 = 980930,56 \quad (5.37)$$

Отже, величина початкових інвестицій, які інвестор повинен вкласти для впровадження та комерціалізації науково-технічної розробки складає 980930,56 гривні.

Розрахуємо абсолютний економічний ефект $E_{\text{абс}}$ для інвестора від можливого впровадження та комерціалізації науково-технічної розробки:

$$E_{\text{абс}} = ПП - PV \quad (5.38)$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV – теперішня вартість початкових інвестицій, грн.

$$E_{abc} = 7027630,23 - 980930,56 = 6046699,67 \quad (5.39)$$

Отже, значення абсолютного економічного ефекту для інвестора від можливого впровадження та комерціалізації науково-технічної розробки складає 6046699,67 гривень. Значення E_{abc} є великим додатнім значенням, що свідчить про потенційну зацікавленість інвестора у впровадженні та комерціалізації науково-технічної розробки.

Для остаточного прийняття рішення по впровадженню науково-технічної розробки необхідно розрахувати внутрішню економічну дохідність E_B вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування τ_{\min} , яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішню економічну дохідність E_B розраховується за формулою:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1 \quad (5.40)$$

де E_{abc} – абсолютний економічний ефект від вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, роки.

Відповідно до описаної вище формули розрахуємо внутрішню економічну дохідність при абсолютному економічному ефекту 6046699,67 гривень, теперішній вартості інвестицій 980930,56 гривні і життєвому циклі 3 роки.

$$E_B = \sqrt[3]{1 + \frac{6046699,67}{980930,56}} - 1 = 0,93 \quad (5.41)$$

Отже, внутрішня економічна дохідність вкладених інвестицій дорівнює 0,92.

Визначимо бар'єрну ставку дисконтування $\tau_{\text{мін}}$ за формулою:

$$\tau_{\text{мін}} = d + f \quad (5.42)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках, $d = 0,45$;

f – показник, що характеризує ризикованість вкладення інвестицій, $f = 0,4$.

Відповідно до описаної вище формули розрахуємо бар'єрну ставку дисконтування:

$$\tau_{\text{мін}} = 0,45 + 0,4 = 0,85 \quad (5.43)$$

Отже, значення величини $E_B > \tau_{\text{мін}}$, тому інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок.

Розрахуємо період окупності інвестицій $T_{\text{ок}}$, які можуть бути вкладені інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{\text{ок}} = \frac{1}{E_B} \quad (5.44)$$

де E_B – внутрішня економічна дохідність вкладених інвестицій, $E_B = 0,93$.

Відповідно до описаної вище формули розрахуємо період окупності інвестицій:

$$T_{\text{ок}} = \frac{1}{0,93} = 1,08 \quad (5.45)$$

Отже, період окупності інвестицій складає 1,08. Значення величини $T_{\text{ок}} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

5.4 Висновок до розділу 5

В даному розділі було проведено розрахунок витрат на розробку та виготовлення нового технічного рішення, сума яких складає 490465,28 гривень та спрогнозовано витрати по кожній статті витрат. Також було розраховано чистий прибуток який може отримати інвестор від реалізації нового технічного рішення, розраховано економічний ефект і термін окупності вкладених інвестицій. В результаті розрахунків можна зробити висновок, що розробка є дешевшою та перспективнішою за аналог. Період окупності інвестицій складає 1,08 роки, що робить розробку комерційно привабливою.

ВИСНОВКИ

При виконанні магістерської кваліфікаційної роботи розв'язано задачу розробки інформаційної технології та програми розв'язання задачі виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж.

У першому розділі магістерської роботи було проведено аналіз предметної області розв'язання задачі виявлення озброєних людей, а саме – сформульовано постановку поставленої задачі, проведено огляд таких методів вирішення задачі виявлення озброєних людей: лінійна регресія, дерево рішень, метод опорних векторів, класифікатор Баєса та нейронні мережі. На основі проведеного аналізу було обрано метод на основі нейронних мереж. Проведено аналіз програмних засобів виявлення озброєних людей та обґрунтовано вибір програми-аналогу *Weapon detection*. Недоліками аналогу є низька достовірність виявлення зброї і низькі функціональні можливості, звідси випливає мета даної роботи – підвищення функціональних можливостей та достовірності виявлення озброєних людей.

У другому розділі магістерської кваліфікаційної роботи було проаналізовано різні архітектури згорткових нейронних мереж здатних до роботи в режимі реального часу та обґрунтовано вибір нейромережі типу *YOLO* для розв'язання поставленої задачі. Розроблено структуру обраного типу нейромережі та описано призначення використаних шарів. Також розроблено алгоритм та діаграму діяльності роботи інформаційної технології виявлення озброєних людей у відеопотоці.

У третьому розділі обґрунтовано створення навчального набору даних для навчання та тестування нейронної мережі, а також представлено розбиття набору даних на класи зброї необхідні для виявлення. Також розглянуто та обґрунтовано вибір мови програмування *Python* та бібліотеки *PyTorch* для виконання поставленої задачі. У результаті було розроблено програму виявлення озброєних людей у відеопотоці і описано кроки створення та навчання нейронної мережі в сервісі *Google Colab*.

У четвертому розділі було проведено тестування роботи програми виявлення озброєних людей у відеопотоці. Аналіз отриманих результатів показує, що достовірність виявлення озброєних людей в розробленій інформаційній технології вища на 4,81% (98,51% проти 93,7%) ніж в аналога. Також в розробленого інформаційної технології в порівнянні з аналогом влучність вища на 3,03% (97,13% проти 94,1%), а повнота вища на 3,7% (96,9% проти 93,2%). Також програма здатна розпізнавати 4 класи зброї, ніж 2 класи в аналога. Тобто мета магістерської кваліфікаційної роботи досягнута – достовірність виявлення озброєних людей підвищена на 4,81%, а кількість класів виявлення збільшена до 4 класів.

