НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет електроніки
(повна назва інституту/факультету)

Кафедра акустичних та мультимедійних електронних систем
(повна назва кафедри)

«На правах рукопису»
УДК 004.032.6

«До захисту допущено»
Завідувач кафедри
_____ Сергій НАЙДА
(підпис) (ініціали, прізвище)

“11” травня 2021 р.

Магістерська дисертація

зі спеціальності (спеціалізації) 171 Електроніка (Електронні системи мультимедіа та засоби Інтернету-речей)
(код і назва спеціальності)

на тему: «Дослідження методів та засобів розпізнавання елементів зображення для систем відеоспостереження».

Виконав студент II курсу, групи ДВ-92мн
(шифр групи)

Борисов Гліб Олександрович
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник к.т.н., доц. Трапезон К.О.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент доц. кафедри ЕПС доц., к.т.н. Михайлов С.Р. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент Борисов Г.О

Київ – 2021 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут (факультет) Факультет електроніки
(повна назва)

Кафедра акустичних та мультимедійних електронних систем
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою

Спеціальність (освітня програма) 171 Електроніка

(Електронні системи мультимедіа та засоби Інтернету речей)

(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій НАЙДА
(підпис) (ініціали, прізвище)
«11» травня 2021 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Борисову Глібу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Дослідження методів та засобів розпізнавання елементів зображення для систем відеоспостереження».

Науковий керівник дисертації к.т.н., доц. Трапезон Кирило Олександрович .
(науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

затверджені наказом по університету від « 15 » березня 2021 р. № 835-с

2. Строк подання студентом дисертації 05.05.2021 р.

3. Об'єкт дослідження: камера системи відеоспостереження з аналізом зображень на основі нейронної мережі .

4. Предмет дослідження (Вхідні дані – для магістерської дисертації за освітньо-науковою програмою): кількість класів розпізнавання – 80; кількість нейронних систем для практичної частини дослідження – 3; частота зйомки для секвенції – 24 кадри за секунду; кількість статичних зображень для аналізу – не менше 9; умови проведення експерименту – стаціонарні і незмінні.

5. Перелік завдань, які потрібно розробити: розглянути основні ознаки, за якими можна провести розпізнавання об’єктів на площині зображення; визначити основні типи систем розпізнавання; розглянути основні підходи з реалізації класифікатора системи розпізнавання; дослідити архітектуру нейронної мережі дотично до систем розпізнавання об’єктів; розробити програмний алгоритм та провести практичне тестування нейронних мереж для набору статичних зображень та секвенції, що отримано з камери відео спостереження за різних умов дорожнього руху.
6. Перелік графічного (ілюстративного) матеріалу: 15 слайдів презентації, основними назвами плакатів якої є постановка проблеми, особливості методів та засобів розпізнавання елементів зображення, архітектура нейронної мережі, результати проведеного практичного експерименту та їх аналіз.
7. Орієнтовний перелік публікацій: 1) Борисов Г.О., Трапезон О.Г., Трапезон К.О. Особливості оброблення сигналів із застосуванням програмних засобів IOS// Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. – 2019. – Том. 30 (69). – №6. – Частина 1. - С. 22-26. DOI:https://doi.org/10.32838/2663-5941/2019.6-1/04 (ISSN 2663-5941. Журнал відноситься до категорії Б (технічні науки). 2) Борисов Г.О., Гумен Т.Ф., Трапезон К.О. Дослідження програмних особливостей об’єднання Android things на основі концепції Інтернету речей// Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки – 2020. – Том.31 (70). – №1. - С. 29-35. DOI:https://doi.org/10.32838/2663-5941/2020.1-1/06 (ISSN 2663-5941. Журнал відноситься до категорії Б (технічні науки).
8. Дата видачі завдання 10. 09. 2019 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Аналіз задач розпізнавання зображень. Програмні підходи. ОС IOS та Android Things.	11.09.2019 – 30.12.2019	Виконано
2	Дослідження принципів реалізації класифікаторів розпізнавання зображень.	1.01.2020 – 31.01.2020	Виконано
3	Аналіз нейронних мереж, принципів та підходів навчання для систем розпізнавання елементів зображення	1.02.2020 – 31.05.2020	Виконано
5	Практична частина дослідження. Формулювання даних до проведення практичного експерименту	01.09.2020 – 31.10.2020	Виконано
7	Оформлення пояснювальної записки	01.11.2020 – 05.05.2021	Виконано
8	Підготовка та оформлення презентації для доповіді	06.05.2021 – 12.05.2021	Виконано

Студент Гліб Борисов

(підпис) (ініціали, прізвище)

Науковий керівник Кирило Трапезон

(підпис) (ініціали, прізвище)

УДК 004.032.6

РЕФЕРАТ

Борисов Г.О. Дослідження методів та засобів розпізнавання елементів зображення для систем відеоспостереження: магістерська дис. : 171 Електроніка. Київ, КПІ ім. Ігоря Сікорського, 2021. 102 с.

Магістерська дисертація: 102 с., 69 рис., 2 табл., 1 дод., 19 джерел.

Ключові слова: мережа, зображення, об’єкт, клас, секвенція, моделювання, методика, мультимедіа.

Актуальність дослідження. Системи відеоспостереження в умовах щільної забудови відіграють не останню роль в системі безпеки та контролю інфраструктури міста. Окрему роль в цьому відіграють транспортні артерії міста і контроль за дорожнім рухом. Так, дорожньо-транспортні пригоди трапляються майже щодня і їх фіксація та аналіз відіграють дуже важливе значення. Сучасні прилади контролю за рухом транспортних засобів потребують не лише автономності роботи, якісної картинки, незалежності від природніх умов, але й швидкого аналізу транспортної ситуації на основі отриманого візуального контенту. Щоб забезпечити ефективність цього процесу було запроваджено використання нейронних мереж для розпізнавання елементів зображення.

Мета дослідження полягає у створенні програмного алгоритму в спеціалізованій програмі, який б дозволив застосувати різні нейронні мережі для аналізу дорожньої ситуації міста в межах функціонування інтелектуальної камери контролю за рухом транспортних засобів.

Завдання для досягнення мети: розглянути основні ознаки, за якими можна провести розпізнавання об’єктів на площині зображення; визначити основні типи систем розпізнавання; розглянути основні підходи з реалізації класифікатора системи розпізнавання; дослідити архітектуру нейронної мережі дотично до систем розпізнавання об’єктів; розробити програмний алгоритм та провести практичне тестування нейронних мереж для набору статичних зображень та секвенції, що отримано з камери відео спостереження за різних умов дорожнього руху.

Об’єкт дослідження: система розпізнавання елементів зображення.

Предмет дослідження: інструменти та програмні засоби роботи з цифровим зображенням.

Методи дослідження: алгоритми та методи, які визначені в основі функціонування програмного алгоритму на мові Python в середовищі PyCharm.

Наукова новизна отриманих результатів: 1) розроблено алгоритм створення програми для розпізнавання зображень об’єктів на основі нейронної мережі; 2) проведено аналіз трьох нейронних мереж на різному візуальному матеріалі з формулюванням відповідних рекомендацій до впровадження в складі системи відеоспостереження.

Практичне значення одержаних результатів: результати роботи можуть бути використані при створенні інтелектуальних систем відеоспостереження в національній поліції України.

Апробація результатів дисертації: публікація двох статей у фаховому журналі з технічних наук категорії Б.

SUMMARY

Master's dissertation: 102 pp., 69 figs., 2 tables, 1 supplements, 19 sources.

Keywords: network, image, object, class, sequence, modeling, technique, multimedia.

Relevance of research. Video surveillance systems in densely built-up areas play not the least role in the system of security and control of the city's infrastructure. A special role in this is played by the city's transport arteries and traffic control. Yes, traffic accidents happen almost every day and their recording and analysis are very important. Modern devices for monitoring the movement of vehicles require not only autonomy, quality picture, independence from natural conditions, but also a rapid analysis of the transport situation on the basis of the received visual content. To ensure the effectiveness of this process, the use of neural networks to recognize image elements was introduced.

The purpose of the work is to create a software algorithm in a specialized program that would allow the use of various neural networks to analyze the road situation of the city within the operation of an intelligent camera for monitoring the movement of vehicles.

Objectives to achieve the goal: to consider the main features that can be used to recognize objects in the image plane; identify the main types of recognition systems; consider the main approaches to the implementation of the classifier of the recognition system; to study the architecture of a neural network in relation to object recognition systems; develop a software algorithm and conduct practical testing of neural networks for a set of static images and sequences obtained from a video surveillance camera under different traffic conditions.

Object of research: image element recognition system.

Subject of research: tools and software for working with digital images.

Research methods: algorithms and methods that are defined in the basis of the operation of a software algorithm in Python in PyCharm.

Scientific novelty of the obtained results: 1) the algorithm of creation of the program for recognition of images of objects on the basis of a neural network is developed; 2) the analysis of three neural networks on different visual material with the formulation of appropriate recommendations for implementation in the video surveillance system.

Practical significance of the obtained results: the results of the work can be used in the creation of intelligent video surveillance systems in the National Police of Ukraine.

Approbation of dissertation results: publication of two articles in a professional journal of technical sciences of category B.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 9

ВСТУП 10

1 ЗАДАЧІ РОЗПІЗНАВАННЯ ЗОБРАЖЕННЯ 11

1.1 Ознаки розпізнавання зображень 11

1.1.1 Ланцюгові коди 12

1.1.2 Сплайн - апроксимація кордону об’єкту 13

1.2 Типи систем розпізнавання 14

1.3 Задачі розпізнавання на зображеннях 16

1.4 Опис камери відеонагляду. Структурна схема 17

2 ПРИНЦИПИ РЕАЛІЗАЦІЇ КЛАСИФІКАТОРІВ РОЗПІЗНАВАННЯ ЗОБРАЖЕННЯ 20

2.1 Байєсівський підхід розпізнавання образів 20

2.2 Розпізнавання образів на основі принципу кластеризації 22

2.3 Логічні методи в розпізнаванні образів 23

2.4 Метод градієнтного спуску 24

2.5 Порівняння методів реалізації класифікаторів розпізнавання 25

3 НЕЙРОННІ МЕРЕЖІ 28

3.1 Проблеми розпізнавання об'єктів зображення 28

3.2 Поняття нейрону 29

3.2 Класифікація нейронних мереж 31

3.3. Парадигми навчання нейронних мереж 33

3.4 Алгоритми навчання нейронних мереж 36

3.4.1 Алгоритм навчання одного нейрону Хебба 36

3.4.2 Персептроний метод навчання 37

3.4.3 Навчання багатошарової нейронної мережі методом зворотного поширення помилки 37
3.5 Переваги нейронних мереж в системах розпізнавання об’єктів 40
4 ПРАКТИЧНА ЧАСТИНА ДОСЛІДЖЕННЯ 42
4.1 Вихідні дані до експерименту 42
4.1.1 Нейронна мережа RetinaNet 44
4.1.2 Нейронна мережа YOLO 49
4.1.3 Нейронна мережа TinyYOLO 53
4.2 Алгоритм створення програми для розпізнавання зображень об’єктів на основі нейронної мережі 53
4.2.1 Підготовчий етап дослідження 53
4.2.2 Алгоритм створення програми для тестування нейронних мереж 57
4.3 Обробка та аналіз результатів експерименту 62
4.3.1 Аналіз статичних зображень з камер відеоспостереження 62
4.3.2 Аналіз секвенції кадрів з камери відеоспостереження 83
4.3.3 Перевірка моделі нейронної мережі до функції навчання 88
ВИСНОВКИ 92
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 95
ДОДАТОК А 97

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

sRGB – Standard Red Green Blue;

FPS – Frames Per Second;

IDE – Integrated Development Environment;

JPEG – Joint Photographic Experts Group;

LED – Light-Emitting Diode;

VGA – Video Graphics Array;

ГБ – Гігабайт;

НМ – Нейронна Мережа;

ОЗП – Оперативна память;

ПК – Персональний Комп’ютер;

ЦСП – Цифровий Сигнальний Процесор;

ШІ – Штучний Інтелект.

ВСТУП

Системи відеоспостереження в умовах щільної забудови відіграють не останню роль в системі безпеки та контролю інфраструктури міста. Окрему роль в цьому відіграють транспортні артерії міста і контроль за дорожнім рухом. Так, дорожньо-транспортні пригоди трапляються майже щодня і їх фіксація та аналіз відіграють дуже важливе значення. Сучасні прилади контролю за рухом транспортних засобів потребують не лише автономності роботи, якісної картинки, незалежності від природніх умов, але й швидкого аналізу транспортної ситуації на основі отриманого візуального контенту. Щоб забезпечити ефективність цього процесу було запроваджено використання нейронних мереж для розпізнавання елементів зображення. Нейронні мережі широко використовуються для розпізнавання образів, вони навчаються на основі наданих їм даних і розпізнають об’єкти. Програми які реалізують нейронну мережу для розпізнавання розгортаються для роботи на смарт камерах. Існує багато різних побудованих нейронних мереж які вирішують цю задачу найбільш поширені у світі нейронні мережі RetinaNet, YOLO v.3, TinyYOLO. Дана магістерська дисертація присвячена насамперед на виявлення технічних особливостей, які погіршують або поліпшують роботу нейронної мережі в цілому і які надалі слід обов’язково враховувати при проектуванні прототипів систем розпізнавання образів з елементами навчання.

1 ЗАДАЧІ РОЗПІЗНАВАННЯ ЗОБРАЖЕННЯ

1.1 Ознаки розпізнавання зображень

Формування ознак - перший етап в будь-якій системі розпізнавання образів. І якість всієї системи виявляється залежною від того, наскільки добре підібрані ознаки для опису об'єкта - в даному випадку, те що знаходиться на площині зображення.

Одним із способів визначення ознак можна вважати підхід на основі використання геометричних характеристик представлених на зображенні об'єктів. Це можуть бути, наприклад, такі ознаки:

- геометричні розміри зображеного об'єкта по вертикалі або горизонталі;
- відстань між найбільш віддаленими точками на зображеному об'єкті;
- периметр і площа зображеного об'єкта;
- компактність об'єкта (як співвідношення між його площею і периметром);
- числові характеристики описаних або вписаних в зображення об'єкта геометричних фігур, таких як кола, багатокутники тощо.

Досить специфічними ознаками, але які широко використовуються в геометричній групі є ознаки, пов'язані з представленням геометрії контуру об'єкта. Як відомо контур - одна з найбільш важливих характеристик зображеного об'єкта при його сприйнятті зором людини. Тому опис контуру - один з поширених завдань, які вирішуються в процесі аналізу і обробки зображень [2]. До найбільш типових методів опису контуру, використовуваних також і для вирішення завдань розпізнавання, відносяться наступні.

1.1.1 Ланцюгові коди

Метод ланцюгового кодування для представлення контуру об'єкта був запропонований Фриманом [2]. Він полягає в тому, щоб границі об'єкта, розташованого на дискретній сітці, можна було б представити у вигляді набору елементарних відрізків. Тоді повною характеристикою кордону в кожній точці є напрямок необхідного відрізка. Рисунок 1.1 а ілюструє даний метод. В даному випадку мається на увазі, що точки на границі є тільки 4-х зв'язковими (відрізок відкладається лише в 4-х напрямках). Іноді застосовують модифікацію даного методу, що використовує 8-зв'язну модель, як на малюнку 1.1 б.

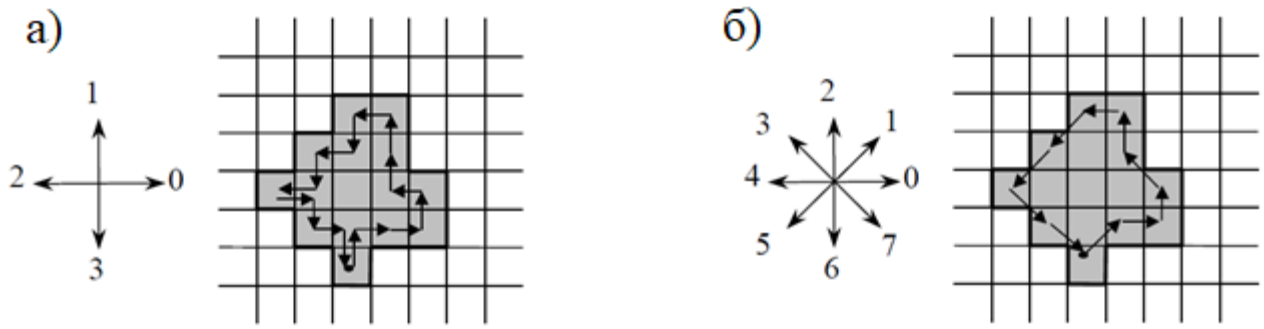


Рисунок 1.1 – Приклади побудови ланцюгового коду:
а) 4 - зв'язкова модель кордону б) 8 - зв'язкова модель кордону [2]

Безсумнівною перевагою визначення кордону зображуваного об'єкта ланцюговим кодом є простота реалізації алгоритму його опису, простота отримання на основі цього опису деяких інших геометричних характеристик об'єкта (наприклад: периметр, площа, лінійні розміри по вертикалі і горизонталі), можливість досягнення інваріантності опису до перетворенням подібності - масштабування зображення, його переносу і повороту. Основним недоліком є висока нестійкість одержуваних описів до спотворень в зображеннях.

1.1.2 Сплайн - апроксимація кордону об’єкту

Апарат сплайнів досить широко використовується для практичних застосувань, зокрема, для вирішення завдання опису кордону. Суть методу сплайн апроксимації - уявлення кордону об'єкта у вигляді кусково-поліноміальної функції часто з виконанням ряду обмежень, що накладаються на її гладкість. Найбільш популярними є сплайни третього порядку, оскільки володіють найменшим ступенем, при якій поліноміальна функція може змінювати знак кривизни. Останнє дозволяє домогтися певної гладкості в точках з'єднання сплайнів за рахунок рівності перших похідних в цих точках.

При використанні найбільш простого завдання $k(s)$ в вигляді горизонтальних прямих з розривом в точках зміни напрямку кордону, отримуємо уявлення контуру у вигляді полілиній - з'єднуємих між собою відрізків прямих (рисунок 1.2). Точки розбивки полілінії вибираються, виходячи з міркувань найкращого представлення контуру [2].

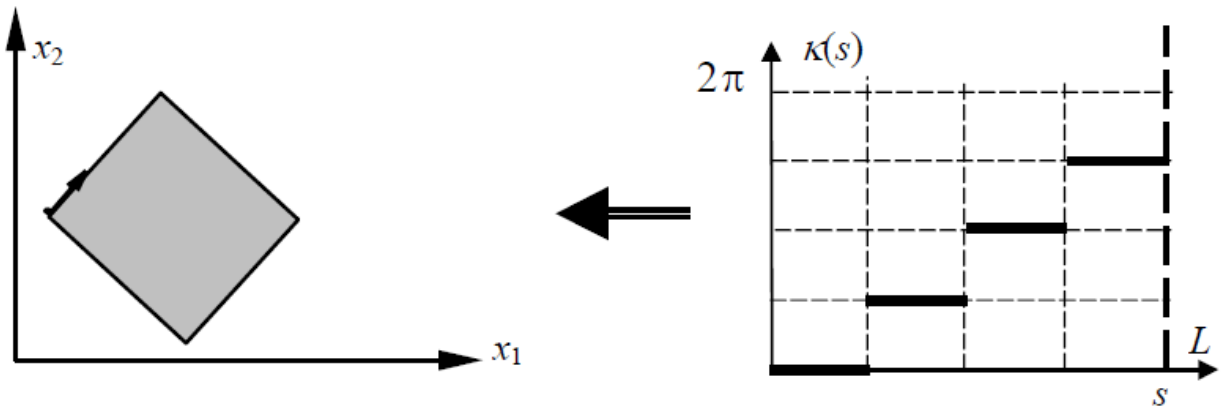


Рисунок 1.2 – Подання контуру об'єкта полілінією [2]

Треба відзначити, що викладені методи і алгоритми далеко не вичерпують весь комплекс методів представлення контуру об'єкта для його розташування, але є найбільш часто використовуваними в практичних додатках.

1.2 Типи систем розпізнавання

Можна виділити кілька критеріїв класифікації систем розпізнавання. Один з таких критеріїв - за характером інформації про ознаки [3]:

- детерміністські;
- імовірнісні;
- логічні;
- структурні;
- комбіновані.

Інший критерій - за кількістю апріорної інформації про розпізнавані об'єкти.

Розрізняють три основних типи систем розпізнавання.

1. Системи без навчання. Кількість апріорної інформації достатньо точна для визначення алфавіту ознак (повного набору ознак), формування словника ознак (тобто визначення мінімального набору ознак, достатнього для вирішення завдань розпізнавання) і визначення меж класів [3]. В цьому випадку в системі розпізнавання відсутній блок «навчання».

2. Системи, на основі навчання з “вчителем”. Кількості апріорної інформації достатньо лише для вибору алфавіту ознак і формування словника ознак, але не для визначення кордонів між класами. Системі розпізнавання пред'являється деяка безліч об'єктів $E = \{x_1, \dots, x_n\}$, яка називається навчальною безліччю (навчальна виборка), із зазначенням, до яких класів ці об'єкти належать. Система сама повинна налаштувати параметри правил класифікації таким чином, щоб виконувалася умова мінімальності помилки неправильної класифікації. Наприклад, (рис. 1.3) безліч об'єктів навчальної вибірки можна розділити двома прямими так, що об'єкти x_1, x_2, x_3 потраплять в перший клас w_1, об'єкти x_4, x_5, x_6 – в другий клас w_2, об'єкти x_7, x_8 – в третій клас w_3. За допомогою процедури навчання може бути також вирішена задача зменшення словника ознак [3].

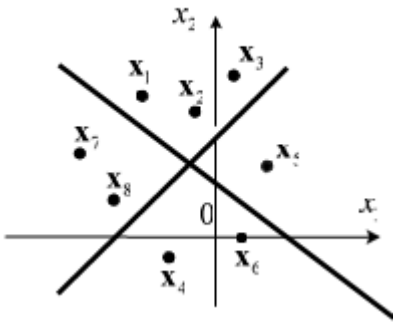


Рисунок 1.3 – Поділ об’єктів за класами

3) Системи, які засновані на самонавчанні (на поясненні). Кількості апріорної інформації недостатньо навіть для формування словника ознак. В цьому випадку в систему розпізнавання образів вводиться список правил, що пояснює завдання розпізнавання образів. Цей список правил виробляється, як правило, експертами - фахівцями в даній області знань, такі системи називають експертними (інтелектуальними). Система розпізнавання, виходячи з цього набору правил, повинна сама сформувати словник ознак і визначити межі класів. При цьому, як правило, використовуються логіко-лінгвістичні методи обробки даних. У такій системі процес пошуку класифікаційного рішення називають логічним висновком або висновком на знаннях. Типовою сферою застосування таких систем є медична діагностика. Наприклад, система розпізнавання для кардіологів (рис. 1.4, 1.5) [3].

Рисунок 1.4 – Амплітуда серцебиття [3]

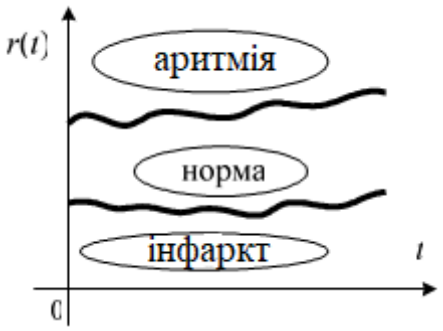


Рисунок 1.5 – Спрощений аналіз амплітуди серцебиття [3]

Тут $a(t)$ - зміна амплітуди серцебиття в залежності від часу, $r(t)$ - зміна відстані між двома максимумами амплітуди $a(t)$.

1.3 Задачі розпізнавання на зображеннях

Задачі які стосуються процедури розпізнавання об’єктів на зображенні можна поділити на дві групи: розпізнавання або класифікація зображень; пошук і розпізнавання об’єктів на зображеннях. Перша група передбачає аналіз зображення в цілому. Тобто, тут все зображення відносять до певного класу на основі фіксації певного набору загальних ознак. Іншими словами, визначається певний набір ознак в зображенні і за цим набором встановлюється певний номер класу для зображення. Прикладом задач першої групи – розпізнавання обличчя, розпізнавання відбитків пальців.

Друга група задач полягає у тому, що процес розпізнавання пов’язаний з пошуком геометричних об’єктів на усієї області спостереження. Об’єкти в даному випадку – відносно невеликі локальні області, форми, поява котрих може відбутись в будь-якій точці зображення. Результатом цієї задачі є не лише клас об’єкту, а й його положення, розмір, колір, орієнтація в площині зображення, тощо. Прикладом задач цієї групи можна назвати дешифрування знімків з висоти повітря. В даному випадку на аналіз проходить кожний фрагмент зображення. І за цим фрагментом формуються ознаки, і визначається приналежність фрагменту до певного класу. В практичній частині дисертації (розділ 4) саме задача другої групи буде об’єктом дослідження.

Оскільки, за другою групою процедури розпізнавання об’єктів необхідно враховувати ще і особливості самих знімків, або секвенції кадрів, то доцільно в рамках дисертації розглянути структуру камери відеоспостереження. Зрозуміло, що якість розпізнавання не в останню чергу залежить і від параметрів самого зображення. Так, як було зазначено вище безпосередній вплив на точність визначення об’єктів, їх характеристик впливають наявні спотворення зображень (шум, зернистість, розмитість, спотворення границь контурів, спотворення передачі кольорів, тощо). А оскільки причиною цих спотворень може бути функціонування самої камери фіксації дорожнього руху, то варто визначити які структурні елементи камери слід враховувати при проведенні процедури розпізнавання об’єктів на зображенні.

1.4 Опис камери відеонагляду. Структурна схема

Так звані “інтелектуальні” камери являють собою систему, в якій додатково до захоплення зображення формується інформація, яка використовується для формування опису подій або прийняття рішень. Зазвичай Smart-камеру розглядають як автономну систему, розміщену в корпусі промислової відеокамери, яка містить цифровий процесор зі своїм програмним забезпеченням і всі необхідні комунікаційні інтерфейси, наприклад Ethernet (рис. 1.6) [1].

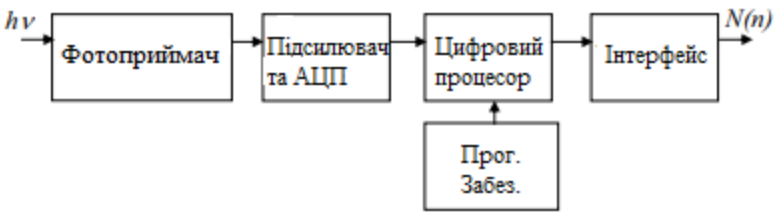


Рисунок 1.6 – Структурна схема інтелектуальної камери

Smart-камери являють собою компактну архітектуру системи технічного зору. Вони дозволяють створювати більш складні системи, що складаються з декількох камер, які мають більш низьку вартість в порівнянні з системами

складаються з традиційних ТБ-камер і ПК. Хоча smart-камери зазвичай використовуються для простих додатків, але сучасні камери можуть змагатися комп'ютером з точки зору обчислювальної потужності і функціональності.

Smart-камери, як правило, складаються з декількох (але не обов'язково всі) компонент:

- Фотоприймач (сенсорна світлочутлива матриця);
- Кодек для оцифровування зображення;
- Пам'ять для зображень;
- Процесор (часто цифровий сигнальний процесор ЦСП);
- Пам'ять програм і пам'ять даних (ОЗП, незалежна FLASH);
- Інтерфейси зв'язку (RS-232, Ethernet, Camera Link, ін.);
- Додаткові лінії зв'язку для управління і синхронізації;
- вбудований об'єктив (зазвичай C, CS або M-монтаж);
- Фонове підсвічування (зазвичай LED-діоди);
- Додатковий відеовихід (наприклад, VGA) [1].

Враховуючи наведений перелік структурних складових камери варто відмітити, з огляду на аналіз присутності на площині зображення спотворень, так елементи як об’єктив, фотоприймач і процесор. Останній елемент може бути використаний в якості арифметично-логічного пристрою для розв’язання задач розпізнавання зображень. До того ж, особливість smart-камер полягає в тому, що розробники можуть створити і завантажити в камери власні програми для вирішення задач машинного зору. І ця особливість власне може бути використана при реалізації систем розпізнавання зображення, що і буде використано у четвертому розділі дисертації при розгляді нейронних систем.

Висновки до 1 розділу. В першому розділі дисертації наведені основні ознаки, які використовуються при проведенні процедури розпізнавання зображення. Додатково визначено методи опису контурів об’єкту на зображенні та надано опис основних поширених систем розпізнавання. Крім цього, наведена структурна схема камери відеоспостереження, оскільки саме цей елемент є основним в системі відеоспостереження і використовується при

розв’язанні задач розпізнавання зображення. Наведені дані далі будуть використані при проведенні практичного експерименту дослідження.

2 ПРИНЦИПИ РЕАЛІЗАЦІЇ КЛАСИФІКАТОРІВ РОЗПІЗНАВАННЯ ЗОБРАЖЕННЯ

Розглянемо в цій частині дослідження найбільш типові реалізації класифікаторів образів об’єктів зображення, на основі існуючих математичних підходів та принципів. Основний розгляд в цьому розділі прикутий саме до першої групи з окреслених задач розпізнавання елементів на зображенні. Тобто, в першій частині розділу переважно буде йти мова про задачу віднесення зображення в цілому до певного класу.

2.1 Байєсівський підхід розпізнавання образів

Байєсівський підхід пов’язаний з статистичної природи спостережень. За основу береться припущення про існування імовірнісної міри на просторі образів, яка або відома, або може бути оцінена. Мета полягає в розробці такого класифікатора, який буде правильно визначати найбільш ймовірний клас для пробного образу. Тобто, завдання полягає у визначенні "найбільш ймовірного" класу [4].

Розглянемо в рамках дослідження приклад для двох класів Ω_1 та Ω_2. Природно вибрати вирішальне правило таким чином: об'єкт відносимо до того класу, для якого апостеріорна ймовірність вище. Загальновідомо, що таке правило класифікації за максимумом апостеріорної ймовірності називається Байєсовим правилом: якщо $P(\Omega_1|x) > P(\Omega_2|x)$, то x класифікується в Ω_1, інакше в Ω_2. Таким чином, для байєсівського вирішального правила необхідно отримати апостеріорні ймовірності $P(\Omega_i|x), i = 1, 2,$ Це можна зробити за допомогою математичної формули Байеса.

Формула Байєса дозволяє обчислити апостеріорні ймовірності подій через апріорні ймовірності і функції правдоподібності.

Нехай $A_1, A_2, \dots, A_n$– повна група несумісних подій; $\bigcup_{i=1}^{n} A_i = \Omega$. $A_i \cap A_j = \emptyset$, при і $\neq j$. Тоді апостеріорная ймовірність має вигляд:

$$P(A_i \mid B) = \frac{P(A_i)P(B \mid A_i)}{\sum_{i=1}^{n} P(A_i)P(B \mid A_i)},$$

де $P(A_i)$ – апріорна ймовірність події A_i, $P(B \mid A_i)$ – умова ймовірності події В за умови, що сталася подія A_i.

