

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТАВРІЙСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ  
В.І.ВЕРНАДСЬКОГО  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ МУНІЦИПАЛЬНОГО  
УПРАВЛІННЯ ТА МІСЬКОГО ГОСПОДАРСТВА  
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Оцінка після захисту роботи в ЕК:

Національна шкала \_\_\_\_\_

Кількість балів \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до бакалаврської кваліфікаційної роботи освітнього ступеня **“бакалавр”**

з галузі знань 12 «Інформаційні технології»

спеціальності 122 «Комп'ютерні науки»

на тему: Система пошуку та детекції об'єктів на відео

Студента групи КН – 41 Капітан Владислав Володимирович \_\_\_\_\_  
(шифр групи) (прізвище, ім'я, по батькові) (підпис)

Керівник роботи д.т.н. професор Дорошенко Ю.О. \_\_\_\_\_  
(вчені ступінь та звання, прізвище, ініціали) (підпис)

**Консультанти:**  
охорона праці та навко-  
лишнього середовища доцент Гуйда О.Г. \_\_\_\_\_  
(вчені ступінь та звання, прізвище, ініціали) (підпис)

**Київ – 2024**

**ТАВРІЙСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ  
В.І.ВЕРНАДСЬКОГО**

Навчально-науковий інститут муніципального управління та міського  
господарства

Перший (бакалаврський) освітній рівень

Галузь знань 12 «Інформаційні технології»

(шифр і назва)

Спеціальність 122 «Комп'ютерні науки»

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр ГУЙДА

“ \_\_\_ ” \_\_\_\_\_ 2023 р.

**З А В Д А Н Н Я  
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ**

студента

Капітана Владислава Володимировича

(прізвище, ім'я, по батькові)

**1 Тема роботи:** Система пошуку та детекції об'єктів на відео

керівник роботи

д.т.н. професор Дорошенко Ю.О.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені ректором Університету від “ 06” листопада 2023 року

**2 Строк подання студентом роботи “31” травня 2024 р.**

**3 Вихідні дані до роботи**

Комп'ютерні інформаційні технології в системах детекції об'єктів на відео

**4 Зміст розрахунково-пояснювальної записки:**

- 4.1 Виконати аналіз літератури по темі систем пошуку та детекції об'єктів.
- 4.2 Визначити актуальні проблеми систем пошуку та детекції.
- 4.3 Вибрати методи та алгоритми для реалізації системи.
- 4.4 Розробити систему для детекції об'єктів на відео на основі обраних методів.
- 4.5 Пропрацювати охорону праці та навколишнього середовища.

## 5 Перелік графічного матеріалу

Графічна робота виконана у вигляді мультимедійної презентації

## 6 Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Загальна частина	професор Юрій ДОРОШЕНКО		
Технологічна частина	професор Юрій ДОРОШЕНКО		
Спеціальна частина	професор Юрій ДОРОШЕНКО		
Охорона праці та навколишнього середовища	доцент Олександр ГУЙДА		
Графічна частина	ст. викладач Олена ФУРТАТ		

7 Дата видачі завдання «7» листопада 2023 року

## КАЛЕНДАРНИЙ ПЛАН

Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
<i>Загальна частина</i>	<i>березень</i>	
<i>Технологічна частина</i>	<i>квітень</i>	
<i>Спеціальна частина</i>	<i>травень</i>	
<i>Охорона праці та навколишнього середовища</i>	<i>травень</i>	
<i>Графічна частина</i>	<i>травень</i>	

Студент \_\_\_\_\_  
( підпис )

Керівник роботи \_\_\_\_\_  
( підпис )

Владислав КАПТАН  
(прізвище та ініціали)

Юрій ДОРОШЕНКО  
(прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до БКР: робота містить 72 с., 21 рисунок, 21 джерело.

Метою роботи є розробка та апробація працездатності системи пошуку та детекції об'єктів на відео з використанням сучасних методів та алгоритмів комп'ютерного зору. В роботі застосовано методи згорткових нейронних мереж, зокрема алгоритм YOLO v8, для забезпечення високої точності та швидкості обробки відеоданих в реальному часі.

В процесі дослідження виконано аналіз існуючих підходів до пошуку та детекції об'єктів на відео, вибір оптимального методу для реалізації системи, програмну реалізацію системи з використанням JavaScript та HTML5 Canvas, а також проведення експериментальної апробації прототипа програми. Результати тестування показали, що розроблена система забезпечує високу точність детекції об'єктів навіть у складних умовах освітлення та наявності рухомого фону.

Рекомендації щодо використання розробленої системи включають її впровадження в системах відеоспостереження, розумних транспортних системах та інших застосуваннях, де необхідна обробка відеоданих в реальному часі. Результати дослідження можуть бути використані для подальшого вдосконалення алгоритмів детекції об'єктів та розробки нових систем на їх основі.

## ЗМІСТ

ВСТУП.....	6
1 ПЕРЕДУМОВИ ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ .....	8
1.1 Аналіз існуючих методів та алгоритмів .....	8
1.2 Основні проблеми існуючих методів .....	9
1.3 Вимоги до створюваного програмного прототипу .....	10
2 МЕТОДИ І АЛГОРИТМИ СИСТЕМ ПОШУКУ І ДЕТЕКЦІЇ ОБ'ЄКТІВ НА ВІДЕО .....	12
2.1 Огляд основних методів пошуку та детекції об'єктів на відео .....	12
2.2 Вибір методів та алгоритмів для розробки програмного прототипу .....	19
2.3 Опис обраного підходу та обґрунтування вибору .....	21
3 РОЗРОБКА ТА ЕКСПЕРИМЕНТАЛЬНА АПРОБАЦІЯ ПРОТОТИПУ ПРОГРАМИ.....	25
3.1 Використані технології та інструменти для реалізації програмного забезпечення .....	25
3.2 Опис архітектури системи.....	26
3.3 Опис реалізованих алгоритмів та характеристика їх роботи .....	30
3.4 Експериментальна апробація прототипу програми .....	43
3.5 Рекомендації щодо подальшого вдосконалення системи.....	58
4 ОХОРОНА ПРАЦІ ТА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....	62
4.1 Аналіз потенційних ризиків та загроз для користувачів програмної системи.....	62
4.2 Заходи з охорони праці та безпеки користування .....	63
4.3 Утилізація комп'ютерної техніки.....	65
ЗАГАЛЬНІ ВИСНОВКИ.....	69
ПЕРЕЛІК ПОСИЛАНЬ .....	71

					<b>КР.122.006.ПЗ</b>			
Зм	Арк.	№ докум.	Підп.	Дата	Система пошуку та детекції об'єктів на відео	Літ.	Аркуш	Аркушів
Розроб.	Капітан В.В.	Дорошенко Ю.О.				У	5	72
Перев.					Пояснювальна записка	ТНУ ім.В.І.Вернадського ННІМУМГ Гр. КН-41		
Н.конт	Фургат О.В.							
Затв.								

## ВСТУП

Системи пошуку та детекції об'єктів на відео знаходять застосування в широкому спектрі областей, включаючи охорону, безпеку та розумні транспортні системи. Зі зростанням обсягів відеоданих, згенерованих цифровими камерами та іншими джерелами, виникає критична потреба у вдосконаленні технологій, які можуть ефективно аналізувати ці дані в реальному часі. Сучасні системи мають здатність адаптуватися до різних умов освітлення і погоди, забезпечуючи точність і швидкість обробки при мінімальних обчислювальних витратах.

Основна мета дослідження полягає у створенні та впровадженні ефективної системи пошуку та детекції об'єктів на відео. Розробка такої системи передбачає реалізацію сучасних алгоритмів та методів обробки зображень, що дозволить забезпечити високу точність і швидкість обробки в різноманітних умовах використання. Особлива увага приділяється адаптації системи для роботи в реальному часі та забезпеченню її надійності та ефективності.

Для досягнення поставленої мети, дослідження включає наступні ключові завдання:

1) Аналіз існуючих методів та алгоритмів:

- Огляд сучасних підходів до пошуку та детекції об'єктів на відео.
- Визначення актуальних проблем та недоліків існуючих рішень.

2) Вибір методу для розробки системи:

- Аналіз та вибір найбільш ефективного алгоритму для реалізації системи детекції об'єктів на відео.
- Обґрунтування вибору алгоритму YOLO v8.

3) Розробка та впровадження системи:

- Реалізація системи детекції об'єктів на основі обраного алгоритму.
- Інтеграція системи у веб-середовище з використанням JavaScript та HTML5 Canvas.

					БКР 122.006. ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

- Оптимізація продуктивності системи для зменшення обчислювального навантаження та покращення часу відгуку.

4) Тестування та валідація:

- Проведення випробувань для оцінювання точності та швидкості реакції системи в різних умовах.

- Аналіз результатів тестування для виявлення можливих напрямків покращення та налаштування системи.

Об'єкт дослідження: процеси аналізу відеоданих та системи автоматичної детекції об'єктів на відео, які включають техніки обробки зображень та відеоаналітики.

Предмет роботи: процес розробки ефективної системи пошуку та детекції об'єктів на відео з акцентом на оптимізацію точності та швидкості обробки в різноманітних умовах використання.

					БКР 122.006. ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

# 1 ПЕРЕДУМОВИ ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

## 1.1 Аналіз існуючих методів та алгоритмів

Огляд сучасних підходів до пошуку та детекції об'єктів на відео показує, що існує кілька основних методів, які використовуються для цих завдань. Традиційні методи, такі як фонове видалення (Background Subtraction), оптичний потік (Optical Flow) і контурне виявлення (Edge Detection), ефективні в деяких сценаріях, але мають свої обмеження[12].

Фонове видалення базується на створенні моделі фону та виявленні рухомих об'єктів шляхом порівняння кожного нового кадру з цією моделлю. Цей метод добре працює в стабільних умовах освітлення, але може давати неточні результати при різких змінах освітлення або наявності рухомого фону.

Оптичний потік визначає рух об'єктів шляхом аналізу зміни інтенсивності пікселів між послідовними кадрами. Цей метод є ефективним для відстеження плавних рухів, але може давати неточні результати при швидкому або складному русі об'єктів.

Контурне виявлення використовує алгоритми, такі як Canny, для ідентифікації країв об'єктів на зображенні. Цей метод добре підходить для виділення чітких країв, але може бути обмежений у складних візуальних сценах з великою кількістю деталей.

Сучасні методи на основі штучного інтелекту, такі як каскадні класифікатори (Cascade Classifiers) і згорткові нейронні мережі (CNN), забезпечують значно вищу точність та швидкість обробки даних. Каскадні класифікатори, такі як алгоритм Viola-Jones для детекції облич, працюють швидко, але можуть бути обмежені у своїй ефективності залежно від типів об'єктів[12].

Згорткові нейронні мережі, включаючи архітектури, такі як AlexNet, VGG, та ResNet, здатні визначати складні особливості в даних і забезпечувати високу точність у складних умовах. Однак, вони вимагають значних обчислювальних ресурсів для тренування та виконання.

					БКР 122.006. ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		



Методи, такі як YOLO (You Only Look Once) і SSD (Single Shot Multibox Detector), дозволяють проводити детекцію в реальному часі з високою точністю та швидкістю. YOLO розбиває зображення на сітку і передбачає об'єкти в кожній комірці, що дозволяє швидко обробляти зображення за один прохід через нейронну мережу. SSD використовує багаторівневу архітектуру для передбачення обмежувальних рамок і класів об'єктів, забезпечуючи високу продуктивність і точність[13].

## 1.2 Основні проблеми існуючих методів

Основні проблеми існуючих методів детекції об'єктів включають:

### 1) Чутливість до змін в умовах освітлення та рухомого фону:

- Традиційні методи, такі як фонове видалення, сильно залежать від стабільності фону і освітлення. Різкі зміни в освітленні або наявності рухомих елементів у фоні можуть призводити до значних помилок в детекції об'єктів.

### 2) Необхідність великих обчислювальних ресурсів:

- Сучасні методи на основі глибокого навчання, такі як згорткові нейронні мережі, вимагають значних обчислювальних потужностей для тренування і виконання, що може бути проблематичним для реального часу застосувань на пристроях з обмеженими ресурсами.

### 3) Обмеження у відстеженні швидко рухомих об'єктів:

- Методи, такі як оптичний потік, можуть бути неточними при швидкому русі об'єктів, що призводить до розмиття і артефактів, ускладнюючи точне відстеження траєкторії руху.

### 4) Труднощі в інтеграції та масштабуванні:

- Інтеграція складних алгоритмів детекції в існуючі системи може вимагати значних зусиль і витрат, а також адаптації до конкретних умов застосування. Масштабування системи для обробки великого обсягу відеоданих може також стати викликом.

					БКР 122.006. ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

### 1.3 Вимоги до створюваного програмного прототипу

Для розробки працездатного програмного прототипу системи детекції об'єктів на відео необхідно враховувати сучасні тенденції та потреби користувачів, включаючи:

1) Висока швидкість обробки:

- Система повинна забезпечувати обробку відеопотоків у реальному часі, з мінімальною затримкою, що дозволить оперативно реагувати на події.

2) Висока точність детекції:

- Необхідно забезпечити високу точність виявлення об'єктів, з мінімальною кількістю помилкових спрацьовувань та невиявлених об'єктів.

3) Гнучкість і легкість інтеграції:

- Система повинна легко інтегруватися в існуючі IT-інфраструктури за допомогою веб-технологій, забезпечуючи широкий доступ та зручність використання.

4) Масштабованість:

- Система повинна бути масштабованою для роботи з різними типами об'єктів і умовами освітлення, що забезпечить її універсальність та адаптивність до різних сценаріїв використання.

5) Оптимізація обчислювальних ресурсів:

- Важливо забезпечити ефективне використання обчислювальних ресурсів, щоб зменшити навантаження на систему і підвищити продуктивність, особливо на пристроях з обмеженими можливостями.

### Висновки з першого розділу

Таким чином, аналіз існуючих методів детекції об'єктів на відео показав, що сучасні підходи, такі як згорткові нейронні мережі та методи реального часу, є найперспективнішими для створення ефективної та надійної системи. Традиційні методи мають обмеження в умовах змінного освітлення та складного фону, тоді як сучасні методи демонструють значні переваги в точності та швидкості обробки.

					БКР 122.006. ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

Основні проблеми існуючих методів включають чутливість до змін умов освітлення, необхідність великих обчислювальних ресурсів та труднощі у відстеженні швидко рухомих об'єктів. Ці проблеми визначають вимоги до нової системи, яка повинна забезпечувати високу швидкість обробки, точність детекції, гнучкість, легкість інтеграції та масштабованість. Результати стануть підґрунтям для розробки алгоритмічного і програмного забезпечення

					БКР 122.006. ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 МЕТОДИ І АЛГОРИТМИ СИСТЕМ ПОШУКУ І ДЕТЕКЦІЇ ОБ'ЄКТІВ НА ВІДЕО

### 2.1 Огляд основних методів пошуку та детекції об'єктів на відео

Розвиток методів пошуку та детекції об'єктів на відео є ключовим у контексті сучасних вимог до автоматизації та безпеки. Ці методи включають в себе різноманітні технології, кожна з яких має свої особливості та призначення, що дозволяє їх застосування у широкому діапазоні сценаріїв. Від точного визначення розташування об'єктів у міському середовищі до ідентифікації та класифікації рухомих об'єктів на виробництві, ці технології спрямовані на підвищення ефективності обробки відеоданих.

Наукове дослідження в цій галузі включає розробку нових алгоритмів, що можуть ефективно функціонувати в реальному часі, що має значний вплив на оперативні процеси в різних сферах. Так, алгоритми пошуку та детекції об'єктів дозволяють не тільки виявляти та відстежувати рухомі об'єкти, але й проводити їх глибший аналіз, що сприяє покращенню систем безпеки, автоматизації управління трафіком та оптимізації логістичних потоків[6].

#### 2.1.1 Традиційні методи детекції

##### 1)Фонове видалення (Background Subtraction)

Фонове видалення — це широко використовуваний метод в системах відеонагляду та інших застосунках, де потрібно виявляти рухомі об'єкти на статичному фоні. В основі методу лежить процес видалення статичного фону з поточного кадру відео для виділення об'єктів, що рухаються. Такий підхід дозволяє чітко ідентифікувати і відстежувати рухомі об'єкти навіть у складних сценах[17].

Принцип роботи:

Алгоритм аналізує кілька початкових кадрів для створення базової моделі фону. Ця модель може оновлюватися з часом для адаптації до змін в освітленні чи змін у самому фоні. Кожен новий кадр порівнюється з моделлю фону, і пікселі, що значно відрізняються, класифікуються як частина рухомого об'єкта.

					БКР 122.006. ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

Різниця може бути визначена через різноманітні методи, включаючи просте віднімання пікселів, статистичні тести чи використання складніших алгоритмів на основі машинного навчання.

Ефективність у контрольованих умовах: Фонове видалення особливо ефективно в умовах, де фон залишається незмінним, що робить його ідеальним для застосувань усередині приміщень або в інших стабільних середовищах. Метод є відносно простим для реалізації та не вимагає великої обчислювальної потужності, що робить його доступним для широкого спектру систем.

Чутливість до змін в освітленні та погоді: Фонове видалення може бути неточним у випадках, коли освітлення змінюється раптово, наприклад, через хмари, що проходять перед сонцем, або коли погодні умови варіюються. Складнощі виникають також тоді, коли фон містить рухомі об'єкти, такі як рухаючі дерева, штори тощо, що може вести до хибно позитивних детекцій.

Фонове видалення продовжує бути популярним вибором для багатьох застосувань завдяки своїй простоті та ефективності в певних сценаріях, але вимагає додаткових налаштувань та адаптацій для роботи в більш складних або змінних середовищах.

## 2) Оптичний потік (Optical Flow)

Оптичний потік — це метод в аналізі відео, який використовується для визначення руху об'єктів між послідовними кадрами на основі зміни інтенсивності пікселів. Цей метод особливо корисний для відстеження плавних рухів і часто застосовується в таких областях, як автоматизоване водіння, відеонагляд, а також у взаємодії людина-комп'ютер[15].

Принцип роботи:

Оптичний потік оцінює вектори переміщення пікселів між кадрами. За кожним пікселем у початковому кадрі слідкується, де він з'являється в наступному кадрі. Результатом є векторне поле, де кожен вектор показує напрямок та швидкість переміщення пікселів. Ці дані можуть використовуватись для аналізу руху, визначення траєкторій та навіть для реконструкції тривимірного руху в двовимірному просторі.

					БКР 122.006. ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

Оптичний потік ефективний для аналізу відео в умовах, де освітлення стабільне, що дозволяє точно відстежувати навіть незначні рухи. Від анімації і симуляції до розробки допоміжних систем для водіїв і робототехніки, оптичний потік дозволяє реалізовувати складні функції аналізу руху.

Чутливість до шуму та освітлення: Метод може бути неточним при сильних коливаннях освітлення або при високому рівні шуму в відео, що може призводити до хибних детекцій руху. При швидкому русі об'єктів можуть виникати розмиття та інші артефакти, що ускладнюють відстеження точного руху.

Оптичний потік є важливим інструментом у розробці алгоритмів для аналізу відео, який незамінний у багатьох технічних застосуваннях завдяки своїй здатності до детального і точного відстеження рухів. Однак, важливо зазначити, що ефективне використання цього методу вимагає ретельного вибору сцен та умов зйомки, а також подальшої обробки даних для мінімізації впливу можливих помилок та шуму.