В п'ятому розділі було проведено розрахунок витрат на розробку та виготовлення нового технічного рішення, сума яких складає 490465,28 гривень та спрогнозовано витрати по кожній статті витрат. Також було розраховано чистий прибуток який може отримати інвестор від реалізації нового технічного рішення, розраховано економічний ефект і термін окупності вкладених інвестицій. В результаті розрахунків можна зробити висновок, що розробка є дешевшою та перспективнішою за аналог. Період окупності інвестицій складає 1,08 роки, що робить розробку комерційно привабливою.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Янковський Є. В., Колесницький О. К. «ПРОГРАМНИЙ МОДУЛЬ ВИЯВЛЕННЯ ОЗБРОЄНИХ ЛЮДЕЙ У ВІДЕОПОТОЦІ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ», в матеріали конференції «ЛІІ Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2023)», Вінниця, 2023, [Електронний ресурс]. Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/17346/14429>.
2. Logistic Regression in Machine Learning [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/logistic-regression-inmachine-learning>.
3. Decision Tree Classification Algorithm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/machine-learningdecision-tree-classification-algorithm>.
4. Використання "дерева рішень" [Електронний ресурс] – Режим доступу до ресурсу: https://pidru4niki.com/10780621/ekonomika/vikoristannya_dereva_rishen.
5. Support Vector Machine Algorithm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/machine-learning-supportvector-machine-algorithm>.
6. Naïve Bayes Classifier Algorithm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/machine-learning-naive-bayesclassifier>.
7. Kalyani Rajesh Gulalkari, Dr. Atul Joshi. Fog Enabled Forest Fire Management System using IoT and Machine Learning International Journal of Science and Research (IJSR) . Volume 11 Issue 3, March 2022. pp.1-5.
8. Bengio Y. Convolutional Networks for Images, Speech, and Time-Series [Електронний ресурс] / Y. Bengio, Y. Lecun. – 1997. – Режим доступу до ресурсу: https://www.researchgate.net/profile/Yann_Lecun/publication/2453996

[Convolutional Networks for Images Speech and TimeSeries/links/0deec519dfa2325502000000.pdf](https://arxiv.org/abs/1708.02002).

9. Khandelwal R. Convolutional Neural Network(CNN) Simplified [Электронный ресурс] / R. Khandelwal. – 2018. – Режим доступа до ресурсу: <https://medium.datadriveninvestor.com/convolutional-neural-network-cnn-simplified-ecafd4ee52c5>.

10. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>.

11. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016; pp. 779-788.

12. Faster R-CNN | ML [Электронный ресурс] – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/faster-r-cnn-ml/>.

13. Як працює RetinaNet? [Электронный ресурс] – Режим доступа до ресурсу: <https://developers.arcgis.com/python/guide/how-retinanet-works/>.

14. Feature Pyramid Networks for Object Detection [Электронный ресурс] / [L. Tsung-Yi, D. Piotr, G. Ross та ін.]. – 2016. – Режим доступа до ресурсу: <https://arxiv.org/abs/1612.03144>.

15. Focal Loss for Dense Object Detection [Электронный ресурс] / L. Tsung-Yi, G. Priya, G. Ross, H. Kaiming. – 2017. – Режим доступа до ресурсу: <https://arxiv.org/abs/1708.02002>.

16. EfficientDet [Электронный ресурс] – Режим доступа до ресурсу: <https://paperswithcode.com/method/efficientdet>.

17. EfficientDet: Scalable and Efficient Object Detection [Электронный ресурс] – Режим доступа до ресурсу: <https://medium.com/analytics-vidhya/efficientdet-scalable-and-efficient-object-detection-384a5df9011a>.

18. CenterNet [Электронный ресурс] – Режим доступа до ресурсу: <https://paperswithcode.com/method/centernet>.

19. How Object Detection Evolved (Part 3) [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@pandrii000/how-object-detection-evolved-part-3-94d6cccc4a45>.

20. YOLO: Algorithm for Object Detection Explained [+Examples] [Електронний ресурс] – Режим доступу до ресурсу: <https://www.v7labs.com/blog/yolo-object-detection>.

21. Introduction to YOLO Algorithm for Object Detection [Електронний ресурс] – Режим доступу до ресурсу: <https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/>.

22. Understanding YOLOv8 Architecture, Applications & Features [Електронний ресурс] – Режим доступу до ресурсу: <https://www.labellerr.com/blog/understanding-yolov8-architecture-applications-features/>.

23. Improving Detection Capabilities of YOLOv8-n for Small Objects in Remote Sensing Imagery: Towards Better Precision with Simplified Model Complexity [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/371845645_Improving_Detection_Capabilities_of_YOLOv8-n_for_Small_Objects_in_Remote_Sensing_Imagery_Towards_Better_Precision_with_Simplified_Model_Complexity.

24. UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mdpi.com/1424-8220/23/16/7190>.

25. Pistol Detection [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaggle.com/datasets/vaibhavtalekar/pistol-classification>.

26. Weapon Detection Dataset [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaggle.com/datasets/snehilsanyal/weapon-detection-test>.

27. 9 найкращих мов програмування, які використовуються в ШІ [Електронний ресурс] – Режим доступу до ресурсу: <https://hashdork.com/uk/programming-languages-used-in-ai/>.

28. Які мови програмування вивчати для роботи з AI? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.imena.ua/blog/what-programming-languages-to-learn-to-work-with-ai/>.

29. Keras vs Tensorflow vs Pytorch: Key Differences Among Deep Learning [Електронний ресурс] – Режим доступу до ресурсу: <https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article>.

30. Tensorflow, PyTorch or Keras for Deep Learning [Електронний ресурс] – Режим доступу до ресурсу: <https://domino.ai/blog/tensorflow-pytorch-or-keras-for-deep-learning>.

31. OpenCV – Overview [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/opencv-overview/>.

32. PyQt5 tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://build-system.fman.io/pyqt5-tutorial>.

33. Козловський В. О. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт [Електронний ресурс] / В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – 2021. – Режим доступу до ресурсу: https://epvm.vntu.edu.ua/wp-content/uploads/2021/09/Ekonom_magister_tehn_2021.pdf.

34. Яровий А. А. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / А. А. Яровий, О. К. Колесницький. – 2023. – Режим доступу до ресурсу: https://iq.vntu.edu.ua/method/getfile.php?fname=6786.pdf&card_id=23946&id=6786.

Додаток А (обов'язковий)

Протокол перевірки кваліфікаційної роботи на наявність текстових
запозиченьПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія виявлення озброєних людей у відеопотоці з використанням згорткових нейромереж

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 81,1% Схожість 18,9%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи

Янковський Є.

