

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 2022 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**РОЗПІЗНАВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ ЗА
ДОПОМОГОЮ ТЕХНОЛОГІЇ КОМП'ЮТЕРНОГО
ЗОРУ**

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810105

Виконала студентка 4-го курсу, групи 401
_____ *А. В. Вишневецька*
«13» червня 2022 р.

Керівник: канд. техн. наук, доцент
_____ *Г. В. Кондратенко*
«13» червня 2022 р.

Миколаїв – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 20__ р.

З А В Д А Н Н Я
на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Вишневецькій Анастасії Володимирівні.

1. Тема кваліфікаційної роботи «Розпізнавання транспортних засобів за допомогою технології комп'ютерного зору».

Керівник роботи Кондратенко Галіна Володимирівна, канд. техн. наук, доцент.

Затв. наказом Ректора ЧНУ ім. Петра Могили від « ____ » _____ 20__ р. № ____

2. Строк представлення кваліфікаційної роботи студентом « ____ » _____ 20__ р.

3. Вхідні (початкові) дані до роботи: відеозапис руху транспортних засобів на камері.

Очікуваний результат: система розпізнавання транспортних засобів за допомогою комп'ютерного зору.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

– аналіз предметної області та алгоритмів машинного навчання для розв'язку поставленої задачі;

– опис методів та алгоритмів для ідентифікування транспортних засобів;

– реалізація розпізнавання транспортних засобів за допомогою комп'ютерного зору;

– порівняльний аналіз з іншими моделями для розпізнавання транспорту.

5. Перелік графічного матеріалу: презентація.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	ст. викладач каф. екології Макарова О.В.	

Керівник роботи канд. техн. наук, доцент Кондратенко Г. В.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Завдання прийнято до виконання Вишневецька А.В.
(прізвище та ініціали)

_____ (підпис)

Дата видачі завдання « 27 » _____ листопада _____ 2021 р.

КАЛЕНДАРНИЙ ПЛАН
виконання бакалаврської кваліфікаційної роботи

Тема: Розпізнавання транспортних засобів за допомогою технології комп'ютерного зору

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників БКР	27.10.2021	30.10.2021	
2	Отримання завдання на виконання БКР	08.11.2021	10.11.2021	
3	Складання календарного плану роботи на весь період виконання БКР	06.12.2021	08.12.2021	
4	Отримання завдання на переддипломну практику	13.05.2022	17.05.2022	
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	23.05.2022	04.06.2022	
6	Розробка звіту з переддипломної практики	04.06.2022	06.06.2022	
7	Виконання БКР: опис методів та алгоритмів для ідентифікування транспортних засобів, розробка ПЗ	28.02.2022 та 06.06.2022	27.03.2022 та 19.06.2022	
8	Попередній захист БКР на засіданні комісії кафедри	30.05.2022	31.05.2022	
9	Доробка та остаточне оформлення БКР	02.06.2022	20.06.2022	
10	Подання БКР рецензенту	16.06.2022	18.06.2022	
11	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	20.06.2022	22.06.2022	
12	Захист БКР перед екзаменаційною комісією (ЕК)	27.06.2022	29.06.2022	

Розробила студентка Вишневецька А.В.
(прізвище, ім'я, по батькові студента) (підпис)

Керівник роботи канд. техн. наук, доцент Г. В. Кондратенко
(посада, прізвище, ім'я, по батькові) (підпис)

« 12 » _____ 12 _____ 2021 р.

АНОТАЦІЯ

**бакалаврської кваліфікаційної роботи студентки групи 401 ЧНУ ім. Петра
Могили**

Вишневецької Анастасії Володимирівни

**Тема: «Розпізнавання транспортних засобів за допомогою технології
комп'ютерного зору»**

Метою роботи є розпізнавання транспортних засобів за допомогою машинного навчання, а саме комп'ютерного зору.

Об'єктом розробки є процеси розпізнавання та ідентифікації транспортних засобів.

Предметом дослідження є методи та технології розпізнавання транспортних засобів, зокрема з використанням комп'ютерного зору.

Завданням у БКР є реалізація інформаційної системи для розпізнавання транспортних засобів за допомогою інструментів та бібліотек машинного навчання.

Робота складається з фахового розділу і спеціальної частини з охорони праці. Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі розглядається аналіз предметної області та алгоритмів машинного навчання для розв'язку поставленої задачі.

У другому розділі описано методи та алгоритми для ідентифікування транспортних засобів.

У третьому розділі реалізацівано розпізнавання транспортний засобів за допомогою комп'ютерного зору.

У четвертому розділі наведено порівняльний аналіз з іншими моделями для розпізнавання транспорту.

Бакалаврська кваліфікаційна робота містить 87 сторінок, 20 рисунків, 11 таблиць, 29 використаних джерел та 1 додаток.

Ключові слова: машинне навчання, комп'ютерний зір, фреймворк.

ABSTRACT

Bachelor's qualification work of a student of group 401 of the Petro Mogyla

Black Sea National University

Vyshnevetska Anastasiia Volodimirovna

Topic: "Vehicle recognition using computer vision technology"

The aim of the work is to recognize vehicles using machine learning, namely computer vision.

The object of development is the processes of vehicle recognition and identification.

The subject of the research is Methods and technologies of vehicle recognition, in particular using computer vision.

The task in The Bachelor's qualification work is to implement an information system for vehicle recognition using machine learning tools and libraries. The work consists of a professional section and a special part on labor protection. The explanatory note consists of an introduction, four sections, conclusions and appendices.

The first chapter discusses the analysis of the subject area and machine learning algorithms for solving this problem.

The second section describes methods and algorithms for identifying vehicles.

In the third chapter, recognition of transrotechnical means using computer vision is implemented.

The fourth section provides a comparative analysis of my and other transport recognition models.

The Bachelor's qualification work contains 87 pages, 20 figures, 11 tables, 29 sources used and 1 appendice.

Keywords: machine learning, computer vision, framework.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, МЕТОДІВ ТА АЛГОРИТМІВ	
МАШИННОГО НАВЧАННЯ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	
1.1 Основні поняття та визначення	9
1.2 Останні дослідження та публікації	11
1.3 Огляд існуючих аналогів	13
1.4 Постановка задачі.....	14
2 ОПИС МЕТОДІВ ТА ПІДХОДІВ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ	
РОЗПІЗНАВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ.....	
2.1 Роль глибинного навчання у розпізнаванні графічних об'єктів	16
2.2 Convolutional Neural Network.....	17
2.3 Recurrent Neural Network	20
2.3.1 Архітектура рекурентної нейронної мережі	21
2.3.2 Типи рекурентних нейронних мереж.....	22
2.3.3 Алгоритм зворотного розповсюдження помилки у часі.....	24
2.3.4 Проблеми рекурентної нейромережі	24
2.4 LP Детекція та Розпізнавання	25
2.5 Набори даних і протоколи оцінки	28
2.6 Порівняльний приклад систем ALPR – Автоматичних систем розпізнавання номерних знаків	29
2.7 Огляд функціоналу бібліотек C# для вирішення задачі розпізнавання.....	30
3 РЕАЛІЗАЦІЯ РОЗПІЗНАВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ ЗА	
ДОПОМОГОЮ КОМП'ЮТЕРНОГО ЗОРУ	
3.1 Опис обраних методів.....	35

3.2 Реалізація розпізнавання транспортних засобів на С#.....	41
4 ДОСЛІДЖЕННЯ ТА ПОРІВНЯННЯ З ІНШИМИ МОДЕЛЯМИ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ТРАНСПОРТУ	53
4.1 Багатопотокове динамічне злиття	57
4.2 Майбутні перспективи	59
4.2.1 Гіпотеза про лотерейний квиток	59
4.2.2 Навчання з нульовим пострілом.....	60
4.2.3 Самокеровані автомобілі	60
5 СПЕЦІАЛЬНИЙ РОЗДІЛ З ОХОРОНИ ПРАЦІ	63
ВИСНОВКИ.....	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	75
ДОДАТОК А КОД СИСТЕМИ РОЗПІЗНАВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ	79

ПЕРЕЛІК СКОРОЧЕНЬ

БКР – бакалаврська кваліфікаційна робота.

ЗНМ – згортова нейрона мережа.

ІТС – інтелектуальна транспортна система.

ПЗ – програмне забезпечення.

ALPR - Automatic License Plate Recognition.

API - Application Programming Interface.

CNN – Convolutional Neural Network.

GPS - Global Positioning System.

ITS – Intellectual Transport System .

LP detection - License Plate detection.

ReLU - rectified linear activation unit.

RNN - Recurrent Neural Networks.

Пояснювальна записка

до кваліфікаційної роботи

на тему:

РОЗПІЗНАВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ КОМП'ЮТЕРНОГО ЗОРУ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810105

Виконала студентка 4-го курсу, групи 401

_____ *А. В. Вишневецька*

«13» червня 2022 р.

Керівник: канд. техн. наук, доцент

_____ *Г. В. Кондратенко*

«13» червня 2022 р.

Миколаїв – 2022

ВСТУП

Використання транспортних засобів у нашому житті зростає в геометричній прогресії через швидкий економічний розвиток. Тож підтримка зручності транспортування і безпеки на дорогах має важливе значення за багатьма причинами: компанії перевозять свої товари по дорозі, очікується, що громадський транспорт прибуде вчасно, а власники транспортних засобів можуть стати об'єктом викрадення. Отже, існує гостра потреба у високопродуктивних інтелектуальних транспортних системах (ІТС) [1].

ІТС - це система, де інформаційні та комунікаційні технології застосовуються в області автомобільного транспорту, включаючи інфраструктуру, транспортні засоби та користувачів, а також в управлінні дорожнім рухом.

ІТС використовує ряд даних стільникового зв'язку, таких як метод мобільної тріангуляції, при якому телефони періодично передають інформацію про свою присутність в мережу мобільного зв'язку.

Потім дані перетворюються в інформацію про потік трафіку. Також використовуються Детектори Bluetooth і RFID, що встановлюються на бічних стійках.

Іншим більш актуальним джерелом інформації є смартфони. Вони пропонують багатий набір сенсорних даних від GPS до акселерометрів для відстеження швидкості і щільності руху [2]. Нарешті, радар - дуже поширений датчик для моніторингу доріг.

Однак у цій роботі увагу зосереджено на ІТС, заснованих на баченні.

Камери відеоспостереження є найбільш надійним джерелом даних, оскільки вони передають точне зображення реальності. Однак розмір записаних відеозаписів робить їх аналіз стомлюючим, а іноді і нездійсненним завданням для людей. Таким чином, автоматичний аналіз відео викликав значний інтерес у дослідників та інженерів.

Завдання спостереження, такі як виявлення інцидентів і підрахунок

транспортних засобів, вирішуються або за допомогою алгоритмів, заснованих на датчиках, або на зорі.

Крім того, візуальні дані містять безліч шаблонів, які можуть бути проаналізовані та інтерпретовані для отримання більш розширених завдань, такі як виявлення транспортних засобів і розпізнавання номерних знаків. Напроти, методи, засновані на радарних датчиках, можуть виявляти транспортні засоби тільки на мінімальній площі, в той час як методи, засновані на відео, можуть виявляти кілька транспортних засобів одночасно. Однак жоден алгоритм аналізу відео не наблизився до людського рівня виявлення транспортних засобів, класифікації або будь-який іншого природно простого завдання для людей.

Метою роботи є розпізнавання транспортних засобів за допомогою машинного навчання, а саме комп'ютерного зору.

Об'єктом розробки є процеси розпізнавання та ідентифікації транспортних засобів.

Предметом дослідження є методи та технології розпізнавання транспортних засобів, зокрема з використанням комп'ютерного зору.

Завданням у БКР є реалізація інформаційної системи для розпізнавання транспортних засобів за допомогою інструментів та бібліотек машинного навчання.

Функції, які виконує розроблена система:

- отримання зображень;
- попередня обробка;
- виділення деталей;
- детектування/сегментація;
- високорівнева обробка.

В роботі планується використовувати бібліотеку Accord, яка ідеально підходить для створення програм комп'ютерного зору. Accord.NET Framework — це платформа машинного навчання .NET у поєднанні з бібліотеками обробки аудіо та зображень, повністю написаними на C#. Це повна структура для

Кафедра інтелектуальних інформаційних систем
Розпізнавання транспортних засобів за допомогою технології комп'ютерного зору

побудови комп'ютерного зору, комп'ютерного прослуховування, обробки сигналів і статистичних додатків навіть для комерційного використання . Повний набір прикладів програм забезпечує швидкий старт для швидкого запуску та роботи, а обширна документація та вікі допомагають заповнити деталі.

Реалізація систем комп'ютерного зору залежить від області їх застосування, апаратної платформи та вимог щодо продуктивності. Деякі системи є автономними і вирішують специфічні проблеми детектування та виміру.

Докладніше про ці методи та інформаційні технології буде представлено у наступному розділі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, МЕТОДІВ ТА АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1 Основні поняття та визначення

При зйомці на відкритому повітрі за допомогою системи перегляду часто виявляється і відстежується кілька об'єктів, які рухаються у різних напрямках у різних місцях зображеної сцени. Використання стаціонарної відеокамери з грубим полем зору має сенс у таких випадках, якщо вдається надійно виявляти і відслідковувати одиночні об'єкти, що рухаються, які потім будуть зображені дуже маленькими. Труднощі, пов'язані з дрібністю зображення об'єктів для відстеження, виникають, коли об'єкти тимчасово частково або повністю приховані, або коли вони змінюють свій рух та, таким чином, зображення об'єктів, а також, при необхідності, їх тіні напроїжджної частини більше не можуть вважатися постійним у часі.

Зазвичай камери обліку трафіку встановлюють на такому боці дороги, щоб фіксувати транспортні засоби принаймні при їх наближенні та збільшенню у розмірах. Коли автомобілі збільшуються в розмірах на камері, алгоритм відстеження руху може легко зафіксувати їх.

Але система, яку було розроблено, відстежує навпаки. Автомобілі з'являються в нижній частині кадру і стають менше, коли вони віддаляються від камери. Також доводиться мати справу з випадковими мотоциклами, автобусами та зустрічним рухом на іншій смузі.

Завдання спостереження, такі як виявлення інцидентів Оскарбскі та ін. і підрахунок транспортних засобів, вирішуються або за допомогою алгоритмів, заснованих на датчиках, або на зорі [3]. Крім того, візуальні дані містять безліч шаблонів, які можуть бути проаналізовані та інтерпретовані для отримання більш докладної інформації. розширені завдання, такі як виявлення транспортних засобів і розпізнавання номерних знаків. Напроти, методи,

засновані на радарних датчиках, можуть виявляти транспортні засоби тільки на мінімальній площі, в той час як методи, засновані на відео можуть виявляти кілька транспортних засобів одночасно. Однак жоден алгоритм аналізу відео не наблизився до людського рівня виявлення транспортних засобів, класифікації або будь-який інший природно просте завдання для людей. Все це радикально змінилося з появою CNN [4]. Натхненний біологічними процесами, патерн зв'язку між нейронами нагадує організацію зорової кори головного мозку тварин. CNN може імітувати здатність мозку аналізувати візуальні дані, а в деяких випадках і перевершувати її.

Глибокі об'єкти відносяться до ієрархічної природи витягнутих об'єктів. CNN складаються з шарів, розташованих один за іншим. Перші шари витягують основні об'єкти, такі як лінії і криві, в той час як шари в кінці розпізнають форми і об'єкти, як показано на рисунку 1.1.

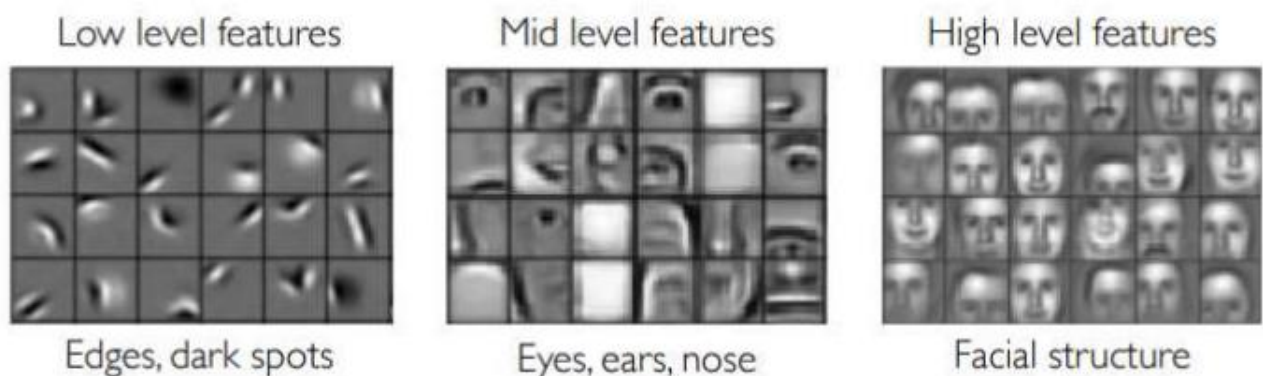


Рисунок 1.1 - Ієрархічне представлення функцій CNN людських облич від країв до структури обличчя

Глибокі шари CNN зазвичай складаються з декількох згорткових шарів, шарів об'єднання, повністю пов'язаних шарів і шарів нормалізації. Кожен з них призначений для вивчення іншого рівень представництва в ієрархії. Недоліком цієї архітектури є те, щочим більше шарів має CNN, тим більше вона вимагає обчислювальних витрат.

1.2 Останні дослідження та публікації

Для аналізу методів та способів вирішення задачі було опрацьовано низку наукових статей про машинне навчання, серед яких хотілося би виділити роботу автора Mohamed Dhia Elhak Besbes «Vehicle identity recognition using deep learning» [5]. В роботі автора було розглянуто згорткову нейронну мережу (CNN). Однак як він наголошує, ці методи мають ряд обмежень, оскільки вони виділяють неструктуровані характеристики автомобіля, які використовуються для завдання розпізнавання. У цій статті автор пропонує більш структурований метод вилучення функцій, використовуючи надійну багатопотокову архітектуру глибоких мереж. Мухамад використовує нову техніку динамічного комбінування, щоб об'єднати різні елементи автомобіля з усім зображенням. Це дозволяє поєднувати глобальне представлення з локальними функціями. Представлена система, яка була оцінена на загальнодоступних наборах даних, здатна вивчати дуже дискримінантне представлення і досягає найсучасніших результатів.

Іншою видатною публікацією є стаття Arthur A. Goshtasby «Computer Vision» [6]. Вона охоплює основні прийоми для процесів зору низького та середнього рівня. Техніки включають: геометрію camera, фільтрацію зображень і виявлення країв, аналіз кольору і текстур, зображення сегментації, узгодження зображень, аналіз руху та сприйняття глибини стерео.

Австралійські вчені також не залишилися у стороні та представили видання «A Perspective on Range Finding Techniques for Computer Vision». Автори: R. A. Jarvis, Department of Computer Science, Australian National University, Canberra, Australia [7]. У даній публікації наголошується, що останнім часом виявлено великий інтерес до отримання даних про дальність для підтримки аналізу сцени, що веде до дистанційного (безконтактного) визначення конфігурації та ступеня заповнення простору тривимірних об'єктів. У цій статті розглядаються різноманітні підходи до узагальненого визначення дальності та представлена точка зору на їх застосування та недоліки в контексті досліджень

комп'ютерного зору.

Нарешті, для повного розуміння доцільності використання методів комп'ютерного зору для розпізнавання зображень та динамічних об'єктів рекомендовано розглянути дослідження «Computer Vision: Evolution and Promise» автора T. S. Huang при університеті University of Illinois at Urbana-Champaign. У цій статті надаються причини, чому комп'ютерний зір є дуже складним дослідженням. Автор стверджує, що однією з основних причин труднощів у розробці цих методів є те, що зорова система людини є просто занадто високорозвиненою для багатьох завдань (наприклад, розпізнавання обличчя), тому системи комп'ютерного зору не можуть впоратись з поставленими задачами. Людина може розпізнавати обличчя при будь-яких варіаціях освітлення, точки зору і т. д. У більшості випадків нам не важко впізнати друга на фотографії, зробленій багато років назад. Крім того, здається, що немає обмежень у тому, скільки облич ми можемо зберігати у своєму мозку для подальшого розпізнавання. Тому ймовірність побудови автономної системи з такою зіровою продуктивністю є досить малою.