### 3) Контурне виявлення (Edge Detection)

Контурне виявлення (Edge Detection) є одним з фундаментальних методів у області обробки зображень і аналізу відео, який використовується для ідентифікації меж об'єктів на зображенні. Цей метод заснований на виявленні різких змін інтенсивності пікселів, які зазвичай відповідають краям об'єктів[15].

#### Основи методу:

Алгоритми контурного виявлення, такі як Canny, використовуються для точного визначення країв об'єктів. Canny, зокрема, є одним з найефективніших і найпопулярніших методів виявлення країв завдяки своїй здатності мінімізувати помилкові детекції та втрати реальних країв.

#### Процес виявлення країв:

Перший крок в алгоритмі Canny полягає в застосуванні гауссівського фільтра для зменшення шуму на зображенні. Це допомагає уникнути неправильної ідентифікації шуму як країв. Застосування оператора Собеля або іншого оператора для виявлення градієнтів інтенсивності в горизонтальному та

					БКР 122.006. ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

вертикальному напрямках. Це виявляє напрямок найбільшої зміни інтенсивності, що вказує на ймовірність краю. Алгоритм Canny використовує два пороги для визначення сильних та слабких країв. Сильні краї вважаються справжніми, тоді як слабкі краї зберігаються лише, якщо вони з'єднані з сильними краями.

Canny ефективно виявляє краї, забезпечуючи чіткі та виразні лінії, що робить його ідеальним для застосувань, де важлива точність країв. Метод ефективний навіть в умовах з певним рівнем шуму, забезпечуючи стабільні результати.

Чутливість до шуму та змін в освітленні: Хоча алгоритм добре справляється з шумом, надмірне згладжування може призвести до втрати важливих деталей зображення. Висока обчислювальна складність може бути проблемою для реального часу в застосуваннях, які вимагають швидкої обробки зображень.

Контурне виявлення є важливим інструментом у відеоаналізі та обробці зображень, який забезпечує основу для багатьох подальших аналітичних застосувань, таких як виявлення об'єктів, аналіз текстури та сегментація зображень.

## 2.1.2 Сучасні методи на основі машинного навчання

### 1) Каскадні класифікатори (Cascade Classifiers)

Каскадні класифікатори є популярним методом у комп'ютерному зорі, який використовується для визначення об'єктів в зображеннях, особливо ефективний для детекції облич. Метод був розроблений Полом Віолою та Майклом Джонсом і згодом удосконалений у їх спільних роботах. Основна ідея полягає у використанні послідовності простих класифікаторів, що працюють каскадом для швидкого відкидання неможливих об'єктів, зосереджуючись на більш детальному аналізі потенційно можливих кандидатів.

Принцип роботи каскадних класифікаторів заснований на використанні навчених на основі позитивних та негативних зразків "слабких" класифікаторів, зазвичай заснованих на простих характеристиках, таких як Хаар-подібні

					БКР 122.006. ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

характеристики. Кожен класифікатор в каскаді оцінює зображення та вирішує, чи пройшов об'єкт цей етап відсіювання. Навчання класифікаторів відбувається на основі виявлення певних характеристик в зображеннях, таких як прості форми, краї, кольори чи текстури. Після тренування класифікатори організовуються у каскад, де кожен наступний класифікатор виконує більш складне та глибоке відсіювання. Початкові класифікатори швидко відкидають більшість негативних випадків, дозволяючи складнішим класифікаторам обробляти лише найбільш імовірні кандидати[15].

Каскадні класифікатори мають переваги в швидкості обробки, оскільки здатні швидко обробляти великі зображення, та ефективності для специфічних об'єктів, таких як обличчя або певні типи транспортних засобів, завдяки зосередженню на характерних характеристиках. Однак вони можуть страждати від високої кількості помилкових спрацьовувань та пропусків, особливо в складних візуальних сценах, і залежать від якості навчального набору даних. Каскадні класифікатори залишаються важливим інструментом у відеоаналізі та обробці зображень, забезпечуючи швидке та ефективне рішення для певних застосувань, де потрібна швидка відповідь системи і відомі параметри об'єктів для детекції.

## 2) Згорткові нейронні мережі (CNN)

Згорткові нейронні мережі (CNN)[11] відіграють ключову роль у сучасній обробці зображень та аналізі відео, завдяки їх здатності виявляти складні візуальні характеристики об'єктів на різних рівнях абстракції. CNN є особливо ефективними в задачах, пов'язаних з розпізнаванням образів, детекцією об'єктів та класифікацією зображень завдяки своїй архітектурі, яка імітує спосіб, яким людський зір обробляє візуальну інформацію[21].

Основні принципи роботи CNN включають використання згорткових шарів, які застосовують фільтри до вхідних зображень для створення карти ознак. Ці карти ознак визначають важливі характеристики, такі як краї, кути та інші текстурні елементи, які важливі для подальшої ідентифікації та класифікації об'єктів. Pooling (згортання) зменшує розміри карти ознак, зберігаючи при цьому

					БКР 122.006. ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		



важливу інформацію, що допомагає зменшити кількість параметрів і обчислень у мережі та запобігає перенаванчанням. На останніх етапах CNN повнозв'язні шари використовуються для класифікації об'єктів на основі визначених раніше характеристик, інтегруючи усі навчені ознаки в кінцеві результати, такі як ідентифікація конкретних об'єктів або класів об'єктів[11],[20].

Згорткові нейронні мережі відомі своєю здатністю досягати високої точності у складних завданнях розпізнавання та класифікації зображень, що робить їх незамінними в сучасних системах штучного інтелекту. Вони можуть адаптуватися до різноманітних візуальних завдань, а також до змін у вхідних даних, що робить їх ефективними в різних умовах і середовищах. Однак для ефективного навчання згорткових нейронних мереж потрібні великі обсяги розмічених даних, що може бути складно та дорого отримати в деяких умовах, і вони вимагають значних обчислювальних ресурсів, особливо для тренування, що може бути проблематичним для впровадження в мобільні або вбудовані системи. Згорткові нейронні мережі продовжують бути на передньому краї інновацій в області комп'ютерного зору, забезпечуючи потужні інструменти для аналізу та інтерпретації візуальної інформації.

### 2.1.3 Детектори об'єктів в реальному часі

Детектори об'єктів в реальному часі, такі як YOLO (You Only Look Once) та SSD (Single Shot Multibox Detector), представляють собою передові технології в області комп'ютерного зору, що дозволяють швидко і ефективно виявляти об'єкти у відеопотоках. Ці системи розроблені для забезпечення високої продуктивності і точності, що є критично важливим для застосувань, де необхідна миттєва реакція, таких як автомобільні системи безпеки, системи відеоспостереження та робототехніка[19].

1) YOLO (You Only Look Once):

YOLO (You Only Look Once) є передовим алгоритмом детекції об'єктів, який відрізняється своєю здатністю обробляти зображення в реальному часі з високою точністю. Розроблений Джозефом Редмоном і його командою, YOLO

					БКР 122.006. ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

радикально змінив підходи до детекції об'єктів, впровадивши метод, що дозволяє аналізувати зображення за один прохід через нейронну мережу[13].

YOLO використовує єдину згорткову нейронну мережу для одночасного передбачення декількох класів і локалізації об'єктів. Мережа ділить зображення на сітку (наприклад, 13x13 клітин), і кожна клітина сітки відповідає за передбачення об'єктів, які центрально розташовані в ній. Кожна клітина сітки може передбачати декілька обмежувальних рамок об'єкта і відповідні ймовірності того, що рамка містить певний клас об'єкта. Обмежувальні рамки характеризуються чотирма параметрами: центром (x, y), шириною та висотою. Крім того, кожна рамка має "показник достовірності" (confidence score), який відображає впевненість у тому, що рамка містить об'єкт і точність цієї рамки. Для кожної обмежувальної рамки передбачається ймовірність належності до кожного з можливих класів. Таким чином, система може одночасно ідентифікувати різні об'єкти та їхнє розташування на зображенні[13].

YOLO може обробляти зображення набагато швидше, ніж інші методи детекції об'єктів, що робить його ідеальним для систем, де критично важлива швидкість, наприклад, в системах автомобільного відеонагляду або в реальних робототехнічних застосуваннях. Завдяки глобальному аналізу зображення, YOLO зменшує кількість помилкових спрацьовувань у випадках, коли об'єкти мають складні перекриття або знаходяться в складних сценах. Через свою єдину мережеву структуру, YOLO легше інтегрувати і масштабувати в порівнянні з іншими більш складними системами. YOLO v8 вніс значні покращення в архітектуру і ефективність мережі, включаючи краще врахування контекстуальної інформації, покращення точності локалізації об'єктів та зниження кількості помилкових детекцій. Ці удосконалення роблять YOLO v8 ще більш потужним інструментом для реального часу детекції в різних областях застосування.

## 2) Нейронна мережа SSD (Single Shot MultiBox Detector):

SSD (Single Shot MultiBox Detector) є високоефективним методом для детекції об'єктів, який поєднує швидкість з високою точністю, роблячи його

					БКР 122.006. ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

популярним вибором для застосувань у реальному часі. Розроблений Вей Ліу та його колегами, SSD здатний виконувати детекцію об'єктів за один прохід через нейронну мережу, ефективно комбінуючи передбачення обмежувальних рамок та класифікацію об'єктів[18].

Основна особливість SSD полягає у його способі використання кількох різних рівнів згорткових шарів для передбачення обмежувальних рамок. Це дозволяє SSD ефективно виявляти об'єкти різних розмірів: великі об'єкти визначаються на більш ранніх шарах, де карта ознак є більшою, а маленькі об'єкти визначаються на пізніших шарах, де карта ознак менша. Кожен з рівнів у архітектурі SSD виробляє певну кількість фіксованих обмежувальних рамок для кожної області зображення і надає оцінки для наявності кожного класу об'єктів у рамці. Це дозволяє SSD бути надзвичайно швидким, оскільки весь процес відбувається за один прохід[18].

SSD вирізняється своєю здатністю швидко обробляти зображення при збереженні високої точності детекції, що робить його ідеальним для застосувань у реальному часі, де швидкість обробки є критичною. Завдяки своїй архітектурі, SSD може бути адаптований до різних розмірів вхідних даних та різних об'єктів, що робить його використання універсальним в багатьох сценаріях застосування. Хоча SSD є швидким, для досягнення оптимальної продуктивності він потребує значних обчислювальних ресурсів, особливо для тренування мережі. Незважаючи на використання багаторівневих карт ознак, SSD може мати певні труднощі з дуже малими об'єктами, які можуть загубитися на великих зображеннях[18].

## **2.2 Вибір методів та алгоритмів для розробки програмного прототипу**

У рамках розробки системи пошуку та детекції об'єктів на відео було обрано використання нейронної мережі YOLO v8, яка впроваджується через JavaScript. Вибір цього методу зумовлений кількома ключовими чинниками, що

					БКР 122.006. ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

забезпечують високу ефективність і адаптивність системи до потреб користувачів та технічних умов застосування.

#### Критерії вибору:

1. Швидкість обробки: YOLO v8 демонструє високу швидкість обробки зображень, що є критично важливим для реалізації систем відеонагляду та інших застосунків у реальному часі. Це дозволяє системі обробляти відеопотік з мінімальними затримками, забезпечуючи оперативну реакцію на події.

2. Точність детекції: Оновлені алгоритми та покращена архітектура YOLO v8 забезпечують високу точність в ідентифікації об'єктів. Це важливо для забезпечення надійності системи, знижуючи кількість помилкових спрацьовувань та невиявлених об'єктів.

3. Мінімізація помилок: Вдосконалення в алгоритмах YOLO v8 допомагають знизити кількість помилкових детекцій та пропусків, що підвищує загальну надійність системи. Це особливо важливо в умовах змінного освітлення та складного фону.

4. Легкість інтеграції та масштабування: Використання JavaScript для реалізації YOLO v8 дозволяє легко інтегрувати систему в існуючі IT-інфраструктури та масштабувати її залежно від змінних потреб користувачів. Це забезпечує гнучкість у розгортанні та адаптації системи.

#### Вибрані методи та їх застосування:

- YOLO v8 через JavaScript: Цей метод вибрано як основний інструмент для детекції об'єктів на відео у реальному часі. Він буде використовуватися для аналізу відеопотоків, що надходять з різних джерел, та ідентифікації об'єктів, таких як люди, транспортні засоби та інші значущі елементи в контексті застосування системи.

#### Обґрунтування вибору:

Використання YOLO v8 підтверджується її здатністю до швидкої адаптації до різних умов відеоспостереження та високим рівнем точності, що забезпечується завдяки глибокому навчанню та оптимізації алгоритмів. Такий

					БКР 122.006. ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

підхід дозволяє ефективно забезпечувати безпеку та контроль за об'єктами у різних сценаріях з мінімальними помилками та затримками у виявленні.

### 2.3 Опис обраного підходу та обґрунтування вибору

Для реалізації системи пошуку та детекції об'єктів на відео обраний підхід базується на використанні згорткової нейронної мережі YOLO v8, інтегрованої через JavaScript. Цей підхід був обраний з урахуванням специфічних вимог проекту, таких як необхідність швидкої обробки відеоданих та високої точності детекції об'єктів в реальному часі[16].

Технічні аспекти:

#### 1) Використання згорткових нейронних мереж (YOLO v8)

YOLO v8 (You Only Look Once, версія 8) представляє собою останню ітерацію відомої серії алгоритмів YOLO, яка спеціалізується на швидкій та точній детекції об'єктів у відеопотоках.

Покращена архітектура: YOLO v8 включає більш глибокі та широкі згорткові шари, що дозволяє моделі краще розпізнавати складні візуальні ознаки на різних масштабах. Це забезпечує високу точність детекції об'єктів навіть у складних сценах.

Ефективне виявлення: Модель оптимізована для розпізнавання ширшого спектра об'єктів з вищою точністю, що знижує кількість помилкових позитивів та невиявлених об'єктів. Завдяки цьому, YOLO v8 може точно ідентифікувати об'єкти різних розмірів та форм.

Краща загальна продуктивність: YOLO v8 оптимізована для ефективного використання обчислювальних ресурсів, що дозволяє системі працювати швидше та ефективніше навіть на обмежених платформах. Це особливо важливо для застосувань в реальному часі, де швидкість обробки відеоданих є критичною.

#### 2) Реалізація програмного прототипу на основі використання мови програмування JavaScript

Використання мови JavaScript для впровадження нейронної YOLO v8 дозволяє інтегрувати її в програмний прототип.

					БКР 122.006. ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

Крос-платформеність: JavaScript є універсальною мовою, що підтримується більшістю веб-браузерів та мобільних платформ, що спрощує розгортання системи на різноманітних кінцевих пристроях.

Швидка інтеграція: Можливість легкої інтеграції з існуючими веб-інтерфейсами та додатками забезпечує широку доступність та зручність використання.

Обробка відеоданих в реальному часі: Завдяки технологіям, реалізованим у YOLO v8, система може ефективно обробляти відеодані з мінімальною часовою затримкою.

Мінімальні затримки: Алгоритми оптимізовані для забезпечення мінімальної затримки між отриманням відеокадру та виявленням об'єктів, що життєво важливо для реагування в реальному часі.

Здатність обробки великого обсягу даних: Система здатна обробляти великі потоки даних без втрати продуктивності, забезпечуючи стабільну роботу навіть під високим навантаженням[12].

Обґрунтування вибору:

Основними чинниками, що вплинули на вибір YOLO v8 як основної технології для системи детекції об'єктів на відео, є її висока ефективність, здатність мінімізувати помилки, а також гнучкість і масштабованість рішення.

1) Висока ефективність

Оновлені алгоритми та архітектура YOLO v8 демонструють значні поліпшення у точності та швидкості обробки, що важливо для реалізації системи детекції в реальному часі. Нова архітектура використовує більш ефективні згорткові операції та алгоритми навчання, що знижують навантаження на обчислювальні системи та забезпечують швидшу обробку даних. Вдосконалені механізми класифікації та локалізації в YOLO v8 забезпечують краще розпізнавання об'єктів із нижчим рівнем помилок, що особливо важливо для надійності системи.

2) Мінімізація помилок

					БКР 122.006. ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

Зниження кількості помилкових спрацьовувань та пропусків об'єктів критично важливо для застосувань, де висока точність є обов'язковою. Система стає більш надійною для застосувань у сферах безпеки та моніторингу, де помилки можуть призвести до серйозних наслідків. Покращені алгоритми дозволяють системі ефективно функціонувати в різноманітних умовах освітлення та навколишнього середовища, що забезпечує високу універсальність застосування.

### 3) Гнучкість та масштабованість

Інтеграція YOLO v8 через JavaScript забезпечує велику гнучкість в застосуванні та можливість масштабування системи. JavaScript дозволяє легко інтегрувати систему в існуючі веб-та мобільні платформи, забезпечуючи широкий доступ до технології без необхідності спеціалізованого обладнання. Можливість масштабування від невеликих до великих систем дозволяє використовувати технологію в різних застосуваннях — від приватних домоволодінь до великих промислових об'єктів.

## Висновки з другого розділу

У розділі було проведено комплексний аналіз існуючих методів та алгоритмів пошуку та детекції об'єктів на відео, розглянуто їх переваги, недоліки та можливості застосування в різних умовах. Серед традиційних методів розглянуто фонове видалення, оптичний потік та контурне виявлення, які мають свої специфічні застосування та обмеження.

Окрему увагу приділено сучасним методам на основі згорткових нейронних мереж (CNN), каскадних класифікаторів та детекторам об'єктів у реальному часі, таким як YOLO (You Only Look Once) та SSD (Single Shot MultiBox Detector). Ці методи показують високу ефективність у складних завданнях детекції об'єктів завдяки їх здатності обробляти зображення швидко та точно.

На основі проведеного аналізу та критеріїв вибору було обрано нейронну мережу YOLO v8 для реалізації системи пошуку та детекції об'єктів на відео. Цей

					БКР 122.006. ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

метод був обраний через його високу швидкість обробки, точність детекції, мінімізацію помилок та легкість інтеграції у веб-середовище через JavaScript.

Вибір YOLO v8 та його інтеграція через JavaScript дозволяє створити ефективну, надійну та масштабовану систему, яка здатна забезпечити високу продуктивність та точність обробки відеопотоків у реальному часі. Цей підхід відповідає вимогам сучасних систем відеоспостереження, безпеки та автоматизації, забезпечуючи надійність і ефективність у різних умовах використання.

					БКР 122.006. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24



## 3 РОЗРОБКА ТА ЕКСПЕРИМЕНТАЛЬНА АПРОБАЦІЯ ПРОТОТИПУ ПРОГРАМИ

### 3.1 Використані технології та інструменти для реалізації програмного забезпечення

Мови програмування і фреймворки:

Проект розроблено з використанням мови програмування JavaScript, яка служить основою для створення інтерактивних клієнтських додатків. В контексті даної роботи, JavaScript використовувався для розробки скрипту `object\_detector.js`, який забезпечує інтеграцію з веб-воркерами (`worker.js`) для асинхронної обробки відеоданих[7].

Для побудови користувальницького інтерфейсу застосовані HTML та CSS, що дозволило реалізувати візуальну складову системи, включаючи відеопотік і візуалізацію результатів обробки на HTML5 Canvas[6],[5].