Керівник роботи

Колесницький О.К.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку

Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```

import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget,
QVBoxLayout, QPushButton, QFileDialog, QLineEdit, QLabel, QComboBox,
QGraphicsView, QGraphicsScene, QGraphicsPixmapItem
from PyQt5.QtGui import QImage, QPixmap, QColor, QPainter, QFont
from PyQt5.QtCore import Qt, QTimer
import cv2
import time
from pytube import YouTube
from ultralytics import YOLO
from ultralytics.utils.plotting import Annotator
from ultralytics.utils.plotting import Colors
from PIL import Image

colors = Colors()
model = YOLO("best.pt")

def annotate_frame(results, frame):
    for result in results:
        annotator = Annotator(frame)
        boxes = result.boxes
        for box in boxes:
            bbox = box.xyxy[0]
            cls = int(box.cls)
            annotator.box_label(box=bbox, label=model.names[cls],
color=colors(cls))
        return annotator.result()

class VideoPlayer(QMainWindow):
    def __init__(self):
        super().__init__()

        self.setWindowTitle("Videostream weapon detector")
        self.setGeometry(100, 100, 1280, 720)

        self.colors = Colors()
        self.model = YOLO("best.pt")

        self.video_sources = ["З комп'ютера", "З YouTube", "З камери"]
        self.selected_source = "З комп'ютера"

        self.initUI()

        self.video_source = 0

```



```

self.vid = cv2.VideoCapture(self.video_source)
self.is_playing = False
self.last_time = time.time()
self.frame_count = 0

self.timer = QTimer(self)
self.timer.timeout.connect(self.update)
self.timer.start(10)

def initUI(self):
    central_widget = QWidget()
    self.setCentralWidget(central_widget)

    layout = QVBoxLayout()

    self.canvas = QGraphicsView()
    layout.addWidget(self.canvas)

    source_label = QLabel("Виберіть джерело відео:")
    layout.addWidget(source_label)

    self.source_combo = QComboBox()
    self.source_combo.addItem(self.video_sources)
    layout.addWidget(self.source_combo)

    self.url_entry = QLineEdit()
    self.url_entry.setPlaceholderText("Введіть URL YouTube-відео")
    self.url_entry.setDisabled(True)
    layout.addWidget(self.url_entry)

    open_button = QPushButton("Відкрити відео")
    open_button.clicked.connect(self.open_video)
    layout.addWidget(open_button)

    close_button = QPushButton("Закрити відео")
    close_button.clicked.connect(self.close_video)
    layout.addWidget(close_button)

    self.fps_label = QLabel()
    layout.addWidget(self.fps_label)

    central_widget.setLayout(layout)

def open_video(self):
    selected_source = self.source_combo.currentText()
    if selected_source == "З комп'ютера":
        options = QFileDialog.Options()
        options |= QFileDialog.ReadOnly
        file_name, _ = QFileDialog.getOpenFileName(self, "Відкрити

```

```

video", "", "Відео файли (*.mp4 *.avi *.mkv)", options=options)
    if file_name:
        self.video_source = file_name
        self.vid = cv2.VideoCapture(self.video_source)
        self.is_playing = True
    elif selected_source == "3 YouTube":
        youtube_url = self.url_entry.text()
        try:
            yt = YouTube(youtube_url)
            stream = yt.streams.filter(adaptive=True,
file_extension='mp4').first()
            stream_url = stream.url
            self.vid = cv2.VideoCapture(stream_url)
            self.is_playing = True
        except Exception as e:
            print("Помилка при відкритті YouTube-відео:", str(e))
    elif selected_source == "3 камери":
        self.vid = cv2.VideoCapture(0)
        self.is_playing = True

def close_video(self):
    self.vid.release()
    self.is_playing = False
    self.clear_canvas()
    self.url_entry.clear()

def clear_canvas(self):
    scene = QGraphicsScene()
    self.canvas.setScene(scene)

def update(self):
    if self.is_playing:
        ret, frame = self.vid.read()
        if ret:
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            height, width, _ = frame.shape
            aspect_ratio = width / height
            if aspect_ratio > 1280 / 720:
                new_width = 1280
                new_height = int(1280 / aspect_ratio)
            else:
                new_height = 720
                new_width = int(720 * aspect_ratio)
            frame = cv2.resize(frame, (new_width, new_height))
            results = self.model(source=frame, stream=True)
            frame = annotate_frame(results, frame)
            image = QImage(frame.data, frame.shape[1],
frame.shape[0], frame.shape[1] * 3, QImage.Format_RGB888)
            pixmap = QPixmap.fromImage(image)

```

```
        scene = self.canvas.scene()
        if not scene:
            scene = QGraphicsScene()
            self.canvas.setScene(scene)

        item = QGraphicsPixmapItem(pixmap)
        scene.clear()
        scene.addItem(item)

        self.frame_count += 1
        current_time = time.time()
        elapsed_time = current_time - self.last_time
        if elapsed_time >= 1.0:
            fps = self.frame_count / elapsed_time
            self.fps_label.setText(f"FPS: {fps:.1f}")
            self.last_time = current_time
            self.frame_count = 0


        selected_source = self.source_combo.currentText()
        self.url_entry.setDisabled(selected_source != "3 YouTube")


