

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

Пояснювальна записка

до комплексної магістерської кваліфікаційної роботи

на тему «Інформаційна технологія розпізнавання нетипових ситуацій на відео: детектування об'єктів»

Виконав: студент 2 курсу,
групи 2КН-19 м
спеціальності 122 «Комп'ютерні науки»
Преподобний Мендеш да Майа А. А.

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

Рецензент: к. т. н., доц. Рейда О. М.

Вінниця
2020

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

(підпис)
“ _____ ” _____ 2020 року

ЗАВДАННЯ

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

08-22.МКР.026.19.000.ПЗ

Магістранта групи 2КН-19м Преподобного Мендеша да Майа Артура Амадейовича

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія розпізнавання нетипових ситуацій на відео: детектування об'єктів»

Вхідні дані: Формат вхідного відео - MP4, AVI, MKV; потік фреймів з камери формату AHD, CVI, TVI, IP мінімального піксельного розміру 448x448, об'єм навчальної вибірки - не менше 100 зображень; об'єм контрольної вибірки - не менше 100 зображень; вихідні дані - чітке вихідне зображення, на якому зображені об'єкти, набір визначених об'єктів – список з координатами та розмірами, налаштування користувача – значення порогів для визначення нетипових ситуацій.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: загальна структурна схема програми, структура згорткової нейронної мережі, структура інформаційної технології розпізнавання нетипових ситуацій на відео у частині детектування об'єктів, часова діаграма активності програми, приклади роботи програми.

2. Текстова (пояснювальна записка): вступ, аналіз предметної області детектування об'єктів на відео для розпізнавання нетипових ситуацій, розробка інформаційної технології розпізнавання нетипових ситуацій у частині детектування об'єктів на відео, програмна реалізація інформаційної технології розпізнавання нетипових ситуацій у частині детектування об'єктів на відео, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного рівня інформаційних технологій детектування об'єктів на відео. Постановка задач дослідження			Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Побудова математичних моделей функціонування нейронної мережі			Математичні моделі, розділ 2
3	Практичне застосування та оцінка ефективності розроблених моделей			розділ 3
4	Підготовка економічної частини			розділ 4
5	Апробація та/або впровадження результатів дослідження			тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____ О. К. Колесницький
ініціали та прізвище

2. Економічна частина _____ канд. екон. наук, доц., доц. каф. ЕПВМ
(підпис) наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____ М. В. Бальзан
ініціали та прізвище

Дата попереднього захисту роботи “ ____ ” _____ 20__ р.

Рецензент _____ канд. техн. наук, доц., доц. кафедри ПЗ
(підпис) наук. ступінь, вчене звання (посада)
 _____ О. М. Рейда
ініціали та прізвище

Завдання видав науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
 _____ О. К. Колесницький
ініціали та прізвище
 “ ____ ” _____ 20__ р.

Завдання отримав магістрант _____ А. А. Преподобний
(підпис) ініціали та прізвище
 “ ____ ” _____ 20__ р.

АНОТАЦІЯ

Дана комплексна магістерська кваліфікаційна робота присвячена розробці інтелектуальної інформаційної технології та програмного забезпечення розпізнавання нетипових ситуацій на відео для моніторингу дорожнього руху і автоматизованого виявлення аварійних випадків. У цій частині розробляються засоби детектування об'єктів (транспортних засобів) на відео. В основі технології лежить згортова нейронна мережа, для якої розроблено архітектуру на основі моделі YOLO. В ході практичної реалізації інформаційної технології було обрано мову програмування Python та середовище програмування Visual Studio Code. Серед бібліотек та фреймворків для детектування об'єктів були обрані OpenCV, Tensorflow та Darknet.. Розроблене програмне забезпечення розпізнавання нетипових ситуацій на відео порівняно з аналогом має кращу на 4% достовірність детектування окремих об'єктів і кращу на 3% достовірність детектування об'єктів по відео в цілому.

ABSTRACT

This complex master's degree work is devoted to the development of intelligent information technology and software for recognizing atypical situations on video for traffic monitoring and automated accident detection. This section develops tools for detecting objects (vehicles) on video. The technology is based on a convolutional neural network, for which an architecture based on the YOLO model has been developed. During the practical implementation of information technology, the Python programming language and the Visual Studio Code programming environment were chosen. OpenCV, Tensorflow and Darknet were selected as libraries and frameworks for object detection. The developed software for recognizing atypical situations on video in comparison with the analogue has 4% better detection accuracy of individual objects and 3% better detection accuracy of object detection on video as a whole..

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ НА ВІДЕО ДЛЯ РОЗПІЗНАВАННЯ НЕТИПОВИХ СИТУАЦІЙ.....	13
1.1 Постановка задачі детектування об'єктів на відео при розпізнаванні нетипових ситуацій	13
1.2 Огляд відомих методів детектування об'єктів на відео та обґрунтування вибору нейромережевого методу	14
1.3 Обґрунтування вибору аналогу до програмного забезпечення детектування об'єктів на відео	17
1.4 Висновок.....	21
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ НЕТИПОВИХ СИТУАЦІЙ У ЧАСТИНІ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ НА ВІДЕО	22
2.1 Обґрунтування вибору типу нейронної мережі	22
2.2 Розробка архітектури згорткової нейронної мережі	24
2.3 Метод навчання згорткової нейронної мережі	31
2.4 Структура інформаційної технології розпізнавання нетипових ситуацій на відео у частині детектування об'єктів.....	33
2.5 Розробка структурної схеми програмної системи розпізнавання нетипових ситуацій на відео	35
2.6 Висновок.....	38
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ НЕТИПОВИХ СИТУАЦІЙ У ЧАСТИНІ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ НА ВІДЕО	39
3.1 Обґрунтування вибору мови програмування	39
3.2 Вибір бібліотек та фреймворків для детектування об'єктів	41
3.3 Розробка UML діаграм програми виявлення нетипових ситуацій	47
3.4 Програмна реалізація процесу навчання нейронної мережі.....	49
3.5 Розробка навчальної вибірки для нейронної мережі	53

3.6 Тестування та аналіз результатів роботи програмного засобу розпізнавання нетипових ситуацій у частині детектування об'єктів на відео	55
3.7 Висновок	58
4 ЕКОНОМІЧНА ЧАСТИНА.....	59
4.1 Оцінювання комерційного потенціалу розробки.....	59
4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи	62
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки	65
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності ..	67
4.5 Висновок	70
ВИСНОВКИ.....	71
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	73
Додаток А Інструкція користувача.....	76
Додаток Б Лістинг програми.....	80
Додаток В Графічна частина.....	85

ВСТУП

Актуальність теми дослідження. 21 століття називають віком інформаційних технологій, така негласна назва утворилась тому що розвиток науки та техніки переживає значний стрибок у розвитку. Автоматизація та комп'ютеризація процесів все більше і більше вплітається в повсякденне життя.

Проект спостереження за транспортним рухом є зовсім не новим. Вже доволі немалий період часу в багатьох містах світу розташовують камери відеоспостереження, які записують рух транспорту по дорогам. Але для аналізу отриманих відео записів необхідно час, дуже багато часу. В ідеалі необхідно щоб людина яка проглядає відео не відволікалась від відео, щоб максимально знизити можливість пропустити щось не типове. Тому маємо просту арифметику, скільки камер, стільки і людей. Очевидно що дана система не є життєздатною. Одна камера може захопити в кадр досить малий, відносно всіх доріг міста, шматок дороги, тому для повноцінної роботи потрібно багато камер. І як наслідок багато працівників, що візьмуть на себе обов'язок слідкувати за цими камерами.

Для підтримання роботи будь-яка система потребує ресурсів. В нашому випадку необхідні фінанси на постачання енергії для роботи обладнання та оплату праці досить великого штату робітників. Для наближення до реальності та зменшення витрат використовують декілька ідей. Серед них оптимізація розташування камер, оптимізація режимів роботи камер, збільшення камер за які відповідає одна особа.

На даний момент вважається що одна особа може відповідати за 8 камер, без значного ризику пропустить щось важливе. Дана методика дає змогу зменшити необхідні витрати на оплату праці робітників. Але використовуючи по максимуму інформаційні технології можливо отримати що кращі результати.

Інформаційні технології можна використовувати для аналізу відео та пошуку нетипових ситуацій. Це зменшить необхідність постійного контролю з боку людини та надасть додаткові методи оптимізації процесу

Отже, задача виявлення нетипових ситуацій носить важливий прикладний характер, і для її розв'язання відсутні достатньо ефективні класичні методи, що надає можливість застосування засобів штучного інтелекту для досягнення кращих показників достовірності та швидкодії виявлення нетипових ситуацій.

Тема магістерської кваліфікаційної роботи є актуальною, оскільки на даний момент існує потреба у автоматизованих системах спостереження та виявлення нетипових ситуацій. Дана технологія буде корисною для державних та приватних установ та служб, особливо в умовах значної кількості камер та обмеженого ресурсу персоналу.

Використання методів штучного інтелекту для розв'язання поставленої задачі є актуальним через те, що надає великі можливості для модифікацій і налаштування різних методів під дану задачу, а отже дозволяє таким чином досягати кращий у порівнянні з традиційними існуючими методами результат.

Таким чином, застосування для розв'язання задачі виявлення нетипових ситуацій сучасних методів штучного інтелекту сприяє подальшому розвитку комп'ютерного зору та ряду інших суміжних систем, а також є темою подальших дослідження нових методів використання штучного інтелекту.

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

Мета і завдання досліджень. Метою магістерської кваліфікаційної роботи є підвищення достовірності та швидкодії детектування об'єктів на відео для розпізнавання нетипових ситуацій за рахунок застосування згорткових нейронних мереж.

Для досягнення мети розробки необхідно виконати такі задачі:

– провести аналіз проблеми детектування об'єктів при виявленні нетипових ситуацій;

- розглянути існуючі методи вирішення задачі детектування об'єктів при розпізнаванні нетипових ситуацій та обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- розробити архітектуру згорткової нейронної мережі для детектування об'єктів при розпізнаванні нетипових ситуацій та удосконалити її згідно з метою роботи;
- сформулювати стадії інформаційної технології та на їх основі розробити структуру та алгоритм роботи програмного засобу;
- виконати програмну реалізацію запропонованої інформаційної технології розпізнаванні нетипових ситуацій в частині детектування об'єктів;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

Об'єкт дослідження – процес детектування об'єктів для розпізнавання нетипових ситуацій на відео з використанням штучних нейронних мереж.

Предмет дослідження – інформаційна технологія та програмні засоби розпізнавання нетипових ситуацій на відео в частині детектування об'єктів з використанням згорткових нейронних мереж та достовірність їх роботи.

Методи дослідження. У роботі використані наступні методи наукових досліджень: системний аналіз для аналізу структури інформаційної системи, теорія штучних нейронних мереж для реалізації інформаційної технології розпізнавання нетипових ситуацій на відео, методи математичної статистики для обробки результатів тестування програми, об'єктно-орієнтованого програмування для реалізації програмного продукту..

Наукова новизна одержаних результатів.

1. Вперше запропоновано інформаційну технологію розпізнавання нетипових ситуацій на відео, яка використовує для детектування та трекінгу об'єктів глибоку згорткову нейронну мережу, що дозволило підвищити достовірність розпізнавання нетипових ситуацій на відео.

2. Удосконалено архітектуру згорткової нейронної мережі, яка відрізняється введенням додаткового згорткового фільтра для передбачення

класу об'єкта, що дозволило підвищити достовірність детектування об'єктів на відео.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення розпізнавання нетипових ситуацій на відео.

Запропонована інформаційна технологія сприяє підвищенню достовірності процесу розпізнавання нетипових ситуацій на відео:

- розроблено структурну схему програмного забезпечення розпізнавання нетипових ситуацій на відео на основі згорткової нейронної мережі в частині детектування об'єктів;

- розроблено програмні засоби для розпізнавання нетипових ситуацій на відео на основі згорткової нейронної мережі в частині детектування об'єктів на відео.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології розпізнавання нетипових ситуацій на відео в частині детектування та трекінгу об'єктів;. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз методів детектування та трекінгу об'єктів; [1], аналіз процесу детектування та трекінгу об'єктів на відео на основі згорткової нейронної мережі [2,3] та методів підвищення достовірності розпізнавання нетипових ситуацій на відео [4].

Апробація результатів роботи. Результати досліджень апробовані на XII Міжнародній науково-практичній конференції ІОН-2020, 26-29 травня, 2020, Вінниця: ВНТУ [4], а також на XLVIII Науково-технічній конференції

підрозділів Вінницького національного технічного університету (2019)», Вінниця, 2019 [2], та на XLIX Науково-технічній конференції підрозділів Вінницького національного технічного університету (2020)», Вінниця, 2020 [3].

Публікації. За результатами досліджень опубліковано статтю у фаховому журналі [1]., троє тез доповідей на науково-технічних конференціях [2,3,4] та подано заявку на авторське свідоцтво на твір (програму).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ НА ВІДЕО ДЛЯ РОЗПІЗНАВАННЯ НЕТИПОВИХ СИТУАЦІЙ

1.1 Постановка задачі детектування об'єктів на відео при розпізнаванні нетипових ситуацій

Детектування об'єктів являє собою окремий випадок загального процесу розпізнавання образів.

Розпізнавання образів – процес, при якому на підставі численних характеристик (ознак) деякого об'єкта визначаються одна або декілька найбільш істотних, але недоступних для безпосереднього визначення, його характеристик, зокрема його приналежність до певного класу об'єктів.

Розпізнавання образів один з головних розділів кібернетики, що розвиває теоретичні основи й методи класифікації і ідентифікації предметів, явищ, процесів, сигналів, ситуацій і т. п. об'єктів, які характеризуються кінцевим набором деяких властивостей і ознак.

Замість терміну “розпізнавання” можна вживати інший – “класифікація”. Ці два терміни у багатьох випадках розглядаються як синоніми, але не є повністю взаємозамінюваними. Кожний з них має свої сфери застосування, і інтерпретація обох термінів часто залежить від специфіки конкретної задачі. Класифікація є більш широким терміном, так як в даному випадку необхідно окрім знаходження об'єктів ще і зрозуміти що саме це за об'єкт та віднести його до певного класу чи типу за рядом ознак. Тому в даній роботі використаємо саме термін розпізнавання так як потрібно шукати лише певний один тип образів, пошук додаткових образів лише збільшить навантаження на апаратне забезпечення та не приведе до значних покращень роботи програми..