Клієнтські технології:

Використання `HTML5 Canvas` дозволяє ефективно відображати детекційні рамки навколо об'єктів на відео. Клієнтська частина також включає HTML контролери для управління відеопотоком. Асинхронну обробку даних здійснює веб-воркер, запущений зі скрипту `worker.js`, що дозволяє знизити навантаження на основний потік браузера і покращити загальну відгук системи[4],[10].

Серверні та інтеграційні технології:

Локальне розгортання сервера для тестування і розробки виконується через розширення Code Runner у Visual Studio Code. Це забезпечує можливість швидкого запуску та перевірки змін у коді, що є ідеальним для швидкої ітерації та прототипування.

Інструменти розробки:

Visual Studio Code виступає як основне середовище розробки, підтримуючи різноманітні інструменти для веб-розробників. Це IDE забезпечує

					БКР 122.006. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

розширений набір функцій, включаючи підтримку синтаксису JavaScript, HTML, CSS.

### 3.2 Опис архітектури системи

Структурно-функціональна схема системи програмного прототипу:

Структурно-функціональна схема демонструє, як різні компоненти системи взаємодіють між собою на функціональному рівні, зосереджуючись на їх функціях та зв'язках.

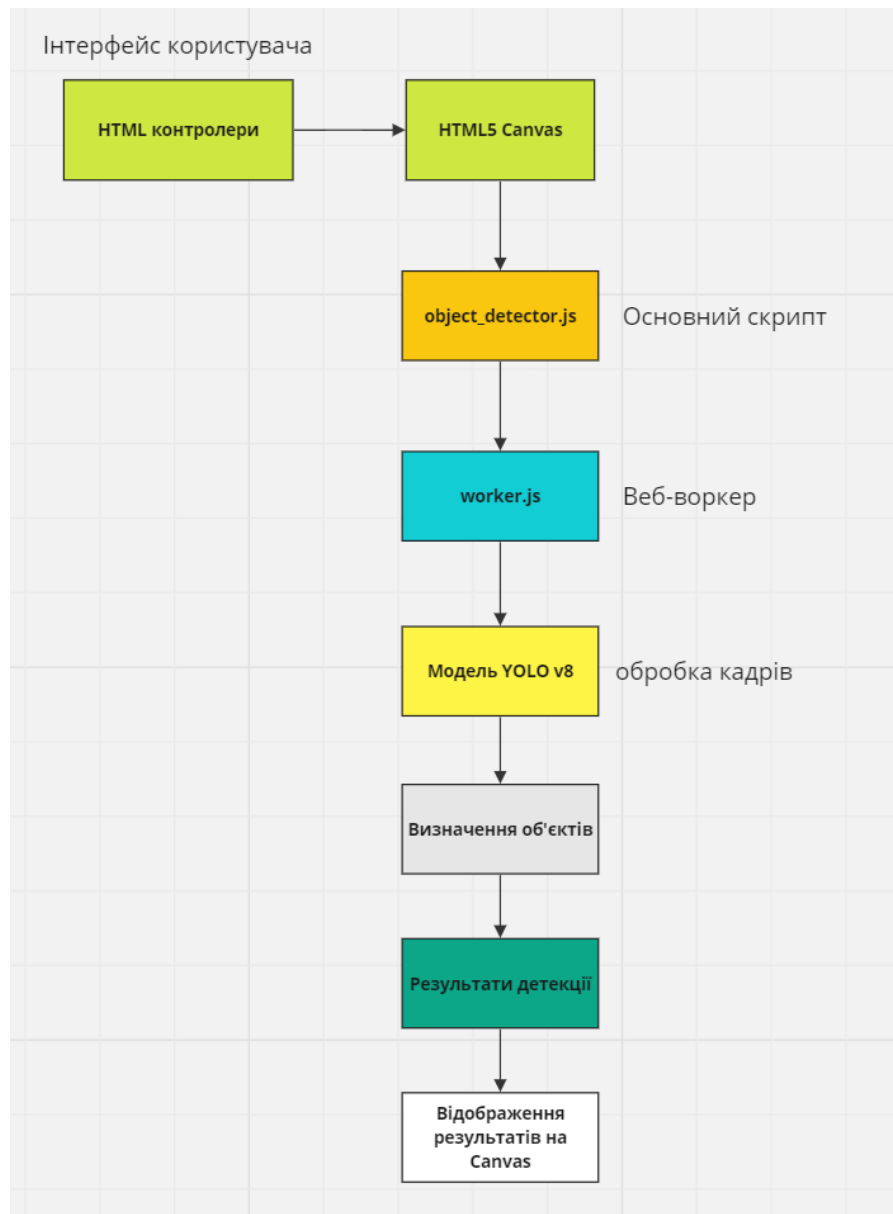


Рис.3.1 - структурно-функціональна схема системи детекції об'єктів на відео

### 3.2.1 Опис функцій складових програмного прототипу

#### 1) Інтерфейс користувача (User Interface)

- HTML контролери: Забезпечують управління відеопотоком, дозволяють завантажувати відео, починати та зупиняти обробку.
- HTML5 Canvas: Використовується для відображення відеопотоку та результатів детекції об'єктів у режимі реального часу.

#### 2) Основний скрипт (object\_detector.js)

- Ініціалізація веб-воркера: Основний скрипт запускає веб-воркер для обробки відеокадрів.
- Передача кадрів: Скрипт передає кадри відео до веб-воркера для обробки.
- Отримання результатів: Отримує результати обробки від веб-воркера і відображає їх на Canvas.

#### 3) Веб-воркер (worker.js)

- Асинхронна обробка кадрів: Веб-воркер обробляє кадри відео у фоновому режимі, не блокуючи основний потік браузера.
- Виклик моделі YOLO v8: Веб-воркер використовує модель YOLO v8 для детекції об'єктів на кадрах відео.

#### 4) Модель YOLO v8

- Обробка відеокадрів: Модель аналізує кожен кадр відео для виявлення об'єктів.
- Класифікація об'єктів: Визначає тип і позицію об'єктів на відео.

#### 5) Результати детекції

- Отримання результатів від веб-воркера: Основний скрипт отримує оброблені дані від веб-воркера.
- Відображення на Canvas: Результати детекції (рамки навколо об'єктів) відображаються на HTML5 Canvas.

### 3.2.2 Опис архітектури системи

#### 1) Інтерфейс користувача

					БКР 122.006. ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

- Реалізований з використанням HTML та CSS для створення контролерів управління відеопотоком і відображення відео та результатів детекції на HTML5 Canvas.

- HTML контролери: Елементи для управління відеопотоком (кнопки, слайдери і т.д.).

- HTML5 Canvas: Полотно для відображення відео і результатів детекції об'єктів.

## 2) Основний скрипт (object\_detector.js)

- Відповідає за ініціалізацію веб-воркера та передачу кадрів відео для асинхронної обробки. Він також отримує результати обробки та відображає їх на Canvas.

- Ініціалізація веб-воркера: Запуск веб-воркера для обробки відеоданих.

- Передача кадрів: Відправка кадрів відео до веб-воркера для обробки.

## 3) Веб-воркер (worker.js)

- Здійснює асинхронну обробку кадрів відео, що дозволяє знизити навантаження на основний потік браузера. Він використовує модель YOLO v8 для детекції об'єктів у кадрах.

- Асинхронна обробка кадрів: Обробка відеокadrів у фоновому режимі без блокування основного потоку.

- Виклик YOLO v8: Використання моделі YOLO v8 для детекції об'єктів на кадрах відео.

## 4) Модель YOLO v8

- Використовується для обробки кадрів відео та визначення об'єктів. Вона забезпечує високу точність та швидкість детекції об'єктів.

- Обробка кадрів відео: Аналіз кожного кадру для виявлення об'єктів.

- Класифікація об'єктів: Визначення типу і позиції об'єктів на відео.

## 5) Відображення результатів

- Результати детекції передаються від веб-воркера до основного скрипту, який відображає їх на HTML5 Canvas.

					БКР 122.006. ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

- Отримання результатів від веб-воркера: Основний скрипт отримує оброблені дані від веб-воркера.

- Відображення на Canvas: Результати детекції (рамки навколо об'єктів) відображаються на HTML5 Canvas.

Взаємодія компонентів:

- 1) Завантаження відео через HTML контролери.
- 2) Передача відеокадрів від основного скрипта до веб-воркера.
- 3) Обробка кадрів веб-воркером за допомогою моделі YOLO v8.
- 4) Отримання результатів детекції від веб-воркера.
- 5) Відображення результатів на HTML5 Canvas.

					БКР 122.006. ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

## 3.3 Опис реалізованих алгоритмів та характеристика їх роботи

### 3.3.1 Структура та вміст файлу `index.html`

Файл `index.html` створює базовий користувацький інтерфейс для системи детекції об'єктів на відео. Він включає HTML-компоненти, стилі та скрипти для забезпечення функціоналу системи.

Програмний код:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Object Detector</title>
6   <style>
7     body {
8       font-family: Arial, sans-serif;
9       background-color: #f4f4f9;
10      display: flex;
11      justify-content: center;
12      align-items: center;
13      height: 100vh;
14      margin: 0;
15    }
16    video {
17      display: none;
18    }
19    canvas {
20      border: 1px solid #ccc;
21      margin-top: 20px;
22    }
23    button {
24      color: white;
25      background-color: #8a220b;
26      border: none;
27      padding: 10px 20px;
28      margin: 10px;
29      border-radius: 5px;
30      cursor: pointer;
31      transition: background-color 0.3s ease;
32    }
33    button:hover {
34      background-color: #d42c06;
35    }
36    button:focus {
37      outline: none;
38    }
39  </style>
40 </head>
41 <body>
42   <video src="video.mp4" controls></video>
43   <canvas></canvas>
44   <div>
45     <button id="play">Play</button>
46     <button id="pause">Pause</button>
```

					БКР 122.006. ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

```
47     </div>
48     <script defer src="object_detector.js"></script>
49 </body>
50 </html>
```

Опис компонентів:

1) Метадані та стилі:

- `<!DOCTYPE html>`: Встановлює тип документа як HTML5.
- `<html lang="en">`: Визначає мову документа.
- `<meta charset="UTF-8">`: Встановлює кодування символів для коректного відображення тексту.
- `<style>`: Внутрішні стилі забезпечують базове форматування сторінки, включаючи фоновий колір, шрифти, стилі для кнопок та `canvas`.

2) HTML-елементи: убрать большие буквы

- `<video>`: Прихований елемент для відтворення відео, який використовується для завантаження та обробки відеопотоку.
- `<canvas>`: Полотно для відображення результатів детекції, на якому малюються рамки навколо виявлених об'єктів.
- `<button>`: Кнопки для керування відтворенням відео (Play та Pause), що забезпечують інтерфейс для взаємодії користувача з відеопотоком.

3) Підключення JavaScript:

- `<script defer src="object_detector.js"></script>`: Підключає скрипт `object\_detector.js` з атрибутом `defer`, що забезпечує його завантаження та виконання після завантаження HTML-документа.

Цей файл надає основу для взаємодії користувача з системою детекції об'єктів, забезпечуючи інтерфейс для завантаження та керування відео, а також область для відображення результатів.

### 3.3.2 Структура та вміст файлу `object\_detector.js`

Файл `object\_detector.js` є основним скриптом для обробки відеопотоку, виконання детекції об'єктів за допомогою веб-воркера, та відображення

					БКР 122.006. ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

результатів на канвасі. Він відповідає за інтеграцію відеопотоку, підготовку даних для моделі YOLO v8, обробку результатів детекції та візуалізацію.

### Програмний код:

```
1 const videoElement = document.querySelector("video");
2
3 const detectionWorker = new Worker("worker.js");
4 let detectionBoxes = [];
5 let detectionInterval;
6 let isBusy = false;
7
8 videoElement.addEventListener("play", () => {
9     const canvasElement = document.querySelector("canvas");
10    canvasElement.width = videoElement.videoWidth;
11    canvasElement.height = videoElement.videoHeight;
12    const canvasContext = canvasElement.getContext("2d");
13    detectionInterval = setInterval(() => {
14        canvasContext.drawImage(videoElement, 0, 0);
15        renderDetectionBoxes(canvasElement, detectionBoxes);
16        const inputTensor = prepareInputTensor(canvasElement);
17        if (!isBusy) {
18            detectionWorker.postMessage(inputTensor);
19            isBusy = true;
20        }
21    }, 30);
22 });
23
24 detectionWorker.onmessage = (event) => {
25     const outputData = event.data;
26     const canvasElement = document.querySelector("canvas");
27     detectionBoxes = processModelOutput(outputData, canvasElement.width,
28 canvasElement.height);
29     isBusy = false;
30 };
31
32 videoElement.addEventListener("pause", () => {
33     clearInterval(detectionInterval);
34 });
35
36 const playButton = document.getElementById("play");
37 const pauseButton = document.getElementById("pause");
38 playButton.addEventListener("click", () => {
39     videoElement.play();
40 });
41 pauseButton.addEventListener("click", () => {
42     videoElement.pause();
43 });
44
45 function prepareInputTensor(imgElement) {
46     const tempCanvas = document.createElement("canvas");
47     tempCanvas.width = 640;
48     tempCanvas.height = 640;
49     const tempContext = tempCanvas.getContext("2d");
50     tempContext.drawImage(imgElement, 0, 0, 640, 640);
51     const imageData = tempContext.getImageData(0, 0, 640, 640).data;
```

					БКР 122.006. ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		



```

52     const redChannel = [], greenChannel = [], blueChannel = [];
53     for (let i = 0; i < imageData.length; i += 4) {
54         redChannel.push(imageData[i] / 255);
55         greenChannel.push(imageData[i + 1] / 255);
56         blueChannel.push(imageData[i + 2] / 255);
57     }
58     return [...redChannel, ...greenChannel, ...blueChannel];
59 }
60
61 function processModelOutput(output, imageWidth, imageHeight) {
62     let detectedBoxes = [];
63     for (let i = 0; i < 8400; i++) {
64         const [classId, probability] = [...Array(yoloClasses.length).keys()]
65             .map(col => [col, output[8400 * (col + 4) + i]])
66             .reduce((max, item) => item[1] > max[1] ? item : max, [0, 0]);
67         if (probability < 0.5) {
68             continue;
69         }
70         const label = yoloClasses[classId];
71         const xCenter = output[i];
72         const yCenter = output[8400 + i];
73         const width = output[2 * 8400 + i];
74         const height = output[3 * 8400 + i];
75         const x1 = (xCenter - width / 2) / 640 * imageWidth;
76         const y1 = (yCenter - height / 2) / 640 * imageHeight;
77         const x2 = (xCenter + width / 2) / 640 * imageWidth;
78         const y2 = (yCenter + height / 2) / 640 * imageHeight;
79         detectedBoxes.push([x1, y1, x2, y2, label, probability]);
80     }
81     detectedBoxes = detectedBoxes.sort((box1, box2) => box2[5] - box1[5]);
82     const filteredBoxes = [];
83     while (detectedBoxes.length > 0) {
84         filteredBoxes.push(detectedBoxes[0]);
85         detectedBoxes = detectedBoxes.filter(box =>
86 calculateIoU(detectedBoxes[0], box) < 0.7 || detectedBoxes[0][4] !==
87 box[4]);
88     }
89     return filteredBoxes;
90 }
91
92 function calculateIoU(box1, box2) {
93     return calculateIntersection(box1, box2) / calculateUnion(box1, box2);
94 }
95
96 function calculateUnion(box1, box2) {
97     const [x1a, y1a, x2a, y2a] = box1;
98     const [x1b, y1b, x2b, y2b] = box2;
99     const area1 = (x2a - x1a) * (y2a - y1a);
100    const area2 = (x2b - x1b) * (y2b - y1b);
101    return area1 + area2 - calculateIntersection(box1, box2);
102 }
103
104 function calculateIntersection(box1, box2) {
105     const [x1a, y1a, x2a, y2a] = box1;
106     const [x1b, y1b, x2b, y2b] = box2;
107     const x1 = Math.max(x1a, x1b);
108     const y1 = Math.max(y1a, y1b);
109     const x2 = Math.min(x2a, x2b);

```

					БКР 122.006. ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

```

110     const y2 = Math.min(y2a, y2b);
111     return (x2 - x1) * (y2 - y1);
112 }
113
114 function renderDetectionBoxes(canvasElement, boxes) {
115     const ctx = canvasElement.getContext("2d");
116     ctx.strokeStyle = "#0000FF";
117     ctx.lineWidth = 3;
118     ctx.font = "18px Arial";
119     boxes.forEach(([x1, y1, x2, y2, label]) => {
120         ctx.strokeRect(x1, y1, x2 - x1, y2 - y1);
121         ctx.fillStyle = "#0000FF";
122         const textWidth = ctx.measureText(label).width;
123         ctx.fillRect(x1, y1, textWidth + 10, 25);
124         ctx.fillStyle = "#FFFFFF";
125         ctx.fillText(label, x1, y1 + 18);
126     });
127 }
128
129 const yoloClasses = [
130     'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train',
    'truck', 'boat',
    'traffic light', 'fire hydrant', 'stop sign', 'parking meter', 'bench',
    'bird', 'cat', 'dog', 'horse',
    'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'backpack',
    'umbrella'
];

```

Опис компонентів:

### 1) Ініціалізація відеоелемента

```
const videoElement = document.querySelector("video");
```

-Отримання посилання на елемент `video`, який буде використовуватися для завантаження та відтворення відеопотоку.

### 2) Ініціалізація веб-воркера та змінних

```
const detectionWorker = new Worker("worker.js");
let detectionBoxes = [];
let detectionInterval;
let isBusy = false;
```

- Ініціалізація веб-воркера для асинхронної обробки відеокадрів.

- Змінні `detectionBoxes`, `detectionInterval` та `isBusy` використовуються для зберігання результатів детекції, інтервалу обробки та статусу обробки відповідно.

### 3) Обробка події `play` відеоелемента

					БКР 122.006. ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

```

videoElement.addEventListener("play", () => {
    const canvasElement = document.querySelector("canvas");
    canvasElement.width = videoElement.videoWidth;
    canvasElement.height = videoElement.videoHeight;
    const canvas

Context = canvasElement.getContext("2d");
detectionInterval = setInterval(() => {
    canvasContext.drawImage(videoElement, 0, 0);
    renderDetectionBoxes(canvasElement, detectionBoxes);
    const inputTensor = prepareInputTensor(canvasElement);
    if (!isBusy) {
        detectionWorker.postMessage(inputTensor);
        isBusy = true;
    }
}, 30);
});

```

- Підключення обробника події `play` для відеоелемента.
- Ініціалізація канвасу для відображення відео та встановлення його розмірів відповідно до розмірів відео.
- Налаштування інтервалу обробки відеокадрів кожні 30 мс, копіювання кадрів на канвас, підготовка тензора для моделі та надсилання його до веб-воркера.

#### 4) Обробка повідомлень від веб-воркера

```

detectionWorker.onmessage = (event) => {
    const outputData = event.data;
    const canvasElement = document.querySelector("canvas");
    detectionBoxes = processModelOutput(outputData,
canvasElement.width, canvasElement.height);
    isBusy = false;
};

```

- Отримання результатів детекції від веб-воркера.
- Обробка даних моделі та оновлення результатів детекції.
- Відновлення статусу `isBusy` для дозволу подальшої обробки.

#### 5) Обробка події `pause` відеоелемента

```

videoElement.addEventListener("pause", () => {
    clearInterval(detectionInterval);
});

```

- Зупинка інтервалу обробки при паузі відтворення відео.

					БКР 122.006. ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

## 6) Обробка кнопок Play та Pause

```
const playButton = document.getElementById("play");
const pauseButton = document.getElementById("pause");
playButton.addEventListener("click", () => {
  videoElement.play();
});
pauseButton.addEventListener("click", () => {
  videoElement.pause();
});
```

- Додавання обробників подій для кнопок Play та Pause для керування відтворенням відео.

## 7) Функція підготовки вхідного елемента

```
function prepareInputTensor(imgElement) {
  const tempCanvas = document.createElement("canvas");
  tempCanvas.width = 640;
  tempCanvas.height = 640;
  const tempContext = tempCanvas.getContext("2d");
  tempContext.drawImage(imgElement, 0, 0, 640, 640);
  const imageData = tempContext.getImageData(0, 0, 640, 640).data;
  const redChannel = [], greenChannel = [], blueChannel = [];
  for (let i = 0; i < imageData.length; i += 4) {
    redChannel.push(imageData[i] / 255);
    greenChannel.push(imageData[i + 1] / 255);
    blueChannel.push(imageData[i + 2] / 255);
  }
  return [...redChannel, ...greenChannel, ...blueChannel];
}
```

- Створення тимчасового канвасу для нормалізації вхідних даних.

- Зчитування даних зображення та нормалізація значень кольорових каналів.

## 8) Функція обробки результатів моделі

Функція `processModelOutput` відповідає за обробку результатів, отриманих від моделі YOLO v8, і визначення координат об'єктів на відео. Вона також виконує фільтрацію результатів для уникнення перекриття детекцій.