Розглянемо отримання апостеріорної ймовірності $P(\Omega \mid B)$, знаючи $P(\Omega)$ та $P(x \mid \Omega)$

$$P(AB) = P(A|B)P(B), P(AB) = P(B|A)P(A),$$

$$P(A|B)P(B) = P(A|B)P(A),$$

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

Якщо P(A) та P(A|B) описуються щільністю ймовірності $p(x)$ та $p(x|B)$, то

$$P(B|x) = \frac{p(x|B)P(B)}{p(x)} \rightarrow P(\Omega_i|x) = \frac{p(x|\Omega_i)P(\Omega_i)}{p(x)}.$$

При перевірці класифікації порівняння $P(\Omega_1|x)$ та $P(\Omega_2|x)$ еквівалентно порівнянню $p(x|\Omega_1)P(\Omega_1)$ та $p(x|\Omega_2)P(\Omega_2)$. У випадку, коли $P(\Omega_1|x) = P(\Omega_2|x)$, вважається, що міра безлічі x дорівнює нулю.

Таким чином, завдання порівняння по апостеріорної ймовірності зводиться до обчислення величин $P(\Omega_1), P(\Omega_2), p(x|\Omega_1), p(x|\Omega_2)$.

Будемо вважати, що у нас достатньо даних для визначення ймовірності приналежності об'єкта кожному з класів $P(\Omega_i), i = 1, 2$. Такі ймовірності називаються апріорними ймовірностями класів. А також будемо вважати, що відомі функції розподілу вектора ознак для класу $P(x|\Omega_i), i = 1, 2$. Вони в теорії називаються функціями правдоподібності x по відношенню до Ω_i. Якщо апріорні ймовірності і функції правдоподібності невідомі, то їх можна оцінити методами математичної статистики на безлічі прецедентів.

Наприклад, $P(\Omega_i) \approx \frac{N_i}{N}$, да $N_i -$ число прецедентів з $\Omega_i, i = 1, 2. N -$ загальне число прецедентів. $P(x|\Omega_i)$ може бути наближене гистограммой розподілу вектора ознак для прецедентів з класу Ω_i [4].

Отже, Байєсівський підхід до статистичних завдань ґрунтується на припущенні о існуванні деякого розподілу ймовірностей для кожного

параметра. Недоліком цього методу є необхідність постулювання як існування апріорного розподілу для невідомого параметра, так і знання його форми.

2.2 Розпізнавання образів на основі принципу кластеризації

Принцип кластеризації відповідає “предикатному” способу завдання множин, тобто системи обмежень за значеннями ознак. Принцип кластеризації застосовується в тих випадках, коли кожен образ представлений набором з *n* чисельних характеристик - параметрів. В цьому випадку його можна розглядати як вектор в *n*-вимірному просторі параметрів [5]:

$$x = \begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix}.$$

Такий *n*-мірний простір називають простором ознак (ПО). Типовим прикладом образів є пікселі мультиспектрального зображення. В цьому випадку ПО являє собою простір спектральних яскравостей, де кожен з векторів *n*-мірного базису відповідає каналу мультиспектрального зображення. Такі вектори зазвичай називають спектральними сигнатурами пікселів [5].

У зв'язку з появою гіперспектральних зображень, які дозволяють отримати для кожного пікселя практично безперервну залежність яскравості від довжини хвилі, спектральною ознакою пікселя найчастіше називають саме таку спектральну криву. Можна, таким чином, сказати, що спектральна сигнатура є дискретну апроксимацію спектрального образу пікселя.

У методах класифікації, заснованих на описаній моделі, передбачається, що кожному класу відповідає певна область в просторі ознак. Тобто, щоб розділити ці класи, потрібно побудувати границю відповідних областей. У задачах класифікації мультиспектральних зображень безліч спектральних сигнатур, що визначають кожну таку область в ПО, називають спектральної сигнатурою класу.

Тобто, щоб використати цей принцип для розпізнавання елементів, необхідно мати додаткову інформацію про саме зображення.

2.3 Логічні методи в розпізнаванні образів

Методи математичної логіки застосовуються в задачах розпізнавання образів в тих випадках, коли відсутні відомості про кількісний розподіл класів об'єктів по вимірюваним характеристикам. Тобто, навіть якщо є якісь приписані об'єктам значення параметрів, немає ніякої інформації, що дозволяє надійно визначити положення і кордони областей класів в просторі ознак. Але при цьому в нашому розпорядженні є відомості про логічні зв'язки між досліджуваними об'єктами і класами, які можуть бути записані з використанням булевих змінних і логічних операцій.

Такі ознаки об'єктів можуть приймати тільки два значення: 0 (*false*) і 1 (*true*). Значення *true* може означати або факт наявності певної ознаки, або істинність деякого укладення (висловлювання) про об'єкт; відповідно, *false* - або відсутність ознаки, або хибність висновку (висловлювання) [5].

Істинне значення висловлювання може полягати, наприклад, в прийнятті ознакою-параметром певного значення ($x_j = a$), в перевищенні цим значенням заданого порогу ($x_j > d$), або навіть просто в наявності у класу ознаки x_i. У цьому випадку в якості ознак в логічній системі розпізнавання ми будемо мати набори булевих змінних, кожна з яких визначає, виконується чи ні відповідна умова [5].

Індикаційні ознаки класів теж можуть, в свою чергу, бути записані у вигляді логічних висловлювань. Тобто в логічній системі розпізнавання і класи, і ознаки є функціями булевих змінних.

Один з найбільш поширених логічних методів розпізнавання - алгоритм обчислення оцінок (метод голосування) [6]. Він може застосовуватися в тих випадках, коли виділяються класи характеризуються наявністю певних комбінацій ознак-індикаторів, причому деякі з цих ознак або комбінацій мають різну значимість (вага) для різних класів.

2.4 Метод градієнтного спуску

Ще один з методів, який можна використати при процедурі розпізнавання образів це метод градієнтного спуску. Виділення контурів з довільними напрямками яскравості робиться за допомогою модуля градієнта яскравості:

$$|gradu(x,y)| = \sqrt{\left(\frac{\partial u(\mathrm{x},\mathrm{y})}{\partial \mathrm{x}}\right)^2 + \left(\frac{\partial u(\mathrm{x},\mathrm{y})}{\partial \mathrm{y}}\right)^2},$$

який пропорційний максимальній (у напрямку) швидкості зміни функції яскравості в даній точці і не залежить від напрямку контуру.

Градієнтні методи визначають розриви в інтенсивності представлення образу об'єкта. В ідеальному випадку ці методи визначають пікселі, що лежать на кордоні між об'єктом і фоном. На практиці даний ряд пікселів рідко повністю характеризує кордон через шум і розривів на кордоні. Таким чином, алгоритми виявлення контурів супроводжуються процедурами побудови меж об'єктів з відповідних послідовностей пікселів. Нижче розглянуто кілька методів, придатних для цієї мети.

Одним з найбільш простих підходів з'єднання точок контуру є аналіз характеристик пікселів в невеликій околиці (наприклад, в околиці розміром 2×2) кожної точки (x, y) образу, який вже піддався процедурі виявлення контуру. Всі точки, які є подібними, з'єднуються, утворюючи кордон з пікселів, що володіють деякими властивостями.

При такому аналізі для встановлення подібності пікселів контуру необхідно визначити:

1. Величину градієнта, необхідного для побудови контурного пікселя $G\,[f\,(x, y)]$;

2. Напрямок градієнта.

Таким чином, піксель контуру з координатами (x', y') подібний за величиною в певній раніше околиці (x, y) пікселу з координатами (x, y), якщо справедливо нерівність.

$$|G[u(x,y)] - G[u(x',y')]| \leq T,$$

де T - порогове значення. Напрямок градієнта встановлюється за кутом вектора градієнта

$$\theta = arctg\left[\frac{G_y}{G_x}\right]$$

де θ - кут (щодо осі x), уздовж якого швидкість зміни має найбільше значення. Тоді можна сказати, що кут пікселя контуру з координатами (x', y') в деякому околі (x, y) подібний до розі пікселя з координатами (x, y) при виконанні наступного нерівності

$$|\theta - \theta'| < A,$$

де A - граничне значення кута.

Враховуючи ці припущення, ми з'єднуємо точку в деякій околиці (x, y) з пікселем, які мають координати (x, y), якщо задовольняються критерії за величиною і напрямком. Рухаючись від пікселя до пікселя і представляючи кожну приєднувану точку як центр міста, процес повторюється для кожної точки образу. Враховуючи, що цей метод, як правило, використовується в задачах розпізнавання образів з використання нейронних мереж, то визначимо його основним в даному дослідженні.

2.5 Порівняння методів реалізації класифікаторів розпізнавання

Для розглянутих вище методів, за якими можна створити класифікатор системи розпізнавання зображення можна сформулювати наступні особливості:

1) Байєсівський метод має недолік в необхідність постулювання як існування апріорного розподілу для невідомого параметра, так і знання його форми.

2) Метод кластеризації, а саме кластерний аналіз має певні недоліки і обмеження. Зокрема, склад і кількість кластерів залежить від обраних критеріїв розбиття. При зведенні вихідного масиву даних до більш компактного особливу увагу можуть виникати певні спотворення, а також можуть втрачатись індивідуальні риси окремих об'єктів за рахунок заміни їх характеристиками узагальнених значень параметрів кластера. При проведенні

класифікації об'єктів ігнорується дуже часто можливість відсутності в даній сукупності будь-яких значень кластерів.

3) Логічний метод може застосовуватися в тих випадках, коли виділяються класи характеризуються наявністю певних комбінацій ознак-індикаторів, причому деякі з цих ознак або комбінацій мають різну значимість (вагу) для різних класів. Тому цей метод можна використовувати в наступних випадках:

- при створенні тематичних карт на основі результатів автоматизованої обробки аерокосмічної інформації, коли прийняті в аналізованої предметної області поняття і категорії не збігаються з тематичними класами, що виділяються на знімках автоматичними методами;
- при просторовому агрегування даних (класифікація ділянок території по атрибутам знаходяться на цих ділянках об'єктів), в тому числі при генералізації тематичних карт (угруповання різнотипних об'єктів за комплексом властивостей);
- при складанні тематичних карт в деяких слабо формалізованих предметних областях (ландшафтоведенії, грунтознавстві, геоботаніці і т.п.), коли класи і ознаки описуються сукупністю характеристик-індикаторів без вказівки їх кількісних співвідношень;
- в експертних системах різного призначення.

5) Метод градієнтного спуску являє собою один з найпростіших методів виділення об’єктів. І головною його особливістю може бути необхідність обробки зображення перед використанням метода.

Таким чином, окрім розглянутих методів є досить багато інших методів, які вирішують задачу розпізнавання, але на мою думку, найрезультативнішими і перспективними з точки зору точності, швидкодії є ті, які ґрунтуються саме на основі нейронних мереж. Крім того, вони дозволяють дуже просто реалізувати задачі машинного зору для розпізнавання об’єктів лише на основі відповідних бібліотек та програмних скриптів на мові Python. Можна відзначити, що методи розпізнавання, які не ґрунтуються на нейронних мережах, як правило не мають

функції навчання, а отже не можна виправляти помилки. Крім цього, для класичних методів розпізнавання важко зробити систему з ознаками універсальності. Тобто, при збільшенні класів розпізнавання невпинно зростає технічна складність реалізації. І технічні труднощі при використанні розглянутих методів вже виникають коли число класів стає більшим за десять. Натомість, при побудові нейронної мережі питання про обмеження кількості класів для класифікації образів насправді не стоїть. Через це, у наступному розділі дисертації розглянемо спочатку теоретичні основи створення нейронної мережі а потім проведемо практичне тестування декількох мереж для обраного контенту з камер відеоспостереження.

Висновки до 2 розділу. В цій частині дослідження розглянуто базові підходи до створення класифікаторів розпізнавання образів з визначенням відповідних особливостей та обмежень. Наприклад, байєсівський підхід вбачає для аналізу зображення наявність ймовірнісної міри на просторі образів зображення. Що у ряді випадків, може бути перепоною при проведенні практичних дослідів з розпізнавання елементів зображення.

3 НЕЙРОННІ МЕРЕЖІ

3.1 Проблеми розпізнавання об'єктів зображення

Розпізнавання образів являє собою класифікацією вхідних даних за допомогою виділення істотних ознак, якими характеризуються об’єкти, які підлягають розпізнаванню. Розпізнавання образів являється одним із найфундаментальніших проблем в теорії інтелектуальних інформаційних систем в рамках теорії машинного навчання. Проблема розпізнавання образів набула суттєвого значення в умовах інформаційного світу, коли людина просто не може контролювати всю інформацію яка надходить з певного джерела. Також задача розпізнавання образів має велике практичне значення. Розпізнавання образів широко використовується на дорожньо-транспортних шляхах, і в приміщеннях спеціального призначення, наприклад, аеропорти, на кордонах країн та інші.

Розпізнавання образів в конкретній ситуації пов’язано з врахуванням невизначеностей різної природи. У найпростішому випадку коли образи однозначно визначаються набором ознак, межі класів точно описуються, а самі класи не перетинаються, степінь невизначеності можна вважати мінімальною. Однією з основних задач розпізнавання образів є задача оптимального вибору системи ознак, які найбільш інформативно описують образи. Інша важлива задача – опис самих класів і формулювання умов, за якими той або інший образ можна віднести до відповідного класу. Тобто, повинна бути сформульована математична постановка задачі розпізнавання.

Функцію розпізнавання образів для систем відеоспостереження можна реалізувати за допомогою нейронних мереж. Всі вони мають різну будову та архітектуру, відмінності в алгоритмах навчання, якість розпізнавання об’єктів, тощо. Елементами таких мереж є нейрони, які виконано, наприклад, на основі бінарних порігових перетворювачах. Ці елементи функціонують за принципом “все або нічого”. І виявилось, що такі мережі з нейронами можна навчити розпізнаванню образів та узагальненню інформації. Фактично мають ознаки,

які характерні для живого мозку людини. Далі розглянемо особливості функціонування біологічного нейрону людини і на їх основі визначимо відповідну математичну модель нейрону і як власне можна описати нейронну мережу для впровадження функції розпізнавання об’єктів.

3.2 Поняття нейрону

Кожен нейрон реалізує деякі функції над вхідними значеннями. Якщо значення функції перевищує певний поріг, нейрон збуджується і формує вихідний сигнал для передавання його іншим нейронам. В роботі [7] зазначено: “мозок людини і її нервова система складаються з нейронів, з'єднаних нервовими волокнами. Між нейронами передаються електричні імпульси за допомогою нервових волокон. Всі дії, які відбуваються з живим організмом, все роздратування шкіри, очей, біль, процеси мислення - є взаємодія між нейронами”. Будова біологічного нейрона представлена на рисунку 3.1 [7].

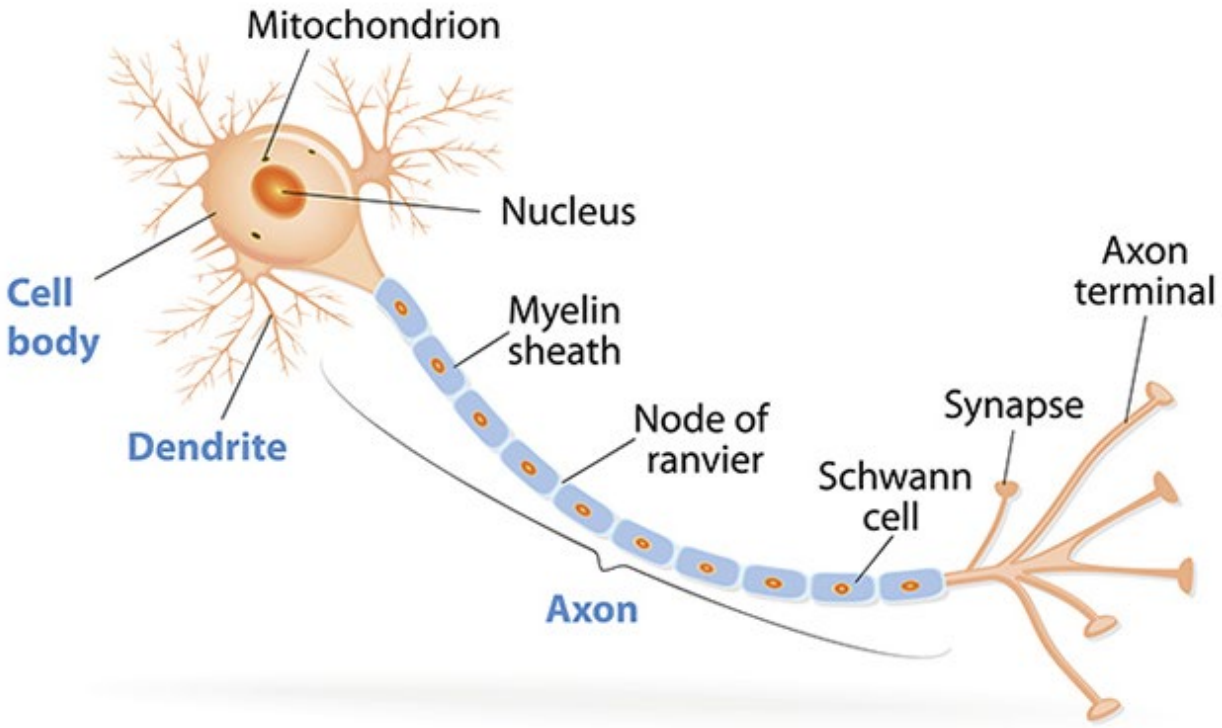


Рисунок 3.1 – Біологічний нейрон [9]

Дендрити - приймає імпульси нейрона; аксон - передає імпульс нейрона; синапси – утворення, що впливають на силу імпульсу, для контакту аксона і дендрита. Імпульс при проходженні синапсу змінює свою силу в певне число разів (вага синапсу). Коли до ядра нейрона з декількох дендритів надходять імпульси, то вони підсумовуються. Якщо у сумарного імпульсу перевищено поріг, то нейрон переходить в стан збудження, формує власний імпульс і надсилає його далі по аксону. Поведінка відповідного нейрона може змінюватися, так як ваги синапсів мають властивість змінюватися з часом [7].

Математична модель описаного процесу представлена наступним чином (рисунок 3.2):

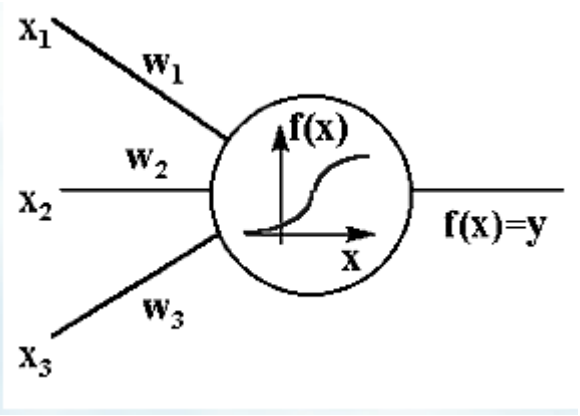


Рисунок 3.2 – Математична модель штучного нейрону [10]

Дана модель описує нейрон з трьома входами (дендритами), де синапси мають ваги w_1, w_2, w_3, до яких надходять вхідні сигнали x_1, x_2, x_3 відповідно. До нейрона надходять імпульси $x_1 w_1, x_2 w_2, x_3 w_3$ після проходження синапсів і дендритів [7].

Отриманий сумарний імпульс $x = x_1 w_1 + x_2 w_2 + x_3 w_3$ нейрон перетворює відповідно до функції передачі $f(x)$.

Таким чином, нейрон отримує набір чисел x_k (вектор). Далі нейрон, після деяких перетворень, видає певне число у на виході та передає його іншому нейрону.

Штучний нейрон це складова нейронної мережі який являє собою спрощену модель природного нейрона.

Під штучною нейронної мережею розуміється математична модель, а також її програмна і апаратна реалізація, побудована за принципом біологічних нейронних мереж - нервових клітин живого організму. Це поняття виникло при спробі змоделювати процеси, що протікають в мозку людини [7].

Штучна нейронна мережа являє собою систему штучних нейронів, з'єднаних і взаємодіючих між собою. Кожен нейрон мережі має справу з сигналами, які періодично надходять або передаються іншим нейронам [7].

Велика мережа здатна вирішувати найскладніші завдання в найкоротші терміни. Нейронна мережа в апаратно-програмних системах може використовуватись в якості арифметично-логічного пристрою, який входить в центральний процесор обладнання. Ця мережа в даній структурі пов’язує мережу та блок навчання. Пам’яттю в таких апаратно-програмних системах можна вважати набір вагів міжнейронних зв'язків, який формується в процесі навчання. У функціонуванні такої системи розрізняють два режими – робочий та режим навчання. Задача навчання полягає у такому налаштуванні коефіцієнтів між нейронних зв’язків, при якому забезпечується мінімізація помилки представлення за усією множиною, тобто коли кожному значенню вхідного образу відповідає потрібне значення вихідного образу. Аналогією таких систем можна вважати електронну систему відеоспостереження. В задачах розпізнавання об’єктів така мінімізація полягає у точному представленні на зображенні об’єктів відповідним їх типам класів. Аналіз роботи за цією функції і буде виконано у 4 розділі дисертації.

3.2 Класифікація нейронних мереж

Нейронні мережі розділяють по зв'язкам між нейронами, особливостям навчання мережі, особливостям моделі нейрона,. За структурою нейронні мережі можна розділити (рисунок 3.3) на неповнозв'язні і повнозв'язні, з регулярними і випадковими зв'язками, з симетричними і несиметричними зв'язками [8].

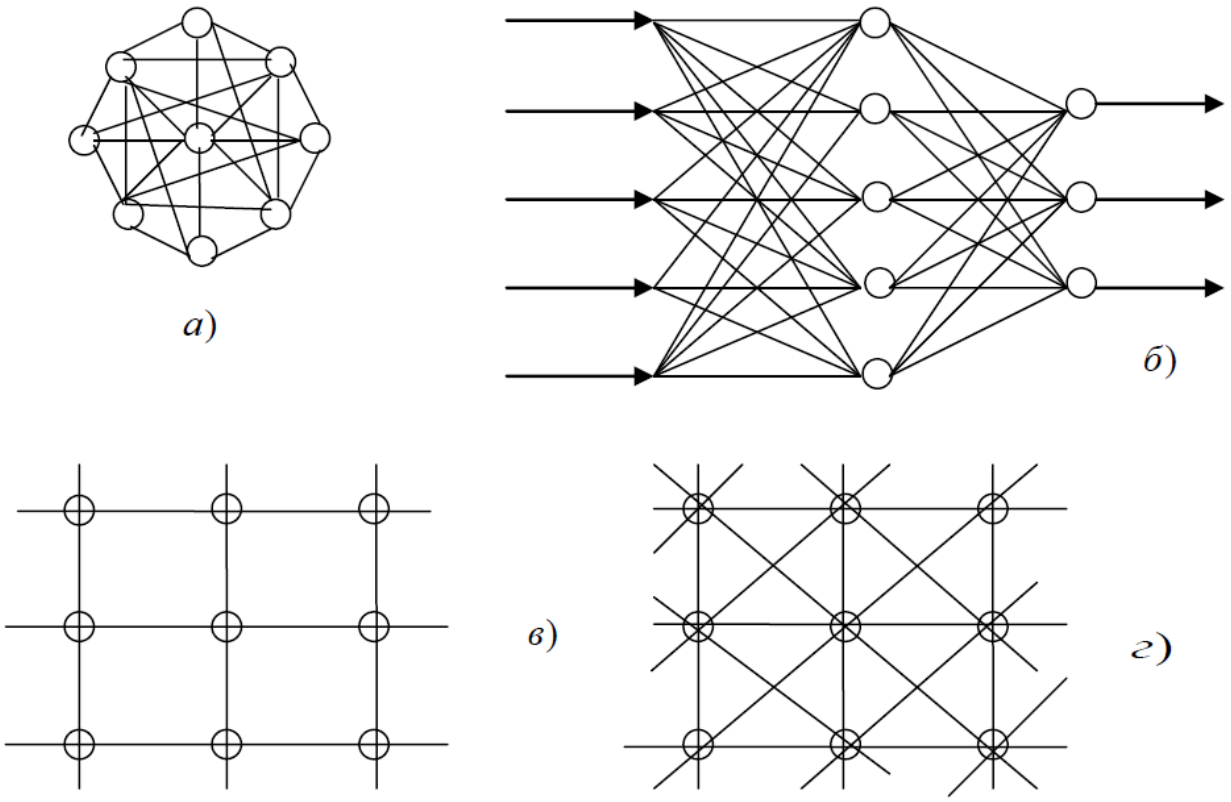


Рисунок 3.3 – Архітектури нейронних мереж: а) повнозв'язна мережа; б) багатошарова мережа з послідовними зв'язками; в, г) - слабкозв'язні мережі [8]

Неповнозв'язні нейронні мережі описуються неповнозв'язним орієнтованим графом, і зазвичай їх називають персептрони. Неповнозв'язні нейронні мережі поділяються на одношарові (найпростіші персептрони) і багатошарові, з прямими, перехресними і зворотними зв'язками. Класичним варіантом неповнозв'язних мереж є мережі прямого поширення (рисунок 3.4).

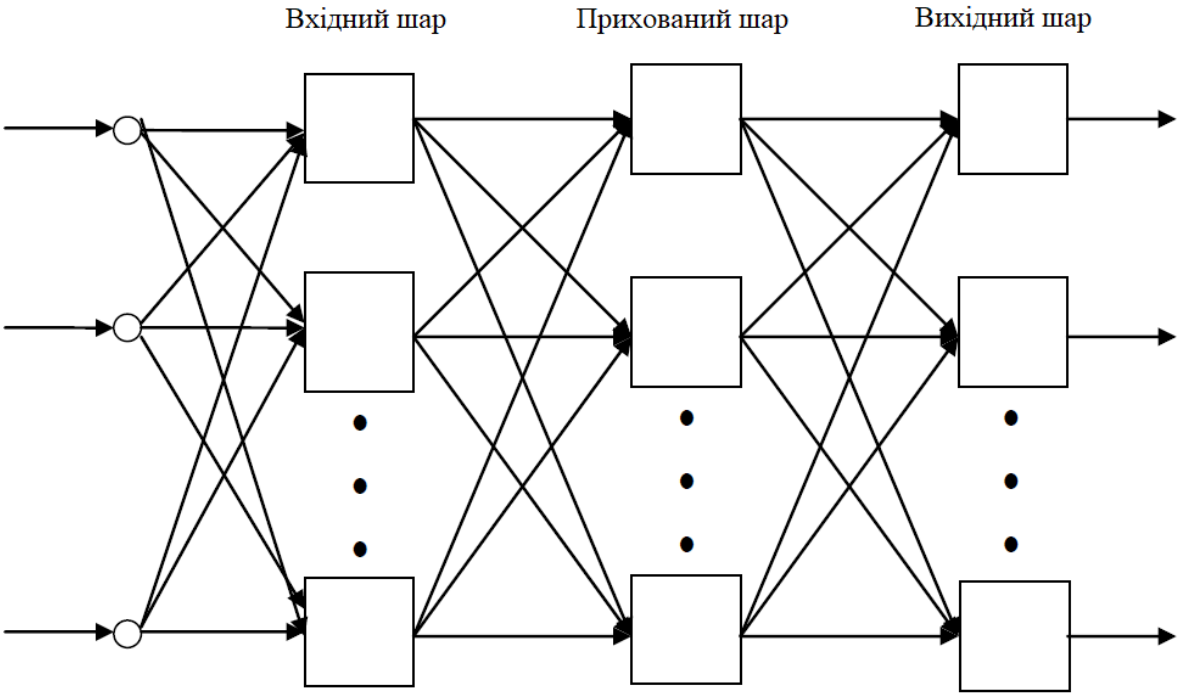


Рисунок 3.4 – Двошарова мережа прямого поширення [8]

У нейронних мережах з прямими зв'язками нейрони *j*-ого шару на входах можуть з'єднуватися тільки з нейронами *i*-х шарів, де $j > i$, тобто з нейронами нижчих шарів. У нейронних мережах з перехресними зв'язками допускаються зв'язку всередині одного шару, тобто вище наведене нерівність замінюється на $j >=$ і. В нейронних мережах із зворотними зв'язками використовуються зв'язки *j*-ого шару по входах з *i*-м при $j < i$. Крім того, за видом зв'язків розрізняють персептрони з регулярними і випадковими зв'язками [8].

3.3. Парадигми навчання нейронних мереж

Навчання нейронної мережі визначається наступною процедурою. Мережі послідовно надаються елементи навчальної вибірки і розраховується величина помилки, тобто відхилення результату роботи мережі від очікуваної реакції. Міжнейронні зв’язки корегуються пропорційно величині цього відхилення (корегуються насправді ваги зв'язків між нейронами). Далі надається наступний образ. Процедури навчання таким чином реалізуються

знаходження мінімуму деякого функціонала помилки методом градієнтного спуску. Для різних мереж можуть бути різні функціонали помилки і різна реалізація методу градієнтного спуску.

Таким чином, нейронну мережу можна навчити регулюючи значення вагів між елементами (нейронами). Зазвичай мережа навчається таким чином, щоб частковий вхід призводив до цільового виходу. Навчання полягає у порівнянні виходу та цілі, тобто до відповідності мережевого виходу та цілі. Зазвичай при такому підході виділяють багато навчальних пар вхід/ціль.

Організацію навчання нейронної мережі розрізняють на навчання з учителем та навчання без учителя.

1) Навчання з вчителем.

У загальному випадку, наявність вчителя можна розглядати як наявність знань про навколишнє середовище у вигляді пар вхід-вихід (питання, відповідь, оцінка відповіді, запам'ятовування правильної відповіді і корекція зв’язків до правильної відповіді). Це за умови, що зовнішнє середовище незнайома нейронної мережі. На початковому нейронну мережу подається вхідний навчальний вектор і результат який має бути сформований нейронною мережею на виході. Надання нейронній мережі результату передбачає вчинення нейронною мережею оптимальних дій по досягненню бажаної інформації на виході. Іншими словами, параметри мережі коригуються в залежності від вхідного навчального вектора і сигналу помилки. Сигналом помилки називають різницю між бажаним відгуком і дійсним вихідним значенням сигналу нейронної мережі. Імітуючи поведінку вчителя, штучна нейронна мережа виконує покрокове коригування всіх своїх параметрів. Цей процес повинен привести до отримання найбільшої схожості дій нейронної мережі з діями вчителя. Можна сказати, що процес навчання нейронної мережі - це процес передачі знань вчителя нейронній мережі. Після проведення тестування мережі і отримання задовільних результатів, необхідно відключити вчителя і дати можливість мережі працювати самостійно. Процес навчання можна схематично показати на рисунку 3.5. Набір питань, наприклад, це може бути алфавіт літер. Різниця між відповіддю мережі та істинним значенням називається вектором

помилки. Алгоритм зворотного розповсюдження помилки – набір формул, який дозволяє за вектором помилки розрахувати кореговані значення вагів мережі. Після багатократного надання питань ваги мережі стабілізуються, мережа надає привальні відповіді і вважається, що вона навчена. У програмних реалізаціях в процесі навчання величина помилки (сума квадратів помилок по усім виходам) поступово зменшується. І коли величина помилки досягає певного мінімуму, то вважається що мережа є навченою до даного банку прикладів.

