

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДІВ РОЗПІЗНАВАННЯ

ДОРОЖНІХ ЗНАКІВ УКРАЇНИ

(тема)

Виконав:
студент 2 курсу, групи ІНФМ-21-1

Темчур К.О.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Темчур Карині Олексіївні
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження методів розпізнавання дорожніх знаків України

затверджена наказом по університету від 9 листопада 2022 року № 1469Ст

2. Термін подання студентом роботи до екзаменаційної комісії 21 листопада 2022 р.3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, відкрита програмна бібліотека для машинного навчання TensorFlow, відкрита нейромережна бібліотека Keras, відкриті набори даних дорожніх знаків Європи для навчання нейронної мережі, відкрита модель YOLOv3, мова Python для створення інтерфейсу користувача.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючих методів розпізнавання дорожніх знаків.2. Аналіз моделей згорткових нейронних мереж та моделі YOLO.3. Моделювання нейронної мережі та підготовка до її навчання.4. Програмна реалізація та аналіз результатів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми розпізнавання дорожніх знаків, мета роботи, постановка задачі, етапи виконання роботи, результати тестування, перспективи подальшої роботи.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	09.11.2022	
2	Аналіз завдання, підбір літератури	09.11.22-10.11.22	
3	Аналіз літератури з досліджуваної проблеми	11.11.21	
4	Аналіз методів розпізнавання дорожніх знаків	12.11.21	
5	Розробка методів розпізнавання	13.11.21	
6	Програмна реалізація	13.11.21-15.11.21	
7	Оформлення пояснювальної записки	15.11.21-18.11.21	
8	Перевірка на плагіат	21.11.2021	
9	Рецензування	22.11.2021	
10	Підготовка презентації та доповіді	22.11.2021	
11	Занесення роботи в електронний архів	29.11.2021	
12	Попередній захист кваліфікаційної роботи	30.11.2021	

Дата видачі завдання 9 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Творошенко І.С.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 97 с., 4 табл., 41 рис., 1 дод., 81 джерело.

КОМП'ЮТЕРНИЙ ЗІР, РОЗПІЗНАВАННЯ ТА КЛАСИФІКАЦІЯ ОБРАЗІВ, ГЛИБОКЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, TRANSFER LEARNING, YOLO.

Об'єктом дослідження є процес розроблення застосунку для розпізнавання та класифікації дорожніх знаків України.

Метою дослідження є розроблення програмного застосунку із застосуванням методів глибокого машинного навчання, спрямованого на розпізнавання та класифікацію дорожніх знаків.

Проведено дослідження класичних методів розпізнавання та класифікації образів та методів на основі нейронних мереж. Досліджено можливості нейронних моделей YOLO, особливості роботи CNN. Проведено дослідження методів на основі згорткових нейронних мереж та моделі YOLO. У результаті роботи отримано програмний застосунок для розпізнавання та класифікації дорожніх знаків з використанням обох моделей.

Результати роботи апробовано у вигляді 2 тез доповідей під час XXX Міжнародної науково-практичної конференції MicroCAD-2022, XXXVI Міжнародної науково-практичної «The main prospects for the development of science in modern life».

COMPUTER VISION, IMAGE RECOGNITION AND CLASSIFICATION, DEEP LEARNING, NEURAL NETWORKS, CONVOLUTIONAL NEURAL NETWORKS, YOLO, TRANSFER LEARNING.

The object of the research is the process of developing an application for recognizing and classifying Ukrainian road signs.

The purpose of the research is to develop a software application using deep machine learning methods aimed at recognizing and classifying road signs.

Classical methods of image recognition and classification and methods based on neural networks have been studied. The capabilities of the YOLO models, the features of CNN work, were studied. A study of methods based on CNN and the YOLO model was conducted. As a result of the work, a software application was obtained for recognition and classification of road signs using both models.

The results of the work were tested in the form of 2 theses during the XXX International scientific-practical conference MicroCAD-2022 and XXXVI International Scientific and Practical Conference «The main prospects for the development of science in modern life».

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Аналіз існуючих методів розпізнавання дорожніх знаків	9
1.1 Аналіз сучасних систем розпізнавання дорожніх знаків.....	9
1.2 Класифікація та аналіз існуючих методів розпізнавання дорожніх знаків	15
1.3 Особливості дорожніх знаків України.....	24
1.4 Аналіз літературних джерел щодо апробації результатів розпізнавання дорожніх знаків	27
1.5 Постановка задачі дослідження.....	39
2 Особливості розроблення методів розпізнавання дорожніх знаків.....	41
2.1 Механізм оброблення зображення дорожніх знаків	41
2.2 Принципи розпізнавання складових дорожніх знаків	46
2.3 Методика розпізнавання дорожніх знаків.....	49
3 Реалізація та дослідження методів розпізнавання дорожніх знаків	69
3.1 Вибір інструментального засобу для реалізації методів розпізнавання дорожніх знаків	69
3.2 Етапи програмної реалізації методів розпізнавання дорожніх знаків	72
3.3 Тестування розроблених застосунків та аналіз результатів.....	81
3.4 Перспективи подальшої роботи	85
Висновки	86
Перелік джерел посилання	87
Додаток А Список класів дорожніх знаків для навчання моделі.....	96

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект

ТЗ – транспортний засіб

TSR – Traffic Sign Recognition (розпізнавання дорожніх знаків)

FPS – Frames Per Seconds (частота кадрів)

ПЗ – програмний застосунок

HOG – Histogram of Oriented Gradients (гістограма напрямлених градієнтів)

SIFT – Scale-Invariant Feature Transform (масштабонезалежне перетворення ознак)

SVM – Support Vector Machine (метод опорних векторів)

CNN – Convolutional Neural Network (згорткова нейронна мережа)

YOLO – You Only Look Once

SSD – Single Shot Multibox Detector

ДТП – дорожньо-транспортна пригода

ІМС – інтегральні мікросхеми

ІС – інформаційна система

HSV – Hue, Saturation, Value (тон, насиченість, значення)

SURF – Speeded Up Robust Features (прискорені надійні функції)

PCA – Principal Component Analysis (метод головних компонент)

LDA – Linear Discriminant Analysis (лінійний дискримінантний аналіз)

BoF – Bag of Freebies (пакет безкоштовних подарунків)

BoS – Bag of Specials (пакет спеціальних пропозицій)

ВСТУП

Вміння людини за допомогою зору знаходити та розпізнавати образи в житті, на фотографіях або навіть в абстракціях – дивовижне вміння, що формувалося тисячі років, та приголомшує ефективністю та, водночас, складністю процесів, що ховаються за цим вмінням.

Багато років вчені, що вивчають та вдосконалюють роботу обчислювальних систем таких, як комп'ютер, намагаються надати їм можливість перейняти деякі вміння, що притаманні людині. І якщо сам по собі комп'ютер не може «бачити» та розрізнявати об'єкти, то технології комп'ютерного зору вивчають створення штучних систем, що вже можуть вдало імітувати можливості зору людини.

Комп'ютерний зір – це досить молода галузь комп'ютерних наук, що вивчає інтелектуальну обробку та аналіз візуальних даних, таких як зображення та відео.

До типових задач комп'ютерного зору відносять пошук образів на зображенні, розпізнавання образів, їх класифікація, відновлення візуальних матеріалів або їх генерація та інші.

Дорожні знаки є невід'ємною частиною дорожнього середовища. Вони призначені для регулювання потоку транспортних засобів, надавати конкретну інформацію учасникам руху, або застерігають від несподіваних дорожніх обставин. Сприйняття та швидке тлумачення дорожніх знаків має вирішальне значення для безпека водія.

Комунальні служби, відповідальні за утримання інфраструктури дорожнього руху, встановлюють знаки на стовпах на узбіччях, над смугами автомагістралі та в інших місцях таким чином, щоб їх було легко помітити, не відволікаючи увагу водія від керування транспортним засобом.

Піктограми знаків також розроблені та стандартизовані відповідно до правила максимальної простоти та відмінності.

Однак, певні обставини, наприклад, сильні візуальні перешкоди, несприятливе освітлення або погані погодні умови, можуть значно погіршити сприйняття дорожніх знаків.

Відомо, що суто фізіологічні фактори, такі як хвилювання, роздратування або втома, ще більше знижують зорову концентрацію людини і, отже, можуть поставити під загрозу життя водія, особливо під час їзди на високих швидкостях.

З наведених вище причин автоматизація процесу виявлення, розпізнавання та класифікації дорожніх знаків була визнана одним з найважливіших напрямів, як тільки обробка відео та фото стала доступною на комп'ютері.

Сьогодні це вважається критичним завданням у сучасних системах підтримки водія, але у даній сфері все ще залишається великий простір для вдосконалення підходів вирішення проблеми.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

1.1 Аналіз сучасних систем розпізнавання дорожніх знаків

Останніми роками, у зв'язку з розповсюдженням штучного інтелекту (ШІ), система керування автомобілем оновила попередній режим водіння. Отримуючи інформацію про стан доріг у реальному часі, система миттєво нагадує водіям про необхідність виконувати точні операції, тим самим запобігаючи автомобільним аваріям через втому водія. Окрім допоміжних систем водіння, розробка автономних транспортних засобів також вимагає швидкого та точного виявлення дорожніх знаків із цифрових зображень.

Ще з моменту появи перших бортових комп'ютерів вчені та розробники почали працювати над задачею автоматичного розпізнавання дорожніх перешкод, розмітки та дорожніх знаків, оскільки дана сфера є перспективною у контексті розвитку систем автоматичного керування наземними транспортних засобів (ТЗ). Протягом багатьох років велись розробки, було створено велику кількість методів і комп'ютерних програм, для того, аби все ж таки вирішити проблему розпізнавання об'єктів у реальному часі [1].

На сьогоднішній день у цій сфері ведеться активний розвиток, щороку великі компанії створюють нові та оновлюють діючі системи з розпізнаванням об'єктів на відео. В основному, такі системи направлені на розпізнавання великих перешкод, таких як інші транспортні засоби, а також на розпізнавання дорожніх знаків [2].

Розпізнавання дорожніх знаків (Traffic Sign Recognition або TSR) призначене для виявлення розташування дорожніх знаків із цифрових зображень або відеокадрів за певною класифікацією. Методи TSR в основному використовують візуальну інформацію, таку як форма та колір дорожніх знаків. Однак, звичайні алгоритми TSR стикаються з недоліками в тестах у реальному часі, наприклад, їх легко обмежити умовами водіння, включаючи

освітлення, кут камери, перешкоди, швидкість руху тощо. Також дуже важко досягти багатоцільового виявлення, легко пропустити візуальні об'єкти через повільне розпізнавання.

Завдяки постійному вдосконаленню комп'ютерного обладнання обмеження штучних нейронних мереж було значно пом'якшено, що призвело машинне навчання до золотого періоду розвитку. Глибоке навчання є різновидом методів машинного навчання.

Модель глибокої нейронної мережі моделює нейронну структуру нашого людського мозку під час обробки інформації. Використання цієї моделі нейронної мережі для вилучення ефективних характеристик із зображення дороги є набагато кращим, ніж звичайні алгоритми TSR, що має потенціал для покращення надійності та узагальнення алгоритмів.

Результати дослідження в TSR не тільки дозволяють уникнути дорожньо-транспортних пригод і захистити водіїв, але й допомагають ефективно й точно перевіряти дорожні знаки на дорогах, що зменшує непотрібну робочу силу та ресурси. Крім того, також забезпечується технічна підтримка безпілотного та допоміжного водіння. Тому дослідницька робота, що заснована на глибокому навчанні, має величезне значення і неоціненна для повсякденного життя людей.

Розробки в цій галузі почалися близько століття тому з відмови від оптичного розпізнавання тексту. У 1950 році після аналізу проблеми перетворення друкованих повідомлень на машинну мову для комп'ютерної обробки Девід Х. Шепард, криптоаналітик Агентства безпеки збройних сил США, створив машину, яка вирішила цю проблему.

Отримавши патент США, Шепард заснував компанію з інтелектуальних машин, яка незабаром виготовила перші в світі комерційні системи оптичного розпізнавання символів.

З кожним десятиліттям технологія розпізнавання образів знаходила все більше застосувань, а словниковий запас розпізнаних слів поступово збільшувався.

Справжнім проривом у технології розпізнавання образів стало машинне навчання. З появою глибокого навчання кількість помилок розпізнавання різко зменшилася.

У 2006 році під час проведення змагань Face Recognition Grand Challenge була зроблена оцінка ефективності розроблених на той час алгоритмів розпізнавання облич. У тестах використовувалися зображення високої роздільної здатності, 3D-сканування обличчя та зображення райдужної оболонки ока.

Результати показали, що нові алгоритми в 10 разів точніші, ніж алгоритми розпізнавання облич 2002 року, і в 100 разів точніші, ніж алгоритми 1995 року.

Деякі алгоритми перевершили людей у розпізнаванні облич і змогли показати коректну відповідь навіть у випадках при розпізнаванні однойцевих близнюків [3]. Наприклад, заснований у 2013 році застосунок Lookserу дозволяє змінювати зовнішній вигляд користувачів [4].

У транспортній сфері дані про появу систем розпізнавання образів не є відкритими. Можна зауважити, що перші згадки про подібну технологію датовані 2005 роком [5].

На сьогоднішній день системи розпізнавання дорожніх перешкод застосовуються у досить багатьох моделях автомобілів таких виробників, як Tesla, BMW, Mercedes, Ford, Nissan. Це – функції розпізнавання дорожніх знаків, інших автомобілів та пішоходів, системи зберігання полоси тощо. Вони, здебільшого, спрямовані на допомогу водієві у процесі керування, хоча деякі найсучасніші розробки уже здатні повністю взяти на себе функцію управління автомобілем.

Розробка системи розпізнавання знаків дорожнього руху є складним завданням. Є дуже багато факторів, які можуть стати причиною низької точності та повноти системи розпізнавання знаків дорожнього руху.

У загальному проблеми будь-якої системи розпізнавання об'єктів можна розділити на такі групи:

– недостатня освітленість об'єкту є одною з найбільш основних проблем в розробці систем розпізнавання об'єктів, в тому числі і системи розпізнавання елементів дорожнього руху. Різні дорожні знаки мають різні кольори для того, щоб дорожній знак був більш помітний. Більшість існуючих алгоритмів розпізнавання елементів дорожнього руху використовують інформацію про колір об'єкту для ідентифікації знаків.

Інформація про колір об'єкту є дуже чутлива до умов освітлення та терміну експлуатації знаку, відомо, що на сонці дорожні знаки мають властивість висвітлюватись (рис. 1.1);



Рисунок 1.1 – Справа приклад висвітленого сонцем знаку на трасі E40 (Київ-Чоп), для порівняння, по ліву частину дороги – новий знак

– деформовані або пошкоджені знаки є проблемою для сучасних алгоритмів розпізнавання елементів дорожнього руху, оскільки в алгоритмах використовується інформація про форму заданого об'єкту, що може призвести до неправильної класифікації або ж взагалі до того, що знак не буде розпізнано (рис. 1.2). Також бувають випадки, коли знак частково закритий іншими об'єктами. Найпоширенішими перешкодами для чіткого визначення дорожнього знаку є, наприклад, елементи природного ландшафту. Здебільшого дерева або кущі.