Основна складність полягає в тому, що алгоритми комп'ютерного зору майже всі тендітні; алгоритм може працювати в одних випадках, але не в інших. На думку автора, щоб застосунок комп'ютерного зору був потенційно успішним, він повинен задовольняти двом критеріям:

- можливості людської взаємодії;
- умова допущення деяких помилок є задовільною.

Оцінювані за цими двома критеріями, застосунками, які потенційно можуть бути дуже успішними, є:

- бази даних зображень / відео - індексація та пошук на основі вмісту зображень;
- людино-комп'ютерний інтерфейс на основі комп'ютерного зору - наприклад, використання жестів (в поєднанні з промовою) при взаємодії з віртуальними середовищами;
- віртуальний агент / актор - створення сцен штучної людини на основі параметрів, витягнутих з відеопослідовностей реальної людини.

1.3 Огляд існуючих аналогів

Система виявлення транспортних засобів за допомогою каскадів Хаара

Каскад хаара був навчений з використанням 526 зображень автомобілів ззаду (360 x 240 пікселів, без масштабу). Зображення були витягнуті з набору даних автомобілів, запропонованого Бредом Філіпом і Полом Апдайком, зроблених на автострадах Південної Каліфорнії [9].

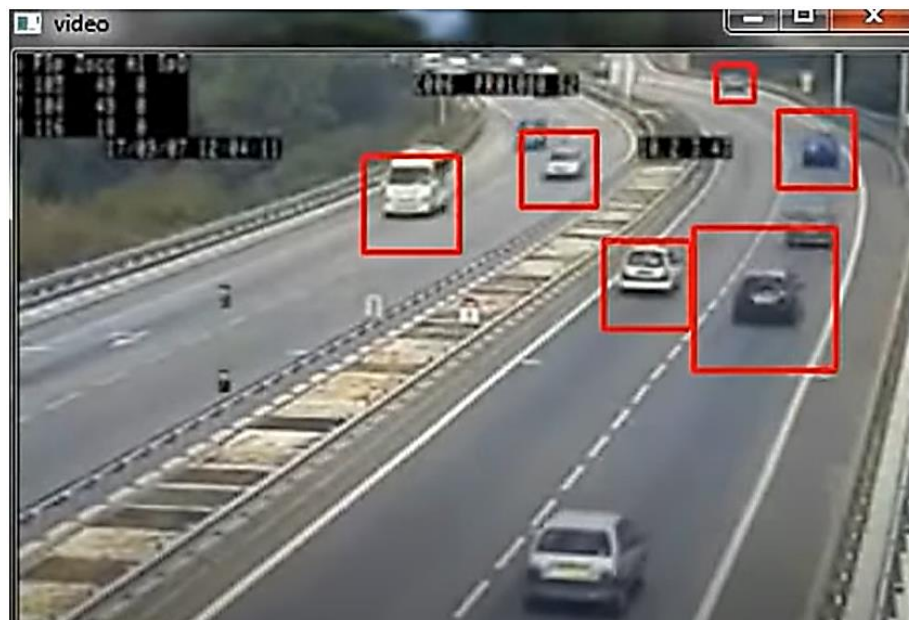


Рисунок 1.2 - Система виявлення транспортних засобів за допомогою каскадів Хаара

Система ідентифікації транспортних засобів TDS поєднує передові технології зчитування номерних знаків і визначення марки/моделі, щоб

забезпечити дуже потужний інструмент для ідентифікації транспортних засобів, що цікавлять, з мобільної платформи, наприклад, поліцейського автомобіля [10].

Система розпізнавання ідентифікації транспортного засобу складається з системи камер, встановленої на даху поліцейського автомобіля, яка взаємодіє з комп'ютерним процесором, розташованим у багажнику. Система включає в себе GPS і модем 4G LTE, що дозволяє зв'язуватися з віддаленими серверами для хмарної обробки, оновлювати чорний список/інші та повідомляти про події за потреби. Система розроблена для роботи в повністю автоматизованому режимі та буде безперервно отримувати й обробляти зображення транспортних засобів, зняті камерою на даху. Зображення оброблятимуться за допомогою програмного забезпечення автоматизованого розпізнавання номерних знаків (ALPR), наданого TDS, яке точно визначає номерний знак. Система порівнює отримані номери ліцензій, щоб визначити, коли відбувається збіг із попередньо встановленим чорним списком. Коли сірник знайдено, система додатково визначає марку та модель автомобіля (система VMM) на основі зображень, отриманих для цього автомобіля. Інформація про номерний знак, марку/модель та чорний список використовується для створення сповіщення, яке передається через WiFi для відображення як попередження на дисплеї планшета поліцейського. Відповідну інформацію про подію можна додатково пересилати на віддалений сервер і може включати GPS і мітку часу. Розташування поліцейського автомобіля можна періодично оновлювати до центральної системи розташування автомобіля, де можна створити та відобразити карту розташування всіх поліцейських автомобілів у системі для командування та контролю. Процесор, розташований у магістралі, працює під керуванням операційної системи Linux і може запускати додаткові програми, пов'язані з поліцією. офіцер поліції.

1.4 Постановка задачі

Актуальністю побудови нейронних мереж для розпізнавання зображень та відео не можна знехтувати, адже жоден алгоритм аналізу відео не наблизився до

людського рівня виявлення транспортних засобів, класифікації або будь-який іншого природно простого завдання для людей.

Метою роботи є розпізнавання транспортних засобів за допомогою машинного навчання, а саме комп'ютерного зору.

Об'єктом розробки є процеси розпізнавання та ідентифікації транспортних засобів.

Предметом дослідження є методи та технології розпізнавання транспортних засобів, зокрема з використанням комп'ютерного зору.

Завданням у БКР є реалізація інформаційної системи для розпізнавання транспортних засобів за допомогою інструментів та бібліотек машинного навчання.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- проаналізувати сучасний стан задачі розпізнавання об'єктів за допомогою комп'ютерного зору;
- окреслити існуючі технології для вирішення поставленої задачі;
- реалізація розпізнавання транспортних засобів за допомогою технологій та методів машинного навчання.
- Розробка системи та графічного інтерфейсу для представлення руху транспорту;
- порівняльний аналіз з іншими моделями для розпізнавання динамічного транспорту на дорозі.

Висновки до розділу 1

Отже, у даній бакалаврській кваліфікаційній роботі детально вивчаються кілька моделей комп'ютерного зору, щоб досягти кращих результатів і подолати колишні недоліки, такі як навчання на меншому кількості прикладів, загальні міжкраїнові норми рішення і стійкість до складних умових. Складність розпізнавання актуальна і досі, так як досконале та точне ідентифікування потребує багато часу та ресурсів для навчання.

2 ОПИС МЕТОДІВ ТА ПІДХОДІВ ДЛЯ ВИРШЕННЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ

2.1 Роль глибинного навчання у розпізнаванні графічних об'єктів

Традиційне машинне навчання складається з вилучення об'єктів, створених вручну. Навпаки, глибинні нейронні мережі вивчають ієрархічні уявлення об'єктів від ліній і кривих до об'єктів, що підвищують надійність і точність. Своїми візуальними функціями, вони привернуло багато уваги в останні кілька років. У багатьох застосунках відбувся стрибок вперед з точки зору продуктивності, таких як класифікація зображень, виявлення об'єктів, генерація зображень і сегментація зображень; і це лише деякі з них.

У цій БКР глибинні моделі в основному досліджуються для ідентифікації транспортних засобів. Виявлення та розпізнавання динамічних об'єктів є фундаментальним завданням при ідентифікації транспортних засобів, але вона стикається з багатьма проблемами, головним чином тому, що рухаючі об'єкти зливаються з навколишнім середовищем і мають невеликий простір зображення, що робить їх чутливими до змін якості зображення, освітленості, точок огляду і перекриттів.

Ці проблеми були вирішені за допомогою багатопотокової надійної архітектури для вилучення і об'єднання як локальних, так і глобальних уявлень об'єктів: по-перше, попередньо навчений детектор витягує деталі транспортного засобу, кількість виявлених деталей варіюється від одного зображення до іншому, звідси і надійність системи. По-друге, процес вибору відфільтровує зображення автомобіля з деякими виявленими деталями для оптимізації продуктивності. Потім кожна частина проходить через інший потік в спеціалізований екстрактор функцій, що дозволяє системі виявляти тонкі відмінності між класами. Нарешті, всі витягнуті об'єкти об'єднуються за допомогою нової технології злиття.

2.2 Convolutional Neural Network

Згорткові мережі відрізняються дуже високою здатністю до розпізнавання патернів на зображеннях. Тому і завдання, яке розглядається в даній роботі, може вирішуватися за допомогою даного методу.

Згорткові нейронні мережі працюють на основі фільтрів, які займаються розпізнаванням певних характеристик зображення (наприклад, прямих ліній). Фільтр - це колекція кернелів; іноді у фільтрі використовується один кернел. Кернел - це звичайна матриця чисел, званих вагами, які "навчаються" (підлаштовуються, якщо вам так зручніше) з метою пошуку на зображеннях певних характеристик. Фільтр переміщається вздовж зображення і визначає, чи є певна потрібна характеристика в конкретній його частині. Для отримання відповіді такого роду відбувається операція згортки, яка є сумою творів елементів фільтра та матриці вхідних сигналів.

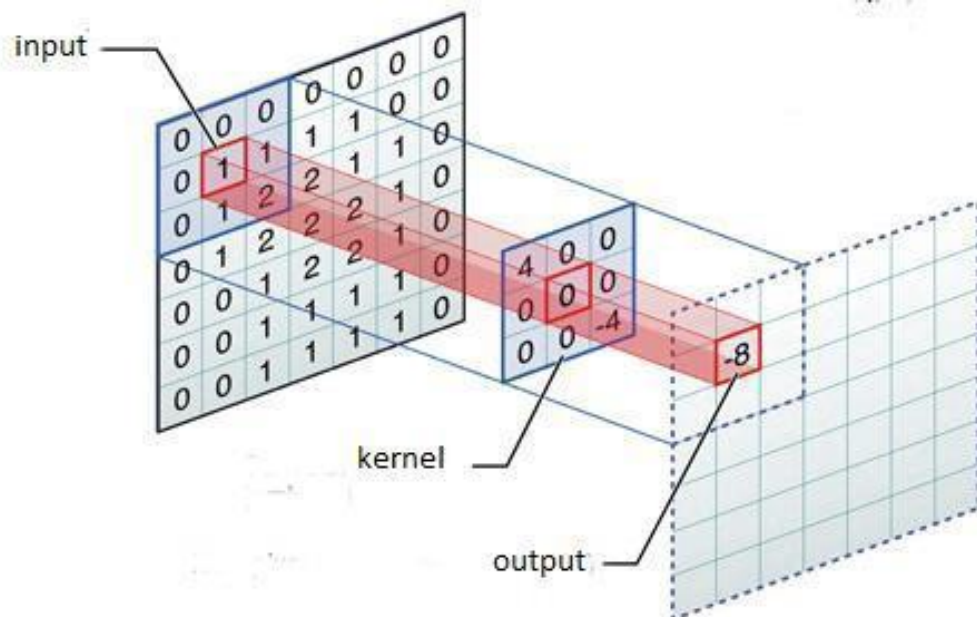


Рисунок 2.1 - Операція згортки

Якщо деяка потрібна характеристика присутня у фрагменті зображення, операція згортки на виході видаватиме число з відносно великим значенням. Якщо ж характеристика відсутня, вихідне число буде невеликим.

Фільтр може переміщатися вздовж матриці вхідних сигналів з кроком, відмінним від одиниці. Крок переміщення фільтра

називається **страйдом** (stride). Страйд визначає, на скільки пікселів повинен зміститися фільтр за один раз. На наступній формулі представлено розрахунок кількості вихідних значень після операції згортки:

$$n_{out} = \text{floor} \left(\frac{n_{in} - f}{s} \right) + 1, \quad (2.1)$$

де n_{in} - кількість вхідних пікселів;

f - кількість пікселів у фільтрі;

s - страйд.

Для того, щоб навчання ваг, укладених у кернах, було ефективним, результати згорток слід ввести деяке зміщення (bias) і нелінійність.

Зміщення — це статична величина, яку слід “змістити” вихідні значення. По суті це звичайна операція складання кожного елемента вихідної матриці з величиною усунення. Якщо пояснювати дуже поверхово, це потрібно для того, щоб вивести нейронну мережу з тупикових ситуацій, що мають суто математичні причини.

Нелінійність являє собою **функцію активації**. Завдяки ній картина, що формується за допомогою операції згортки, отримує деяке спотворення, що дозволяє нейронній мережі чіткіше оцінювати ситуацію. Цю необхідність дуже грубо можна порівняти із необхідністю людям зі слабким зором носити контактні лінзи. А взагалі така необхідність пов'язана з тим, що вхідні дані за своєю природою нелінійні, тому потрібно навмисне спотворювати проміжні результати, щоб відповідь нейронної мережі був відповідним.

Часто як функцію активації використовують ReLU (Rectified Linear Unit). Її графік зображено на рис. 2.2.

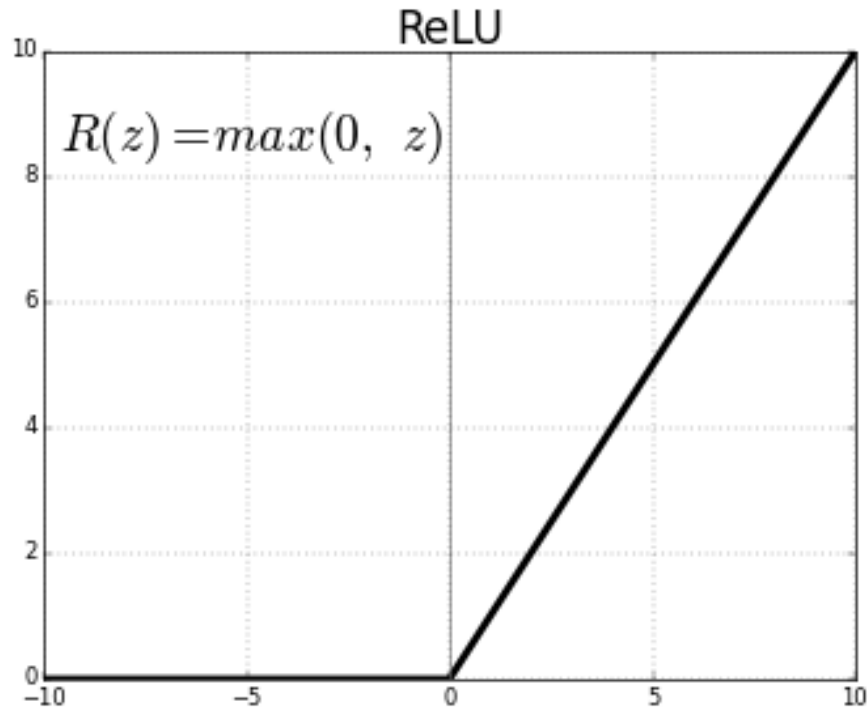


Рисунок 2.2 - Функція ReLU

Як ви можете бачити, ця функція досить проста. Вхідні значення, менші або рівні нулю, перетворюються на нуль; значення, що перевищують нуль, не змінюються.

Зазвичай у згорткових шарах використовується більше одного фільтра. Коли це має місце, результати роботи кожного з фільтрів збираються вздовж деякої осі, що дає тривимірну матрицю вихідних даних.

Кожен згортковий фільтр являє собою функцію; це означає, що згортка має властивість бути трансляційно інваріантною. Потім CNN може містити результуюче передбачення на основі цих витягнутих ознак. Практично прогнозування не залежить від того, де розташовані об'єкти, а тільки в тому випадку, якщо вони присутні, щоб об'єкт міг змінити положення на зображенні і при цьому бути розпізнаним або виявленим. Згортаючі фільтри управляються безліччю параметрів, таких як крок, заповнення нулем і глибина каналу. Крок - це кількість пікселів, на які ковзає фільтр. При більш значному кроці виходять карти об'єктів меншого розміру, і навпаки. Нульові відступи дозволяють нам певною мірою контролювати розмір карт об'єктів. Він заповнює вхідну матрицю нулями навколо

кордону. Глибина каналу-це кількість фільтрів, що використовуються при згортанні. Чим більше фільтрів, тим більше об'єктів зображення витягується.

Фільтри можуть бути чутливими до шуму, тому використовується метод згладжування, який називається об'єднанням або субдискретизацією. Це може бути досягнуто шляхом отримання середніх або максимальних значень за зразком зображення або карти об'єктів. На рис. 2.3 показаний приклад двох методів об'єднання.

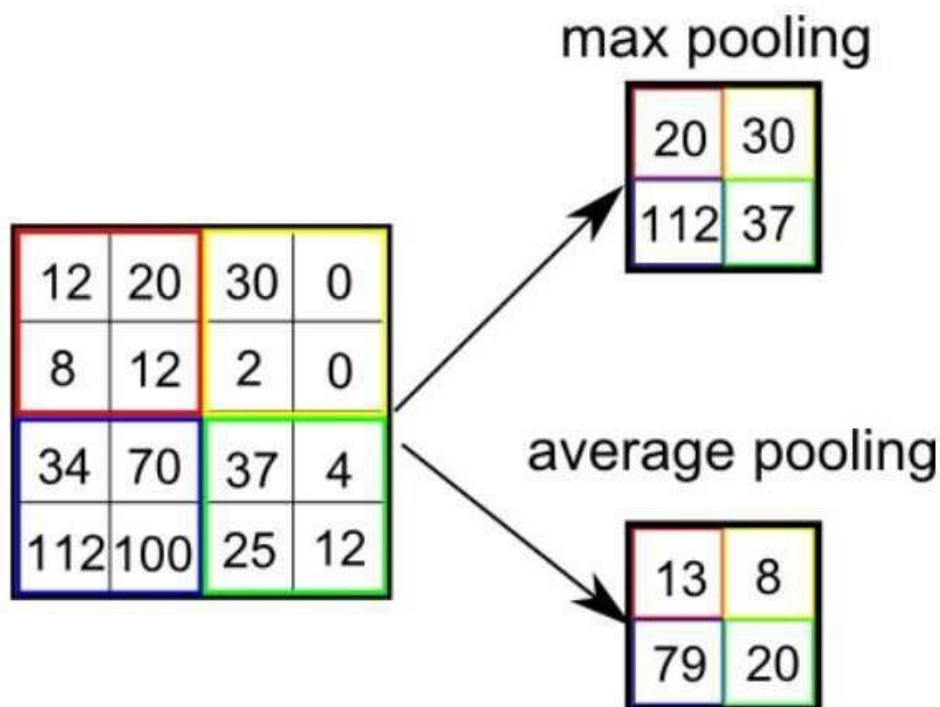


Рисунок 2.3 - Методи середнього об'єднання і максимального об'єднання

2.3 Recurrent Neural Network

Рекурентні нейронні мережі були створені для роботи з динамічними послідовними даними, такими як текст, аудіо і відео, тобто будь-якими тимчасовими послідовностями. Для цього використовуються зв'язки між прихованими шарами.

На відміну від звичайних нейромереж з прямим зв'язком рекурентні мережі можуть працювати з контекстом, а також зберігати, запам'ятовувати та використовувати інформацію про контекст у послідовності будь-якої довжини. Головна відмінність від звичайних нейромереж, яка дозволяє досягти цього - це зв'язок

між станом прихованого шару в один момент та станом цього ж шару в наступний момент часу.

2.3.1 Архітектура рекурентної нейронної мережі

На рисунку 2.4 показано, як рекурентна мережа розгортається на повну мережу. По суті, за допомогою цього можна просто показати скільки шарів (тобто скільки разів ми повторюємо обчислення) в нашій мережі.