Рисунок 3.5 – Алгоритм навчання мережі

Будь-яку взаємодію системи з учителем можна представити у вигляді конкретної точки на поверхні помилок. Ознакою підвищення продуктивності системи є зміщення точки помилки на поверхні помилок в бік мінімуму.

2) Навчання без вчителя.

Під час навчання без вчителя мережа самостійно здійснює налаштування вагових коефіцієнтів. При такому підході відсутні приклади, за допомогою яких мережа змогла би досягти необхідних для успішного результату дій.

Навчання без вчителя не передбачає втручання зовнішнього коректора в процес навчання і також може називатися самоорганізацією штучної нейронної мережі. Для мережі існує тільки допустимий рівень помилки, якого вона повинна досягти за допомогою оптимізації своїх вільних параметрів. Навчена мережа здатна виявляти статистичні закономірності вхідних даних і формувати

внутрішнє уявлення таким чином, щоб під час вступу нових вхідних даних відбувалося автоматичне формування нових класів, вільних параметрів мережі.

3.4 Алгоритми навчання нейронних мереж

Будь-який нейронна мереже має алгоритм навчання, тобто зведення помилки до мінімума. Розглянемо приклади алгоритмів навчання нейронних мереж.

3.4.1 Алгоритм навчання одного нейрону Хебба

У 1949 році Дональд Хебб, досліджуючи механізми функціонування центральної нервової системи, припустив, що навчання нервових клітин мозку відбувається шляхом посилення зв'язків між тими нейронами, які синхронно збуджуються. Іншими словами, між одночасно активованими нейронами мережі пороги синаптичних зв'язків знижуються. В результаті утворюються «нейронні ансамблі», які все швидше активуються при кожному черговому повторенні входу. Це спостереження лягло в основу одного з перших правил навчання нейронної мережі, відомого як правило Хебба. І хоча в біологічних системах цей механізм не завжди виконується, при навчанні штучних нейронних мереж він виявився дуже ефективним [3].

Алгоритм Хебба описується наступними кроками:

1) Ініціюються початкові значення вагового вектора - вектора синапсів: $w^{(0)}$.

2) Для всіх пар $(x_i,\ y_i)$, виконується корекція вагового вектора за формулою $w^{(k+1)} = w^{(k)} + x_i\ y_i$.

3) Перевіряється умова зупинки для знайденого вагового вектора w, а саме, для кожної пари (x_i, y_i), обчислюються значення s_i = sgn((*w,* x_i)). Якщо s_i = y_i для всіх *i* = 1, ..., *N*, то алгоритм припиняє свою роботу, в іншому випадку - перехід до пункту 2.

3.4.2 Персептроний метод навчання

В алгоритмі навчання персептрона для уніфікованих векторів навчальної вибірки $E' = \{x_1,..., x_N\}$ (Тобто. $x'_i = x_i$ якщо $x_i \in \overline{w_1}$ та $x'_i = -x_i$ якщо $x_i \in \overline{w_2}$) мінімізується функціонал помилки $F(w) = \sum_{x'_i \in E(w)}(w, x')$, де E (w) - підмножина векторів навчальної вибірки, які класифікуються неправильно для даного вектора w, тобто $(w, x') \leq 0$. Тоді $\partial F/\partial w_i = -\sum_{x'_i \in E(w)} x'_i$ і $grad(F) = -\sum_{x'_i \in E(w)} x'_i = -\sum_{x'_i \in E'} x'\eta(-(w, x'))$, де η – функція Хевісайда. Таким чином, величина корекції пропорційна векторної сумі неправильно класифікованих векторів. На практиці при навчанні нейронної мережі корекцію здійснюють після виявлення чергового неправильно класифікованого вектора і корегують на величину цього вектора. Така варіація методу градієнтного спуску робить його частково випадковою, що покращує збіжність алгоритму навчання [3].

3.4.3 Навчання багатошарової нейронної мережі методом зворотного поширення помилки

Метод зворотного поширення помилки був запропонований Румельхартом, Хінтон і Вільямсом в 1986 році і призначений для навчання багатошарової мережі. Припустимо, що є мережа, що складається з M шарів, навчальна вибірка $E = \{x_1,..., x_N\}$. а також сукупність правильних відгуків мережі $y_1,..., y_N$, де y_i - вектор значень, який повинен бути отриманий на останньому шарі при надходженні на перший шар мережі вектора x_i. (рисунок 3.6) [3].

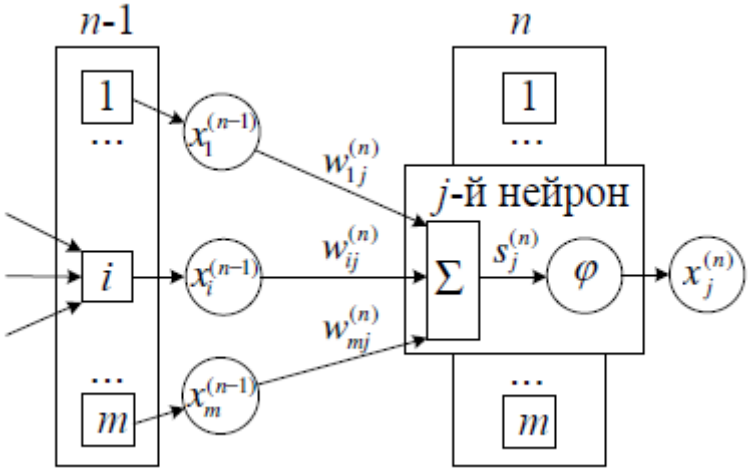


Рисунок 3.6 – Багатошарова нейронна мережа [3]

Нехай між i - м n - 1-го шару та j -м нейроном n -го шару існує випадкові зв'язки $w_{ij}^{(n)}, x^{(n)}$ - вектор на виході n -го шара, $n =$ 1,..., M ($x^{(M)} -$ вектор на виході останнього шару і всієї мережі в цілому). Мета навчання M – шарової мережі полягає в знаходженні такої кількості вагових матриць $W^{(n)} = w_{ij}^{(n)}, n =$ 2,..., M , щоб середньоквадратична помилка неправильної класифікації була мінімальною, тобто

$$F\left(W^2, \dots, W^{(M)}\right) = \sum_{x \in E} \left\| x^{(M)} - y \right\|^2 = \sum_{x \in E} \sum_{j} (x_j^{(M)} - y_j)^2 \to min.$$

Мінімізацію цього функціоналу помилки здійснюють методом градієнтного спуску, шляхом корекції ваг кожного шару. Перетворення вхідного сигналу в j-му нейроні n -го шару здійснюється за формулою $x_j^{(n)} = \varphi\left(s_j^{(n)}\right)$, де $s_j^{(n)} = \left(w_j^{(n)}, x\right)$, φ - обрана функція активації. На n-му шарі корекція ваги $w_{ij}^{(n)}$ змінюється на величину.

$$\Delta w_{ij}^{(n)} = -h \frac{\partial \mathrm{F}}{\partial w_{ij}^{(n)}} = -h \frac{\partial \mathrm{F}}{\partial x_j^{(n)}} \frac{dx_j^{(n)}}{ds_j^{(n)}} \frac{\partial s_j^{(n)}}{\partial w_{ij}^{(n)}} = -h \delta_j^{(n)} x_i^{(n)}, \qquad (3.1)$$

де $\delta_j^{(n)} = \frac{\partial \mathrm{F}}{\partial x_j^{(n)}} \frac{dx_j^{(n)}}{ds_j^{(n)}}$. Величина $\delta_j^{(n)}$ уособлює величину помилки на n-му шарі в j -му нейроне. має сенс помилки на n-му шарі в j -му нейроне.

Розглянемо, як змінюється величина помилки $\delta_j^{(n)}$ при переході з n - го шару на n + 1-ий шар. Маємо

$$\delta_j^{(n)} = \frac{\partial \mathrm{F}}{\partial x_j^{(n)}} \frac{dx_j^{(n)}}{ds_j^{(n)}} = \left(\sum_k \frac{\partial \mathrm{F}}{\partial x_k^{(n+1)}} \frac{dx_k^{(n+1)}}{ds_k^{(n+1)}} \frac{\partial s_k^{(n+1)}}{\partial x_j^{(n+1)}} \right) \frac{\partial x_j^{(n)}}{\partial s_j^{(n)}}$$

$$= \left(\sum_k \delta_k^{(n+1)} w_{jk}^{(n+1)} \right) \frac{\partial x_j^{(n)}}{\partial s_j^{(n)}},$$

так як $\frac{\partial s_k^{(n+1)}}{\partial x_j^{(n+1)}} = w_{jk}^{(n+1)}$. Таким чином, маємо ітераційну формулу для обчислення коефіцієнтів $\delta_j^{(n)}$:

$$\delta_j^{(n)} = \left(\sum_k \delta_j^{(n+1)} w_{jk}^{(n+1)} \right) \frac{\partial x_j^{(n)}}{\partial s_j^{(n)}}, n = M-1, M-2, \dots, 1. \quad (3.2)$$

На останньому шарі маємо

$$\delta_j^{(M)} = \left(x_j^{(n)} - y_j \right) \frac{\partial x_j^{(M)}}{\partial s_j^{(M)}}. \quad (3.3)$$

Зауважимо, що похідні $dx_j^{(n)}/ds_j^{(n)} = \varphi'\left(s_j^{(n)}\right)-$ це похідні функцій активації φ. Обчислення величин $\delta_j^{(n)}$ за формулами (3.2) і (3.3) можна інтерпретувати як зворотне поширення помилки від останнього шару до першого [3].

Алгоритм навчання нейронної мережі методом зворотного поширення помилки описується наступними кроками:

1. Ініціюються початкові значення вагових матриць: $W^2, \dots, W^M$.

2. На вхід першого шару мережі подається черговий навчальний вектор x. У звичайному режимі функціонування обчислюються всі значення $s_j^{(n)} = \sum_i w_i^{(n-1)} w_{ij}^{(n)}, n = 1, \dots, M, j = 1, \dots, N$.

3. За формулою (3.2) (або (3.3) для останнього шару) обчислюються значення помилок $\delta_j^{(n)}$.

4. За формулою (3.1) здійснюється корекція ваги на даному шарі.

5. Аналогічно, виконуючи зворотне поширення помилки за формулою (3.2), коригуємо ваги на інших шарах.

6. Перевіряємо умову зупинки - стабілізацію критерію мінімізації F (тобто $F^{k+1}= F^{k}$): якщо F стабілізувався, то алгоритм завершує роботу, в іншому випадку - перехід до пункту 2.

3.5 Переваги нейронних мереж в системах розпізнавання об’єктів

Нейронна мережа для систем розпізнавання об’єктів дозволяє вирішити задачу приналежності вхідного образу об’єкту, який можна визначити через вектор ознак, до одного або декілька класів, які попередньо визначені. В робо техніці, як однієї з напрямів сучасної електроніці, одним з технічних додатків є розпізнавання об’єктів з відеоінформації чи зображення, які отримано від системи технічного зору. Модель на основі нейронної мережі має наступні основні властивості:

- однорідність (елементи нейронної мережі однакові і прості, функції мережі визначаються структурою зв’язків);
- надійність (забезпечується завдяки надмірній кількості зв’язків між елементами);
- голографічність (тобто зберігання властивостей навіть у випадку руйнування частини моделі);
- паралельність дій (швидкість аналізу забезпечується величезною кількістю паралельно функціонуючих нейронів та між нейронних зв’язків);
- дані які аналізуються надалі можуть бути використані для навчання мережі (корегування порігу спрацювання на приналежність об’єкту до класу);
- для накопичування знань мережею використовуються зв’язки між нейронами (синаптичні ваги);

Висновки до розділу 3. Здатність нейронної мережі вчитися - ось що робить її настільки корисною в області штучного інтелекту. Завдяки нейронним мережам стало можливо будувати складні системи для розпізнавання образів. Для розпізнавання образів використовується парадигма навчання з учителем. Тобто на вхід нейронної мережі подається зображення номерного знаку та результат який має вийти.

Для навчання нейронної мережі існує багато алгоритмів і кожен з них показує різну ефективність в різних задачах. Нейронні мережі навчені за допомогою різних алгоритмів ведуть себе по різному в одних і тих самих задачах. Тому доцільно дослідити нейронні мережі для розпізнавання зважаючи на їх переваги (швидкість, точність, різноманіття об'єктів, здатність до навчання).

4 ПРАКТИЧНА ЧАСТИНА ДОСЛІДЖЕННЯ

У практичній частині дослідження дисертації розглянемо три найбільш поширені у світі нейронні мережі RetinaNet, YOLO v.3, TinyYOLO, які використовуються в системах розпізнавання образів і можуть бути обрані в якості основних електронних систем машинного читання у приладах електронного відеоспостереження при створенні систем контролю за дорожнім рухом. Зазначимо, що в якості чистоти проведення експерименту були налаштовані нейронній системи при однакових вихідних умовах. Тестування систем, перевірка можливості їх глибокого навчання відбувалось за умови наявності однакового візуального матеріалу. Вибір цього матеріалу обумовлений завданням на дисертацію і спрямований насамперед на виявлення технічних особливостей, які погіршують або поліпшують роботу нейронної мережі в цілому і які надалі слід обов'язково враховувати при проектуванні прототипів систем розпізнавання образів з елементами навчання.

4.1 Вихідні дані до експерименту

В якості вихідних даних для перевірки трьох обраних нейронних мереж було обрано два види візуального контенту – набір статичних зображень у кількості 11 одиниць, які характеризують різноманітні дорожні ситуації та секвенція кадрів, з кінофільму, яка визначає різноманітні динамічні ситуації, які впливають на якість роботи нейронної мережі, як складової системи розпізнавання об'єктів. Зокрема, серед таких ситуації можна зазначити, швидкість руху автомобілів, віддаленість об'єкту від об'єктиву кінокамери, умови освітлення сцени, сторонні об'єкти сцени, тощо. В якості частоти зйомки обрана стандартна частота – 24 кадри за секунду. Загальна кількість кадрів секвенції – 84 кадри, формат зображення HD_720 (1280 пікселів×720 пікселів), кольорова система – sRGB; графічний формат зображень – JPEG.

Вибір статичних зображень для проведення експерименту полягав насамперед з міркувань того, щоб визначити як нейронні мережі працюють, чи здатній вони до навчання і який метод чи спосіб використовується в основі їх функціонування. Крім цього, зазначимо, що завантаження як статичних зображень, так і динамічної секвенції відбувалось на одній і тій самій графічній станції за однакових умов (ноутбук Dell G3 15, 6 ядерний процесор Intel Core i7 8 покоління, 32 ГБ оперативної пам'яті, дискретна відео карта з 4 ГБ відео пам'яті). В графічній станції є підтримка технології CUDA. Додатково, перевірялась швидкість зчитування та розпізнавання об’єктів на зображенні за умови однакового алгоритму та однакових підключених бібліотек. Скрипт і алгоритм його створення на мові Python в середовищі PyCharm також буде наведено у практичній частині дослідження.

Алгоритм, за яким буде проведено експеримент доцільно поділити на послідовність наступних дій: вибір з зображення з характерними ознаками; завантаження бібліотеки об’єктів нейронної мережі, запуск мережі на опрацювання зображення, проведення оцінювання точності спрацювання мережі. При цьому, для наочності дослідження з кожного зображення серед отриманих результатів для трьох нейронних мереж буде обрано три однакові об’єкти (для секвенції кадрів - два). Отримані результати буде зведено у відповідні графічні ілюстрації у формі стовпчиків, при цьому колір кожного стовпчика для зручності буде визначати певний тип нейронної мережі. Перед проведенням експерименту спочатку наведемо певні характерні особливості зазначених нейронних мереж, які визначають самі розробники з метою їх перевірки безпосередньо на практиці в рамках дослідження. Далі наведемо алгоритм створення моделі дослідження і після цього перейдемо до аналізу отриманих результатів.

4.1.1 Нейронна мережа RetinaNet

Архітектура нейронної мережі складається з 4 ланок, кожна з яких виконує певні функції в загальній процедурі розпізнавання об’єктів на зображенні. На рисунку 4.1 наведена архітектура цієї мережі.

Перша ланка мережі носить назву “backbone” і це є основним елементом мережі, який використовується для пошуку та фіксації ознак з зображення, яке є предметом аналізу. Ця частина мережі є зі змінною структурою і в її основі можуть входити інші класифікаційні мережі, наприклад ResNet, EfficientNet [11]. Перший тип класифікаційної мережі якраз і наведено на рисунку 4.1.

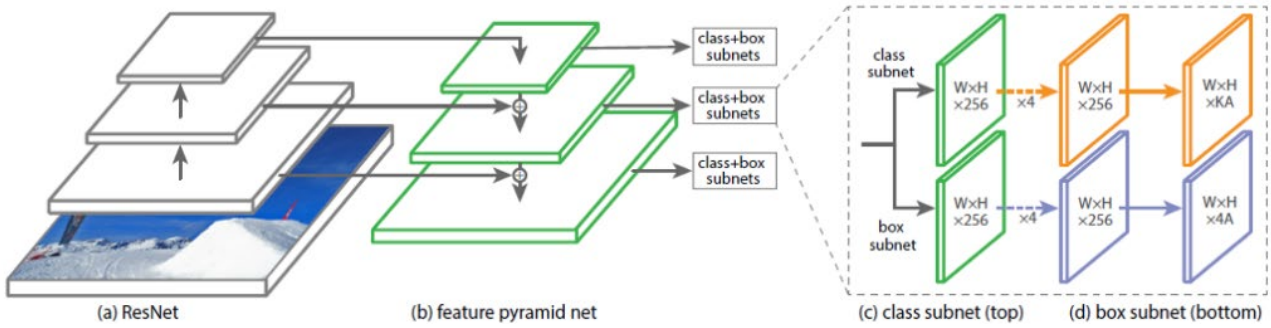


Рисунок 4.1 – Архітектура мережі RetinaNet (варіант) [12,13]

На першій ланці на основі варіативної мережі проводиться аналіз зображення з виділенням важливих ознак і ця отримана інформація далі буде використана в мережі як для навчання, так і для створення класифікаційної таблиці. Тому вибір змінної структури необхідно обирати так, щоб модель мала велику точність виявлення та класифікації ознак в палітрі зображення.

Друга ланка архітектури отримала назву Feature Pyramid Net (FPN) [12,13] і за структурою являє собою карту характеристик різного масштабу на деяких рівнях у формі піраміди. Ця ланка у формі згортувальної підмережі використовується для об’єднання переваг карт ознак на нижніх та верхніх рівнях мережі. Нижній рівень (шар) має високу роздільну здатність і низьку семантичну здатність. Верхній рівень (шар) навпаки – низька роздільна

здатність, але висока семантична здатність (значення). На рисунку 4.2 ці особливості карт ланки наведено. Фактично на цій ланці нейронної мережі маємо ієрархічну піраміду – послідовність рівнів (шарів) з поступовим зменшенням роздільної здатності зображення, яке аналізується. При цьому ознаки які проходять крізь аналіз з різною роздільною здатністю мережа отримує з першої ланки. Верхній шар ланки називається bottom-up pathway, нижній шар має назву top-down pathway і ще є бічні з'єднання (lateral connections). В нашому випадку на цій ланці аналіз зображення проходить у два шляхи з верхньої області до нижньої і потім навпаки з нижньої до верхньої (рис.4.3). При цьому на верхньому рівні можуть бути втрачені деталі зображення, наприклад через високій рівень шумів оригінального зображення. На нижньому рівні, що характерно для отримання високої роздільної здатності карти ознак збільшуються вдвічі на основі методу "найближчого сусіда" за напрямком донизу (рис.4.4).

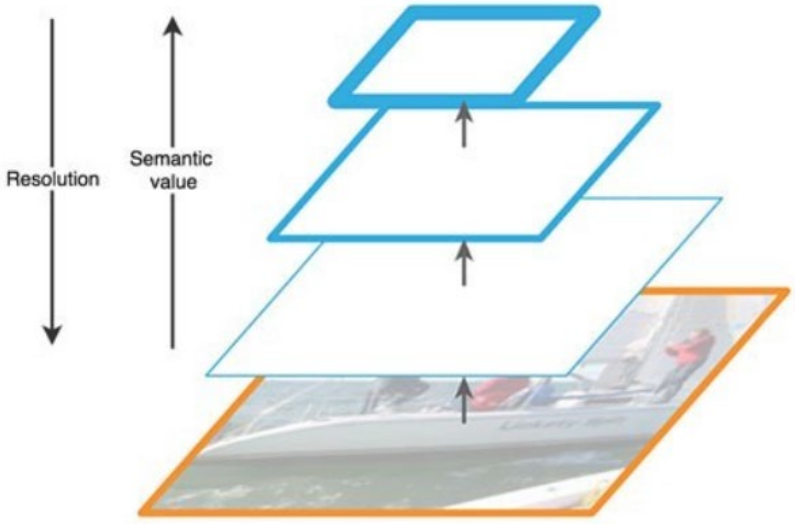


Рисунок 4.2 – Різні рівні мережі FPN

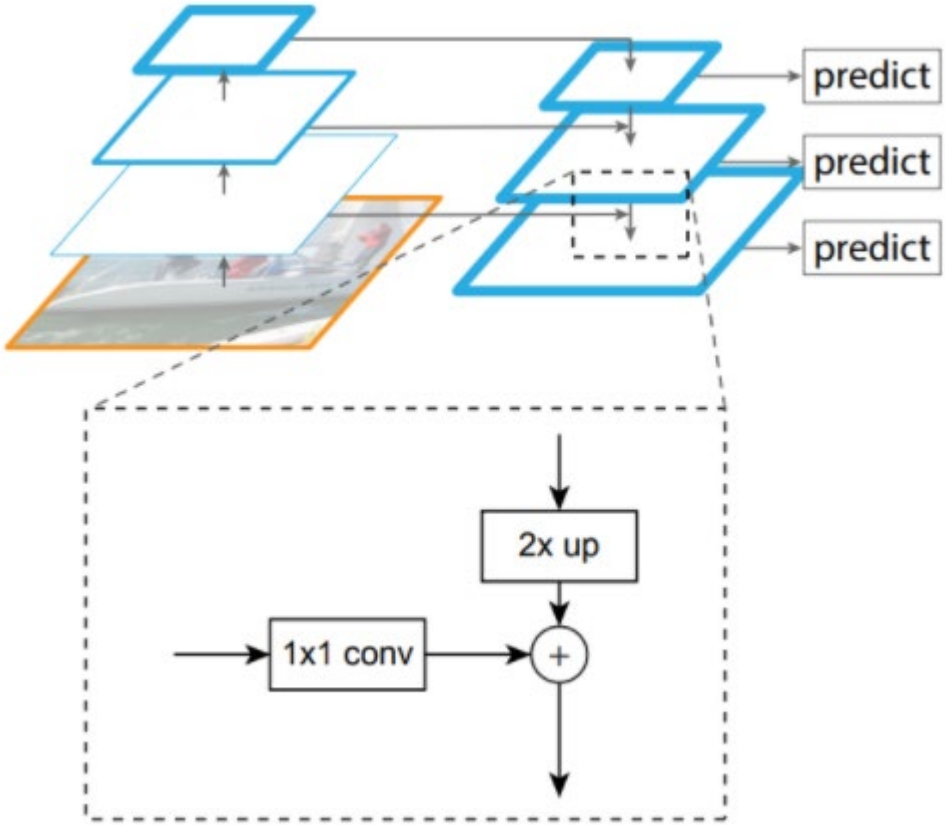


Рисунок 4.3 – Алгоритм аналізу ознак

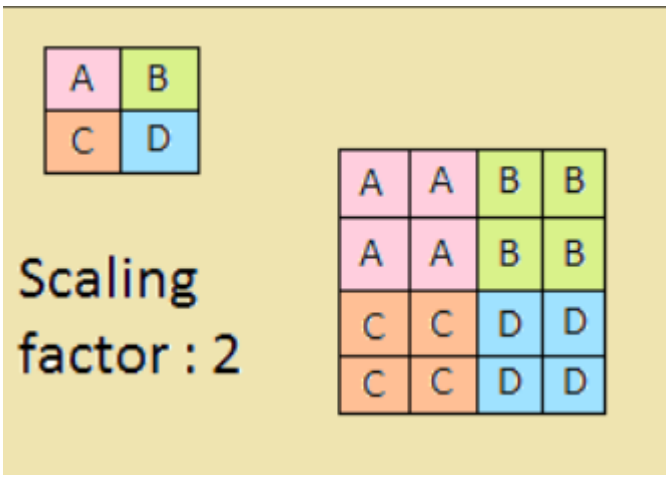


Рисунок 4.4 – Зображення збільшення роздільної здатності зображення

З рисунку 4.3 можна підкреслити, що через наявність бічних з’єднань карти ознак відповідних пірамід по-елементно додаються, причому карти з верхнього рівня проходять ще і через згортку 1×1 conv. Таким чином, такий підхід роботи ланки на основі двох шляхів проходження дозволяє суміщати семантично важливу інформацію, яка отримана наприкінці першої піраміди, і

більш детальну інформацію яка отримана раніше. Отримана інформація переходить на третю ланку RetinaNet, яка складається з двох підмереж, які називаються класифікаційна (Classification Subnet) та регресійна відповідно (Regression Subnet) (рис.4.1). Класифікаційна підмережа з FPN визначає інформацію про класи об'єктів розпізнавання, розв'язуючи тим самим задачу класифікації. Регресійна підмережа у свою чергу вилучає з карти ознак інформацію про розташування об'єктів, їх координати на зображенні. З рисунку 4.1 можна відмітити, що ці дві підмережі відрізняються лише останнім шаром. Так на 5 шарі кількість карт ознак змінюється – регресійна модель має $4A$ карт ознак, тоді як класифікаційна модель відзначається лише KA кількістю карт ознак, де A – кількість якірних рамок, K – кількість класів об'єктів за класифікацією. У шостому шарі (рис.4.5) кожна карта ознак перетворюється у набір векторів. Якірна рамка (anchor box) – параметр нейронних мереж, у формі детектора прямокутника відносного якого працює мережа. На рисунку 4.6 наведено для карти ознак розміром 3×3, де кожна з 9 комірок має відповідно 9 якірних рамок. При цьому кожна з цих рамок має різний розмір та різне співвідношення сторін.

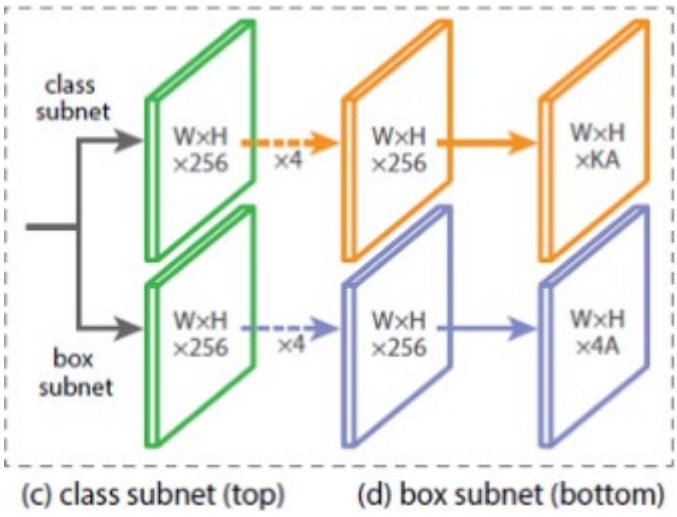


Рисунок 4.5 – Класифікаційна (class subnet) та регресійна (box subnet) підмережі (шість шарів) [12,13]

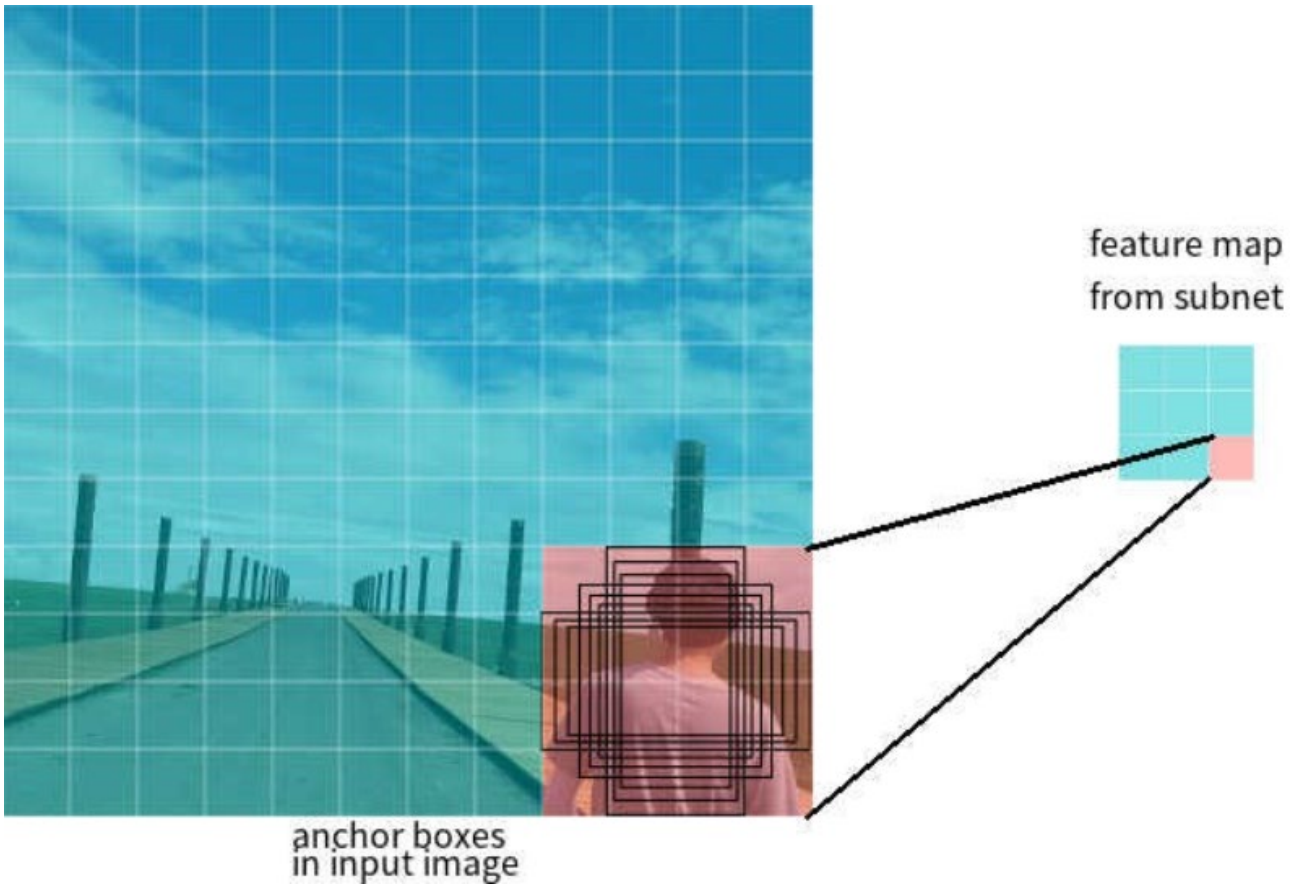


Рисунок 4.6 – Комірка карти ознак з нанесеними якірними рамками [12]

Під час навчання нейронної мережі RetinaNet кожній цільовій рамці підбирається відповідна якірна рамка з набору. Якщо показник співпадіння (параметр IoU, див п.4.1.2) має значення більше за 50 відсотків то якірна рамка вважається цільовою, а якщо менше цього значення, то вважається що рамка вважається фоном і надалі не використовується для навчання мережі. Класифікаційна підмережа навчається відносно призначення (клас об’єкту або фон), регресійна відносно координат якірної рамки. Помилка роботи RetinaNet складається з двох значень – помилка регресії (локалізації) і помилка класифікації [14]. Значення помилки регресії визначається в мережі через 4 числа – Pos_{ix}, Pos_{iy}, Pos_{iw}, Pos_{ih}. Перші два числа визначають прогнозовану різницю між координатами центрів для якірної та цільової рамок, тоді як третя та четверта координата визначають прогнозовані різницю між висотою та шириною двох рамок. В результаті, для кожної цільової рамки розраховується відповідна різниця з якірною. Помилка класифікації розраховується через спеціальну функцію, в яку входить ймовірність передбачення класу, цільове

призначення класу, кількість класів та коефіцієнт зміщення. Ця функція є функцією крос-ентропії (функція ентропійної втрати). Під час навчання мережі більша частина об’єктів вважаються класифікатором фоном, який виведено в окремий клас [15]. І теоретично, мережа може краще визначати фон, аніж потрібні об’єкти розпізнавання.