Рисунок 1.2 – Приклад деформованого дорожнього знаку

– також важливою проблемою є *оптичні ефекти, розмитість та висвітленість*, що мають місце в поганих погодних умовах, що можуть зменшувати точність та повноту алгоритму розпізнавання та ідентифікації. Цей ефект також має місце в сучасних системах, оскільки для отримання зображення використовують камери, які під час руху автомобіля на високій швидкості можуть давати розмиті фотографії. Для зменшення цієї проблеми використовують камери з великою характеристикою FPS (Frames per seconds). Наприклад, камери з можливістю зйомки 240 кадрів у секунду;

– *встановлення зони розміщення знаку на картинці*. Деякі конструкції, які є дуже схожі на елементи дорожнього руху можуть спричинювати проблеми для сучасних алгоритмів при розпізнаванні дорожніх знаків. Наприклад, рекламні щити схожої форми можуть бути розпізнані системою як дорожні знаки;

– *погана видимість*. Тінь від інших машин призводить до спотворення картинки при розпізнаванні, наприклад, лінії руху. Такі природні явища як

дощ, сніг та туман також є серйозними проблемами для сучасних алгоритмів розпізнавання елементів дорожнього руху.

Світовим лідером у створенні програмного забезпечення (ПЗ) для розпізнавання дорожніх об'єктів є Tesla. На рисунку 1.3 [6] зображено рекламу з описанням сенсорів, що беруть участь у автоматичному пілотуванні автомобіля.

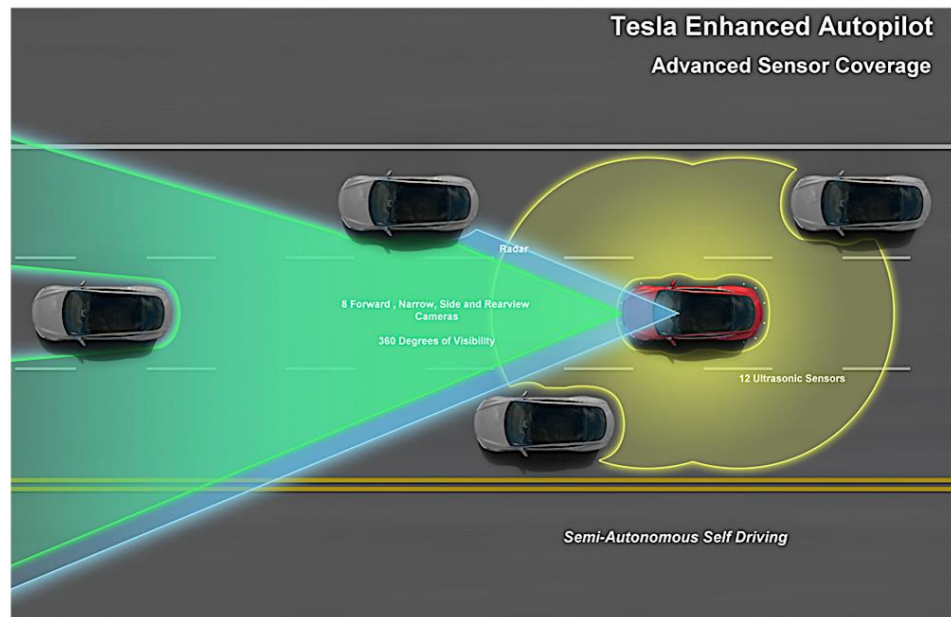


Рисунок 1.3 – Рекламний проспекти автопілоту компанії Tesla

Їх ПЗ включає у себе розпізнавання транспортних засобів, дорожніх знаків, використання сторонніх даних про ситуації на дорогах, аналіз розмітки та дорожнього покриття. Ці технології роблять можливою автоматичну їзду у будь-якій країні світу, однак проблематика аналізу дорожнього покриття навіть у світового лідера галузі є актуальною [6].

BMW 5 – продукт компанії BMW, що дозволяє використання автопілоту та була протестована на німецьких дорогах.

BMW Connected Drive, Nissan Around View Monitor та інші подібні системи дають можливість кругового огляду, попередження зіткнення та розпізнавання дорожніх знаків, але функція аналізу покриття не реалізована.

1.2 Класифікація та аналіз існуючих методів розпізнавання дорожніх знаків

Розпізнавання дорожніх знаків на зображеннях можна логічно розділити на два основних етапи:

- виявлення дорожнього знака на зображенні;
- розпізнавання дорожніх знаків.

Основний принцип цих методів полягає в порівнянні кожної області зображення з певним шаблоном, тим самим визначаючи взаємну кореляцію. Шаблони можуть бути задані вручну або визначені будь-якою функцією та можуть відповідати всьому об'єкту або окремим компонентам об'єкта.

Якщо коефіцієнт кореляції перевищує заданий поріг, область дослідження зображення позначається як така, що містить зображення дорожнього знака. Існують також різні вдосконалення цього підходу, які використовують кілька шаблонів, що відповідають окремим компонентам об'єктів, які є областями у вигляді ліній.

Робота цих алгоритмів складалася з кількох етапів, кожен з яких визначав достовірність знайдених ділянок. Коли область зображення збігається з одним із шаблонів, вона позначається як область інтересу для подальшого аналізу.

В наслідок цього підвищується ступінь схожості областей зображення із заданим шаблоном. Далі виявлені ділянки порівнювалися з іншими шаблонами, що дозволяло визначити наявність потрібного об'єкта. Данні методи в основному використовувалися в ранніх роботах з класифікації та виявлення об'єктів на зображеннях. Це були перші спроби формування особливостей образу предметів [7].

Сучасні модифікації шаблонних методів демонструють високу точність розпізнавання. Недоліком цих методів є низька швидкість роботи і висока чутливість до спотворення об'єктів на зображеннях.

Для реалізації кожного з цих етапів використовуються індивідуальні методи та алгоритми.

Багато існуючих методів пасивної безпеки автомобілів є застарілими, а алгоритми, які в них використовуються – неефективними. Існуючі системи пасивної безпеки є дуже дорогими, а їхні алгоритми запатентовані та у закритому доступі.

TSR завжди була актуальною темою дослідження в останні роки. З цією метою TSR досліджується для виявлення області дорожніх знаків і зони без дорожніх знаків у складній сцені зображень, TSR має виділити специфічні особливості, представлені шаблонами дорожніх знаків. Існуючі методи TSR в основному згруповані у дві категорії: одна базується на традиційних методах, інша пов'язана з методами глибокого навчання.

Основні кроки методів TSR, заснованих на кольорі та формі даного зображення, полягають у вилученні візуальної інформації, що міститься в кандидатській області, захопленні та сегментуванні дорожніх знаків на зображенні та правильному позначенні знаків за допомогою шаблонної класифікації.

Хоча TSR вимагає інформації про колір і форму, яка використовується для підвищення точності розпізнавання. Проблеми зміни освітленості або згасання кольору дорожніх знаків, а також деформація та закривання дорожніх знаків досі залишаються невирішеною проблемою.

Звичайні методи машинного навчання зазвичай вибирають певні візуальні характеристики та використовують їх для класифікації класів дорожніх знаків. Специфічні функції включають функції, подібні до Haar-like features, функції HOG або The Histogram of Oriented Gradients, функції SIFT (Scale-Invariant Feature Transform) і так далі.

Звичайні методи TSR базуються на зіставленні шаблонів, які потребують вилучення та використання незмінних і схожих візуальних характеристик дорожніх знаків, алгоритми зіставлення запускаються для класифікації шаблонів. Репрезентація ознак цих методів повинна бути добре визначена, що

є складною проблемою для точного опису візуальних особливостей через варіації дорожніх знаків.

Нейронні мережі, байєсівський класифікатор, випадковий ліс і опорна векторна машина (SVM або Support Vector Machine) використовуються як класифікатори. Однак продуктивність звичайних методів машинного навчання залежить від зазначених функцій, вони схильні пропускати ключові функції. Крім того, для різних класифікаторів необхідна відповідна інформація про опис ознак. Отже, традиційні методи машинного навчання мають обмеження, їхня продуктивність у реальному часі не є відносно порівняльною.

Глибоке навчання використовує багат шарову нейронну мережу для автоматичного вилучення та вивчення особливостей візуальних об'єктів, що має переваги для обробки зображень [8].

Моделі CNN (Convolutional Neural Network або Згортова Нейронна Мережа) є одними з найпопулярніших підходів глибокого навчання для TSR. Алгоритми TSR базуються на регіональних пропозиціях, також відомих як двоетапний алгоритм виявлення. Основною ідеєю є вибірковий пошук [9], його перевагами є висока продуктивність виявлення та позиціонування, але вартість полягає у великій кількості обчислень і високому продуктивне апаратне забезпечення для обчислень. Моделі CNN інкапсулюють R-CNN, Fast R-CNN і Faster R-CNN. Швидший R-CNN поєднує регресію обмежувальних рамок і класифікацію об'єктів, використовує наскрізні методи для виявлення візуальних об'єктів, які не тільки підвищують точність виявлення об'єктів, але й підвищують швидкість розпізнавання об'єктів.

У китайському дослідженні [10] для вхідного зображення було застосовано керовану фільтрацію зображення для видалення артефактів зображення, таких як туман та імла.

Оброблене зображення імпортується в запропоновані мережі для навчання моделі.

Алгоритми TSR, засновані на регресії, також відомі, як одноетапний алгоритм виявлення. Цей тип алгоритмів TSR усуває ідею мережі регіональних пропозицій і безпосередньо виконує регресію та класифікацію в мережі. You Only Look Once (YOLO) і Single Shot MultiBox Detector (SSD) належать до категорії одноступеневих.

Візуальне виявлення об'єктів складається з двох завдань: класифікації та позиціонування. До появи YOLO ці два завдання відрізнялися візуальним виявленням об'єктів. У моделях YOLO виявлення об'єкта просто перетворюється на задачу регресії. Крім того, YOLO слідує наскрізній структурі нейронних мереж для візуального виявлення об'єктів, яка отримує координати передбачуваних обмежувальних рамок, впевненість цілі та ймовірність класу [11].

У 2020 році було випущено три версії YOLO, тобто YOLOv4, YOLOv5 та PP-YOLO. Коли було випущено YOLOv4, її вважали швидшою та точнішою моделлю виявлення об'єктів у реальному часі, яка успадкувала Darknet і отримала чітку середню точність (AP) на основі набору даних Microsoft COCO, а також досягла високої швидкості виявлення на основі Tesla V100. Порівняно з YOLOv3, AP і FPS (тобто кількість кадрів за секунду або частота кадрів відео) були значно покращені.

Інформаційна система розпізнавання дорожніх знаків в основному складається з наступних апаратних рішень і програмних модулів:

- фотокамера для зйомки зображень;
- апаратний модуль захоплення;
- модуль виявлення;
- модуль класифікації;
- бази даних.

Зображення з камери надходить на вхід системи. Потім за допомогою алгоритму визначається положення дорожнього знака. Після цього знак буде розпізнаний програмою розпізнавання. База даних може мати різну схему в залежності від поставлених перед системою завдань.

Система Opel Eye – це система розпізнавання дорожніх знаків. Система Opel Eye працює за таким принципом: відеокамера високої роздільної здатності з широким кутом огляду розташована між лобовим склом і дзеркалом заднього виду. Ця камера постійно отримує інформацію про те, що знаходиться попереду автомобіля.

Відеокамера може передавати до тридцяти кадрів в секунду на два процесори обробки сигналів. База даних дорожніх знаків зберігається в пам'яті. Якщо зображення, передане камерою, відповідає символу, що зберігається в базі даних, система відобразить відповідний символ на панелі приладів. Наприклад, це може бути знак заборони обгону, знак обмеження швидкості тощо. На рисунку 1.4 показано приклад системи Opel Eye.



Рисунок 1.4 – Приклад роботи Opel Eye

Система Opel Eye допомагає водієві: якщо з якоїсь причини водій не помітив обмеження швидкості (або будь-який інший дорожній знак), система Opel Eye не пропускає цей знак і відображає його на панелі приладів. Зараз багато автомобілів використовують навігаційну систему, яка вже містить повідомлення про обмеження швидкості на різних ділянках маршруту.

Система Opel Eye принципово відрізняється від інших навігаційних систем тим, що оцінює реальну ситуацію на дорозі в реальному часі. Також

варто відзначити, що система розпізнавання символів Opel Eye має такі доповнення, як FCA і LDW. FCA – це система попередження про лобове зіткнення. LDW – це система, яка контролює зміну смуги руху. Якщо існує ризик зіткнення з транспортним засобом, що їде попереду, або якщо пішохід переходить смугу руху, всередині автомобіля лунає звуковий сигнал. Разом із звуковим сигналом на приладовій панелі також відображається відповідне повідомлення. Точність розпізнавання символів різко падає в разі нахилу або бруду, загальна точність становить близько 70%.

Нові автомобілі S-, CL-, CLS- і E-Class оснащені новою системою Speed Limit Assist – системою, яка допомагає водієві не перевищувати обмеження швидкості. Він призначений для нагадування водієві про обмеження швидкості на ділянці дороги, яку він збирається проїхати. На рисунку 1.5 показаний приклад працюючого помічника обмеження швидкості.



Рисунок 1.5 – Приклад роботи Speed limit assist

Остання версія Speed Limit Assist здатна не тільки розпізнавати дорожні знаки, але й аналізувати, чи можна передати отриману інформацію вантажівкам.

Також система розпізнає знаки закінчення обмеження швидкості. При цьому, якщо через деякий час знак обмеження швидкості не буде знято, навігаційна система відобразить збережені в ньому дані для цієї ділянки дороги.

Система в режимі реального часу оцінює інформацію, отриману від камери на лобовому склі. Розташування знака обмеження швидкості – на узбіччі або вище – значення не має.

Завдяки доопрацюванню технології розпізнавання символів, що надходять з камери в систему Speed Limit Assist, аналіз інформації займає не більше секунди, а результат відразу ж передається на дисплей.

За словами німецьких експертів, система підтримки водія у дотриманні швидкісного режиму допоможе уникнути або принаймні зменшити кількість ДТП (дорожньо-транспортна пригода).

Статистика показує, що багато серйозних аварій на дорогах стаються через перевищення дозволеної швидкості.

Інформаційна система дорожніх знаків допомагає водієві запам'ятати дорожні знаки, які він проїхав. Одночасно відображає знак обмеження швидкості та заборону обгону. Також може показувати знаки «Дорога для автомобілів» і «Автобан».

При виявленні знака обмеження швидкості на шкалі швидкості ставиться червоний трикутник з обмеженням.

Символи скидаються, коли виявлено кінець символу обмеження тощо. Є помилкові тривоги.

Програмний комплекс може розпізнавати такі знаки:

- обмеження швидкості;
- обгін заборонено;
- закінчення всіх обмежень;
- автобан / кінець автобану;
- інформаційні щити.

На рисунку 1.6 демонструється робота системи.



Рисунок 1.6 – Приклад роботи системи Road sign information

Точність виявлення буде значно знижена, якщо знак буде нахилений, частково перекритий, забруднений чи повернутий.

У таблиці 1.1 наведено порівняльну характеристику існуючих систем розпізнавання.

Можна зробити висновок, що окремо реалізовані інформаційні системи розпізнавання дорожніх знаків зустрічаються дуже рідко, більшість ІМС (інтегральних мікросхем) не є окремо вбудованою функцією, а здебільшого входять до заводської комплектації автомобіля.