if __name__ == "__main__":
    app = QApplication(sys.argv)
    player = VideoPlayer()
    player.show()
    sys.exit(app.exec_())
```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВІЯВЛЕННЯ ОЗБРОЄНИХ ЛЮДЕЙ У
ВІДЕОПОТОЦІ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ

Виконав: студент 2-го курсу,
групи ЗКН-22м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)
 Янковський Є.В.
(прізвище та ініціали)

Керівник: к.т.н., проф. каф. КН
 Колесницький О.К.
(прізвище та ініціали)
« 07 » 12 2023 р.

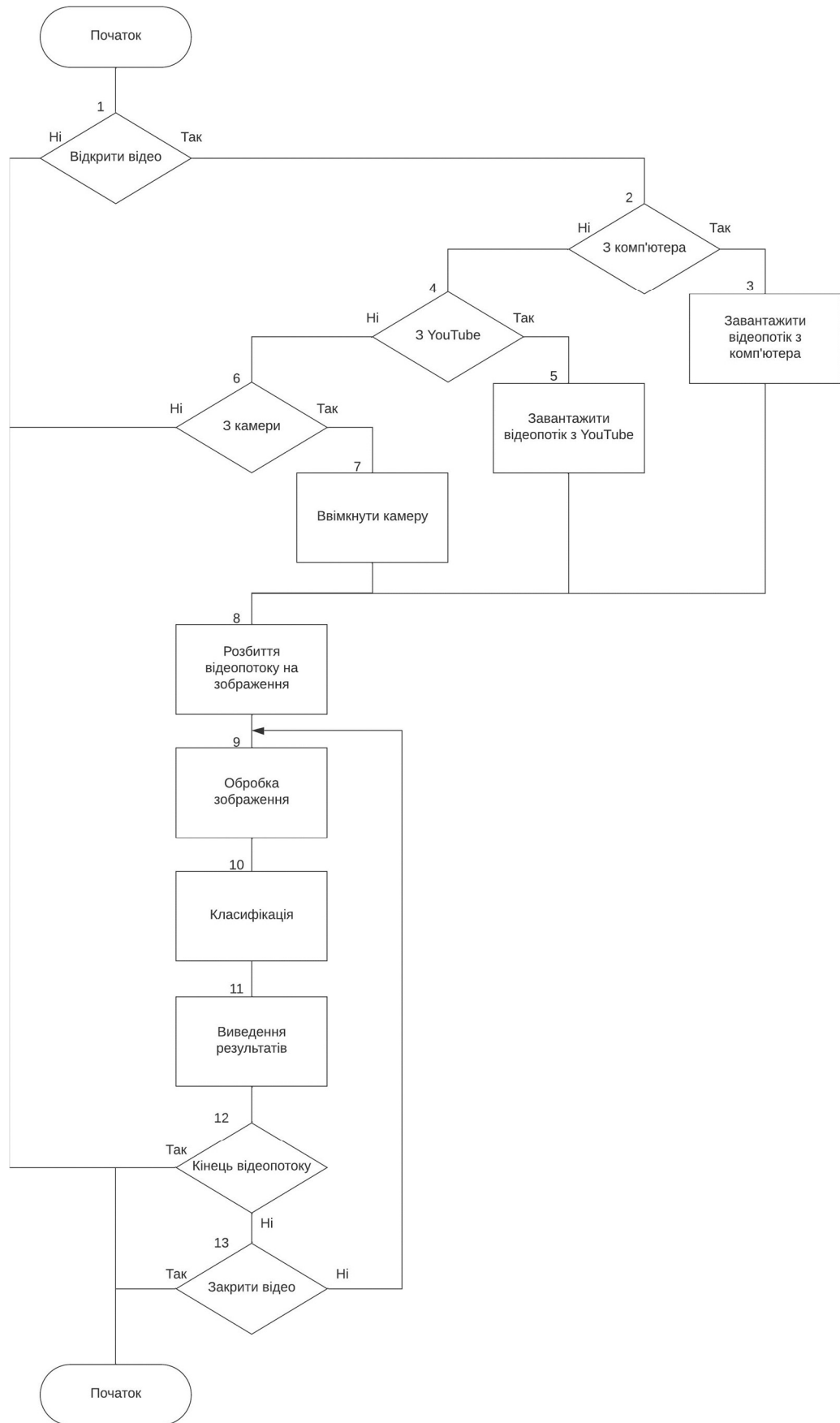


Рисунок В.1 – Схема алгоритму роботи інформаційної технології виявлення озброєних людей у відеопотоці