За свідченням розробників цієї нейронної мережі вона здатна в умовах невеликих об’єктів (ширина декілька пікселів), з частковим закриттям об’єктів у тіні забезпечувати високу середню точність розпізнавання, яка складає більше ніж 78 відсотків. Перевіримо це твердження нижче в рамках практичного експерименту.

4.1.2 Нейронна мережа YOLO

Нейронна мережа YOLO (You Only Look Once) виходячи зі своєї назви відрізняється від інших нейронних мереж тим, що в процесі аналізу зображення ця мережа використовується один раз відразу до аналізу усієї площини зображення. Тобто, нейронна мережа поділяє зображення на своєрідну сітку з однаковими комірками і прогнозує своєрідні області bounding box тим, що саме в цій області з певною ймовірністю є шуканий об’єкт для кожної ділянки зображення.

Перевагою такого підходу є те, що при аналізі всього зображення мережа враховує відразу повний інформаційний контекст при детектуванні та розпізнаванні. Тобто, на відміну від мережі RetinaNet (див.п.4.1.1) нема двобічного проходження аналізу зображення, то дана мережа повинна працювати значно швидше. Так, наприклад, за словами розробника Джозефа Редмона стверджується, що базова модель мережі обробляє зображення в режимі реального часу зі швидкістю 45 кадрів за секунду часу. Водночас, відомо, що ця мережа важко локалізує об’єкти з високою точністю, але це без умови навчання. Ці особливості перевіримо нижче у дослідженні шляхом

запуску часового лічильника при аналізі одного й того самого зображення різними нейронними мережами.

Іншими словами, кожна комірка відповідає за прогнозування певного класу об’єкту і визначається для неї показник впевненості (confidence). Цей показник математично є ймовірністю того, що в даній комірці міститься шуканий об’єкт. В результаті такого аналізу отримуємо карту усіх об’єктів і набір рамок з ймовірностями через показник впевненості. На рисунку 4.7 наведено описаний вище алгоритм розпізнавання об’єктів. Крім цього, кожна комірка відповідає за прогнозування ймовірності конкретних класів об’єктів зображення, для пошуку яких використовується нейронна мережа [17].

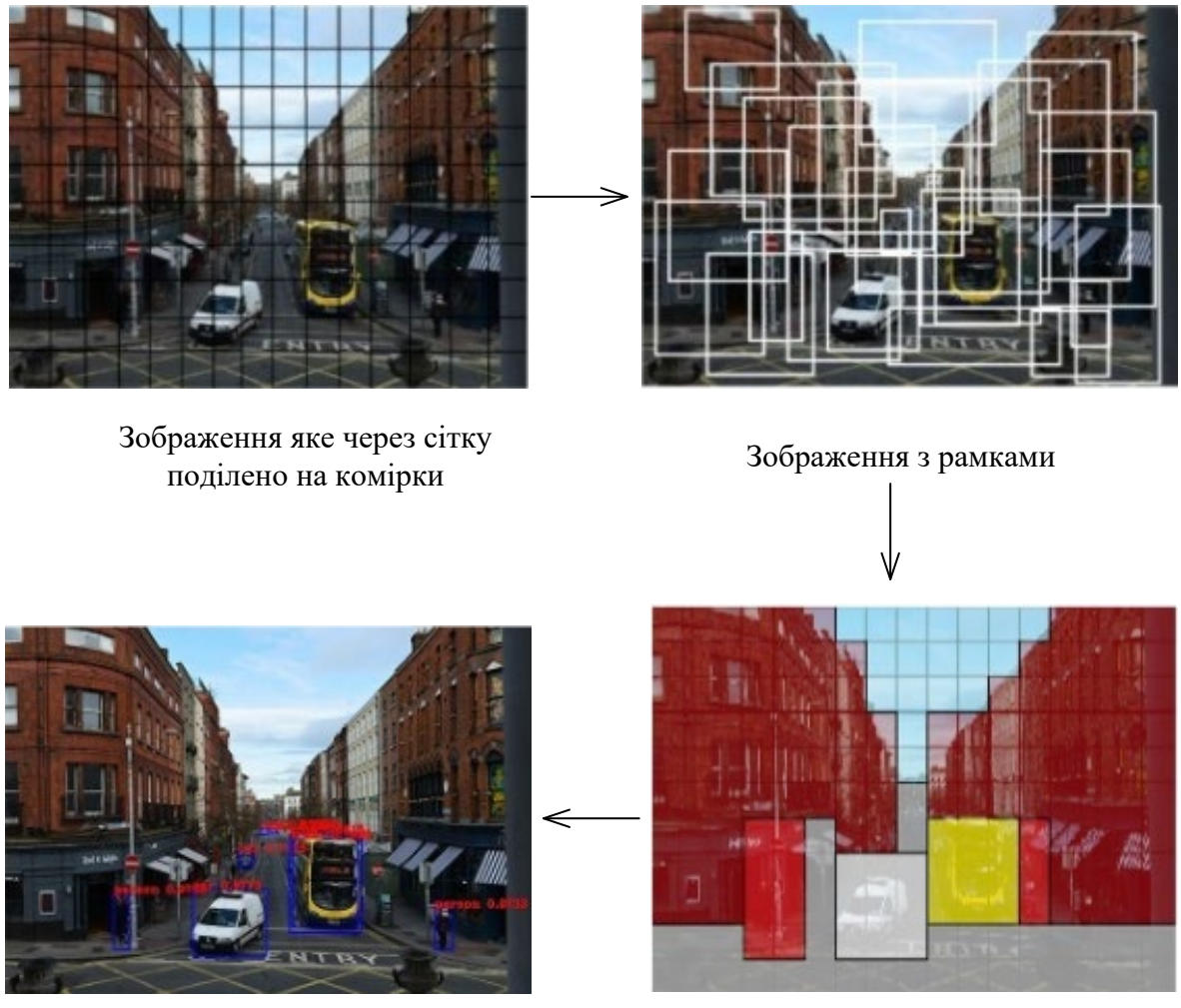


Рисунок 4.7 – Алгоритм роботи мережі YOLO

Мережа YOLO версії 3 складається з 106 шарів (рівнів згортки) і за твердженням розробників з високою точністю розпізнає невеликі об’єкти зображення. Головна особливість мережі у тому, що на виході мережі є 3 шари, кожен з яких розрахований на виявлення об’єктів різного розміру. На рисунку 4.8 наведено схематично принцип роботи нейронної мережі.

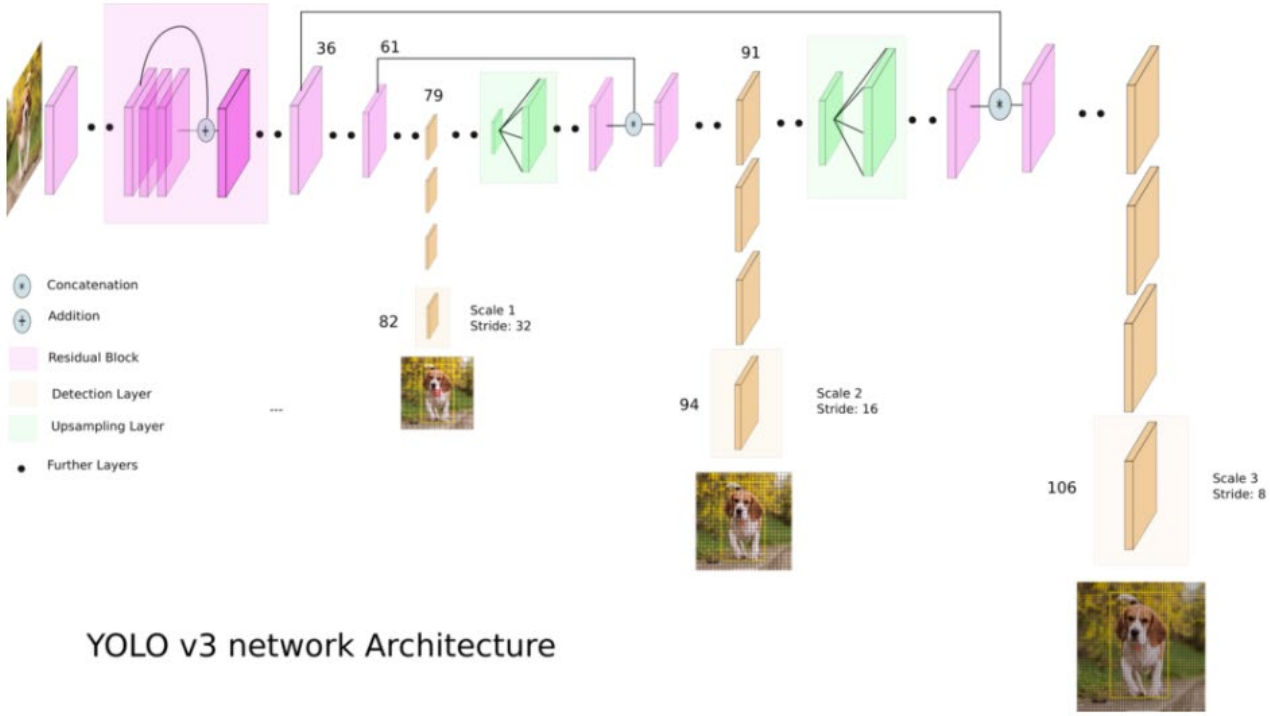


Рисунок 4.8 – Принцип роботи мережі YOLO v.3 [17]

В мережі YOLO розпізнавання об’єктів реалізовано як єдина задача регресії до роздільних обмежувальних рамок, з якими пов’язані ймовірності приналежності до різних класів об’єктів на зображенні. Тобто, прослідковується безпосередній зв’язок пікселів зображення і координат рамок та ймовірностей прогнозування класу об’єкту. Для прогнозування рамок на вміст об’єктів використовуються якірні рамки (anchor boxes). Їх основне призначення у визначенні двох рамок які визначають форму пошуку (рис.4.9).

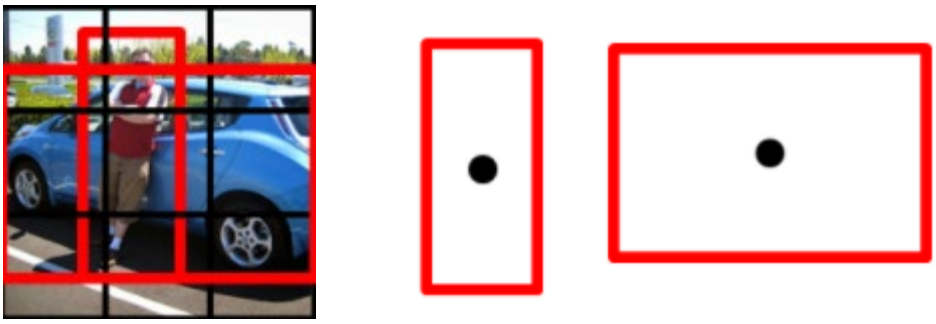


Рисунок 4.9 – Принцип прогнозування

Таким чином, на площині зображення створюється сітка з комірками і для кожної комірки маємо в результаті роботи мережі 4 координати - Pos_{ix}, Pos_{iy}, Pos_{iw}, Pos_{ih}, показник впевненості (є чи немає об’єкт в області комірки) і деяку кількість ймовірностей класів. В загальному випадку якірних рамок може бути і більше двох (до п’яти за алгоритмом). В алгоритмі мережі YOHO також як і в моделі RetinaNet використовується поняття “перетин відносно об’єднання” (intersection over union, IoU). Це відношення площі перетину прямокутників до площі їх об’єднання (на рисунку 4.10 це відношення розмірів областей жовтого кольору до синього кольору).

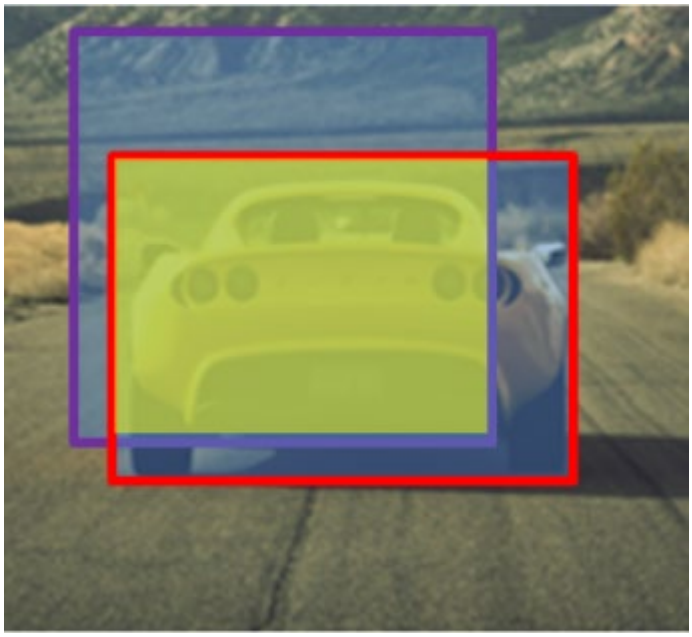


Рисунок 4.10 – Визначення параметру IoU

До того ж, в результаті аналізу всієї площини зображення можуть виникнути дублікати результатів. Для того щоб їх позбутись, в мережі YOHO передбачено так званий алгоритм “усунення не-максимумів”. За цим алгоритмом знаходиться рамка з максимальним значенням показника впевненості, і інші найближчі рамки, які розташовані поруч з максимальним перетином відносно об’єднання з першою рамкою просто будуть проігноровані.

4.1.3 Нейронна мережа TinyYOLO

Це спрощена версія розглянутої вище архітектури YOLO і вона складається з меншого числа на виході шарів. Для цієї реалізації їх лише два, на відміну від архітектури мережі YOLO (див. рис.4.8). Через це, як свідчать численні експерименті ця мережа гірше проводить розпізнавання дрібних деталей зображення, але при цьому вона функціонує швидше аніж її повна версія. Наприклад, при аналізі відео з мобільного телефону iPhone XS швидкість при роботі мережі TinyYOLO складала в середньому 30-32 кадрів за секунду. Тоді як для мережі YOLO це значення було в межах 23-25 кадрів за секунду. Крім цього, логічно зробити висновок, що дана версія буде мати нижчу точність розпізнавання об’єктів. Перевіримо ці особливості роботи мережі далі під час експерименту з аналізу статичних зображень об’єктів з автомобільних доріг.

4.2 Алгоритм створення програми для розпізнавання зображень об’єктів на основі нейронної мережі

4.2.1 Підготовчий етап дослідження

Для створення програми, яка могла б провести тестування нейронної мережі використаємо бібліотеку ImageIA – інструмент розпізнавання різних об’єктів на рухомому зображенні чи фотографії на основі мови програмування Python. Алгоритм створення програми можна поділити на наступні етапи.

1. Для завантаження бібліотеки встановлюємо програму Python з пакетним менеджером pip для отримання можливості встановлювати додаткові пакети в середовище розробки.

2. Відкриваємо IDE текстовий редактор PyCharm і в ньому створюємо новий проект з назвою image detection (рис.4.11).

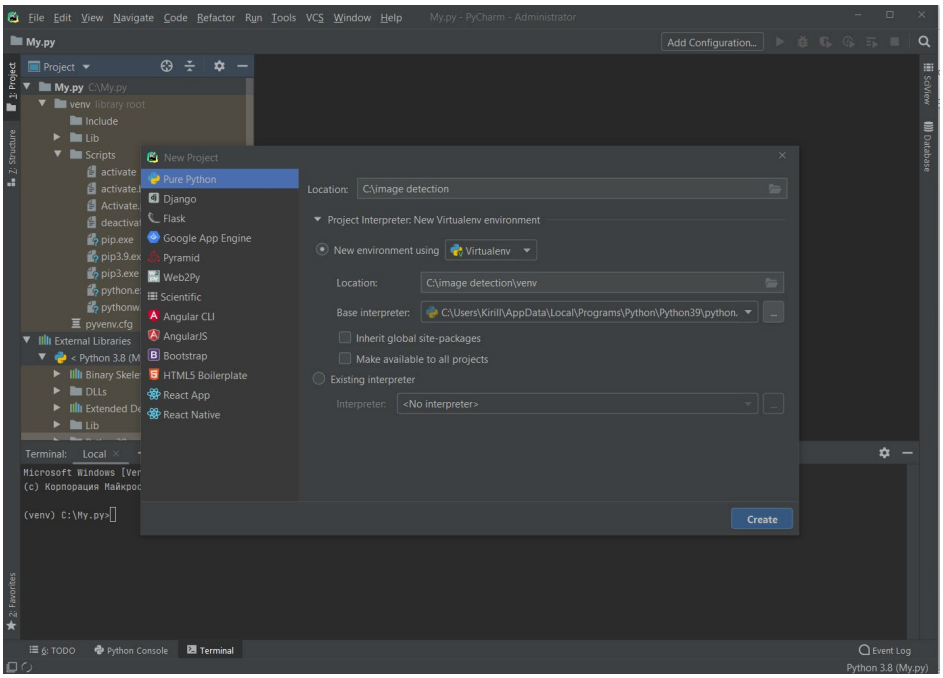


Рисунок 4.11 – Налаштування інтерфейсу для тестування нейронної мережі системи розпізнавання об’єктів

В результаті отримаємо робоче поле. Зверху через рядок venv створюємо новий файл – Diplom.py.

3. Після створення проекту та файлу отримуємо поле для створення програми з розпізнавання елементів зображення (рис.4.12).

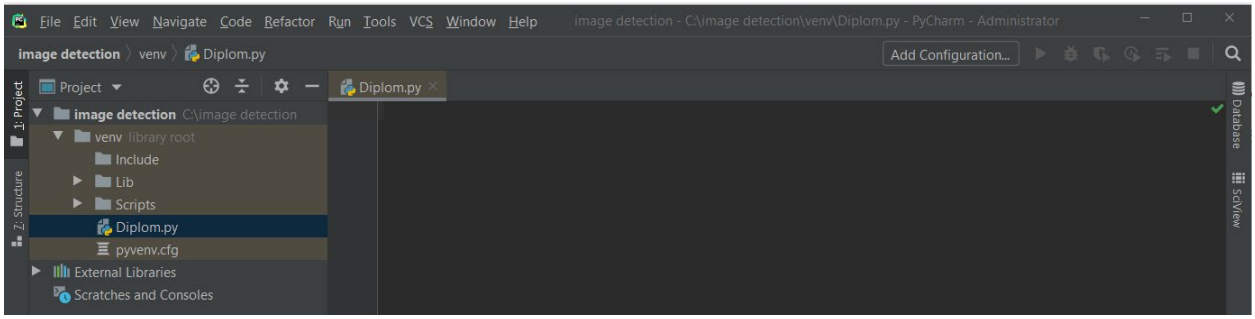


Рисунок 4.12 – Налаштування

4. Далі перевіряємо додаток через тестовий запис “Diplom Borisov Gleb”. В результаті виконання команди можна побачити, що відбулось з’єднання програми та бібліотек (рис.4.13).

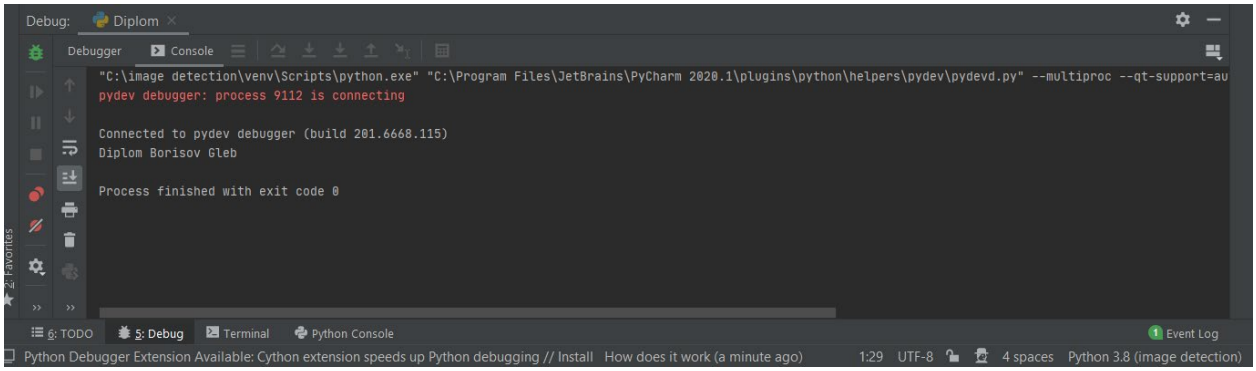


Рисунок 4.13 – Налаштування IDE

5. Наступним кроком є встановлення та завантаження необхідних бібліотек для забезпечення роботи і тестування нейронних мереж. Зокрема, підключимо бібліотеку Image AI: Object Detection. Для її роботи слід попередньо встановити додаткові модулі (бібліотеки). Ці модулі прописують в терміналі програми pyCharm (рис.4.14).

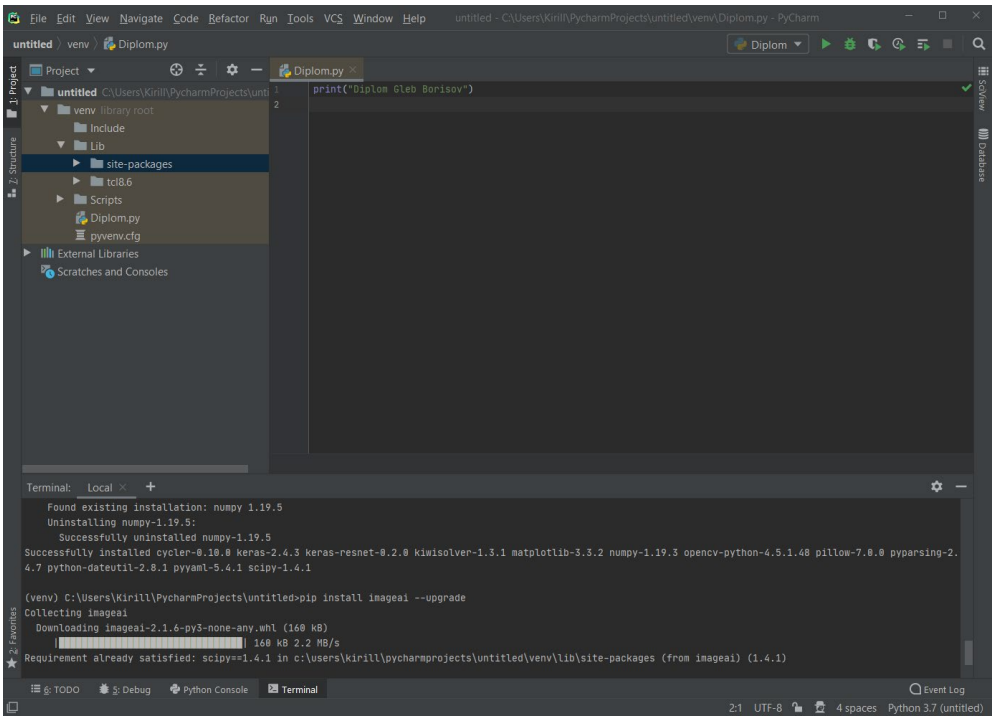


Рисунок 4.14 – Завантаження додаткових модулів

6. Після цього, необхідно встановити модель RetinaNet – модель нейронної мережі з глибоким навчанням. Ця модель візуалізує що саме знаходиться на зображенні або на відео файлі, розуміти як саме описано об’єкти (текстова інформація характеристики об’єкту на зображенні). Об’єкти, які виявляє ця нейронна мережа наведено у таблиці 4.1. Відмітимо, що кількість таких класів визначена на рівні 80 елементів.

Таблиця 4.1 – Перелік класів об’єктів розпізнавання моделі мережі RetinaNet [13]

Умовні номери класів об’єктів	Перелік класів
1	person, bicycle, car, motorcycle, airplane, bus, train, truck, boat, traffic light, fire hydrant, stop_sign;
2	parking meter, bench, bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe, backpack, umbrella, handbag, tie, suitcase, frisbee, skis, snowboard;
3	sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket;
4	bottle, wine glass, cup, fork, knife, spoon, bowl, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donot, cake, chair, couch, potted plant, bed, dining table, toilet, tv, laptop, mouse, remote, keyboard, cell phone, microwave, oven, toaster, sink, refrigerator, book, clock, vase, scissors, teddy bear, hair dryer, toothbrush.

Згідно з завданням на дослідження нас з таблиці 4.1 буде цікавити саме клас об’єктів розпізнавання під номером 1. Крім цього, наведений перелік може свідчити про те, що нейронні мережі можна використовувати не лише для контролю дорожньої ситуації в системах відеоспостереження, але й у побуті. Отже, цю модель далі необхідно розмістити у ту ж папку, де знаходиться файл

Diplom.py. Цим власне завершується підготовчий етап до створення програми з тестування нейронних мереж. Далі наведемо кроки з створення скрипта програми на мові Python.

4.2.2 Алгоритм створення програми для тестування нейронних мереж

Визначимо послідовні кроки до створення програмного коду, який дозволить для набору тестових зображень та відео секвенції провести тестування визначених вище нейронних мереж з метою перевірки зазначених розробниками переваг та недоліків цих мереж. Іншою метою можна вважати формулювання рекомендацій на основі отриманих результатів за результатами їх обговорення.

Таким чином, для створення програми з розпізнавання об’єктів для електронної системи відеоспостереження необхідно виконати наступні послідовні кроки.

1. З завантаженої бібліотеки ImageAI яка реалізує нейронну функцію розпізнавання, з класу detection імпортуємо клас ObjectDetection (рис.4.15).

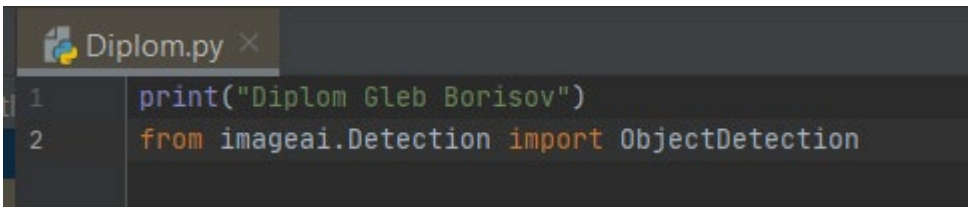


Рисунок 4.15 – Завантаження класу ObjectDetection

Зазначимо, що клас ObjectDetection при аналізі зображення або відео дозволяє виявляти різні об’єкти. Імпортуємо додатково бібліотеку os. Це вбудована бібліотека у мові програмування Python і вона дозволяє працювати з операційною системою.

2. Створюємо нову змінну diplom_path і в неї розмістимо команду os.getcwd() – рядок-функція дозволяє вказати шлях до створеного проекту

(рис.4.16). Це зазначення шляху необхідно щоб можна було виявляти різні об’єкти, наприклад модель Retina.net.

Рисунок 4.16 – Створення змінної

3. На основі класу ObjectDetection створюємо декілька властивостей. Тобто, створимо змінну Detector і в нею помістимо імпортований клас ObjectDetection. Іншими словами, створюємо об’єкт на основі класу ObjectDetection (рис.4.17).

Рисунок 4.17 – Створення об’єкту

4. Використовуючи створену змінну можна звернутись до функції setModelTypeAsRetinaNet з класу ObjectDetection. Тим самим встановлюється те, що нейронна функція-модель RetinaNet використовується для виявлення об’єкту. Вказуємо для створеної змінної та залученої моделі шлях. Так, у змінній diplom_path знаходиться повний шлях до проекту диплому. І далі, у дужках вказуємо який файл слід відкрити для використання моделі RetinaNet (вказуємо шлях до моделі).

5. Завантажуємо модель RetinaNet (рис.4.18). Далі необхідно визначити процедуру співпадіння на зображення. Для цього, створюємо нову змінну з назвою list, де будуть розміщуватись виявлення на зображенні. І в цій змінній, беремо результат відпрацювання функції detectObjectsFrom Image і у дужках цієї функції вказуємо параметр input_image – тобто вихідне зображення, з якого буде проходити нейронний алгоритм виявлення та розпізнавання об'єктів. Для коректної роботи завантаження зображення (розташування зображення на комп'ютері) аналізу необхідно вказати – модуль os, змінну path, функцію join. Функція Join необхідна щоб об'єднати декілька рядків – змінну diplom_path, та назву зображення для аналізу. Наприклад Picture.jpeg. Вказуємо у цьому ж рядку функцію output_image_path – тобто, місце, де буде збережена фотографія, як результат аналізу з характерними ознаками розпізнавання (рис.4.19).

Аналогічним чином завантажують моделі і двох інших нейронних мереж. Так програма, яка дозволяє проаналізувати зображення чи секвенцію на основі нейронної мережі YOLO v.3 або TinyYOLO лише назвою моделі мережі. В першому випадку, слід в скрипті прописати yolo.h5, у другому випадку – yolo-tiny.h5.

Diplom.py

```
print("Diplom Gleb Borisov")
from imageai.Detection import ObjectDetection
import os

diplom_path=os.getcwd()

detector=ObjectDetection()
detector.setModelTypeAsRetinaNet()
detector.setModelPath(diplom_path,"resnet50_coco_best_v2.0.1.h5")
detector.loadModel()
```

Рисунок 4.18 – Завантаження моделі нейронної мережі

Diplom.py

```
print("Diplom Gleb Borisov")
from imageai.Detection import ObjectDetection
import os

diplom_path=os.getcwd()

detector=ObjectDetection()
detector.setModelTypeAsRetinaNet()
detector.setModelPath(diplom_path,"resnet50_coco_best_v2.0.1.h5")
detector.loadModel()

list=detector.detectObjectsFromImage(input_image=os.path.join(diplom_path, "Picture.jpg"),
                                     output_image_path=os.path.join(diplom_path, "Rezultat.jpg"
)

```

Рисунок 4.19 – Програма аналізу зображення на основі нейронної мережі

З метою перевірки правильності створення скрипту візьмемо тестове зображення (рис.4.20) і проаналізуємо його в трьох зазначених нейронних мережах. Результати відпрацювання створеної моделі та алгоритму наведено на рисунках 4.21-4.23.