З цього випливає, що їх комерційна цінність занадто висока.

Усі встановлені системи демонструють невеликий відсоток розпізнавання дорожніх знаків.

Відсоток розпізнавання зменшується, коли дорожній знак знаходиться в не ідеальних умовах (нахил, часткове перекриття, забруднення знака), оскільки зображення зазнають спотворень дизайну.

Під час тестів надійно і точно розпізнаються лише чіткі символи з контрастним контуром.

Неможливо з'ясувати, на яких алгоритмах базуються ці ІС (інформаційні системи), оскільки ця інформація не є у вільному доступі.

Таблиця 1.1 – Порівняльна характеристика систем розпізнавання

Параметр	Opel Eye	Speed Limit Assist	Road Sign Information
Заявлена точність розпізнавання	90%	95%	96%
Освітленість знаку не менше	50	50	50
Розпізнавання знаків обмеження швидкості	Так	Так	Так
Розпізнавання інших заборонних знаків	Так	Ні	Так
Інша інформація	Початок роботи від швидкості руху > 60		
Реальна точність розпізнавання	75%	70%	75%

Розробка методу ідентифікації стійких до таких перетворень дорожніх знаків підвищить точність розпізнавання, що є актуальним при:

- похилому дорожньому знаку;
- маневруванні транспортним засобом в зоні розпізнавання;
- високій швидкості автомобіля.

1.3 Особливості дорожніх знаків України

Зараз в Україні дорожні знаки поділяються на вісім груп:

- попереджувальні знаки;
- знаки пріоритету;
- заборонні знаки;
- нормативні знаки;
- спеціальні знаки;
- інформаційні покажчики;
- знаки обслуговування;
- знаки додаткової інформації.

Розглянемо основні з них.

Заборонні дорожні знаки є основними знаками в Україні. Заборонений дорожній знак забороняє водієві виконувати будь-які дії, запропоновані знаком. Найчастіше аварійні ситуації виникають при ігноруванні цих ознак.

На рисунку 1.7 наведено приклад знаків заборони руху.



Рисунок 1.7 – Приклад заборонних дорожніх знаків

Прикладом можуть бути знаки обмеження швидкості, обгону, в'їзду, паркування та зупинки.

Як відомо, червоний колір відразу впадає в око, адже заборонні знаки мають круглу форму і яскраво-червоний колір, щоб водій міг їх бачити за будь-яких умов і погодних умов.

Попереджувальні дорожні знаки (рис. 1.8).

Основна функція цих знаків – попереджати водіїв про небезпеку та дорожньо-транспортні пригоди.

Зазвичай такі знаки встановлюють на небезпечних ділянках доріг. Основою знака в більшості випадків є трикутник з червоним контуром.



Рисунок 1.8 – Приклад попереджувальних дорожніх знаків

Наказові дорожні знаки використовуються для того, щоб показати водіям єдину дозволена дію: рухатися прямо або лише повертати ліворуч (але в цьому випадку розворот також дозволений).

Такі знаки дозволяють лише певним учасникам дорожнього руху виконувати певні дії.

На рисунку 1.9 показано приклад адміністративних символів.



Рисунок 1.9 – Приклад наказових дорожніх знаків

Знаки пріоритету встановлюють порядок проїзду перехресть, перехресть або вузьких ділянок дороги. *Попереджувальні знаки* можуть поєднувати елементи наказових і заборонних знаків.

Приклад спеціальних приписних знаків наведено на рисунку 1.10.



Рисунок 1.10 – Приклад знаків пріоритету та попереджувальних знаків

При цьому найважливішими з перших п'яти груп є заборонні знаки, такі як знаки обмеження швидкості або знаки заборони обгону. За статистикою ігнорування заборонних знаків є найпоширенішою причиною ДТП.

1.4 Аналіз літературних джерел щодо апробації результатів розпізнавання дорожніх знаків

Автоматичне виявлення та розпізнавання дорожніх знаків є важливою частиною вдосконаленої системи допомоги водієві. Дорожні символи мають кілька відмінних ознак, які можна використовувати для їх виявлення та ідентифікації. Вони розроблені в певних кольорах і формах, з текстом або символом у високому контрасті з фоном.

Оскільки дорожні знаки, як правило, орієнтовані вертикально та спрямовані до камери, кількість обертальних і геометричних спотворень обмежена.

Інформацію про дорожні символи, таку як форма та колір, можна використовувати для розміщення дорожніх символів у певні групи; однак є кілька факторів, які можуть перешкоджати ефективному виявленню та розпізнаванню дорожніх знаків. Ці фактори включають варіації в перспективі, варіації в освітленні (включаючи варіації, спричинені зміною рівня освітленості, сутінками, туманом і тінями), оклюзію знаків, розмитість руху та погіршення якості знаків через погоду.

Дорожні сцени також зазвичай дуже зашарашені та містять багато яскравих геометричних фігур, які можна легко класифікувати як дорожні знаки. Точність є ключовим фактором, тому що навіть один неправильно класифікований або непомічений знак може негативно вплинути на водія.

Опубліковано значну кількість робіт, присвячених розпізнаванню дорожніх знаків на основі ідеограм у реальних дорожніх сценах [11–15].

Найпоширеніший підхід, цілком розумно, складається з двох основних етапів: виявлення та розпізнавання. Етап виявлення ідентифікує цікаві регіони та здебільшого виконується за допомогою сегментації кольорів, а потім певної форми розпізнавання форми. Виявлені кандидати потім або ідентифікуються, або відхиляються на етапі розпізнавання за допомогою, наприклад, зіставлення шаблонів [16] або певної форми класифікатора, наприклад, SVM [2], [8], [17] або нейронні мережі [10], [18].

Більшість систем використовують інформацію про колір як метод сегментації зображення [9], [11], [12], [19], [20]. Ефективність визначення дорожніх знаків за кольором часто знижується в сценах із сильним освітленням, поганим освітленням або несприятливими погодними умовами, як-от туман. Колірні моделі, такі як значення відтінку – насиченості (HSV або Hue, Saturation, Value) [8], [12], [19], YUV [20] і CIECAM97 [21], були використані для спроби подолання цих проблем.

Наприклад, турецькі дослідники [22] виконали сегментацію, застосовуючи канали кольоровості U та V простору YUV, де U був позитивним, а V був негативним для червоних кольорів. Цю інформацію

використовували в поєднанні з каналом відтінків колірному простору HSV для сегментації червоних дорожніх знаків.

Китайські дослідники [23] застосували метод гістограми чотирьох дерев, щоб сегментувати зображення на основі значень відтінку та кольоровості колірної моделі CIECAM97.

Індійські дослідники [24] обмежили канал відтінків колірному простору HSV для сегментації червоних дорожніх знаків.

Навпаки, існує декілька підходів [7], [25], які повністю ігнорують інформацію про колір і замість цього використовують лише інформацію про форму із зображень у відтінках сірого.

Наприклад, у роботі [26] запропоновано систему, яка використовувала локальну радіальну симетрію для виділення цікавих точок на кожному зображенні та виявлення восьмикутних, квадратних і трикутних дорожніх знаків.

Деякі новітні методи, такі як [27] і [28], використовують функції HOG для вилучення ознак дорожніх знаків.

У роботі [29] розширили алгоритм HOG, щоб включити інформацію про колір, використовуючи простори кольорів CIELAB і YCbCr.

У дослідженні [30] представили два варіанти формулювання ознак HOG для виявлення знаків швидкості в Новій Зеландії.

Переважає більшість існуючих систем складається з класифікаторів, які були навчені з використанням вручну позначених реальних зображень, наприклад [25], [31], що є повторюваним, трудомістким і схильним до помилок процесом. Запропонований метод дозволяє уникнути збору та ручного маркування навчальних даних, оскільки він вимагає лише синтетичних графічних зображень знаків, які були отримані з онлайн бази даних дорожніх знаків [32].

Крім того, хоча багато існуючих систем повідомляють про високі показники класифікації, загальна кількість розпізнаних класів дорожніх

знаків, як правило, дуже обмежена, наприклад, сім класів у [16], 42 класи в [17] або 20 класів у [33].

Запропонована система використовує всі екземпляри дорожніх символів на основі ідеограм, що використовуються у Великобританії, і, отже, виконує відповідність у цьому більшому наборі. Очікується, що даний підхід буде однаково функціональним, якщо його застосувати до баз даних дорожніх знаків інших країн, отриманих подібним чином.

Багато запропонованих систем страждають від низької швидкості, що робить їх непридатними для застосування до проблем реального часу.

Деякі методи повідомляють про час обробки в кілька секунд для одного кадру [7], [11], [25], [34]. Розглянута система працює із середньою швидкістю 20 кадрів за секунду.

На ринку існує декілька комерційних систем розпізнавання дорожніх знаків, зокрема [35] і [36].

Такі комерційні системи також розпізнають дуже обмежений набір дорожніх знаків, наприклад, система, розроблена Mobileye [35], виявляє лише знаки обмеження швидкості та знаки заборони обгону. Порівняти з цими комерційними системами важко, враховуючи, що інформації про їхню продуктивність мало.

Об'єкти на зображеннях являють собою впорядкований набір ознак. Набір об'єктів, які мають спільні властивості, називається класом.

Машинне розпізнавання об'єктів на зображеннях полягає в здатності автоматично класифікувати необроблені дані та відносити їх до певного класу, підкреслюючи характерні ознаки об'єкта.

Проблема автоматичного виявлення об'єктів на зображеннях відносно нова, вона виникла в другій половині 20 століття.

Перші системи розпізнавання використовували прості геометричні моделі і вимагали участі адміністратора, який виділяв ознаки об'єкта на зображенні. Потім система зробила чисельні вимірювання розмірів і відстаней вибраних об'єктів відносно контрольних точок.

У даний час розпізнавання об'єктів на зображеннях базується на складних математичних уявленнях про існуючі процеси і користується великим попитом у сферах комп'ютерного зору, обробки та аналізу зображень, біометрії, систем безпеки та відеоспостереження.

Сучасні методи, які вирішують завдання розпізнавання об'єктів, використовуються для вирішення найрізноманітніших завдань: розпізнавання облич, відбитків пальців, сітківки ока, друкованих символів, автомобільних номерних знаків, маркування на поверхнях різних об'єктів і т.д.

На сьогоднішній день досягнуто значних успіхів у вирішенні проблем розпізнавання об'єктів і символів на зображеннях, але існує ряд труднощів, які значно знижують достовірність використання сучасних методів:

- низький дозвіл зображення. У результаті особливості об'єктів можуть бути погано помітні;

- наявність складної фонові структури в зображеннях. Це передбачає наявність на зображеннях сторонніх об'єктів, які можуть мати візуальні властивості, подібні до шуканого об'єкта;

- різні спотворення, що виникають при реєстрації зображення. Реєстрація зображення може бути виконана під поганими кутами, поганими погодними умовами, різними ракурсами та умовами освітлення. В результаті зображення можуть мати різні шумові артефакти, символи можуть бути схильні до афінних і проекційних спотворень;

- кількість шуканих об'єктів на знімку заздалегідь невідома. Все це вимагає використання різноманітних алгоритмів попередньої обробки, що, в свою чергу, ускладнює процес розпізнавання, робить його більш громіздким і збільшує кількість і час обчислювальних процесів.

Тому сьогодні все ще існує потреба в розробці методів і алгоритмів, які вирішують проблеми, згадані вище.

Щоб усунути згадані вище ускладнення, потрібно кілька додаткових підходів до попередньої обробки, що збільшує обчислювальну складність системи.

З цього можна зробити висновок, що сьогодні необхідно впроваджувати методи та алгоритми, які дозволять зменшити вплив вищезазначених труднощів на процес виявлення.

Розпізнавання дорожніх знаків на зображеннях можна логічно розділити на два основних етапи:

- розпізнавання положення дорожнього знака на зображенні;
- розпізнавання дорожніх знаків.

Для реалізації кожного з цих етапів використовуються індивідуальні методи та алгоритми.

Основний принцип цих методів полягає в порівнянні кожної області зображення з певним шаблоном, тим самим визначаючи взаємну кореляцію. Шаблони можуть бути задані вручну або визначені будь-якою функцією та можуть відповідати всьому об'єкту або окремим компонентам об'єкта.

Якщо коефіцієнт кореляції перевищує заданий поріг, область дослідження зображення позначається як така, що містить зображення дорожнього знака.

Існують також різні вдосконалення цього підходу, які використовують кілька шаблонів, що відповідають окремим компонентам об'єктів, які являють собою області у вигляді ліній.

Робота цих алгоритмів складалася з кількох етапів, кожен з яких визначав достовірність знайдених ділянок. Коли область зображення збігається з одним із шаблонів, вона позначається як область інтересу для подальшого аналізу.

Таким чином, підвищується ступінь схожості областей зображення із заданим шаблоном. Далі виявлені ділянки порівнювалися з іншими шаблонами, що дозволяло визначити наявність потрібного об'єкта.

Ці методи в основному використовувалися в ранніх роботах з класифікації та виявлення об'єктів на зображеннях. Це були перші спроби формування особливостей образу предметів [37].

Сучасні модифікації шаблонних методів демонструють високу точність розпізнавання. Недоліком цих методів є низька швидкість роботи і висока чутливість до спотворення об'єктів на зображеннях.

Метод перетворення Гафа. Цей метод був запропонований П. Гафом у 1962 році. Метод полягає в пошуку на зображенні об'єктів, що належать до певного класу геометричних фігур: прямих, кіл тощо.

Пошук здійснюється за допомогою процедури голосування, застосованої до простору параметрів зображення. Шуканий об'єкт представляється у вигляді параметричного рівняння, параметри якого представляють так званий фазовий простір.

Суть перетворення Гафа полягає в тому, що кожна точка на зображенні може належати серії ліній.

Для покращення просторових частот на початковому етапі цього процесу виконується посилення контрасту, бінаризація або виділення країв на зображенні. Далі виконується послідовний аналіз кожного пікселя і його сусідів. Одночасно розраховується вага кордону в цій точці.

Якщо вага відповідає вказаній межі, параметри рядка обчислюються і значення у відповідній клітинці масиву збільшується.

Далі досліджуються максимальні значення в масиві, за які голосує більша кількість пікселів зображення. Таким чином, знаходять параметри рівнянь шуканого об'єкта.

Дескриптори локальних особливостей. Дескриптори локальних ознак часто використовуються, щоб підкреслити особливості об'єктів на зображеннях.

Дескриптор – це набір параметрів, які можна використовувати для опису, наприклад, властивостей зображення, таких як колір, текстура тощо. Початковим етапом розпізнавання об'єктів на зображенні за допомогою дескрипторів є визначення характерних точок.

Характерні точки – це точки з високими локальними інформативними параметрами, які не змінюються при різноманітних фотометричних і

геометричних перетвореннях зображення. Детектори розраховуються для знайденого набору точок на зображенні [38].

Одним із найпопулярніших алгоритмів, який включає дескриптор і детектор для точок ознак зображення, є метод SIFT (Scale Invariant Feature Transform) [39].

Цей алгоритм є локальною гистограмою напрямків градієнта зображення. Принцип роботи алгоритму SIFT полягає в обчисленні згортки вихідного зображення за допомогою гаусового ядра зі змінним параметром згладжування. Після цього зображення перетворюються на однаковий розмір і обчислюється їх різниця.