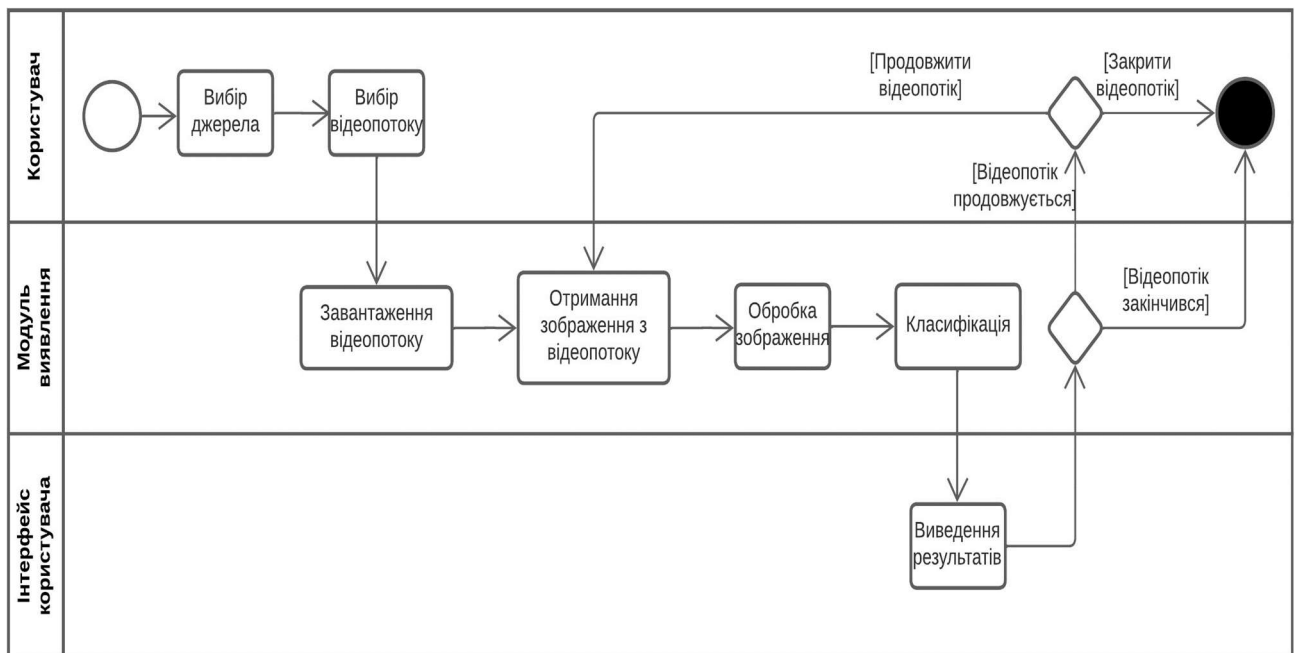


Рисунок В.2 – Діаграма діяльності

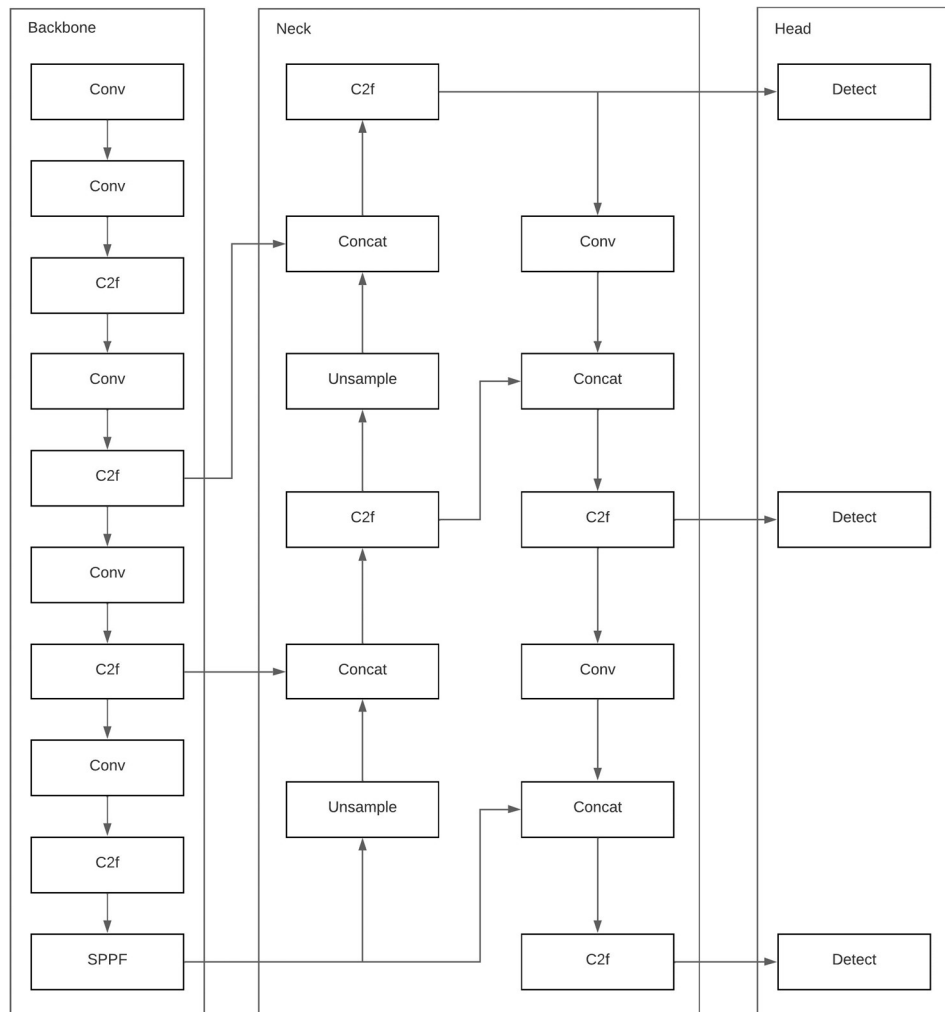


Рисунок В.3 – Структура згорткової нейронної мережі

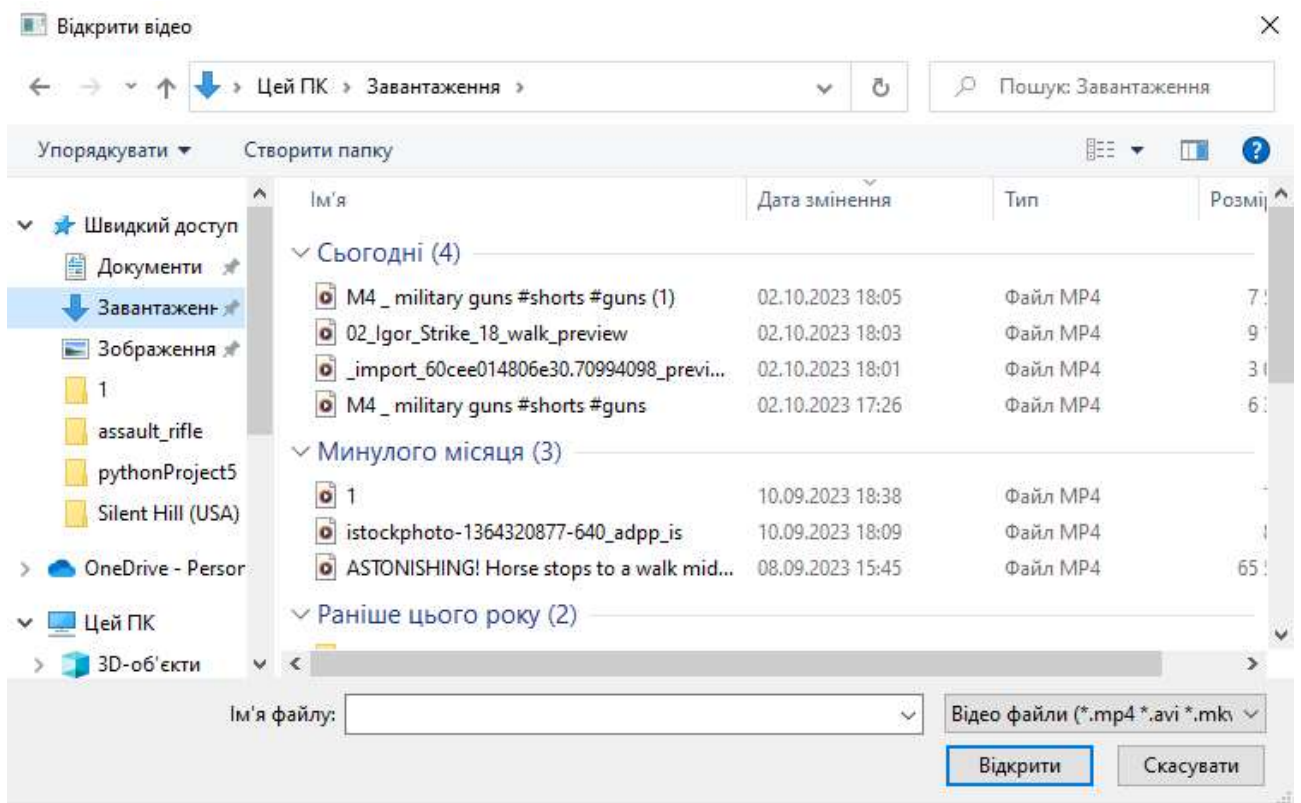
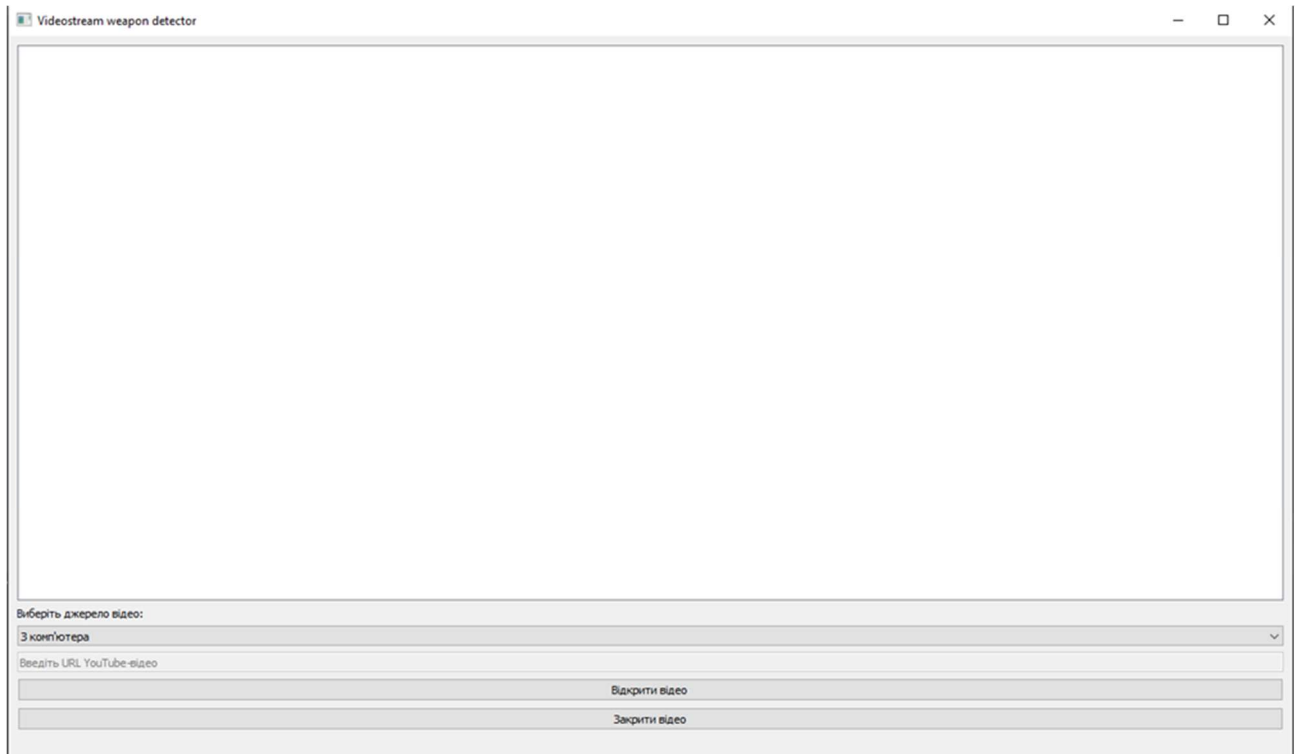


Рисунок В.4 – Робочі вікна програми виявлення озброєних людей

	pistol	rifle	shotgun	machinegun
pistol	196	0	2	5
rifle	2	195	3	3
shotgun	2	2	194	2
machinegun	0	3	1	192

	Аналог(Weapon detection)	Запропонована інформаційна технологія
Достовірність	93,7%	98,51%
Влучність	94,1%	97,13%
Повнота	93,2%	96,9%