Рисунок 4.20 – Тестове зображення з набору вихідних

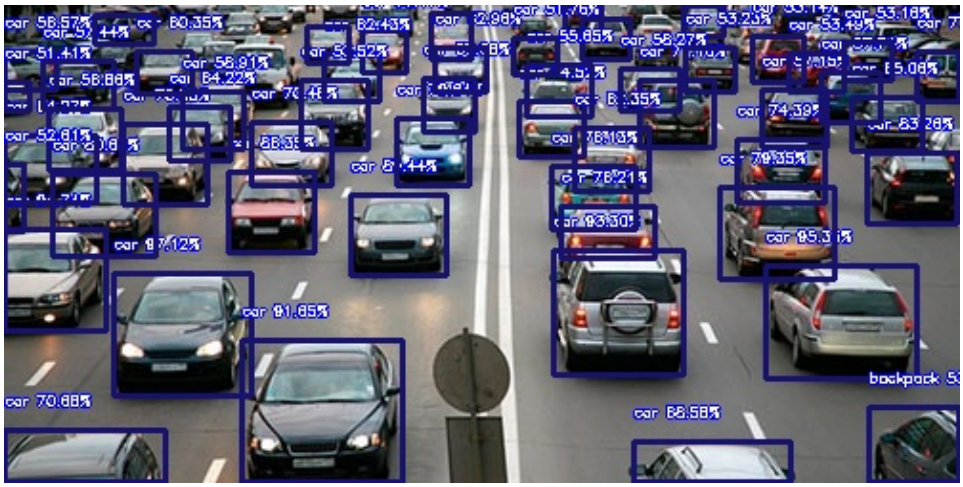


Рисунок 4.21 – Результат на основі моделі RetinaNet

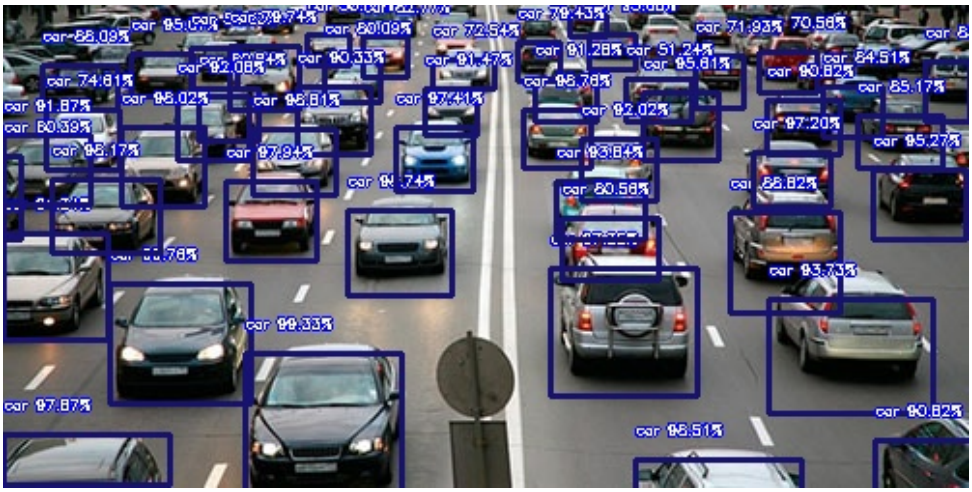


Рисунок 4.22 – Результат на основі моделі YOLO v.3

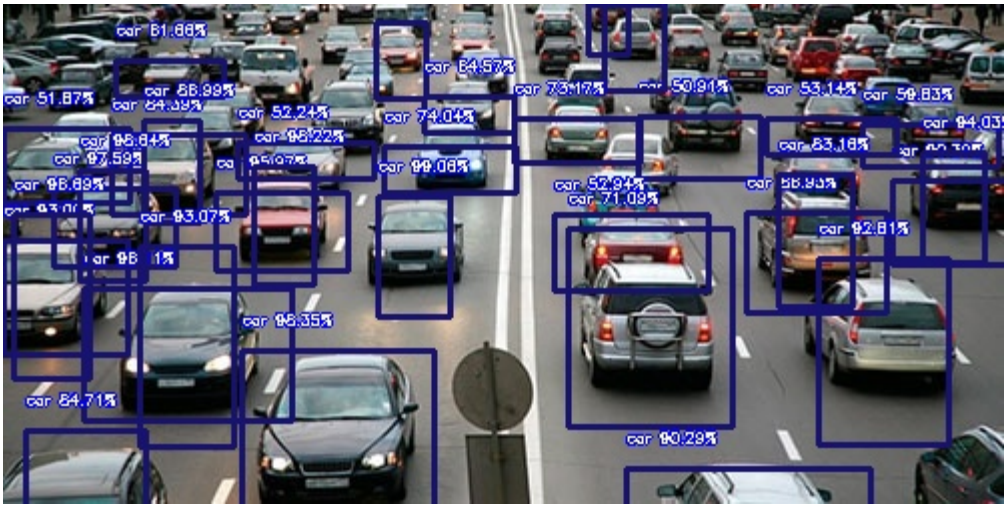


Рисунок 4.23 – Результат на основі моделі TinyYOLO v.3

Аналізуючи отримані результати можна переконатись що розглянуті моделі по-різному проводять процедури розпізнавання і тому далі проведемо комплексний практичний експеримент з тестування мереж на предмет виявлення сильних та слабких сторін нейронних мереж та перевірки особливостей, які були нами наведено у п.4.1.1-4.1.3 дисертації.

4.3 Обробка та аналіз результатів експерименту

4.3.1 Аналіз статичних зображень з камер відеоспостереження

Проведемо спочатку аналіз набору статичних зображень, які визначають певні дорожні ситуації і які, як правило фіксуються вуличними камерами системи відеоспостереження. При цьому при виборі 15 зображень враховувались певні наявні особливості, а саме: різні види освітлення, різні форми об’єктів, їх розташування та віддаленість від центру об’єктива тощо. Головною метою дослідження є визначення переваг та недоліків трьох нейронних мереж, які описано в п.4.1 та визначення передумов, які призводять до необхідності навчання нейронної мережі задля отримання покращених результатів розпізнавання об’єктів на зображенні.

Спочатку розглянемо ситуацію напруженого дорожнього трафіку (див. рис.4.20). В якості об’єктів для дослідження оберемо три автомобілі, які знаходяться на різній відстані від камери відеоспостереження і відповідно позначені цифрами 1,2,3 (рис.4.24). Відмітимо, що для даного зображення (позначимо це зображення під номером 1) об’єкт під номером 1 знаходиться в ідеальних умовах для розпізнавання, оскільки він не має перешкод і знаходиться найближче до точки зйомки. Водночас, об’єкт 2 показано не повністю (закритий іншим транспортним засобом) і в значній перспективі. Відповідно об’єкт 3 зафіксовано з тилової сторони і тут також є перетин з іншим автомобілем.

Рисунок 4.24 – Обрані об’єкти для аналізу (зображення 1)

Результат роботи нейронних мереж в практичному експерименті побудуємо у формі вертикальних стовпчиків з поділом на три області. Ці області будуть визначити тип об’єкту. Висота стовпчиків визначає точність розпізнавання об’єктів, колір стовпчика – тип нейронної мережі. Результат для зображення з рисунку 4.24 наведено на рисунку 4.25.

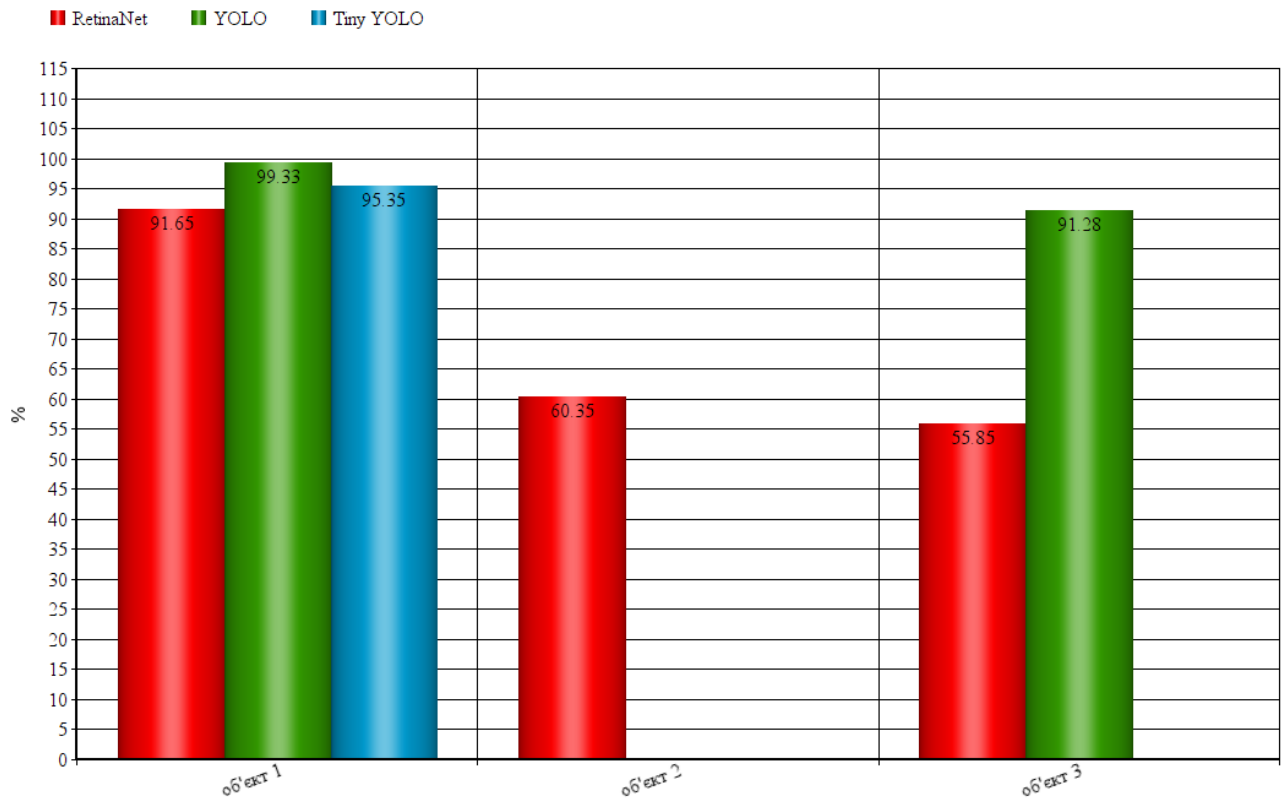


Рисунок 4.25 – Отриманий результат для зображення 1

З отриманих результатів можна підкреслити, що в умовах насиченого дорожнього руху нейронна мережа TinyYOLO для фіксації автомобілів, які або затіняються або знаходяться поодаль не коректно проводить розпізнавання. Для даного зображення 1 мережа для об’єктів 2 та 3 не змогла навіть визначити клас об’єкту. Крім цього, можна побачити, що і розширена версія мережі YOLO для випадку, коли автомобіль припарковано не завжди спрацьовує. І нарешті, для моделі RetinaNET хоч і отримані результати для трьох об’єктів, але вони для об’єктів 2 та 3 є значно менші за точністю за величину 78%. Таким чином, теза про те, що нейронна мережа YOLO дозволяє отримати високу точність для невеликих об’єктів у випадку, якщо є перешкоди на жаль не виконується. Підсумовуючи отримані результати, можна дійти висновку, що в умовах насиченого трафіку нейронні мережі повинні проходити крізь багатошарову процедуру навчання.

Візьмемо ще одну дорожню ситуацію, але в даному випадку спостерігається значна менша кількість транспортних засобів. В якості об’єктів оберемо три автомобілі, які не мають перешкод, але знаходяться на різних відстанях від камери (рис.4.26). Зазначимо, що обраний кадр характеризується майже ідеальними умовами освітлення, адже він отриманий в сонячну ясну погоду.

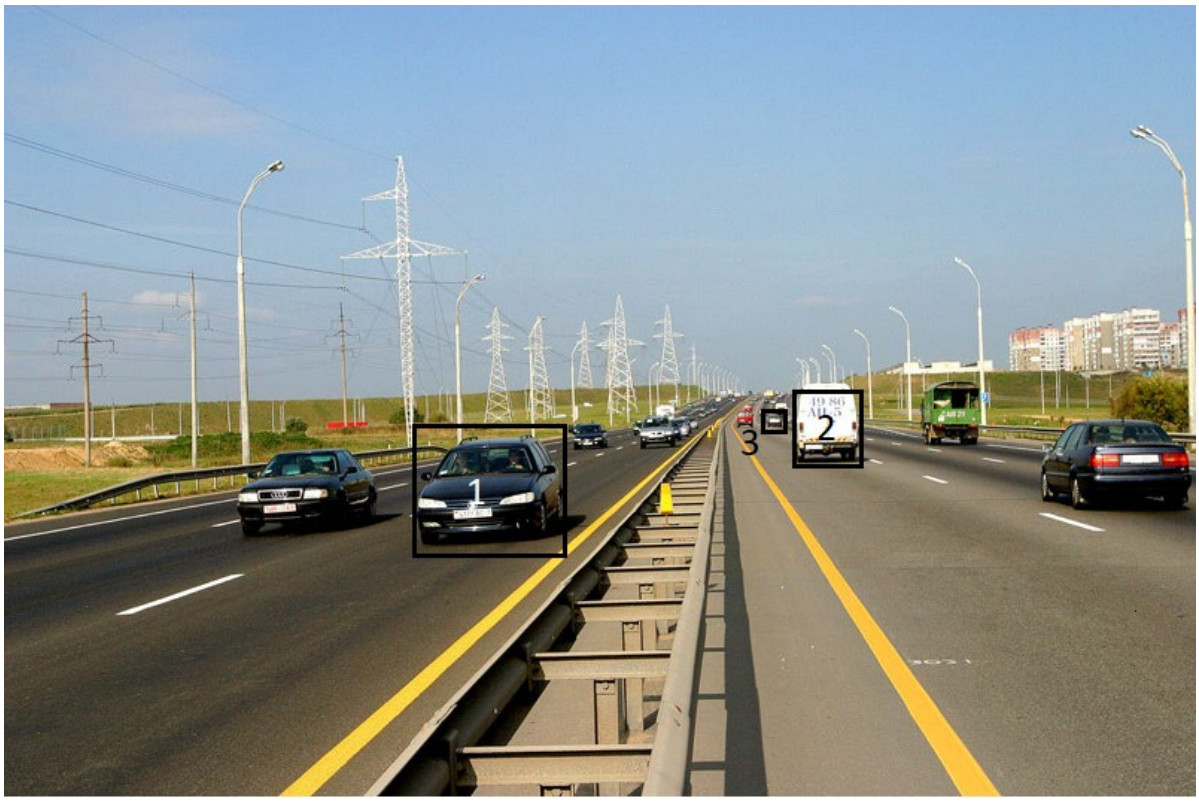


Рисунок 4.26 – Обрані об’єкти для зображення 2

Результати функціонування нейронних мереж для зображення 2 та обраних об’єктів показано на рисунку 4.27. На відміну від зображення 1 для даного випадку маємо ситуацію, коли об’єкт 3 не був визначений і нейронною мережею RetinaNet і нейронною мережею TinyYOLO. Більше того, об’єкт 2 не ідентифікувався нейронною мережею TinyYOLO. Разом з тим, для об’єкту 1 отримані результати за точністю досить високі. Таким чином, як і для випадку зображення 1 ситуація така сама – нейронна мережа TinyYOLO погано розпізнає віддаленні об’єкти від камери. І при цьому немає значення чи є перешкоди для зйомки, чи немає. Натомість, нейронна мережа YOLO за умови отримання чітких, якісних знімків дає досить високу точність.

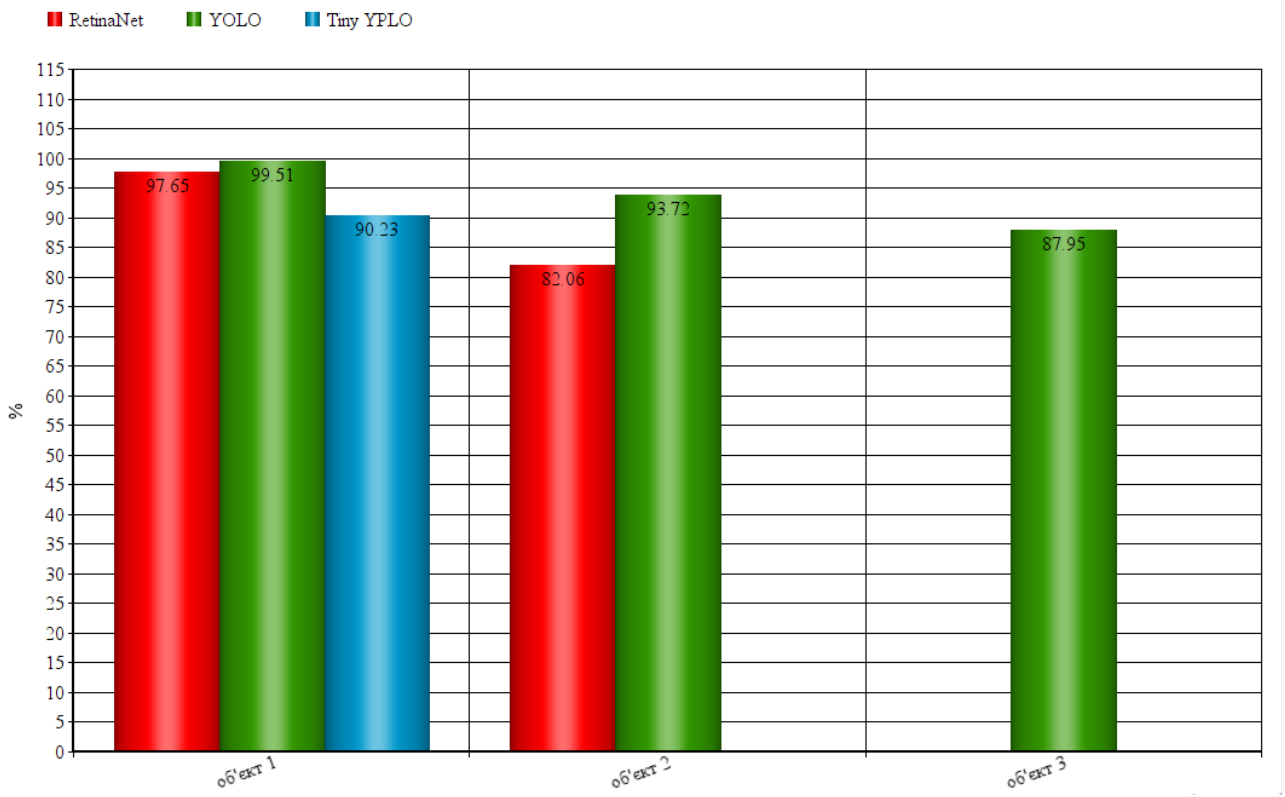


Рисунок 4.27 – Отриманий результат для зображення 2

Розглянемо варіант, коли дорожня ситуація приблизно така сама як і для зображення 2, але камера фіксує кадри з низькою роздільною здатністю (рис.4.28).

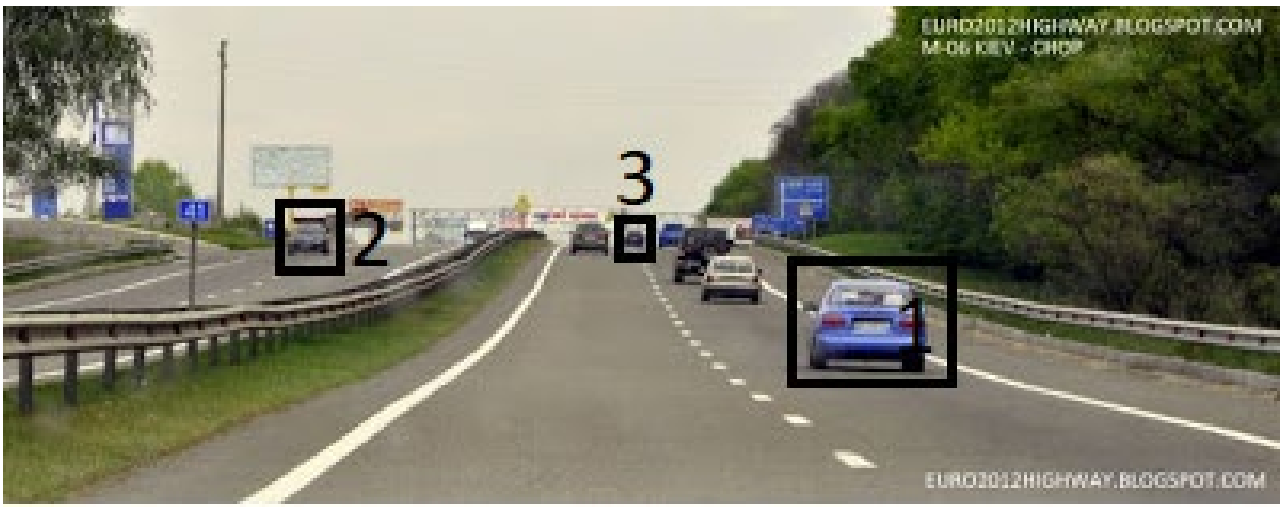


Рисунок 4.28 – Зображення 3 з виділеними об’єктами

Відзначимо, що в зображенні 3 обрано відразу два віддаленні об’єкти – об’єкт 2 та об’єкт 3. Результати відпрацювання нейронних мереж з відключеним режимом навчання наведено на рисунку 4.29. Аналіз отриманих

результатів підтвердив факт, що нейронна мережа TinyYOLO, якість її функціонування залежить і від якості об'єктива камери відеоспостереження. Крім цього, отримано підтвердження що ця мережа дуже погано проводить процедуру розпізнавання віддалених об'єктів, розміри яких складають в ширину (по горизонту зйомки) до 20-30 пікселів. Крім цього, можна відмітити що мережа RetinaNET також ідентифікує невеликі об'єкти. Для зображень 2 та 3 це об'єкти під номером 3. І в цьому випадку можна передбачити, що якість зйомки не відіграє особливо важливу роль.

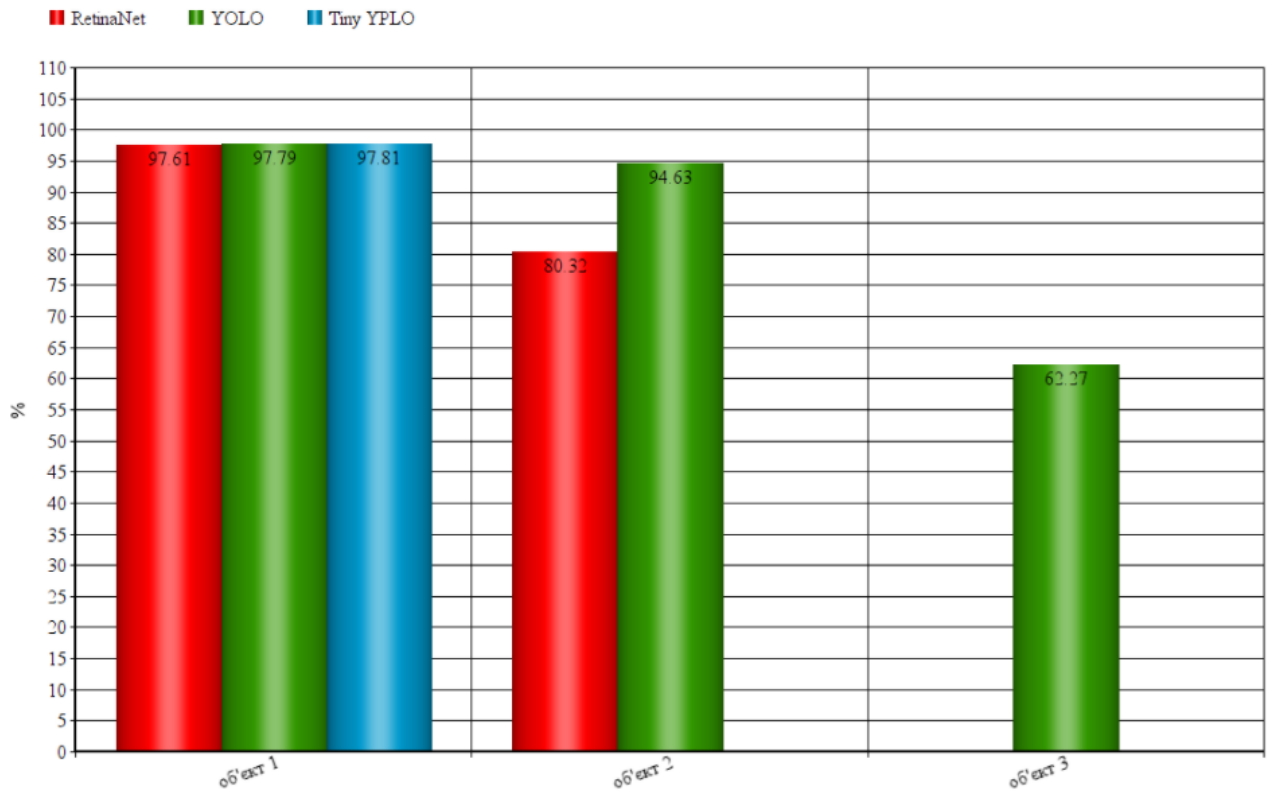


Рисунок 4.29 – Отриманий результат для зображення 3

В умовах контролю за дорожнім рухом нейронні мережі здатні проводити розпізнавання і інших об'єктів сцени. Наприклад, візьмемо зображення 4, де в якості об'єкту 3 візьмемо дорожні ліхтарі (рис.4.30). За визначенням роботи мереж цей клас об'єктів називається traffic light (див. табл.4.1). Результат роботи нейронної мережі YOLO версії 3 показано на рисунку 4.31. А зведені результати для роботи трьох мереж показано у графічному вигляді на рисунку 4.32. З них можна дійти висновку, що нейронна мережа RetinaNet не

ідентифікувала вуличні дорожні ліхтарі. І справа тут, напевно, не в розмірах об’єкту. Ну і додатково отримано, як і раніше що мережа TinyYOLO не визначає віддаленні об’єкти в перспективі.

З аналізу чотирьох розглянутих зображень найбільш стабільну роботу, за виключенням зображення 1, показала нейронна мережа YOLO.

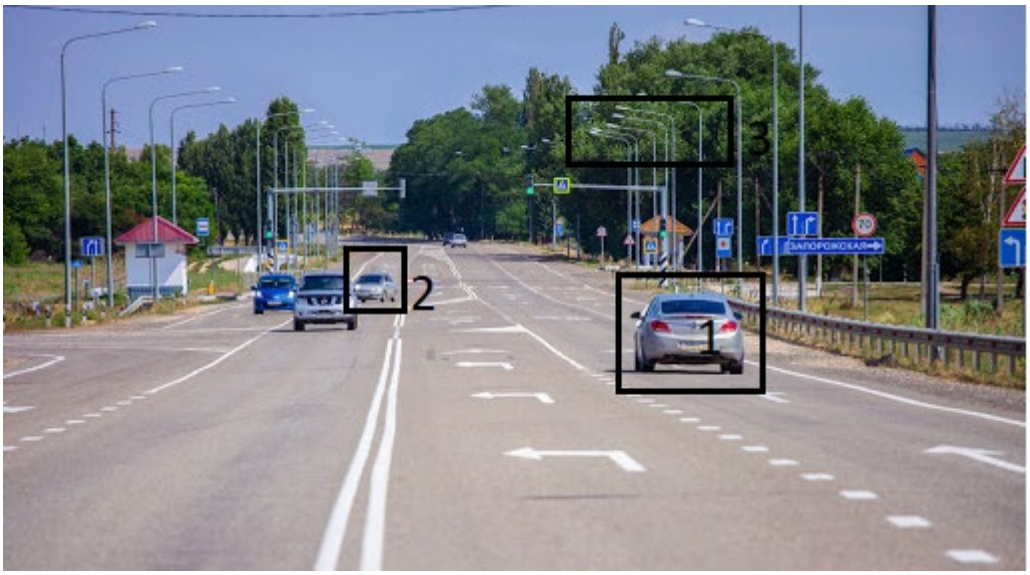


Рисунок 4.30 – Зображення 4 з виділеними об’єктами

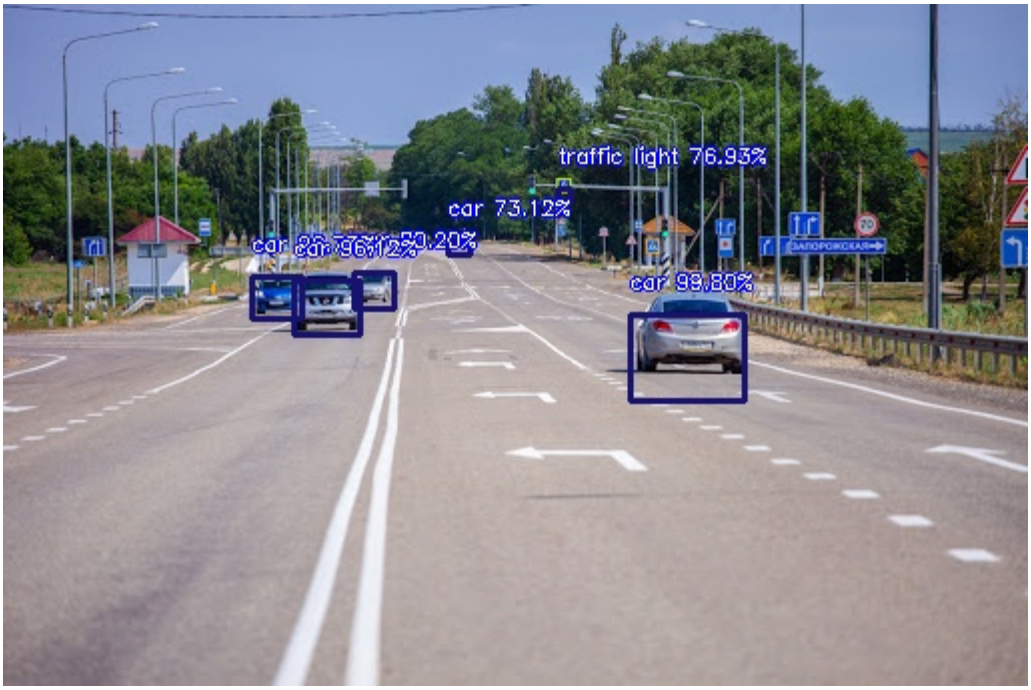


Рисунок 4.31 – Результат роботи мережі YOLO для зображення 4

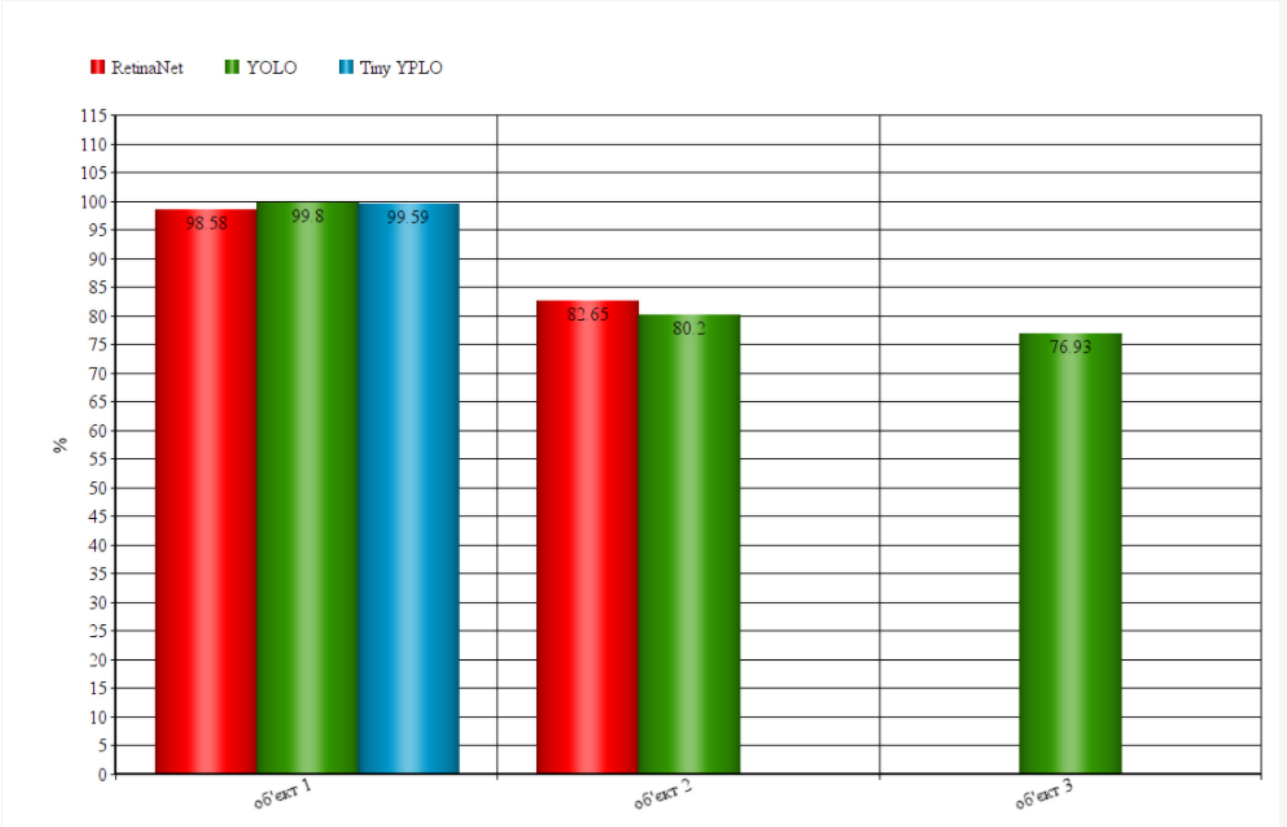


Рисунок 4.32 – Отриманий результат для зображення 4

Далі в експерименті змінимо умови зйомки (похмура погода) та тип об’єктів для розпізнавання. Тобто, візьмемо зображення 5, де визначимо не лише легкові автомобілі, але й вантажні автомобілі (рис.4.33).