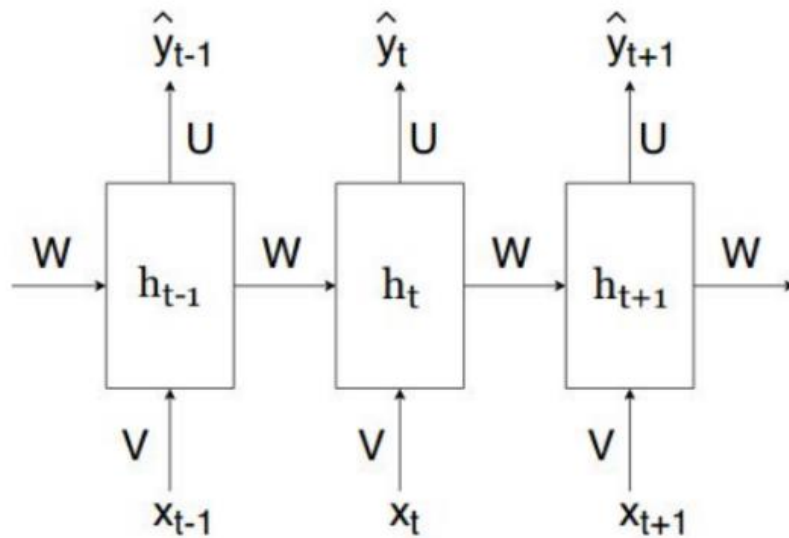


Рисунок 2.4 – Розгортання рекурентної мережі

Формули

$$h_t = f(Vx_t + Wh_{t-1} + b) \quad (2.2)$$

$$y_t = g(Uh_t + b) \quad (2.3)$$

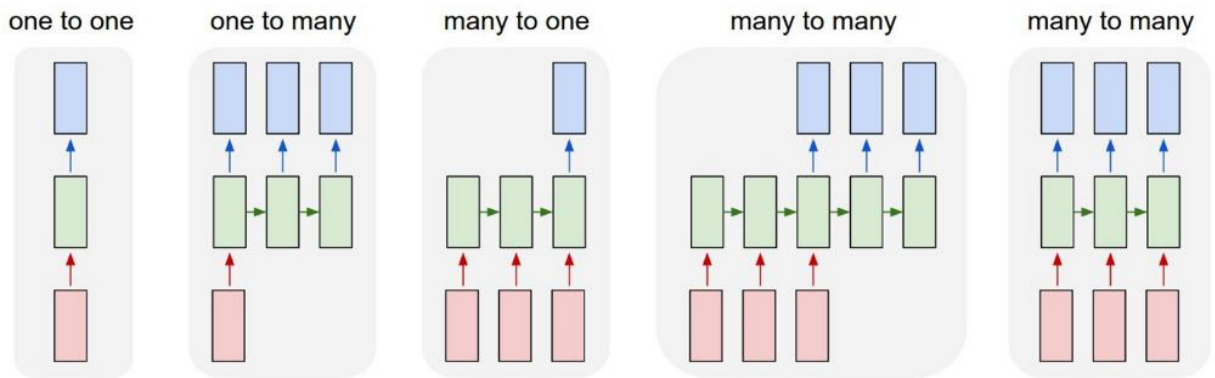
де,

- x_t – вхід на кроці t . Наприклад, для роботи з текстом це послідовність ембедінгів речень.
- h_t - прихований стан на кроці t . Цей стан передається в наступну комірку і є пам'яттю мережі. Прихований стан залежить від усіх попередніх станів та змінюється після кожного кроку.
- y_t – вихід на кроці t . Його може і не бути, наприклад, у задачі класифікації текстів вихід буде лише один на самому кінці нейромережі.

- U, V, W - навчальні параметри мережі. На відміну від традиційних нейромереж, у яких на кожному шарі були свої параметри, дані параметри є спільними для всіх шарів рекурентної мережі.
- $g(x), f(x)$ - нелінійні функції активації (найчастіше ReLU або tanh).

2.3.2 Типи рекурентних нейронних мереж

Рекурентні нейронні мережі розрізняються за кількістю входів та виходів. Наприклад, до типу "один до багатьох" можна віднести генерацію аудіо-запису, якщо на вхід подавати жанр музики, а за принципом "багато хто до багатьох" працює машинний переклад.



Andrej Karpathy, 2015, The Unreasonable Effectiveness of Recurrent Neural Networks

Рисунок 2.5 – Типи рекурентних мереж

RNN були успішно застосовані для вирішення безлічі завдань: створення підписів до зображень, мовного моделювання і розпізнавання дій. Далі розглянемо важливі концепції та методи RNN [11].

Vanilla RNN

Для даних типу послідовності для правильного прогнозування необхідні як поточні входні дані, так і попередні стани. Наприклад, для поточного стану t для обчислення h_t використовуються як входні дані X_t , так і попередній стан h_{t-1} :

$$h_t = f(X_t, h_{t-1}), \quad (2.4)$$

де h_t - прихований стан на кроці t ;

x_t – вхід на кроці t .

Однак ця модель не підходить для довгострокових залежностей.

Long Short-Term Memory

Мережі довготривалої короткочасної пам'яті (LSTM) націлені на обробку навчання для довгострокових залежностей. LSTM розбиває h_t на 2 змінні h_t і C . Вони мають три входи для контролю того, яка інформація буде проходити через них:

$$gate_{forget} = \delta(W_{fx}X_t + W_{fh}X_{t-1} + b_f) \quad (2.5)$$

$$gate_{input} = \delta(W_{ix}X_t + W_{ih}h_{t-1} + b_i) \quad (2.6)$$

$$gate_{output} = \delta(W_{ox}X_t + W_{oh}h_{t-1} + b_o), \quad (2.7)$$

де $gate_{input}$ керує тим, яка частина нової інформації буде додана до стану C ;

$gate_{output}$ керує частиною стану комірки, яка відобразиться як прихований стан;

$gate_{forget}$, як випливає з назви, керує тим, яка частина попереднього стану комірки буде видалена.

Прихований стан і стан комірки будуть оновлені наступним чином:

$$\tilde{C} = \tanh(W_{cx}X_t + W_{ch}h_{t-1} + b_c) \quad (2.8)$$

$$C_t = gate_{forget} \times C_{t-1} + gate_{input} \times \tilde{C}$$

$$h_t = gate_{output} \times \tanh(C_t) \quad (2.9)$$

C_t формується шляхом забування частини попереднього стану комірки при додаванні нового речення C .

На рисунку 2.6 показана Архітектура LSTM [12].

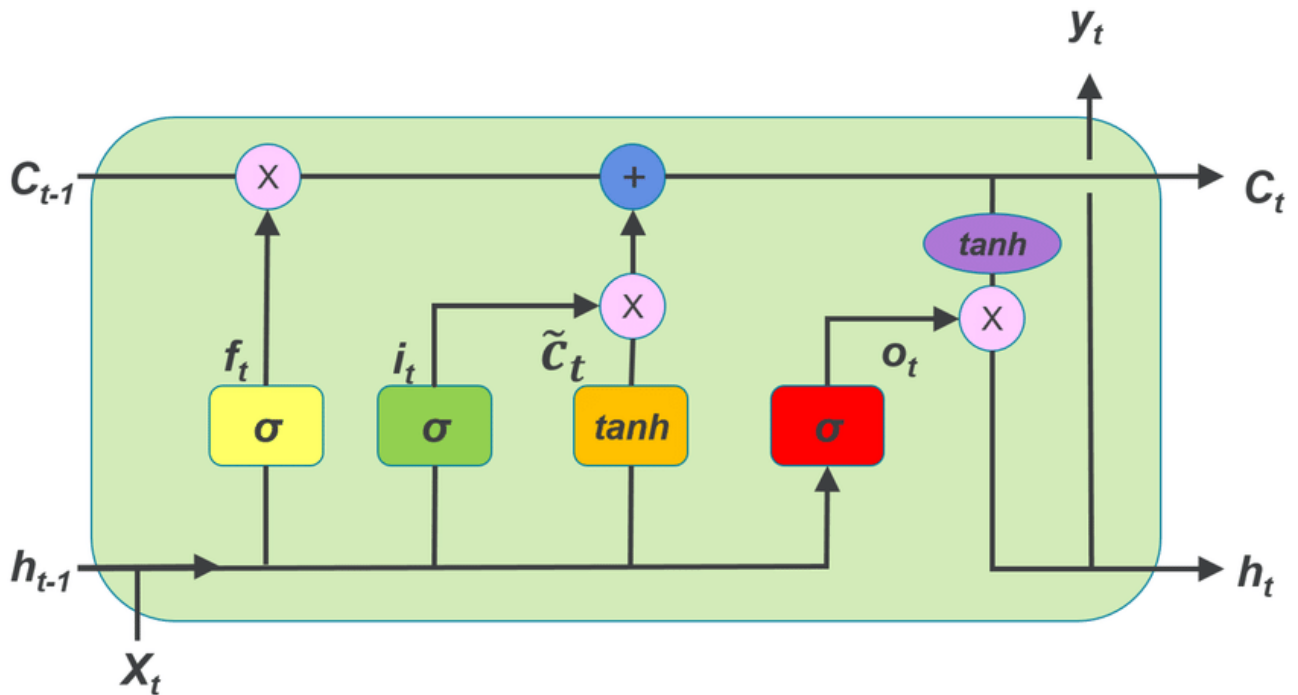


Рисунок 2.6 - Архітектура Long Short-Term Memory

2.3.3 Алгоритм зворотного розповсюдження помилки у часі

Алгоритм зворотного поширення помилки у часі відрізняється від звичайного тим, що для його застосування необхідно "розгорнути" рекурентну нейронну мережу, порахувати на кожному кроці помилки, оновити ваги і потім знову згорнути мережу. Ця операція може бути досить витратною для глибоких мереж, тому іноді доцільно використати усічений алгоритм зворотного розповсюдження помилки.

2.3.4 Проблеми рекурентної нейронної мережі

Попередні розглянуті моделі розширили базову структуру простого RNN, щоб поліпшити його продуктивність при вивченні контекстуальних залежностей одновимірних послідовностей. Однак існує кілька проблем, які вимагають розуміння контекстуальних залежностей за кількома вимірами. Найбільш популярні мережеві архітектури використовують CNN для вирішення цих проблем.

Включення рекурентних з'єднань в кожен згортковий шар може сформувати рекурентну згорткову нейронну мережу. (RCNN) [13]. Активація одиниць в RCNN розвивається з часом, оскільки вони залежать від сусіднього підрозділу. Цей підхід може інтегрувати контекстну інформацію, важливу для

завдань розпізнавання об'єктів. Такий підхід збільшує глибину моделі, в той час як кількість параметрів залишається постійним за рахунок розподілу ваги між шарами. Використання повторюваних підключень з виходу на вхід прихованого рівня дозволяє мережі моделювати залежності міток і згладжувати свої власні вихідні дані на основі своїх попередніх вихідних даних. До проблем рекурентної нейронної мережі відносять вибух градієнта та загасання градієнта. Найчастіше зустрічається друга проблема. Насправді це означає, що нейромережа зможе запам'ятовувати дуже довгі послідовності і навчання нічого очікувати відбуватися ефективно.

Для вирішення проблеми вибуху градієнта був винайдений **Gradient clipping** (обмеження норми градієнта), а для боротьби з загасанням градієнта були придумані моделі LSTM і GRU.

2.4 LP Детекція та Розпізнавання

Виявлення об'єктів

Виявлення об'єкта - це ідентифікація об'єкта на зображенні, а також його локалізація і класифікація. Детектор об'єктів Віюлі-Джонса-перший детектор об'єктів; він вийшов у 2001 році. Він використовувався для розпізнавання обличчя і широко використовувався в той час. Однак з недавнім успіхом підходів до глибокого навчання були запропоновані нові методи з набагато більшою надійністю і точністю. Перша модель детектора об'єктів глибокого навчання називалася Over feat Network, в якій використовувався підхід CNN з ковзаючим вікном.

З 2012 року моделі детекторів об'єктів зазнали безліч змін; першим проривом у виявленні об'єктів став RCNN: він складається з трьох основних частин: екстрактора регіонів, екстрактора ознак і, нарешті, класифікатора. Алгоритм пропозиції регіону витягує область інтересу (ROI), потім кожен регіон передається в класифікатор, і, нарешті, витягнуті об'єкти класифікуються.

Незабаром після цього з'явився швидкий RCNN зі значним поліпшенням. Модель RCNN бере кожен пропозицію регіону і проганяє їх через згорткову базу.

Це досить неефективно. Швидкий RCNN спрямований на скорочення цих накладних витрат за рахунок одноразового запуску згорткової бази. Швидший rcnn вийшов після статті Fast rcnn. У ньому був запропонований детектор, який навчається наскрізним способом.

Виявлення транспортних засобів

Для виявлення транспортних засобів були використані різні ознаки зовнішнього вигляду. Такі функції, як край, тінь і симетрія, використовуються для виявлення транспортних засобів. Проте в останні роки для виявлення транспортних засобів використовуються загальні та надійні Набори рис, а не прості елементи зображення, такі як краї та симетрія. Ці набори функцій дозволяють здійснювати пряму класифікацію і виявлення об'єктів на зображеннях, що в даний час широко поширене в літературі по комп'ютерному зору. Особливості Екстрактори, такі як HOG і Haar-like, надзвичайно добре представлені в літературі з виявлення транспортних засобів, а також в літературі з виявлення об'єктів. Щоб витягти об'єкти HOG, краї оцінюються по зображенню, а потім дискретизуються і відкидаються орієнтації інтенсивностей країв в гістограму. Особливості HOG-це виразні особливості зображення, що демонструють гарне уявлення виявлення в ряді завдань комп'ютерного зору. Для виявлення транспортного засобу використовується симетрія об'єктів HOG, виділених в даній ділянці зображення, поряд з самими об'єктами HOG [14]. У деяких випадках положення транспортного засобу також визначається особливостями свині. Основним недоліком функцій HOG є те, що вони досить повільно обчислюються.

У літературі ми розрізняємо два види підходів до розпізнавання LP: підходи, засновані на сегментації, і підходи, не засновані на сегментації.

Підходи, засновані на сегментації, спочатку витягують кожен символ в LP. Потім виконується алгоритм розпізнавання тексту для розпізнавання кожного з них. Існуючі роботи з сегментації LP можна розділити на дві основні категорії: на основі проєкцій і на основі підключених компонентів. Підходи, засновані на проєкції, використовують той факт, що символи і фон мають явно різні кольори

в LP, що дає протилежні значення в двійковому зображенні. Гістограми вертикальної і горизонтальної проекцій пікселів потім можуть бути використані для сегментації символів. На такі підходи може легко вплинути обертання LP. Підходи, засновані на пов'язаних компонентах, виконують сегментацію, позначаючи всі пов'язані пікселі в двійковому зображенні на компоненти. Цей тип методу стійкий до обертання, але він не може правильно сегментувати символи, коли вони з'єднані разом або розбиті. Оскільки символи сегментовані, етап розпізнавання може бути виконаний як завдання класифікації з одним класом для кожного буквено-цифрового символу. Існуючі алгоритми можна розділити на дві категорії: зіставлення шаблонів і методи, засновані на навчанні.

Методи, засновані на зіставленні шаблонів, полягають у порівнянні подібності даного символу і шаблонів. Потім вибирається найбільш схожий шаблон. Було запропоновано кілька заходів подібності, включаючи відстань Махаланобіса і відстань Хеммінга. Ці методи зазвичай застосовуються до двійкових зображень і обмежені, оскільки вони працюють для розміру та шрифту одного символу, і вони не підтримують поворот або переривчасті символи.

Методи, засновані на навчанні, більш надійні і можуть працювати з символами різного шрифту, освітлення або повороту. Вони використовують методи машинного навчання для розпізнавання символів з використанням одного або декількох ознак, таких як щільність кордонів, градієнт, масштабно-інваріантне перетворення (SIFT) і т.д.

Для методів без сегментації розпізнавання виконується на глобальному зображенні LP без сегментації символів. Зазвичай ковзаюче вікно над вхідним зображенням генерує безліч попередніх символів невеликими кроками. Кожен попередній символ потім використовується розпізнавачем. Коли ковзаюче вікно повністю переміщує вхідне зображення, прогнозовані результати аналізуються, і визначається остаточна послідовність. Послідовні однакові символи вважаються одним символом, а пробіл використовується для розділення інших

символів. У контексті розпізнавання LP в декількох роботах були запропоновані підходи без сегментації, засновані на методах DL.

В цілому, ми відзначаємо, що використання методів глибокого навчання для виявлення LP і визнання як і раніше обмежена, і обмежена конкретними умовами (конкретні країни, одномовні платівки, спеціальні формати LP ...). Лише в декількох роботах ALPR виконується комплексно, і більшість пропонує методів засновані на функціях, створених вручну [15].

2.5 Набори даних і протоколи оцінки

Набір даних GAP-LP

Зображення з набору даних GAP-LP були отримані за допомогою камер різної якості при різних дозволах, кутах огляду і при денному світлі, щоб краще перевірити надійність алгоритмів виявлення / розпізнавання LP. Набір даних GAP-LP 1 знаходиться у вільному доступі для дослідницької спільноти. Він складається з 9175 повністю анотованих зображень як для виявлення, так і для розпізнавання LP. Набір даних розділений наступним чином: 7117 зображень для навчання, 456 для перевірки і 1602 зображення для тесту. На рис. 2.7 показані деякі зразки зображень з набору даних GAP-LP [16].



Рисунок 2.7 - Зображення зразків GAP-LP і відповідні їм номери

Набір даних RADAR

Набір даних RADAR складається з 6448 кольорових зображень у форматі JPEG, отриманих з радарів на різних дорогах Тунісу. Зображення були взяті з природних сцен зі складним фоном і при різних умовах освітлення (день, ніч,

сонце, дощ,...), різних ракурсах і положеннях, а також декількох формах LP (квадрат і прямокутник) і нормах. Набір даних розділений на три набори: 3998 зображень для навчання, 2000 для тестування і 450 для перевірки. На рис. 3.3 показані деякі зразки зображень з набору даних радара [17].



Рисунок 2.8 - Зображення зразків RADAR

2.6 Порівняльний приклад систем ALPR – Автоматичних систем розпізнавання номерних знаків

Для прикладу проведемо експерименти для перевірки ефективності запропонованої системи ALPR. Всі експерименти проводилися на графічному процесорі NVIDIA GTX 1080 Ti (3584 ядра CUDA і 11 ГБ оперативної пам'яті). Експерименти були проведені в двох наборах даних, зібраних з реальних середовищ спостереження за дорогами та контролю доступу до парковок.

З таблиці 2.1 видно, що метод LP детекції досяг вражаючих показників виявлення з точністю 100% і відкликанням 100% в наборі даних GAP-LP і 99,09 в наборі даних Radar. Тут виявлення вважається правильним, якщо перекриття між виявленням і обмежувальною рамкою істинності більше 0,5.

Таблиця 2.1 – LP детекція: точність та швидкість відкликання для порогу 0.5

Dataset	GAP-LP	Radar
Recall	100%	99.09%
Precision	100%	100%

У таблиці 2.2 наведені значення відкликання при виявленні LP з урахуванням різних допустимих значень (від 0,3 до 0,8). Як можна помітити, метод не дуже чутливий до різних варіацій, за винятком дуже високих значень.

Таблиця 2.2 – LP детекція: точність та швидкість відкликання для порогу від 0.3 до 0.8

DatabaseIOU	30	40	50	60	70	80
Radar (Recall %)	100	99.79	99.09	97.93	85.49	65.88
GAP-LP (Recall%)	100	100	100	99.06	93.70	67.83

Результати підтверджують надійність запропонованого підходу до виявлення.