Рисунок В.5 – Результати роботи програми розпізнавання озброєних людей та порівняння з аналогом

Додаток Г (довідниковий)

Інструкція користувача

Після запуску програми відкривається головне вікно програми (рис. Г.1). Головне вікно має такі елементи керування: випадаючий список для вибору джерела відеопотоку та кнопки для відкриття та закриття відеопотоку.

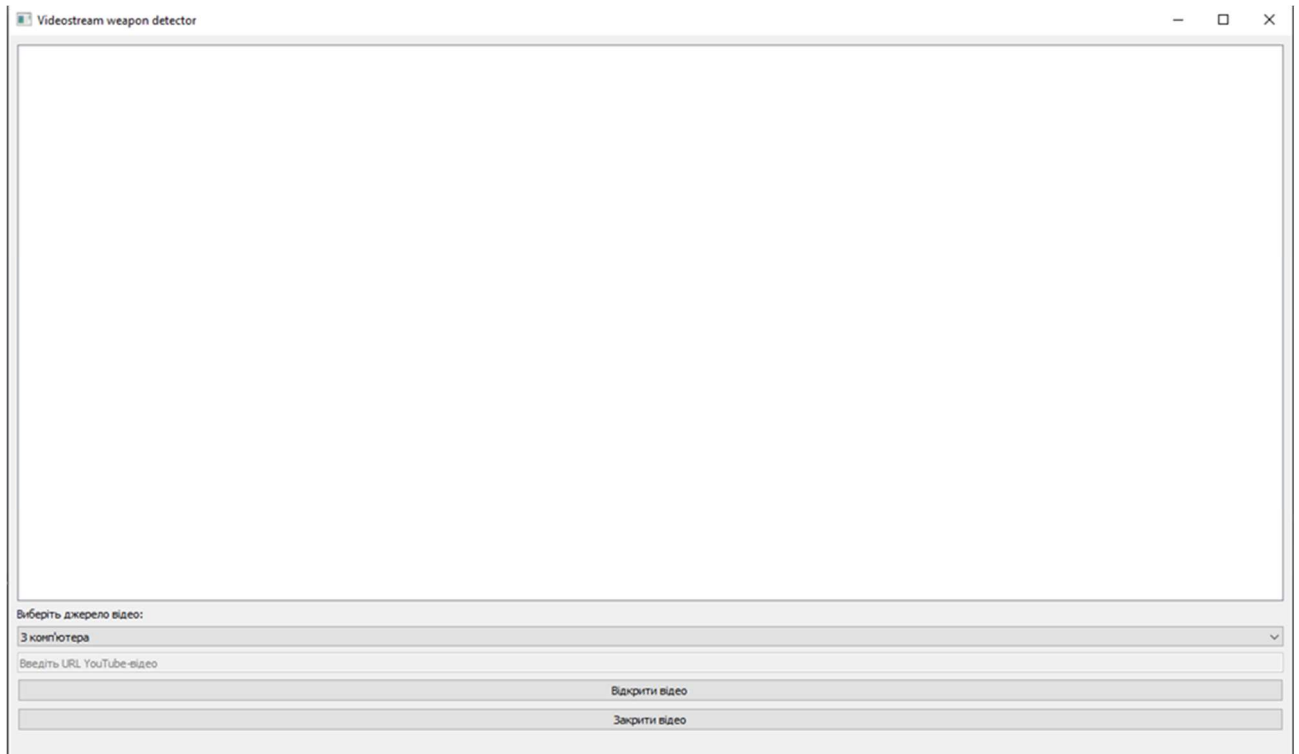


Рисунок Г.1 – Головне вікно програми

Випадаючий список дозволяє користувачу вибрати такі джерела відеопотоку: З комп'ютера, З YouTube та З камери. Якщо вибрано джерело відеопотоку З комп'ютера, після натискання на кнопку Відкрити