Далі кожен піксель зображення порівнюється з вісьмома сусідніми пікселями в тих самих параметрах і масштабі, з дев'ятьма сусідніми пікселями в більшому масштабі та з дев'ятьма в меншому масштабі.

Пікселі вибираються як характерні точки, для яких локальні екстремуми перевищують заздалегідь визначений поріг.

Для кожної обраної точки обчислюється певний локальний дескриптор, що характеризує напрямок градієнтів у заданому околі пікселів. У 2005 році французькі дослідники представили алгоритм HOG [40].

Принцип дії цього алгоритму полягає в тому, що зображення подається у вигляді щільної сітки з рівномірно розташованих чарунків. Гістограми напрямків градієнта розраховуються для пікселів у кожній клітинці.

На основі отриманих параметрів будується дескриптор. Для підвищення точності в цьому алгоритмі нормалізація використовується для перекриття локального контрасту гістограм.

Нормалізовані дескриптори мають підвищену стійкість до змін інтенсивності освітлення.

У 2006 році вперше було представлено алгоритм SURF (Speeded Up Robust Features) [41], який містить дескриптор і детектор для характерних пікселів. У цьому алгоритмі при обчисленні характерних точок використовуються не гістограми зважених градієнтів, а цілочисельні

прямокутні фільтри різних масштабів. Це забезпечує стійкість до поворотів об'єкта і до зміни масштабу.

Пікселі, в яких локальні екстремуми перевищують заданий поріг, вибираються як характерні точки, і в них обчислюються локальні дескриптори. Навколо точки будується квадратна площа, яка розділена на кілька підгілок.

У кожній підгілці розраховуються відповіді на два типи вейвлетів – горизонтально і вертикально спрямовані. Отриманий зворотний зв'язок зважений і підсумований за Гаусом.

Перераховані дескриптори зазвичай використовуються для розпізнавання різних об'єктів, у тому числі номерних знаків. Ці дескриптори забезпечують високий ступінь інваріантності до геометричних перетворень і змін масштабу зображення.

Недоліком використання дескрипторів є низька стабільність роботи в різних умовах освітлення, на поверхнях, що відбивають, а також при різних кутах виявлення об'єктів [42].

Методи аналізу гістограмного зображення. Ці методи засновані на припущенні, що частотна характеристика області розташування об'єкта, що містить символічні зображення, відрізняється від інших областей зображення і має більш високу піксельну інтенсивність.

Для покращення просторових частот і придушення шуму на початковому етапі цього підходу виконується посилення контрасту, бінаризація або вибір меж зображення. Це робить краї об'єктів на зображенні більш контрастними, а фон темнішим.

Для таких завдань, як виявлення маркування на технічних об'єктах і автомобільних номерних знаках, бажаним об'єктом, зазвичай, є серія темних символів і ліній на світлому фоні, або навпаки. Тому ця область зображення має високу контрастність.

Ідея цих методів полягає в наступному: зображення сканується і обчислюється середнє значення яскравості пікселів у кожному рядку

зображення (іноді обчислюється по стовпцях). У точці, де розташований потрібний об'єкт, середня інтенсивність пікселів значно відрізняється від інших областей зображення. Максимальне значення отриманої проекції може відповідати розташуванню об'єкта.

Перевагою цих методів є простота реалізації та висока швидкість роботи. Недоліком цих методів є їх чутливість до будь-якої області зображення, яка має параметри інтенсивності пікселів, схожі на дорожній знак.

Метод головних компонентів. Метод аналізу головних компонент (PCA або Principal Component Analysis) був запропонований у 1901 році.

Цей метод використовується в різних сферах, наприклад в обробці та аналізі зображень, розпізнавання образів, стиснення даних тощо. Метод PCA спрямований на зменшення розмірності даних із мінімальною можливою втратою інформації, що є дуже актуальним у сфері аналізу та обробки зображень.

Для ефективного використання цього методу зображення об'єктів повинні бути отримані в однакових умовах реєстрації. Перевага методу PCA полягає в зменшенні розмірності даних, що є важливим фактором при роботі з великими базами даних зображень.

Лінійний дискримінантний аналіз. Лінійний дискримінантний аналіз (LDA або Linear Discriminant Analysis) спрямований на вирішення задачі класифікації об'єктів на кілька класів. На відміну від PCA, цей метод не ставить за мету мати найменший розмірний простір ознак, але ефективно описує навчальний набір даних. Основне завдання цього методу – знайти проекцію ознак у простір, де різні класи об'єктів найбільше відрізняються один від одного.

Для кожного об'єкта зображення, що належить до певного класу, є характерні ознаки, які виявляють загальну схожість з ознаками інших об'єктів цього класу. Набір таких об'єктів утворює шаблон навчання, де відома

приналежність кожного об'єкта до певного класу. Набір ознак визначається проєктуванням простору ознак у підпростір менших розмірів.

Крім того, вектори ознак різних класів об'єктів повинні мати параметри, що максимально відрізняються один від одного, щоб мінімізувати їх внутрішньокласові відмінності та збільшити відмінності між класами. Таким чином формуються максимально компактні кластери характерних ознак об'єктів, які відповідають різним класам, що мають мінімально можливі перекриття між собою [43]. Завдання класифікації зводиться до виявлення характерних ознак вхідного образу і співвіднесення їх з наявними ознаками, сформованими в результаті навчання. Таким чином визначається належність даного об'єкта зображення до певного класу.

Алгоритм AdaBoost. Алгоритм AdaBoost (Adaptive Boosting) був запропонований Йоавом Фройндом і Робертом Шейпіром у 1999 році [44]. Цей алгоритм є алгоритмом машинного навчання та успішно використовується для вирішення проблем класифікації об'єктів у зображеннях і відеорядах. Алгоритм AdaBoost має адаптивний принцип роботи і являє собою каскадну структуру слабких класифікаторів, кожен з яких вчиться на помилках попереднього. Кожен слабкий класифікатор фокусується на певному наборі ознак і може дати дві відповіді: «вірно» або «хибно». У кожному каскаді визначається сума значень слабких класифікаторів і порівнюється з заданим порогом. Таким чином, каскадний класифікатор видаляє з розгляду всі ділянки зображення, де немає ознак шуканого об'єкта. Це залишає лише ті області, які, швидше за все, містять потрібний об'єкт.

У 2001 році Пол Віола і Майкл Джонс запропонували вдосконалений алгоритм – алгоритм Віоли-Джонса на основі AdaBoost. Цей алгоритм дозволяє виявляти об'єкти на зображеннях за ознаками Хаара. Ознаки Хаара є прямокутними ділянками, що складаються з кількох прилеглих частин.

Різні класи об'єктів мають індивідуальні властивості, які можна виразити через розподіл властивостей Хаара. У кожній області зображення розраховується кілька тисяч можливостей для розміщення об'єктів залежно

від економічного статусу та масштабу. Інтенсивності пікселів у чорних і білих областях ознак Хаара підсумовуються, а потім обчислюється різниця між сумами [45].

Найчастіше алгоритм Віоли-Джонса використовується для розпізнавання людей, але також відомо, що він використовується для вирішення проблем розпізнавання номерних знаків автомобіля. Даний алгоритм показує досить високу точність виявлення, а також високу швидкість роботи. Недоліком цього алгоритму є відносно низька інваріантність до афінних і проєкційних спотворень об'єктів на зображеннях і рівнях освітленості [46].

Метод виявлення об'єктів Віоли-Джонса є найпопулярнішим алгоритмом, який дозволяє високошвидкісно визначити положення об'єктів на зображеннях. Його запропонували Пол Віола та Майкл Джонс у 2001 році. Хоча основним завданням алгоритму є розпізнавання людей, його можна використовувати для розпізнавання різних класів об'єктів. Існує багато реалізацій, у тому числі в бібліотеці комп'ютерного зору OpenCV.

Згортові нейронні мережі (CNN або Convolutional Neural Networks). У 1998 році дослідники Ю. Лекусн, Л. Ботту, Ю. Бенгіо і П. Хаффнер запропонували тип нейронних мереж, які працюють за принципом зорової системи людини, які називаються згортковими нейронними мережами.

CNN – особливий клас багат шарових перцептронів, які мають двовимірну структуру і добре підходять для обробки зображень з високим ступенем інваріантності до трансляції, обертання, масштабування та інших спотворень вхідних даних.

Структура CNN є послідовністю двох типів шарів: згортання і субселективності. Кожен шар складається з набору рівнів (характерних карт), які, у свою чергу, складаються з нейронів. Кожен нейрон згорткового шару має зв'язок з невеликою групою нейронів попереднього шару (локальне рецептивне поле).

Локальні рецептивні поля нейронів згорткового шару частково перекриваються за принципом тайлінгу.

Значення нейронів з локального рецептивного поля множаться на матрицю синаптичних коефіцієнтів і результат записується у відповідний нейрон згорткового шару [47].

Завдяки своїм перевагам CNN досі активно використовуються для виявлення людей та інших об'єктів на зображеннях і відеорядах.

1.5 Постановка задачі дослідження

Актуальність теми зумовлена тим, що дорожні знаки є невід'ємною частиною дорожнього середовища. Вони призначені для регулювання потоку транспортних засобів, надавати конкретну інформацію учасникам руху, або застерігають від несподіваних дорожніх обставин. Сприйняття та швидке тлумачення дорожніх знаків має вирішальне значення для безпека водія.

Автоматизація процесу виявлення, розпізнавання та класифікації дорожніх знаків була визнана одним з найважливіших напрямів, як тільки обробка відео та фото стала доступною на комп'ютері. Сьогодні це вважається критичним завданням у сучасних системах підтримки водія, але у даній сфері все ще залишається великий простір для вдосконалення підходів вирішення проблеми.

Об'єктом дослідження є процес розроблення застосунку для розпізнавання та класифікації дорожніх знаків України.

Метою дослідження є розроблення програмного застосунку із застосуванням методів глибокого машинного навчання, спрямованого на розпізнавання та класифікацію дорожніх знаків.

Враховуючи мету дослідження, необхідно вирішити такі завдання:

– проаналізувати сучасний стан систем розпізнавання дорожніх знаків;

- провести класифікацію та аналіз існуючих методів розпізнавання дорожніх знаків;
- проаналізувати особливості дорожніх знаків України;
- провести аналіз літературних джерел щодо апробації результатів розпізнавання дорожніх знаків;
- розглянути механізм оброблення зображень дорожніх знаків;
- дослідити принципи розпізнавання складових дорожніх знаків;
- розробити методику розпізнавання дорожніх знаків;
- провести вибір інструментальних засобів для реалізації методів розпізнавання дорожніх знаків;
- навести детальний опис етапів програмної реалізації методів розпізнавання дорожніх знаків;
- провести тестування розроблених застосунків та проаналізувати результати;
- виявити перспективи подальшої роботи;
- зробити висновки щодо виконаної роботи.

2 ОСОБЛИВОСТІ РОЗРОБЛЕННЯ МЕТОДІВ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

2.1 Механізм оброблення зображення дорожніх знаків

Робота з зображенням, а особливо розпізнавання та класифікація об'єкту на ньому – досить складна задача для комп'ютеру. Якщо для людини виділити та миттєво розпізнати, що саме зображено для фотографії – це справа, яка не потребує великих зусиль, то для комп'ютера потрібен цілий ряд складних методів, кожен з яких може дати різну ефективність на різних даних [26].

Зазвичай, механізм обробки зображень дорожніх знаків складається з двох послідовних процесів:

- етап розпізнавання конкретної заданої області, тобто області на картинці, де ймовірно може бути елемент дорожнього руху;
- етап ідентифікації елемента, тобто встановлення типу елемента, наприклад, дорожній знак, а згодом, встановлення конкретно, який це знак.

Основною задачею розпізнавання конкретної області – це виділення певних регіонів з картинки, які найбільш ймовірно є дорожніми знаками.

Для вирішення цієї задачі можуть використовуватись різні підходи, які використовують основні характеристики елементів дорожнього руху, такі як форма і колір [30].

У загальному методи розпізнавання області можна розбити на три групи:

- розпізнавання об'єкта за допомогою кольору;
- розпізнавання об'єкта за допомогою форми;
- розпізнавання об'єкта гібридними методами.

Розпізнавання об'єкта за допомогою кольору. Використання інформації про колір об'єкту є одним із найпоширеніших підходів до розпізнавання елементів дорожнього руху. Ця інформація знаходиться за допомогою порогових або розширених методів сегментації і використовується для пошуку областей, де найбільш ймовірно може бути дорожній знак.

Однак, головним недоліком цього методу – є висока чутливість до нестабільних умов освітлення, тому цей метод є дуже залежним від часу доби та погодних умов. У сучасних системах цей метод використовується з різними типами кольорових просторів, найбільш поширеними є RGB, HSI/HSV, YUV, YCbCr, CIE Lab та CIECAM97.

Методи розпізнавання об'єктів на основі характеристик мають високу швидкодію, хорошу стійкість та інші характеристики, які можуть певною мірою поліпшити продуктивність розпізнавання об'єктів на картинці [30].

У таких методах розпізнавання об'єктів, зазвичай, використовуються кольорові картинки високої роздільної здатності, але такі методи дають низьку точність на чорно-білих картинках.

У цих підходах до розпізнавання елементів, вхідні картинки розглядають, як набір пікселів, в яких кожен піксель має певні характеристики, що визначають його колір, потім набір пікселів ділять на окремі ділянки, в яких сусідні пікселі мають схожі характеристики кольору.

Знаки дорожнього руху виявляють за допомогою порогування кольору та алгоритмом розпізнавання.

У таких підходах велику роль відіграє вибір простору кольору, тобто рівняння, яке визначає характеристики кольору пікселя.

Вхідні картини системи розпізнавання дорожніх знаків, зазвичай, трансформують у вибраний простір кольору, де елементи, що шукаються, більш виразні [48].

Існуючі алгоритми розпізнавання дорожніх знаків використовують такі простори кольорів:

- RGB (червоний, зелений, синій);
- HSV (відтінок, насичення, значення);
- HSI (відтінок, насичення, інтенсивність) та інші [26].

Для вилучення червоного або синього кольору з вхідного зображення використовується колірний простір HLS (Hue, Luminance and Saturation).

Спочатку зображення конвертується з кольорового простору «RGB» в «HLS» (рис. 2.1).



Рисунок 2.1 – Перетворення кольорового простору

Після перетворення кольорів виконується видалення червоного і синього кольору.

Це призводить до того, що зображення стає фактично монохромним (чорно-білим), при чому всі об'єкти, що мали червоний або синій колір, стають білого кольору (рис. 2.2).



Рисунок 2.2 – Видалення червоного та синього кольорів

Розпізнавання об'єкта за допомогою форми. Ще однією важливою характеристикою елементів дорожнього руху – є форма (кругла, трикутна, восьмикутна або ж прямокутна).

У порівнянні з методом розпізнавання об'єкту за допомогою кольору, даний метод не має залежності від погодних умов і змін яскравості завдяки тому, що не використовує інформацію про колір об'єкту.

Саме тому багато сучасних систем використовують інформацію про форму об'єкту для його ідентифікації [29].

Для розпізнавання знаків дорожнього руху за формою знаку використовують різні механізми розпізнавання форми, наприклад, пошук контурів і їх наближення для отримання остаточного рішення, виходячи з кількості контурів.