2.7 Огляд функціоналу бібліотек C# для вирішення задачі розпізнавання

Бібліотека AForge

.NET AForge.NET — це бібліотека з відкритим вихідним кодом, розроблена мовою C#, яка використовується розробниками та дослідниками при вирішенні завдань, пов'язаних із комп'ютерним зором. А також у бібліотеці AForge.NET є можливості для вирішення задач у галузі штучного інтелекту. Діапазон засобів, що застосовуються бібліотекою, досить різноманітний: обробка зображень, нейронні мережі, генетичні алгоритми, нечітка логіка, машинне навчання, робототехніка та багато іншого.

Основні компоненти бібліотеки AForge.NET представлені нижче:

- AForge.Imaging — бібліотека, призначена для роботи із зображеннями та фільтрами;
- AForge.Vision - бібліотека, що застосовує методи комп'ютерного зору;
- AForge.Video — пакет бібліотек для виконання робіт, пов'язаних із відеоданими; – AForge.Neuro - бібліотека, в якій використовуються можливості нейронних мереж;
- AForge.Genetic — бібліотека, призначена для вирішення різноманітних завдань із застосуванням генетичних алгоритмів;
- AForge.Fuzzy - бібліотека працює з нечіткою логікою;
- AForge.Robotics — бібліотека підтримує методи, що використовуються в галузі робототехніки;

– AForge.MachineLearning — бібліотека, де застосовуються елементи машинного навчання. Бібліотека AForge.NET знаходиться у процесі постійного покращення та розвитку. За цією бібліотекою існує багато прикладів, що показують її роботу, а також є html-документація, яка може знадобитися розробникам-початківцям, які захочуть використовувати цю бібліотеку у своїх проектах.

Accord.NET Framework

Accord.NET Framework — порівняно нова програмна платформа для AForge.NET, відомого пакету бібліотек у галузі комп'ютерного зору та штучного інтелекту, повністю створена мовою C#.

Бібліотека Accord.NET використовує засоби та можливості, яких раніше не було у AForge.NET: застосування векторних машин на рівні ядра; дискримінантний та проєктивний аналіз; підтримує приховані марківські моделі; застосування нових методів навчання нейронних мереж; нові фільтри зображень та інші можливості. Бібліотека Accord.NET розроблена мовою об'єктно-орієнтованого програмування C#. Призначена для створення програм, що застосовують можливості комп'ютерного зору, комп'ютерної обробки звуку, обробки сигналів та різних статистичних методів. Accord.NET Framework може використовуватись у додатках сімейства Microsoft Windows, Xamarin, Unity3D, Windows Store, Linux, а також у мобільних пристроях [19]. Після злиття з проектом AForge.NET, в даний час платформа пропонує уніфікований API, для навчання машинним моделям, який є простим у використанні та розширюваним.

Проект Accord.NET Framework спочатку створювався для розширення можливостей AForge.NET Framework, але з того часу включив AForge.NET у себе. Нові версії об'єднали обидва фреймворки під ім'ям Accord.NET. Accord.NET містить набір бібліотек, доступних у вихідному коді, а також за допомогою інсталяторів і пакетів NuGet, що виконуються. Вихідний код проекту доступний за умов Gnu Lesser Public License, версія 2.1. у репозиторії на GitHub. Можливості Accord.NET Framework

Можливості бібліотеки Accord.NET Framework можна розділити на кілька основних груп:

- класифікація (а саме метод опорних векторів, логістична регресія, нейронні мережі);
- регресія (особливо, лінійна регресія з безліччю параметрів; регресія, що використовує метод опорних векторів; логістична регресія); – кластеризація (зокрема, методом k-середніх);
- розподілу (більше 40 видів);
- перевірка гіпотез;
- ядерні методи;
- обробка зображень;
- обробка звуку та сигналів;
- алгоритми комп'ютерного зору.

Повний набір прикладів програм дозволяє легко і швидко почати розробляти програми, а велика документація та вікі допомагає заповнити всі деталі. Accord.Vision – виявлення та відстеження обличчя в режимі реального часу, а також загальні методи для виявлення, відстеження та перетворення об'єктів у потоки зображень. Містить визначення каскаду, трекари Camshift та Dynamic Template Matching. Включає заздалегідь створені класифікатори для людських осіб та деякі риси обличчя, наприклад, такі як носи.

Accord.Imaging – містить детектори точок інтересу (наприклад, Harris, SURF, FAST і FREAK), фільтри зображень, методи зіставлення зображень та стикування зображень, а також екстрактори функцій, такі як гістограми орієнтованих градієнтів та описові функції Haralick.

Accord.Controls.Vision — компоненти Windows Forms та елементи керування для відстеження рухів голови, обличчя та рук та інших завдань, пов'язаних із комп'ютерним зором.

Accord.Controls.Imaging – елементи керування Windows Forms для відображення та обробки зображень. Містить зручний елемент керування

ImageBox, який імітує традиційний MessageBox для швидкого відображення або перевірки зображення.

Accord.MachineLearning — застосування методу опорних векторів, дерев прийняття рішень, наївних байєсівських моделей, моделей Gaussian Mixture та загальних алгоритмів, таких як Ransac, перехресна перевірка та Grid-Search для додатків, де використовується машинне навчання.

Accord.Math - містить бібліотеку розширень матриці, поряд з набором чисельних матричних методів декомпозиції, чисельних алгоритмів оптимізації для обмежених та необмежених завдань, спеціальних функцій та інших інструментів для наукових додатків.

Accord.Statistics - містить розподілу ймовірностей, тестування гіпотез, статистичні моделі та такі методи, як лінійна та логістична регресія, приховані марківські моделі, метод випадкових полів, метод головних компонентів, регресія методом найдрібніших дробових квадратів, лінійний дискримінантний аналіз та багато інших.

Accord.Vision.Detection містить детектори об'єктів, такі як метод Віоли-Джонса (метод Хаара). Каскади Хаара повністю сумісні з визначеннями, що генеруються OpenCV, а складання поставляється з прямою підтримкою пов'язаних шаблонів обличчя та носа.

Accord.Vision.Detection.Cascades - вбудовані каскади Хаара для використання з детектором об'єктів Хаара. Ці каскади можна викликати безпосередньо з коду без завантаження файлів XML.

Застосування AForge. Net та Accord. NET Framework у завданнях розпізнавання осіб

Для захоплення відео з пристрою відео введення використовують клас AForge.Video.DirectShow.VideoCaptureDevice. Цьому класу потрібно вказати маркер пристрою, з якого буде здійснюватися захоплення. Також потрібно призначити обробник події NewFrame. Ця подія з'являється в той час, коли з

камери надіслано новий кадр. Потім цей кадр відправляється у вигляді об'єкта Bitmap в обробник події, де вже відбувається його.

Висновки до розділу 2

У даному розділі було розглянуто стек методів та підходів, які можна ефективно використовувати при вирішенні задачі розпізнання. Виявлення та розпізнавання динамічних об'єктів є фундаментальним завданням при ідентифікації транспортних засобів, але вона стикається з багатьма проблемами, головним чином тому, що рухаючі об'єкти зливаються з навколишнім середовищем і мають невеликий простір зображення, що робить їх чутливими до змін якості зображення, освітленості, точок огляду і перекриттів. Тому досі розробляється багато нових технологій для точного ідентифікування зображень. Серед описаних бібліотек, які включають різноманітні класифікатори та підходи, було звернено увагу на Accord.NET Framework, а також її компоненти, призначені для вирішення завдань комп'ютерного зору.

3 РЕАЛІЗАЦІЯ РОЗПІЗНАВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНОГО ЗОРУ

3.1 Опис обраних методів

Виявлення транспортного засобу є ключовим завданням системи моніторингу транспорту, оскільки після виявлення транспортного засобу інші програми можуть бути використані більш ефективним чином. Для автоматичного виявлення та класифікації транспортних засобів використовуються різні технології. Різні алгоритми (засновані на формі, розмірі, текстурі, кольорі і т.д.) використовуються для класифікації різних транспортних засобів з відео, використовуючи обробку зображень, потім ця інформація передається в централізовану систему управління і моніторингу.

На рис. 3.1 показаний процес, який виконується в системі моніторингу дорожнього руху, заснованої на відео, яка має камеру, встановлену на узбіччі дороги, що спостерігає за дорожнім рухом. Камера працює як сенсорний пристрій, що використовується для відеозйомки дорожнього руху. Захоплені відеозображення перетворюються в цифрову форму за допомогою процесора. Зображення в їх цифровій формі обробляються і досліджуються з використанням методів обробки зображень. Витягнута інформація потім може бути використана будь-яким зовнішнім користувачем, таким як центр управління дорожнім рухом, для моніторингу та управління дорожнім рухом і т. д.

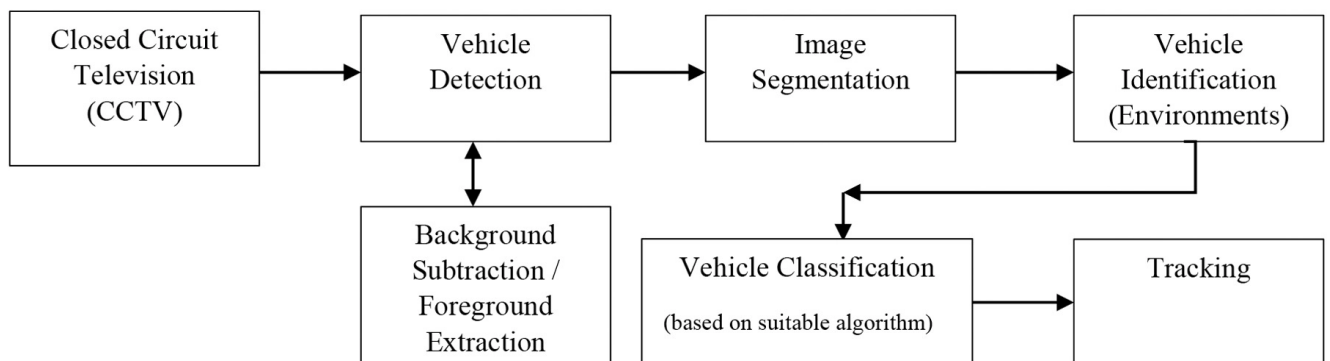


Рисунок 3.1 - Структурна схема виявлення та класифікації транспортних засобів

Віднімання фону, також називається виявленням переднього плану, є однією з життєво важливих завдань, за допомогою якої можуть бути реалізовані багато інших додатків, такі як відстеження транспортних засобів, розпізнавання і виявлення нерівностей. Він використовується для сегментації руху в статичній сцені. Для виявлення руху попиксельне зображення віднімається з фонового зображення, яке розглядається як еталонне зображення. Певний пороговий рівень визначається, якщо різниця перевищує пороговий рівень, то зображення вважається переднім планом. Формування фонового зображення розпізнається як фонове моделювання. Деяка морфологічна постобробка виконується після створення карт пікселів переднього плану, таких як ерозія, розширення і закриття, щоб зменшити шум і поліпшити виявлену область. Еталонне фонове зображення час від часу оновлюється, і зображення, на якому показаний ділянку дороги без транспортного засобу, служить. Існують різні методи, за допомогою яких віднімання фону може бути застосовано з точки зору виявлення переднього плану.

Хейккіла і Сільвен дали дуже просту концепцію фонові техніки, в якій піксель в розташуванні (x, y) на поточному зображенні позначається як передній план, якщо виконується

$$|I_t(x, y) - B_t(x, y)| > Th, \quad (3.1)$$

де попередньо заданий визначений поріг позначається як Th .

Фільтр імпульсної характеристики оновлює фонове зображення, як показано в наведеному нижче рівнянні:

$$B_{t+1} = \alpha I_t + (1 - \alpha)B_t, \quad (3.2)$$

де α -константа, яка знаходиться в діапазоні від 0 до 1.

Побудова піксельної карти переднього плану супроводжується морфологічним закриттям і усуненням областей малого розміру. За допомогою цієї простої фонові моделі пікселі, що відповідають рухомим транспортним засобам на передньому плані, можуть бути виявлені за допомогою порогового значення будь-якої з функцій відстані.

Найзручніший метод розробки системи для ідентифікації об'єктів є використання бібліотеки та фреймворку, у якому зібрано низку необхідних інструментів для синтезу обраних класифікаторів та детекторів, які виконують функції розпізнавання. Саме тому у даній БКР буде зосереджено увагу на передовій бібліотеці Accord.NET Framework, яка використовує засоби та можливості, яких раніше не було у її попередників.

Accord.NET Framework — порівняно нова програмна платформа для AForge.NET, відомого пакету бібліотек у галузі комп'ютерного зору та штучного інтелекту, повністю створена мовою C#.

Бібліотека Accord.NET використовує засоби та можливості, яких раніше не було у AForge.NET: застосування векторних машин на рівні ядра; дискримінантний та проєктивний аналіз; підтримує приховані марківські моделі; застосування нових методів навчання нейронних мереж; нові фільтри зображень та інші можливості. Бібліотека Accord.NET розроблена мовою об'єктно-орієнтованого програмування C#. Призначена для створення програм, що застосовують можливості комп'ютерного зору, комп'ютерної обробки звуку, обробки сигналів та різних статистичних методів. Accord.NET Framework може використовуватись у застосунках сімейства Microsoft Windows, Xamarin, Unity3D, Windows Store, Linux, а також у мобільних пристроях. Після злиття з проектом AForge.NET, в даний час платформа пропонує уніфікований API, для навчання машинним моделям, який є простим у використанні та розширюваним.

Можливості бібліотеки Accord.NET Framework можна розділити на кілька основних груп:

- класифікація (а саме метод опорних векторів, логістична регресія, нейронні мережі);
- регресія (особливо, лінійна регресія з безліччю параметрів; регресія, що використовує метод опорних векторів; логістична регресія);
- кластеризація (зокрема, методом k-середніх);
- розподілу (більше 40 видів);

- перевірка гіпотез;
- ядерні методи;
- обробка зображень;
- обробка звуку та сигналів;
- алгоритми комп'ютерного зору.

З перелічених методів знадобляться каскадний класифікатор та метод опорних векторів, обробка зображень та алгоритми комп'ютерного зору. Коротко поясню з приводу **метода опорних векторів**.

Лінійний класифікатор - це один із способів вирішення поставленого завдання.

Ідея полягає в наступному: знайти пряму, яка відокремлює всі помаранчеві точки від зелених точок. Якщо вдасться знайти таку пряму, то класифікувати кожну нову точку можна буде наступним чином: якщо точка лежить вище прямої, то вона помаранчева, якщо нижче — зелена. Формалізуємо цю класифікацію: необхідно знайти вектор w такий, що для деякого граничного значення b і нової точки x_i виконується:

$$w \times x_i > b \Rightarrow y_i = 1; \quad (3.2)$$

$$w \times x_i < b \Rightarrow y_i = -1, \quad (3.3)$$

де рівняння $w \times x_i = b$ описує гіперплощину, що розділяє класи в просторі R^n .

Якщо скалярний добуток вектора w на x_i більший за допустиме значення b , то нова точка належить першій категорії, якщо менша — другій. Насправді вектор w перпендикулярний шуканій розділяємій прямій, а значення b залежить від найкоротшої відстані між розділяємою прямою і початком координат.

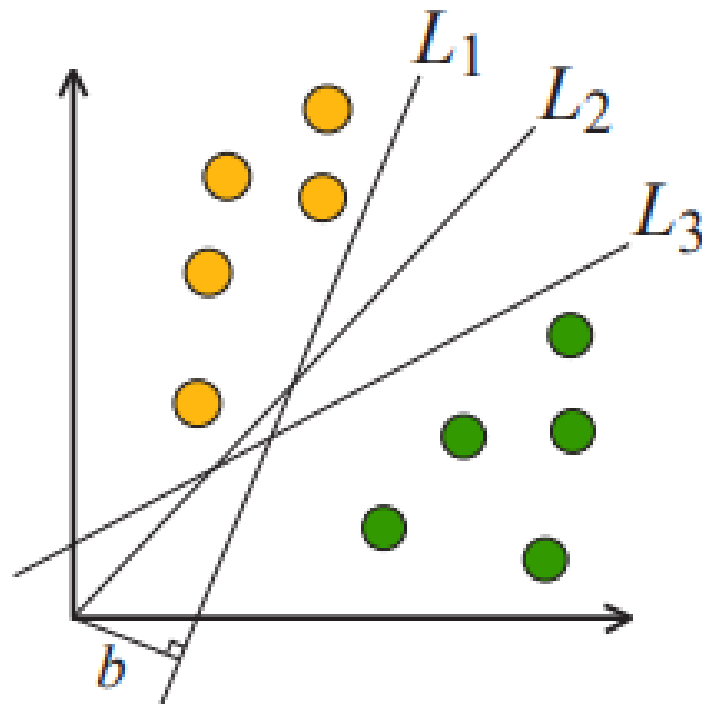


Рисунок 3.2 - Приклад класифікуючих прямих

Розглянемо приклад на рисунку 3.3. Для прямої L_2 межа b дорівнює 0, а для прямої L_1 — довжині перпендикуляра, опущеного на L_1 з початку координат. Метод опорних векторів зводить навчання класифікатора до оптимізаційної задачі, яка розв'язується евристичними алгоритмами. Для побудовання нелінійних класифікаторів використовується розширення простору та функції ядер. Метод SVM у тестах перемагає інші методи за швидкістю та точністю категоризації. При різному підході до вибору ядер метод може емулювати роботу інших математичних методів. SVM може працювати як нейронна мережа, проте таке використання обмежує його призначення, оскільки метод значно перевершує їх за можливостями.

Каскадний класифікатор побудований на основі дерев та реалізує алгоритм виявлення осіб Віоли-Джонса.

Метод складається з двох частин – навчання та розпізнавання. Перед тим, як почати обробляти цільові зображення, необхідно провести навчання та сформувати деяку базу даних, що складається з ознак. Як ознаки

використовуються ознаки Хаара, які були так названі через візуальну схожість з вейвлетом Хаара (рис. 3.3 і 3.4).

Висока швидкість обробки зображень досягається за рахунок використання інтегрального представлення зображень: зображення, що зберігається, представлене у вигляді матриці, кожен елемент якої дорівнює сумі всіх елементів, які розташовані вище і лівіше. Інтегральне уявлення корисно тим, що дозволяє швидко обчислювати суму деякого набору осередків матриці, незалежно від розміру області, що обчислюється, знаходження суми завжди займає константний час. Ознаки Хаара застосовуються до будь-якої області зображення, прямокутник із чорною та білою областю накладається як маска на зображення, знаходяться суми вмісту осередків темної та світлої області. Потім знаходиться різниця між темною та світлою областю, і в результаті виходить число – воно називається значенням ознаки.