Наведемо деякі типові постановки задач класифікації:

Задача ідентифікації, яка полягає в тому, щоб вирізнити певний конкретний об'єкт серед йому подібних.

Даний тип постановки задачі не підходить так як необхідно шукати не якийсь один конкретний предмет а цілий вид предметів по заданим ознакам.

Кластерний аналіз, який полягає в поділі заданого набору об'єктів на класи – групи об'єктів, схожі між собою за тим чи іншим критерієм. Ця постановка задачі має на увазі декілька класів різних об'єктів. Оскільки ми шукаємо лише один тип об'єктів дане формулювання задачі нам не підходить.[5]

Типізація об'єктів. Віднесення об'єкта до того чи іншого типу. Це може бути, наприклад, задача розпізнавання літер або прийняття рішення про наявність дефекту у деякій технічній деталі. Віднесення об'єкта до певного класу відображає найтипівішу проблему класифікації, і, коли говорять про розпізнавання образів, найчастіше мають на увазі саме цю проблему. Дане формулювання найбільш точно відображає потрібну нам задачу. Тому що вхідними даними нашої програми є відеоряд чи статична картинка, на якій зображено проїзна частина, тому на ній можуть бути присутні різні об'єкти такі як люди, дерева, автомобілі тощо. Серед усього різноманіття об'єктів програма має виділити лише один тип образів автомобілі.

1.2 Огляд відомих методів детектування об'єктів на відео та обґрунтування вибору нейромережевого методу

Для виконання поставлених задач можна використати такі методи:

1) Метод потенціалів.

Щоб цей метод можна було використовувати для розпізнавання образів потрібно зробити деякі введення в стандартний метод. Нехай джерелами потенціалів є група точок, які відповідають деякому образу а. Тоді можна вважати, що середній потенціал, які створюють у деякій точці простору джерела цього образу, характеризує віддаль від цієї точки до усього образу у цілому. Спираючись на гіпотезу компактності, можна запропонувати наступне правило розпізнавання:

Точку відносимо до того образу, середній потенціал якого у цій точці є максимальним. Даний видозмінений метод дасть змогу розпізнати об'єкт. Але, оскільки в нашій інформаційній технології потрібно розпізнавати образи автомобілів на дорозі, то в багатьох випадках автомобілі будуть знаходитись досить близько чи частково перекривати один одного. Тому даний метод не зможе в таких випадках розпізнати дві машини як окремі об'єкти.

2) Метод нормалізації плоских зображень.

Також для виконання поставлених задач можна використати метод нормалізації плоских зображень. Оскільки в даному методі зображення розкладається і представляється як множина проєкцій то ми не зможемо розпізнавати автомобілі, які не знаходяться на різній відстані від камери що їх знімала так як для роботи такої системи необхідно буде прорахувати усі можливі групи проєкцій для різних форм автомобілів, їхніх точок перспектив та випадків віддаленості від камери тому дана множина прямуватиме до нескінченності.[6]

3) Структурний метод.

Ще одним методом вирішення задачі класифікації образів є структурний метод. Суть цього методу полягає у накопиченні напрямків векторів для утворення складної кривої. Приклад на рис. 1.1

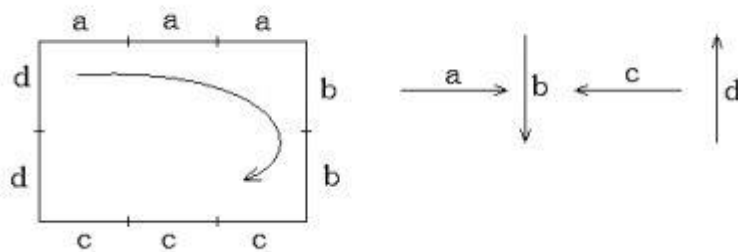


Рисунок 1.1 - Приклад розкладання кривої на вектори

Цей метод для пошуку будь якого об'єкту можна використати через те що будь який складний об'єкт являється набором простих структур.

Використовуючи алгоритми цього методу можливо розпізнати потрібні нам об'єкти та завдяки тому що більшість машин мають досить багато спільного можна з деякою точністю стверджувати що всі потрібні нам образи можливо розкласти на одні й ті самі прості структури.

Недоліками даного методу є те що якщо хоча б незначна частина шуканого об'єкту буде схована за перешкодою, ми не зможемо отримати потрібний нам результат.

4) Аналітичний Метод.

Також для пошуку об'єкту на зображенні можна застосувати аналітичний метод перебору. При цьому методі шукатимемо контури різних геометричних фігур та за їх взаємним розташуванні та відношенням можна зробити висновок про те чи є шуканим об'єктом даний набір знайдених контурів. Недоліками цього методу є його складність та велике навантаження на апаратне забезпечення. Все одно потрібно використовувати додаткові методи для пошуку геометричних фігур та подальший аналіз також не буде вважатись досить точним при великому нагромадженні об'єктів [7].

5) Нейронні мережі.

Поставлені задачі виконуватимуться при використанні нейромережевого підходу штучного інтелекту. В даному випадку для розв'язання задачі буде використана штучна нейронна мережа, яка має змогу шукати об'єкти різних масштабів та відокремлювати декілька близько розташованих однакових об'єктів [8]. Дана технологія також називається комп'ютерний зір.

Обробка зображень і аналіз зображень в основному зосереджені на роботі з двовимірними зображеннями, тобто як перетворити одне зображення в інше. В проєктованій нейронній мережі було використано попіксельні операції збільшення контрастності, операції з виділення країв, усунення шумів. Дані операції присутні тому, що обробка/аналіз зображення діють незалежно від вмісту самих зображень [8].

Використання комп'ютерного зору необхідне для розроблюваного інтелектуальної технології в розв'язанні таких задач як:

- Розпізнавання: один чи декілька попередньо заданих чи вивчених об'єктів або класів об'єктів можуть бути розпізнані, зазвичай разом з їх двовимірним положенням на зображенні чи тривимірним положенням в сцені.
- Ідентифікація: розпізнається індивідуальний екземпляр об'єкта. Приклади: ідентифікація визначеного людського обличчя чи відбитка пальців чи автомобіля.

Розглянувши відомі методи детектування об'єктів на відео, а також їх переваги і недоліки, оберемо для використання у даній МКР саме нейромережеві методи детектування об'єктів як найбільш перспективні.

1.3 Обґрунтування вибору аналогу до програмного забезпечення детектування об'єктів на відео

На сьогоднішній день існують програми які збирають різноманітні дані з камер відеоспостереження, ці дані збираються в спеціальні бази даних та на їх основі можливо аналітично робити деякі висновки. До таких проектів належать: Analytics, Hikvision, ADRTCMS

1. Analytics – проект створений для розпізнавання, трекінгу та відслідковування об'єктів [9].

Даний проект дозволяє перетворити будь-яку камеру, яка працює в системі Latitude VMS, в інтелектуальну і аналізувати відео потік в режимі реального часу, незалежно від виробника. Завдяки прописаними правилами і алгоритмами аналізу, система гарантує роботу навіть в найскладніших умовах і дозволяє персоналу ситуаційного центру точно визначати і реагувати на виникаючі події.

Приклад роботи даної програми відображено на рис.1.2

Запропоноване рішення здатне ефективно фільтрувати фонові перешкоди і ігнорувати непотрібні рухи або об'єкти в зоні відстеження. На цих факторах система може навчатися, підвищуючи якість розпізнавання подій і зводячи до

мінімуму кількість помилкових спрацьовувань. Правило Dependency забезпечує високу ймовірність виявлення події з вкрай низьким рівнем помилкових тривог.



Рисунок 1.2 - Інтерфейс працюючої програми Analytics

Даний проект є досить перспективним але його напрямок відрізняється від поставлених нами цілей.

2. Іншим проектом даної тематики є Hikvision [10]. Даний проект спеціалізується на відстеженні автомобільних номерів. Проект Hikvision заснувала одноіменна компанія-виробник камер відеоспостереження. Приклад роботи даної програми відображено на рис.1.3

Компанія Hikvision, лідируючий виробник обладнання для відеоспостереження, представляє професійні мережеві відеокамери з апаратним функціоналом відео аналітики. 3-мегапіксельна серія професійних відеокамер за допомогою інтегрованого Smart кодека здійснює захоплення відео з високою якістю зображення і передає HD потік зі швидкістю 50 кадрів / с. Повний набір функцій відео аналітики, таких як: визначення напрямку руху, виявлення вторгнень, виявлення перетину загородження, виявлення осіб людей, підрахунок об'єктів - дозволять тонко налаштувати камеру під різні завдання. Проект дає змогу легко розрізняти автомобілі один від одного, але через те, що

він відслідковує лише номери, він стає неефективним у відслідковуванні аварійних ситуацій.

The screenshot displays the Nikvision software interface. On the left, there are two video feeds showing a snowy outdoor area with a fence. The top feed shows a license plate 'E234MK 35' and the bottom feed shows 'У973ОВ 178'. On the right, a search result for 'У973ОВ 178' is shown, including the date and time '15.02.2018 12:51:58' and the vehicle model 'Beward B2710R'. Below this is a table titled 'ЖУРНАЛ РАСПОЗНАВАНИЯ' (Recognition Log) with columns for license plate number, date, time, video channel, user, server, duration, direction, and vehicle name. The table contains several rows of data, including entries for license plates like 'E234MK35', 'У973ОВ178', and 'У387Н435'.

Номер ТС	Дата/Время	Видеоканал	Пользователь	Сервер	Длительность пребывания	Направление	Наименование	Год выпуска а/т	Государство	Ф.И.О. водителя
E234MK35	1411:39 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
B411H435	1405:23 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
E191YK35	1400:02 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
B466YK35	1342:04 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
E234MK35	1337:18 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889	1 дн., 4 час., 18 мин., 3 сек.	Выезд				
E278YK35	1324:56 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
B372H435	1303:21 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889-2	2 дн., 4 час., 20 мин., 30 сек.	Выезд				
У973ОВ178	1303:10 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
У973ОВ178	1302:07 15.02.2018	MDC-M629070R	Администратор	DESKTOP-190889	11 мин., 8 сек.	Долгой				
E191YK35	1252:36 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
E191YK35	1252:34 15.02.2018	MDC-M629070R	Администратор	DESKTOP-190889	2 дн., 22 час., 31 мин., 48 сек.	Долгой				
У973ОВ178	1251:58 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
B411H435	1234:44 15.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
B411H435	1234:41 15.02.2018	MDC-M629070R	Администратор	DESKTOP-190889	1 дн., 3 час., 18 мин., 23 сек.	Долгой				
T177H435	1042:03 14.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				
У387Н435	1020:06 14.02.2018	Beward B2710R	Администратор	DESKTOP-190889		Выезд				

Рисунок 1.3 - Процесс работы программы Nikvision

3. ADRTCMS – У даній роботі представлена автоматична система контролю дорожнього руху, що використовує камеру В & W. Важлива інформація про дорожній рух, така як середня швидкість, розмірність та підрахунок транспортних засобів, отримується за допомогою методів комп'ютерного зору. По-перше, рухомі об'єкти витягуються зі сцени за допомогою алгоритму диференціювання кадру і інформації про текстуру на основі інтенсивності шкали сірого. Проте тіні рухомих об'єктів також належать до переднього плану. Тіні видаляються з об'єктів переднього плану за допомогою перетворень та морфологічних операторів. Нарешті, об'єкти відслідковуються в процесі фільтрації Калмана, і вимірюються такі параметри, як положення, розміри, відстань і швидкість рухомих об'єктів. Потім згідно з цими параметрами рухомі об'єкти класифікуються як транспортні засоби (вантажівки або автомобілі) або неприємні артефакти.

Для візуалізації результатів 3D-модель проектується на транспортні засоби в площині зображення. Показано деякі експериментальні результати з використанням реальних відкритих послідовностей зображень. Ці результати демонструють точність запропонованої системи при денних міжміських умовах руху [11]. На рисунку 1.4 зображено сайт IEEE з якого було взято дану статтю.

The image shows a screenshot of the IEEE Xplore digital library page for the article "Automatic Daytime Road Traffic Control and Monitoring System". The page includes the title, authors (P.F. Alcantarilla, M.A. Sotelo, L.M. Bergasa), and a list of document sections. The abstract describes a system for road traffic control using computer vision. The page also displays citation statistics (10 paper citations, 396 full text views) and various icons for downloading and sharing the document.

Abstract	Abstract: This paper presents an automatic road traffic control and monitoring system for day time sequences using a B & W camera. Important road traffic information such as mean speed, dimension and vehicles counting are obtained using computer vision methods. Firstly, moving objects are extracted from the scene by means of a frame-differencing algorithm and texture information based on grey scale intensity. However, shadows of moving objects belong also to the foreground. Shadows are removed from the foreground objects using top hat transformations and morphological operators. Finally, objects are tracked in a Kalman filtering process, and parameters such as position, dimensions, distance and speed of moving objects are measured. Then, according to these parameters moving objects are classified as vehicles (trucks or cars) or nuisance artifacts. For results visualization, a 3D model is projected onto vehicles in the image plane. Some experimental results using real outdoor sequences of images are shown. These results demonstrate the accuracy of the proposed system under daytime interurban traffic conditions.	
Document Sections	<ul style="list-style-type: none"> I. Introduction II. Initialization Setup III. Moving Objects Extraction IV. Shadows Detection VI. Perspective Camera Model 	
Authors	Published in: 2008 11th International IEEE Conference on Intelligent Transportation Systems	
Figures	Date of Conference: 12-15 Oct. 2008	INSPEC Accession Number: 10452258
References	Date Added to IEEE Xplore: 30 December 2008	DOI: 10.1109/ITSC.2008.4732530
Citations	▶ ISBN Information:	Publisher: IEEE
Keywords	▶ ISSN Information:	Conference Location: Beijing, China

Рисунок 1.4 - Зображення сайту IEEE

Даний проект є найперспективнішим з трьох представлених та, спираючись на його опис, можна зробити висновок, що він задовольняє цілі, заради яких створювалася наша інформаційна технологія. Але даний проект так і не вийшов у реліз та був зупинений на стадії розробки.