Даний метод також може використовуватись як детектор дорожніх знаків, завдяки тому, що він дозволяє знаходити області інтересу з вхідної картинки, а потім їх окремо розглядати для подальшої ідентифікації знаку. Проте, такий підхід може бути затратним по пам'яті, якщо вхідна картинка має велику роздільну здатність.

Існує багато різних механізмів розпізнавання форми елемента дорожнього знаку. Як, наприклад, трансформація Хафа, Similarity Detection, Distance Transform Matching, виявлення характеристик контурів, Haar-Like Features [45].

Найбільш поширеним є – трансформація Хафа. Алгоритм Хафа може використовуватись для розпізнавання ліній, кіл, прямокутників чи інших кривих. Цей метод вперше був введений в 1962 році, і його першим застосуванням було пошук ліній в послідовності зображень [49].

Основна перевага трансформації Хафа – це його стійкість до шумів, масштабування та обертання.

Деякі автори [49], [50] використовували трансформацію Хафа для кіл – для розпізнавання круглих дорожніх знаків, а для розпізнавання трикутних знаків використали трансформацію Хафа для розпізнавання прямих ліній.

Наступним кроком є видалення «шумів» і знаходження контурів дорожніх знаків (рис. 2.3) [49].

Подальшим етапом є фільтрація і видалення непотрібних контурів. Будь-який дорожній знак можна описати прямокутником, який матиме відповідні співвідношення сторін.



Рисунок 2.3 – Знаходження контурів, де може знаходитися знак та фільтрація контурів

Для даних типів знаків співвідношення сторін, враховуючи схильність або поворот знака, лежить в діапазоні від 0,7 до 1,3.

Відповідно, відфільтрувавши контури, отримаємо потрібні контури. Ідентифікація контуру відбувається шляхом порівняння контурів за шаблоном (Template Matching).

Для кожного знака був зроблений шаблон (рис. 2.4) [51], з яким порівнюється контур.



Рисунок 2.4 – Шаблони деяких знаків

Розпізнавання об'єкта гібридними методами. Як відомо, методи розпізнавання елементів дорожнього руху за допомогою кольору та форми мають свої переваги та недоліки.

Тому багато систем використовують різні комбінації цих методів для досягнення максимальної точності розпізнавання. Системи розпізнавання елементів дорожнього руху включають в себе крок сегментації кольору, за яким слідує етап розпізнавання форми елемента [38].

2.2 Принципи розпізнавання складових дорожніх знаків

Існує багато принципів розпізнавання образів на зображеннях. Розуміння даних методів важливо для вирішення різного роду завдань.

По-перше, важливо розуміння теорії розпізнавання образів.

Основними термінами є:

- клас – множина об'єктів, що мають загальні властивості, класів може бути необмежена кількість;
- класифікація – процес призначення міток класу об'єктів, відповідно до деякого опису властивостей цих об'єктів;
- класифікатор – пристрій, який в якості вхідних даних отримує набір ознак об'єкта, а в якості результату видає мітку класу;
- верифікація – процес зіставлення примірника об'єкта з однією моделлю об'єкта або описом класу;
- ознака – кількісний опис тієї чи іншої властивості досліджуваного предмета або явища;
- простір ознак – це N -мірний простір, визначений для даної задачі розпізнавання, де N – фіксоване число вимірюваних ознак для будь-яких об'єктів.

Вектор з простору ознак x , відповідний об'єкту завдання розпізнавання це N -мірний вектор з компонентами (x_1, x_2, \dots, x_N) , які є значеннями ознак для даного об'єкта [28].

Таким чином, вся задача розпізнавання зводиться до виділення істотних ознак для кожного класу i , в кінцевому підсумку, віднесення вхідних даних до одного з них за допомогою виявлення ключових ознак в оригінальному зображенні.

Тобто задачу розпізнавання образів можна розділити на декілька окремих завдань, а саме:

- отримання вхідних даних, за допомогою сенсорів, камер відеоспостереження, добірок даних;
- первинна обробка зображень така, як нормалізація даних, фільтрація шумів, виявлення ознак;
- формування векторів ознак, за допомогою вибору найбільш значущих ознак, можна виділити непересічні множини класів;
- класифікація чи прогнозування на основі отриманих даних про класи.

Існуючі алгоритми розпізнавання складових дорожніх знаків можна умовно розбити на дві категорії: впроваджені в комерційне виробництво системи розпізнавання дорожніх знаків і науково-дослідні роботи.

Системи розпізнавання, що розробляються і впроваджуються різними автомобільними виробниками [6].

Розглянемо лише частину алгоритмів функціонування розпізнавання знаків (обмеження швидкості).

Алгоритм розпізнавання складових дорожніх знаків.

Етап 1: виявлення.

Використовуючи кольоровий простір $YCrCb$, з вихідного зображення витягується канал Cr , потім отримане півтонування адаптивно бінаризується, що забезпечує стійкість до перепадів яскравості в межах знака.

Виділені білим кольором на бінарному зображенні групи пікселів використовуються для детектування кіл на основі RANSAC-технології (метод оцінки параметрів моделі на основі випадкових вибірок), яка дозволяє визначити, чи відповідає пов'язаний компонент заданої моделі геометричній фігурі, використовуючи для перевірки лише частину точок компонента.

Таким чином, сильне зашумлення зображення або часткове перекриття знаку не перешкоджає його виявленню.

Етап 2: розпізнавання.

З вихідного зображення, перетвореного в градації сірого, витягується фрагмент, що обмежується облямівкою, виконується його бінаризація з граничним значенням, рівним середньому значенню пікселів вхідного зображення.

Даний принцип дає хороше виявлення темних цифр на світлому фоні знаку. Далі отримана картинка масштабується до розміру 20×20 пікселів і подається на вхід нейронної мережі.

Для класифікації використовується класична багатошарова мережа з прямим розповсюдженням сигналів.

Структура мережі включає 400 у вхідному шарі (тому що розмір нормалізованого зображення 20×20), 30 нейронів в прихованому і 12 у вихідному шарі.

Кількість нейронів у прихованому шарі підбиралося емпіричним шляхом. Перші десять виходів відповідають значенням максимальної швидкості: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. 11 вихід приписаний знаку скасування обмежень швидкості, 12 використовується для оповіщення про те, що об'єкт на зображенні це не знак.

Сигнали на виходах мережі приймають значення від 0 до 1 і вказують на ймовірність відповідності вхідного об'єкта відповідного класу. Класифікація вважається успішною, якщо різниця між двома найбільшими вихідними значеннями більше 0,5, в іншому випадку результат ігнорується.

Продуктивність даного методу дозволяє обробляти до 27 кадрів в секунду при використанні одноядерного процесора з тактовою частотою 1,6 GHz.

Розглянутий принцип має достатню стійкість до шумів і часткового перекриття знаків сторонніми об'єктами, при цьому демонструючи прийнятну продуктивність. Проте, він має свої недоліки:

- не забезпечує інваріантності повороту знаків у площині зображення;
- не забезпечує стійкість до змін ракурсу;
- алгоритм розпізнає тільки знаки з формою кола.

2.3 Методика розпізнавання дорожніх знаків

Можна визначити декілька методів, що були успішно застосовані дослідниками стосовно питання розпізнавання дорожніх знаків, та показали себе ефективними для подальшого вивчення проблематики.

Перший метод Histogram of Oriented Gradients (HOG). Гістограма напрямлених градієнтів – це дескриптори особливих точок, які використовуються в комп'ютерному баченні та обробці зображень з метою розпізнавання об'єктів. Дана техніка заснована на підрахунку кількості напрямків градієнта в локальних областях зображення [28].

Дескриптор HOG зосереджується на структурі або формі об'єкта. У разі крайових особливостей визначаємо лише, чи є піксель краєм, чи ні. HOG також може забезпечити напрямок краю. Це робиться шляхом вилучення градієнта та орієнтації (або можна сказати величини та напрямку) країв.

Крім того, ці орієнтації обчислюються «локалізованими» частинами. Це означає, що повне зображення розбивається на менші області, і для кожної області розраховуються градієнти та орієнтація. Тоді HOG створить гістограму для кожної з цих областей окремо.

Гістограми створюються з використанням градієнтів та орієнтацій значень пікселів, звідси назва «Гістограма орієнтованих градієнтів».

Отже, дескриптор функції HOG підраховує випадки орієнтації градієнта в локалізованих частинах зображення.

Крок 1. Попередня обробка даних [28].

Попередня обробка даних є вирішальним кроком у будь-якому проєкті машинного навчання, і це не відрізняється при роботі з зображеннями.

Потрібно попередньо обробити зображення і зменшити співвідношення ширини до висоти до 1:2. Переважно розмір зображення повинен бути 64×128 . Це тому, що будемо розділяти зображення на патчі 8×8 та 16×16 для вилучення об'єктів. Наявність зазначеного розміру (64×128) зробить усі наші розрахунки досить простими.

Крок 2. Обчислення градієнтів (напрямок x та y).

Наступний крок – обчислення градієнта для кожного пікселя зображення.

Виділяємо значення одного пікселя. Тепер, щоб визначити градієнт (або зміну) у напрямку x , потрібно відняти значення зліва від значення пікселя справа. Аналогічно, щоб обчислити градієнт у напрямку y , віднімемо значення пікселя нижче від значення пікселя над вибраним пікселем.

Цей процес дасть дві нові матриці: одна зберігає градієнти в напрямку x , а інша – у напрямку y . Це схоже на використання ядра Собеля розміром 1. Величина буде вищою при різкій зміні інтенсивності, наприклад, по краях.

Цей процес повторюється для всіх пікселів на зображенні. Наступним кроком буде визначення величини та орієнтації за допомогою цих значень.

Крок 3. Обчислити величину та орієнтацію.

Використовуючи градієнти, визначається величину та напрямок для кожного значення пікселя.

Далі потрібно створити гістограму, використовуючи ці градієнти та орієнтації.

Крок 4. Обчислити гістограму градієнтів у клітинках 8×8 .

Гістограми, створені в дескрипторі функцій HOG, не генеруються для всього зображення.

Натомість зображення поділяється на чарунку розміром 8×8 , і гістограма орієнтованих градієнтів обчислюється для кожної клітинки.

Таким чином, отримуємо функції (або гістограму) для менших ділянок, які, у свою чергу, представляють ціле зображення.

Якщо розділимо зображення на чарунки 8×8 і сформуємо гістограми, отримаємо матрицю 9×1 для кожної клітинки.

Візьмемо значення кожного пікселя, знайдемо величину градієнта пікселя та оновимо таблицю частот, що позначає градієнт та появу на зображенні. При цьому, більша вага повинна бути у значення чарунку, який ближче до орієнтації. Після того, як створили НОГ для патчів 8×8 на зображенні, наступним кроком є нормалізація гістограм.

Крок 5. Нормалізація градієнтів у чарунку 16×16 (36×1).

Граденти зображення чутливі до загального освітлення. Це означає, що для певного зображення деяка частина зображення буде дуже яскравою порівняно з іншими частинами. Можемо зменшити цю зміну освітлення, нормалізувавши градієнти, взявши блоки 16×16 .

Отже, буде чотири матриці 9×1 або одна матриця 36×1 . Щоб нормалізувати цю матрицю, розділимо кожне з цих значень на квадратний корінь із суми квадратів значень (рис. 2.5) [28].

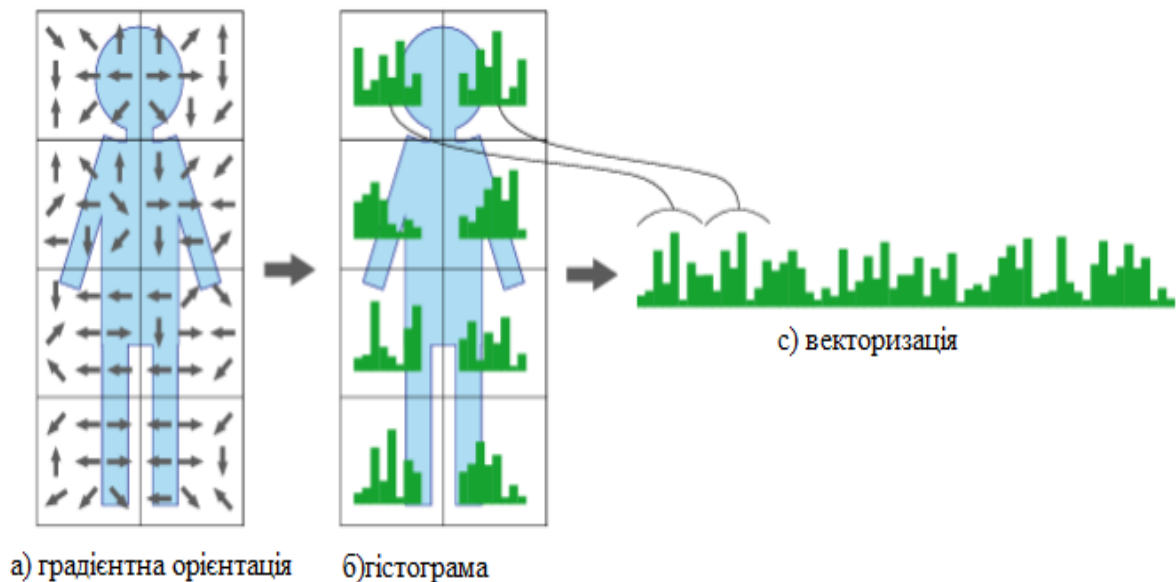


Рисунок 2.5 – Робота алгоритму НОГ

НОГ має дві переваги для виявлення об'єктів. Однією з переваг є стійкість проти дисперсії освітленості, оскільки орієнтації градієнта

обчислюються на основі місцевої різниці інтенсивності. А також стійкість проти деформацій, оскільки незначні зсуви та афінні деформації вносять невеликі зміни у значення гістограми.

Co-HOG – гістограми спільного виникнення орієнтованих градієнтів (функція використовує пари орієнтацій градієнта як одиниці, з яких вона буде гістограми).

Матриця спільного виникнення (гістограма) виражає розподіл орієнтацій градієнта при заданому зміщенню над зображенням (рис. 2.6).

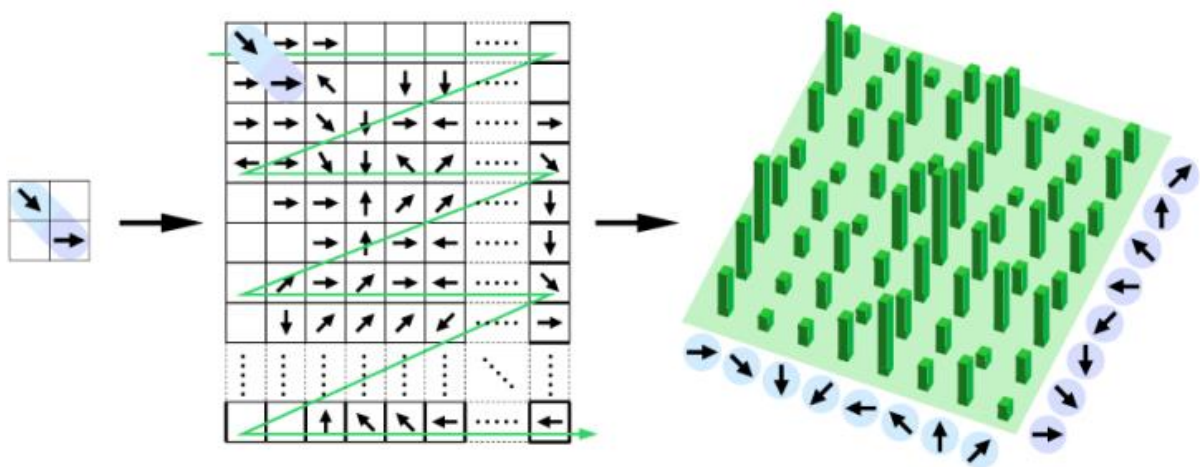


Рисунок 2.6 – Матриця спільного виникнення

Комбінації сусідніх орієнтацій градієнта можуть детально виразити форми. Тож він є інформативним для класифікації об'єктів під кутом.