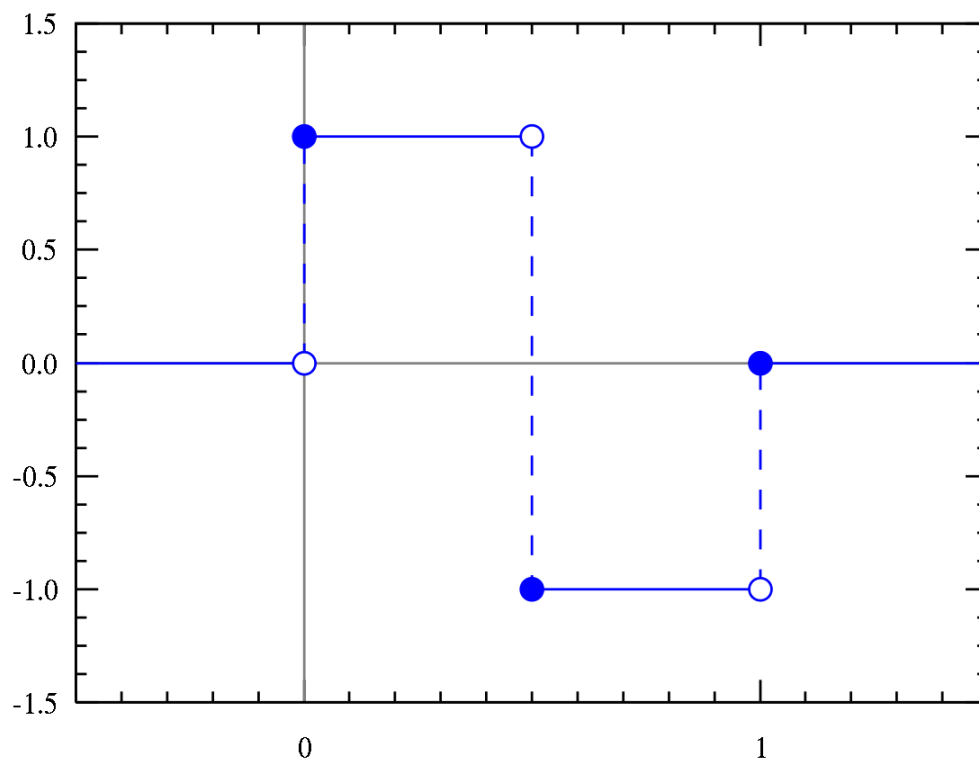


Рисунок 3.3 - Вейвлет Хаара

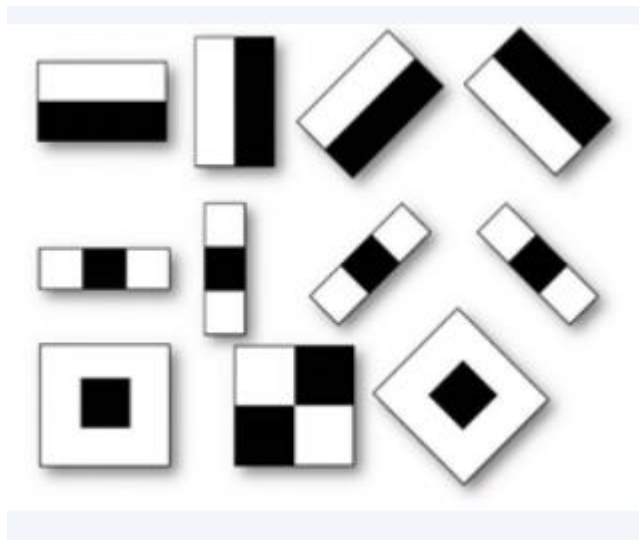


Рисунок 3.4 - Ознаки Хаара

Каскадні класифікатори, засновані на алгоритмі Віоли-Джонса, мають як переваги, так і недоліки. Вони працюють швидко через застосування інтегрального уявлення, тому їх можна використовувати для обробки відео в реальному часі, але для навчання потрібна велика вибірка. Працює дуже точно, якщо об'єкт, що шукається, не дуже відрізняється від зображень об'єктів, які були використані при навчанні, якщо ж відмінності сильні (для осіб приклад – сильний нахил голови, взаємне розташування частин особи порушується), тоді об'єкт взагалі не розпізнається.

3.2 Реалізація розпізнавання транспортних засобів на C#

Для розробки своєї системи використано середовище Microsoft Visual Studio Windows Forms та мова програмування C#.

Windows Forms - це інтерфейс для створення програм для настільних ПК Windows. Він пропонує один з найбільш продуктивних способів створення настільних застосунків на основі візуального дизайнера, що надається у Visual Studio. Функціональні можливості, такі як перетягування та розміщення візуальних елементів керування, дозволяють легко створювати настільні програми. а допомогою Windows Forms ви розробляєте графічно багаті програми, які легко розгортати, оновлювати та працювати в автономному режимі

або під час з'єднання з Інтернетом. Застосунки Windows Forms можуть отримати доступ до локальної апаратної та файлової системи комп'ютера, на якому запущена програма.

Концепція виявлення транспортних засобів полягає в тому, щоб вибрати зону інтересу щодо того, де транспортний засіб знаходиться на дорозі в режимі реального часу. Система має дуже високу складність через отримання зображень в режимі реального часу, оскільки оточення місця руху постійно змінюється. Складність виявлення транспортних засобів також залежить від зміни освітлення, погодних умов, тіні, складних фонових перешкод, параметрів камери, дрейфу та оклюзії транспортних засобів тощо. Тому виявити транспортні засоби в цих змінних умовах дуже важко. Для цього транспортні засоби повинні бути вилучені з послідовності зображень. Характеристика «рухомого» використовується, щоб відрізнити рухомий транспортний засіб від нерухомого фону. Точність або точність виявлення транспортних засобів має велике значення для вираження руху відстеження транспортних засобів і розуміння поведінки, і є основою для подальшої обробки. Як правило, алгоритм, що використовується для виявлення транспортних засобів, поділяється на чотири етапи: проблемна область, отримання зображення або відео, сегментація кадру та розпізнавання транспортних засобів, як показано на рис. 3.5.



Рисунок 3.5 - Узагальнена структура виявлення та розпізнавання транспортних засобів

На рис. 3.5 показана структура виявлення транспортних засобів на основі обробки відео. Використовуючи інформацію про проблемну область у відео, автомобіль ідентифікується з послідовності зображень. Потім, за допомогою вбудованої технології та розумної камери, відбувається отримання зображення.

Оскільки транспортний засіб рухається, отримання зображень може бути дещо складним.

Мій застосунок відтворює дорожнє відео та розміщує жовті мітки на кожній машині, яку він розпізнає. Під час відтворення відео застосунок веде поточний підрахунок всіх автомобілів, помічених на даний момент.



Рисунок 3.6 – Виявлення автобусу

Як можна побачити, застосунок розпізнає як машини, так і автобуси з фургонами. Також на зустрічній смузі можна побачити мотоцикліста, який опиняється у кадрі зовсім випадково. Тож спробуємо оптимізувати застосунок для якісного відбору.

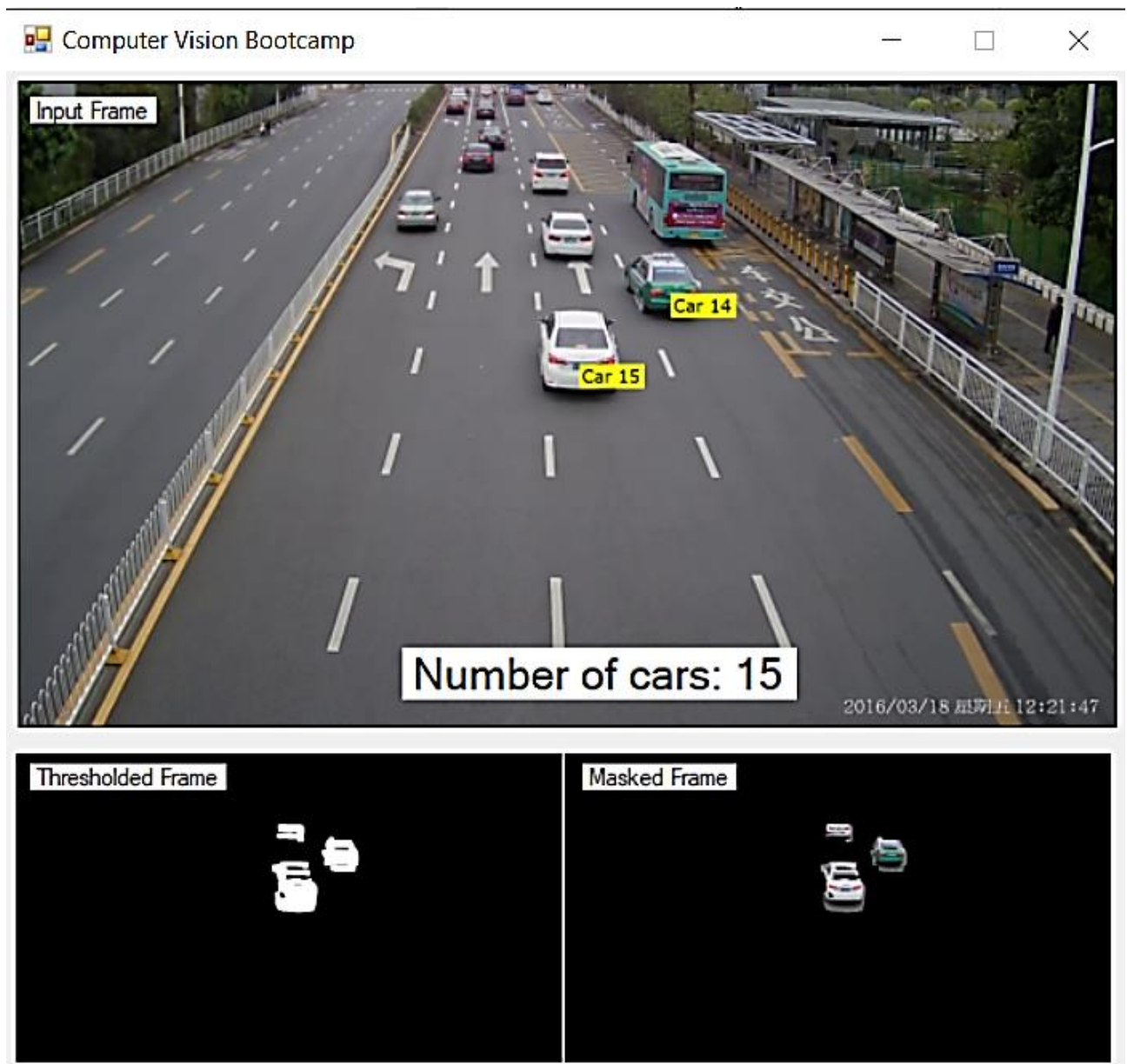


Рисунок 3.7 – Повний вигляд застосунку

Це складне завдання, тому що зазвичай камери для підрахунку трафіку встановлюються на іншій стороні дороги. У міру того, як автомобілі на камері збільшуються в розмірах, алгоритм відстеження руху може легко зафіксувати їх.

Але тут все навпаки. Автомобілі з'являються в нижній частині кадру і стають менше в міру віддалення від камери.

Нам також доводиться мати справу з випадковими мотоциклами, автобусами та зустрічним рухом на іншій смугі.

Як застосунок може впоратися з усім цим?

Нижче представлено, як вирішити цю проблему на C#. Буде використано чудову бібліотеку Accord, яка ідеально підходить для створення програм комп'ютерного зору.

Найбільша проблема — знайти машини на цьому відео. Контраст між автомобілями та асфальтом невеликий, тому досить складно побудувати конвеєр обробки зображень, який міг би чітко розділити їх.

Тим не менш, є один аспект автомобілів, який, безумовно, можна використати на свою користь: автомобілі рухаються.

Було взято два наступних відеокадри і зберегти тільки ті пікселі, які перемістилися. Це робить функції класу `ThresholdedEuclidianDifference`.

У підсумку отримано зображення з великою кількістю візуального шуму, тому використовується клас `BlobFiltering`, щоб зберегти лише великі рухомі краплі (транспортні засоби). Тоді можна збільшити результат у пару разів, щоб зробити машини більш помітними. Це дасть досить гарне підсумкове зображення.

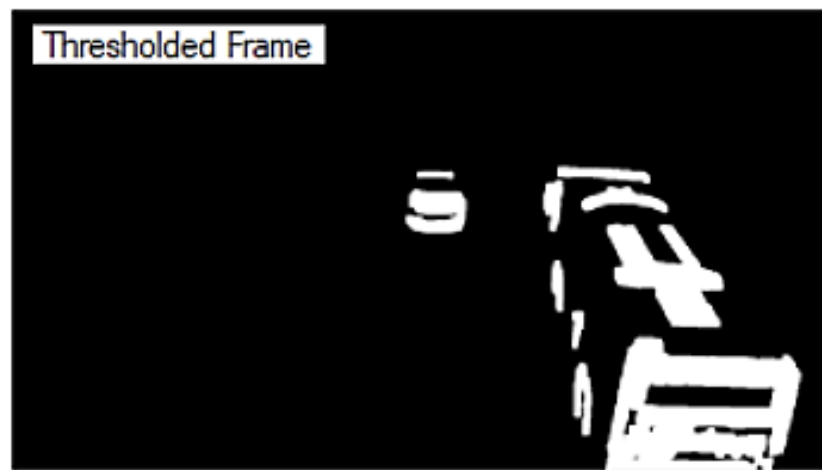


Рисунок 3.8 – Фільтрація та віднімання фону

Віднімання фону, яке також називають виявленням переднього плану, є одним із життєво важливих завдань, за допомогою яких можна реалізувати багато інших застосунків, таких як відстеження транспортних засобів, розпізнавання та виявлення нерівностей. Він використовується для сегментації руху в статичній сцені

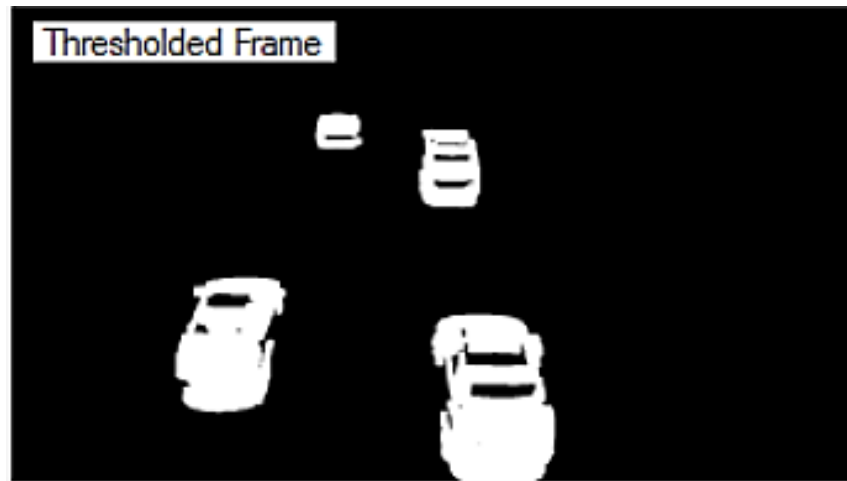


Рисунок 3.9 – Доповнення до рисунку 3.8

Тепер наявний оброблений відеокадр з автомобілями білого кольору, а все інше — чорним. Можна використовувати це зображення як маску з класом `ApplyMask`, щоб видалити все з відео, крім автомобілів, що рухаються.

Повний конвеєр обробки зображень виглядає так:

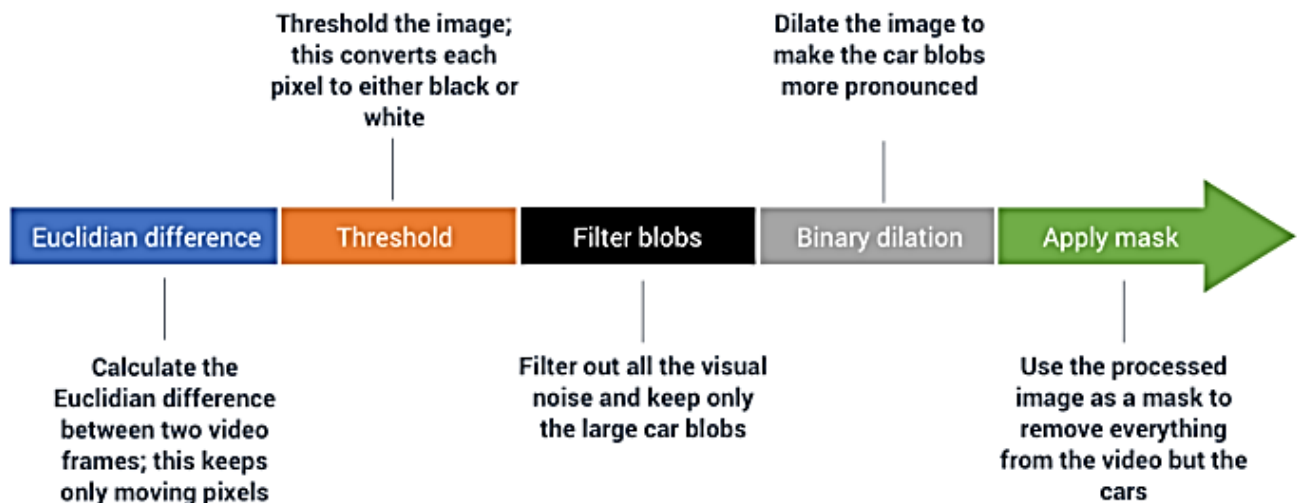


Рисунок 3.10 – Блок-схема конвеєру обробки зображень

Було відстежено поточний відеокадр і попередній кадр. Завантажено їх обидва в фільтр `ThresholdedEuclidianDifference` з пороговим значенням 40. Це призводить до отримання чорного зображення з тільки рухомими пікселями білого кольору.

На жаль, у відео є багато артефактів JPEG, які також відображаються як рухомі пікселі. Так що в результаті буде дуже багато шуму, і нам доведеться його прибрати.

Клас `BlobsFiltering` виконує виявлення великих двійкових об'єктів, а потім відкидає всі великі двійкові об'єкти, розмір яких менше заданого. Було вказано йому відкинути все, що менше 50x50 пікселів. Це позбавляє від шуму, але зберігає машини.

Потім тричі застосовується клас `BinaryDilation3x3`, щоб заповнити будь-які "дірки" у великих двійкових об'єктах і полегшити їх виявлення.

Двійкове розширення (`Binary dilation`) - це фільтр зображення, який додає застосунку пікселі до кордону фігури. Це класний трюк, що дозволяє закрити будь-які "дірки" у фігурі або зробити так, щоб дві різні фігури зливалися в одну.

Останній крок - використовувати клас `ApplyMask` для вилучення всіх пікселів, що відповідають розширеним автомобілям, з вихідного зображення. Це те, що можна побачити в правому нижньому вікні мого додатка.

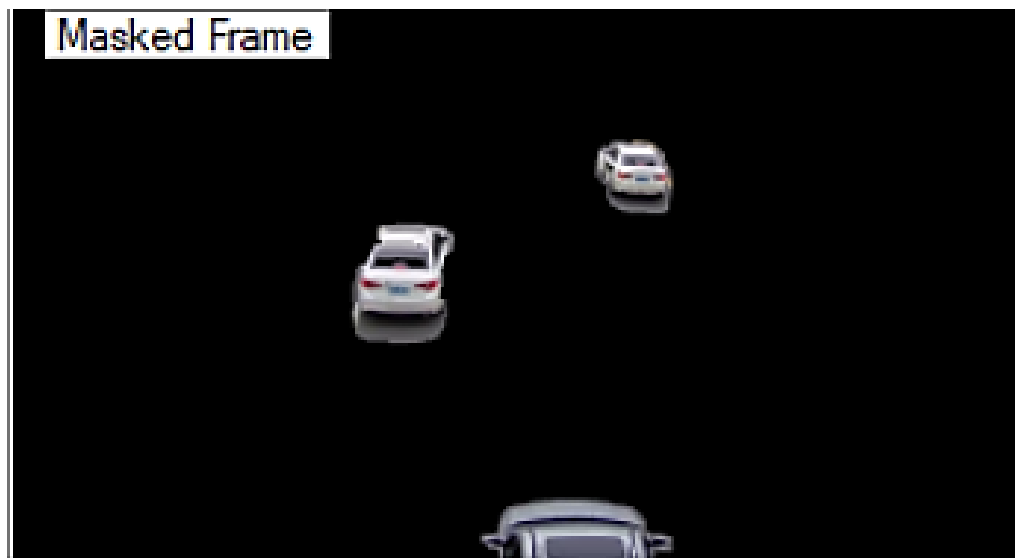


Рисунок 3.11 – Застосування класу `Apply Mask`



Рисунок 3.12 – Доповнення до рисунку 3.11

Отже, це повний конвеєр обробки зображень.

Наступний крок - створити своєрідний код відстеження, щоб стежити за окремими автомобілями, коли вони їдуть по дорозі. Завдання полягає в тому, щоб продовжувати відстежувати ті самі автомобілі від кадру до кадру замість того, щоб помилково ідентифікувати їх як нові автомобілі кожного разу.

Фільтр відстеження руху, такий як Camshift, тут ідеальний. Він може фіксуватися на рухомому двійковому об'єкті і відстежувати його від одного відеокадра до наступного.