відео, відкривається діалогове вікно (рис. Г.2) для вибору відеопотоку необхідного для обробки. Якщо вибрано джерело відеопотоку З YouTube, необхідно в текстове поле (рис. Г.3) ввести адресу відеопотоку необхідного для обробки. Якщо вибрано джерело відеопотоку З камери, відеопотік береть з веб-камери.

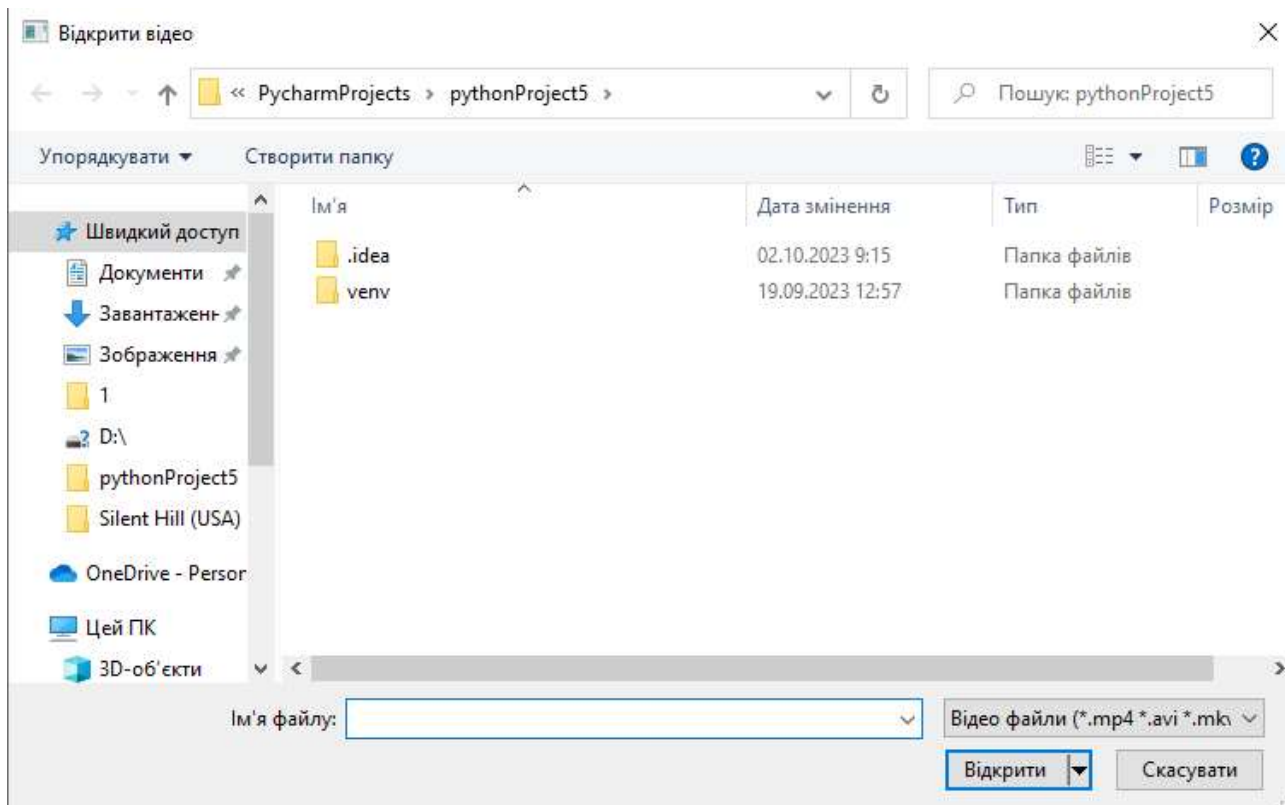


Рисунок Г.2 – Діалогове вікно для вибору відеопотоку

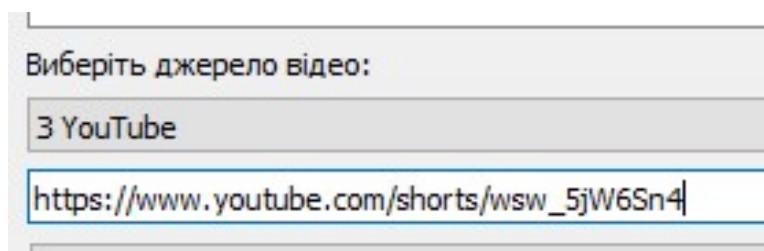
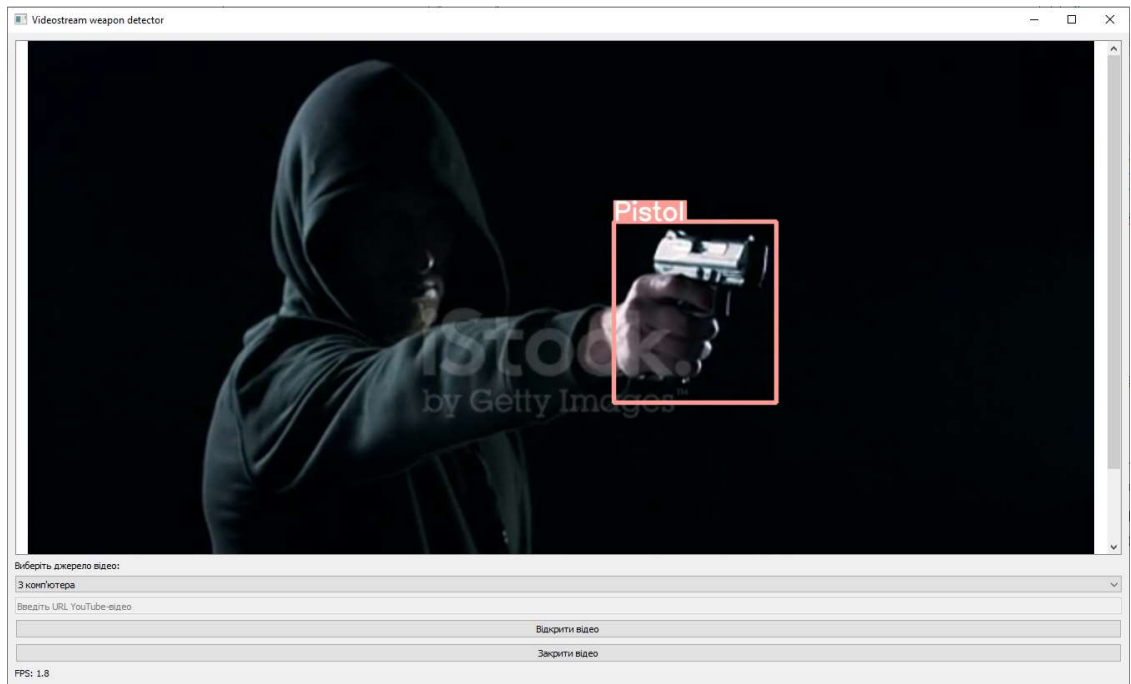
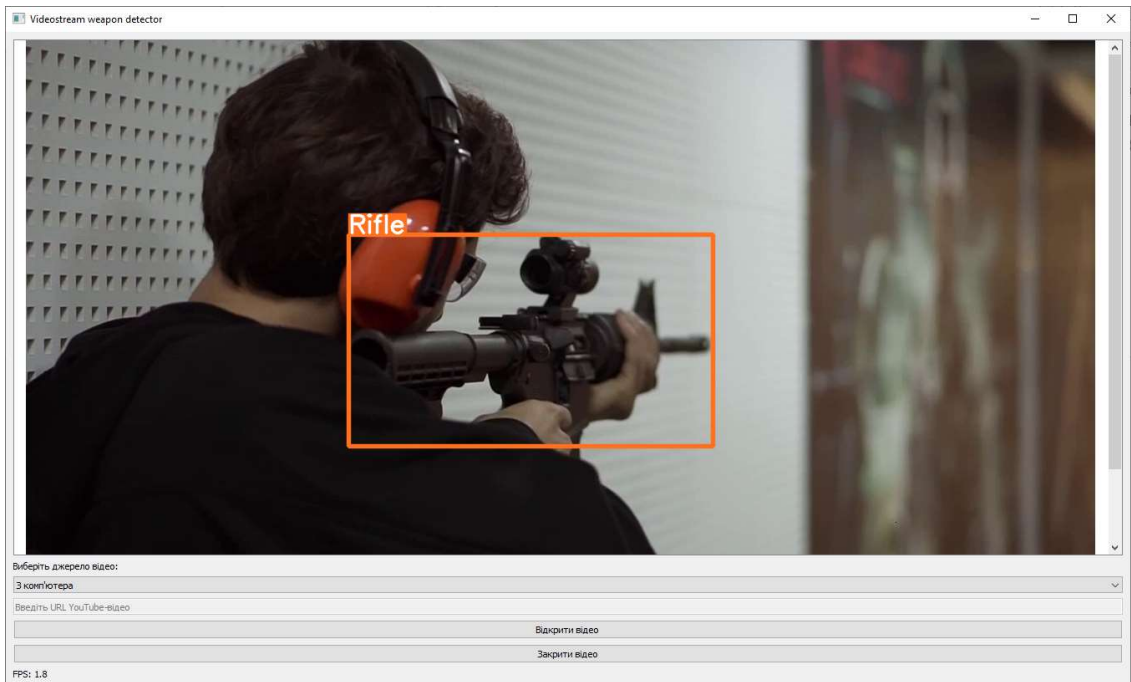


Рисунок Г.3 – Текстове поле вводу адреси відеопотоку