```
function processModelOutput(output, imageWidth, imageHeight) {
  let detectedBoxes = [];
  for (let i = 0; i < 8400; i++) {
    const [classId, probability] =
[...Array(yoloClasses.length).keys()]
      .map(col => [col, output[8400 * (col + 4) + i]])
      .reduce((max, item) => item[1] > max[1] ? item : max, [0,
0]);
    if (probability < 0.5) {
      continue;
    }
  }
}
```

					БКР 122.006. ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    }
    const label = yoloClasses[classId];
    const xCenter = output[i];
    const yCenter = output[8400 + i];
    const width = output[2 * 8400 + i];
    const height = output[3 * 8400 + i];
    const x1 = (xCenter - width / 2) / 640 * imageWidth;
    const y1 = (yCenter - height / 2) / 640 * imageHeight;
    const x2 = (xCenter + width / 2) / 640 * imageWidth;
    const y2 = (yCenter + height / 2) / 640 * imageHeight;
    detectedBoxes.push([x1, y1, x2, y2, label, probability]);
  }
  detectedBoxes = detectedBoxes.sort((box1, box2) => box2[5] -
box1[5]);
  const filteredBoxes = [];
  while (detectedBoxes.length > 0) {
    filteredBoxes.push(detectedBoxes[0]);
    detectedBoxes = detectedBoxes.filter(box =>
calculateIoU(detectedBoxes[0], box) < 0.7 || detectedBoxes[0][4] !== box[4]);
  }
  return filteredBoxes;
}
}

```

- Ініціалізація масиву `detectedBoxes`\*
  - Порожній масив для зберігання детектованих об'єктів.
- Цикл обробки результатів:
  - Проходить через кожен з 8400 можливих об'єктів, щоб отримати ймовірність класу та визначити координати об'єкта.
    - Використовує методи `map` та `reduce` для визначення класу з найвищою ймовірністю.
  - Фільтрація за ймовірністю:
    - Якщо ймовірність класу менше 0.5, об'єкт ігнорується.
  - Обчислення координат об'єкта:
    - Визначає центральні координати, ширину та висоту об'єкта, а також обчислює координати обмежувальної рамки (x1, y1, x2, y2).
  - Додавання детектованих об'єктів до масиву:
    - Додає об'єкт з його координатами та міткою до масиву `detectedBoxes`.
  - Сортування та фільтрація об'єктів:
    - Сортує об'єкти за ймовірністю в порядку спадання.
    - Фільтрує об'єкти, щоб уникнути перекриття детекцій, використовуючи функцію `calculateIoU`.

					БКР 122.006. ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

## 9) Функції обчислення IoU та візуалізації рамок

Ці функції відповідають за обчислення Intersection over Union (IoU) для визначення ступеня перекриття між об'єктами та за візуалізацію рамок навколо виявлених об'єктів на канвасі.

```
function calculateIoU(box1, box2) {
    return calculateIntersection(box1, box2) / calculateUnion(box1,
box2);
}

function calculateUnion(box1, box2) {
    const [x1a, y1a, x2a, y2a] = box1;
    const [x1b, y1b, x2b, y2b] = box2;
    const area1 = (x2a - x1a) * (y2a - y1a);
    const area2 = (x2b - x1b) * (y2b - y1b);
    return area1 + area2 - calculateIntersection(box1, box2);
}

function calculateIntersection(box1, box2) {
    const [x1a, y1a, x2a, y2a] = box1;
    const [x1b, y1b, x2b, y2b] = box2;
    const x1 = Math.max(x1a, x1b);
    const y1 = Math.max(y1a, y1b);
    const x2 = Math.min(x2a, x2b);
    const y2 = Math.min(y2a, y2b);
    return (x2 - x1) * (y2 - y1);
}

function renderDetectionBoxes(canvasElement, boxes) {
    const ctx = canvasElement.getContext("2d");
    ctx.strokeStyle = "#0000FF";
    ctx.lineWidth = 3;
    ctx.font = "18px serif";
    boxes.forEach(([x1, y1, x2, y2, label]) => {
        ctx.strokeRect(x1, y1, x2 - x1, y2 - y1);
        ctx.fillStyle = "#0000FF";
        const textWidth = ctx.measureText(label).width;
        ctx.fillRect(x1, y1, textWidth + 10, 25);
        ctx.fillStyle = "#FFFFFF";
        ctx.fillText(label, x1, y1 + 18);
    });
}
```

Функція `calculateIoU`:

- Викликає функції `calculateIntersection` та `calculateUnion` для обчислення IoU, що визначає ступінь перекриття між двома об'єктами.

Функція `calculateUnion`:

- Обчислює площу об'єднання двох прямокутників. Визначає площу кожного прямокутника та віднімає площу їх перетину.

					БКР 122.006. ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

Функція `calculateIntersection`:

- Обчислює площу перетину двох прямокутників. Визначає координати області, де прямокутники перекриваються.

Функція `renderDetectionBoxes`:

- Відповідає за візуалізацію результатів детекції на канвасі.  
- Використовує контекст 2D канвасу для малювання обмежувальних рамок навколо виявлених об'єктів.

- Малює прямокутники та заповнює їх текстовими мітками, що показують клас об'єкта.

### 10) Класи об'єктів для моделі YOLO v8

```
const yoloClasses = [  
  'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus',  
'train', 'truck', 'boat',  
  'traffic light', 'fire hydrant', 'stop sign', 'parking meter',  
'bench', 'bird', 'cat', 'dog', 'horse',  
  'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'backpack',  
'umbrella'  
];
```

- Перелік класів об'єктів, які може розпізнавати модель YOLO v8.

### 3.3.3 Структура та вміст файлу `worker.js`

Файл `worker.js` виконує ключову роль у системі детекції об'єктів, забезпечуючи асинхронну обробку відеокадрів за допомогою моделі YOLO v8. Веб-воркер дозволяє виконувати обчислення у фоновому режимі, що знижує навантаження на основний потік браузера.

#### Програмний код

```
importScripts("https://cdn.jsdelivr.net/npm/onnxruntime-web/dist/ort.min.js");  
  
let model = null;  
  
onmessage = async(event) => {  
  const input = event.data;  
  const output = await runModel(input);  
  postMessage(output);  
}  
  
async function runModel(input) {
```

					БКР 122.006. ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    if (!model) {
      model = await ort.InferenceSession.create("./yolov8n.onnx");
    }
    input = new ort.Tensor(Float32Array.from(input), [1, 3, 640, 640]);
    const outputs = await model.run({ images: input });
    return outputs["output0"].data;
  }
}

```

## Опис компонентів:

### 1) Імпорт бібліотеки ONNX Runtime:

```

importScripts("https://cdn.jsdelivr.net/npm/onnxruntime-web/dist/ort.min.js");

```

- Цей рядок імпортує бібліотеку ONNX Runtime для роботи з ONNX моделями в браузері.

### 2) Ініціалізація змінної моделі:

```

let model = null;

```

- Змінна `model` використовується для зберігання завантаженої моделі YOLO v8.

### 3) Обробка повідомлень від основного потоку:

```

onmessage = async(event) => {
  const input = event.data;
  const output = await runModel(input);
  postMessage(output);
}

```

- Веб-воркер слухає повідомлення від основного потоку. Отримані дані передаються до функції `runModel`, після чого результати обробки надсилаються назад до основного потоку.

### 4) Функція `runModel`:

```

async function runModel(input) {
  if (!model) {
    model = await ort.InferenceSession.create("./yolov8n.onnx");
  }
  input = new ort.Tensor(Float32Array.from(input), [1, 3, 640,
640]);