1.4 Висновок

В першому розділі було сформульовано постановку задачі детектування об'єктів на відео при розпізнаванні нетипових ситуацій. Було проведено аналітичний огляд відомих методів і засобів детектування об'єктів на відео при розпізнаванні нетипових ситуацій, виявлені їх можливості, переваги і недоліки. Обґрунтована перспективність застосування методу детектування об'єктів на відео на основі штучних нейронних мереж. Також було проаналізовано існуючі програми-аналоги до програмного забезпечення детектування об'єктів на відео.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ НЕТИПОВИХ СИТУАЦІЙ У ЧАСТИНІ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ НА ВІДЕО

2.1 Обґрунтування вибору типу нейронної мережі

Даний компонент повинен відповідати за виявлення автомобілів на оброблених фреймах з потоку. Для нього можна використати такі нейронні мережі як рекурентна нейронна мережа, нейронна мережа Гопфілда та згортова нейронна мережа.

Рекурентні нейронні мережі — це клас штучних нейронних мереж, у якому з'єднання між вузлами утворюють орієнтований цикл. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічну поведінку в часі. На відміну від нейронних мереж прямого поширення, РНМ можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей входів. Але як результат даних можливостей виникає велике навантаження на потужності комп'ютера тому її використання не є можливим при обробці відео [12].

Нейронна мережа Гопфілда — це тип рекурентної, повнозв'язної, штучної нейронної мережі з симетричною матрицею зв'язків. У процесі роботи динаміка таких мереж сходиться (конвергує) до одного з положень рівноваги. Ці положення рівноваги є локальними мінімумами функціоналу, що називається енергія мережі (у найпростішому випадку — локальними мінімумами негативно певної квадратичної форми на n -вимірному кубі). Така мережа може бути використана як автоасоціативна пам'ять, як фільтр, а також для розв'язання деяких завдань оптимізації. Але оскільки вона де факто являється різновидом рекурентних мереж тому має її недоліки, а також її використання не є доцільним при незмінному класі шуканих об'єктів. Тому дана мережа також не підходить для використання в даному проекті.

Згорткова нейронна мережа (ЗНМ) складається з шарів входу та виходу, а також із декількох прихованих шарів. Приховані шари ЗНМ зазвичай складаються зі згорткових шарів, агрегувальних шарів, повноз'єднаних шарів та шарів нормалізації [13].

Цей процес описують в нейронних мережах як згортку за домовленістю. З математичної точки зору вона є радше взаємною кореляцією, ніж згорткою. Це має значення лише для індексів у матриці, й відтак які ваги на якому індексі розташовуються.

Згорткові шари застосовують до входу операцію згортки, передаючи результат до наступного шару. Згортка імітує реакцію окремого нейрону на зоровий стимул. Кожен згортковий нейрон обробляє дані лише для свого рецептивного поля.

Хоч повнозв'язні нейронні мережі прямого поширення й можливо застосовувати як для навчання ознак, так і для класифікування даних, застосування цієї архітектури до зображень є непрактичним. Було би необхідним дуже велике число нейронів, навіть у поверхневій (протилежній до глибинної) архітектурі, через дуже великі розміри входу, пов'язані з зображеннями, де кожен піксель є відповідною змінною. Наприклад, повноз'єднаний шар для (маленького) зображення розміром 100×100 має 10 000 ваг. Операція згортки дає змогу розв'язати цю проблему, оскільки вона зменшує кількість вільних параметрів, дозволяючи мережі бути глибшою за меншої кількості параметрів. Наприклад, незалежно від розміру зображення, області заможування розміру 5×5 , кожна з одними й тими ж спільними вагами, вимагають лише 25 вільних параметрів. Таким чином, це розв'язує проблему зникання або вибуху градієнтів у тренуванні традиційних багатошарових нейронних мереж з багатьма шарами за допомогою зворотного поширення.

Згорткові мережі можуть включати шари локального або глобального агрегування, які об'єднують виходи кластерів нейронів одного шару до одного нейрону наступного шару. Наприклад, максимізаційне агрегування

використовує максимальне значення з кожного з кластерів нейронів попереднього шару. Іншим прикладом є усереднювальне агрегування, що використовує усереднене значення з кожного з кластерів нейронів попереднього шару.

Повнозв'язні шари з'єднують кожен нейрон одного шару з кожним нейроном наступного шару. Це, в принципі, є тим же, що й традиційна нейронна мережа багатшарового перцептрону.

ЗНМ використовують спільні ваги в згорткових шарах, що означає, що для кожного рецептивного поля шару використовується один і той же фільтр (банк ваг); це зменшує обсяг необхідної пам'яті та поліпшує продуктивність.

2.2 Розробка архітектури згорткової нейронної мережі

Як було вирішено у першому розділі для реалізації поставленої задачі буде використано згортковцунейронну мережу. Дана нейронна мережа являється одною з тих у яких найбільша швидкодія. Що дає змогу використовувати її не тільки з зображеннями а й з відео файлами.

На сьогоднішній день існує багато різноманітних архітектур згорткових нейронних мереж, наприклад, Alex-NET, VGG-16, R-CNN, Fast R-CNN, YOLO, SSD та інші. Нам треба взяти за основу ту, що найефективніше підходить саме для розв'язання задачі детектування об'єктів на відео.

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

У моделі YOLO запропоновано визначити виявлення об'єктів як одне регресійне завдання - прямо від пікселів зображення до координат обмежувальних прямокутників і ймовірностей класів для кожного об'єкта [14].

Одна згорткова мережа одночасно передбачає безліч обмежувальних прямокутників і ймовірності класів для них.

Модель YOLO має такі переваги:

- Висока швидкість виявлення. Мережа має швидкодію 45 кадрів в секунду, а її швидка версія (Fast YOLO) більш ніж 150 кадрів в секунду.

- YOLO робить прогноз по всьому зображенню, на відміну від методів ковзного вікна і методів, заснованих на областях пропозиції. Це дозволяє моделі кодувати контекстуальну інформацію про класи. Завдяки цьому YOLO робить в два рази менше помилок на тлі зображення в порівнянні з R-CNN, так як модель R-CNN може робити помилки на тлі зображення, тому що не може побачити більший контекст.

- Завдяки тому, що YOLO вивчає узагальнені подання об'єктів, при навчанні на природних зображеннях і тестуванні на різних творах мистецтва, YOLO набагато перевершила найкращі методи виявлення (R-CNN, Fast R-CNN).

Подальший розвиток ідеї YOLO отримали в моделі SSD, що використовує такий же принцип виявлення об'єктів відразу по всьому зображенню, що відображено в її назві - Single Shot Detector [15]. Модель SSD - перша глибока мережа для виявлення об'єктів, що не використовує ознаки, які складаються з обмежувальних прямокутників. Це призводить до значного збільшення продуктивності і виявлення об'єктів з більш високою точністю. Основне поліпшення швидкості відбувається при відмові від використання обмежувальних прямокутників. Також поліпшення точності відбувається шляхом використання невеликого згорткового фільтра для передбачення класу об'єкта і згорткового фільтра для корекції положення обмежувальних прямокутників з різним співвідношенням сторін і розміром. Ці згорткові фільтри застосовуються до кількох карт ознак, як на ранніх, так і на більш пізніх шарах мережі, що дозволяє виявляти об'єкти різного розміру.

На рис. 2.1 зображено порівняння архітектур YOLO і SSD. У моделі SSD додаються додаткові шари, що дозволяють виявляти об'єкти різного розміру.

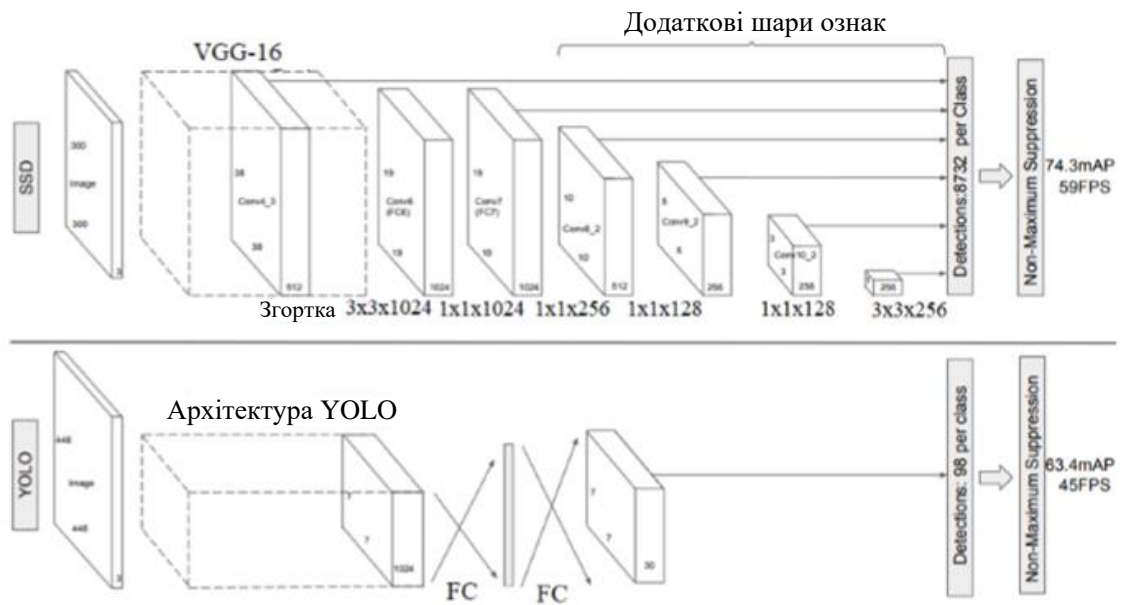


Рисунок 2.1 - Порівняння архітектури SSD (зверху) і YOLO (знизу)

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

- Модель вилучення ознак (VGG, ResNet, Inception, MobileNet).
- Дозвіл вхідного зображення.
- Як пророкування виключається з розрахунку функції втрат.
- Число передбачень в моделі.
- Кодування обмежувальних прямокутників.
- Розширення (аугментація) наборів даних.
- Набір даних, що використовується для навчання.
- Використання багатомасштабних зображень в навчанні або тестуванні.
- Локалізація функції втрат.
- Робота з програмним забезпеченням для глибокого навчання.
- Параметри навчання, такі як розмір батча (кількість даних для навчання на одному кроці), крок навчання, падіння швидкості навчання.

На рис. 2.2 показаний порівняльний результат точності моделей [15]. Там видно, що найбільшу точність дає модель SSD512.

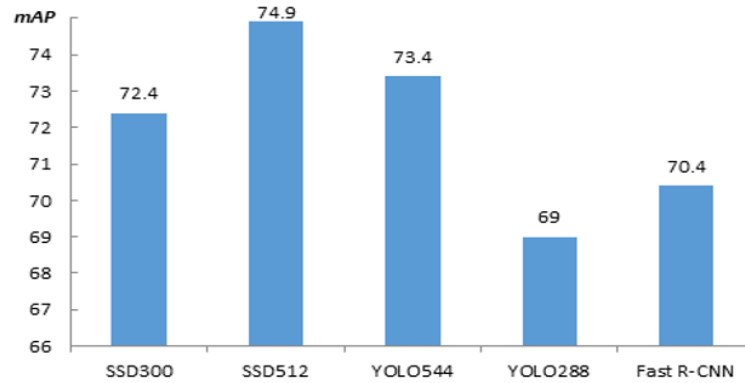


Рисунок 2.2 – Результати точності різних моделей згорткових нейромереж

На рисунку 2.3 показаний порівняльний результат продуктивності, що вимірюється в FPS (кадрів в секунду) [15]. Навчальні та перевірочні набори даних аналогічні попереднім. Там видно, що найбільшу продуктивність має модель YOLO(high).

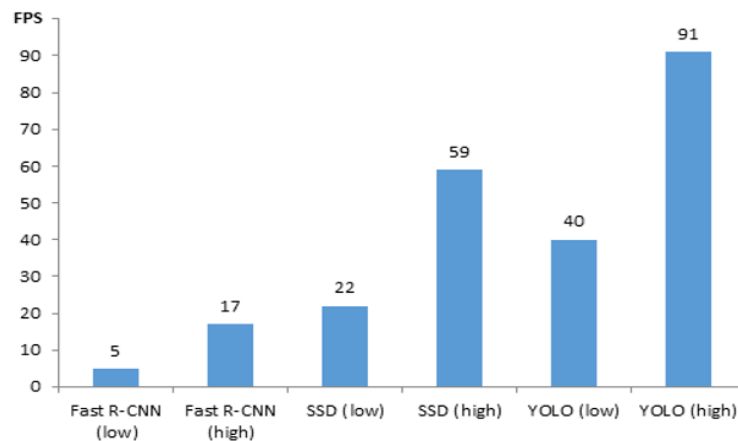


Рисунок 2.3 - Порівняння швидкодії різних моделей згорткових нейромереж

Таким чином, найбільшу продуктивність забезпечує модель YOLO, а найвищу точність має модель SSD512. Оскільки наша система має працювати в реальному часі, тому більше значення в нашому випадку має швидкодія.

Оскільки максимальну швидкість має архітектура YOLO і при цьому має достатньо високу точність, то цю архітектуру візьмемо за основу для розробки інформаційної технології розпізнавання нетипових ситуацій у частині детектування об'єктів на відео.

Розробляючи будь яку нейронну мережу потрібно щоб вона містила такі шари як:

- Шар входу
- Шар виходу
- Приховані шари

У випадку проектування та побудови саме згорткової нейронної мережі потрібно щоб були присутні такі приховані шари:

- Згорткові шари
- Агрегуювальні шари
- Повноз'єднані шари

Побудований нами шар входу нестиме функцію приймання зображень. Оскільки для повноцінної правильної роботи нейронної мережі необхідно щоб на кожен піксель зображення входу відводили окремий нейрон постає проблема того що у різних відео різне розширення зйомки. Провівши аналіз різних відео та камер спостереження, було вирішено, що розширення, до якого будуть перетворюватись усі вхідні данні буде 448x448 пікселів, так як це розширення є стандартом запису камер спостереження тому являється найрозповсюдженішим. І саме тому розміри вхідного шару також будуть 448 на 448.

Після шару входу мають слідувати приховані шари нейронної мережі. Оскільки згорткова нейронна мережа для покращення швидкості роботи використовує принцип рецептивних полів, тобто окремі нейрони реагують лише на обмежену область усіх входів, приклад принципу роботи згорткової мережі відображено на рисунку 2.4.