Отриманий відеопотік розбивається на зображення, які подаються на нейронну мережу, результати розпізнавання виводяться на головне вікно у вигляді відеопотоку. Якщо на зображенні в відеопотоці виявлено зброю, виявлена зброя виділяється і виводиться клас до якого зброя належить. Результати розпізнавання зброї показано на рисунку Г.4 (а, б, в, г).



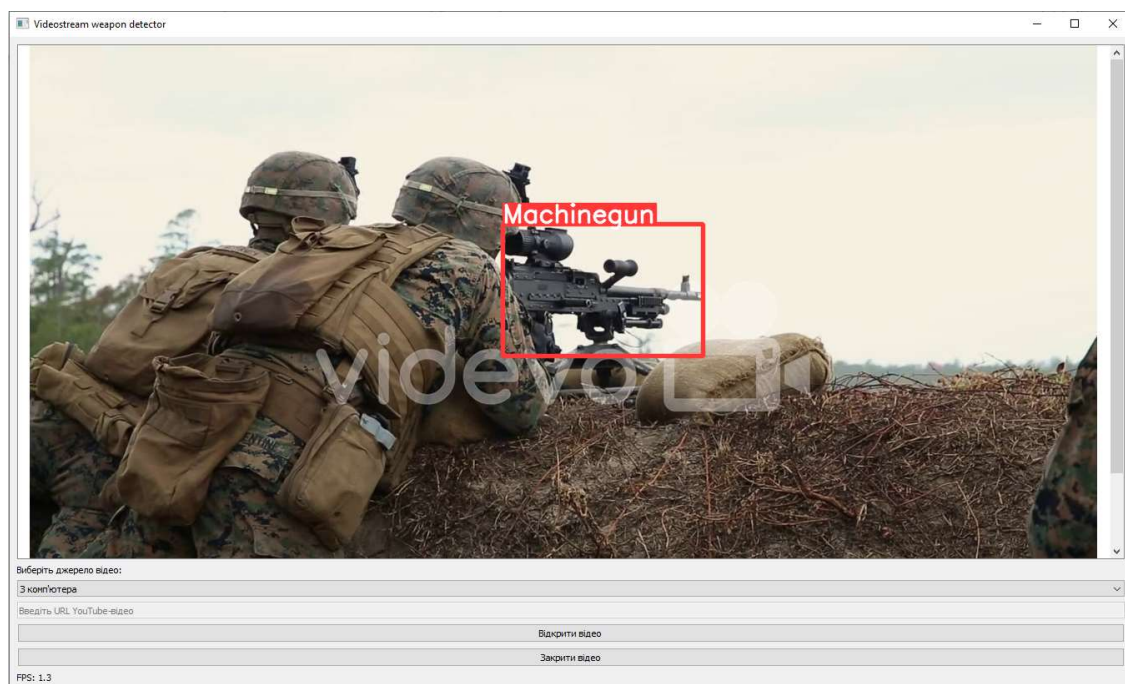
(a)



(b)



(В)



(Г)

Рисунок Г.4 – Результати виявлення зброї:

(а) – клас «пістолет»; (б) – клас «гвинтівка»; (в) – клас «дробовик»; (г) – клас «кулемет»

Процес виявлення зброї продовжується доки відеопотік не закінчиться або доки не буде натиснуто кнопку Закрити відео.