  const outputs = await model.run({ images: input });
  return outputs["output0"].data;
}

```

					БКР 122.006. ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		



- Завантаження моделі YOLO v8 (якщо вона ще не завантажена) та створення сесії для інференції.

- Перетворення вхідних даних у тензор, сумісний з ONNX Runtime.

- Виконання інференції на моделі та повернення результатів обробки.

Файл `worker.js` забезпечує ефективну та асинхронну обробку відеоданих за допомогою моделі YOLO v8, що дозволяє знижувати навантаження на основний потік браузера та покращувати продуктивність системи.

### 3.3.4 Структура та вміст файлу `ort-wasm-simd.wasm`

Файл `ort-wasm-simd.wasm` є компонентом бібліотеки ONNX Runtime для WebAssembly (Wasm). Цей файл забезпечує виконання моделей машинного навчання у браузері з використанням технології WebAssembly, що дозволяє досягати високої продуктивності обчислень[8].

Основні аспекти:

#### 1) WebAssembly (Wasm):

- WebAssembly є низькорівневою мовою програмування, що дозволяє виконувати високопродуктивний код у браузерах. Він забезпечує майже нативну швидкість виконання, що є важливим для обробки великих обсягів даних у реальному часі[9].

#### 2) SIMD (Single Instruction, Multiple Data):

- SIMD — це метод паралельної обробки, який дозволяє виконувати одну і ту ж операцію одночасно над кількома даними. Використання SIMD у WebAssembly дозволяє значно покращити продуктивність обчислень[9].

Функціональність:

- Файл `ort-wasm-simd.wasm` завантажується бібліотекою ONNX Runtime для виконання інференції моделей машинного навчання у браузері. Це дозволяє виконувати складні обчислення без необхідності передавати дані на сервер, що покращує швидкість і знижує затримки.

					БКР 122.006. ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3.3.5 Структура та вміст файлу `yolov8n.onnx`

Файл `yolov8n.onnx` містить модель YOLO v8 у форматі ONNX (Open Neural Network Exchange). ONNX — це відкрита стандартна специфікація для представлення моделей машинного навчання, що дозволяє легко переносити моделі між різними фреймворками[13],[18].

Основні аспекти:

1) YOLO v8 (You Only Look Once, версія 8):

- YOLO — це одна з найпопулярніших моделей для детекції об'єктів у реальному часі. Вона відома своєю швидкістю та високою точністю.

- YOLO v8 включає покращення у архітектурі, що дозволяють краще розпізнавати об'єкти на різних масштабах та зменшують кількість помилкових спрацьовувань.

2) ONNX (Open Neural Network Exchange):

- ONNX є відкритим форматом для представлення моделей машинного навчання, який підтримується багатьма фреймворками, включаючи PyTorch, TensorFlow та інші. Це дозволяє легко обмінюватися моделями між різними системами та середовищами.

Функціональність:

- Файл `yolov8n.onnx` використовується для інференції у веб-воркері. Модель завантажується бібліотекою ONNX Runtime і використовується для детекції об'єктів на відеокадрах у реальному часі.

					БКР 122.006. ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3.4 Експериментальна апробація прототипу програми

#### 3.4.1 Сценарії тестування та вибір датасетів

Основною метою тестування даної системи детекції об'єктів на відео є перевірка її здатності ефективно детектувати та локалізувати об'єкти в різноманітних та складних умовах відеозйомки. Тестування має на меті оцінити наступні аспекти системи:

1) Точність детекції: Визначення, наскільки точно система може виявляти об'єкти в відео, що включає визначення типів об'єктів із заданого набору класів.

2) Локалізація об'єктів: Оцінка здатності системи точно визначати місцеположення об'єктів на відео, включаючи їхні координати та розміри, що критично важливо для подальших застосувань, таких як моніторинг або навігація.

3) Робота в різних умовах освітлення: Перевірка здатності системи стабільно функціонувати в умовах низької освітленості, зміни освітленості, а також при яскравому світлі, що є типовими умовами для зовнішнього використання.

4) Стійкість до змін у динамічному середовищі: Аналіз ефективності системи при виявленні і слідкуванні за об'єктами, які швидко рухаються або коли сама камера рухається.

#### Тестові сценарії для подальшого тестування системи:

##### Сценарій 1: Різні умови освітлення

- Мета: Перевірка адаптивності системи до змін у рівнях освітлення.
- Опис: Тестування системи в умовах яскравого світла, сутінків та ночі для оцінки здатності системи точно детектувати об'єкти при різних режимах освітлення.

##### Сценарій 2: Різноманітність об'єктів

- Мета: Оцінка точності системи у виявленні різних типів об'єктів.
- Опис: Тестування системи на здатність ідентифікувати пішоходів, транспортні засоби та тварин, щоб підтвердити універсальність детекції.

##### Сценарій 3: Динамічні сцени

					БКР 122.006. ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

- Мета: Випробування системи в умовах динамічних змін у відеокадрах.
- Опис: Оцінка ефективності системи при виявленні та слідкуванні за об'єктами, які швидко рухаються, або коли камера змінює положення.

Сценарій 4: Варіабельність погодних умов

- Мета: Перевірка надійності системи в різних погодних умовах.
- Опис: Тестування системи в умовах дощу, туману та снігу для оцінки здатності системи розпізнавати об'єкти незалежно від погоди.

Сценарій 5: Взаємодія та перекриття об'єктів

- Мета: Оцінка здатності системи розрізняти та точно локалізувати об'єкти, що перекриваються.
- Опис: Аналіз як система розрізняє окремі об'єкти в масивних або переповнених сценах, особливо в урбаністичних умовах.

Сценарій 6: Тестування стійкості до шуму

- Мета: Оцінка ефективності системи при високому рівні шуму на відео.
- Опис: Випробування системи на відеоматеріалах з різними рівнями шуму, щоб визначити, наскільки добре система може відфільтрувати помилкові сигнали.

					БКР 122.006. ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3.4.2 Експериментальна апробація та аналіз отриманих результатів

#### Сценарій 1: Різні умови освітлення

- Мета: Перевірка адаптивності системи до змін у рівнях освітлення.
- Опис: Тестування системи в умовах яскравого світла, сутінків та ночі для оцінки здатності системи точно детектувати об'єкти при різних інтенсивностях освітлення.

#### Результати тестування:

##### 1) Яскраве світло:

- Кількість тестових відео: 2
- Виявлені об'єкти: 90% точність детекції
- Проблеми: Проблеми не спостерігаються.
- Візуалізація результатів:

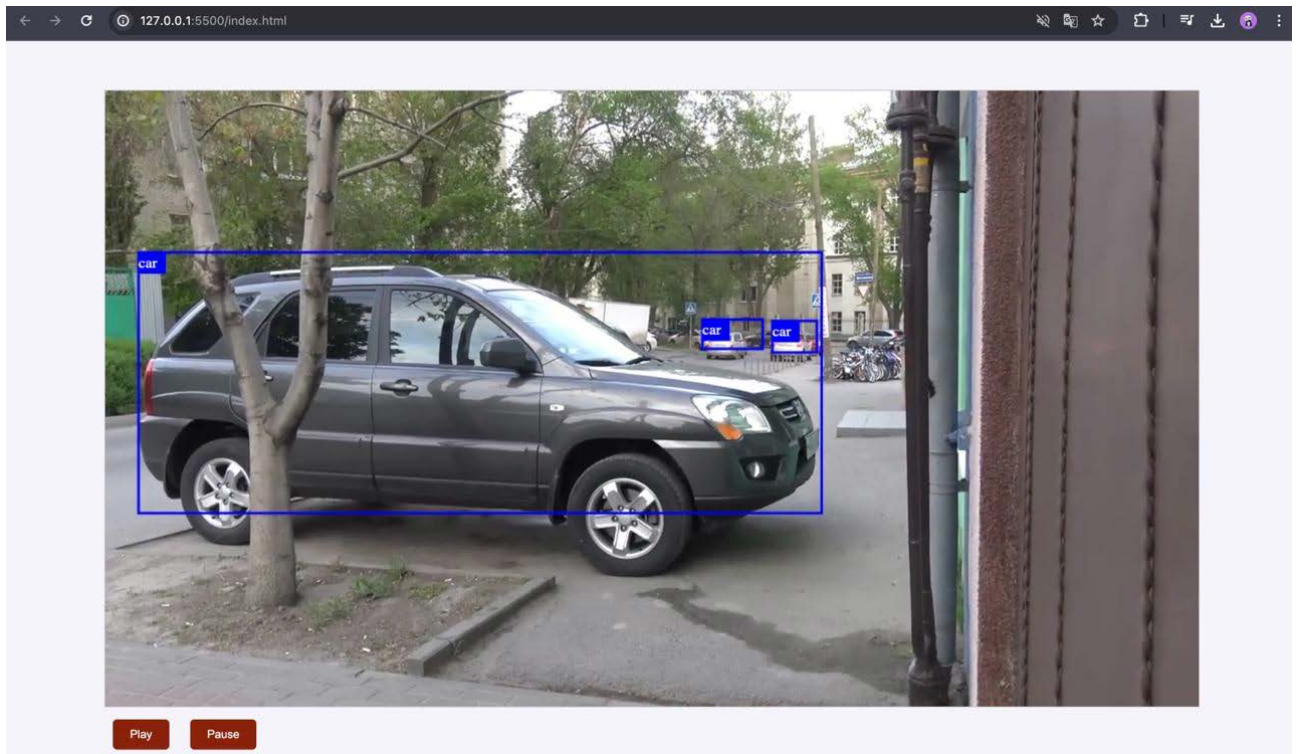


Рис.3.2 - результати роботи системи з об'єктами з яскравим світлом на першому відео

					БКР 122.006. ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

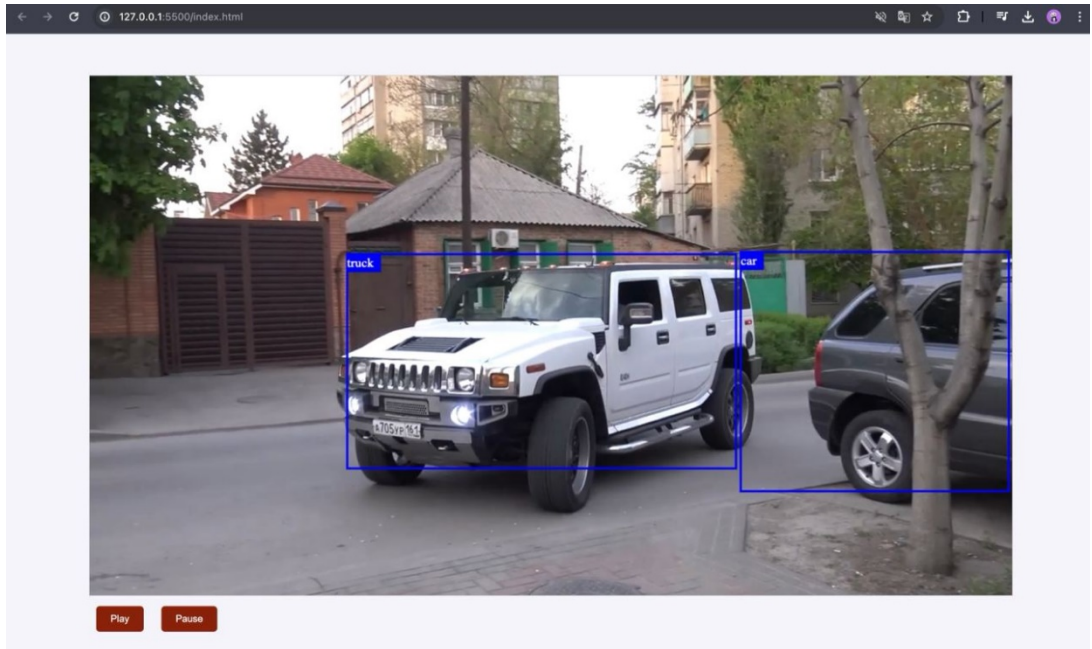


Рис.3.3 - результати роботи системи з об'єктами з яскравим світлом на другому відео

2) Сутінки:

- Кількість тестових відео: 1 відео
- Виявлені об'єкти: 70-80% точність детекції
- Проблеми: При поганому освітленні система не всі об'єкти знаходить в кадрі, якщо використовувати відеоматеріал з меншою кількістю об'єктів, то система розпізнає майже досконально.

- Візуалізація результатів:

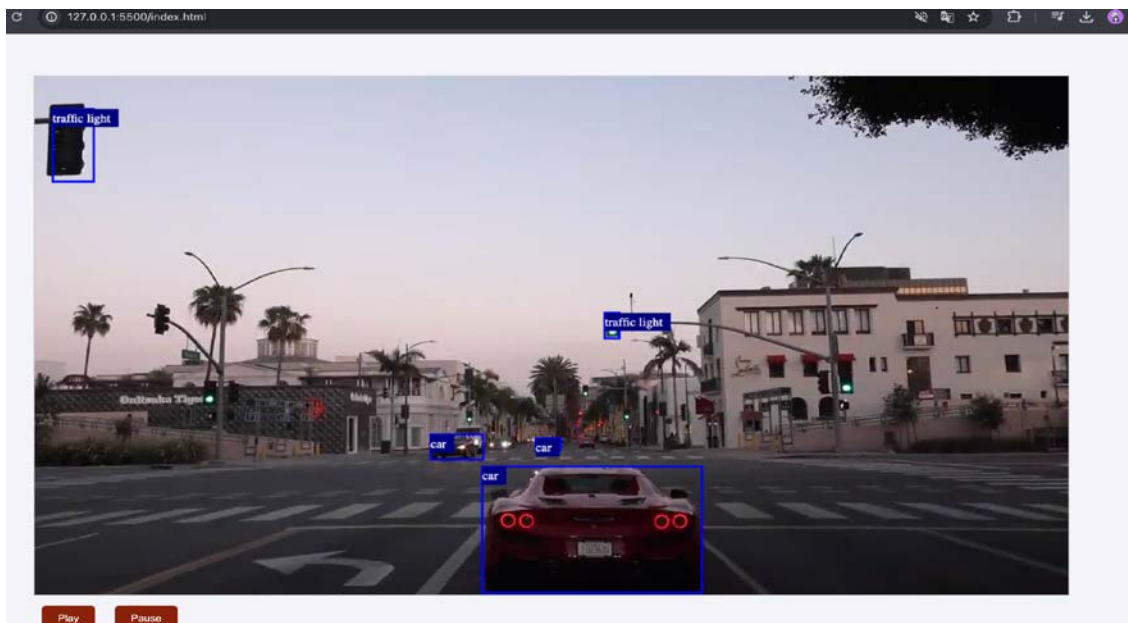


Рис.3.4 - тестування системи в сутінках 1 скрін

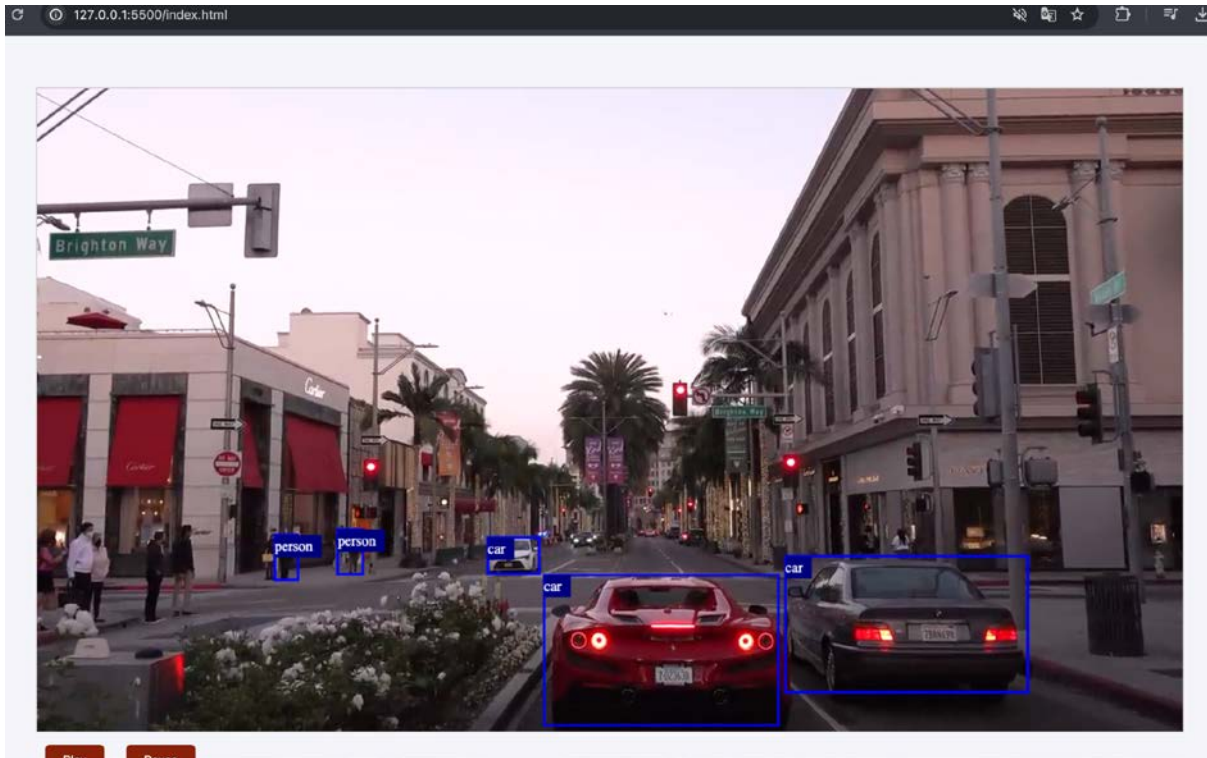


Рис.3.5 - тестування системи в сутінках 2 скрін

3)Ніч:

- Кількість тестових відео: 1
- Виявлені об'єкти: 70% точність детекції
- Проблеми: Спостереження за впливом недостатнього освітлення на якість детекції.

					БКР 122.006. ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

- Візуалізація результатів:

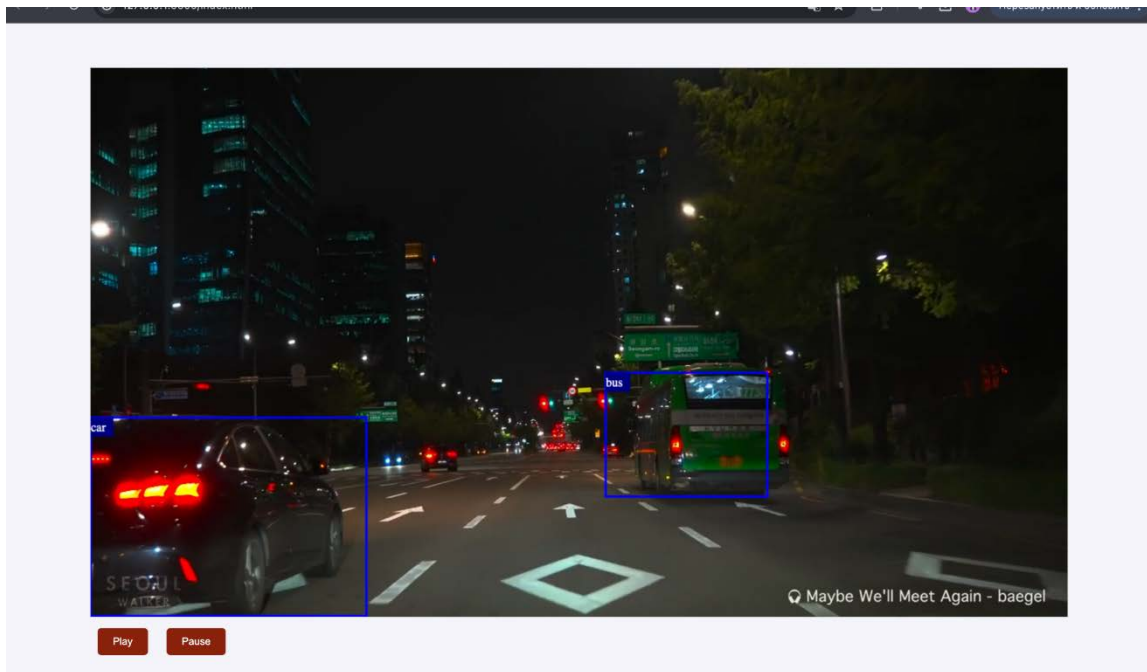


Рис.3.6 - тестування системи вночі скрін 1

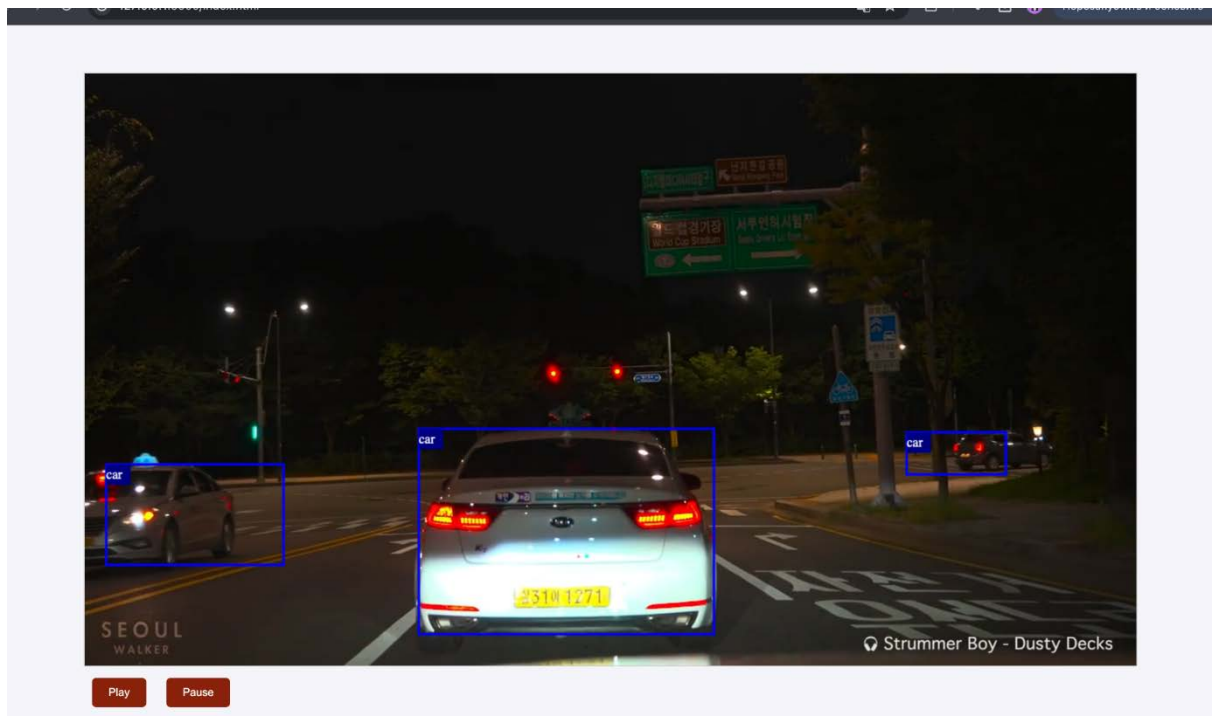


Рис.3.7 - тестування системи вночі скрін 2

Аналіз результатів: Вдень система справляється зі своєю задачею “на всі 100”, якщо в кадрі не знаходиться забагато об’єктів. В сутінках система також справляється доволі добре зі своєю задачею, але спостерігаються проблеми з

					БКР 122.006. ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		



детекцією об'єктів які мають досить погане освітлення. Вночі система справляється з детекцією об'єктів, але при недостатньому освітленні, особливо це помітно коли об'єкти знаходяться на віддаленні й мають основний чорний колір.

### Сценарій 2: Різноманітність об'єктів

- Мета: Оцінка точності системи у виявленні різних типів об'єктів.
- Опис: Тестування системи на здатність ідентифікувати пішоходів, транспортні засоби та тварин, щоб підтвердити універсальність детекції.

### Результати тестування:

#### 1) Різноманітність об'єктів в кадрі на вулиці:

- Кількість тестових відео: 2
- Виявлені об'єкти: 95% точність детекції
- Проблеми: Проблеми спостерігаються при великому скупченні об'єктів в одному кадрі.
- Візуалізація результатів:

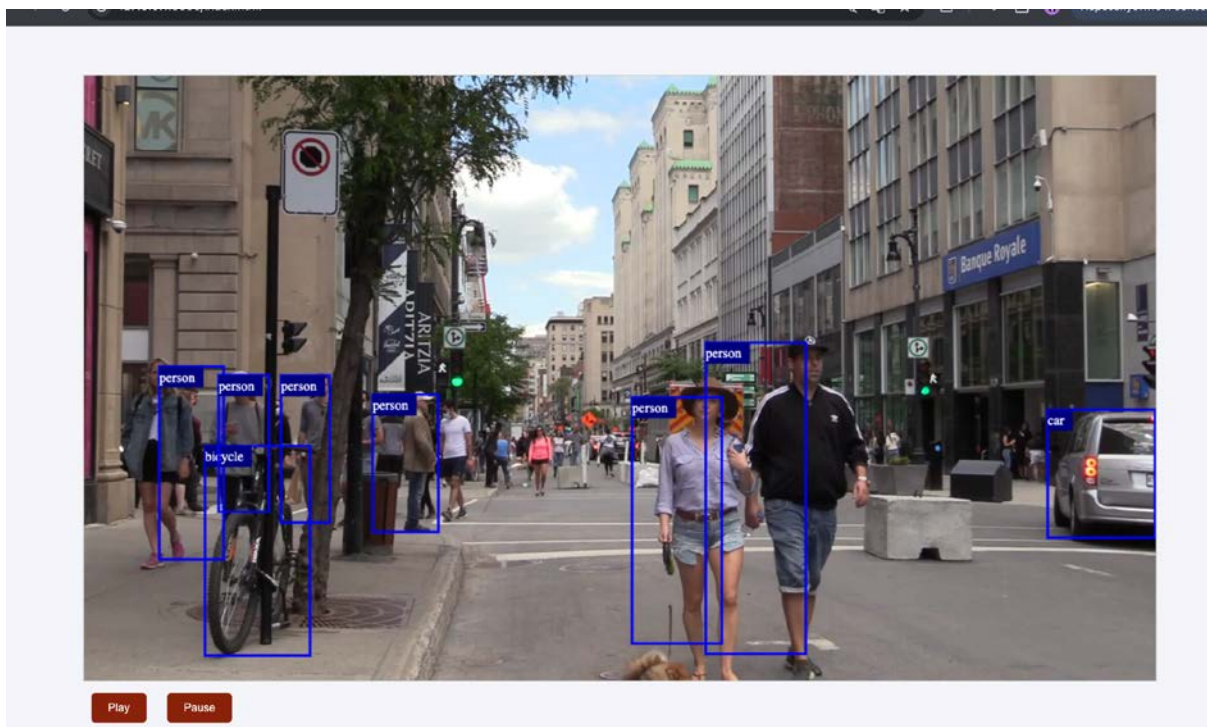


Рис.3.8 - тестування системи з різними об'єктами в кадрі, скрін 1

					БКР 122.006. ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

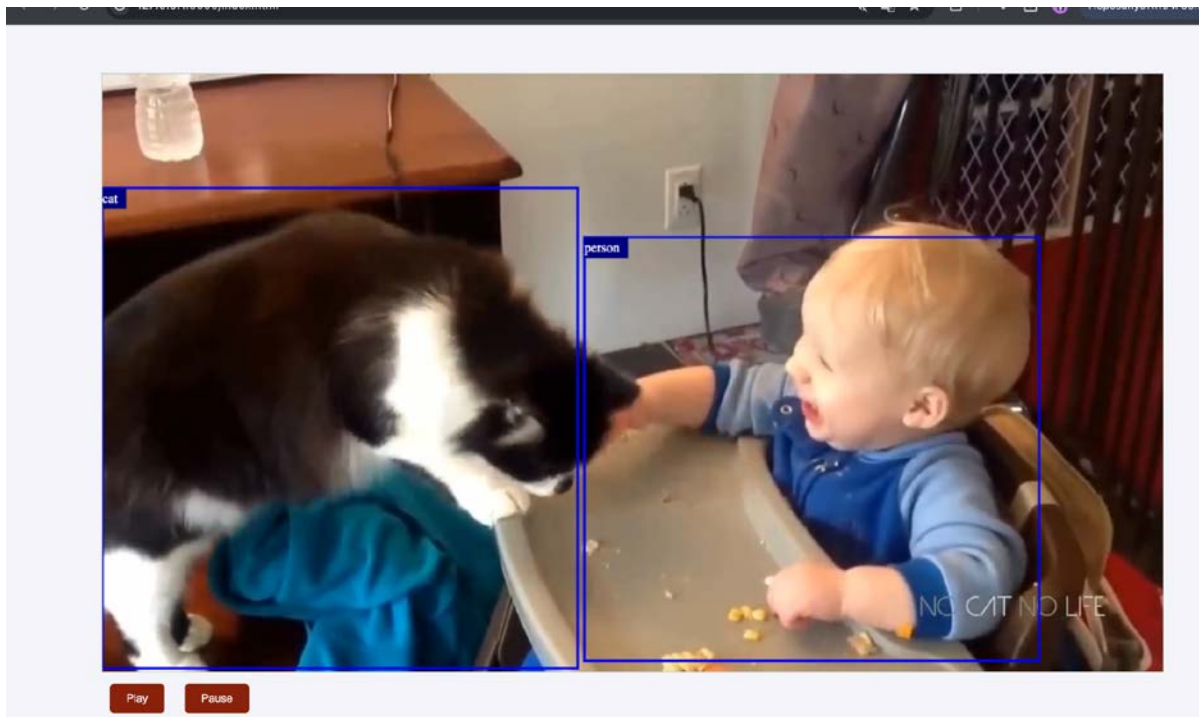


Рис. 3.9 - детекція різноманітних об'єктів в одному кадрі

Аналіз результатів: Система має доволі високу точність при детекції різних об'єктів в одному кадрі.