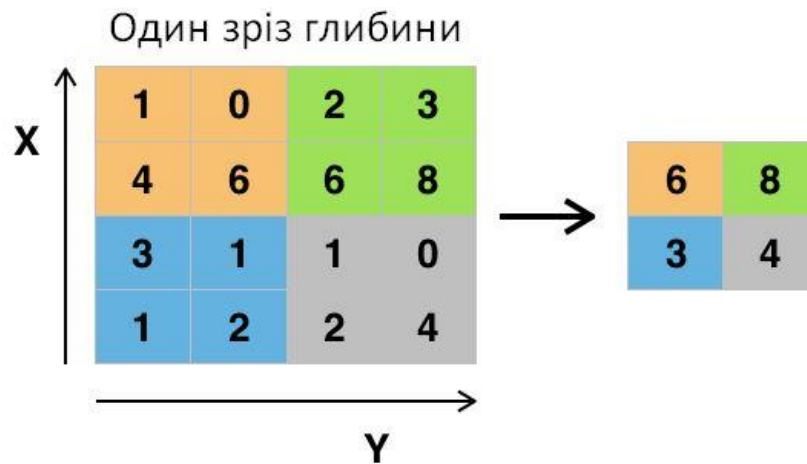


Рисунок 2.4 - Принцип роботи згорткової нейронної мережі

Для підвищення точності роботи системи додамо часткове перекриття рецептивних полів у проектованій нейронній мережі.

Тип шарів, який відповідає за реагування кожного нейрону на окрему ділянку зображення, називається згортковим шаром. Дані шари відповідають за операцію згортки зображення, передаючи результат до наступного шару. Згортка імітує реакцію окремого біологічного нейрону на зоровий стимул з певної частини зорового поля. Кожен згортковий нейрон обробляє дані лише для свого рецептивного поля [16].

Також використаємо так звані агрегувальні шари. Згорткові мережі можуть включати шари локального або глобального агрегування, які об'єднують виходи кластерів нейронів одного шару до одного нейрону наступного шару. Наприклад, максимізаційне агрегування використовує максимальне значення з кожного з кластерів нейронів попереднього шару. Іншим прикладом є усереднювальне агрегування, що використовує усереднене значення з кожного з кластерів нейронів попереднього шару в нашому випадку використаємо власну функцію:

$$S(i, j) = (I \cdot K)(i, j) \equiv \sum_{m=0}^i \sum_{n=0}^j I(m, n)K(i - m, j - n), \quad (2.1)$$

де m, n – поточне положення ядра відносно зображення $I(i, j)$ з розміром $i \times j$

Приклад роботи агрегувальних шарів наведено на рисунку 2.5.

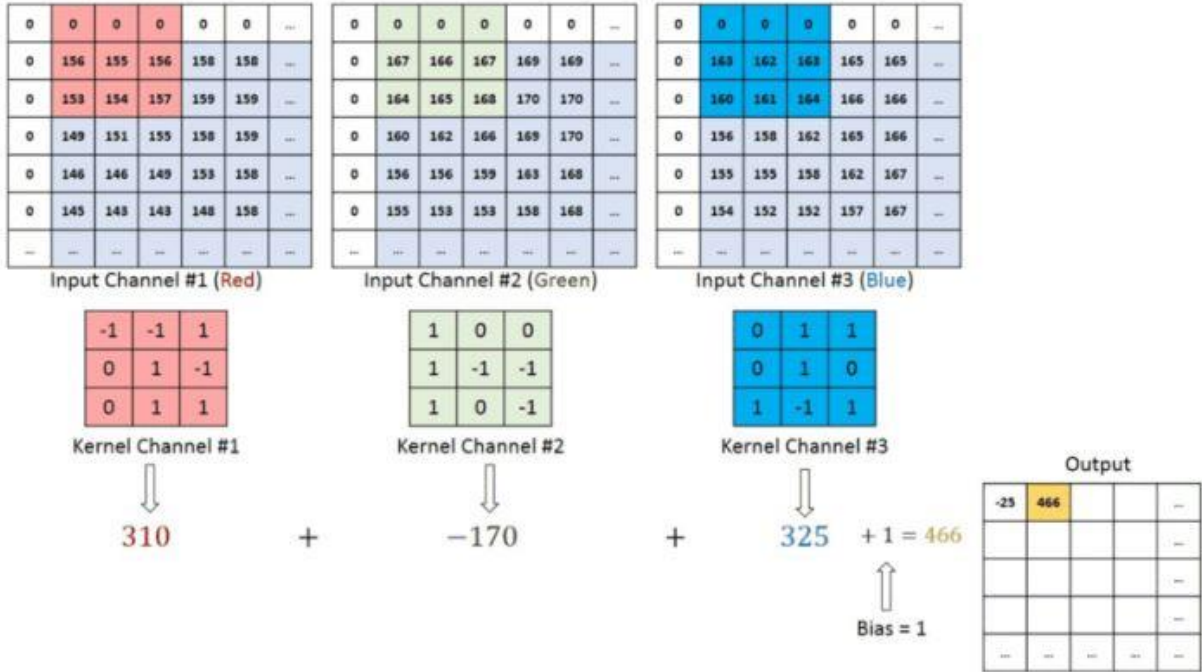


Рисунок 2.5 – Приклад роботи агрегувальних шарів

Останім типом прихованих шарів є повноз'єднані шари які будуть використовуватись як ядро нашої нейронної мережі для визначення зображення чи його частини. Повноз'єднані шари з'єднують кожен нейрон одного шару з кожним нейроном наступного шару. Це, в принципі, є тим же, що й традиційна нейронна мережа багатозарового перцептрону. Загальний вид згорткової нейронної мережі наведено на рисунку 2.6.



Рисунок 2.6 – Загальний вигляд згорткової нейронної мережі

Після розрахунку кількості шарів та їх розстановки отримуємо схему, зображену на рисунку 2.7.

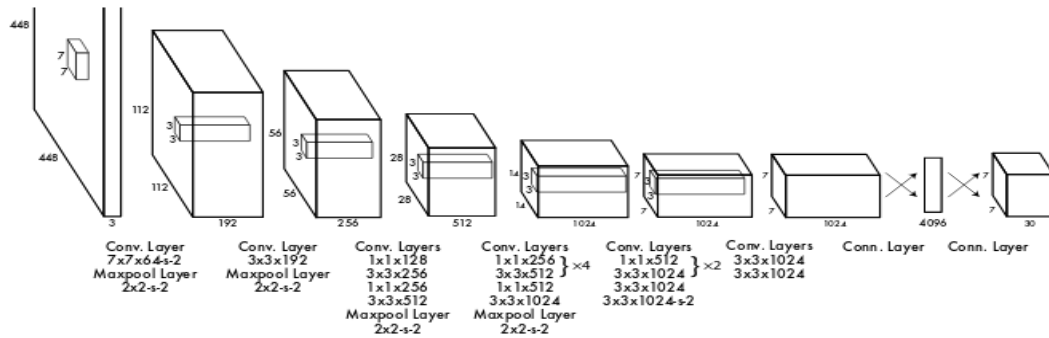


Рисунок 2.7 – Структура згорткової нейронної мережі, що проектується

Для економії пам'яті, затрачуваної на роботу нейронної мережі використаємо спільні ваги в згорткових шарах.

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

2.3 Метод навчання згорткової нейронної мережі

Метою навчання мережі алгоритмом зворотного поширення помилки є корекція ваг зв'язків для максимізації точності результату. При навчанні передбачається, що для кожного вхідного вектора існує парний йому цільовий вектор, що задає необхідний вихід. Разом вони називаються навчальною парою. Мережа навчається на багатьох парах.

Позначимо за T і OUT очікуваний та реальний виходи нейронної мережі відповідно. А за p та q нейрон, з якого виходить зв'язок та у який входить зв'язок відповідно.

Введемо величину σ , яка дорівнює різниці між необхідним T_q і реальним OUT_q виходами, помноженої на похідну логістичної функції активації:

$$\sigma_q = f' * (T_q - OUT_q), \text{ де} \quad (2.2)$$

f' – похідна функції активації,

T_q – очікуване значення виходу мережі,

OUT_q – реальне значення виходу мережі.

Тоді, ваги шару після корекції будуть дорівнювати:

$$\omega_{p-q}(i + 1) = \omega_{p-q}(i) + \alpha * \sigma_q * OUT_p, \text{ де} \quad (2.3)$$

i – номер поточної ітерації,

ω_{p-q} – величина зв'язк нейрона p з нейроном q ,

α – швидкість навчання,

OUT_p – вихід нейрона p .

На першому кроці навчання мережі вагові коефіцієнти ініціалізуються довільними значеннями. Після цього, сигнал поширюється від попередніх шарів до наступних.

Другим кроком навчання мережі є поширення помилки, отриманої на першому кроці, в зворотному напрямку, тобто від наступних шарів до попередніх. При цьому відбувається корегування ваг і порогових значень кожного нейрона за формулами 2.2 та 2.3.

2.4 Структура інформаційної технології розпізнавання нетипових ситуацій на відео у частині детектування об'єктів

Структура інформаційної технології розпізнавання нетипових ситуацій на відео у частині детектування об'єктів на основі згорткової нейронної мережі, детально описана у попередніх підпунктах, зображена на рис. 2.8.

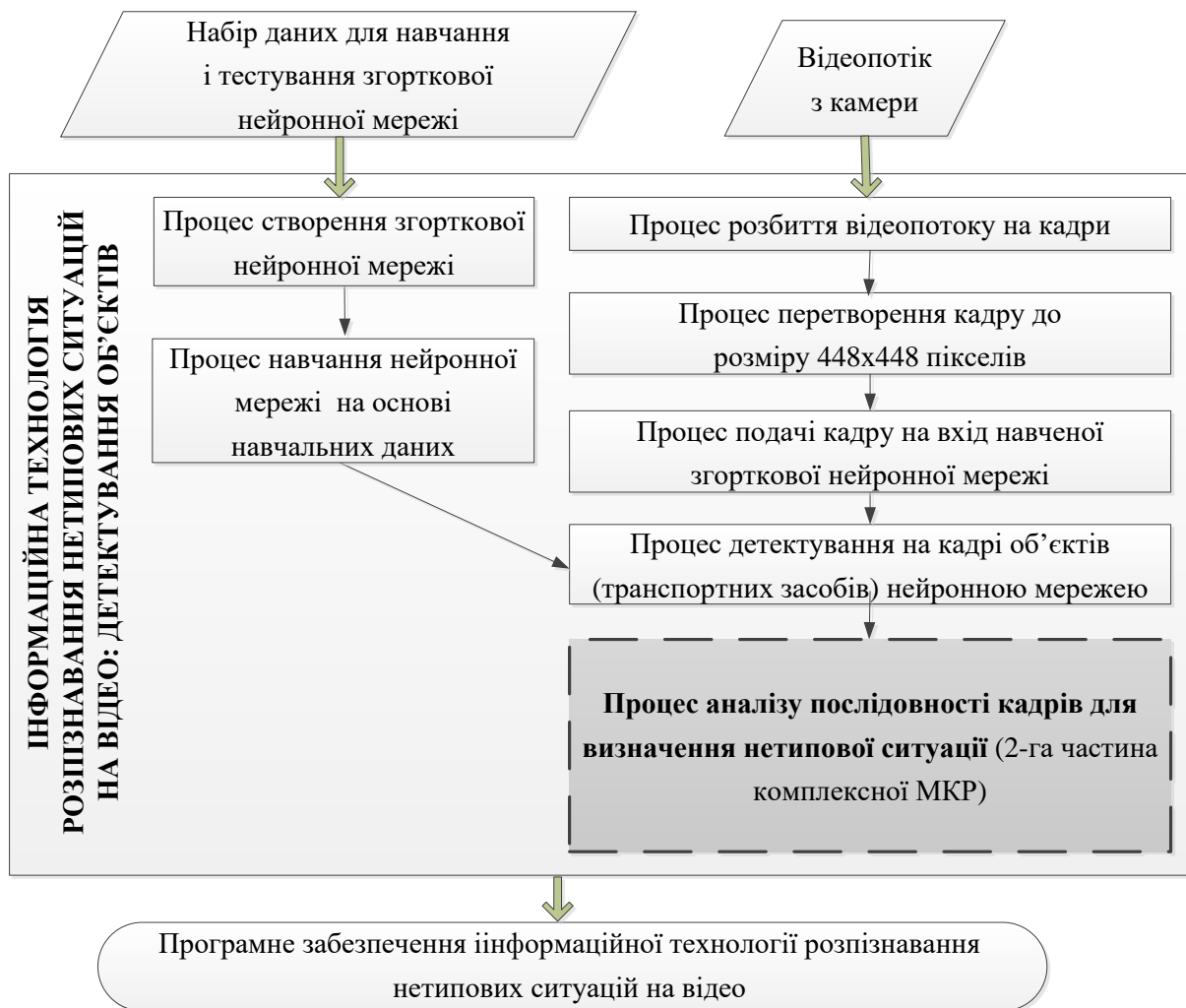


Рисунок 2.8 - Структура інформаційної технології розпізнавання нетипових ситуацій на відео у частині детектування об'єктів

Інформаційна технологія побудована на основі згорткової нейронної мережі. Оскільки кожна нейронна мережа має два режими: навчання та функціонування, то структура інформаційної технології має дві частини. Ліва

частина структури на рис. 2.8 відображує навчання згорткової нейронної мережі, а права частина структури на рис. 2.8 відображує функціонування. Для роботи інформаційної технології розпізнавання нетипових ситуацій на відео на основі згорткової нейронної мережі в режимі навчання вхідною є інформація у вигляді набору зображень, на яких зображені різноманітні транспортні засоби (легкові автомобілі, вантажівки, мотоцикли, велосипеди, пішоходи і т.п.) для навчання і тестування. В цій частині спочатку відбувається процес створення (генерації) згорткової нейронної мережі згідно обраної архітектури YOLO. Потім відбувається процес навчання нейронної мережі на основі набору даних. Після цього згорткова нейронна мережа готова до детектування об'єктів (транспортних засобів) на відео.

Для роботи інформаційної технології розпізнавання нетипових ситуацій на відео на основі згорткової нейронної мережі в режимі функціонування вхідною є інформація у вигляді відео з камери чи файлу. Спочатку відбувається процес розбиття відеопотоку на кадри. Потім здійснюється процес перетворення кожного кадру до розміру 448x448 пікселів, що відповідає розмірності входу згорткової нейронної мережі. Далі проводиться процес подачі кадру на вхід навченої згорткової нейронної мережі. Після цього йде процес детектування на кадрі об'єктів (транспортних засобів) нейронною мережею. Вихідними даними цього процесу мають бути набір зображень (фреймів), з виділеними на них рамкою шуканими образами автомобілів та їх габарити і координати розташування.