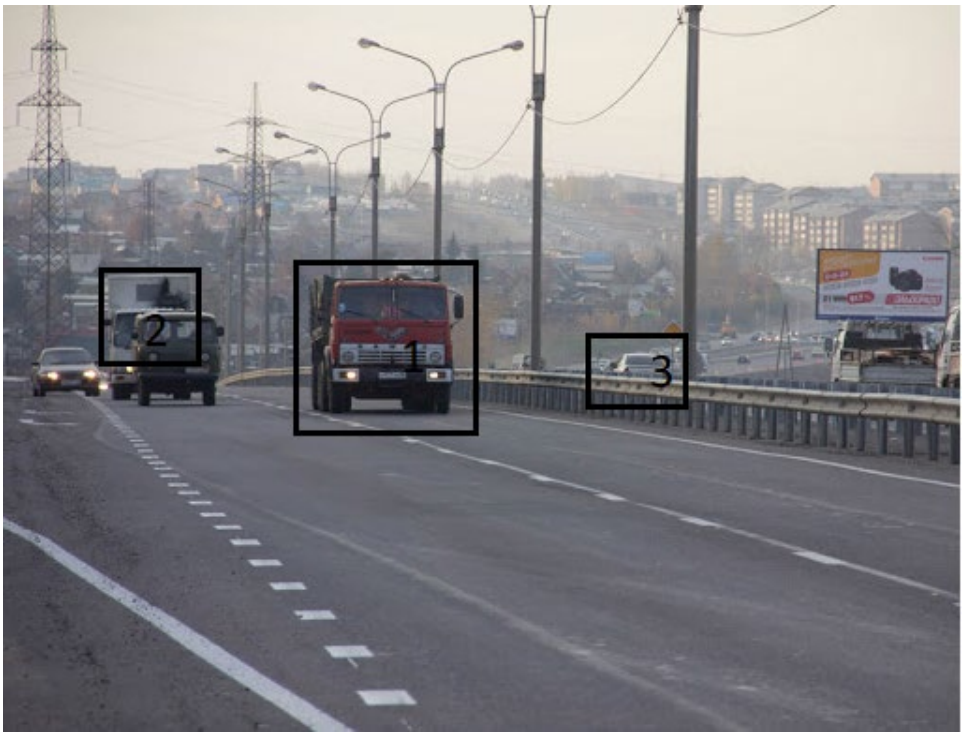


Рисунок 4.33 – Зображення 5 з виділеними об’єктами

Отримані результати аналізу наведено на рисунку 4.34. Тут зокрема, можна відмітити, що мережа RetinaNet не розпізнала об’єкт 2, виділивши при цьому сусідній елемент зображення (рис.4.35). Цей факт ще раз підтверджує те, що віддаленні об’єкти ця мережа за умови сторонніх об’єктів перетинання може визначати з помилкою. Іншими словами, ця мережа потребує додаткового навчання для поліпшення отриманих результатів за точністю. Також з отриманих результатів випливає, що на якість роботи TinyYOLO впливає не лише віддаленість елементів, але й перетин зі сторонніми об’єктами.

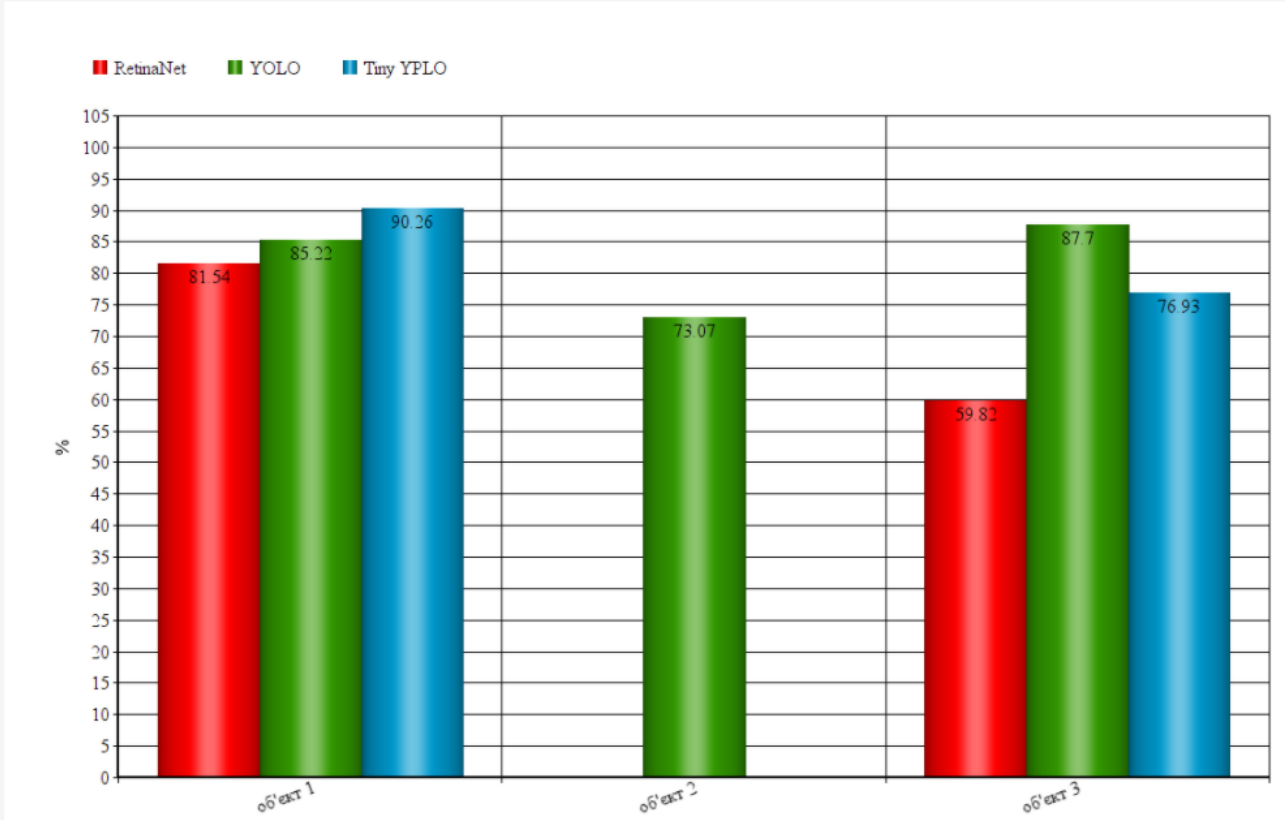


Рисунок 4.34 – Отриманий результат для зображення 5

Змінимо трохи умови вибору об’єктів і для зображення 6 в якості останніх виберемо не лише інший тип транспортного засобу, але й об’єкт у вигляді керманича мотоцикла. Крім цього, перевіримо чи впливає на роботу нейронних мереж засвітлення кадру зйомки (рис.4.36). Отримані результати аналізу наведено на рисунку 4.37. Аналізуючи отримані дані можна помітити, що мережі ідентифікували за двома класами водія на мотоциклі. Це може свідчити про те, що моделі мереж працюють виключно за класами без

елементів навчання і до того ж, це фактично є помилкою розпізнавання з огляду на об’єкт 3 того самого зображення 6. До того, ж порівнюючи між собою нейронні мережі RetinaNet та YOLO можна дійти висновку, що для визначених об’єктів мережа YOLO є більш точною, і в середньому її показники точності або ймовірності приналежності об’єкту до класу та його розташування в певній області (локалізації) є вищими на 10 відсотків і ця точність збільшується при віддаленні об’єкта аналізу від камери фіксації.

Рисунок 4.35 – Результат роботи мережі RetinaNet для зображення 5

Рисунок 4.36 – Зображення 6 з виділеними об’єктами

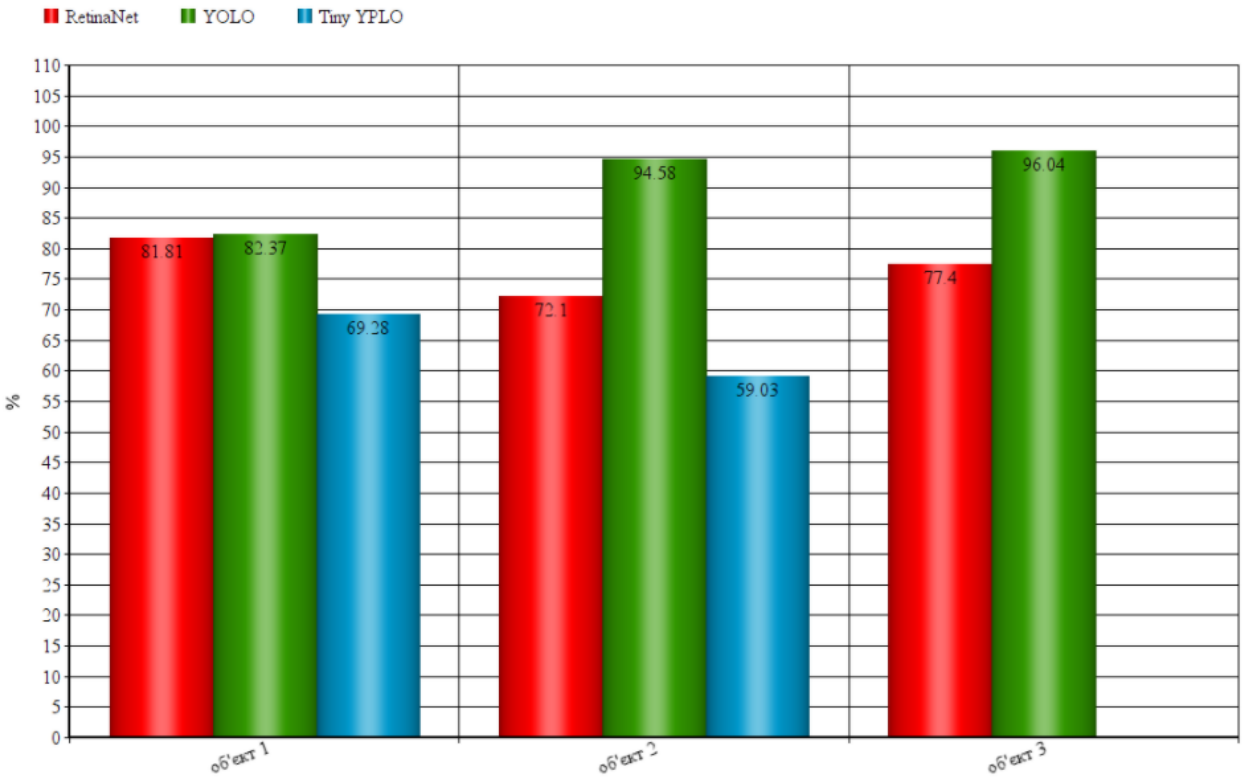


Рисунок 4.37 – Отриманий результат для зображення 6

Далі, дещо ускладнимо умови практичного експерименту і візьмемо для аналізу не просто дорожній рух транспортних засобів, а дорожньо-транспортну пригоду. Така ситуація показана на рисунку 4.38 (зображення 7).

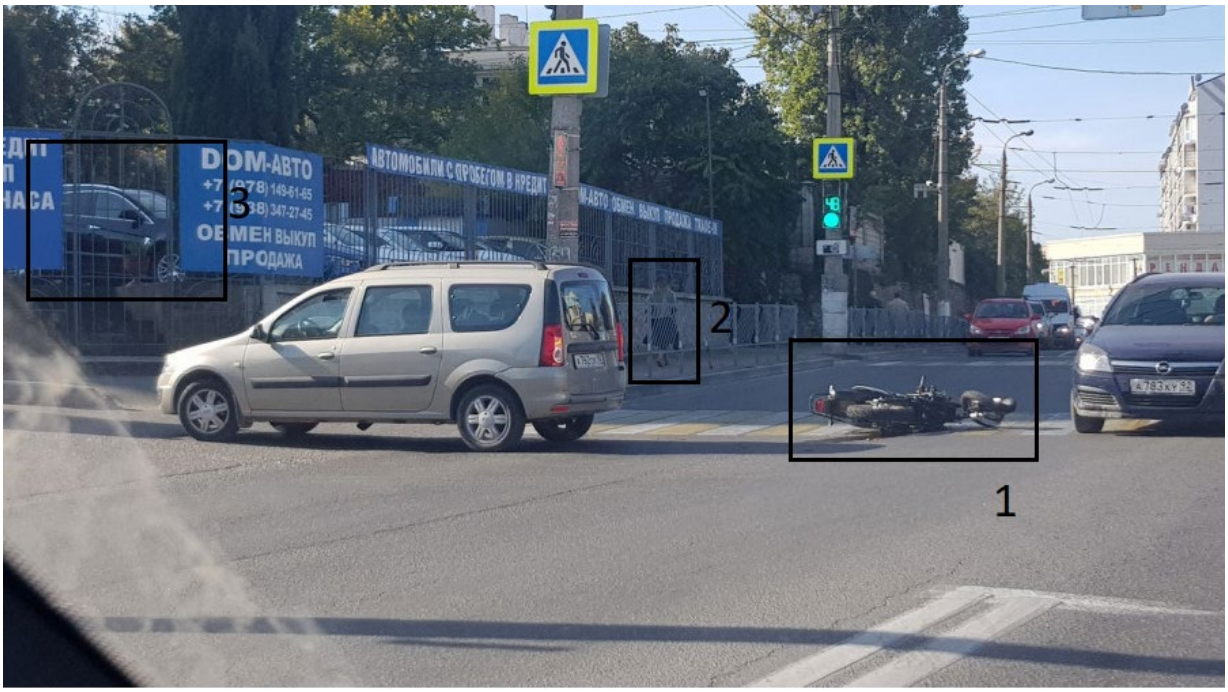


Рисунок 4.38 – Зображення 7 з виділеними об’єктами

В якості об’єктів обрано перекинутий мотоцикл, автомобіль та стороння людина в умовах затінку. Результати відпрацювання нейронних мереж показано на графіку (рис.4.39). За цими результатами можна побачити, що мережа TinyYOLO не розпізнала жодного обраного об’єкту (рис.4.40). До того ж, отримані дані точності за усіма моделями знаходяться на не достатньому рівні 55-58 %. Додатково, модель YOLO не змогла ідентифікувати а ні об’єкт 2, а ні об’єкт 3 (рис.4.41). Враховуючи все вище сказане, варто відзначити що ен типові дорожні ситуації при відео та фото фіксації значно знижують точність роботи нейронної мережі. І для підвищення якості роботи системи відео фіксації необхідно проводити додаткове навчання мережі.

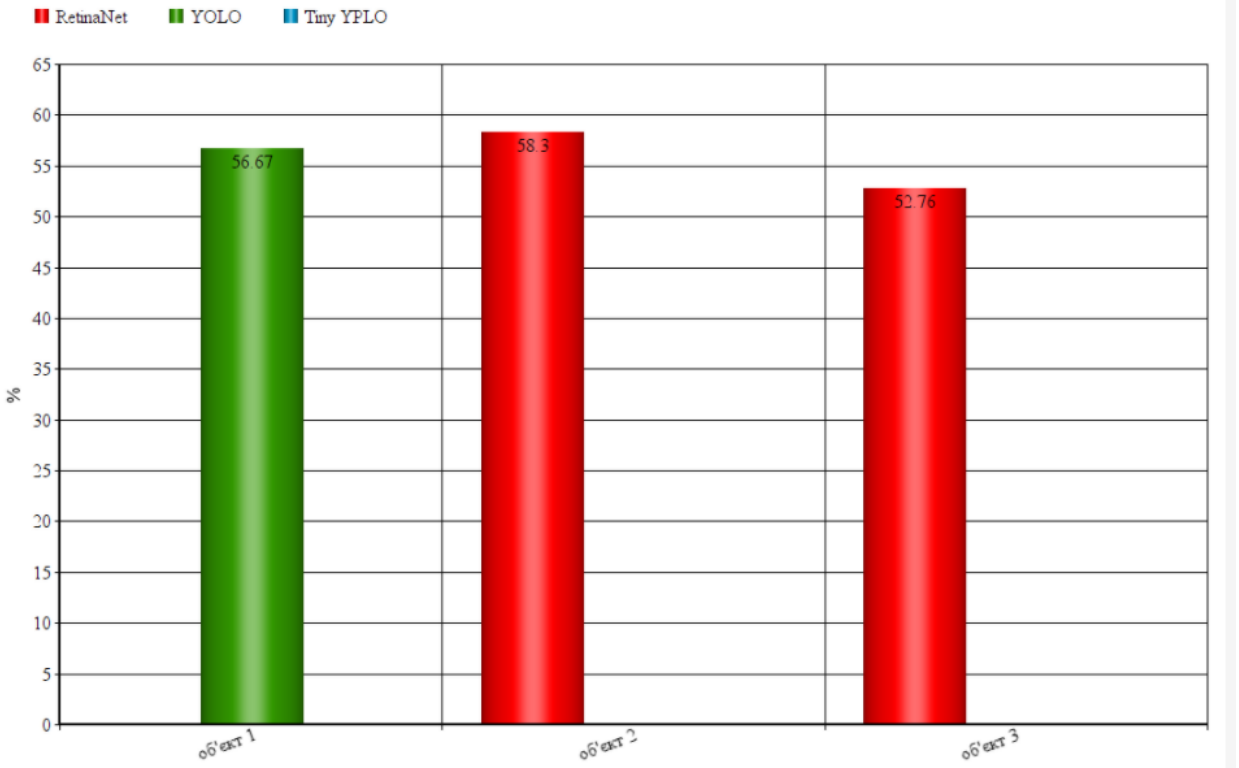


Рисунок 4.39 – Отриманий результат для зображення 7

Рисунок 4.40 – Результат роботи мережі TinyYOLO для зображення 7

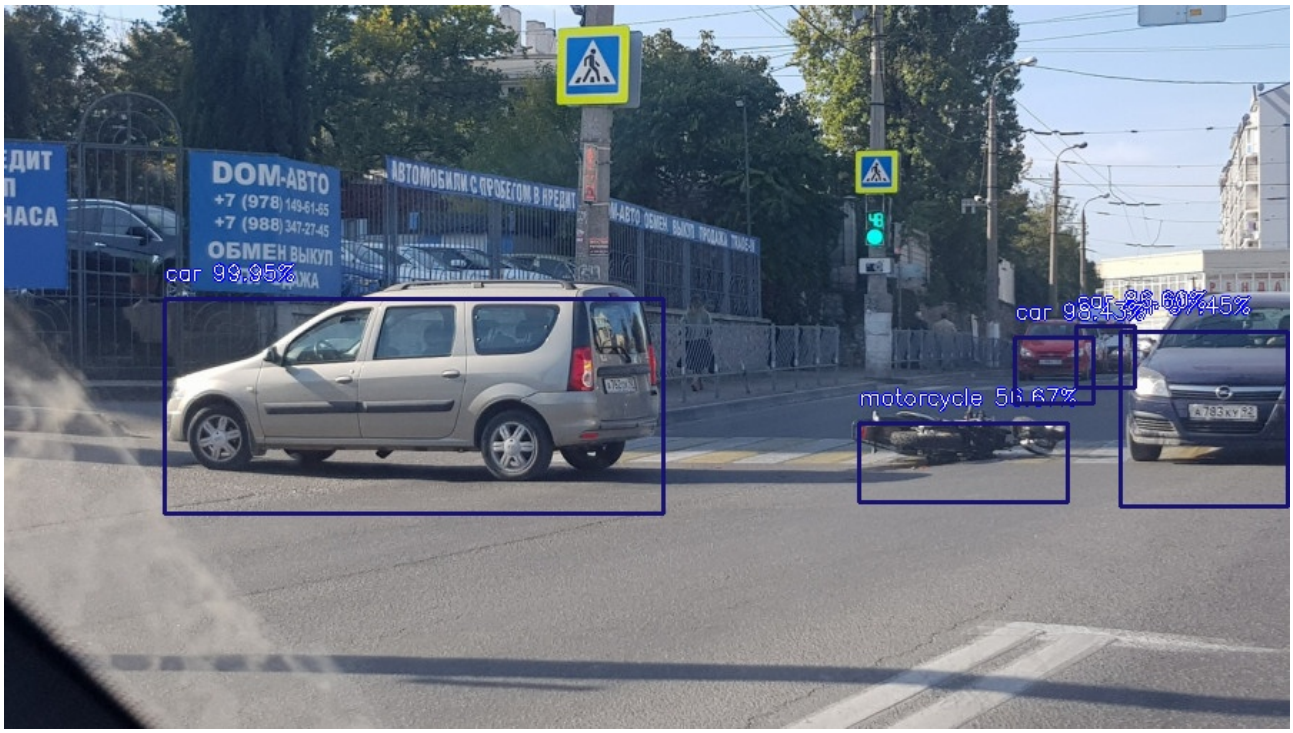


Рисунок 4.41 – Результат роботи мережі YOLO для зображення 7

Далі в межах дослідження, перевіримо як нейронні мережі відпрацьовують ситуації, коли об’єкти є віддаленими і візуально схожі на транспортний засіб. Розглянемо зображення 8, де поодаль є силует схожий на автомобіль, але насправді це просто кущі (рис.4.42).

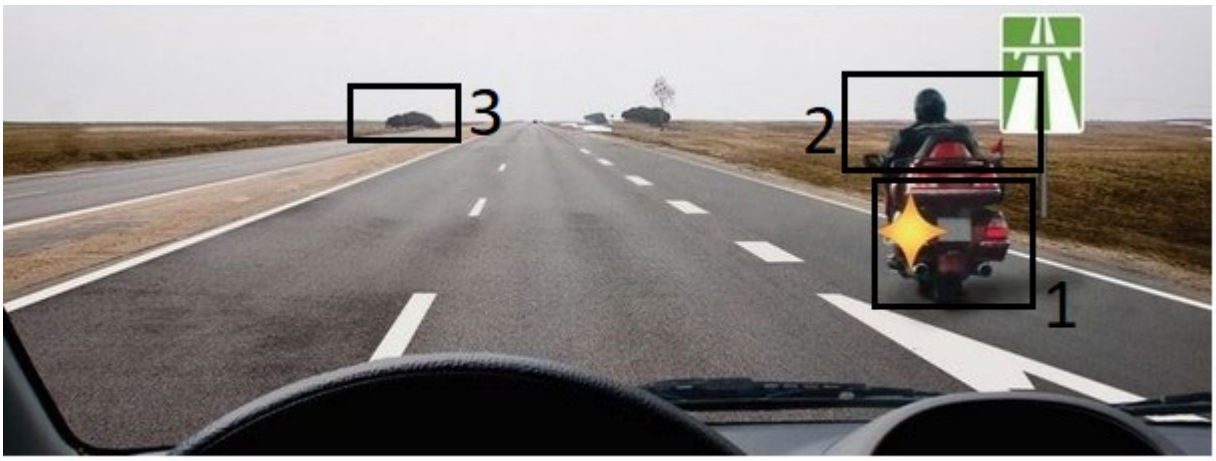


Рисунок 4.42 – Зображення 8 з виділеними об’єктами

Результати розпізнавання елементів на зображенні наведено на рисунку 4.43. І тут цікавою особливістю роботи нейронних мереж, як обрані об’єкти булі

ідентифіковані за класами з таблиці 4.1. Так, наприклад мережа YOLO визначила об’єкт 3 як автомобіль, хоча насправді це кущі або невеличке дерево уздовж узбіччя дороги (рис.4.44). Тобто, потрібно проводити навчання цієї мережі.

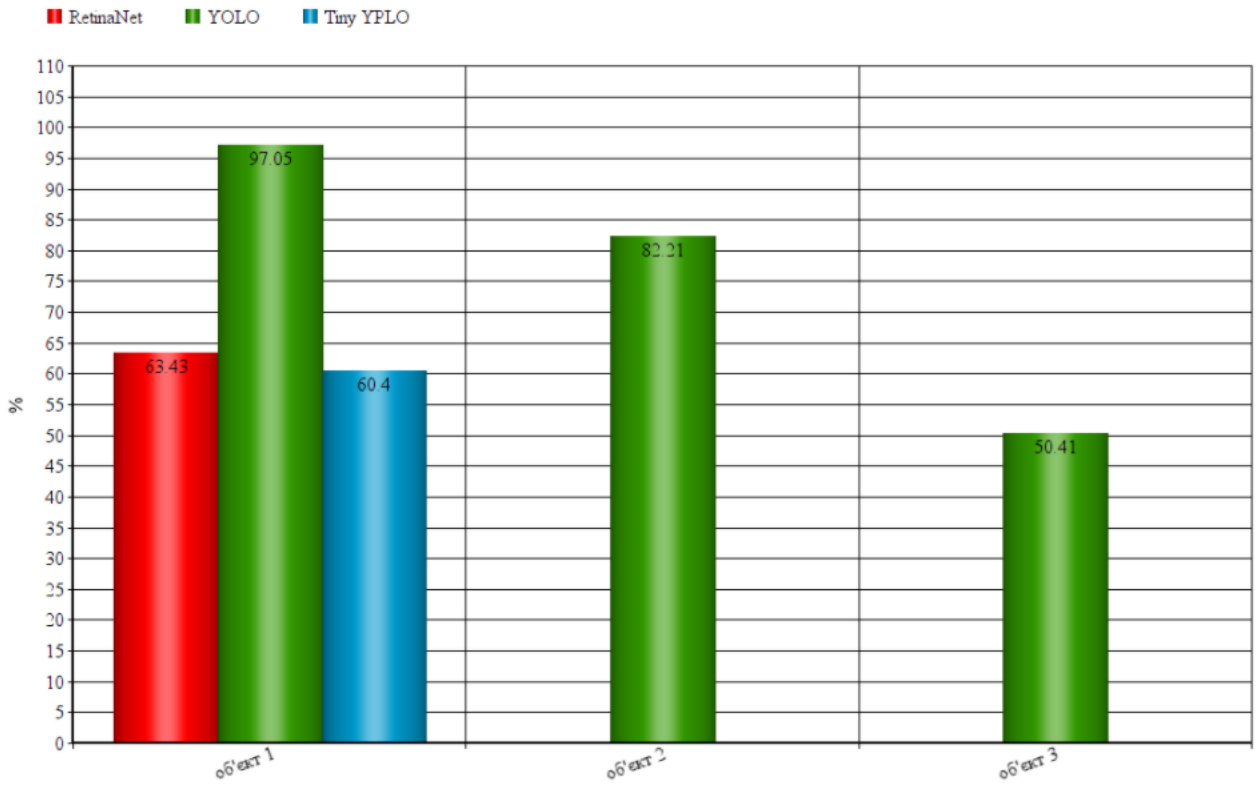


Рисунок 4.43 – Отриманий результат для зображення 8

Рисунок 4.44 – Результат роботи мережі YOLO для зображення 8

Цю обставину обробки зображення слід, напевно, враховувати при використанні зазначеної нейронної мережі та інших мереж у розгалуженій

транспортній інфраструктурі міста. Для перевірки цього розглянемо відповідно зображення 9 та 10.

Для прикладу спочатку розглянемо зображення 9, яке показано на рисунку 4.45.

Рисунок 4.45 – Зображення 9 з виділеними об’єктами

Відмітимо, що в зображенні 9 є фактори які повинні, начебто, ускладнювати процес розпізнавання – частина об’єкту, яка закрита іншими об’єктами, відбиття зображення мотоцикліста, сторонні транспортні засоби. Результати відпрацювання нейронних мереж за цим зображенням показано на рисунку 4.46. Звідси можна підкреслити, що мережа TinyYOLO не коректно працює в умовах насиченої інфраструктури міста (жоден об’єкт з обраних не ідентифіковано). Крім цього, отримано що мережа RetinaNet має вкрай низькі значення точності розпізнавання обраних об’єктів, що ніяк не корелюється з показником у 78 відсотків (див.п.4.1.1). З мереж найкращою в даному випадку виявилась мережа YOLO.