### Сценарій 3: Динамічні сцени

- Мета: Випробування системи в умовах динамічних змін у відеокадрах.
- Опис: Оцінка ефективності системи при виявленні та слідкуванні за об'єктами, які швидко рухаються, або коли камера змінює положення.

Результати тестування:

#### 1) Динамічні сцени в кадрі:

- Кількість тестових відео: 2
- Виявлені об'єкти: 73% точність детекції
- Проблеми: При швидкій зміні ракурсу камери системі потрібен деякий час для детекції об'єкта в кадрі, також не завжди точний контур при різкій появі об'єкта в кадрі.

- Візуалізація результатів:

					БКР 122.006. ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

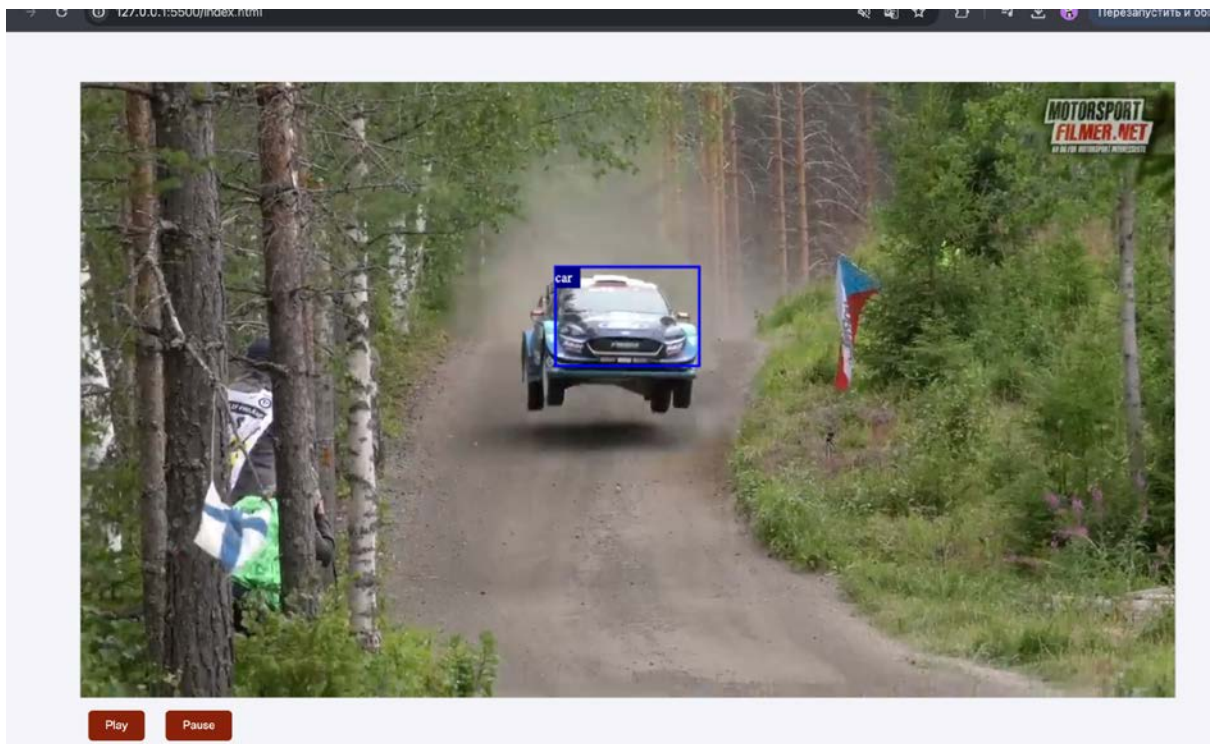


Рис.3.10 - різка поява об'єкта в кадрі й детекція його



Рис.3.11 - різка поява об'єкта в кадрі

					БКР 122.006. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

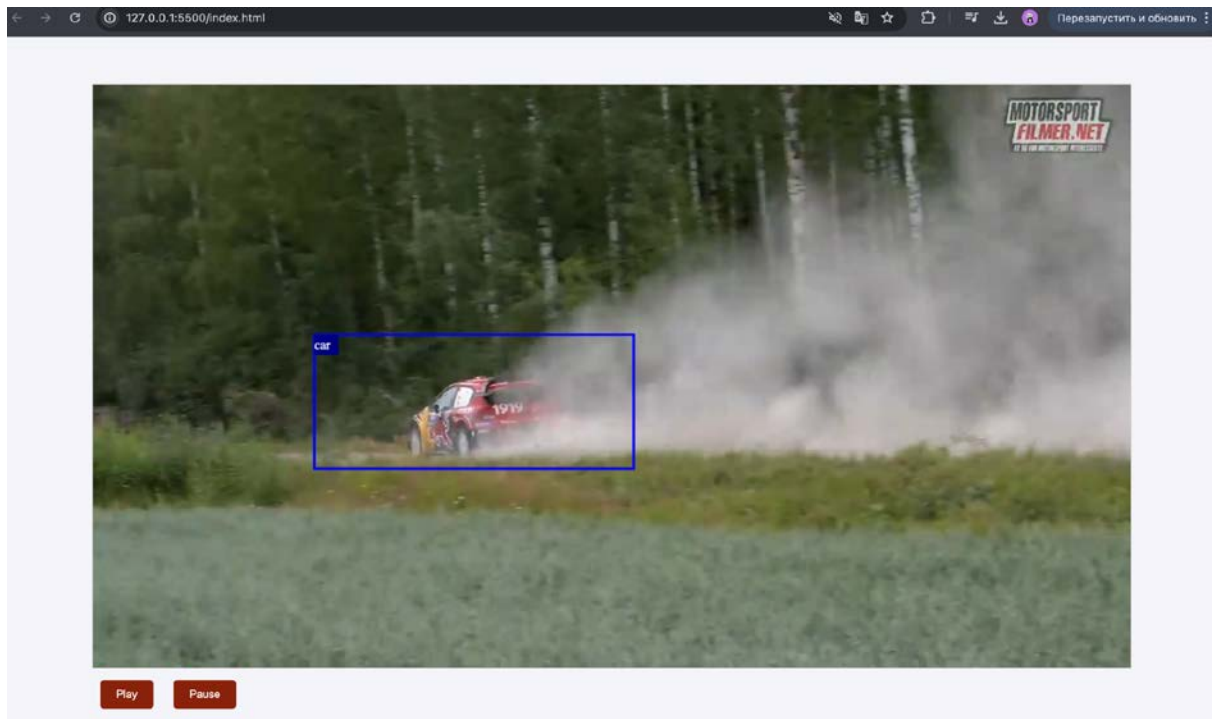


Рис.3.12 - детекція об'єкта після різкої зміни плану й кадру

Аналіз результатів: система доволі точно знаходить об'єкти в кадрі й швидко їх детектить, але при дуже швидкій зміні ракурсу камери системі потрібен мінімальний час для детекції.

#### Сценарій 4: Варіабельність погодних умов

- Мета: Перевірка надійності системи в різних погодних умовах.
- Опис: Тестування системи в умовах дощу та снігу для оцінки здатності системи розпізнавати об'єкти незалежно від погоди.

#### Результати тестування:

##### 1) Тестування системи в умовах дощу і снігу:

- Кількість тестових відео: 3
- Виявлені об'єкти: 65% точність детекції
- Проблеми: проблеми з'являються тільки якщо недостатньо освітлення.
- Візуалізація результатів:

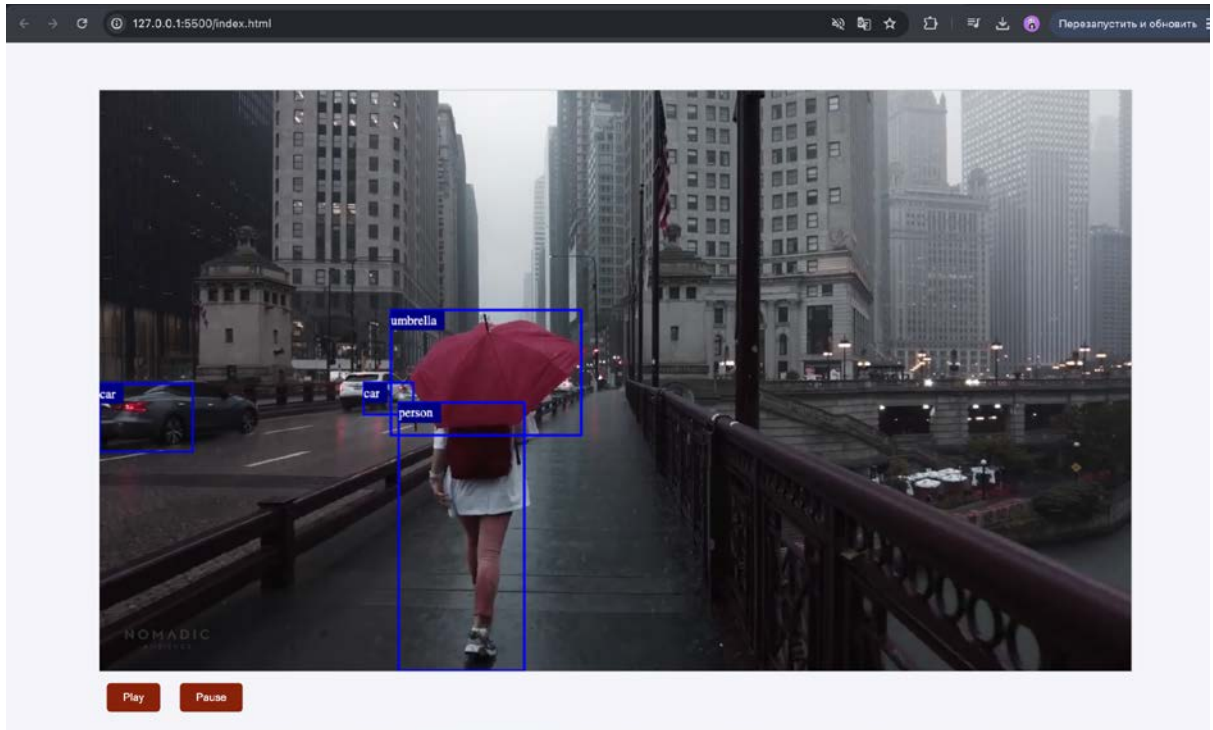


Рис.3.13 - тестування системи в дощову погоду скрін 1.

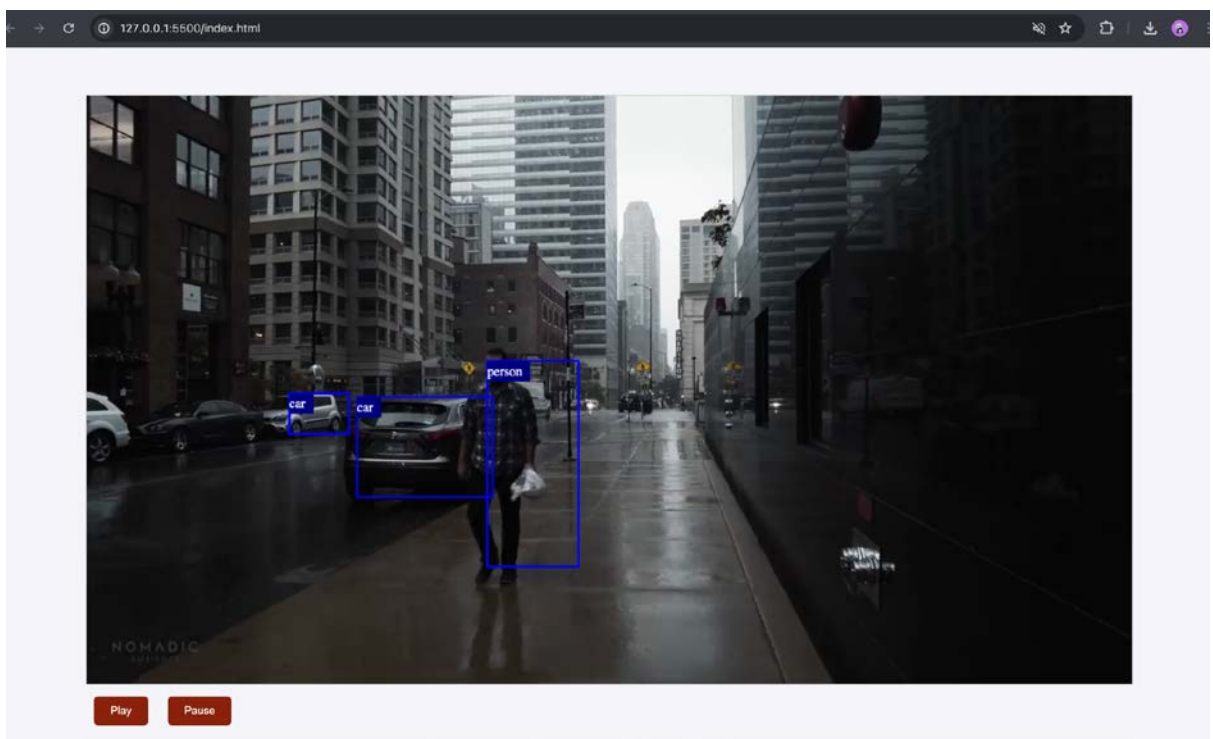


Рис.3.14 - тестування системи в дощову погоду скрін 2.

					БКР 122.006. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

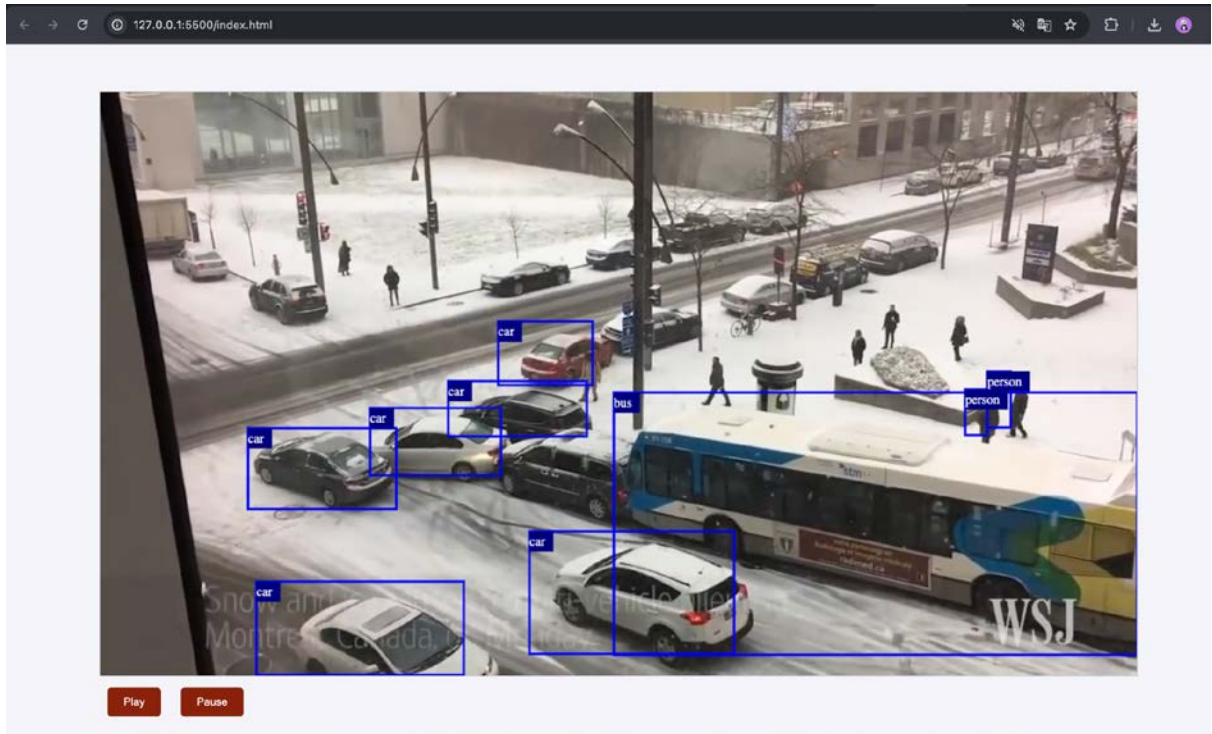


Рис.3.15 - тестування в умовах снігу в кадрі скрін 1

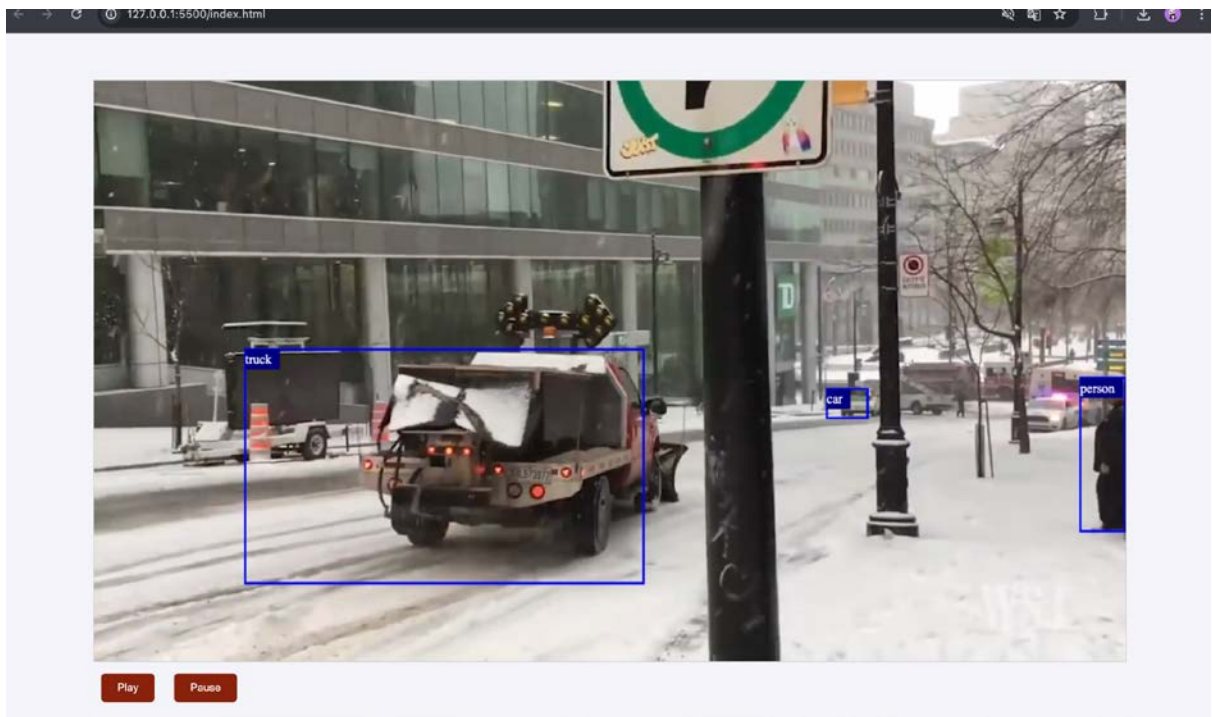


Рис.3.16 - тестування в умовах снігу в кадрі скрін 2

Аналіз результатів: різні погодні умови не дуже сильно впливають на швидкість та якість детекції об'єктів на відео, можливі проблеми лише при поганому освітленні.

					БКР 122.006. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

## Сценарій 5: Взаємодія та перекриття об'єктів

- Мета: Оцінка здатності системи розрізняти та точно локалізувати об'єкти, що перекриваються.

- Опис: Аналіз як система розрізняє окремі об'єкти в масивних або переповнених сценах, особливо в урбаністичних умовах.

Результати тестування:

### 1) Перекриття об'єктів:

- Кількість тестових відео: 1

- Виявлені об'єкти: 65% точність детекції

- Проблеми: при великому об'ємі об'єктів в кадрі система не встигає всіх відстежити через дуже великий потік даних.

- Візуалізація результатів:

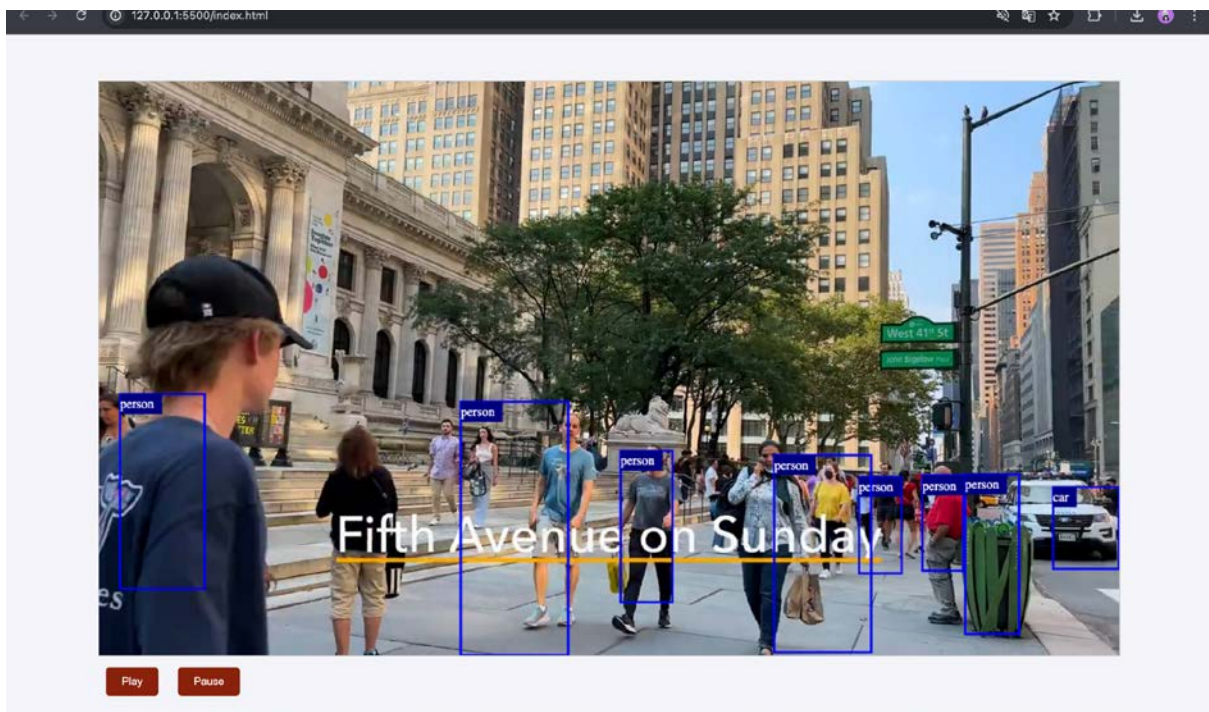


Рис.3.17 - тестування системи в переповнених сценах скрін 1

					БКР 122.006. ПЗ	Арк.
						55
Зм.	Арк.	№ докум.	Підпис	Дата		

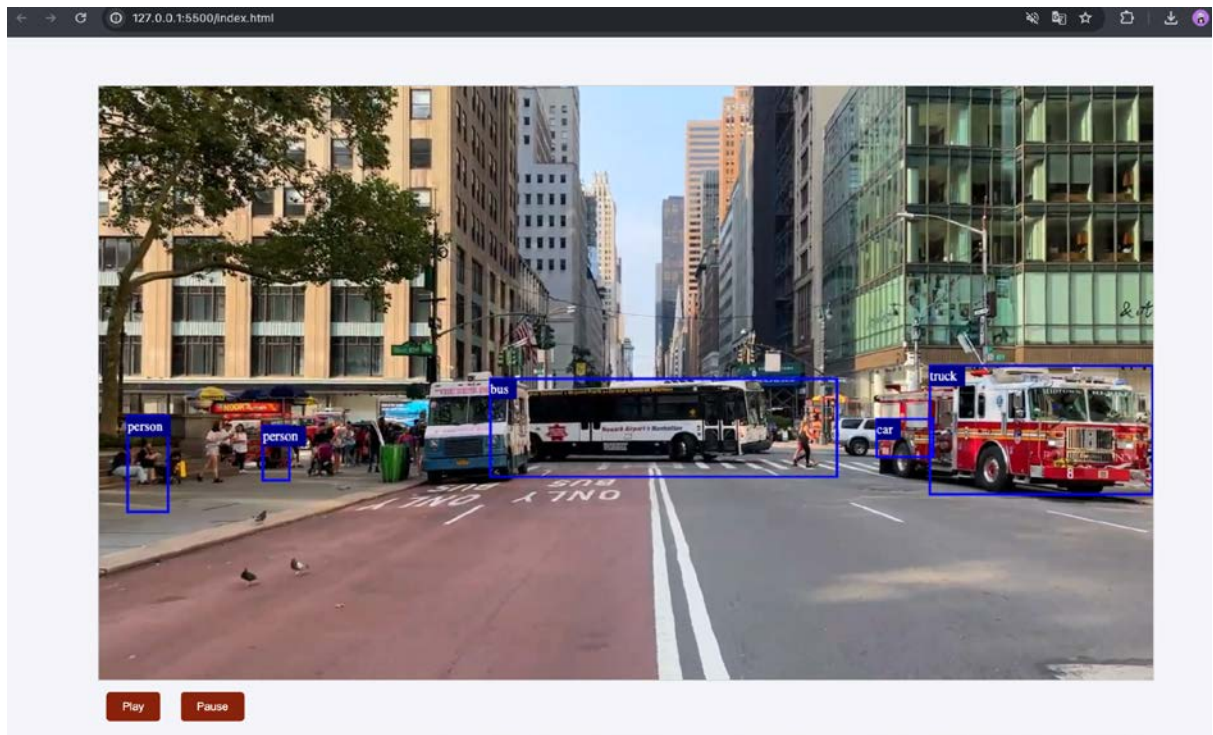


Рис.3.18 - тестування системи з перекриттям об'єктів в кадрі

Аналіз результатів: система майже повністю справляється з великим потоком об'єктів в кадрі та також дуже добре розрізняє об'єкти якщо вони закриті/перекриті іншим об'єктом.

#### Сценарій 6: Тестування стійкості до шуму

- Мета: Оцінка ефективності системи при високому рівні шуму на відео.
- Опис: Випробування системи на відеоматеріалах з різними рівнями шуму, щоб визначити, наскільки добре система може відфільтрувати помилкові сигнали.

#### Результати тестування:

##### 1) Перекриття об'єктів:

- Кількість тестових відео: 3
- Виявлені об'єкти: 70% точність детекції
- Проблеми: при наявності великого об'єму шумів на відео система погано детектить об'єкти.
- Візуалізація результатів:



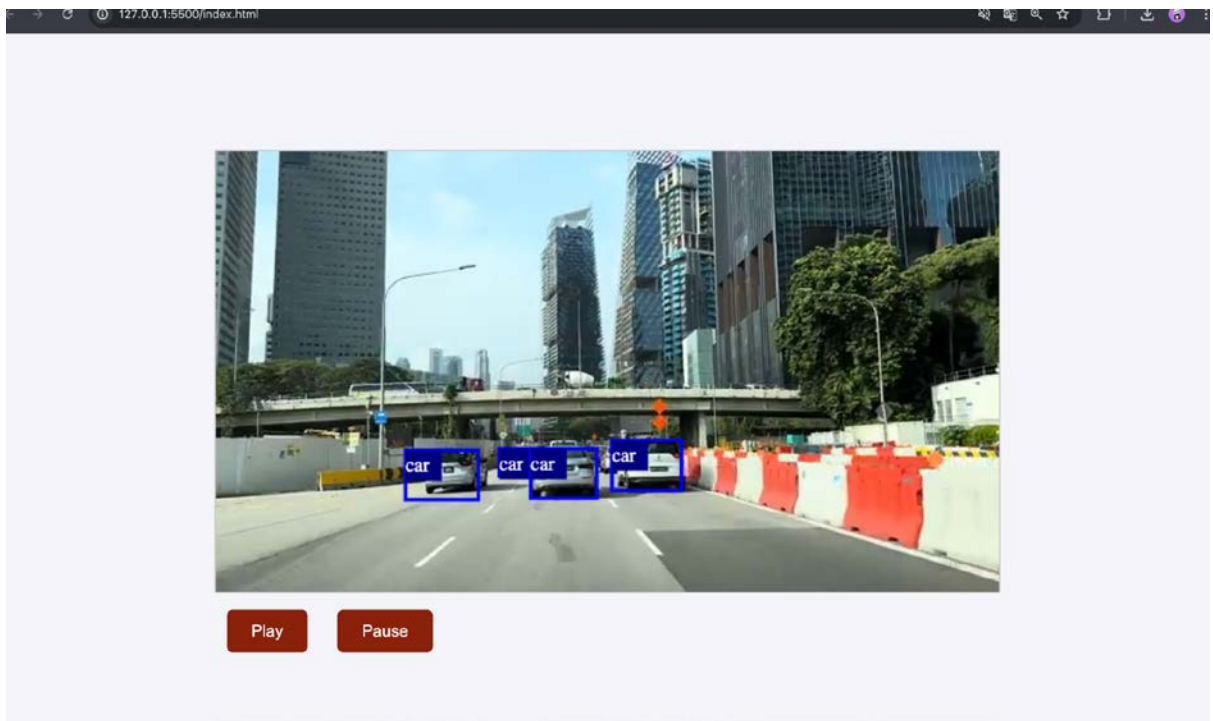


Рис.3.19 - тест системи з невеликою кількістю шумів

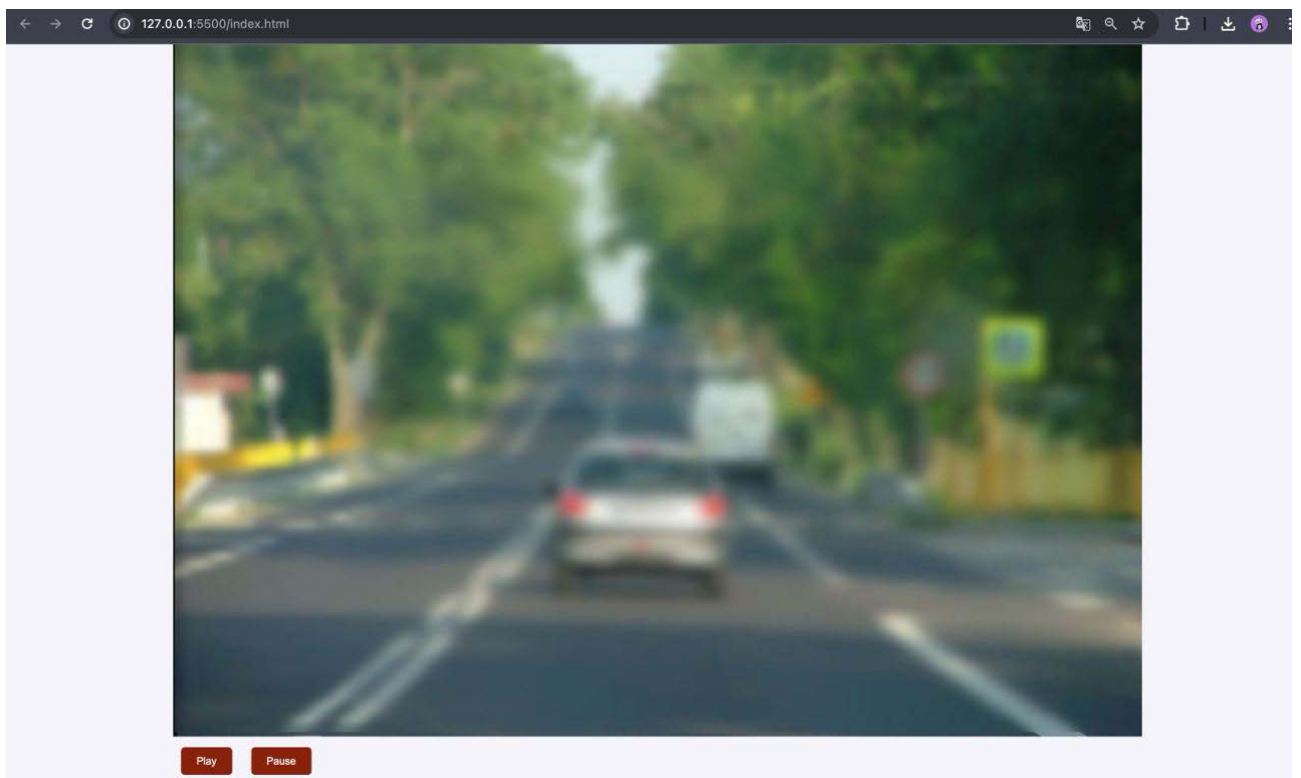


Рис.3.20 - тест системи з великою кількістю шумів

					БКР 122.006. ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

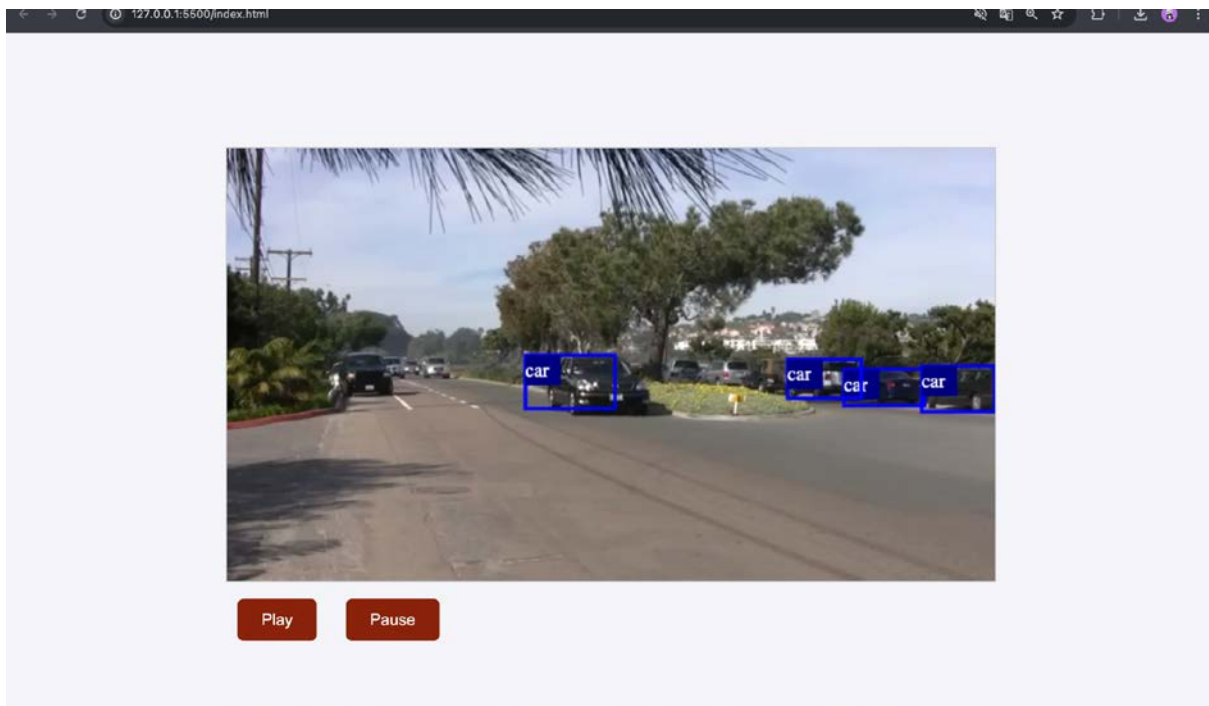


Рис.3.21 - тест системи з більшою кількістю шумів

Аналіз результатів: при використанні системи в відео з невеликою кількістю шумів - результат детекції майже не змінився, але якщо використовувати відео с величезною кількістю шумів - результат 0, але, на мою думку, тут ніяка система б не змогла задетектики об'єкт з такою кількістю шумів в кадрі.

### 3.5 Рекомендації щодо подальшого вдосконалення системи

У процесі розробки системи детекції об'єктів на відео було досягнуто значних результатів, проте є кілька напрямків, які можна вдосконалити для підвищення її ефективності та зручності у використанні. Нижче наведено основні рекомендації щодо подальшого розвитку системи:

#### 1)Покращення точності детекції:

- Оптимізація моделі YOLO v8: Використання більш точних параметрів при тренуванні моделі YOLO v8 для зменшення кількості хибних спрацьовувань та пропущених об'єктів.

					БКР 122.006. ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

- Використання додаткових даних для навчання: Розширення датасету для тренування моделі, включаючи різноманітні умови освітлення, кути огляду та типи об'єктів, щоб покращити загальну точність детекції.

## 2) Покращення продуктивності:

- Зменшення затримок обробки: Оптимізація коду JavaScript у файлі `object_detector.js` для підвищення швидкості обробки відеокадрів. Це може включати використання більш ефективних алгоритмів обробки зображень та багатопотоковість.

- Покращення роботи веб-воркера: Збільшення швидкості обробки у файлі `worker.js` шляхом оптимізації взаємодії з моделлю та зменшення обчислювальних витрат.

- Підвищення швидкості візуалізації: Оптимізація процесу відтворення рамок навколо об'єктів, щоб забезпечити їх своєчасне відображення навіть при швидкому русі об'єктів. Важливо зменшити затримки між детекцією та візуалізацією, щоб рамки не відставали від об'єктів.

## 3) Покращення користувацького інтерфейсу:

- \*нтерактивні елементи: Додавання більше інтерактивних елементів, таких як слайдери для налаштування параметрів детекції, кнопки для збереження результатів та області інтересу для детекції.

- Підтримка різних форматів відео: Додавання підтримки різних відеоформатів та джерел, включаючи стрімінгові відео та відео з локальних файлів.

- Завантаження відео з браузера: Додавання можливості завантажувати відео безпосередньо з браузера з подальшою коректною обробкою.

## 4) Розширення функціональності:

- Статистика об'єктів: Додавання можливості збирати та відображати статистику щодо кількості виявлених об'єктів на відео. Це дозволить користувачам отримувати додаткову інформацію про вміст відео.

					БКР 122.006. ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

- Аналіз поведінки об'єктів: Додавання функціоналу для аналізу поведінки детектованих об'єктів, що може бути корисним у відеоспостереженні та безпеці.

5) Покращення безпеки:

- Захист даних: Впровадження шифрування для збереження та передачі відеоданих, щоб забезпечити конфіденційність та цілісність інформації.

- Аутентифікація користувачів: Додавання системи аутентифікації та авторизації для контролю доступу до системи та її налаштувань.

6) Покращення документації:

- Розширення технічної документації: Докладний опис усіх функцій та компонентів системи, що полегшить інтеграцію та модифікацію коду іншими розробниками.

- Інструкції для користувачів: Створення інструкцій та посібників для кінцевих користувачів, що допоможе швидко освоїтись з системою та її можливостями.

7) Мобільна адаптація:

- Оптимізація для мобільних пристроїв: Забезпечення адаптивного дизайну інтерфейсу для зручного використання на мобільних пристроях, таких як смартфони та планшети.

- Підтримка мобільних браузерів: Тестування та оптимізація роботи системи в популярних мобільних браузерах для забезпечення кросплатформенної сумісності.

**Висновки з третього розділу**

- Використання сучасних технологій та інструментів забезпечило створення ефективної, гнучкої та надійної системи детекції об'єктів на відео.

- Поєднання JavaScript, HTML5 Canvas та асинхронних веб-воркерів дозволило створити швидку та інтерактивну клієнтську частину.

					БКР 122.006. ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

- Серверні інструменти, такі як Visual Studio Code, забезпечили стабільність та масштабованість проекту, дозволяючи швидко та ефективно проводити розробку та тестування.

- Використання WebAssembly та ONNX Runtime для виконання моделей машинного навчання у браузері дозволило досягти високої продуктивності обчислень, що є критично важливим для обробки відеоданих у реальному часі.