У наступній частині інформаційної технології, яка детально описана у 2-ій частині цієї комплексної МКР, відбувається аналіз послідовності кадрів для визначення нетипової ситуації. В цій частині відбуваються процеси відстеження (трекінгу) об'єктів, накопичення параметрів руху (швидкість, прискорення) виявлених об'єктів, статистичної обробки параметрів руху виявлених об'єктів, аналізу статистичних параметрів руху об'єктів з метою виявлення нетипової ситуації та збереження результатів виявлення нетипової ситуації.

Таким чином, розроблена структура інформаційної технології розпізнавання нетипових ситуацій на відео на основі згорткової нейронної мережі може бути використана для подальшої розробки програмних засобів.

2.5 Розробка структурної схеми програмної системи розпізнавання нетипових ситуацій на відео

Проектування програмного забезпечення - це процес вирішення задач та планування для створення програмного рішення. Після того як мета і задача програми описані, розробник повинен створити дизайн проекту. Для поліпшення якості цього процесу та збільшення наочності етапів проектування використаємо схеми. Для відображення головного та найбільш загальною рівня програми використаємо загальну структурну схему.

Структурною називають схему, що відображає склад і взаємодію по управлінню частин майбутнього програмного забезпечення. Структурні схеми пакетів програм не інформативні, оскільки організація програм в пакети не передбачає передачі управління між ними. Тому структурні схеми розробляють для кожної програми пакету окремо, а список програм пакету визначають аналізуючи функції, вказані в технічному завданні. Структурними компонентами програмної системи або програмного комплексу можуть служити програми, підсистеми, бази даних, бібліотеки ресурсів і тому подібне [17].

Загальна структурна схема нашої програми зображена на рисунку 2.9. Як показано на даній схемі, програма міститиме 4 модулі: модуль (попередньої) обробки зображення, модуль детектування об'єктів, модуль відстеження (трекінгу) об'єктів та модуль аналізу сцен (визначення нетиповості ситуації, тобто аварії), а також інтерфейс та сховище даних.



Рисунок 2.9 – Загальна структурна схема програми

Першим є модуль (попередньої) обробки зображення. Вхідними даними цього модуля має бути відео з камери чи файлу. А на виході – набір фреймів (кадрів), приведенних до розмірності 448x448, що відповідає входу нейронної мережі.

Наступним є модуль детектування об'єктів. Вхідними даними цього модуля мають є окремі фрейми відео. Як було визначено у попередніх розділах даної роботи цей модуль міститиме в собі згорткову нейронну мережу, яка буде виконувати задачу розпізнавання образів. Вихідними даними цього модуля

мають бути набір зображень (фреймів), з відзначеними на них шуканими образами автомобілів та їх габарити і координати розташування.

Наступним є модуль відстеження (трекінгу) об'єктів. Вхідними даними цього модуля є набір фреймів, з відзначеними на них шуканими образами автомобілів та їх габарити і координати розташування, а вихідними - набір статистичних даних які будуть характеризувати динаміку руху цих автомобілів. Серед цих даних повинна бути інформація про точне положення кожного окремого об'єкту, його розміри, параметри швидкості та прискорення.

Наступним є модуль аналізу сцен (визначення нетиповості ситуації, тобто аварії), який відповідає за відстеження аварій. Даний модуль повинен на основі наданих попереднім модулем даних визначати, чи відбулась аварія.

Користувач взаємодіє з програмою через інтерфейс. З його допомогою користувач може виконувати такі дії як: задання параметрів та внесення налаштувань роботи програми, підключення відео ресурсу на якому буде проводитись відстеження, отримання звукових та інших сигналів про виявлення аварії чи збір даних по знайденим аваріям. Структурна схема взаємодії користувача з інтерфейсом зображена на рис. 2.10.

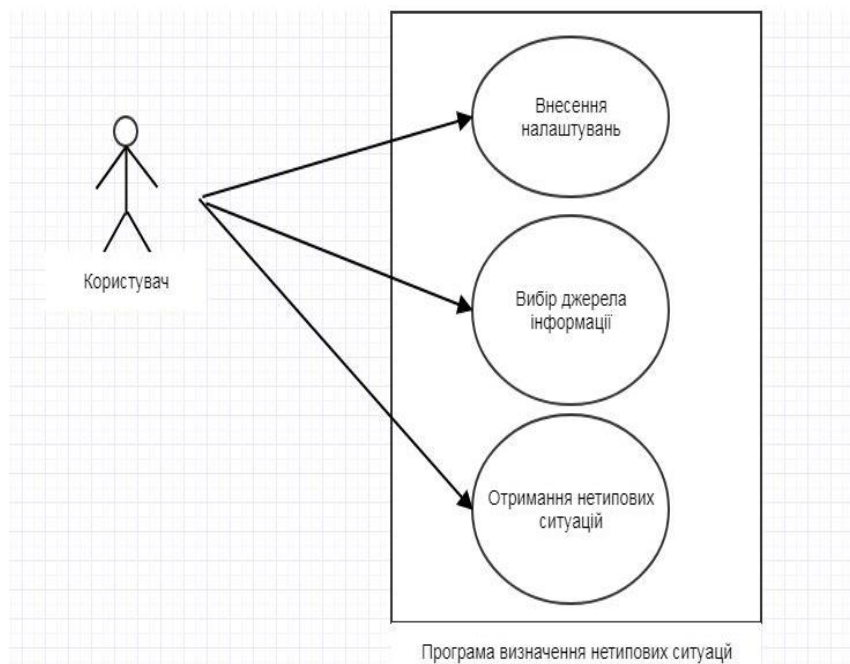


Рисунок 2.10 – Структурна схема взаємодії користувача з інтерфейсом

Сховище даних створюється з метою збереження деякої інформації такої як фрейм на якому відображено момент початку аварії та час відносно потоку фреймів [17]. Дана інформація буде доступна для перегляду через інтерфейс та збереження на інші носії

2.6 Висновок

В даному розділі було обгрунтовано вибір типу нейронної мережі - згорткова, та розроблено архітектуру згорткової нейронної мережі для розв'язання задачі детектування об'єктів на відео, За основу взято архітектуру YOLO, проаналізовано метод та математичну модель навчання згорткової мережі. Мережа складатиметься з вхідного шару розмірністю 448×448 , 6 згорткових шарів, 4 агрегувальних шарів та 2 повнозв'язних вихідних шари. Функція, яка буде використана в шарах агрегування – максимізаційна (max-pooling). У підсумку було розроблено структуру інформаційної технології розпізнавання нетипових ситуацій на відео у частині детектування об'єктів на основі згорткової нейронної мережі та розроблено структурну схему програмної системи розпізнавання нетипових ситуацій на відео.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ НЕТИПОВИХ СИТУАЦІЙ У ЧАСТИНІ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ НА ВІДЕО

3.1 Обґрунтування вибору мови програмування

Для виконання даної роботи нам потрібно використати мову програмування, на якій можна працювати з бібліотеками комп'ютерного зору. Під даний критерій потрапляють C# , Python, Java. Розглянемо кожен із цих варіантів.

C# — мова програмування високого рівня з підтримкою декількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної.

Одними з особливостей мови C# є:

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

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

Можливість роботи на низькому рівні з пам'яттю, адресами, портами.

Можливість обчислення на етапі компіляції, з використанням шаблонів і створення узагальнених алгоритмів для різних типів даних, їх спеціалізація.

Можливість перевантаження операторів.

Можливість перевантаження імен функцій.

Підтримка узагальненого програмування через шаблони.

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

в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

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

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

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

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

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ [18]. Він поширюється під вільною ліцензією Python Software Foundation License, що дозволяє використовувати його без обмежень в будь-яких додатках, включаючи пропрієтарні [18]. Є реалізація інтерпретатора для JVM з можливістю компіляції, CLR, LLVM, інші незалежні реалізації..

До середовища програмування також є ряд вимог які бажано виконати. До цих вимог належать:

- Доповнювання коду.
- Виявлення помилок.
- Можливість легко під'єднувати.

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

3.2 Вибір бібліотек та фреймворків для детектування об'єктів

Детектування об'єктів на зображеннях є досить складним та ресурсоємким процесом. Адже для комп'ютера потрібно виконати ряд послідовних завдань щоб знайти на зображенні об'єкт. Перш за все потрібно якимось чином виділити область зображення яка є «цілісною» та має якусь спільну відмінність від всього залишку площі зображення. Наступним кроком є співставлення даної області зображення з класами чи кластерами інших зображень, якщо відмінність є незначною, тоді дану область зображення комп'ютер також відносить до даного кластеру. Але порівнювати область зображення з цілісним малюнком не є доцільним. Оскільки кожна найменша

відмінність у масштабі, співвідношенні частин, формі зображення частин та інших параметрів шуканого об'єкта можуть і будуть відрізнятися. Комп'ютеру стане надзвичайно складно визначити які з даних відмінностей є важливими, а які з них незначними. Тому для спрощення процесу пошуку було винайдено декілька способів. Першим був метод градієнтів. На зображення накладається матриця градієнтів які вказують локальні напрямки від темного до світлого. Таким чином можна було визначити деякі контури об'єктів з суцільного зображення та приблизно зрозуміти по кольоровій гаммі що це за об'єкт. З розвитком нейрокомп'ютерних мереж стало можливе використання їх як свого роду аналізатори. З того часу стали використовувати інший метод пошуку зображень. Зображення поділяють на певну кількість областей, і в кожній з них нейронна мережа шукає якусь маленьку частину об'єкта. Якщо виявляється, що в декількох сусідніх зонах були знайдені частини одного об'єкта та їх розташування приблизно відповідає правильному, то збільшується вірогідність того що це і є зображення шуканого об'єкту. Даний метод дозволив вирішити питання масштабу та спростив сам процес знаходження об'єкту так як комп'ютеру легше знайти ряд простих об'єктів, ніж один великий. При досягненні зазначеного рівня вірогідності комп'ютер приймає ціле зображення за даний об'єкт. Дану технологію назвали комп'ютерний зір.

Багато з методів та застосувань все ще знаходяться на стадії фундаментальних досліджень, але все більша кількість методів знаходить застосування в комерційних продуктах, де вони часто складають частину складнішої системи, яка може вирішувати складні задачі (наприклад, в галузі медичних зображень або вимірювання та контролю якості в процесах виробництва). У більшості практичних застосувань комп'ютерного зору, комп'ютери попередньо запрограмовано для вирішення окремих задач, але методи, що базуються на знаннях, стають все більше узагальненими.

На сьогоднішній день існує потужна бібліотека Комп'ютерного зору – OpenCV [19].

OpenCV (Open Source Комп'ютерне бачення бібліотека) є відкритим джерелом комп'ютерного бачення і бібліотеки програмного забезпечення машинного навчання. OpenCV був побудований, щоб забезпечити спільну інфраструктуру додатків комп'ютерного зору і прискорити використання сприйняття машин у комерційних продуктах. Будучи ліцензованим продуктом BSD, OpenCV [20] спрощує для підприємств використання та модифікації коду.

Бібліотека має більше 2500 оптимізованих алгоритмів, що включає в себе повний набір як класичних, так і найсучасніших алгоритмів комп'ютерного зору і машинного навчання. Ці алгоритми можуть бути використані для виявлення та розпізнавання облич, визначення об'єктів, класифікації дій людини у відео, відстеження рухів камери, відстеження рухомих об'єктів, вилучення 3D-моделей об'єктів, створення 3D-хмари точок від стереокамер, зшивання зображень для отримання високого дозволу зображення всієї сцени, пошук схожих зображень з бази даних зображень, видалення червоних очей з зображень, зроблених за допомогою спалаху, слідування рухів очей, розпізнавання декорацій та встановлення маркерів для накладання його з розширеною реальністю, і т.д. спільноти та розрахункова кількість завантажень, що перевищує 14 мільйонів. Бібліотека широко використовується в компаніях, дослідницьких групах та урядових органах.

Поряд з відомими компаніями, такими як Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, що використовують бібліотеку, існує багато стартапів, таких як Applied Minds, VideoSurf і Zeitera, які широко використовують OpenCV. Розгорнуте використання OpenCV [21] охоплює діапазон від зшивання зображень вулиць разом, виявлення вторгнень у відео спостереження в Ізраїлі, моніторингу шахтного обладнання в Китаї, надання допомоги роботам у навігації та збиранні об'єктів у Willow Garage, виявлення аварій в потопаючих басейнах в Європі Іспанія і Нью-Йорк, перевіряючи злітно-посадочні смуги для сміття в Туреччині, оглядають етикетки на продуктах на фабриках по всьому світу про швидке виявлення облич в Японії.

Він має C ++, Python, Java і MATLAB інтерфейси і підтримує Windows, Linux, Android і Mac OS. OpenCV спирається в основному на додатки реального часу бачення і використовує переваги MMX і SSE інструкції, коли є. На даний момент активно розвиваються повнофункціональні інтерфейси CUDA і OpenCL. Є більше 500 алгоритмів і приблизно в 10 разів більше функцій, які складають або підтримують ці алгоритми. OpenCV написано на самому початку в C ++ і має інтерфейс шаблонів, який працює з контейнерами STL.

OpenCV була розроблена як крос-платформена бібліотека. Таким чином, бібліотека була написана на C, що робить портативний OpenCV практично в будь-якій комерційній системі, від PowerPC Mac до робототехнічних собак. Починаючи з версії 2.0, OpenCV включає в себе традиційний інтерфейс C, а також новий C ++. У більшості випадків нові алгоритми OpenCV розроблені в C ++. Також обгортки для таких мов, як Python і Java, були розроблені для заохочення широкої аудиторії. OpenCV працює на обох робочих столах (Windows, Linux, Android, MacOS, FreeBSD, OpenBSD) і мобільних пристроях (Android, Maemo, iOS).

У версії 2.2 бібліотека була реорганізована. Замість універсальних модулів `sxcore`, `svaux`, `highGUI` і інших було створено кілька компактних модулів з більш вузькою спеціалізацією:

`opencv_core` - основна функціональність. Включає в себе базові структури, обчислення (математичні функції, генератори випадкових чисел) і лінійну алгебру, DFT, DCT, введення / виведення для XML і YAML і т. д.

`opencv_imgproc` - обробка зображень (фільтрація, геометричні перетворення, перетворення кольорних просторів і т. д.).

`opencv_highgui` - простий UI, введення / виведення зображень і відео.

`opencv_ml` - моделі машинного навчання (SVM, дерева рішень, навчання зі стимулюванням і т. д.).