Обрано стратегія полягає в тому, щоб використовувати лічильник великих двійкових об'єктів для визначення місця розташування кожного великого двійкового об'єкта в кадрі, а потім налаштувати окремі датчики перемикання передач для кожного автомобіля. Додано трохи додаткового коду, щоб ігнорувати будь-які великі бінарні об'єкти, які перекриваються з автомобільними трекарами, які вже створено.

Camshift дійсно бореться з об'єктами, які стискаються, а не ростуть. Так що на горизонті видно, як трекари від'єднуються від своїх машин, збільшуються до величезних розмірів або помилково перестрибують на іншу машину. Тому було додано трохи додаткового коду для попереднього видалення активних трекерів, перш ніж їхні машини стануть занадто малими, щоб їх можна було відстежити.

Ось блок-схема алгоритму виявлення транспорту:

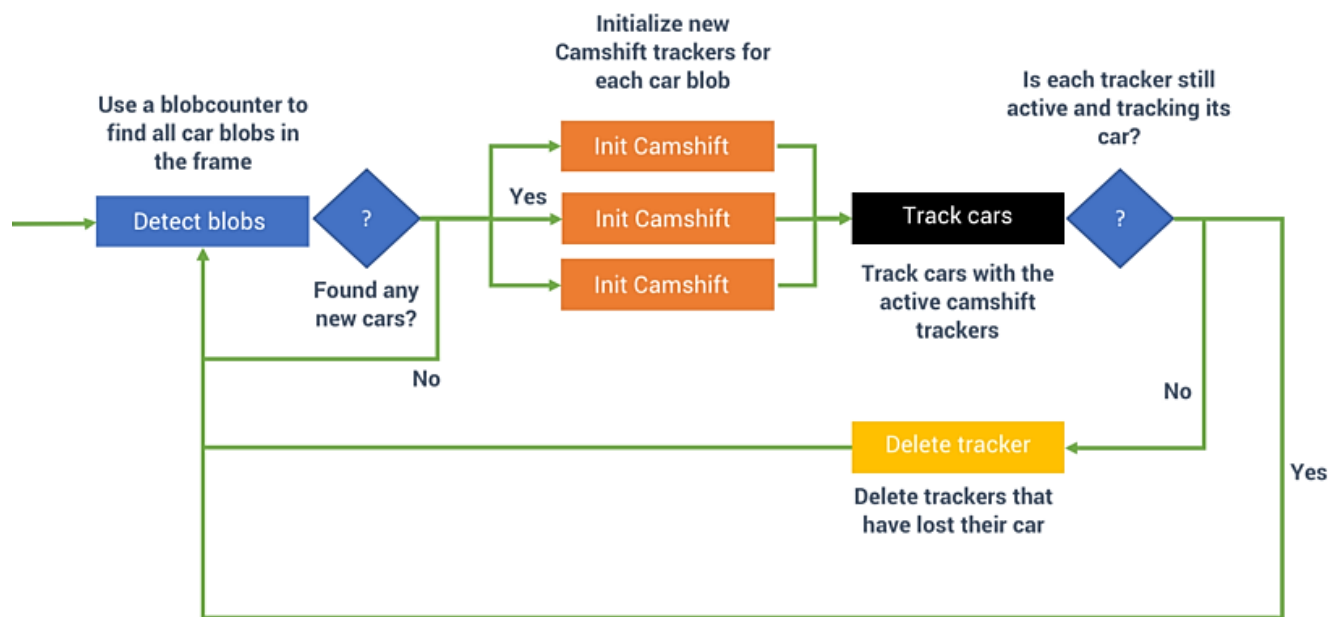


Рисунок 3.13 – Блок-схема алгоритму виявлення транспорту

Detect blobs - використання лічильника об'єктів, щоб знайти всі об'єкти автомобіля в кадрі. Якщо знайдено деякі нові машини, ініціалізуємо нові трекери для кожного автомобільного об'єкту. Якщо кожен трекер ще досі активний, то циклічно повертаємося на початок алгоритму, якщо ні, то видаляємо трекери, які більше не тримаються одного транспортного засобу. Після видалення повертаємося на початок.

Щоб легко створювати трекери перемикавання передач, їх було розміщено у допоміжному класі:

Клас **TrackerType** може відстежувати один автомобіль. Він містить внутрішній трекер перемикавання передач, який може відстежувати рух автомобіля від одного відеокадра до наступного. Він також запам'ятовує індекс відеокадра, коли трекер був вперше створений (зберігається в `startIndex`, тому можна видалити трекер, коли він стає занадто "старим"), і номер запису, який збільшується з 1.

Запускається **BlobCounter** кожні 10 кадрів, щоб прискорити роботу мого застосунку. Для кожного виявленого великого двійкового об'єкта перевірено наступне:

Чи перетинається blob з будь-якими відомими автомобілями?

Чи перекривається великий двійковий об'єкт з будь-яким іншим великим двійковим об'єктом?

Чи знаходиться крапля все ще нижче центральної лінії рамки?

Чи є великий бінарний об'єкт висотою менше 100 пікселів?

Якщо будь-яке з цих умов є правильним, великий бінарний об'єкт буде відкидано. Це залишає тільки нові автомобілі, які увійшли в кадр, тому буде створено нові треки перемикачів передач для кожного з них.

На горизонті перемикачів передач має тенденцію збиватися з пантелику. Він від'єднається від своєї машини, і прямокутник стеження швидко роздується. Тому буде видалено будь-який трекер, висота якого перевищує 360 пікселів або який старше 30 кадрів.

Нарешті, всім іншим трекерам вказано обробляти поточний кадр і відстежувати свої машини, і малюються мітки на кожному з них.

Це працює досить добре. У відео можна побачити, що застосунок ідентифікує більшість автомобілів і бореться лише з двома автомобілями, які не виявлені.

Одна з цих машин наведена на рисунку 3.14, на середній смузі.



Рисунок 3.14 – Недоліки розпізнавання

На наступному рисунку можна побачити, що система виявляє також автомобілі, які прямують у протилежному напрямку на зустрічній смузі. Цей аспект і є головним новаторством, який було виділено у програмі.

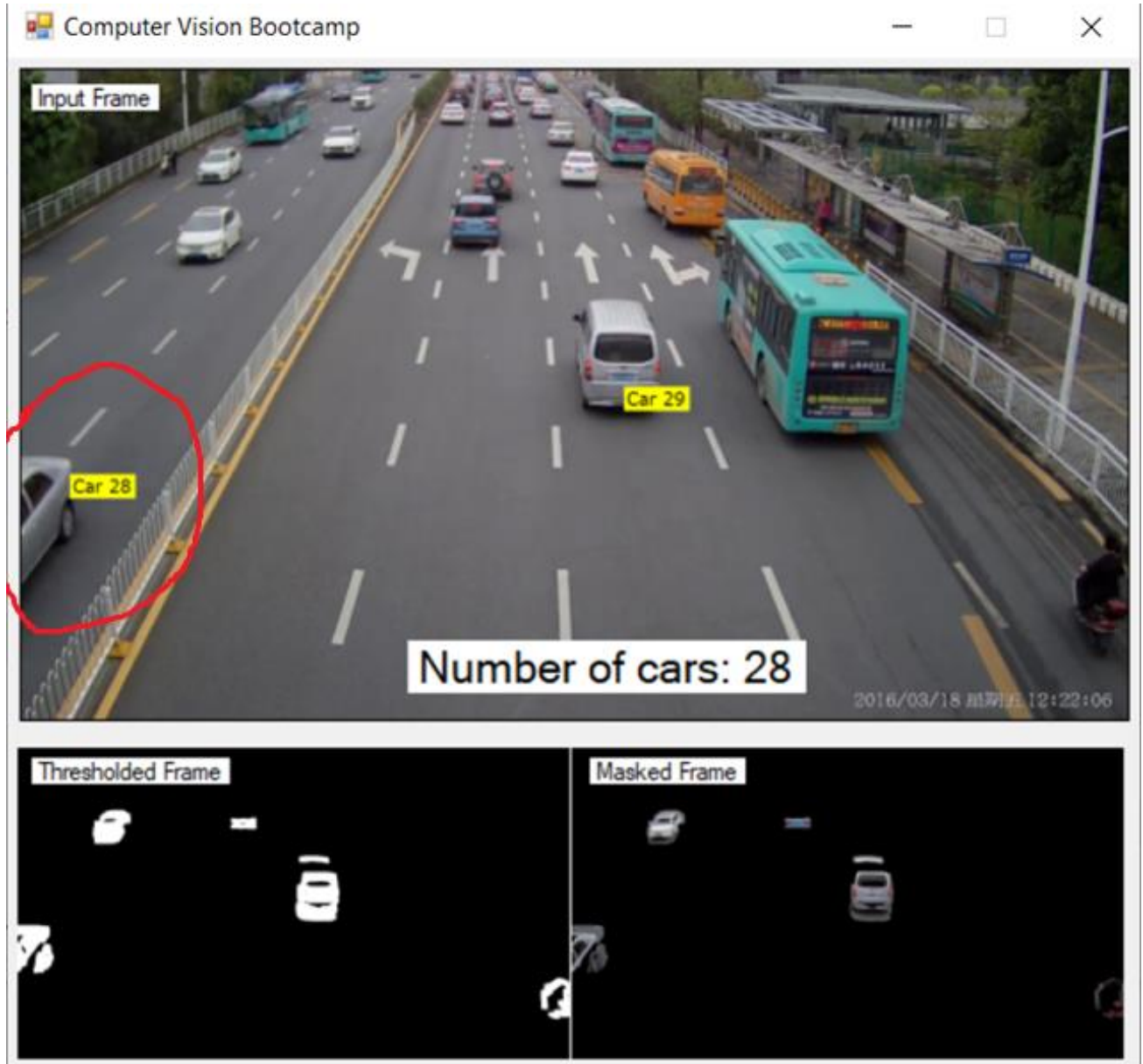


Рисунок 3.15 – Детекція автомобілів на зустрічній смузі

Висновки до розділу 3

Тож з результатів тестування програми було зроблено висновок, що за допомогою сучасних методів та підходів можна розв'язати складнорівневу задачу розпізнавання динамічних об'єктів, навіть в умовах низької контрастності.

Звісно, розроблена система не є ідеальною та не впоралася з деякими чорними рухаючимися транспортними об'єктами. Фільтрація недостатньо чітко сортує автомобілі цього кольору та стискається з труднощами. Можливо є інші, більш пропрацьовані методи глибинного навчання(наприклад LP detection, про який наведено інформацію та приклади у розділі 2), точність яких буде майже 100-відсотковою. Але вивчення та реалізація цих підходів потребує більше часу та глибоке занурення у нейронні мережі. Даний застосунок створений для максимально спрощеного та ефективного процесу рішення поставленої задачі, алгоритм якого буде зрозумілий некваліфікованій у цій науці людині.

4 ДОСЛІДЖЕННЯ ТА ПОРІВНЯННЯ З ІНШИМИ МОДЕЛЯМИ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ТРАНСПОРТУ

Для виявлення транспортних засобів у програмі було використано вбудований набір даних з бібліотеки Accord, а саме – Accord.Vision. Але як можна побачити з результатів виконання системи, є автомобілі, з якими вона не впоралася. Тому для порівняння хотілося оглянути та проаналізувати інші датасети, їх функціонал та ефективність.

AlexNet

AlexNet взяла участь у великомасштабному конкурсі візуального розпізнавання ImageNet у вересні 2012 року. Мережа досягла помилки в топ-5 в розмірі 15,3%, що на 10,8% пунктів нижче, ніж та, що зайняла друге місце. Архітектура, показана на рисунку 4.1, складається з восьми шарів: п'яти згорткових шарів і трьох повністю пов'язаних шарів [20].

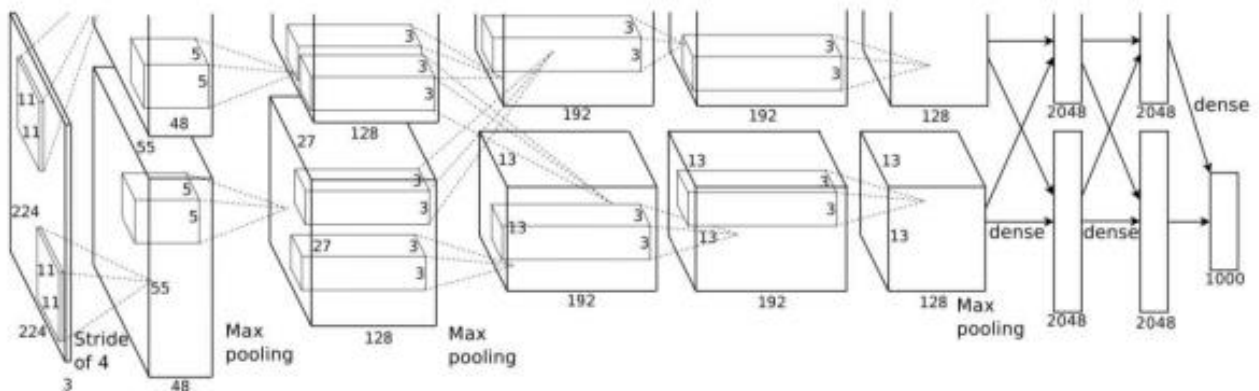


Рисунок 4.1 – Архітектура мережі AlexNet

AlexNet використовує випрямлені лінійні одиниці (ReLU) замість функції tanh. За допомогою ReLU, AlexNet досяг 25% помилки в наборі даних CIFAR-10 в шість разів швидше, ніж CNN використовує tanh. Крім того, AlexNet дозволяє проводити навчання на декількох графічних процесорах, розділяючи нейрони моделі на кілька графічних процесорів. Поліпшення AlexNet стосувалися швидкості і рішення проблеми переоснащення, оскільки AlexNet має 60 мільйонів параметрів. Отож, використовується метод, званий відсівом, який складається з повороту нейронів з заздалегідь певною ймовірністю 40%,

наприклад, це змушує кожен нейрон мати більш надійні функції, які може використовуватися з іншими випадковими нейронами.

VGG-16

VGG-16, запропонований К. Симонян і А. Зіссерман з Оксфордського університету у статті "Дуже глибокі згорткові мережі для великомасштабного розпізнавання зображень", яка була представлена на ILSVRC-2014 [21]. Це покращує AlexNet за рахунок заміни фільтрів великого розміру ядра на кілька фільтрів розміром з ядро 3x3 один за іншим. Модель досягає 92,7% точності тестування, що входить в топ-5 в ImageNet. Архітектура VGG-16 складається в цілому з 16 шарів. Вхідні дані для першого згорткового шару мають розмір 224 x 224 RGB зображення. За яким слід стопка згорткових (conv.) шарів, в яких використовувалися фільтри з малим полем сприйняття зі значенням кроку 1. Просторове об'єднання досягається за допомогою максимального об'єднання з кроком 2. Нарешті, слідує три повністю з'єднаних шарів (FCs). Перші два мають 4096 каналів кожен, а третій, що має кілька каналів в залежності від кількості класів, діє як класифікатор.

	Layer	Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

Рисунок 4.2 – Архітектура VGG-16

Однак мережа VGG-16 повільно навчається, а самі ваги мережевої архітектури досить великі. У порівнянні з іншими мережами, VGG-16 має більше параметрів з меншою точністю.

Як і у випадку з будь-яким іншим штучним об'єктом, транспортні засоби, як правило, слідує певним пропорціям, які не обов'язково мають функціональне призначення, а скоріше екстатичне. На рис. 4.3 показані різні марки і моделі з однаковими пропорціями. Відстань між колесами дорівнює триразовому діаметру колеса; відстань між капотом і колесом також дорівнює одному колесу. Крім того, висота вікна становить одну третину висоти автомобіля. Всі ці обмеження накладають більш незначні відмінності між моделями з боків автомобіля.

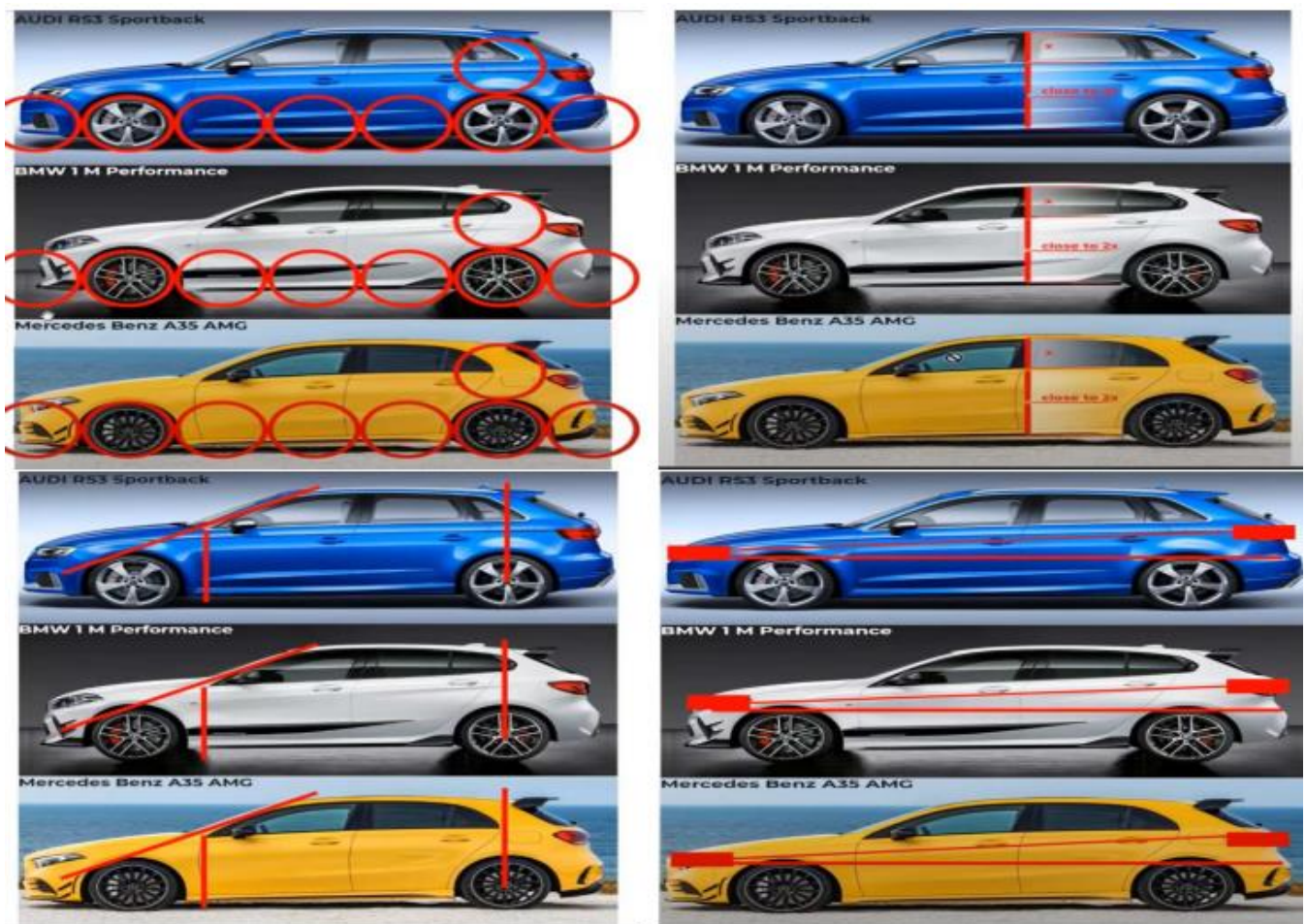


Рисунок 4.3 - Такі самі пропорції дизайну для різних марок і моделей

У таблицях 4.1 і 4.2 наведені результати VGG 16 з пакетною нормалізацією для їх відповідних обрізаних частин [22].

Навчальний набір обрізаних деталей є підмножиною навчального набору; те ж саме відноситься і до підмножини тестового набору.