Аналізуючи представлені результати тестування по кожному сценарію, можна зробити висновок, що система демонструє високу ефективність у розпізнаванні об'єктів при достатньому освітленні, але виявляє труднощі при низькій освітленості та високому рівні шуму. Це вказує на потенційні напрямки для покращення системи, зокрема у сфері обробки відео в умовах поганого освітлення та оптимізації алгоритмів для більш ефективної роботи в "шумних" умовах. Наведені результати підтверджують важливість подальших досліджень та розвитку системи для забезпечення її стійкості та надійності в різних операційних умовах.

Впровадження рекомендацій дозволить підвищити ефективність, зручність та безпеку системи детекції об'єктів на відео, що сприятиме її більш широкому використанню в різних сферах діяльності.

					БКР 122.006. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

## 4 ОХОРОНА ПРАЦІ ТА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

### 4.1 Аналіз потенційних ризиків та загроз для користувачів програмної системи

Потенційні ризики та загрози[3]:

1) Фізичні ризики:

Травматизм під час монтажу обладнання. Може виникнути при встановленні та налагодженні обладнання для обробки відео, зокрема камер та серверів. Пожежна небезпека. Використання електронного обладнання підвищує ризик виникнення пожежі через можливі короткі замикання або перегрів компонентів.

2) Ергономічні ризики:

Неправильна організація робочого місця. Відсутність ергономічних умов для користувачів комп'ютерних систем може призвести до хронічних захворювань опорно-рухового апарату та зору. Тривала робота з відеоданими. Висока візуальна напруга та стрес під час аналізу відеопотоку можуть викликати втоми та зниження працездатності.

3) Психологічні ризики:

Психоемоційне навантаження. Постійне спостереження за відеопотоком та аналіз детекцій може спричинити психоемоційний стрес у користувачів.

Заходи зниження ризиків:

1) Виконання вимог ДСТУ та законодавства[2]:

Дотримання вимог ДСТУ ISO 45001:2019 "Системи управління охороною здоров'я та безпекою праці. Вимоги". Виконання положень Закону України "Про охорону праці" та Кодексу законів про працю (КЗпП).

2) Організаційні заходи:

Проведення регулярних інструктажів з охорони праці для всіх користувачів системи. Створення та підтримка належних умов праці, включаючи ергономічні меблі, оптимальне освітлення та відповідну температуру в

					БКР 122.006. ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

приміщенні. Забезпечення регулярних перерв для зниження втоми та підвищення ефективності праці.

3) Технічні заходи:

Використання якісного та сертифікованого обладнання, що відповідає вимогам пожежної безпеки. Встановлення систем автоматичного пожежогасіння та димової сигналізації. Забезпечення надійного електроживлення та заземлення всіх електронних компонентів.

4) Психологічна підтримка:

Впровадження програм психологічної підтримки для користувачів, що працюють у стресових умовах. Організація тренінгів з управління стресом та покращення психоемоційного стану.

#### 4.2 Заходи з охорони праці та безпеки користування

Основні заходи з охорони праці[1][3]:

1) Ергономічний дизайн інтерфейсу:

Інтерфейс користувача розроблено з урахуванням ергономічних принципів, що забезпечують зручність і безпеку використання. Використання великих кнопок та зрозумілих іконок для полегшення взаємодії. Забезпечення можливості налаштування кольорової схеми та розміру шрифтів для користувачів з різними зоровими потребами.

2) Освітлення робочого місця:

Вимоги до освітлення робочого місця, де використовується система, повинні відповідати стандартам ДСТУ EN 12464-1:2014 «Світло та освітлення. Освітлення робочих місць в приміщеннях». Забезпечення достатнього рівня освітлення для зниження навантаження на зір користувачів.

3) Контроль рівня шуму:

Рівень шуму на робочому місці не повинен перевищувати допустимі норми, встановлені ДСТУ ISO 1999:2009 «Акустика. Визначення та оцінювання впливу шуму на слух». Використання обладнання, що працює тихо, для зниження рівня шуму на робочому місці.

					БКР 122.006. ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

#### 4) Забезпечення зручності робочого місця:

Робочі місця повинні бути обладнані зручними кріслами з регулюванням висоти, що відповідають вимогам ДСТУ ГОСТ 12.2.032-78 «Система стандартів безпеки праці. Робочі місця при виконанні робіт сидячи. Загальні ергономічні вимоги». Забезпечення можливості регулювання моніторів за висотою та нахилом для зручного огляду.

Технічні заходи безпеки[2]:

##### 1) Захист даних:

Використання сучасних засобів захисту інформації, включаючи шифрування даних, автентифікацію користувачів та регулярні оновлення програмного забезпечення. Відповідність системи вимогам ДСТУ ISO/IEC 27001:2015 «Інформаційна технологія. Методи та засоби забезпечення безпеки. Системи менеджменту інформаційної безпеки. Вимоги».

##### 2) Регулярне оновлення системи:

Регулярне оновлення програмного забезпечення для виправлення виявлених вразливостей та покращення функціоналу. Забезпечення автоматичних оновлень для зниження ризику використання застарілого програмного забезпечення.

##### 3) Безпека користувацьких даних:

Обмеження доступу до конфіденційної інформації лише авторизованим користувачам. Впровадження політики резервного копіювання даних для запобігання втраті інформації у разі збою системи.

##### 4) Захист від шкідливого програмного забезпечення:

Використання антивірусних програм та інших засобів захисту від шкідливого програмного забезпечення. Регулярне сканування системи на наявність вірусів та шкідливих програм.

Організаційні заходи безпеки:

##### 1) Інструктажі та навчання:

					БКР 122.006. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64



Проведення регулярних інструктажів з охорони праці та безпеки користувачів. Навчання користувачів основам безпечної роботи з програмною системою.

2) Моніторинг та оцінка ризиків:

Постійний моніторинг використання системи для виявлення потенційних загроз та ризиків. Оцінка ризиків з метою впровадження додаткових заходів захисту у разі необхідності.

3) Розробка плану дій у надзвичайних ситуаціях:

Розробка та впровадження плану дій у випадку виникнення надзвичайних ситуацій, включаючи інструкції щодо евакуації та надання першої допомоги.

### 4.3 Утилізація комп'ютерної техніки

1) Законодавчі вимоги та стандарти

Закон України «Про відходи» визначає загальні принципи та вимоги щодо поводження з відходами, включаючи електронні та електричні відходи. Згідно з цим законом, підприємства та організації зобов'язані організувати утилізацію електронного обладнання відповідно до екологічних стандартів.

ДСТУ ISO 14001:2015 «Системи екологічного управління. Вимоги та настанови щодо застосування» регламентує вимоги до систем екологічного управління, що включають аспекти утилізації електронних відходів. Виконання цих вимог сприяє зменшенню негативного впливу на навколишнє середовище та забезпечує безпечне поводження з небезпечними відходами.

2) Процес утилізації комп'ютерної техніки

Процес утилізації комп'ютерної техніки включає кілька етапів:

Збір та транспортування:

Непридатна до використання техніка збирається та транспортується до спеціалізованих пунктів прийому або підприємств з утилізації електронних відходів. Збір техніки може здійснюватися через організацію пунктів прийому на підприємствах або шляхом співпраці з ліцензованими компаніями, що займаються утилізацією.

					БКР 122.006. ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

### Діагностика та демонтаж:

Після надходження техніки до пункту утилізації проводиться її діагностика для визначення можливості вторинного використання компонентів. Після цього здійснюється демонтаж техніки на складові частини, що включає вилучення цінних матеріалів та небезпечних компонентів, які потребують спеціального поводження.

### Переробка та утилізація:

Комп'ютерна техніка розбирається на складові елементи, зокрема метали, пластмаси та електронні компоненти. Небезпечні відходи, такі як батареї та деякі електронні компоненти, підлягають спеціальній утилізації відповідно до вимог ДСТУ 4219:2003 «Відходи електронного та електричного обладнання. Загальні вимоги до збирання, транспортування, зберігання та оброблення». Переробка металів та пластмас здійснюється з метою їх повторного використання у виробництві.

### 3) Екологічні аспекти утилізації

Важливим аспектом утилізації є мінімізація негативного впливу на навколишнє середовище. Використання сучасних технологій переробки та утилізації дозволяє зменшити кількість відходів, що потрапляють на сміттєзвалища, та забезпечити вторинне використання цінних матеріалів. Відповідно до ДСТУ ISO 14024:2002 «Екологічні маркування та декларації. Екологічне маркування I типу. Принципи та методи», підприємства повинні впроваджувати екологічно чисті технології та сприяти вторинному використанню матеріалів.

### 4) Організаційні заходи

Для забезпечення ефективної утилізації комп'ютерної техніки підприємства та організації повинні розробити та впровадити політику екологічного управління, що включає:

Впровадження системи збору та утилізації техніки на підприємстві:

					БКР 122.006. ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

Організація системи збору використаної техніки, встановлення спеціальних контейнерів для електронних відходів, що забезпечують їх безпечне зберігання та транспортування до пунктів утилізації.