Математично матриця спільного входження визначається над $n \times m$ зображенням, параметризованим зміщенням (x, y) , як

$$C_{x,y}(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & I(p, q) = i \text{ та } I(p+x, q+y) = j, \\ 0, & \text{інакше} \end{cases} \quad (2.1)$$

де C – матриця спільного входження;

n, m – розмір зображення у пікселях;

x, y – зміщення;

i, j – орієнтацій градієнта.

Алгоритм підрахунку гістограми приведено на рисунку 2.7.

Обчислюємо орієнтації градієнта на зображенні.

$$\theta = \arctg v h, \quad (2.2)$$

де v і h – вертикальний та горизонтальний градієнти відповідно, обчислені за допомогою фільтра Собеля, фільтра Робертса тощо.

```

1: given I: an image of gradient orientation
2: initialize H  $\leftarrow$  0
3: for all positions (p,q) inside of the image do
4: i  $\leftarrow$  I(p,q)
5: k  $\leftarrow$  the small region including (p,q)
6: for all offsets (x,y) such that corresponds neighbors do
7: if (p +x,q +y) is inside of the image then
8: j  $\leftarrow$  I(p+x, q+y)
9: H(k, i, j, x, y)  $\leftarrow$  H(k, i, j, x, y) + 1
10: end if
11: end for
12: end for

```

Рисунок 2.7 – Алгоритм підрахунку гістограм

Позначаємо кожен піксель з однією з восьми дискретних орієнтацій або як безградієнтний. Усі орієнтації $0^\circ - 360^\circ$ поділені на вісім орієнтацій через 45° . Без градієнта означає, що $\sqrt{h^2 + v^2}$ менший за поріг.

Обчислюємо матриці спільного входження за рівнянням (2.1). Використовуючи короткі та дальні зміщення, матриця спільного виникнення може виражати локальні та глобальні форми. Не використовуємо половину зміщень, тому що вони ведуть себе так само, як і інші при обчисленні матриці спільного виявлення.

Матриці спільного виникнення обчислюються для кожної невеликої області.

Невеликі прямокутні області розміщені плиткою $N \times M$ без перекриття. Компоненти всіх матриць спільної появи об'єднуються у вектор (рис. 2.8) [10].

Оскільки SoHOG виражає форми в деталях, він є високорозмірним. Матриця спільного виникнення, розрахована з нульовим зміщенням, має лише вісім ефективних значень, оскільки недіагональні компоненти дорівнюють нулю.

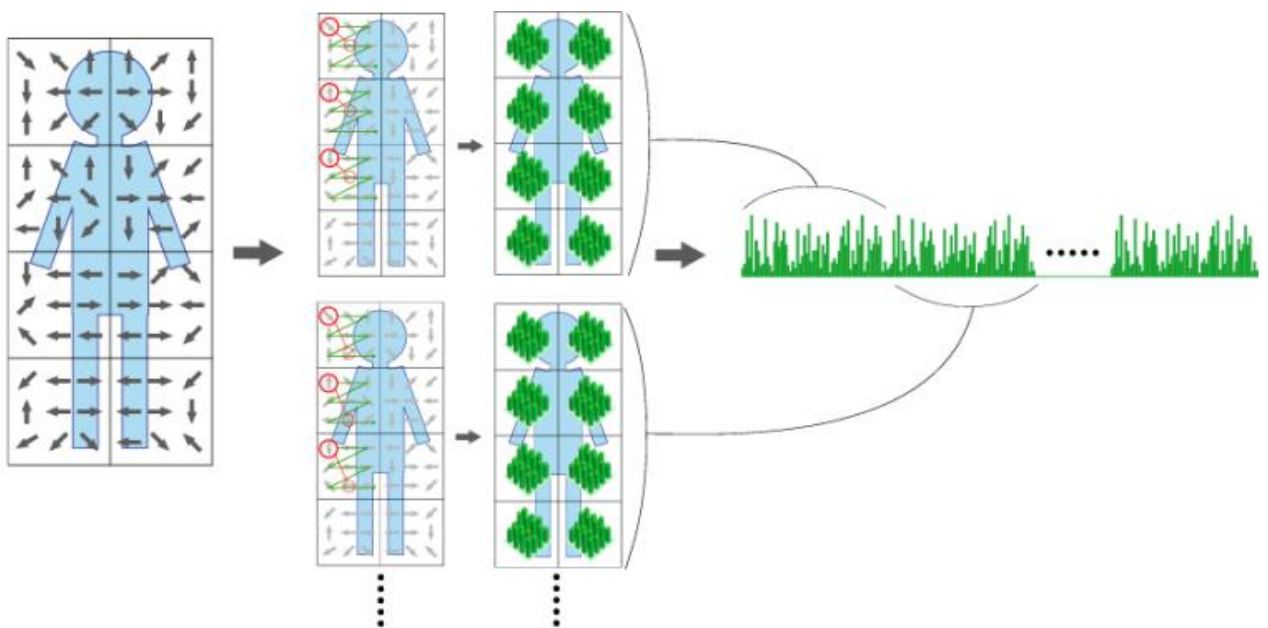


Рисунок 2.8 – Робота алгоритму SoHOG

Таким чином, SoHOG отримує тисячі компонентів із зображення. Насправді ефективних значень менше, оскільки матриці спільного повторення мають кілька компонентів з нульовим значенням.

Компоненти з нульовим значенням не використовуються в класифікації, оскільки їх внутрішній добуток завжди дорівнює нулю. Тим не менш, SoHOG є більш потужним дескриптором функцій, ніж HOG.

Реалізація SoHOG проста. Можемо обчислити SoHOG лише шляхом ітерації для збільшення компонентів матриць спільного появи, тоді як

розрахунок HOG включає більше процедури, такі як зважене голосування за орієнтацією, нормалізація гістограми, перекриття регіонів тощо.

SoHOG має стійкість проти деформації та відхилення освітленості з тих же причин, що і HOG, оскільки SoHOG – це опис характеристик гістограми на основі градієнта.

Останні роки серед багатьох методик, що використовуються для розпізнавання та класифікації зображень, сильно виділяються ті, що використовують можливості машинного навчання та нейронних мереж.

Нейронні мережі дозволяють вирішувати широке коло завдань і представляють із себе структуру з декількох шарів – штучних нейронів (обчислювальних елементів) і зв'язків між ними.

Модель імітує структуру і властивості організації нервової системи живих організмів.

Нейронна мережа отримує на вхід набір сигналів і на виході видає відповідну відповідь (вихідні сигнали), які описують рішення деякої задачі.

Етапи створення та роботи нейронної мережі для розпізнавання мають чітко визначений вигляд (рис. 2.9).

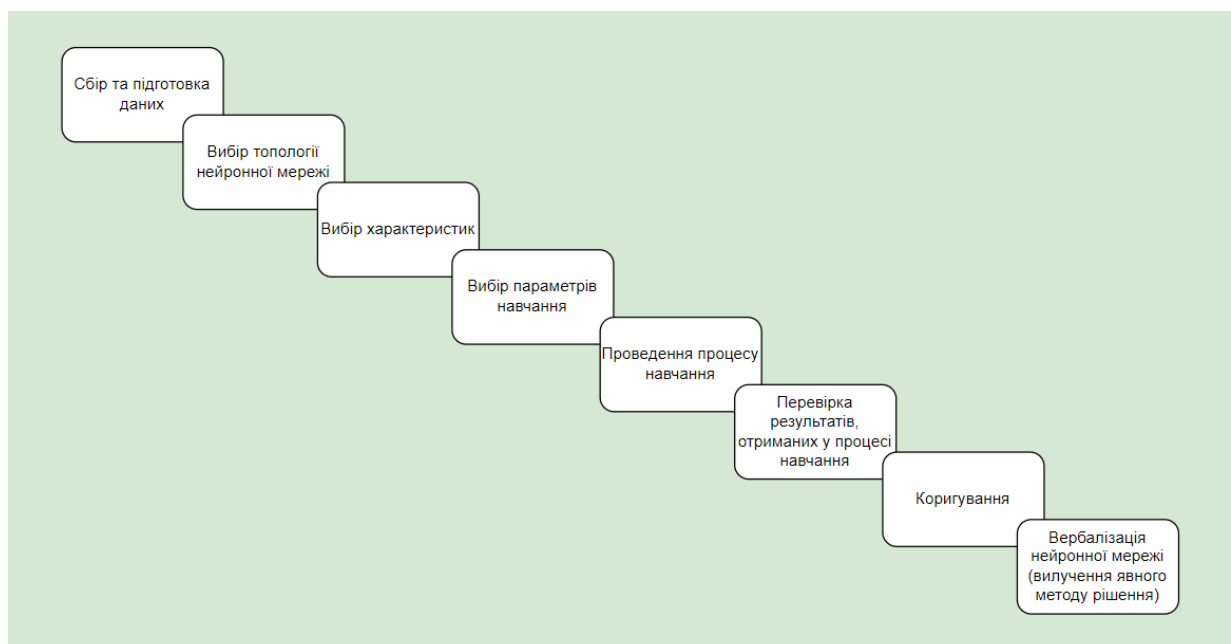


Рисунок 2.9 – Етапи створення та роботи нейронної мережі

Порівнюючи з лінійними методами статистики, нейронні мережі мають змогу ефективно будувати нелінійні залежності, більш точно описують набори даних.

Що ж стосується байєсівського класифікатора, який будує квадратичну розділяючу поверхню, нейронна мережа може побудувати поверхню більш високого порядку. Висока нелінійність розділяє поверхні наївного байєсівського класифікатора (він не використовує коваріаційні матриці класів, як класичний Баєс, а аналізує локальні щільності ймовірності) вимагає значного сумарного числа прикладів для можливості оцінювання ймовірностей при кожному поєднанні інтервалів значень змінних – тоді як нейронна мережа навчається на всій вибірці даних, не фрагментуючи її, що підвищує адекватність налаштування мережі.

Існує багато видів нейронних мереж, що відрізняються і топологією, і характером навчання та зв'язків. Для вирішення задачі розпізнавання розглянемо два види нейронних мереж, що активно використовуються в задачах комп'ютерного зору.

Розглянемо перший вид такої нейронної мережі, як згорткова нейронна мережа.

Згорткова нейронна мережа або CNN відноситься до типу багатосарових мереж прямого поширення, в яких сигнал поширюється в одному напрямку. Принцип роботи мережі бере за основу роботу зорової сенсорної системи тварин та націлений на ефективне розпізнавання образів на зображенні [52].

CNN входить до складу технологій глибокого навчання та є частиною багатьох бібліотек глибокого навчання, таких, як TensorFlow.

Глибоке навчання є підтипом машинного навчання та базується на навчанні ознак даних (feature learning) та побудові ієрархії абстракцій [52]. Якщо алгоритми машинного навчання потребують структурованих даних, то мережі глибокого навчання повністю покладаються на свої шари.

Багаторівневі шари розміщують дані за ієрархією концепцій та вчать на власних помилках [52].

Ідея архітектури CNN полягає в чергуванні згорткових та субдискретизаційних (підвибіркових) шарів. Свою ж назву мережа отримала від операції згортки, при якій кожен фрагмент зображення перемножується на ядро згортки поелементно [10]. Отриманий результат при цьому додається та записується в схожу позицію вихідного зображення.

CNN має певну стійкість до змін масштабу, зсуву, поворотів, ракурсу та інших трансформацій зображення, тобто мережа має властивість інваріантності, що була притаманна ще SURF.

Забезпечити цю властивість нейронній мережі допомагають 3 механізми: локальне вилучення ознак, формування шарів у вигляді набору карт ознак та наявність підвибірки [10].

Розглянемо кожен з механізмів детальніше.

При локальному вилученні ознак кожен нейрон отримує вхідний сигнал від локального рецептивного поля на попередньому шарі. Таким чином, йде вилучення локальної ознаки. Як тільки мережа вилучає ознаку, то її подальше точне місцезнаходження вже не має значення через те, що тепер воно визначається відносно інших знайдених ознак.

Першим шаром в згортковій нейронній мережі є вхідний шар. На вхід до мережі подаються вхідні дані у вигляді зображень. Вхідний шар враховує двовимірну структуру зображень і складається, зазвичай, з трьох карт. Представлення зображення у трьох основних каналах – червоному, синьому та зеленому. У випадку, коли зображення представлено в сірих кольорах, створюється лише одна карта на вхідному шарі.

Вхідні дані пікселів нормалізуються від 0 до 1.

$$f(p, min, max) = \frac{p - min}{max - min}, \quad (2.3)$$

де f – функція нормалізації;

p – колір пікселя в діапазоні від 0 до 255;

min – 0;

max – 255.

Іншим важливим механізмом CNN є формування шарів у вигляді набору карт ознак. Таку особливість переважно приймає згортковий шар мережі. Цей шар є набором таких карт ознак, у кожній з яких є синаптичне або скануюче ядро. Простіше кажучи – фільтр.

Кількість карт визначається постановкою задачі та вимогами до неї, але треба зауважити, що велика кількість карт не тільки підвищує якість та точність розпізнавання, але й збільшує обчислювальну складність [52].

Дослідники радять дотримуватись співвідношення 1:2 [10], при якому карта попереднього шару пов'язана з двома картами наступного. Розмір усіх карт згорткового шару обчислюються за спеціальною формулою

$$(w, h) = (mW - kW + 1, mH - kH + 1), \quad (2.4)$$

де (w, h) – розмір згорткової карти;

mW – ширина попередньої карти;

mH – висота попередньої карти;

kW – ширина ядра;

kH – висота ядра.

Як вже було зазначено, синаптичне або скануюче ядро є фільтром, що проходить по карті та знаходить ознаки об'єктів.

Розмір ядра задається самостійно в межах від 3×3 до 7×7 але так, щоб розмір карти ознак був парним [10]. Це необхідно, щоб не допустити втрат інформації при зменшенні підвибіркового шару.

Карта ознак на згортковому шарі спочатку дорівнює нулю, а вага ядер формується випадково в діапазоні від -0,5 до 0,5 [10]. Вікно розміром величини ядра згортки проходить по зображенню та на кожній ітерації поелементно перемножує область, що досліджується, на значення ядра. Сума результату

множення записується до карти наступного шару. Так проводиться операція згортки, що можна представити у вигляді формули

$$(f * g)[m,n] = \sum_{k,l} f[m - k, n - l] * g[k,l], \quad (2.5)$$

де f – початкова матриця зображення;

g – ядро згортки.

Результат згортки може відрізнятись в залежності від методу обробки границь матриці. Може бути менше початкового зображення, того ж розміру або більшого [52]. На рисунку 2.10 показаний процес згортки з результатом меншим за початкове зображення.