`opencv_features2d` - розпізнавання і опис плоских примітивів (SURF (англ.) рос., FAST і інші, включаючи спеціалізований фреймворк).

`opencv_video` - аналіз руху і відстеження об'єктів (оптичний потік, шаблони руху, усунення фону).

`opencv_objdetect` - виявлення об'єктів на зображенні (перебування осіб за допомогою алгоритму Віоли-Джонса, розпізнавання людей HOG і т. д.).

`opencv_calib3d` - калібрування камери, пошук стерео-відповідності та елементи обробки тривимірних даних.

`opencv_flann` - бібліотека швидкого пошуку найближчих сусідів (FLANN 1.5) і обгортки OpenCV.

`opencv_contrib` - супутній код, ще не готовий для застосування.

`opencv_legacy` - застарілий код, збережений заради зворотної сумісності.

`opencv_gpu` - прискорення деяких функцій OpenCV за рахунок CUDA, створений за підтримки NVidia.

Серед декільком основних алгоритмів та методів, які використовуються у системах комп'ютерного зору, згортова нейронна мережа YOLO показує найкращі результати, тому ми будемо використовувати архітектуру YOLO.

YOLO (Ви тільки дивитесь раз) - це метод / спосіб виявлення об'єкта. Це алгоритм / стратегія, що стоїть перед тим, як код виявляє об'єкти на зображенні. Офіційна реалізація цієї ідеї доступна через DarkNet (реалізація нейронної мережі з нуля).. Раніше рамки навколо об'єктів виявлялися шляхом багаторазового розглядання різних частин зображення в різних масштабах і повторно використовувалися методи класифікації зображень для виявлення об'єктів. Цей підхід був повільним і неефективним. Архітектура ЗНМ YOLO використовує зовсім інший підхід. Все зображення проглядається тільки один раз і проходить через мережу один раз і виявляються об'єкти. Звідси і назва. Це дуже швидко. Саме тому YOLO стала настільки популярною. Є також інші популярні згорткові нейронні мережі для виявлення об'єктів, такі як Faster R-CNN і SSD .

Darknet - це нейронна мережа з відкритим кодом, написана на C та CUDA. Вона швидка, проста в установці і підтримує обчислення на CPU і GPU. Darknet встановлюється тільки з двома необов'язковими залежностями:

OpenCV, якщо користувачі хочуть більш широке розмаїття підтримуваних типів зображень або CUDA, якщо вони хочуть обчислення GPU. Жоден з них не є обов'язковим, але користувачі можуть почати, просто встановивши базову систему, яка була протестована лише на комп'ютерах Linux і Mac. Структура має функцію «Тільки подивитися один раз» (You Only Look Once - YOLO), сучасну систему виявлення об'єктів у реальному часі. На Titan X він обробляє зображення при 40-90 FPS і має mAP на VOC 2007 з 78,6% і mAP 44,0% на тесті CO-CO. Користувачі можуть використовувати Darknet для класифікації зображень для виклику 1000-класу ImageNet. Darknet відображає інформацію, коли вона завантажує конфігураційний файл і ваги, потім класифікує зображення і друкує топ-10 класів для зображення. Більш того, фреймворк може бути використаний для запуску нейронних мереж в зворотному напрямку у відповідній назві Nightmare. Періодичні нейронні мережі є потужними моделями для представлення даних, які змінюються з часом, і Darknet може обробляти їх, не використовуючи CUDA або OpenCV.

Тепер коли ми визначились з бібліотекою, фреймворком та алгоритмом для його запуску та коректної роботи нам необхідні ваги для нейронної мережі. Для даної задачі є спеціальна відкрита бібліотека TensorFlow. TensorFlow - відкрита програмна бібліотека для машинного навчання, розроблена компанією Google для вирішення завдань побудови і тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття. Застосовується як для досліджень, так і для розробки власних продуктів Google. Основний API для роботи з бібліотекою реалізований для Python, також існують реалізації для C ++, Haskell, Java, Go і Swift. Є продовженням закритого проекту DistBelief. Спочатку TensorFlow була розроблена командою Google Brain для внутрішнього використання в Google, в 2015 році система була переведена в вільний доступ з відкритою ліцензією Apache 2.0

3.3 Розробка UML діаграм програми виявлення нетипових ситуацій

Створення структурної схеми у п.2.5 дало змогу відобразити усі компоненти програми на загальному рівні. Наступним кроком є розробка схем та діаграм які допоможуть у відображенні та структуруванні системи на більш глибокому рівні та відображенні потоків даних та взаємозв'язків між модулями. Для даних задач використаємо UML. UML це уніфікована мова моделювання яка використовується для створення абстрактних моделей систем та супроводу багатьох проектів. Оскільки вона має чіткі та легко зрозумілі стандарти, використаємо її для створення часової діаграми активності. Часова діаграма активності це підрозділ діаграм станів який ми використаємо для відображення потоків даних в кожний момент часу. Часова діаграма активності надає змогу відобразити одночасні потоки даних. На рисунку 3.1 відображено часову діаграму активності проекту. Дана діаграма показує прямий та зворотний рухи даних в програмі.

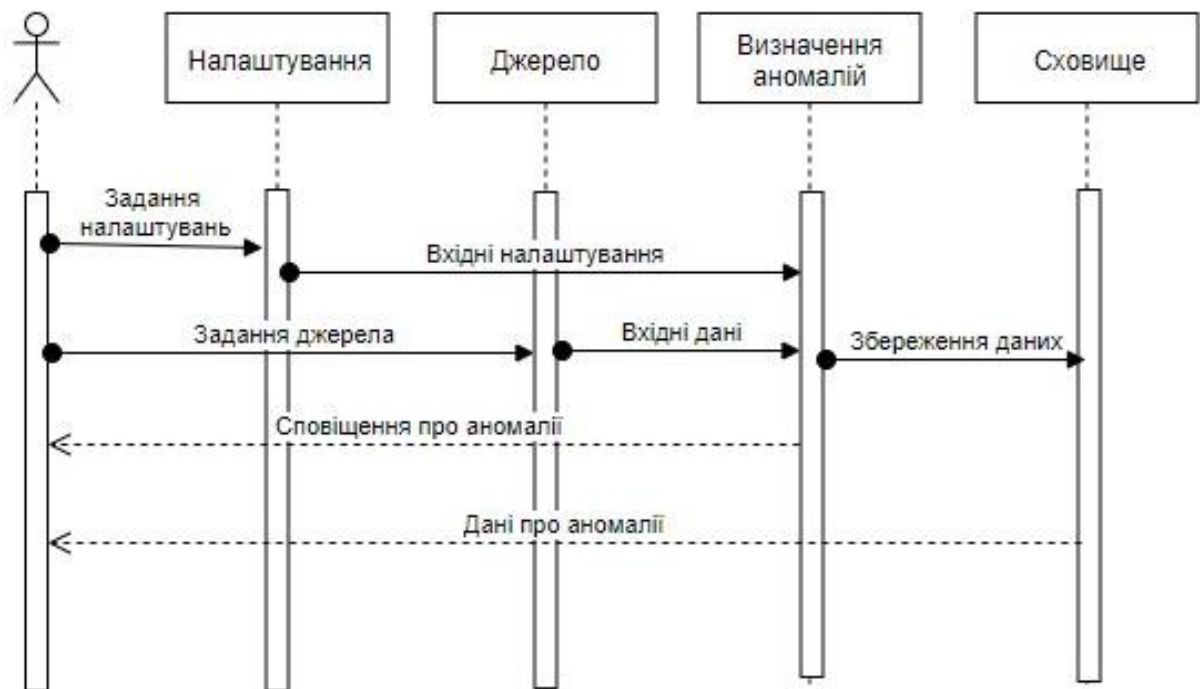


Рисунок 3.1 – Часова діаграма активності проекту.

Для відображення більш глибокого рівня програми використаємо іншу UML схему. Функціональна схема даного проекту відображена на рисунку 3.2.



Рисунок 3.2 – Функціональна схема проекту

Дана функціональна схема добре показує порядок дій, вхідні дані та механізми роботи на кожному з етапів роботи програмного циклу. Як видно на схемі на перший етап роботи програми потрапляє потік фреймів, керується цей процес за допомогою налаштувань заданих користувачем. Це такі параметри як тип фільтрів, розмір перетвореного зображення, тощо. Механізмом на даному етапі стануть різноманітні інструменти обробки інформації.[17]

Наступним етапом роботи є детектування об'єктів. Механізмом роботи або інструментом який ми для цього використаємо стане згортова нейронна мережа, вхідними даними буде вектор даних оброблених зображень. Керування роботи даного елемента буде здійснюватись за допомогою налаштувань мережі, таких як ваги нейронів визначені під час навчання.

Наступний крок це фільтрація об'єктів. Даний етап напряму впливає на точність роботи мережі. Його вхідними даними стануть об'єкти знайдені мережею та ймовірності впевненості мережі, суть цього кроку досить проста

відфільтровувати об'єкти з недостатньою точністю але це не зменшує його важливість так як завдяки цьому відсіюються зайві образи об'єктів а як наслідок і зменшується навантаження на апаратне забезпечення. Керування відбувається за допомогою заданого при налаштуваннях порогу достовірності. Механізмом роботи є фільтрація.

Останій етапом робочого циклу є визначення нетипових ситуацій (аномалій). За допомогою спеціально розроблених алгоритмів які є механізмом роботи доного етапу з вхідних даних які складаються з нових знайдених об'єктів та знайдених попередньо та відслідковуваних до доного моменту на основі прараметрів визначення які є методом керування визначатиметься чи мала місце нетипова ситуація чи ні.

Дана схема дає змогу наглядно побачити етапи робочого циклу програми та методи керування на кожному з них.

3.4 Програмна реалізація процесу навчання нейронної мережі

Відеопотік з камери буде надходити на вхід програми, яка буде розбивати цей потік на кадри. Кожен кадр (фрейм) буде аналізуватись за допомогою алгоритму з бібліотеки OpenCV. Для навчання згорткової нейронної мережі використовується наступна тренувальна конфігурація:

```
{
  "lr": 0.0001,
  "epochs": 10,
  "batch_size": {
    "train": 8
  },
  "input_size": {
    "width": 448,
    "height": 448
  },
  "bn_momentum": 0.01,
  "gpu_devices": [0],
  "data_workers": {
    "train": 3
  }
}
```

```

},
"dataset_tags": {
  "train": "train"
},
"subdivisions": {
  "train": 1
},
"print_every_iter": 10,
"weights_init_type": "continue_training",
"enable_augmentations": true
}

```

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

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

- `confidence_tag_name` - ім'я мітки довіри для передбачених прив'язок.

`mode` - група містить всі налаштування режиму:

- `name` - назва режиму визначає, як застосувати НМ до зображення (наприклад, `full_image` - застосувати НМ до повного зображення)

- `model_classes` - які класи будуть використовуватися, напр. НМ виробляє 80 класів, і ми збираємось використовувати лише декілька і ігнорувати інші. У цьому випадку ми повинні встановити поле `save_classes` зі списком зацікавлених імен класів. Рядок `add_suffix` буде доданий до нового класу для запобігання подібним іменам класів з існуючими класами в проєкті. Якщо ми збираємось використовувати всі класи моделей, просто встановіть `"save_classes": "__all__"`.

Приклад конфігурації повного зображення:

```

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

```

```

    "add_suffix": "_yolo"
  }
}
}

```

Лістинг програмного коду наведено в додатку Б.

Для навчання нейронної мережі спочатку необхідно завантажити множину зображень та на основі цієї множини сформувати кінцевий масив міток класів. Фрагмент функції, яка завантажує множину зображень та повертає вихідний масив зображень та міток класів до яких вони належать, та масив всіх можливих міток класів:

```

def load_data_and_labels(filename):
    df = pd.read_csv(filename, compression='zip',
dtype={'consumer_complaint_narrative': object})
    selected = ['product', 'consumer_complaint_narrative']
    non_selected = list(set(df.columns) - set(selected))
    df = df.drop(non_selected, axis=1)
    df = df.dropna(axis=0, how='any', subset=selected)
    df = df.reindex(np.random.permutation(df.index))
    labels = sorted(list(set(df[selected[0]].tolist())))
    encoded_labels = np.zeros((len(labels), len(labels)), int)
    np.fill_diagonal(encoded_labels, 1)
    label_dict = dict(zip(labels, encoded_labels))

```

Функція `pd.read_csv` зчитує файл з множиною зображень та записує його в змінну `df`. Потім проводиться вибір полів з множини даних, які потрібні для процесу навчання мережі, а зайві поля видаляються викликом функції `df.drop`. Після цього формується масив міток класів у вигляді векторів, де один елемент дорівнює одиниці, а інші – нулю.

Після того, як множина зображень була завантажена, потрібно розбити її на навчальну та тестову вибірки у пропорції 80% : 20%.

```

x_, x_test, y_, y_test = train_test_split(x, y, test_size=0.2)

```

Далі необхідно створити об'єкт згорткової нейронної мережі з бібліотеки TensorFlow та ініціалізувати його:

```
cnn = TransportCNN(
    sequence_length=x_train.shape[1],
    num_classes=y_train.shape[1],
    vocab_size=len(vocab_processor.vocabulary_),
    embedding_size=params['embedding_dim'],
    filter_sizes=list(map(int, params['filter_sizes'].split(","))),
    num_filters=params['num_filters'])
```

Перший та другий параметр об'єкту TransportCNN – це розмірність вхідних даних (448 та 448 відповідно). Третій параметр вказує на кількість класів транспортних засобів. Четвертий параметр вказує на довжину вектора, в який буде перетворено кожний об'єкт вхідного зображення. П'ятий параметр – це масив з розмірністю згорткових фільтрів (3×100 , 4×100 та 5×100). Шостий параметр вказує на кількість фільтрів (32 для кожного згорткового шару).

Після ініціалізації мережі починається процес навчання. Процес навчання складається з певної кількості ітерацій. На кожній ітерації обирається деяке зображення з навчальної вибірки та подається на вхід нейронної мережі.

Процес навчання мережі складається з повторення даного фрагменту коду:

```
optimizer = tf.train.AdamOptimizer(1e-3)
grads_and_vars = optimizer.compute_gradients(cnn.loss)
train_op = optimizer.apply_gradients(grads_and_vars,
    global_step=global_step)
def train_step(x_batch, y_batch):
    feed_dict = {
        cnn.input_x: x_batch,
```

```

cnn.input_y: y_batch,
cnn.dropout_keep_prob: params['dropout_keep_prob']}
_, step, loss, acc = sess.run([train_op, global_step, cnn.loss,
cnn.accuracy], feed_dict)

```

Дана функція на кожній ітерації обчислює значення виходу нейронної мережі на основі зображення з навчальної вибірки та на основі порівняння значення виходу нейронної мережі з бажаним результатом (співставлення міток класів) проводить корекцію ваг мережі на основі методу стохастичного градієнтного спуску. Ці всі операції виконуються після виклику функції `sess.run` всередині бібліотеки `Tensorflow`.