Деякі частини більш описові, ніж інші. Наприклад, в бампері заднього виду базовий рівень VGG16 досягає 92,60%, в той час як базовий рівень vgg16 переднього правого світла досягає 61.76%. Грати і багажник - єдині дві частини, на яких є логотип марки і назва моделі, дає найкращі результати, в той час як ліві і праві фари як спереду, так і ззаду дають найменші результати. Як згадувалося в попередньому розділі, фари в основному складаються зі скла, тому більшість змін залежить від навколишнього оточення, що не допускає великих відмінностей між моделями.

Таблиця 4.1 – Окремі лицьові частини веб-даних CompCars.

Part	VGG16	Train	Test
Bumper	92.60%	7022	6740
Hood	92.26%	6988	6757
Grilles	93.83%	6622	6385
Left Light	64.72%	5875	5683
Right Light	61.76%	5942	5790

Таблиця 4.2 – Окремі задні частини веб-даних CompCars

Part	VGG16	Train	Test
Bumper	89.56%	5603	5312
Trunk	93.90%	5225	5221
Left Light	72.89%	5124	4859
Right Light	82.10%	5587	5300

4.1 Багатопотокове динамічне злиття

У цьому підрозділі порівнюються різні комбінації деталей, в таблиці 4.3 показана швидкість розпізнавання найбільш ефективної комбінації для кожного порту перегляду, всього 19 комбінацій.

В середньому, вид спереду більш розбірливий, ніж вид ззаду: комбінація 9, де всі частини спереду без повного зображення, набагато краще, ніж її аналог ззаду (84,12% - 65,45%). Додавання більшої кількості деталей може знизити продуктивність; наприклад, комбінація 1 краще, ніж комбінація 7, хоча всі деталі в 1 також входять в 7. Деякий деталі можуть значно знизити продуктивність, наприклад, комбінація 19 у порівнянні з комбінацією чотири або комбінацією 5, з 91,23% до 45,36% і 46,85% відповідно [23].

Таблиця 4.3 – Злиття з використанням різних комбінацій у веб-даних CompCars. FB: передній бампер, FH: передній капот, FG: передні решітки, повний: Передній лівий ліхтар, дль: Передній правий ліхтар, RB: задній бампер, RT: задній багажник, RL: задній лівий ліхтар, RRL: Задній правий ліхтар

Index	Combination	VGG16
1	full image + FB	96.12%
2	full image + FH	94.33%
3	full image + FG	92.74%
4	full image + FLL	45.36%
5	full image + FRL	46.85%
6	All Front parts	91.36%
7	full image + FB + FH + FG	92.65%
8	full image + FLL + FRL	90.12%
9	FB + FH + FG + FLL + FRL	84.12%

Index	Combination	VGG16
10	full image + RB	91.11%
11	full image + RT	94.14%
12	full image + RLL	68.12%
13	full image + RRL	65.45%
14	full image + RB + RT + RLL + RRL	88.24%
15	RB + RT + RLL + RRL	65.26%
16	full image + RB + RT + RLL	87.26%
17	full image + RB + RT	89.22%
18	full image + RLL + RRL	62.25%
19	full image	91.23%

Закінчення таблиці 4.3

У таблиці 4.4 наведені останні результати по набору даних CompCars, де підходи без глибоких згорткових мереж дають найгірший результат, наприклад, Yang . VoxCar і Baseline VGG16 покладаються на глибокі мережі для представлення глобальних функцій. Проте найкращі результати, такі як SWP-CNN та WindowResnet, використовували підходи, засновані на деталях. Наш підхід поєднував у собі як глобальне, так і локальне представництво, що робило систему надійною.

Таблиця 4.4 – Порівняння підходів

Approach	CompCars (web)
Yan	76.7%
BoxCars	84.8%
Baseline	92.66%
VGG16	
Ours (VGG16)	95.07%
SWP-CNN	97.6%
WindowResnet	97.8 %

4.2 Майбутні перспективи

Протягом багатьох років аспекти глибокого навчання були повною загадкою, і дослідницька спільнота більше зосереджувалася на тому, що може зробити глибоке навчання, а не на розумінні того, чому воно поводить певним чином, наприклад, чому великі нейронні мережі працюють краще, ніж більш дрібні, і як вони узагальнюються з такою кількістю параметрів. Однак зовсім недавно деякі роботи дають захоплююче уявлення, яке дозволить майбутнім дослідженням розкрити справжній потенціал глибоких нейронних мереж.

4.2.1 Гіпотеза про лотерейний квиток

Хоча гіпотеза лотерейного квитка, запропонована Франклом і Карбіном, є лише протестована на невеликих наборах даних, таких як MNIST, він забезпечує дуже вражаючу оптимізацію, скорочуючи кількість параметрів до 90% [24]. Що вказує на глибоке розуміння внутрішньої роботи нейронних мереж. Гіпотеза розглядає нейронні мережі як гігантські завдяки випадковій ініціалізації конкретні підмережі математично пощастило, і оптимізатор

визнає їх потенціал, в той час як інша частина мережі не вносить великого внеску. Хотілося б попрацювати над аналогічними методами оптимізації, які використовують відмінне розуміння нейронних мереж для більш складних завдань.

4.2.2 Навчання з нульовим пострілом

Навчання з нульовим пострілом-це захоплююча концепція. Нульовий постріл - це коли модель може класифікувати невидимі класи без будь-яких навчальних прикладів. Це може здатися притягнутим за вуха, але людський мозок здатний саме на це [26]. У цій роботі представлено навчання з кількома пострілами, де кожна особистість в задачі повторної ідентифікації має одне позначене зображення.

Хоча методи глибокого навчання в тих же випадках могли перевершити навіть людей у деяких візуальних завданнях, вони завжди покладалися на величезну кількість даних.

Метанавчання

У цій роботі описано деякі парадигми навчання. Хоча MetaLearning є ще новим напрямком(навчаюсь вчитися), це захоплююче. Метанавчання - це наука про спостереження як різні підходи учнів працюють з цілим рядом навчальних завдань, а потім беруть уроки з цього досвіду, щоб освоювати нові завдання набагато швидше, ніж це можливо в іншому випадку. Це дозволяє нам роздумувати і надихатися нашим власним досвідом в якості учнів.

4.2.3 Самокеровані автомобілі

Самокеровані автомобілі - це програма глибокого навчання, яка особливо цікавить у переслідуванні. Це вимагає як швидкості реагування на зміну візуальних даних, так і точність [29]. Навколишнє середовище може змінитися радикально і дуже швидко. Крім того, як тільки транспортний засіб опиниться на дорозі, він виявить об'єкти, з якими він не стикався під час

навчання, на які йому необхідно реагувати. Це вимагає дуже ефективного використання парадигм машинного навчання.

Ця програма може довести методи глибокого навчання до межі. Отже, це дозволяє дослідникам видобувати цінні дані та отримувати уявлення про внутрішню роботу deep learning. Зокрема, налаштування нульового пострілу, розглянута вище.

Висновки до розділу 4

Отже, можна побачити, що було створено багато визначних моделей машинного навчання для розпізнавання, такі як AlexNet та VGG-16. Друга модель покращує AlexNet за рахунок заміни фільтрів великого розміру ядра на кілька фільтрів розміром з ядро 3x3 один за іншим. Вона досягає 92,7% точності тестування, що входить в топ-5 в ImageNet.

Спеціальний розділ
ОХОРОНА ПРАЦІ

до кваліфікаційної роботи

на тему:

**«Розпізнавання транспортних засобів за допомогою
технології комп'ютерного зору»**

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.2181005

Виконала студентка 4-го курсу, групи 401

_____ *А.В.Вишневецька* _____
(підпис, ініціали та прізвище)

«__» червня 2022 р.

*Консультант_ст. викладач каф. екології
Макарова О.В.*

(наук. ступінь, вчене звання)

_____ *(підпис, ініціали та прізвище)*

«__» червня 2022р.

5 СПЕЦІАЛЬНИЙ РОЗДІЛ З ОХОРОНИ ПРАЦІ

Охорона здоров'я та безпеки праці зосереджена насамперед на захисті працівників на робочому місці від нещасних випадків, травм та впливу шкідливих речовин. Незважаючи на те, що нещасні випадки можуть статися в будь-який час, роботодавець несе відповідальність за те, щоб він вжив заходів для зниження ризику інцидентів і підтримки безпечного робочого середовища.

Тож **метою** даного розділу є створення безпечних і здорових умов праці на робочому місці або у виробничому приміщенні та забезпечення безпеки людини у надзвичайних ситуаціях.

В даній дипломній роботі створено програму розпізнавання транспортних засобів за допомогою комп'ютерного зору. З метою створення безпечних і здорових умов праці при створенні цієї програми виконується аналіз умов на робочому місці.

Виходячи з мети можна сформулювати наступні **завдання**:

1. Виконати опис серверного приміщення, робочого місця адміністратора та виробничого обладнання.
2. Оцінити умови праці у серверному приміщенні.
3. Сформулювати рекомендації щодо поліпшення умов праці на робочому місці адміністратора.

Пріоритетність охорони праці у даних умовах має кілька ключових переваг, зокрема:

- зменшення ризику, нещасних випадків чи травм шляхом виявлення та зменшення небезпек;
- підвищення ефективності та продуктивності завдяки тому, що менше співробітників пропускають роботу через хворобу або травму;
- покращені відносини і моральний дух співробітників (безпечніше робоче середовище - це менш стресове робоче середовище);

- зменшення витрат, пов'язаних з нещасними випадками або травмами (витрати на охорону здоров'я та реабілітацію, втрати продуктивності, вплив на добробут співробітників).

5.1 Аналіз умов праці

Характеристика приміщення

Розробка програми велася у приміщенні, план якого приведено на рис.1



Рисунок 5.1 – План приміщення

Приміщення має одностороннє природне освітлення та загальне штучне освітлення.

Ширина приміщення 2,5 м, довжина – 3,5 м, висота стелі – 2,7 м. Кількість робочих місць – одне. Приміщення знаходиться на п'ятому поверсі дев'ятиповерхової будівлі. Площа – 8,75 м², об'єм – 23,6 м³. Виходячи з цього отримано дані наведені в таблиці 1.

Таблиця 5.1 - Фактичні та нормативні значення параметрів приміщення.

Параметр	Норма	Фактичні параметри
Площа, S	Не менше ніж 6 м ²	8,75 м ²
Об'єм, V	Не менше ніж 20 м ³	23,6 м ³

Показники відповідають вимогам нормативних документів.

Мікрокліматичні умови

Відповідно до державних санітарних правила і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [26], ця робота відноситься до категорії легка 1а. Оптимальні та фактичні параметри мікроклімату наведені у таблиці 5.2.

Таблиця 5.2. Значення мікроклімату

Період року	Параметр	Оптимальний	Фактичний
Теплий	Температура	23 – 25	28
	Вологість	60 – 40	50
	Швидкість повітря	< 0,1 м/с	
Холодний	Температура	22 – 24	22
	Вологість	60 – 40	40
	Швидкість повітря	< 0,1 м/с	

Значення відносної вологості повітря в холодний період року знаходиться на межі допустимих значень, доцільно використовувати в цей період зволожувачі повітря.

Температура повітря в теплий період року виходить за межі допустимих значень, для забезпечення вимог [26] необхідно встановлення кондиціонера.

1.3 Освітлення

Норми освітленості регламентуються СНиП 23-05-95 «Природне і штучне освітлення» [27]. Згідно з санітарними нормами мікроклімату виробничих приміщень робота відноситься до розряду Шв. При загальному освітленні показник освітлення робочої поверхні має бути у межах 200 – 400 лк.

У даному приміщенні використовується природне, штучне та змішане освітлення.

Джерелом природного освітлення є вікно шириною 1,7 м і висотою 1,8 м.

Джерелом штучного освітлення є світильник з п'ятьма світлодіодними лампами потужністю 18 Вт, світловий потік 1800 лм.

Для визначення освітленості робочої зони скористаємось методом світлового потоку.

Формула світлового потоку має вигляд:

$$\Phi = \frac{EkSZ}{N\eta}, \quad (5.1)$$

де Φ – світловий потік, Лм;

E – освітленість робочого місця, Лк;

k – коефіцієнт запасу, що враховує зменшення світлового потоку ламп у процесі експлуатації, $k = 1,2$;

Z – коефіцієнт нерівномірності, $Z = 1,1$;

S – площа приміщення, м²;

N – кількість ламп;

η – коефіцієнт використання світлового потоку.

Звідси освітленість на робочому місці дорівнює:

$$E = \frac{\Phi N \eta}{kSZ} \quad (5.2)$$

Для визначення коефіцієнту використання світлового потоку η потрібно розрахувати індекс приміщення i за формулою 5.3:

$$i = \frac{S}{h(A+B)} \quad (5.3)$$

де S – площа приміщення, $S = 8,75$ м²;

h – висота світильників над робочою поверхнею, м;

A - ширина приміщення, $A = 2,5$ м;

B - довжина приміщення, $B = 3,5$ м.

Висота світильників над робочою поверхнею знаходиться за формулою 5.4

$$h = H - h_{\text{св}} - h_{\text{рп}} \quad (5.4)$$

де H – висота приміщення, м;

$h_{\text{св}}$ – висота світильника, м;

$h_{\text{рп}}$ – висота робочої поверхні, м.

$$h = 2,7 - 0,3 - 0,8 = 1,6 \text{ м}$$

$$i = \frac{8,75}{1,6 (2,5 + 3,5)} = 0,91$$

За індексом приміщення та коефіцієнтами світлового потоку від стелі – 70%, стін – 50%, підлоги – 30 % визначаємо значення коефіцієнту використання світлового потоку $\eta = 0,5$.

Підставимо всі значення у формулу 2 для визначення освітленості:

$$E = \frac{1800 \cdot 5 \cdot 0,51}{1,2 \cdot 8,75 \cdot 1,1} = 397 \text{ Лк}$$

Освітленість на робочому місці становить 397 лк, що відповідає вимогам для Шв розряду зорових робіт.

Завдяки двом великим вікнам немає дискомфорту з природнім освітленням. Електроживлення освітлення серверного приміщення та електроживлення телекомунікаційного обладнання, встановленого в серверному приміщенні, подається від різних розподільних електричних щитів. Світильники розміщуються на стелі.

Для управління освітленням користуватися одним або декількома вимикачами і розташовувати їх поряд з дверима на висоті 1,5 м від рівня підлоги.

1. Пожежна безпека

Пожежна безпека в закладах системи освіти України регламентується Правилами пожежної безпеки в Україні, затвердженими наказом Міністерства внутрішніх справ України від 30.12.2014 № 1417 (далі – НАПБ А.01.001-2014) [29] та

галузевими Правилами пожежної безпеки для навчальних закладів та установ системи освіти України, затвердженими наказом Міністерства освіти і науки України від 15.08.2016 № 974 (далі – НАПБ В.01.050-2016/920) [28].

Так, відповідно до розділу II НАПБ А.01.001-2014 та розділу I НАПБ В.01.050-2016/920 на об'єктах із постійним або тимчасовим перебуванням на них 100 і більше осіб або таких, що мають хоча б одне окреме приміщення з одночасним перебуванням 50 і більше осіб, пожежна безпека забезпечується шляхом проведення організаційних і практичних заходів та використання технічних засобів.

Відповідальність за пожежну безпеку закладів та установ освіти покладається на їх власників або уповноважених ними органи або керівників (ректори, директори, начальники, завідувачі; далі – керівники навчальних закладів та установ).

Відповідно до розділу II НАПБ В.01.050-2016/920 керівниками навчальних закладів та установ повинні призначатися особи, відповідальні за пожежну безпеку окремих будівель, споруд, приміщень, інженерного обладнання та утримання й експлуатацію засобів протипожежного захисту, що має бути передбачено у їхніх функціональних обов'язках, посадових інструкціях тощо.

У кожному навчальному закладі та установі освіти наказом чи інструкцією встановлюється протипожежний режим, що містить необхідні для виконання організаційні заходи щодо забезпечення пожежної безпеки, вимоги до утримання територій, будинків, приміщень, споруд, евакуаційних шляхів і виходів; до інженерного обладнання; до технічних засобів протипожежного захисту, вимоги пожежної безпеки під час проведення вогневих, фарбувальних та будівельно-монтажних робіт, а також порядок дій у разі пожежі.

Крім цього, на об'єкті повинні бути розроблені та затверджені керівником схема плану евакуації учнів і вихованців на випадок пожежі та інструкція, що визначає дії працівників закладу чи установи на випадок термінової евакуації учасників навчально-виховного процесу, за якою не рідше одного разу на півроку мають проводитися практичні тренування всіх працівників.

План евакуації та порядок евакуації повинні переглядатися один раз на три роки.

Таблиця 5.1 - Фактори умов праці на робочому місці

№ з/п	Фактор умов праці на робочому місці	Значення показника	Тривалість дії фактора, хв.
1	Температура повітря на робочому місці (РМ) у виробничому приміщенні, °С: - теплий період; - холодний період	22 -	420 -
2	Відносна вологість повітря на РМ, %	70	420
3	Швидкість руху повітря на РМ, м/с	0,25	480
4	Освітленість на РМ, лк	150	240
5	Мінімальний розмір об'єкта розпізнавання, мм	0,5	240
6	Виробничий шум, дБА	62	480
7	Інтенсивність теплового випромінювання, Вт/м ²	180	420
8	Токсична речовина, озон, кратність перевищення ГДК	5	480
9	Виробничий пил (паперовий та ін.), кратність перевищення ГДК	1	480
10	Робоче місце (РМ), поза та переміщення у просторі	Робоче місце стаціонарне, маса переміщення вантажу до 5 кг	480
11	Кількість важливих об'єктів спостереження	6	420
12	Тривалість зосередженого спостереження, % часу зміни	75	360

2. Техніка безпеки

Згідно з нормативним документом НПАОП 40.1-1.01-97 Правила безпечної експлуатації електроустановок (ДНАОП 1.1.10-1.01-97), всі виробничі маніпуляції на працюючих електроустановках повинні оформлятися офіційним завданням на стандартному бланку (т.зв. наряд-допуск), в якому зазначаються:

- зміст і місце проведення робіт;
- початок і кінець маніпуляцій;
- умови електробезпеки для їх здійснення;
- список працівників;
- прізвище відповідального за безпеку.

Також можливе здійснення таких робіт за розпорядженням керівництва і на підставі списку маніпуляцій, необхідних для поточного функціонування електроустановок. Проводити маніпуляції без підписаного наряду-допуску забороняється, не можна також самовільно збільшувати обсяг робіт або складу бригади. Якщо необхідно провести капітальний ремонт електричних агрегатів з напругою понад 1 тис. Вольт без виключення електроенергії або повітряних ліній електропередач, всі роботи повинні здійснюватися відповідно до технологічних карт, підписаними директором підприємства. Порядок робіт на електроустановках з 1 тис. Вольт і вище напругою наступний:

- вимикається напруга з струмопровідних ділянок, які перебувають біля робочого поля (або ставиться огорожу) для виключення випадкового торкання;
- надаватися діелектричне взуття, або використовується електроізолюючими килимок;
- готується інструмент з ізоляцією (наприклад, для викруток необхідно забезпечити електроізоляцію стрижня), або надягають діелектричні рукавички;
- якщо роботи проводяться в приміщеннях з електроприладами, в підземних колодязях, різних тунелях і т.п., надягають захисні каски.