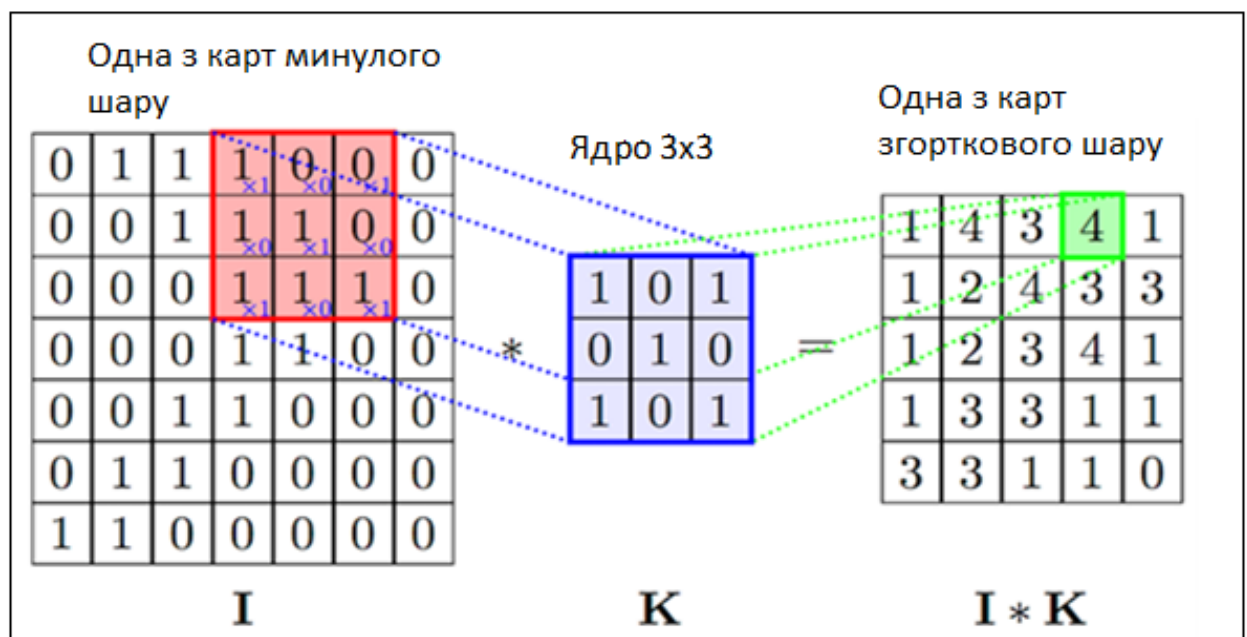


Рисунок 2.10 – Операція згортки [32]

Третім важливим механізмом CNN є підвибірка. Підвибірковий шар йде відразу за згортковим. Він також має карти ознак, але їх кількість так ж, як і в попередньому шарі.

Підвибірковий шар потрібен для того, щоб зменшити розмірність карт ознак. Він фільтрує непотрібні деталі ознак та зменшує вірогідність

перенавчання. Карта ядра підвибіркового шару має розмірність 2×2 та зменшує карти ознак згорткового шару в два рази.

Ця операція має назву *max pooling*, використовує функцію активації, її вигляд приведено на рисунку 2.11 [9].

Функція активації визначає залежність вихідного сигналу від вхідного та відображає числа в проміжку $(0,1)$ – рідше $(-1,1)$. Функція показує вихідне значення нейрону та задає, чи має бути нейрон активним або його можна ігнорувати.

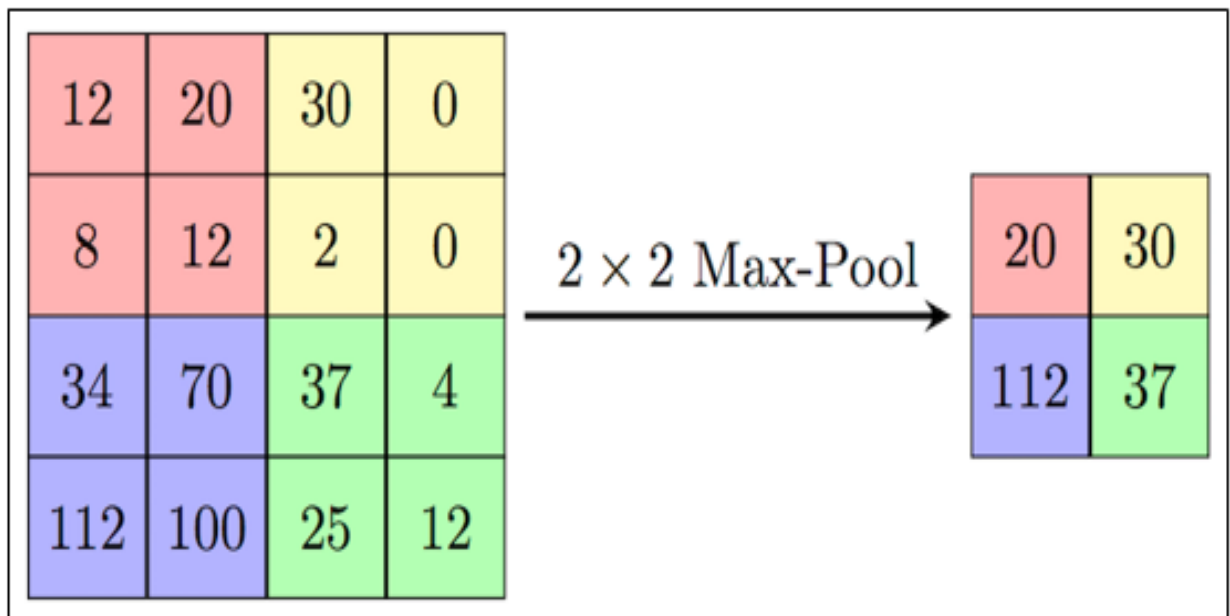


Рисунок 2.11 – Формування карти підвибіркового шару на основі попереднього згорткового шару

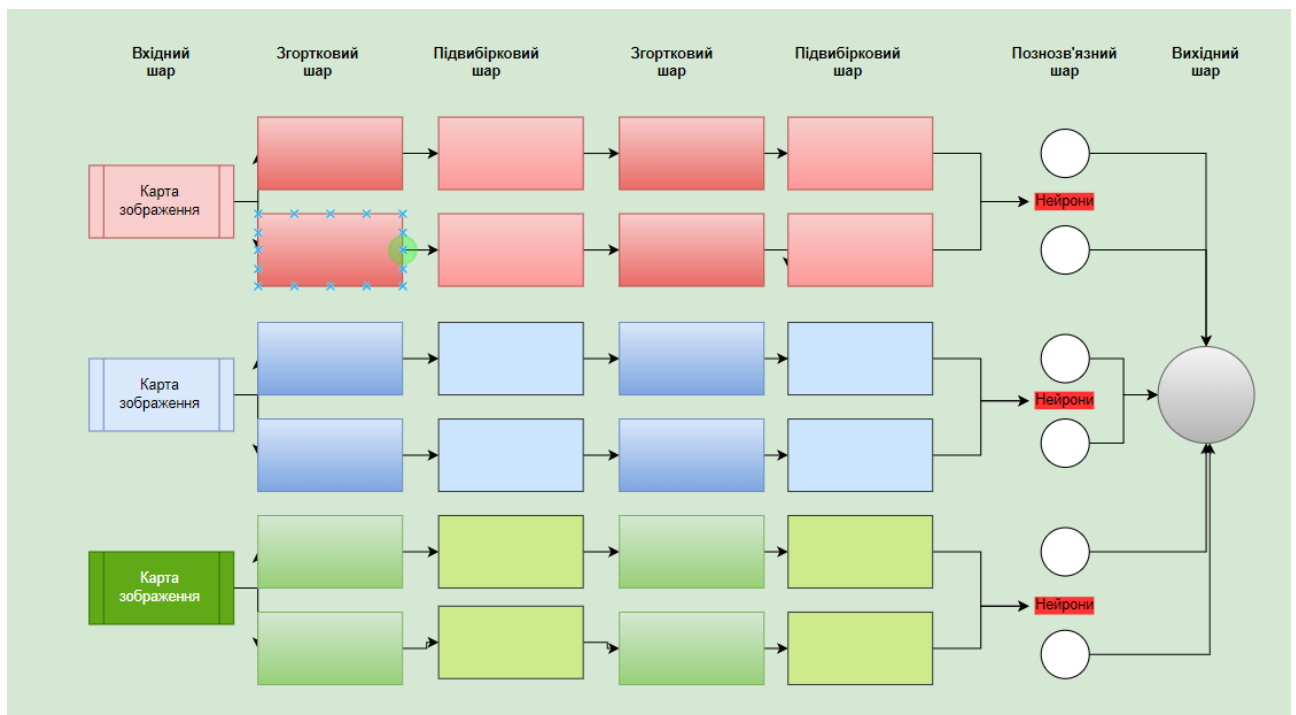
Тобто функція активації визначає функціональні можливості нейронної мережі та її метод навчання [52].

У згортковій нейронній мережі частіше за все використовується функція активації ReLU – *rectified linear unit* або випрямлена лінійна функція. Її похідна дорівнює або 0, або 1, в неї немає ресурсомістких операцій, немає розростання або затухання градієнту, вона сприяє швидкому навчанню, хоч і не завжди надійна і дуже залежить від значень ваги.

Останніми типами шарів CNN є повнозв'язний шар. Мета цього шару – класифікація, за допомогою моделювання складної нелінійної функції, а також вихідний шар [47].

Загальна топологія CNN приведена на рисунку 2.12.

Таким чином, використання CNN є одним з найкращих сучасних підходів до задачі розпізнавання та класифікації образів. Згорткові нейронні мережі володіють властивістю інваріантності, ефективно навчаються та є не дуже складними в використанні. Але при цьому вимагають велику кількість навчальних даних та обчислювальних ресурсів.



Рисунку 2.12 – Топологія згорткової нейронної мережі

При цьому існує велика кількість бібліотек глибокого навчання з відкритим кодом, що надають великий спектр інструментів. Це дозволяє в рази полегшити роботу з нейронними мережами, особливо згортковими.

Тепер розглянемо іншу модель нейронної мережі, що базується на принципах згорткових нейронних мереж.

YOLO – це нейронна мережа, яка за один прохід прогнозує для зображення положення прямокутників, що обмежують, і ймовірності

класифікації. Моделі YOLO здатні обробляти понад 60 кадрів на секунду, а отже, ця архітектура чудово підходить для розпізнавання об'єктів у відео.

Сучасні системи виявлення змінюють призначення класифікаторів для виконання виявлення.

Щоб виявити об'єкт, системи беруть класифікатор для об'єкта, оцінюють його в різних місцях, масштабують на тестовому 44 зображенні. Такі системи, як деформовані деталі моделі (DPM) використовують підхід з розсувним вікном, де класифікатор виконується в рівномірно розташованих місцях по всьому зображенню (рис. 2.13).

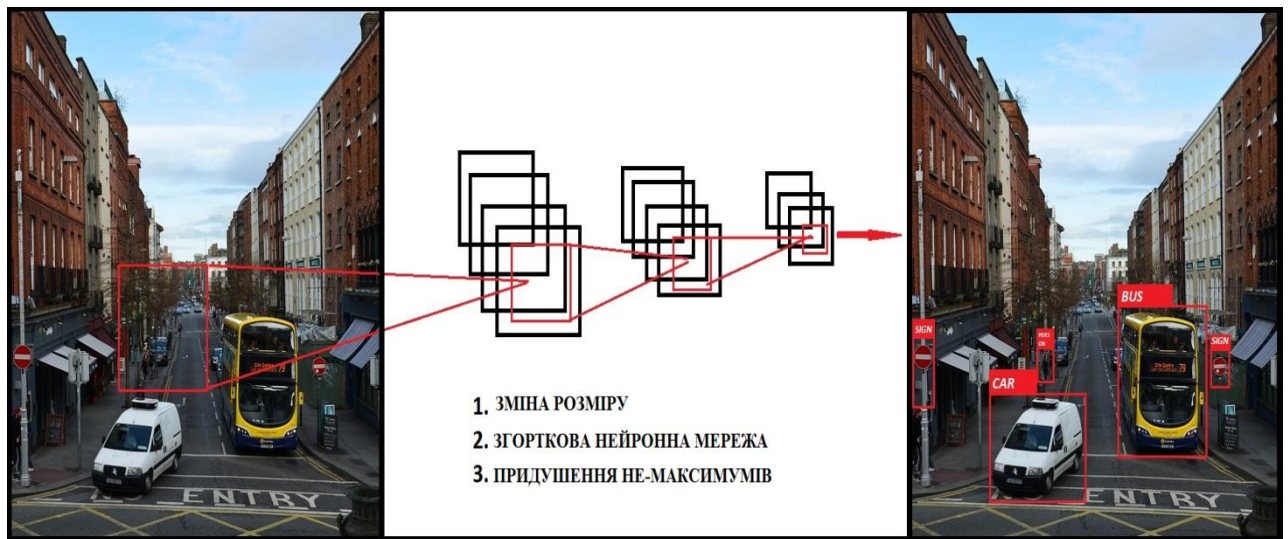


Рисунок 2.13 – Приклад системи виявлення YOLO

Одна згортка мережі одночасно прогнозує кілька обмежувальних прямокутників та ймовірності класів для цих блоків.

Тренування відбувається на повних зображеннях і безпосередньо оптимізується продуктивність виявлення [11]. Ця уніфікована модель має ряд переваг над традиційними методами виявлення об'єктів.

По-перше, YOLO надзвичайно швидкий. Оскільки визначаємо кадр, як проблему регресії, не потрібен складний конвеєр.

По-друге, YOLO міркує глобально про зображення, коли робить прогнози. YOLO бачить ціле зображення під час навчання та тестування, тому

він неявно кодує контекстну інформацію про класи, а також їх зовнішній вигляд.

По-третє, YOLO вивчає узагальнені уявлення об'єктів.

YOLO базується на ідеї сегментації зображення на менші зображення. Зображення розбивається на квадратну сітку розмірами $S \times S$. Клітина, в якій знаходиться центр об'єкта, наприклад, центр собаки, є клітиною, відповідальною за виявлення цього об'єкта.

Даний спосіб продемонстровано на рисунку 2.14.