3.5 Розробка навчальної вибірки для нейронної мережі

Для того щоб нейронна мережа могла виявляти об'єкти її потрібно навчити. Для цього використовують спеціально створені навчальні вибірки. Це набір вхідних даних, які містять відзначені на них шукані об'єкти. За допомогою спеціальних програм можливо зіставляти пари вхідні данні результат роботи [14].

Оскільки в нас згортована нейронна мережа, то для її навчання використаємо алгоритми глибокого навчання.

Ціль створення навчальної вибірки - це покрити як можна більш широкий спектр ситуацій, в нашому випадку це зміна положення точки споглядання на шуканий нами тип об'єктів: автомобілі. Також, оскільки в нашій програмі використовується перетворення розмірів зображення, необхідно до навчальної вибірки додати зображення різних відношень сторін та розмірів.

Отже, навчальною вибіркою для нашої згортової мережі стане множина зображень з присутніми в ній зображеннями автомобілів. Список варіантів зображень, присутніх у навчальній вибірці, наведено в таблиці 3.1.

Оскільки навчальна вибірка складається для того, щоб програма змогла коректно детектувати об'єкти, то склад навчальної вибірки повинен відображати усі випадки, в яких нам знадобиться детектувати об'єкти.

Таблиця 3.1 – Список варіантів зображень, присутніх у навчальній вибірці

Відмінність групи зображень	Кількість зображень з даним типом відмінності
Зображення в денний час	92
Зображення в нічний час	65
Зображення зі значною дальністю до об'єкту	31
Зображення з малою дальністю до об'єкту	39
Частково перекриті зображення	50
Зображення без перешкод	75
Зображення розмірності 400x400	30
Зображення розмірності 448x448	60
Зображення розмірності 680x680	40
Зображення з іншим відношенням сторін	15

Нейрона мережа повинна бути спроможна детектувати об'єкти як у денний, так і у нічний час, тому навчальна вибірка повинна містити обидва випадки світлових періодів дня. При нескладних розрахунках було доведено, що середній час, який триває ніч це 8 годин. Як наслідок, денних годин в середньому 16, тому співвідношення зображень денних і нічних у навчальній вибірці має бути приблизно таким самим. Можливі випадки, коли нейронній

мережі потрібно буде детектувати частково перекритий чимось об'єкт. Тому такі зображення також повинні бути присутні в навчальній вибірці. Зображень перекритих об'єктів у навчальній вибірці менше ніж неперекритих, так нейронна мережа зможе краще детектувати об'єкт за його частинами.

В навчальній вибірці повинні бути присутні зображення об'єкту різної віддаленості. Додамо до вибірки приблизно однакову кількість зображень із значною дальністю до об'єкту та з малою дальністю до об'єкту.

В модулі детектування об'єктів присутні ряд фільтрів, які приводять зображення до потрібних нам розмірів. Тому, щоб нейронна мережа могла коректно працювати із зображеннями різних розмірностей, необхідно, щоб у навчальній вибірці також були присутні зображення багатьох розмірностей та відношень.

3.6 Тестування та аналіз результатів роботи програмного засобу розпізнавання нетипових ситуацій у частині детектування об'єктів на відео

Для коректної роботи програма повинна правильно виконувати ряд поставлених при розробці задач. Ці задачі включають:

- Виявлення об'єктів

- Відстеження їх та збір інформації

- Визначення аномалій

Після розробки прототипу програми було проведено його тестування на різних відео-файлах, які складають тестову вибірку .

Результати тестування представлені на рисунках 3.3 та 3.4.

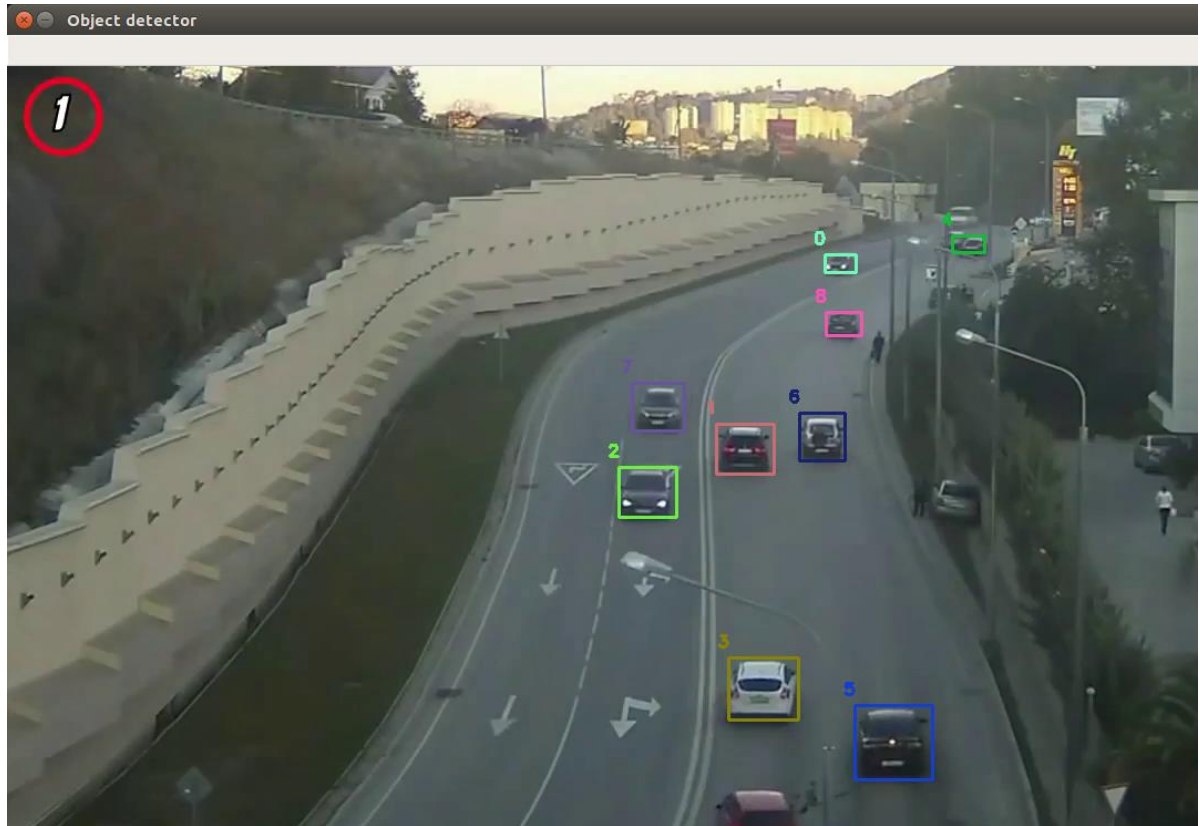


Рисунок 3.3– Демонстрація процесу детектування та маркування об'єктів на дорозі

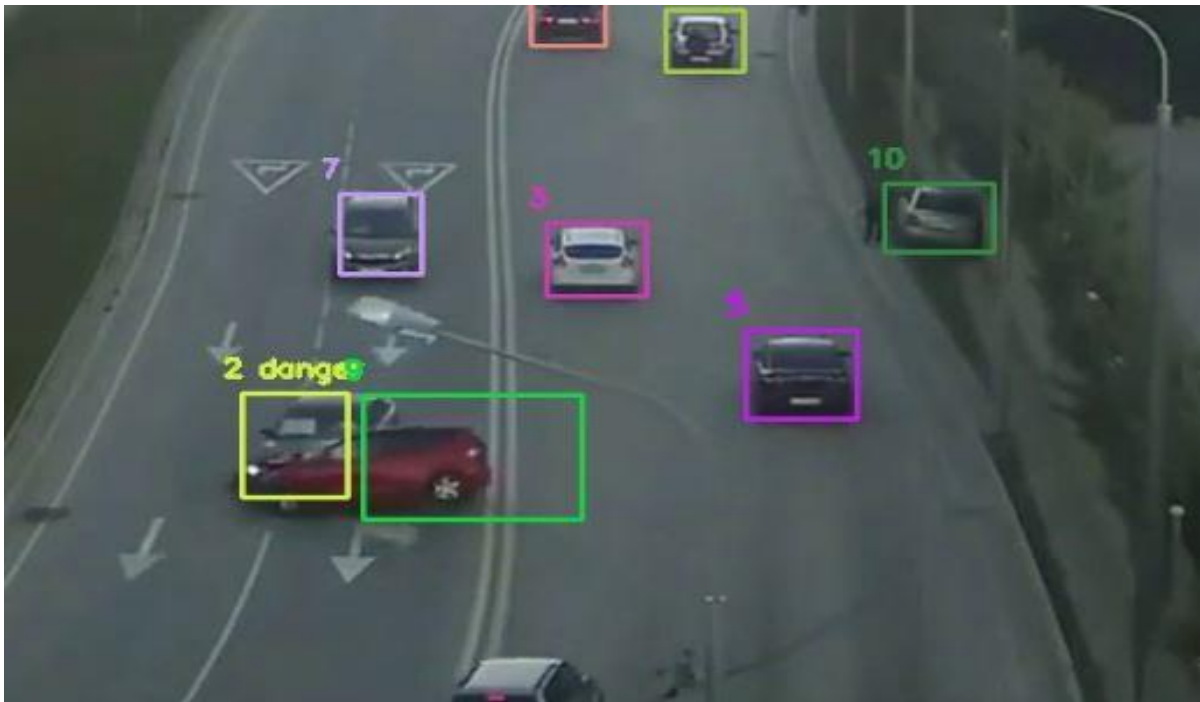


Рисунок 3.4 – Фіксація моменту аварії програмою

На контрольній вибірці окрім нашої програми перевірялась і програма аналог для порівняння результатів. Контрольна вибірка складалась за тими самими критеріями, що і навчальна вибірка, тобто містить відео, де відображені різні можливі ситуації та містить в загальному 100 відео протяжністю від 1хвилини до 15 хвилин. Результати тестування та їх порівняння відображенні в таблиці 3.2

Таблиця 3.2 – Порівняння результатів тестування програм

	Аналог	Наша програма
Кількість відео в тестуванні	100	100
Кількість шуканих об'єктів	482	482
Детектовано об'єктів	321	335
Кількість відео на яких знайдені усі об'єкти	65	68
Достовірність пошуку по об'єктам	68%	72%
Достовірність пошуку по відео	65%	68%

В результаті тестування можна зробити висновок, що програмне забезпечення працює правильно, тобто, воно коректно виявляє об'єкти на дорозі та відслідковує аварії. Для повного тестування були використані різні відео з різних камер, що дозволило протестувати програму у різних умовах. Окрім нашої програми також тестувалась програма Analytics, за результатами тестування було виявлено достовірність пошуку та детектування об'єктів програми, що була розроблена нами складає 72% та 68% по окремим об'єктам та по відео в загальному відповідно. В той час, як достовірність програми аналогу складає 68% та 65% по відповідним критеріям. Що показує різницю в 4% та в 3% по відповідним параметрам. Аналізуючи результати, було виявлено, що наша програма в основному показала кращий результат на відео, записаних у денний час. В той час як на нічних відео і аналог і розроблена в даному проекті програма показали схожі результати. Найкращою перевагою стало

якраз те, що наша програма спрямована на детектування лише автомобілів. В той час як аналог здатен розпізнавати ширший спектр об'єктів, що призвело до зменшення точності пошуку кожного конкретного типу об'єктів.

3.7 Висновок

В ході практичної реалізації інформаційної технології розпізнавання нетипових ситуацій на відео у частині детектування об'єктів у даному розділі було обгрунтовано вибір мови програмування Python та середовища програмування Visual Studio Code. Серед бібліотек та фреймворків для детектування об'єктів були обрані OpenCV, Tensorflow та Darknet. Було розроблено UML діаграму програми виявлення нетипових ситуацій та розроблено навчальну вибірку для навчання згорткової нейронної мережі задачі детектування транспортних засобів. Також було описано роботу основних частин програми та наведено відповідні фрагменти коду. В результаті було розроблено програмне забезпечення розпізнавання нетипових ситуацій на відео, яке порівняно з аналогом має кращу на 4% достовірність детектування окремих об'єктів і кращу на 3% достовірність детектування об'єктів по відео в цілому. Таким чином, мета роботи досягнута – достовірність детектування об'єктів на відео при розпізнаванні нетипових ситуацій підвищена.

4 ЕКОНОМІЧНА ЧАСТИНА

Темою магістерської кваліфікаційної роботи є «Інформаційна технологія розпізнавання нетипових ситуацій: детектування об'єктів». За цією темою в економічній частині проводяться розрахунки економічних показників на розробку програмного продукту та впровадження його на ринок аналогічних товарів.

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Колесницький О.К. та Яровий А.А. Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями, наведеними у таблиці 4.1, за 5-ти бальною шкалою.

Таблиця 4.1 – Критерії оцінювання комерційного потенціалу розробки та їх бальна оцінка

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

Продовження таблиці 4.1

4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 4.1

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	--	---	--	---

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Озеранський В.С.	2. Яровий А.А.
	Бали, виставлені експертами:	
1	4	4
2	0	1
3	4	4
4	2	3
5	4	3
6	2	3
7	2	3
8	3	3
9	4	4
10	4	4
11	4	3
12	4	4
Сума балів	СБ ₁ = 37	СБ ₂ = 39
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 38$	

Отже, з отриманих даних таблиці 4.1 видно, що рівень потенціалу нової розробки – вище середнього. Дана розробка є конкурентоспроможною з аналогами, так як для її розробки було проаналізовано недоліки та переваги аналогових продуктів, та на основі цього впроваджену у розробку. Вона має соціологічний вплив, так як розуміння того що більшість порушень буде знайдено, приведе до зменшення кількості порушень само по собі.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (4.1):

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

де M - місячний посадовий оклад конкретного розробника;

T_p - кількість робочих днів у місяці, $T_p = 20$ днів;

t - число днів роботи розробника, $t = 37$ днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.3.