Гігієна праці і виробнича санітарія

Виходячи з принципів Гігієнічної класифікації праці, умови розподіляють на 4 класи:

- 1) клас - оптимальні умови праці - такі умови, при яких зберігається не лише здоров'я працюючих, а й створюються передумови для підтримання високого рівня працездатності;
- 2) клас - допустимі умови праці - характеризуються такими рівнями факторів виробничого середовища і трудового процесу, які не перевищують встановлених гігієнічних нормативів, а можливі зміни функціонування стану організму відновлюються за час регламентованого відпочинку або до початку наступної зміни та не справляють негативного впливу на стан здоров'я;
- 3) клас - шкідливі умови праці - характеризуються такими рівнями шкідливих виробничих факторів, які перевищують гігієнічні нормативи і здатні чинити несприятливий вплив на організм. Шкідливі умови праці за ступенем перевищення гігієнічних нормативів та вираженості можливих змін в організмі людини поділяють на чотири ступені: перший ступінь характеризується рівнями шкідливих виробничих факторів, які викликають функціональні зміни, що виходять за межі фізіологічних коливань та збільшують ризик погіршення здоров'я; другий ступінь - це умови праці, які характеризуються рівнями шкідливих факторів виробничого середовища та трудового процесу, які викликають функціональні зміни, що виходять за межі фізіологічних коливань та збільшують ризик погіршення здоров'я; третій ступінь - умови праці, які характеризуються рівнями шкідливих факторів виробничого середовища і трудового процесу, які призводять до зростання виробничо обумовленої захворюваності, до розвитку професійних захворювань; четвертий ступінь - умови праці характеризуються рівнями шкідливих факторів виробничого середовища і трудового процесу, які здатні призводити до значного зростання хронічної патології та рівнів захворюваності з тимчасовою втратою працездатності, розвитку важких форм професійних захворювань (з втратою загальної працездатності);

- 4) клас - небезпечні (екстремальні) умови праці - характеризуються такими рівнями шкідливих факторів виробничого середовища і праці, вплив яких протягом робочої зміни створює загрозу для життя, високий ризик виникнення важких форм гострих професійних уражень.

Мікроклімат приміщень - умови внутрішнього середовища цих приміщень, що впливають на тепловий обмін працюючих з оточенням шляхом конвенції, кондукції, теплового випромінювання та випаровування вологи відповідно до ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.

Таблиця 5.2. - Класифікація робіт за важкістю та енерговитратами (ДСН 3.3.6.042-99)

Класифікація робіт	Характеристика робіт	Енерговитрати, Вт, (ккал/год)
I-а - легка	Виконуються сидячи і не потребують фізичного напруження	105- 140 (90-120)
I-б - легка	Виконуються сидячи, стоячи або в русі з незначними фізичними навантаженнями	141 - 175 (121-150)
II-а - середньої важкості	Виконуються в русі при переміщенні вантажів вагою до 1 кг або сидячи чи стоячи з фізичними навантаженнями	176-232(151-200)
II- важка	Пов'язані з хворобою, переміщенням та перенесенням вантажів вагою до 10 кг і супроводжуються помірним фізичним навантаженням	233-290(201-250)
III - важка	Постійне переміщення з перенесенням вантажів (понад 10 кг), що потребують значних фізичних зусиль	291 -349(251-300)

Значення параметрів мікроклімату суттєво впливають на самопочуття працездатність людини. Тривала дія високої температури тіла людини при одночасній підвищеній його вологості призводить до збільшення температури людини до 38-40 град, (гіпертермія). При підвищенні температури значно збільшується потовиділення, внаслідок чого настає різке порушення водного обміну.

З потом із організму виділяється значна кількість солей, головним чином хлористого натрію, калію, кальцію. Зростає вміст у крові молочної кислоти, сечовини, внаслідок чого вона згущується. Перегрів тіла людини супроводжується головними болями, запамороченням, нудотою, загальною слабкістю, часом можуть виникати судоми та втрата свідомості. Негативна дія високої температури збільшується при підвищеній вологості.

Суттєві фізіологічні зміни в організмі можливі також при низькій температурі, яка призводить до переохолодження організму (гіпотермія). Найбільш вираженими реакціями на низьку температуру є звуження судин м'язів та шкіри. Охолодження тіла викликає порушення рефлексорних реакцій, зниження тактильних та інших реакцій, утруднення рухів. Все вищевказане може бути причиною травматизму.

Недостатня вологість повітря (нижче 20%) призводить до підсихання слизових оболонок дихальних шляхів та очей, внаслідок чого зменшується захисна здатність протистояти мікробам.

Висновки до розділу 5

Аналіз умов праці в розглянутому робочому приміщенні показав, що умови праці з ПЕОМ відповідають вимогам, оскільки площа та об'єм не менше нормативних значень, рівні шуму, вібрації і загазованості не перевищують нормативних значень.

Рівень освітлення робочого місця відповідає нормам.

Для підтримання параметрів мікроклімату в приміщенні встановлено радіатор водяної системи центрального опалення, що складається з 12 секцій.

Ергономіка робочого місця і режим зорової роботи задовольняють вимогам і сприяють зниженню втоми.

ВИСНОВКИ

В результаті написання бакалаврської кваліфікаційної роботи було опрацьовано тему виявлення транспортних засобів у складному середовищі, що є ключовим завданням ІТС. У цій роботі розглянуто сучасні та футуристичні методи виявлення транспортних засобів, які використовуються у всьому світі. Дано короткий огляд класифікації транспортних засобів на основі методів машинного навчання та комп'ютерного зору. Виконано аналіз предметної області та алгоритмів машинного навчання для ідентифікації транспортних засобів, реалізовано застосунок розпізнавання транспортних засобів за допомогою комп'ютерного зору, а саме функціоналу фреймворку за допомогою методу опорних векторів, класичного класифікатора та фільтрації зображен. Здійснено порівняльний аналіз з іншими моделями для розпізнавання транспорту, заснованих на різних підходах.

З ростом числа транспортних засобів на дорогах більшість країн впроваджують інтелектуальні транспортні системи (ІТС) для вирішення таких питань, як щільність транспортного потоку, довжина черги, середня швидкість руху і загальна кількість транспортних засобів, що проходять через точку за певний проміжок часу і так далі. ITS, знімаючи зображення і відео дорожнього руху за допомогою камер, допомагає центрам управління дорожнім рухом в моніторингу та управлінні дорожнім рухом.

Незважаючи на те, що доступно багато алгоритмів, заснованих на методологіях, у формі найсучасніших засобів класифікації, виявлення та відстеження транспортних засобів, все ще виникають складнощі через різні проблеми, пов'язані з набором даних рухомих транспортних засобів. Моя система впоралася майже повністю з поставленою задачею, тому після виявлення транспортного засобу інші програми можуть бути використані більш ефективним чином.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Трофімов В.В. Інформаційні технології : підручник. Київ, 2015. 289 с.
2. М. А. Абдель Вахаб. Точний підхід до підрахунку транспортних засобів, заснований на глибоких нейронних мережах. У 2019 році Міжнародна конференція з інноваційних тенденцій в комп'ютерній Інженерії (ITCE): сторінки 1-5, 2019. (дата звернення: 10.04.2022).
3. Da Vinci Instruments *Intuitive / da Vinci Robotic Assisted Surgical Systems* : веб-сайт. URL: <https://www.intuitive.com/en-us/products-and-services/da-vinci/systems> (дата звернення: 10.04.2022).
4. Ерхан Бас, А. Текалп і Ф. Салман. Автоматичний підрахунок транспортних засобів по відео для аналізу транспортного потоку: сторінки 392-397, 07 2007. (дата звернення: 15.04.2022).
5. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speededup robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008: веб-сайт. URL: <http://dx.doi.org/10.1016/j.cviu.2007.09.014>. (дата звернення: 15.04.2022).
6. Amine ben khalifa and Nichem Frigui. A dataset for vehicle make and model recognition. 06 2015. (дата звернення: 15.04.2022).
7. М. Биглари, А. Сулеймані і Х. Хассанпур. Каскадна система на основі деталей для точної класифікації транспортних засобів. *Транзакції IEEE по інтелектуальним транспортним системам*, 19 (1): 273-283, січень 2018. (дата звернення: 15.04.2022).
8. Штучний інтелект (ШІ): що це таке і чому це важливо? *Інтегратор інновацій в Україні Everest* : стаття. URL: <https://www.everest.ua/ai-platform/analytics/shtuchnij-intelekt-ai-shho-ce-take-i-chomu-ce-v/> (дата звернення: 15.04.2022).

9. Томас Бьорклунд, Аттіліо Фіандротті, Мауро Аннарумма, Джанлука Франчіні та Енріко Мальї. Надійне розпізнавання номерних знаків з використанням нейронних мереж, навчених насинтетичних зображеннях. *Розпізнавання образів*, 93: 134-146, 2019 (дата звернення: 15.04.2022).
10. Queen Mary Bigger not necessarily better, when it comes to brains
Your source for the latest research news : наукова стаття.
URL: <https://www.sciencedaily.com/releases/2009/11/091117124009.htm>
(дата звернення: 5.05.2022).
11. J. Sanny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. (дата звернення: 12.05.2022).
12. Каспаров против Деєр Блує | Матч, змінив історію *Chess.com* – шахмат : стаття. URL: <https://www.chess.com/ru/article/view/kasparov-protiv-deep-blue-match-izmenivshii-istoriiu#kasparov-deep-blue-1997-rematch> (дата звернення: 20.05.2022).
13. Чжан чжан Чен, Фан Бай, Юньлу Сюй, Ган Чжен, Шилян Пу і Шуйген Чжоу. Фокусування уваги: на точному розпізнаванні тексту на природних зображеннях, сторінки 5086-5094 (дата звернення: 20.05.2022).
14. М. Чон, у. Лі, К. Юн і м. Пак. Система виявлення транспортних засобів на основі зору з урахуванням місця розташування виявлення. *Транзакції IEEE по інтелектуальним Транспортним системам*, 13 (3):1243-1252, 2012 (дата звернення: 15.04.2022).
15. H. Cho, M. Sung, and B. Jun. Canny text detector: Fast and robust scene text localization algorithm. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (дата звернення: 15.04.2022).
16. M Cormier, LW Sommer, and M Teutsch. Low resolution vehicle re-identification based on appearance features for wide area motion imagery. in *2016 IEEE winter applications of computer vision workshops (wacv)*. IEEE, 1:3, 2016_ (дата звернення: 20.04.2022).

17. Predicting coronary artery disease using different artificial neural network models / Çolak, M Cengiz, Çolak, Cemil, Kocatürk, Hasan, Sagiroglu, Seref, Barutçu, Irfan *ProQuest* : наукова стаття. URL: <https://search.proquest.com/openview/5ff9810dabea172db0d8a71cf5c06c77/1?pq-origsite=gscholar&cbl=106052> (дата звернення: 24.04.2022).
18. Coronary heart disease diagnosis by artificial neural networks including genetic polymorphisms and clinical parameters / Atkov O.Yu., Gorokhova S.G., Sboev A.G., Generozov E.V., Muraseyeva E.V, Moroshkina S.Y., Cherniy N.N. *US National Lib of Medicine* : наукова стаття. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6489351> (дата звернення: 28.04.2022).
19. IBM SPSS Statistics: Features and modules *IBM* : веб-сайт. URL: <https://www.ibm.com/products/spss-statistics/details> (дата звернення: 28.04.2022).
20. The R Language for Statistical Computing *The R Project* : веб-сайт. URL: <https://www.r-project.org/> (дата звернення: 28.04.2022).
21. Н. Далал і б. Тріггс. Гістограми орієнтованих градієнтів для виявлення людини. У 2005 році конференція *IEEE Computer Society з комп'ютерного зору та моделювання. Визнання (CVPR'05)*, том 1, сторінки 886-893, том 1, 2005. (дата звернення: 5.05.2022).
22. H.S. Niranjana Murthy, Dr.M. Meenaksh ANN Model to Predict Coronary Heart Disease Based on Risk Factors *Study documents, essay examples, research ppers*: наукова стаття. URL: <https://studyres.com/doc/6101474/ann-model-to-predict-coronary-heart-disease-based-on-risk> (дата звернення: 5.05.2022).
23. Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm / Zeinab Arabasadi, Roohallah Aliza-

- dehsani, Mohamad Roshanzamir, Hossein Moosaei, Ali Asghar Yarifard *Semantic Scholar | AI-Powered Research Tool* : наукова стаття. URL: <https://www.semanticscholar.org/paper/Computer-aided-decision-making-for-heart-disease-Arabasadi-Alizadehsani/53c5972ee-c92b21f213f2c1d6c2b6e8f23882700> (дата звернення: 5.05.2022).
24. Abdullah Caliskan, Mehmet Emin Yuksel Classification of coronary artery disease data sets by using a deep neural network *ResearchGate* : наукова стаття. URL: https://www.researchgate.net/publication/320862416_Classification_of_coronary_artery_disease_data_sets_by_using_a_deep_neural_network (дата звернення: 5.05.2022).
25. Б. Епштейн, Е.Офек і Ю. Векслер. Виявлення тексту в природних сценах за допомогою перетворення ширини штриха. У 2010 році конференція *IEEE Computer Society з комп'ютерного зору та розпізнавання образів*, сторінки 2963-2970, 2010. (дата звернення: 12.05.2022).
26. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПН 3.3.2.007-98. – К.: Постанова Головного державного санітарного лікаря України, 1998. - № 7.
27. Санітарні норми мікроклімату виробничих приміщень : ДСН 3.3.6.042-99. – К., 2000.- С.16
28. Природне і штучне освітлення : ДБН В.2.5-28:2006. – К. : Міністерство будівництва, архітектури та житлово-комунального господарства України, 2006. – 68 с. – (Національні стандарти України).
29. Санітарні норми виробничого шуму, ультразвуку та інфразвуку. Державні санітарні норми. ДСН 3.3.6.037-99. – К., 1999.

ДОДАТОК А

Код системи розпізнавання транспортних засобів

```
using Accord.Video.FFMPEG;
using Accord.Video.DirectShow;
using System;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;
using Accord.Imaging;
using Accord.Imaging.Filters;
using System.Collections.Generic;
using Accord.Vision.Tracking;

namespace Bootcamp.CompVis.Traffic
{
    /// <summary>
    /// The TrackerType holds data to track a single car.
    /// </summary>
    struct TrackerType
    {
        /// <summary>
        /// The camshift tracker tracking the car.
        /// </summary>
        public Camshift Tracker;

        /// <summary>
        /// The frame index when the car was first discovered.
        /// </summary>
    }
}
```

```
public int StartIndex;

/// <summary>
/// The index number of the car (1...N)
/// </summary>
public int CarNumber;

public TrackerType(Rectangle rect, int index, int carNumber)
{
    Tracker = new Camshift(rect);
    Tracker.Smooth = true;
    Tracker.Conservative = false;
    StartIndex = index;
    CarNumber = carNumber;
}
}

/// <summary>
/// The main application form.
/// </summary>
public partial class MainForm : Form
{
    // switch between camera and video
    private const bool USE_CAMERA = false;

    // the previous video frame
    private Bitmap previousFrame = null;

    // the index of the current frame
```

```

private int frameIndex = 0;

// the index of the last detected car
private int carIndex = 0;

// the list of camshift trackers following cars in the frame
private List<TrackerType> trackers = new List<TrackerType>();

/// <summary>
/// Initialize MainForm.
/// </summary>
public MainForm()
{
    InitializeComponent();
}

/// <summary>
/// Set the video player to read from the built-in camera.
/// </summary>
private void SetCamera()
{
    // use the first video input device
    var deviceName = (from d in new
FilterInfoCollection(FilterCategory.VideoInputDevice)
    select d).FirstOrDefault();
    var captureDevice = new VideoCaptureDevice(deviceName.MonikerString);

    // switch to 1280x720 resolution
    captureDevice.VideoResolution = (from r in captureDevice.VideoCapabilities

```

```
where r.FrameSize.Width == 1280
select r).First();
```

```
videoPlayer.VideoSource = captureDevice;
}

/// <summary>
/// Set the video player to read a video file from disk.
/// </summary>
/// <param name="fileName"></param>
private void SetVideo(string fileName)
{
    var source = new Accord.Video.FFMPEG.VideoFileSource(fileName);
    videoPlayer.VideoSource = source;
}

/// <summary>
/// Draw a label on the specified car.
/// </summary>
/// <param name="frame">The current video frame.</param>
/// <param name="rect">The bounding box around the car.</param>
/// <param name="index">The index number of the car.</param>
private void DrawCarLabel(Bitmap frame, Rectangle rect, int index)
{
    using (Graphics g = Graphics.FromImage(frame))
    {
        string name = $"Car {index}";
        Font fnt = new Font("Verdana", 20, GraphicsUnit.Pixel);
        Brush brs = new SolidBrush(Color.Black);
```

```
var stringSize = g.MeasureString(name, fnt);
var center = rect.Center();
g.FillRectangle(new SolidBrush(Color.Yellow), center.X, center.Y,
stringSize.Width, stringSize.Height);
g.DrawString(name, fnt, brs, center.X, center.Y);
}
}

/// <summary>
/// Called when MainForm loads.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void MainForm_Load(object sender, EventArgs e)
{
    if (USE_CAMERA)
        SetCamera();
    else
        SetVideo("./input.mp4");

    // start the player
    videoPlayer.Start();

    // start the timer
    timer.Start();
}

/// <summary>
/// Called when MainForm is closed.
```



```

/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void MainForm_FormClosed(object sender, FormClosedEventArgs e)
{
    videoPlayer.Stop();
}

/// <summary>
/// Called when videoPlayer receives a new frame.
/// </summary>
/// <param name="sender"></param>
/// <param name="image"></param>
private void videoPlayer_NewFrame(object sender, ref Bitmap frame)
{
    if (previousFrame != null)
    {
        // find the thresholded euclidian difference between two subsequent frames
        ThresholdedEuclideanDifference threshold = new
ThresholdedEuclideanDifference(40);
        threshold.OverlayImage = previousFrame;
        var difference = threshold.Apply(frame);

        // only keep big blobs
        var filter = new BlobsFiltering();
        filter.CoupledSizeFiltering = true;
        filter.MinHeight = 50;
        filter.MinWidth = 50;
        filter.ApplyInPlace(difference);
    }
}

```

```
// dilate remaining blobs
var dilate = new BinaryDilation3x3();
dilate.ApplyInPlace(difference);
dilate.ApplyInPlace(difference);
dilate.ApplyInPlace(difference);
dilate.ApplyInPlace(difference);

// put this image in the thresholded picturebox
thresholdedBox.Image = difference.Clone() as Bitmap;

// use this as a mask for the current frame
var mask = new ApplyMask(difference);
var maskedFrame = mask.Apply(frame);

// put this image in the masked picturebox
maskedBox.Image = maskedFrame.Clone() as Bitmap;

// now find all moving blobs
if (frameIndex % 10 == 0)
{
    var counter = new BlobCounter();
    counter.ProcessImage(difference);

    // only keep blobs that:
    // - do not overlap with known cars
    // - do not overlap with other blobs
    // - have crossed the middle of the frame
    // - are at least 100 pixels tall
```

```

var blobs = counter.GetObjectsRectangles();
var newBlobs = from r in counter.GetObjectsRectangles()
                where !trackers.Any(t =>
t.Tracker.TrackingObject.Rectangle.IntersectsWith(r))
                && !blobs.Any(b => b.IntersectsWith(r) && b != r)
                && r.Top >= 240 && r.Bottom <= 480
                && r.Height >= 100
                select r;

// set up new camshift trackers for each detected blob
foreach (var rect in newBlobs)
{
    trackers.Add(new TrackerType(rect, frameIndex, ++carIndex));
}
}

// now kill all car trackers that have expanded by too much
trackers.RemoveAll(t => t.Tracker.TrackingObject.Rectangle.Height >
360);

// and kill all trackers that have lived for 30 frames
trackers.RemoveAll(t => frameIndex - t.StartIndex > 30);

// let all remaining trackers process the current frame
var img = UnmanagedImage.FromManagedImage(maskedFrame);
trackers
    .ForEach(t => t.Tracker.ProcessFrame(img));

// remember this frame for next iteration

```

```

previousFrame.Dispose();
previousFrame = frame.Clone() as Bitmap;

// draw labels on all tracked cars
var outputFrame = frame.Clone() as Bitmap;
trackers
    .FindAll(t => !t.Tracker.TrackingObject.IsEmpty)
    .ForEach(t => DrawCarLabel(outputFrame,
t.Tracker.TrackingObject.Rectangle, t.CarNumber));

// return the processed frame to the video
frame = outputFrame;
}

// or else just remember this frame for next iteration
else
    previousFrame = frame.Clone() as Bitmap;

frameIndex++;
}

private void timer_Tick(object sender, EventArgs e)
{
    carLabel.Text = $"Number of cars: {carIndex}";
}
}
}

```