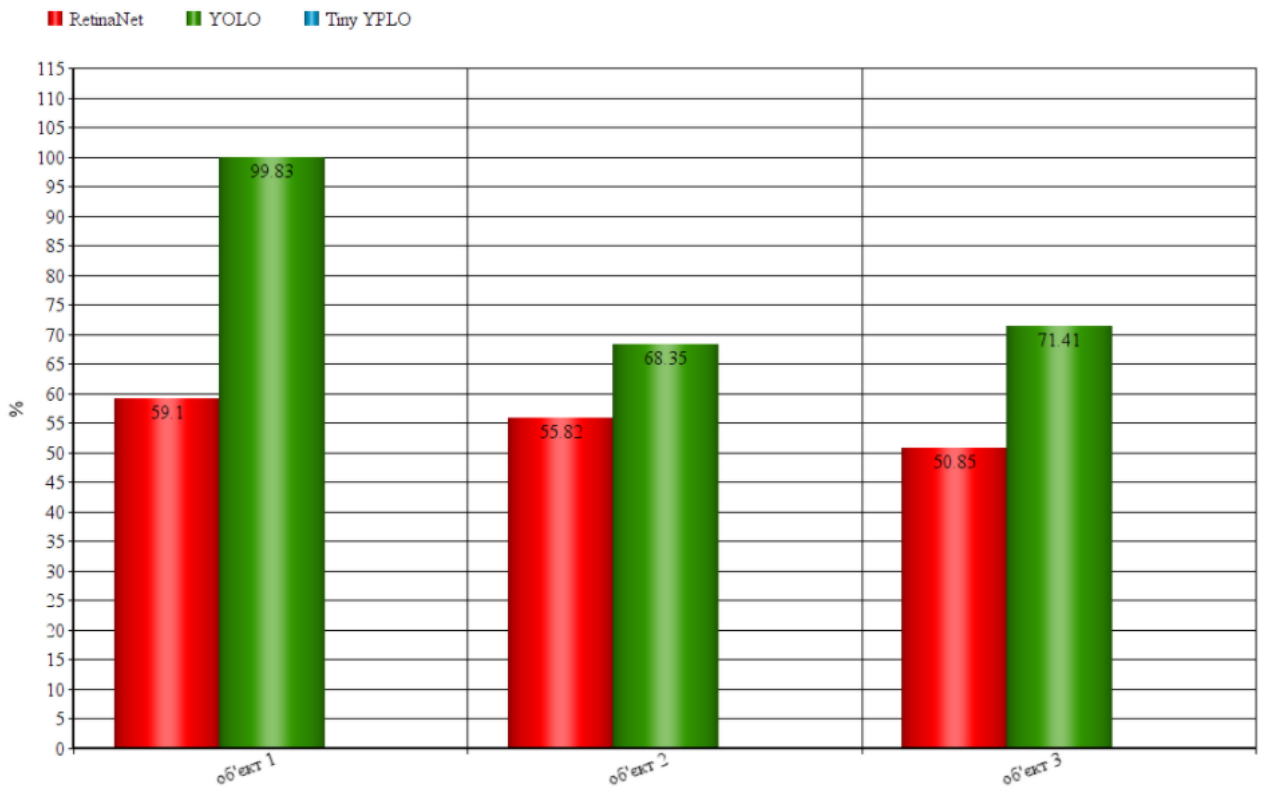


Рисунок 4.46 – Отриманий результат для зображення 9

Наприкінці розгляду статичних зображень, розглянемо ситуацію коли на передньому плані є трамвай (зупинка громадського транспорту, станція Польова, м. Київ) (рис.4.47). Крім цього, додатковою умовою є затінок в центрі кадру та в якості об’єкту 3 обрано рюкзак людини (backpack). Результати відпрацювання моделей показано на рисунку 4.48. Тут можна відмітити, що дві мережі ідентифікували з високою точністю трамвай (за класом), тоді як третя мережа TinyYOLO цей об’єкт не локалізувала.

Рисунок 4.47 – Зображення 10 з виділеними об’єктами

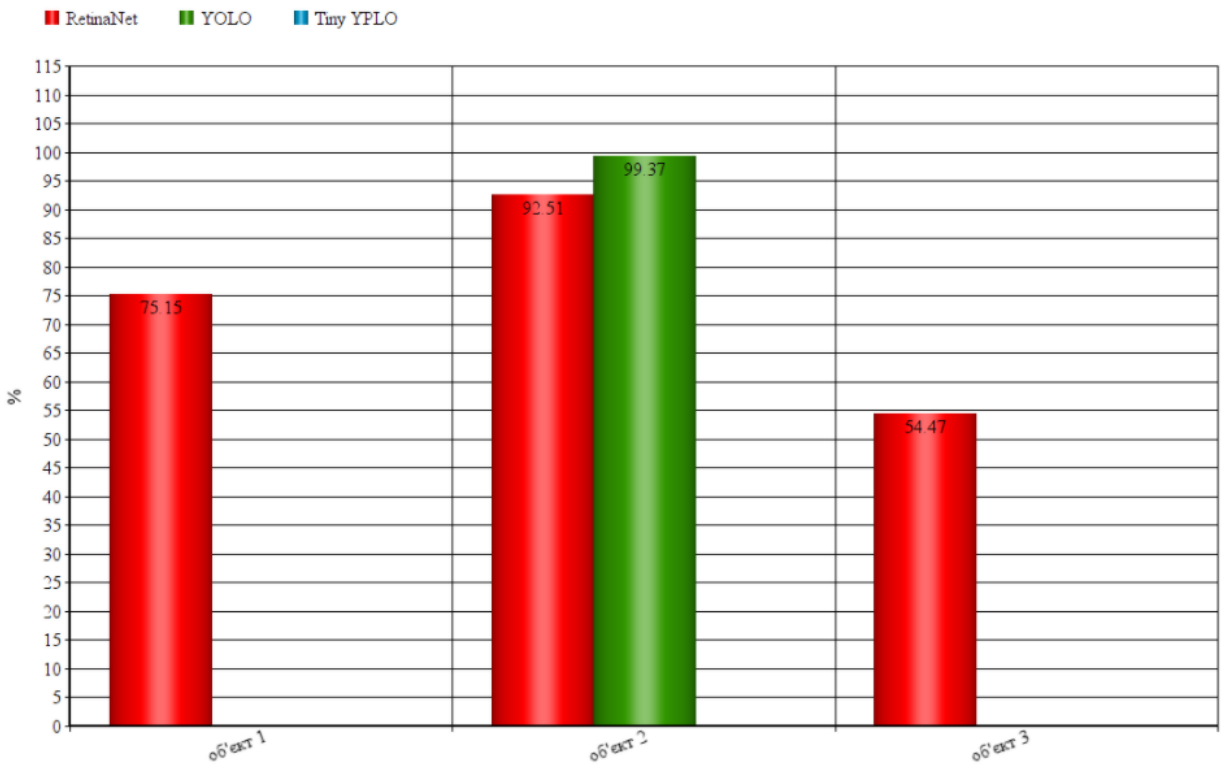


Рисунок 4.48 – Отриманий результат для зображення 10

Для переходу до динамічної секвенції, спочатку розглянемо як нейронні мережі аналізують окремий кадр цієї секвенції. Дане зображення під номером 11 представлено на рисунку 4.49.

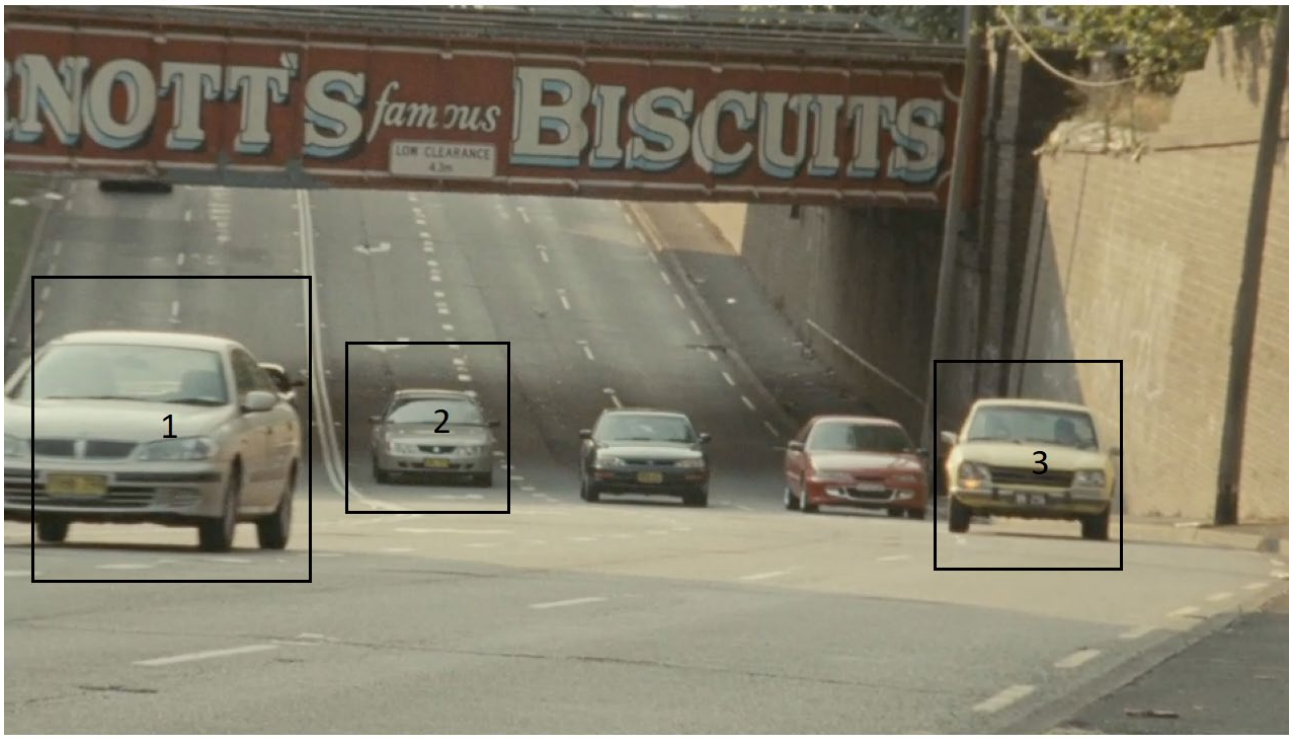


Рисунок 4.49 – Зображення 11 з виділеними об’єктами

Зображення 11 не є складним для нейронних мереж, але виявлено цікавий факт – нейронна мережа TinyYOLO продублювала об’єкт під номером 3. Тобто, фактично визначила що в цій області зображення два об’єкти, коли насправді лише один (рис.4.50). Іншими словами, мережу TinyYOLO виходячи з виявлених недоліків за проведеним аналізом не будемо далі використовувати для динамічної секвенції.

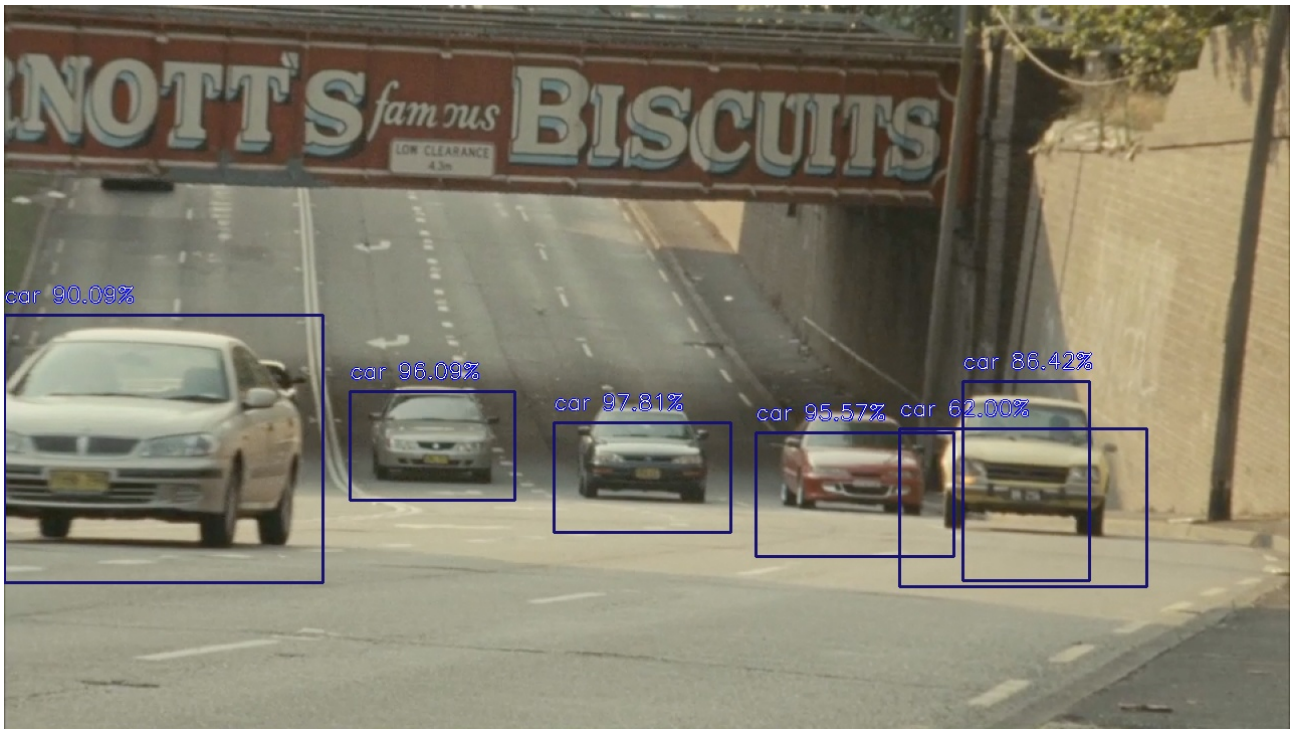


Рисунок 4.50 – Результат роботи мережі TinyYOLO для зображення 11

На основі проведеного практичного експерименту для ряду зображень, які ілюструють різні дорожні ситуації у місті, і які цілком ймовірно можуть виникнути при роботі вуличної системи відеоспостереження доцільно зробити наступні висновки з результатів роботи трьох розглянутих нейронних мереж за умови однакових вихідних даних проведеного експерименту. Зокрема, варто відзначити наступне:

1. Нейронну мережу TinyYOLO не рекомендується використовувати в якості основної при розгортанні вуличної системи відеоспостереження для виконання функцій розпізнавання об’єктів. Наприклад, для зображення 1 дана мережа в умовах насиченого дорожнього руху не коректно ідентифікувала автомобілі, які або затіняються або знаходяться поодаль. Більше того, мережа для об’єктів 2 та 3 не змогла навіть визначити клас об’єкту. Те саме можна сказати і при аналізі зображення 4, коли не були визначені дорожні ліхтарі. Крім цього, мережа TinyYOLO не розпізнає віддалені об’єкти (зображення 2 та 3) не залежно від якості картинки камери, умов зйомки.

2. Нейронна мережа RetinaNet може помилково плутати розпізнавання між щільно розташованими об’єктами (зображення 5). Подібну ознаку, але з

дублюванням об’єктів (створення клонованого об’єкту з оригіналу) може встановлювати мережа TinyYOLO (зображення 11). Ці помилки розпізнавання вочевидь виправляються шляхом додаткового навчання нейронної мережі.

3. Порівнюючи між собою нейронні мережі RetinaNet та YOLO (зображення 6 та 9) можна відмітити, що мережа YOLO є більш точною при розпізнавання однакових об’єктів. Так, для визначених об’єктів мережа YOLO має більші показники точності приналежності об’єкту до класу та його розташування в певній області (локалізація) і вони становлять в середньому на 10 відсотків більше у порівнянні з аналогічними показниками мережі RetinaNet.

4. На основі аналізу зображення 8 знайдено, що мережа YOLO може плутати класи розпізнавання об’єктів. Ця обставина усувається при проведенні навчання мережі.

5. Знайдено, що мережа TinyYOLO не коректно працює в умовах насиченої інфраструктури міста (жоден об’єкт з обраних не ідентифіковано у зображенні 9). Крім цього, отримано що для зображення 9 мережа RetinaNet має вкрай низькі значення точності розпізнавання обраних об’єктів, що ніяк не корелюється з показником у 78 відсотків.

6. Нейронна мережа TinyYOLO може створювати клоновані об’єкти розпізнавання. Така ситуація спостерігається при аналізі зображення 11, коли в реальності замість одного транспортного засобу ідентифіковано два, причому другий в реальності в кадрі відсутній.

Підсумовуючи усе вище отримане можна дійти висновку, що за результатами порівняння трьох нейронних мереж і враховуючи в комплексі усі отримані переваги та недоліки рекомендованою для встановлення у вуличних системах відеоспостереження можна вважати нейронну мережу YOLO. Перевіримо далі в межах практичного експерименту, як ця мережа функціонує коли маємо справу з динамічною секвенцією.

4.3.2 Аналіз секвенції кадрів з камери відеоспостереження

Для перевірки нейронних мереж в динамічному режимі роботи камери розглянемо секвенцію кадрів у кількості 84, яку отримано з частотою зйомки 24 FPS, формат зображення HD_720 (див. п.4.1). В якості тестування оберемо дві нейронні мережі RetinaNet, YOLO v.3. Умови аналізу секвенції такі самі, які і для статичних зображень. Додатково перевіримо швидкість роботи обраних нейронних мереж. Для аналізу виберемо два об'єкти – автомобільні транспортні засоби білого та червоного кольорів (рис.4.51). Зазначимо, що оскільки секвенція визначає неперервну зйомки в динаміці, і зазначені об'єкти переміщаються з певною швидкістю, то не на всіх кадрах можливе тестування обраних нейронних мереж (є кадри, де обрані об'єкти повністю затіняються іншими транспортними засобами). Крім цього, слід враховувати, що в кадрі через швидкість об'єктів є певне розмиття контурів останніх.

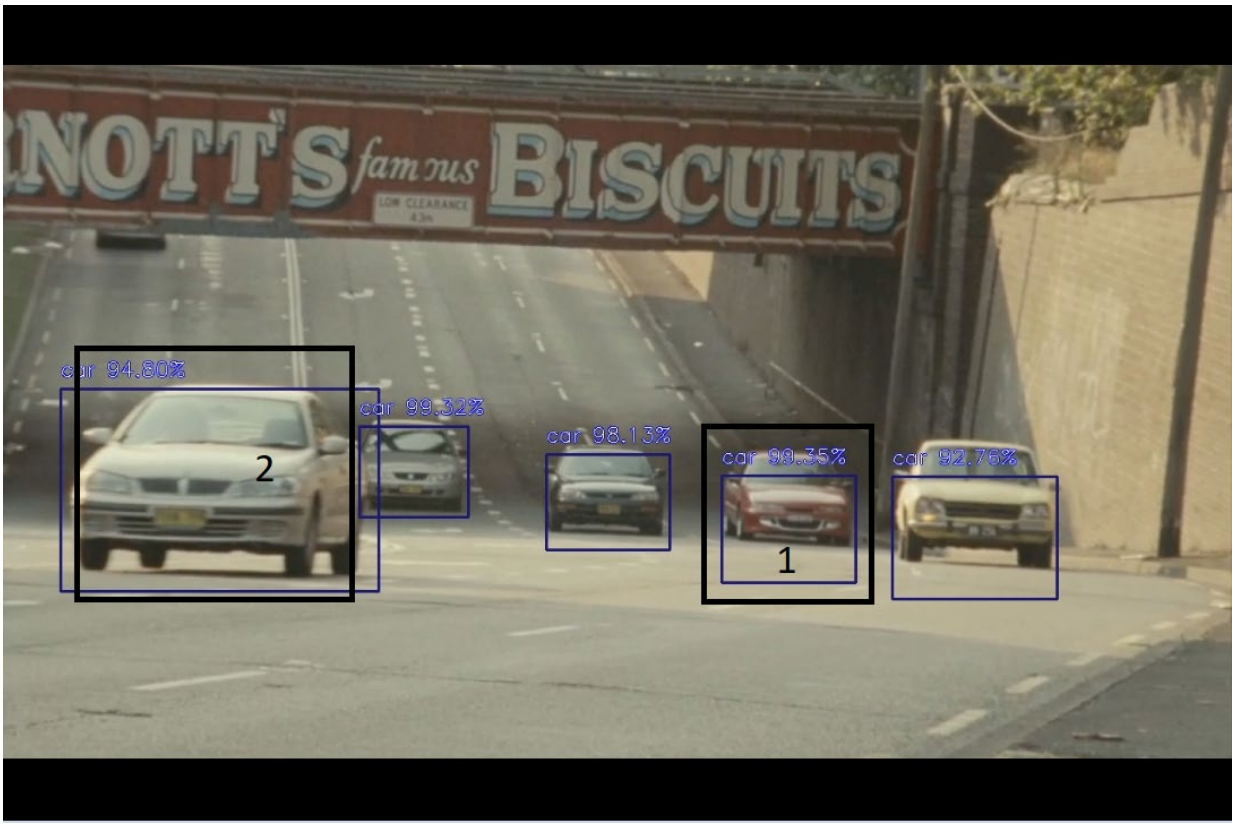


Рисунок 4.51 – Обрані об'єкти відеофайлу

За результатами тестування нейронних мереж отримано графіки окремо для об'єкту 1 та об'єкту 2 (див. рис.4.52 та рис.4.53). За віссю OX визначено

номер кадру секвенції, вісь OY – точність прогнозування об’єкту нейронною мережею заданого типу.

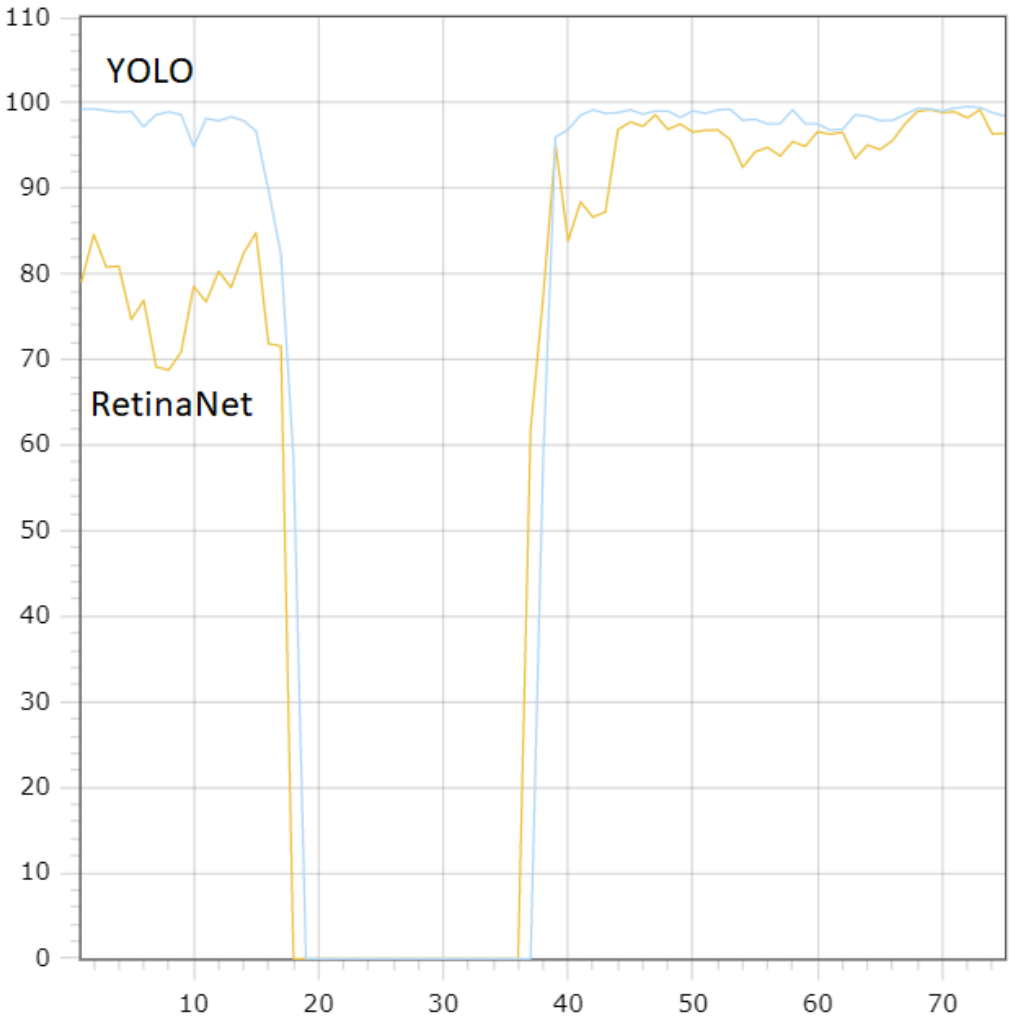


Рисунок 4.52 – Отримані результати для об’єкту 1

З аналізу побудованої кривої (рис.4.52) для об’єкту 1 в областях, де об’єкт можна проаналізувати отримано що точність роботи мережі YOLO є вищою у порівнянні з мережею RetinaNet. Причому, у проміжку від 1 кадру до 20 кадру об’єкт 1 знаходиться віддалено від камери, тобто для віддалених об’єктів, які визначають зменшеними формами нейронна мережа YOLO демонструє кращі результати.

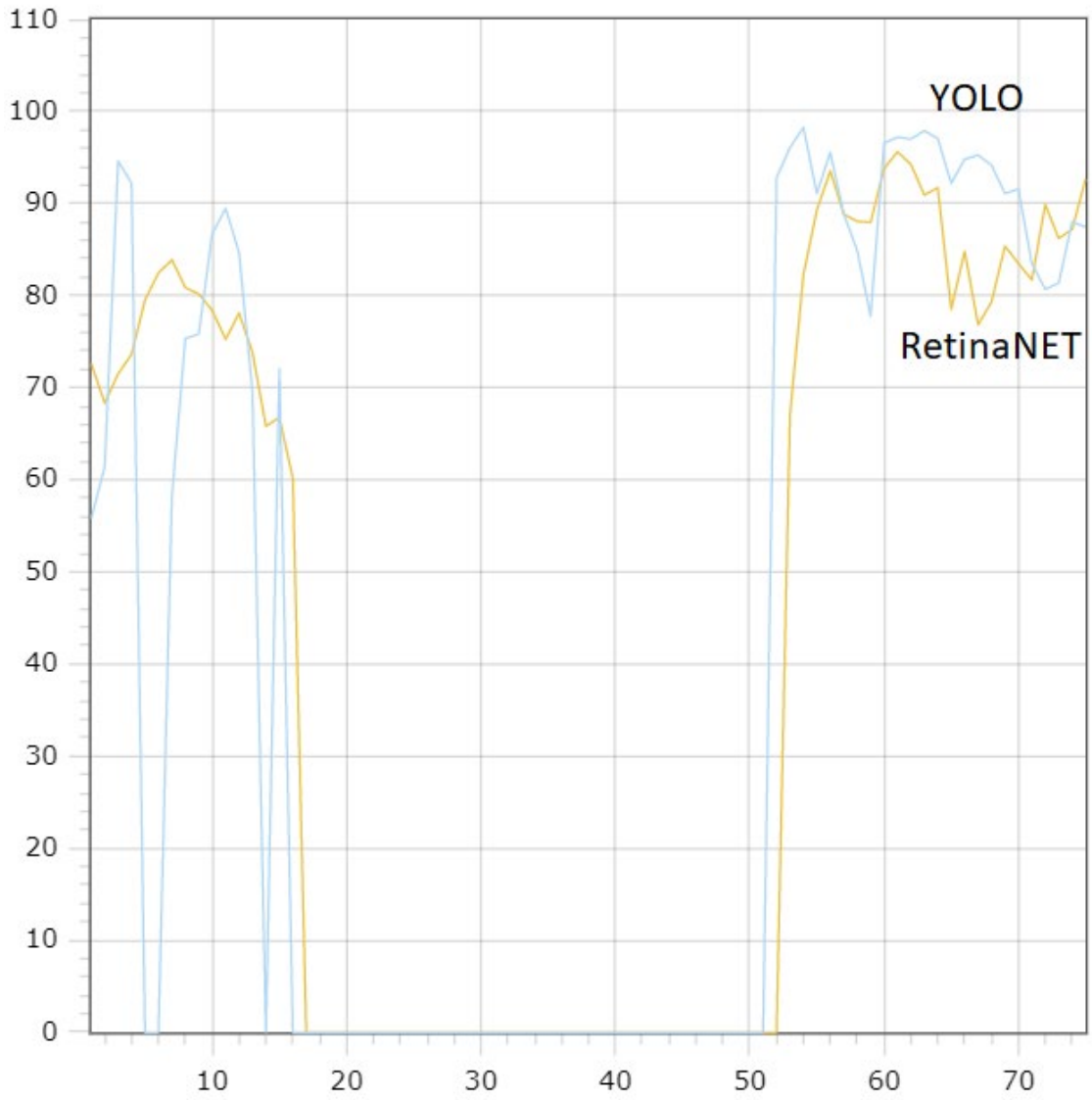


Рисунок 4.53 – Отримані результати для об’єкту 2

В інтервалі починаючи з 37 кадру точність розпізнавання починає зростати і наприклад на 63 кадрі вона для об’єкту 1 в межах моделі RetinaNet є більшою за 75 відсотків (рис.4.54).

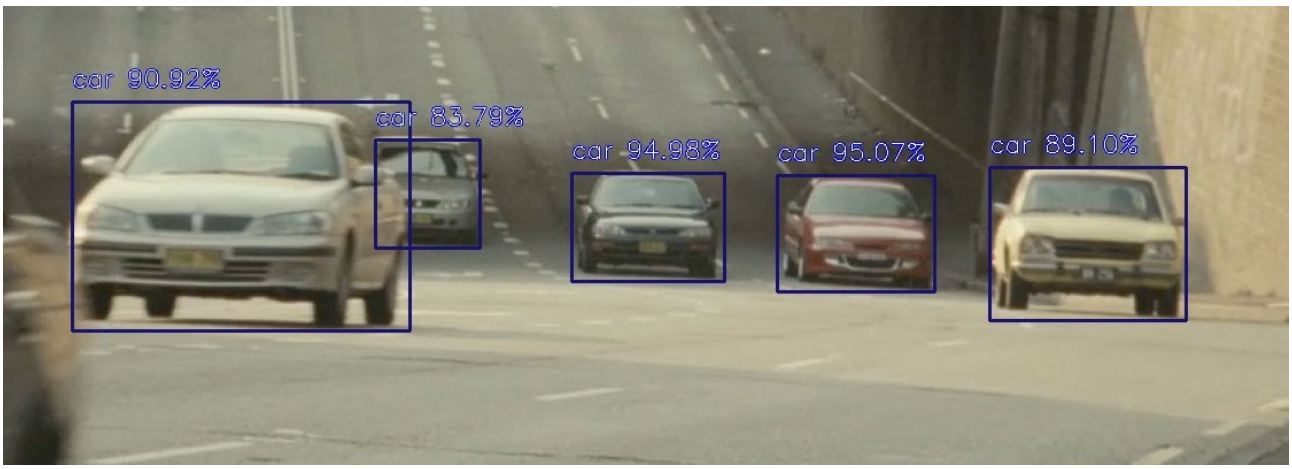


Рисунок 4.54 – Результат моделі RetinaNet (63 кадр)

Крім цього, можна відмітити що при роботі моделі YOLO через значні перетини об’єктів спостерігаються помилки з визначення класу об’єктів. Наприклад за умовами експерименту в секвенції рухаються легкові транспортні засоби і в 29 кадрі нейронна мережа визначає що в класі є вантажний автомобіль з точністю майже 64% (рис.4.55).