Таблиця 4.3 – Розрахунки основної заробітної плати

Працівник	Оклад M , грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	7500	375	4	1500
Інженер-програміст	9600	480	33	15840
Всього:				17340

Розрахуємо додаткову заробітну плату, вона розраховується як 10-12% від суми основної заробітної плати всіх розробників:

$$Z_{\text{дод}} = 0,1 \cdot 17340 = 1734 \text{ (грн.)} \quad (4.2)$$

Нарахування на заробітну плату $H_{\text{зп}}$ для працівників бюджетної сфери становить 22% від суми основної та додаткової заробітної плати:

$$H_{\text{зп}} = (Z_o + Z_p) \cdot \frac{\beta}{100}, \quad (4.3)$$

$$H_{\text{зп}} = (17340 + 1734) \cdot \frac{22}{100} = 4196,28 \text{ (грн.)}$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12}, \quad (4.4)$$

де $Ц$ – балансова вартість обладнання, грн;

H_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – Термін використання ($T=2$ міс.).

Таблиця 4.4 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	12700	25	2	529,1
Мобільний пристрій	4300	15	2	107,5
Всього:				636.6

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n H_i \cdot Ц_i \cdot K_i, \quad (4.5)$$

де n – кількість комплектуючих;

H_i – кількість комплектуючих i -го виду;

$Ц_i$ – покупна ціна комплектуючих i -го виду, грн;

K_i – коефіцієнт транспортних витрат (прийємо $K_i = 1,1$).

Таблиця 4.5 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
USB – microUSB кабель	шт.	75	1	75
Пачка паперу	уп.	100	1	100
Ручка	шт.	4	1	4
Всього з урахуванням транспортних витрат				196,9

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi} ; \quad (4.6)$$

де V – вартість 1кВт-години електроенергії ($V=2.35$ грн/кВт);

Π – установлена потужність комп'ютера ($\Pi=0,6$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=296$ год.);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,7$).

$$V_e = 2,35 \cdot 0,6 \cdot 296 \cdot 0,7 = 292,15 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$. Інші витрати I_v можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (z_o + z_p). \quad (4.7)$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 1 * (17340 + 1734) = 19074 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$B = Z_o + Z_d + H_{зп} + A + K + B_e + I_B \quad (4.8)$$

$$B = 17340 + 1734 + 4196,28 + 636,6 + 196,9 + 292,15 + 19074 = 43\,469,83 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою:

$$B_{заг} = \frac{B_{ін}}{\alpha} \quad (4.9)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{заг} = \frac{43\,469,83}{1} = 43\,469,83 \text{ грн.}$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta} \quad (4.10)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{43\,469,83}{0,9} = 48\,299,8 \text{ грн.}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо кількісно, яку вигоду можна отримати у майбутньому від впровадження результатів виконаної наукової роботи.

Виконання наукової роботи та впровадження її результатів буде здійснюватися протягом одного року. Основні позитивні результати від впровадження розробки очікуються протягом 3-х років після її впровадження.

Одним із основних позитивних результатів є зменшення ціни експлуатації системи.

При реалізації результатів наукової розробки покращується якість програмного продукту, що дозволяє зменшити ціну його експлуатації на 150 грн. Кількість одиниць реалізації програмного засобу також збільшиться: протягом першого року — на 700 шт., протягом другого року — ще на 450 шт., протягом третього року — ще на 300 шт.

Реалізація продукції до впровадження результатів наукової розробки складала 50 шт., а ціна — 250 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового за такою формулою 4.11:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}}\Delta N)_i \quad (4.11)$$

де $\Delta\Pi_0$ – покращення основного оціночного показника від впровадження результатів розробки у даному році. Зазвичай таким показником може бути ціна одиниці нової розробки;

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

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

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

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт 0,8333.

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $= 0,2 \dots 0,3$;

v – ставка податку на прибуток. $v = 18\%$.

Тоді, збільшення чистого прибутку підприємства протягом першого року складе:

$$\Delta\Pi_1 = [150 \cdot 50 + (250 + 150) \cdot 700] \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 39290,09 \text{ грн}$$

Протягом другого року:

$$\Delta\Pi_2 = [150 \cdot 50 + (250 + 150) \cdot (700 + 450)] \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 61\,710 \text{ грн}$$

Протягом третього року:

$$\Delta\Pi_3 = [150 \cdot 50 + (250 + 150) \cdot (700 + 450 + 300)] \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 79900 \text{ грн}$$

Отже, протягом трьох років підприємство може розраховувати на збільшення чистого прибутку від реалізації наукової розробки.

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність E_{abc} вкладених інвестицій розраховується за формулою:

$$E_{abc} = (ПП - PV), \quad (4.12)$$

де ПП – приведена вартість всіх чистих прибутків, які отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій $PV = ZB = 48\,299,8$ грн.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 4.1.



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

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.13)$$

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

τ – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

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

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

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{48\,063,6}{(1+0,1)^0} + \frac{39\,290,09}{(1+0,1)^2} + \frac{61\,710}{(1+0,1)^3} + \frac{79\,900}{(1+0,1)^4} = 181\,474,87 \text{ грн.}$$

Тоді розрахуємо $E_{\text{абс}}$:

$$E_{abc} = 181\,474,87 - 48\,299,8 = 133\,175,07 \text{ грн.}$$

Оскільки $E_{abc} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B за формулою:

$$E_B = \sqrt[T]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.14)$$

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

PV – теперішня вартість інвестицій $PV = 3B$, грн;

T – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_B = \sqrt[3]{1 + \frac{133\,175,07}{48\,063,6}} - 1 = 0,56 \text{ або } 56\%$$

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

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2020 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_B = 56\% > \tau_{\text{мін}} = 0,3 = 30\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ок}}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_B}, \quad (4.16)$$

Якщо $T_{ок} < 3 \dots 5$ -ти років, то фінансування наукової розробки є доцільним.

$$T_{ок} = \frac{1}{0,56} = 1,7 \text{ років}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним, оскільки $T_{ок} < 3$ років.

4.5 Висновок

Результати проведених розрахунків дають можливість зробити висновок про доцільність розробки та впровадження нашої наукової роботи. Це підтверджують такі показники як:

- абсолютна ефективність вкладених інвестицій, яка дорівнює 133 175,07 грн., що є більшим 0 і вказує на те, що інвестор може бути зацікавленим у нашій розробці;

- відносна ефективність наукової розробки становить 56%, що є вищим за мінімальну ставку дисконтування (30%), тому вкласти кошти у нашу розробку є вигідніше, ніж покласти кошти на депозит;

- термін окупності вкладених у реалізацію наукового проекту інвестицій складе 1,7 років, що є менше 3-ох і вказує швидку окупність вкладених інвестицій.

Крім того, розраховано, що наукова розробка зменшить витрати протягом 3-х років за рахунок покращення її якості порівняно з існуючими аналогами.

Усе це, узятє разом, забезпечує прийняття рішення про доцільність виготовлення нового продукту.

ВИСНОВКИ

В результаті виконання комплексної магістерської кваліфікаційної роботи було розв'язано задачу розробки інтелектуальної інформаційної технології та програмного забезпечення розпізнаванні нетипових ситуацій на відео в частині детектування об'єктів на основі згорткової нейронної мережі.

У першому розділі магістерської роботи було сформульовано постановку задачі детектування об'єктів на відео при розпізнаванні нетипових ситуацій. Було проведено аналітичний огляд відомих методів і засобів детектування об'єктів на відео при розпізнаванні нетипових ситуацій, виявлені їх можливості, переваги і недоліки. Обґрунтована перспективність застосування методу детектування об'єктів на відео на основі штучних нейронних мереж. Також було проаналізовано існуючі програми-аналоги до програмного забезпечення детектування об'єктів на відео.

У другому розділі магістерської кваліфікаційної роботи було обґрунтовано вибір типу нейронної мережі - згорткова, та розроблено архітектуру згорткової нейронної мереж для розв'язання задачі детектування об'єктів на відео, За основу взято архітектуру YOLO, проаналізовано метод та математичну модель навчання згорткової мережі. Мережа складатиметься з вхідного шару розмірністю 448×448 , 6 згорткових шарів, 4 агрегувальних шарів та 2 повнозв'язних вихідних шари. Функція, яка буде використана в шарах агрегування – максимізаційна (max-pooling). У підсумку було розроблено структуру інформаційної технології розпізнавання нетипових ситуацій на відео у частині детектування об'єктів на основі згорткової нейронної мережі та розроблено структурну схему програмної системи розпізнавання нетипових ситуацій на відео.

У третьому розділі в ході практичної реалізації інформаційної технології розпізнавання нетипових ситуацій на відео у частині детектування об'єктів було обґрунтовано вибір мови програмування Python та середовища програмування Visual Studio Code. Серед бібліотек та фреймворків для

детектування об'єктів були обрані OpenCV, Tensorflow та Darknet. Було розроблено UML діаграму програми виявлення нетипових ситуацій та розроблено навчальну вибірку для навчання згорткової нейронної мережі задачі детектування транспортних засобів. Також було описано роботу основних частин програми та наведено відповідні фрагменти коду. В результаті було розроблено програмне забезпечення розпізнавання нетипових ситуацій на відео, яке порівняно з аналогом має кращу на 4% достовірність детектування окремих об'єктів і кращу на 3% достовірність детектування об'єктів по відео в цілому. Таким чином, мета роботи досягнута – достовірність детектування об'єктів на відео при розпізнаванні нетипових ситуацій підвищена.

У четвертому розділі було проведено розрахунки, які дають можливість зробити висновок про доцільність розробки та впровадження нашої наукової роботи. Це підтверджують такі показники: абсолютна ефективність вкладених інвестицій дорівнює 133 175,07 грн., що є більшим 0 і вказує на те, що інвестор може бути зацікавленим у нашій розробці; відносна ефективність наукової розробки становить 56%, що є вищим за мінімальну ставку дисконтування (30%), тому вкласти кошти у нашу розробку є вигідніше, ніж покласти кошти на депозит; термін окупності вкладених у реалізацію наукового проекту інвестицій складе 1,7 років, що є менше 3-ох і вказує на швидку окупність вкладених інвестицій. Крім того, розраховано, що наукова розробка зменшить витрати протягом 3-х років за рахунок покращення її якості порівняно з існуючими аналогами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. О. Колесницький, С. Кукунін, М. Дерев'янку, і А. Преподобний Мендеш Да Майа, Розпізнавання нетипових ситуацій на дорозі за допомогою згорткової нейронної мережі, ОЕІЕТ, vol 38, № 2, с. 38-44, Бер 2020. DOI: <https://doi.org/10.31649/1681-7893-2019-38-2-38-44>

2. Преподобний Мендеш да Майа А. А., Колесницький О. К., Дерев'янку М. Ю. «Визначення аномальних ситуацій за допомогою нейронних мереж», в Матеріали конференції «XLVIII Науково-технічна конференція підрозділів Вінницького національного технічного університету (2019)», Вінниця, 2019. С. 801-803 [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/index/pages/view/zbirn2019> Дата звернення: Черв. 2019

3. Колесницький О. К., Преподобний Мендеш да Майа А. А., Дерев'янку М. Ю., «Дослідження ефективності використання згорткової нейронної мережі для визначення нетипових ситуацій на дорозі», в Матеріали конференції «XLIX Науково-технічна конференція підрозділів Вінницького національного технічного університету (2020)», Вінниця, 2020, с. 983-986. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/allvntu/index/pages/view/zbirn2020> Дата звернення: Черв. 2020.

4. Колесницький О., Дерев'янку М., Преподобний Мендеш да Майа А. А. «Використання згорткової нейронної мережі для визначення нетипових ситуацій на дорозі», «ІНТЕРНЕТ-ОСВІТА-НАУКА-2020», XII Міжнародна науковопрактична конференція ІОН-2020, 26-29 травня, 2020 : Збірник праць. – Вінниця : ВНТУ, 2020 – с. 55-57. Режим доступу: <https://conferences.vntu.edu.ua/index.php/allvntu/index/pages/view/zbirn2020> Дата звернення: Черв. 2020

5. 1. Т. Басюк, Я. Пушко Аналіз та класифікація основних методів розпізнавання образів на площині Одеса: Наука і техніка, 2005. 68 с.

6. Мерков А. Б. Распознавание образов. Введение в методы статистического обучения Ульяновск: УлГТУ, 2009. 183 с.

7. Комп'ютерні науки (лекції) [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://knhelp.wordpress.com/2012/04/19/%D0%BB7-%D0%BF%D0%BE%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D>
Назва з екрана

8. В. М. Заяць, Р. М. Камінський Методи розпізнавання образів : Навч. посіб. для студ. /; Нац. ун-т "Львів. політехніка". - Л., 2004. - 173 с

9. Ш.-К. Чэн Принципы проектирования систем визуальной информации М.: Мир, 1994

10. А. С. Довбиш І. В. Шелехов Основи теорії розпізнавання образів Навчальний посібник Частина 1 Сумський Державний Університет 2015р 109с.

11. IEEEExplore Digital Library [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://ieeexplore.ieee.org/document/4732530/authors#authors> Назва з екрана

12. Profesional Solutions LEATER [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://leater.com/services/sistema-intellektualnogo-videoanaliza.html> Назва з екрана

13. Viatec.ua [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://viatec.ua/news?id=125> Назва з екрана

14. Jeannette Lawrence, Sylvia Luedeking Introduction to neural networks :design, theory and applications. Nevada City, Calif. : California Scientific Software, 1994.

15. ResearchGate [Електронний ресурс] : [Веб-сайт]. – Режим доступу: https://www.researchgate.net/publication/224364818_Automatic_Daytime_Road_Traffic_Control_and_Monitoring_System – Назва з екрана.

16. Frank Millstein Convolutional Neural Networks in Python: Beginner's Guide to Convolutional Neural Networks in Python. CreateSpace Independent Publishing Platform, 2018. 120 с.

17. Zhang, Wei. Parallel distributed processing model with local space-invariant interconnections and its optical architecture CreateSpace Independent Publishing Platform, 1990. 107 с.

18. Python. Documentation [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://docs.python.org/3/> (дата звернення 01.02.2019) – Назва з екрана.

19. OpenCV. Open Source Computer Vision [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://docs.opencv.org/master/> (дата звернення 02.02.2019) – Назва з екрана.

20. OpenCV. Open Source Computer Vision [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://docs.opencv.org/master/> (дата звернення 02.02.2019) – Назва з екрана.

21. GitHub. OpenCV issues [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://github.com/opencv/opencv/issues> (дата звернення 03.02.2019) – Назва з екрана.

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