Проведення регулярних інструктажів та навчання працівників щодо правил поводження з електронними відходами:

Проведення навчальних семінарів та інструктажів з питань екологічної безпеки та утилізації електронного обладнання. Це сприятиме підвищенню обізнаності працівників щодо екологічних ризиків та методів їх мінімізації.

Співпраця з ліцензованими підприємствами, що займаються утилізацією та переробкою електронних відходів:

Укладення договорів з ліцензованими компаніями, що спеціалізуються на утилізації електронних відходів, для забезпечення дотримання екологічних стандартів та вимог законодавства.

Моніторинг та оцінка ефективності заходів з утилізації з метою їх подальшого вдосконалення:

Проведення регулярного моніторингу процесу утилізації, аналіз ефективності впроваджених заходів та їх подальше вдосконалення для підвищення екологічної безпеки.

### **Висновки з четвертого розділу**

У розділі розглянуто основні аспекти забезпечення безпеки та здоров'я користувачів програмної системи, а також її впливу на навколишнє середовище. Враховуючи потенційні ризики, пов'язані з використанням програмного забезпечення, зокрема ергономічні, психологічні, електричні та кібербезпекові загрози, визначено необхідність дотримання відповідних стандартів ДСТУ. Серед них: ДСТУ EN 12464-1:2014 (освітлення робочих місць), ДСТУ ISO 1999:2009 (вплив шуму), ДСТУ ГОСТ 12.2.032-78 (ергономічні вимоги) та ДСТУ ISO/IEC 27001:2015 (інформаційна безпека).

Запропоновані заходи охорони праці включають рекомендації щодо забезпечення безпеки робочого середовища, електробезпеки, кібербезпеки та

					БКР 122.006. ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підпис	Дата		

ергономіки. Регулярні інструктажі та навчання користувачів відіграють ключову роль у мінімізації ризиків. Дотримання цих заходів сприятиме створенню безпечних умов праці та захисту здоров'я користувачів, а також мінімізує негативний вплив на навколишнє середовище.

Розглянуто також утилізацію комп'ютерної техніки. Відповідно до ДСТУ ISO 14001:2015 та інших екологічних стандартів, утилізація повинна мінімізувати негативний вплив на довкілля та забезпечити безпечне поводження з небезпечними відходами. Процес утилізації включає збір, транспортування, діагностику, демонтаж, переробку та утилізацію компонентів, дотримуючись законодавчих вимог та стандартів екологічної безпеки.

Виконання рекомендацій розділу дозволить забезпечити високий рівень безпеки та захисту при експлуатації програмного забезпечення, що є важливим для сталого розвитку та підвищення якості життя.

					БКР 122.006. ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

## ЗАГАЛЬНІ ВИСНОВКИ

У рамках даної роботи було розроблено та впроваджено систему пошуку та детекції об'єктів на відео з використанням сучасних алгоритмів машинного навчання та технологій веб-розробки. Основною метою дослідження було створення ефективної системи, здатної працювати в реальному часі та забезпечувати високу точність детекції.

Розпочавши з аналізу існуючих підходів до детекції об'єктів, було обрано модель YOLO v8 завдяки її здатності забезпечувати високу продуктивність і точність. Було здійснено інтеграцію моделі у веб-середовище з використанням JavaScript, що дозволило створити інтерактивний клієнтський додаток.

В процесі реалізації було вирішено кілька ключових завдань:

1) Інтеграція веб-воркерів для асинхронної обробки відео, що дозволило значно знизити навантаження на основний потік браузера і покращити відгук системи.

2) Використання HTML5 Canvas для візуалізації результатів детекції, що забезпечило зручність і наочність інтерфейсу користувача.

3) Розробка алгоритмів обробки відеокадрів та оптимізації продуктивності, що дозволило забезпечити стабільну роботу системи в реальному часі навіть при високому навантаженні.

У ході експериментальної апробації прототипу програми було проведено тестування на різних відеоданих, що дозволило підтвердити ефективність та надійність розробленої системи. Система продемонструвала високу точність виявлення та класифікації об'єктів, а також здатність працювати в різних умовах освітлення та з різними типами відеопотоків.

На основі проведеного дослідження було сформульовано рекомендації щодо подальшого вдосконалення системи, включаючи оптимізацію продуктивності, додавання нових функцій та розширення можливостей інтеграції.

Здійснено аналіз потенційних ризиків та загроз для користувачів програмної системи відповідно до стандартів охорони праці. Впроваджено

					БКР 122.006. ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підпис	Дата		

заходи з охорони праці та безпеки користування, що забезпечують надійність та безпеку експлуатації системи. Розглянуто питання утилізації комп'ютерної техніки відповідно до екологічних стандартів, що дозволяє мінімізувати негативний вплив на довкілля та забезпечити безпечне поводження з небезпечними відходами. Виконання цих заходів сприяє створенню безпечних умов праці, захисту здоров'я користувачів та сталому розвитку.

Загальний підсумок роботи показав, що розроблена система відповідає сучасним вимогам до систем пошуку та детекції об'єктів на відео, демонструючи високу продуктивність, точність та зручність використання. Виконані завдання та отримані результати експериментальної апробації відкривають перспективи для подальшого розвитку та вдосконалення даної технології у різних сферах застосування.

					БКР 122.006. ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ПОСИЛАНЬ

- 1) Закон України "Про охорону праці". Від 14 жовтня 1992 року № 2694-ХІІ. (дата звернення: 04.05.2024).
- 2) Конституція України. Від 28 червня 1996 року. (дата звернення: 01.05.2024).
- 3) Зуєв, А. М. Основи охорони праці: конспекти лекцій. Київ: НМЦ, 2019. 216 с. (дата звернення: 30.04.2024).
- 4) Веб-воркери для асинхронної обробки даних. URL: <https://www.html5rocks.com/en/tutorials/workers/basics/> (дата звернення: 16.05.2024).
- 5) Веб-технології для обробки відеоданих. URL: <https://www.html5rocks.com/en/tutorials/video/basics/> (дата звернення: 07.04.2024).
- 6) Використання HTML5 Canvas для обробки зображень. URL: [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API) (дата звернення: 04.04.2024).
- 7) Документація JavaScript. URL: <https://developer.mozilla.org/uk/docs/Web/JavaScript> (дата звернення: 08.04.2024).
- 8) Документація Microsoft ONNX Runtime for Web. URL: <https://microsoft.github.io/onnxruntime/> (дата звернення: 01.05.2024).
- 9) Документація ONNX Runtime. URL: <https://onnxruntime.ai> (дата звернення: 05.04.2024).
- 10) Документація Web Workers. URL: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Workers\\_API/Using\\_web\\_workers](https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers) (дата звернення: 29.04.2024).
- 11) Використовуємо CNN (Convolutional Neural Networks) для обробки зображень. Частина перша. URL: <https://dou.ua/forums/topic/48368/> (дата звернення: 15.04.2024).
- 12) Нейромережі, штучний інтелект, машинне навчання розробка і впровадження. URL: <https://tqm.com.ua/ua/likbez/interview-ua/shtuchnyj-intelekt-v-ukrayini-oles-petryv> (дата звернення: 10.04.2024).

					БКР 122.006. ПЗ	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дата		

13) Розробка систем детекції об'єктів на основі YOLO // Матеріали міжнародної конференції "Інформаційні технології та комп'ютерна інженерія". 2020. С. 102-110. (дата звернення: 17.04.2024).

14) Сидоренко, В. О. Інтелектуальні системи обробки інформації. Київ: Наукова думка, 2017. 320 с. (дата звернення: 20.04.2024).

15) Brownlee J. Introduction to Neural Networks // Medium. 2018. URL: <https://medium.com/deep-learning-demystified/introduction-to-neural-networks-part-1> (дата звернення: 10.04.2024).

16) Introduction to Object Detection Algorithms // Medium. URL: <https://medium.com/@abuabdul4dev/introduction-to-object-detection-algorithms-5a54e358b0b8> (дата звернення: 02.04.2024).

17) Krizhevsky A., Sutskever I., Hinton G. ImageNet Classification with Deep Convolutional Neural Networks // Communications of the ACM. 2017. Vol. 60, No. 6. P. 84-90. URL: <https://dl.acm.org/doi/10.1145/3065386> (дата звернення: 13.04.2024).

18) Liu W. et al. Single Shot MultiBox Detector // arXiv. 2015. URL: <https://arxiv.org/abs/1512.02325> (дата звернення: 02.04.2024).

19) Redmon J., Farhadi A. YOLOv3: An Incremental Improvement // arXiv. 2018. URL: <https://arxiv.org/abs/1804.02767> (дата звернення: 25.04.2024).

20) Rehm G. Convolutional Neural Networks: A Comprehensive Guide // Medium. 2020. URL: <https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide> (дата звернення: 12.04.2024).

21) Understanding Convolutional Neural Networks: A Beginner's Journey into the Architecture // Medium. URL: <https://medium.com/codex/understanding-convolutional-neural-networks-a-beginners-journey-into-the-architecture-aab30dface10> (дата звернення: 01.04.2024).

					БКР 122.006. ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підпис	Дата		