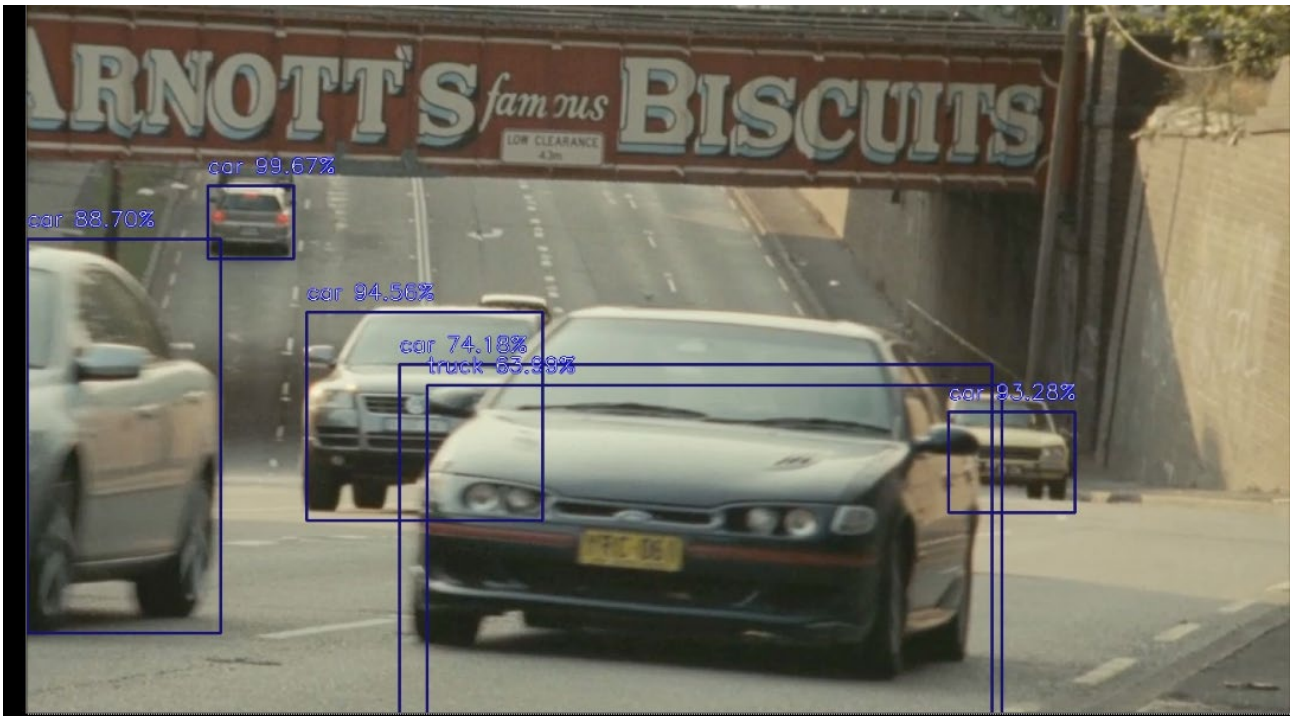
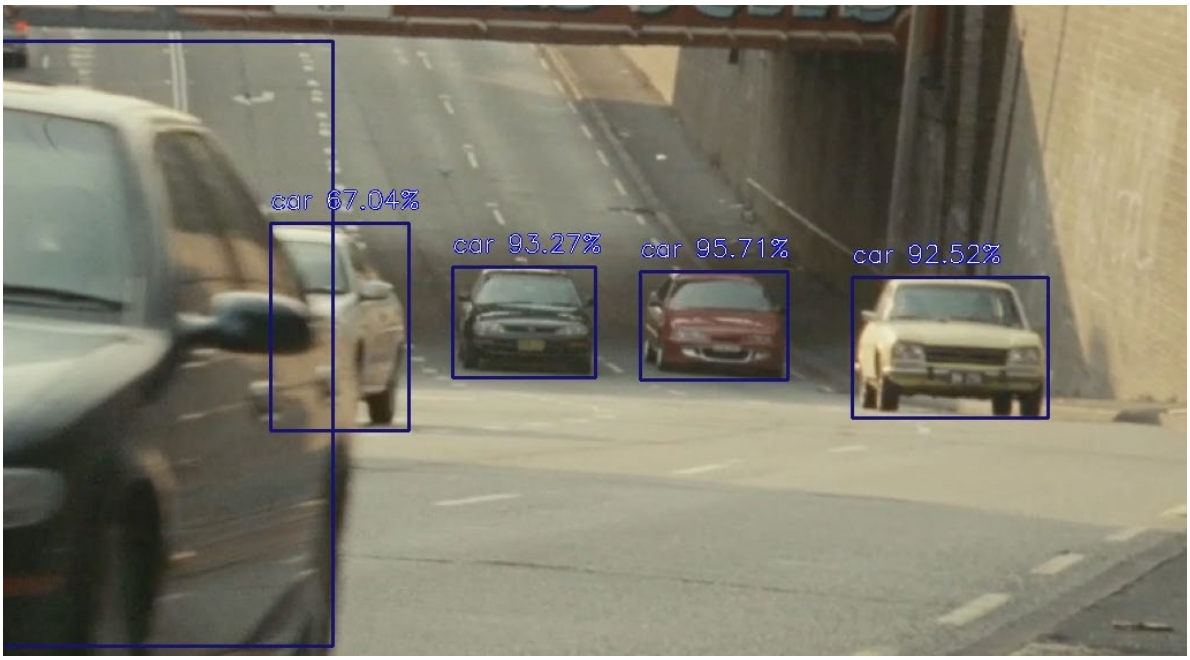
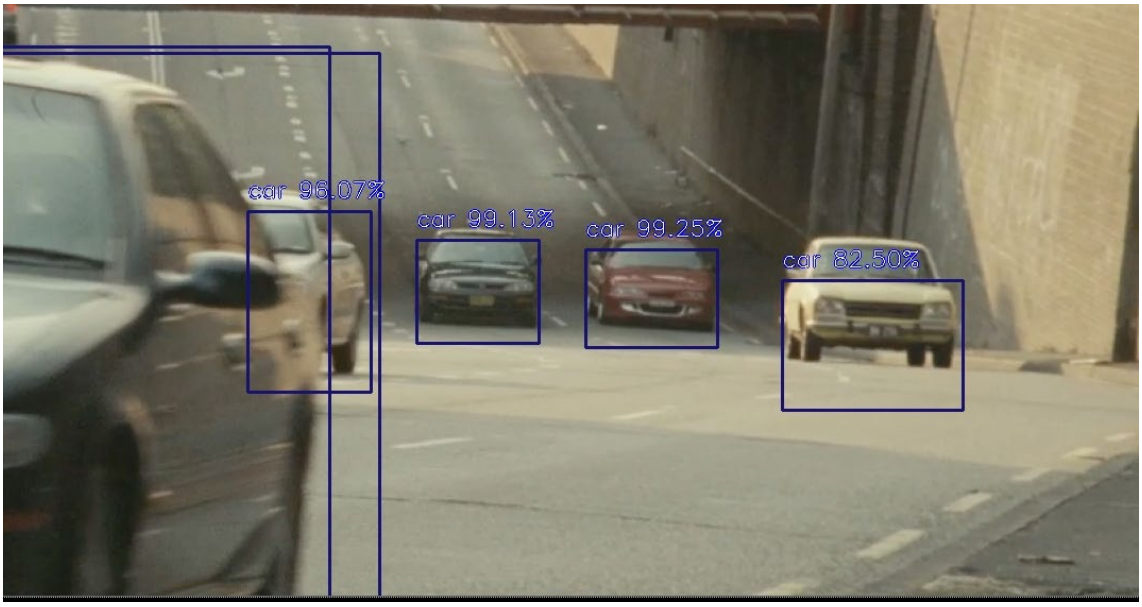


Рисунок 4.55 – Результат моделі YOLO (29 кадр)

Об’єкт 2 починаючи з 53 кадру для моделі RetinaNet має 67,04% точності і для моделі YOLO (рис.4.56) 96,07% точності. При цьому слід зауважити що обраний об’єкт 2 візуалізується не повністю і затіняється стороннім автомобілем (ускладнені умови розпізнавання). Далі, виходячи з побудованих кривих з рисунку 4.53 для об’єкту 2 у проміжку починаючи з 56 кадру ця тенденція вищих значень за точністю для мережі YOLO над значеннями мережі RetinaNet повторюється.



а)



б)

Рисунок 4.56 – Результат роботи нейронних мереж (53 кадр секвенції):
а) – модель RetinaNet; б) - модель YOLO

Крім цього, в рамках дослідження для динамічної секвенції було зафіксовано час (секундомір), який витрачається на аналіз 1 кадру секвенції при роботі нейронної мережі. Знайдено, що за однакових вихідних умов розпізнавання 1 кадру мережею RetinaNet займає приблизно 20-25 сек (час на виконання скрипту програми), тоді як при роботі мережі YOLO ця величина не

перевищує значення у 10 сек. Іншими словами, підтверджено тезу з п.4.1.2 дисертації, що мережа YOLO працює швидше у порівнянні з іншими розглянутими нейронними мережами, зокрема при порівнянні з мережею RetinaNet.

4.3.3 Перевірка моделі нейронної мережі до функції навчання

В цій частині дослідження перевіримо моделі двох нейронних мереж RetinaNet та YOLO на здатність їх до навчання. Для цього використаємо послідовність кадрів з рухомими транспортними засобами (п.4.3.2). В якості об’єкту дослідження розглянемо секвенцію з 25 кадру і прослідкуємо як моделі поводять себе при аналізі об’єкту Car (див. рис.4.57). Такий вибір об’єкту насамперед обумовлений тим, що на початку даний об’єкт розпізнавання не повністю відображається і точність не є високою, але далі транспортний засіб Car висувається на передній план (починаючи з 33 кадру секвенції) і нейронна мережа, якщо має ознаки навчання то результати визначення помилки неправильної класифікації повинні зменшуватись. Відмітимо, що починаючи з 33 кадру обраний об’єкт немає перетинів з іншими засобами та рух його незначний.

Рисунок 4.57 – 25-й кадр послідовності та вибір об’єкту дослідження

Спочатку дослідимо модель нейронної мережі RetinaNet. Результати роботи цієї нейронної моделі по-кадрово по відношенню до об’єкту Car наведено в таблиці 4.1.

Таблиця 4.1 – Точність визначення класу об’єкту нейронною мережею RetinaNet

Номер кадру	Точність визначення класу об’єкту, %	Номер кадру	Точність визначення класу об’єкту, %
25	85,62	37	85,69
26	85,79	38	87,77
27	87,99	39	92,15
28	83,98	40	92,77
29	82,80	41	92,15
30	80,09	42	93,68
31	83,47	43	94,61
32	84,56	44	93,71
33	80,50	45	93,54
34	84,92	46	94,89
35	82,43	47	96,36
36	83,72		

Аналізуючи отримані дані з таблиці 4.1 можна відмітити наступне. У проміжку між 25 кадром до 32 кадру модель незважаючи на неповну візуалізацію обраного об’єкту (перетин контуру з іншим об’єктом) намагається покращити точність оцінки. При цьому, мінімальна точність спостерігається на 30 кадрі, а далі модель намагається скорегувати отриману помилку у сторону зменшення. При цьому відмітимо, що процес покращення результатів відбувається декількома ітераціями. Починаючи з 35 кадру можна помітити, що точність починає зростати і це не залежить від параметрів контуру, адже починаючи з 33 кадру об’єкт Car нічим не затіняється. Так, можна заперечити,

що 33 кадр визначається значенням 80,50% але це напевно, ще вплив стороннього близького об’єкту. Таким чином, аналізуючи дані з 38 кадру по 47 кадр точність збільшується і система проходить процедуру навчання з вчителем. Крім цього, варто відзначити за отриманими даними, що в основі роботи моделі є метод градієнтного спуску, адже точність змінюється наче стрибками (це можна побачити у проміжку між 28 кадром і 34 кадром).

Розглянемо тепер як поводиться при аналізі об’єкту Car нейронна мережа YOLO. Результати роботи цієї моделі представлено у таблиці 4.2.

Таблиця 4.2 – Точність визначення класу об’єкту нейронною мережею YOLO

Номер кадру	Точність визначення класу об’єкту, %	Номер кадру	Точність визначення класу об’єкту, %
25	96,32	37	82,81
26	98,20	38	79,45
27	96.93	39	80,26
28	94,81	40	78,54
29	93,28	41	81,18
30	85,71	42	79,45
31	89,21	43	84,44
32	91,27	44	85,74
33	91,76	45	85,95
34	85,06	46	83,67
35	69,96	47	84,21
36	79,72		

Аналізуючи дані з таблиці 4.2 по-перше слід зазначити що при роботі мережі на 35 та 36 кадрах модель відпрацювала так, ніби в даній області був не один об’єкт а два – Car, Truck. Крім цього, отримані результати дозволяють стверджувати, що дана модель нейронної мережі функціонує в режимі без навчання а основний метод, виходячи зі стрибків отриманих величин теж є

метод градієнтного спуску. Навіть, в даному випадку не можна говорити про режим самонавчання, адже після високих результатів на 37 кадрі далі точність зменшується на окремих кадрах, навіть ще менше (40 та 42 кадр) у порівнянні з 36 кадром.

Висновки до 4 розділу. В 4 розділі дисертації наведено практичний експеримент з перевірки можливостей трьох нейронних багатошарових мереж для різного візуального контенту, який зібрано з вуличних камер відеоспостереження. Зокрема, сформульовано детальний алгоритм перевірки нейронної мережі, визначено теоретичні засади функціонування мережі з зазначенням та подальшої перевіркою певних практичних особливостей. На основі отриманих результатів надані рекомендації з використання нейронних мереж для електронних систем відеоспостереження. Зокрема, за переліком визначених переваг та недоліків знайдено, що найкращі результати роботи за визначенням класу об’єкту та його локалізації видає нейронна мережа YOLO, за умови відсутності попереднього навчання.

ВИСНОВКИ

Наукове дослідження, яке розглянуто в рамках магістерської дисертації присвячено аналізу методів і засобів розпізнавання елементів на кольоровому цифровому зображенні чи секвенції зображень. Серед отриманих результатів у теоретичній і практичній частинах роботи варто відзначити наступні.

1. На початку роботи виконано розгляд ознак, для систем розпізнавання образів. Причому, основна увага приділяється саме геометричним ознакам контурів об’єктів. Такий вибір обумовлений тою обставиною, що морфологічний і семантичний аналіз зображень у програмних середовищах Matlab та Labview найбільш просто реалізувати саме за таким підходом. Серед розглянутих методів опису контуру об’єктів відмічено, що сам метод, його ефективність в системах відеоспостереження може залежати від певних чинників. Наприклад, на вибір методу для розпізнавання зображення впливає не лише характер самого зображення (складність фігур), але й якість цього зображення при аналізі (наявність спотворень). В першому випадку, коли зображення визначається великою кількістю різних образів, недоцільно використовувати методи сплайн-апроксимації. Оскільки в даному випадку значно збільшується масив обробки даних. У другому випадку, коли в зображенні є спотворення не рекомендовано використовувати метод ланцюгового кодування, оскільки точність в такому випадку буде не задовільною. У другій частині першого розділу дисертації представлено основні типи систем розпізнавання. Серед розглянутих типів варто відзначити системи без навчання та системи з вчителем, оскільки висвітлені принципи можна використати при створенні системи розпізнавання елементів зображення на основі підключеної нейронної мережі. Зокрема, система без навчання може бути використана при аналізі статичних зображень. А система з вчителем може бути корисною при роботі з секвенцією кадрів зображення.

2. З'ясовано, що існуючі підходи до реалізації класифікатора розпізнавання образів не можуть у повній мірі забезпечити процедуру точного елементів на площині зображення. Це пояснюється тим, що при аналізі

зображень необхідно мати додаткову інформацію про саме зображення. Наприклад, принцип кластеризації передбачає, що для зображення необхідно побудувати простір ознак, де сам образ складається як вектор з *n*-ма координатами. Зрозуміло, що цей образ буде штучним, адже при зміні характеру зображення, простір доведеться будувати наново. Тобто, для автоматичного режиму розпізнавання з адаптацією цей підхід не є зручним. За результатами проведеного огляду визначено передумови, які дозволяють стверджувати, що найбільш перспективним та ефективним підходом до розпізнавання об'єктів на сьогодні є використання нейронних мереж. Іншими словами, при використанні нейронних мереж, вже не виникає питання щодо обмеження кількості класів класифікації образів на площині зображення. До того ж, в даному випадку у користувача при створенні системи відеоспостереження з'являється можливість пропускати систему через режим навчання. А це, у свою чергу, буде призводити до зменшення помилок неправильної класифікації, що є позитивним моментом.

3. Визначено основні проблеми, які можуть виникнути при розпізнаванні об'єктів зображення. Так, наприклад, це може бути неможливість сформувати набір ознак об'єкту, наявність майже повного перетину різних образів об'єктів на площині зображення. Для мінімізації окреслених проблем запропоновано використання нейронної мережі. Окремо, для розуміння принципів роботи цієї мережі, побудовано аналогію з типовою апаратно-програмною системою, де основними елементами, які впливають на якість роботи, зазначено арифметично-логічний пристрій і пам'ять. В заключній частині третього розділу дослідження сформульовані головні переваги нейронних мереж для систем розпізнавання об'єктів. Тут варто відзначити серед іншого, що така мережа має риси однорідності, надійності та має ознаки паралельності виконуваних операцій. А це, в свою чергу, дозволяє в результаті забезпечити високу точність розпізнавання, швидкість аналізу кольорового зображення та корекцію помилок неправильної класифікації. Що і буде перевірено у практичній частині дисертації.

4. У практичній частині дослідження дисертації було обрано для аналізу три найбільш поширені у світі нейронні мережі RetinaNet, YOLO v.3, TinyYOLO, які використовуються в системах розпізнавання образів і можуть бути обрані в якості основних електронних систем машинного читання у приладах електронного відеоспостереження при створенні систем контролю за дорожнім рухом. На початку розділу були сформульовані вихідні дані до експерименту і тут слід зазначити про однакові умови для тестування. Тобто, перевірка нейронних мереж проводилась на одноковому обладнанні на основі однакового візуального. За результатами проведених випробувань на основі аналізу 11 зображень були окремо сформульовані висновки і виявилось, що для статичних зображень рекомендованою для встановлення у вуличних системах відеоспостереження можна вважати нейронну мережу YOLO. Додатково було проведено експеримент перевірки нейронних мереж в динамічному режимі роботи камери. З'ясувалось, що серед двох нейронних мереж RetinaNet, YOLO v.3 в середньому точність роботи мережі YOLO є вищою у порівнянні з мережею RetinaNet. Причому, у проміжку від 1 кадру до 20 кадру об'єкт 1 знаходиться віддалено від камери, тобто для віддалених об'єктів, які визначають зменшеними формами нейронна мережа YOLO демонструє кращі результати. Так, в інтервалі починаючи з 37 кадру точність розпізнавання починає зростати і наприклад на 63 кадрі вона для об'єкту 1 в межах моделі RetinaNet є більшою за 75 відсотків. Але разом з тим, можна відмітити, що при роботі моделі YOLO через значні перетини об'єктів спостерігаються помилки з визначення класу об'єктів. Наприклад за умовами експерименту в секвенції рухаються легкові транспортні засоби і в 29 кадрі нейронна мережа визначає що в класі є вантажний автомобіль з точністю майже 64%. За результатами швидкості роботи мереж для динамічної секвенції знайдено, що за однакових вихідних умов розпізнавання 1 кадру мережею RetinaNet займає приблизно 20-25 сек (час на виконання алгоритму програми), тоді як при роботі мережі YOLO ця величина не перевищує значення у 10 сек. Тобто, отримано підтвердження з теорії, що мережа YOLO працює швидше у порівнянні з іншими розглянутими нейронними мережами, зокрема при порівнянні з мережею RetinaNet.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Молодяков С. А. Проектирование специализированных цифровых видеокамер. Санкт-Петербург : Изд-во Политехн. ун-та, 2016. 286 с.

2. Коломиец Э. И. Методы распознавания образов и анализа изображений, Самара 2003

3. Лепский А. Е., Броневич А. Г. Математические методы распознавания образов. Таганрог : Изд-во ТТИ ЮФУ, 2009. 155 с.

4. Местецкий Л. М. Математические методы распознавания образов. Москва : МГУ, ВМиК, 2004. 85 с.

5. Чабан Л. Н. Методы и алгоритмы распознавания образов в автоматизированном дешифрировании данных дистанционного зондирования. Москва : МИИГАиК, 2017. 94 с.

6. Адзерихо К. С., Киселевский Л. И., Костюкевич С. Б., Краснопрошин В. В. Физические основы дистанционного зондирования. – Минский ун-т, 1991

7. Шитиков В. К., Розенберг Г. С., Зинченко Т. Д. Методы системной идентификации. Тольятти : ИЭВБ РА, 2003. 463 с.

8. Богатиков В. Н., Дранишников Л. В., Пророков А. Е. Построения систем управления на основе нейронных сетей. Апатиты : Изд-во КФ ПетрГУ, 2011. 41 с.

9. Axons: the cable transmission of neurons. URL: https://qbi.uq.edu.au/brain/brain-anatomy/axons-cable-transmission-neurons (дата звернення: 10.03.2021)

10. Искусственный интеллект. URL: http://ilab.xmedtest.net/?q=node/6327&page=2 (дата звернення: 08.04.2021)

11. Tan M., Le Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 2019. URL: arxiv.org/abs/1905.11946 (дата зверення 24.03.2021)

12. Zeng N. RetinaNet Explained and Demystified. 2018 URL: blog.zenggyu.com/en/post/2018-12-05/retinanet-explained-and-demystified (дата звернення 19.03.2021)

13. Review: RetinaNet — Focal Loss (Object Detection). 2019 URL: towardsdatascience.com/review-retinanet-focal-loss-object-detection-38fba6afabe4 (дата звернення 4.04.2021)

14. Tsung-Yi Lin Focal Loss for Dense Object Detection. 2017. URL: arxiv.org/abs/1708.02002 (дата звернення 1.04.2021)

15. The intuition behind RetinaNet. 2018 URL: medium.com/@14prakash/the-intuition-behind-retinanet-eb636755607d (дата звернення 29.03.2021)

16. Распознавание объектов в режиме реального времени на iOS с помощью YOLOv3. URL: https://habr.com/ru/post/460869/ (дата звернення: 1.28.2021)

17. Распознавание объектов с помощью YOLO v3. URL: https://proglib.io/p/raspoznavanie-obektov-s-pomoshchyu-yolo-v3-na-tensorflow-2-0-2020-11-08 (дата звернення: 29.03.2021)

18. Борисов Г. О., Трапезон О. Г., Трапезон К. О. Особливості оброблення сигналів із застосуванням програмних засобів IOS. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки*. Київ, 2019. Том. 30 (69), №6, Частина 1. С. 22-26. DOI:https://doi.org/10.32838/2663-5941/2019.6-1/04

19. Борисов Г. О., Гумен Т. Ф., Трапезон К. О. Дослідження програмних особливостей об’єднання Android things на основі концепції Інтернету речей. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки*. Київ, 2020. Том. 31 (70), №1. С. 29-35. DOI:https://doi.org/10.32838/2663-5941/2020.1-1/06

ДОДАТОК А

ABSTRACT

In the practical part of the dissertation research we will consider the three most common neural networks in the world RetinaNet, YOLO v.3, TinyYOLO, which are used in image recognition systems and can be selected as the main electronic machine reading systems in electronic video surveillance devices when creating traffic control systems. . Note that as a purity of the experiment were set up the neural system under the same initial conditions. Testing of systems, checking the possibility of their deep learning took place under the condition of having the same visual material. The choice of this material is determined by the dissertation and is aimed primarily at identifying technical features that impair or improve the neural network as a whole and which must be taken into account in the design of prototypes of pattern recognition systems with learning elements.

Two types of visual content were selected as input data to test the three selected neural networks - a set of 11 static images depicting various road situations and a sequence of frames from a film that identifies various dynamic situations that affect the quality of the neural network. as a component of object recognition system. In particular, such situations include the speed of cars, the distance of the subject from the camera lens, the lighting conditions of the stage, foreign objects of the scene, and so on. The default frequency is 24 frames per second. The total number of frames in the sequence is 84 frames, the image format is HD_720 (1280 pixels × 720 pixels), the color system is sRGB; graphic image format - JPEG.

The choice of static images for the experiment was primarily for the purpose of determining how neural networks work, whether they are capable of learning and what method or method is used in the basis of their operation. In addition, we note that the download of both still images and dynamic sequence took place on the same graphics station under the same conditions (laptop Dell G3 15, 6 core Intel Core i7 8 generation, 32 GB of RAM, discrete video card with 4 GB of video memory). The graphics station supports CUDA technology. Additionally, the speed of reading and recognizing objects in the image was tested under the same algorithm and the same connected libraries. The script and algorithm for its creation in Python in PyCharm will also be presented in the practical part of the study.

The algorithm according to which the experiment will be conducted should be divided into a sequence of the following actions: selection from the image with characteristic features; download the library of neural network objects, start the network for image processing, evaluate the accuracy of the network. In this case, for clarity of the study from each image among the obtained results for three neural networks will be selected three identical objects (for the sequence of frames - two). The results will be summarized in the appropriate graphical illustrations in the form of columns, and the color of each column for convenience will determine a certain type of neural network. Before conducting the experiment, we first present certain characteristics of these neural networks, which are determined by the developers in order to test them directly in practice as part of the study. Next, we present an algorithm for creating a research model and then proceed to the analysis of the results.

The architecture of the RetinaNet neural network consists of 4 links, each of which performs certain functions in the general procedure of recognizing objects in the image. Figure 4.1 shows the architecture of this network.

The first link of the network is called "backbone" and it is the main element of the network, which is used to find and capture features from the image that is the subject of analysis. This part of the network has a variable structure and may be based on other classification networks, such as ResNet, EfficientNet.

On the first link on the basis of a variable network the analysis of the image with allocation of important signs is carried out and this received information will be further used in a network both for training, and for creation of the classification table. Therefore, the choice of variable structure must be chosen so that the model has a high accuracy of detection and classification of features in the image palette.

The second link of the architecture is called Feature Pyramid Net (FPN) and the structure is a map of characteristics of different scales at some levels in the form of a pyramid. This link in the form of a convolutional subnet is used to combine the benefits of feature maps at the lower and upper levels of the network. The lower level (layer) has a high resolution and low semantic ability. The upper level (layer) on the contrary - low resolution, but high semantic ability (value). In fact, on this link of the

neural network we have a hierarchical pyramid - a sequence of levels (layers) with a gradual decrease in the resolution of the image being analyzed. Thus signs passing through the analysis with various resolution the network receives from the first link. The top layer is called the bottom-up pathway, the bottom layer is called the top-down pathway, and there are lateral connections. In our case at this link the analysis of the image passes in two ways from the top area to the bottom and then on the contrary from the bottom to the top. At the upper level, image details may be lost, for example due to the high noise level of the original image. At the lower level, which is typical for obtaining a high resolution, the feature maps are doubled based on the method of "nearest neighbor" in the downward direction.

Due to the presence of side connections, the maps of the features of the corresponding pyramids are added element by element, and the maps from the upper level also pass through the convolution of 1 × 1 conv. Thus, this approach to the work of the link on the basis of two paths allows you to combine semantically important information obtained at the end of the first pyramid, and more detailed information obtained earlier. The resulting information goes to the third link RetinaNet, which consists of two subnets, called classification (Classification Subnet) and regression, respectively (Regression Subnet). The classification subnet with FPN defines information about the classes of recognition objects, thus solving the classification problem. The regression subnet, in turn, removes from the feature map information about the location of objects, their coordinates in the image. From Figure 4.1 it can be noted that these two subnets differ only in the last layer. Thus, on layer 5, the number of feature maps changes - the regression model has 4A feature maps, while the classification model is marked only by KA by the number of feature maps, where A is the number of anchor frames, K is the number of object classes by classification. In the sixth layer, each feature map is transformed into a set of vectors. Anchor box - a parameter of neural networks, in the form of a detector of the rectangle relative to which the network operates.

When training a RetinaNet neural network, a corresponding anchor frame from the set is selected for each target frame. If the match rate (parameter IoU, see clause 4.1.2) is more than 50 percent, the anchor frame is considered the target, and if less

than this value, it is considered that the frame is considered the background and is no longer used for network training. The classification subnet learns about the purpose (object class or background), the regression subnet about the anchor frame coordinates. RetinaNet error consists of two values - regression error (localization) and classification error. Knowledge of the regression error is determined in the network through 4 numbers - Posix, Posiy, Posiw, Posih. The first two numbers determine the predicted difference between the coordinates of the centers for the anchor and target frames, while the third and fourth coordinates determine the predicted difference between the height and width of the two frames. As a result, the corresponding difference with the anchor is calculated for each target frame. The classification error is calculated through a special function, which includes the probability of predicting the class, the purpose of the class, the number of classes and the bias factor. This function is a function of cross-entropy (entropy loss function). When learning a network, most objects are considered a background classifier, which is displayed in a separate class. And theoretically, the network can determine the background better than the required recognition objects.

YOLO neural network (You Only Look Once) based on its name differs from other neural networks in that in the process of image analysis, this network is used once immediately before the analysis of the entire image plane. That is, the neural network divides the image into a kind of grid with the same cells and predicts a kind of bounding box by the fact that it is in this area with a certain probability is the desired object for each part of the image.

The advantage of this approach is that when analyzing the entire image, the network immediately takes into account the full information context in detection and recognition. That is, unlike the RetinaNet network there is no two-way image analysis, this network should work much faster. For example, according to developer Joseph Redmon, it is claimed that the basic network model processes images in real time at a speed of 45 frames per second. At the same time, it is known that this network is difficult to locate objects with high accuracy, but this is without the condition of learning. These features will be tested below in the study by running a timer when analyzing the same image by different neural networks.

In other words, each cell is responsible for predicting a certain class of object and determines the confidence indicator for it. This metric is the mathematical probability that a given cell contains the object you are looking for. As a result of such analysis we receive a map of all objects and a set of frames with probabilities through a confidence indicator. In addition, each cell is responsible for predicting the probabilities of specific classes of image objects that the neural network is used to search for.

In the YOLO network, object recognition is implemented as the only regression task to separate constraints, which are associated with the probabilities of belonging to different classes of objects in the image. That is, the direct relationship between image pixels and frame coordinates and object class prediction probabilities is traced. Anchor boxes are used to predict frames for the contents of objects.

TinyYOLO This is a simplified version of the above-mentioned YOLO architecture and it consists of a smaller number of output layers. There are only two of them for this implementation, in contrast to the architecture of the YOLO network. Because of this, as shown by numerous experiments, this network performs worse recognition of small details of the image, but it functions faster than its full version. For example, when analyzing video from an iPhone XS mobile phone, the speed of the TinyYOLO network averaged 30-32 frames per second. Whereas for the YOLO network, this value was in the range of 23-25 frames per second. In addition, it is logical to conclude that this version will have a lower accuracy of object recognition. Let's test these features of the network further during the experiment on the analysis of static images of objects from highways.