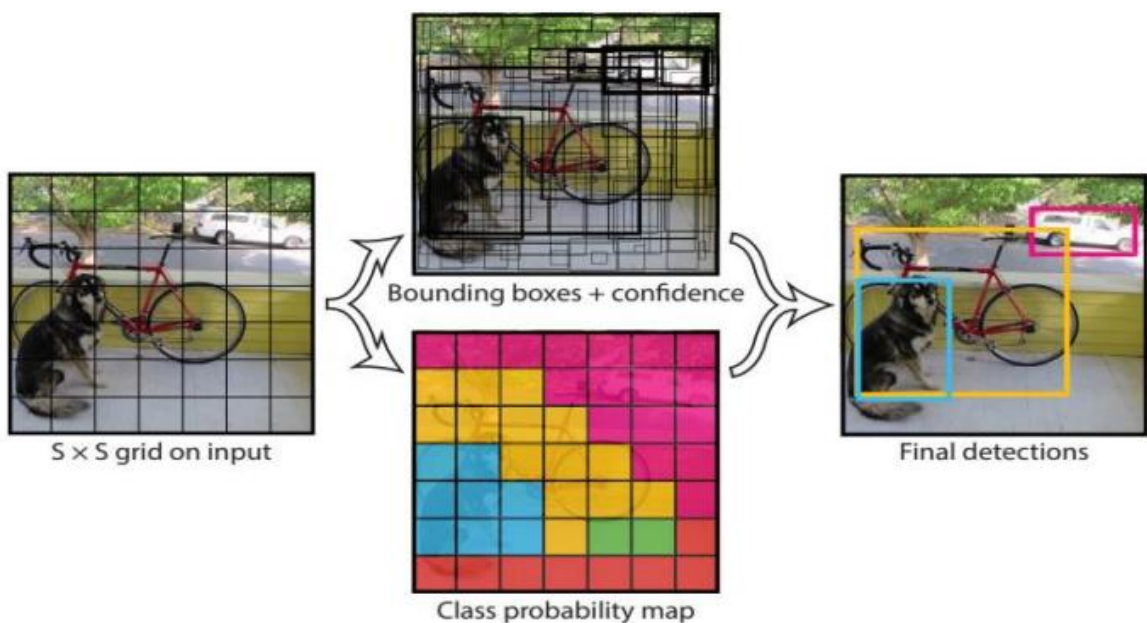


Рисунок 2.14 – Поділ зображення на сітку $S \times S$ і для кожної клітинки сітки передбачає B обмежувальні рамки

Кожен чарунок прогнозує обмежувальні прямокутники, які можна позначити як B , а також оцінку достовірності для кожного прямокутника. За замовчуванням для цієї архітектури модель передбачає дві обмежувальні рамки.

Оцінка класифікації образу буде відбуватись у значеннях від 0,0 до 1,0, де 0,0 – є найнижчим рівнем достовірності, а 1,0 – найвищим. Якщо в цьому чарунку немає жодного об'єкта, оцінка достовірності має бути 0,0, а якщо модель повністю впевнена у своєму передбаченні, оцінка має бути 1,0.

Ці рівні достовірності фіксують впевненість моделі в тому, що в цьому чарунку існує об'єкт і що обмежувальна рамка точна.

Кожна обмежувальна рамка складається з 5 передбачень: x , y , w , h , i – так звана довіра. Координати (x, y) представляють центр квадрата відносно меж клітинки сітки. Ширина і висота прогножуються відносно всього зображення. Прогноз довіри є IOU між прогнозованим боксом і будь-яким наземним боксом істини [12].

Достовірність представляє IOU між передбачуваним обмежувальним прямокутником і фактичним обмежувальним прямокутником, що називається основним полем істинності.

IOU розшифровується як Intersection Over Union і є площею перетину передбачених і базових блоків істинності, поділену на площу об'єднання тих самих передбачених і базових блоків істинності.

Основа модель попередньо натренована на наборі даних класифікації зображень. У наборі конкурентних даних класу ImageNet – 1000 заздалегідь підготовлено 20 із 24 шарів згортки.

Шляхом додаванням слідом до моделі ще 4 згортки, а також 2 повністю з'єднані шари, було показано, що додавання обох згорток і повністю з'єднаних шарів підвищує продуктивність.

Для виявлення більш дрібних деталей збільшилась роздільна здатність з 244×244 до 448×448 пікселів. Останній рівень, який передбачає як імовірності класу, так і координати обмежувальної рамки, використовує функцію лінійної активації для кінцевого шару і всі інші шари використовують наступну лінійну активацію.

Функція втрат є простою квадратичною сумою, але її необхідно модифікувати. Без модифікації модель братиме до уваги помилку локалізації, різницю між прогнозованими та справжніми координатами обмежувальної рамки та помилку передбачення класу.

Функцію втрат можна представити у вигляді формули

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \hat{\Pi}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \hat{\Pi}_{ij}^{obj} \left[(\sqrt{w_i} - \right. \\ & \left. - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^B \hat{\Pi}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \hat{\Pi}_{ij}^{noobj} (C_i - \\ & - \hat{C}_i)^2 + \sum_{i=0}^{S^2} \hat{\Pi}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 . \end{aligned} \quad (2.6)$$

Крім того, коли клітинка сітки не містить об'єкт, її оцінка достовірності наближена до 0, що може перевершити градієнти інших клітинок, які містять об'єкти.

Обидві проблеми вирішуються за допомогою двох коефіцієнтів, λ_{coord} і λ_{noobj} , які множать втрати для координат і втрати об'єкта відповідно.

Для них встановлено значення $\lambda_{coord} = 5$ і $\lambda_{noobj} = 0,5$, що збільшує вагу виявлення та зменшує важливість відсутності втрати об'єкта.

Кожна клітинка сітки передбачає кілька обмежувальних рамок, але лише одна обмежувальна рамка відповідає за виявлення об'єкта.

Відповідальна обмежувальна рамка визначається шляхом вибору прогнозованої обмежувальної рамки з найвищим IOU.

Це відбувається таким чином: якщо існує великий об'єкт, коли кілька обмежувальних рамок прогнозують межі, найкращий отримує винагороду та продовжує покращувати прогнозування великих рамок.

Коли з'являється невеликий об'єкт, попередній об'єкт вже не може передбачити хорошу відповідність, оскільки його обмежувальна рамка завелика.

Однак інший – має кращий прогноз, і він отримує винагороду за вдале обмеження маленького об'єкта.

Подвійна сума означає лише суму всіх клітинок сітки (квадрат $S \times S$) і всіх обмежувальних прямокутників B .

$$\sum_{i=0}^{S^2} \sum_{j=0}^B . \quad (2.7)$$

Квадрат різниці між фактичною координатою x і прогнозованою координатою в клітинці i визначається:

$$(w_i - \hat{w}_i). \quad (2.8)$$

Це повторюється для обох координат x і y , знаходячи квадрат різниці між загальною середньою точкою. Обчислюємо втрати лише для найкращої обмежувальної рамки. Таким чином, перший рядок рівняння є сумою квадратів різниць між передбачуваною та фактичною середніми точками об'єктів у всіх клітинках сітки, які містять об'єкт, і є відповідальною обмежувальною рамкою.

Другий рядок – це сума квадратів різниць квадратних коренів із прогнозованої та фактичної ширини та висоти в усіх клітинках сітки, у яких є об'єкт.

Третій рядок – це лише квадрат різниці між прогнозованими ймовірностями класу та фактичними ймовірностями класу в усіх клітинках, які містять об'єкт.

Четвертий рядок такий самий, але для всіх клітинок, у яких немає об'єкта.

Нарешті, останній рядок – це квадрат різниці між прогнозованим і фактичним класом для всіх чарунків, у яких є об'єкт, що, по суті, просто перевіряє, наскільки далеко була класифікація.

Нарешті, усе це підсумовується, і перші два рядки множаться на коефіцієнт координати, щоб отримати більшу вагу, а четвертий рядок множиться на менший коефіцієнт відсутності об'єктів, щоб зменшити вагу.

Усі моделі YOLO є моделями виявлення об'єктів.

Оригінальний YOLO – YOLO була першою мережею виявлення об'єктів, яка об'єднала проблему малювання обмежуючих прямокутників та ідентифікації міток класів в одній наскрізній мережі, що диференціюється.

YOLOv2 – YOLOv2 зробив ряд ітераційних удосконалень поверх YOLO, включаючи BatchNorm, вищу роздільну здатність та поля прив'язки.

YOLOv3 – YOLOv3 побудований на основі попередніх моделей, додаючи показник об'єктності до передбачення обмежувальної рамки, додаючи підключення до шарів магістральної мережі та роблячи прогнози на трьох окремих рівнях деталізації для підвищення продуктивності на дрібніших об'єктах.

YOLOv4 це двоступінчастий детектор із кількома компонентами. Його розберемо більш детально. YOLOv4 можна реалізувати в будь-якій комбінації магістралі хребта, шиї та голови.

Хребет. Як екстрактори ознак використовуються такі моделі, як ResNet, DenseNet, VGG і т.д. Попередньо навчаються на наборах даних класифікації зображень, таких як ImageNet, та налаштовуються на наборі даних виявлення.

Шия. Це додаткові шари, що проходять між хребтом та головою. Вони використовуються для вилучення різних функцій карт на різних етапах магістралі.

Голова. Це мережа, яка фактично виконує частину виявлення (класифікацію і регресію) прямокутників, що обмежують.

Розрізняють дві категорії методів YOLOv4, які використовуються підвищення точності детектора об'єктів. Вони аналізують різні методи в обох категоріях, щоб отримати нейронну мережу з високою швидкістю роботи та гарною точністю.

Пакет безкоштовних подарунків (BoF або Bag of Freebies). Методи, які можуть збільшити точність детектора об'єктів без збільшення вартості виведення. Ці методи лише змінюють стратегію навчання чи лише збільшують вартість навчання.

Прикладом BoF є збільшення даних, що підвищує здатність моделі до узагальнення.

Для цього можемо робити фотометричні спотворення, такі як: зміна яскравості, насиченості, контрастності та шуму, або можемо робити

геометричне спотворення зображення, обертати його, обрізати і т.д. Ці методи є наочним прикладом VoF і вони допомагають точності детектора.

Пакет спеціальних пропозицій (BoS або Bag of Specials). Ці модулі, що підключаються, і методи постобробки, які лише трохи збільшують вартість виведення, але можуть значно підвищити точність виявлення об'єктів.

Такі модулі, зазвичай, включають: введення механізмів уваги (стиснення, збудження та просторовий модуль уваги), серед іншого, розширення сприйнятливого поля моделі та посилення можливостей інтеграції функцій.

3 РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ МЕТОДІВ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

3.1 Вибір інструментального засобу для реалізації методів розпізнавання дорожніх знаків

Після теоретичного аналізу декількох методів розпізнавання дорожніх знаків для практичної реалізації було обрано два найефективніших – розпізнавання через використання згорткових нейронних мереж та через використання моделі YOLO.

Тому першим етапом вибору інструментального засобу для реалізації методів розпізнавання дорожніх знаків, став вибір інструменту для створення моделей нейронних мереж.

Як правило, для реалізації методів, що відносяться до машинного навчання, переважна більшість дослідників обирають мову програмування Python.

Python – це інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою [53]. Його високорівневі вбудовані структури даних у поєднанні з динамічною типізацією та динамічним зв'язуванням роблять його дуже привабливим для швидкої розробки застосунків, а також для використання як мови сценаріїв.

Простий, легкий для вивчення синтаксис Python підкреслює читабельність і, отже, знижує вартість обслуговування програми. Python підтримує модулі та пакети, що заохочує модульність програми та повторне використання коду.

Інтерпретатор Python і велика стандартна бібліотека доступні у вихідному або двійковому вигляді безкоштовно для всіх основних платформ і можуть вільно поширюватися.

Проекти з використанням нейронних мереж та машинного навчання значно відрізняються від звичайних програм і тому мають свої вимоги до мови

програмування та середовища реалізації. Python, у свою чергу, цілком задовольняє цим вимогам.

Однак, для спрощення побудування нейронної мережі одного Python буде замало, тому було прийнято рішення використати спеціальну бібліотеку, TensorFlow, яка полегшить процес будування та тренування нейронної моделі.

TensorFlow – це відкрита та безкоштовна бібліотека для машинного навчання, що була розроблена компанією Google [54].

TensorFlow можна використовувати в багатьох мовах програмування, включаючи Python, JavaScript, C++ і Java, однак реалізація для Python є найбільш вживаною та підтримується активніше, ніж інші види фреймворку.

Також як допоміжну бібліотеку для TensorFlow часто використовують Keras.

Keras – це бібліотека для мови програмування Python, призначена для глибокого машинного навчання [55]. Вона дозволяє швидше створювати та налаштовувати моделі-схеми, якими поширюється і підраховується інформація під час навчання. Але складних математичних обчислень Keras не виконує та використовується як надбудова над іншими бібліотеками, таких як TensorFlow.

Keras дозволяє швидко побудувати модель, налаштувати її шари, обробити вхід та вихід інформації з моделі, змінити набір даних, що йдуть на вхід моделі, візуалізувати розроблену модель, навчити та протестувати модель, а також провести первинний запуск програми машинного навчання.

Звичайно, все вище наведене можна зробити і без Keras, але за допомогою цієї бібліотеки усі данні етапи робляться в рази швидше та легше, аніж без неї.

Окрім моделей нейронних мереж, що будуть проводити розпізнавання зображень дорожніх знаків, було прийнято рішення додати простий інтерфейс для взаємодії з системою.

Через те, що розробка буде проводитися більше з метою дослідження обраних методів, аніж прямого практичного використання, інтерфейс буде

мати мінімалістичний вигляд. Для його створення будуть обрані вбудовані бібліотеки Python.

Як вже було зазначено раніше, другим методом, що було обрано для дослідження та практичної реалізації є метод з використання моделі YOLO.

Наразі існує декілька версій даної моделі, але для дослідження планується використати модель YOLOv3, тобто третю версію. Вона не є найновішою з тих, що вже існують, але є доволі оптимальною як за швидкістю, так і за точністю.

Порівняно з минулою версією, YOLOv2, третя версія не є швидшою, але в ній було значно покращено точність розпізнавання. У версії YOLOv4 було оптимізовано розпізнавання в режимі real-time, але для даного дослідження питання розпізнавання у реальному часі на відео поки не стоїть.

Саме тому оптимальним рішенням є використання моделі YOLOv3 для її порівняння з результатами роботи звичайної нейронної мережі.

Для навчання та тестування обох моделей необхідно використання вибірки даних. Через те, що робота присвячена розпізнаванню дорожніх знаків України, однією з перших цілей у процесі програмної реалізації стало створення вибірки з українськими дорожніми знаками.

Дорожні знаки України відносяться до європейської групи знаків, тому для власної вибірки були використані дані з декількох вибірок різних країн Європи, які знаходяться у вільному доступі. Здебільшого, вибірка, що була використана в роботі, була створена на основі вибірки дорожніх знаків Німеччини, Франції та Бельгії.

Загалом вибірка ділиться на 43 класи дорожніх знаків та налічує до 50000 зображень. Вибірка вийшла доволі різноманітною. Деякі класи мають більше зображень, ніж інші. Сама вибірка поділена таким чином: 39000 зображень були поділені для тренування, 12000 – для тестування.

Зображення мають різний розмір, різну ступінь освітленості та затемненості, а також різну ступінь наявності шуму та розмиття.

Таким чином, для дослідження обраних методів та їх програмної реалізації були обрані усі вищезазначені інструменти.

3.2 Етапи програмної реалізації методів розпізнавання дорожніх знаків

Першим етапом програмної реалізації є підготовка вибірки даних. Через те, що наразі у відкритому доступі немає вибірки з актуальними українськими дорожніми знаками, першочерговим завданням стає створення вибірки.

Підготовка тестових та навчальних зображень з нуля є дуже складним процесом, що займає дуже велику кількість часу, було прийнято рішення використати вибірки з відкритим доступом та скомбінувати їх між собою.

Дорожні знаки України відносяться до європейського типу знаків, тому мають високу подібність до знаків країн Європи. Для створення вибірки, що містить дорожні знаки України, були використані частини вибірки дорожніх знаків Німеччини, Франції та Бельгії. Для процесу злиття окремих частин вибірок були використані вбудовані функції мови Python.

Більшою мірою була використана вибірка зі знаками Німеччини. В процесі аналізу були виявлені класи зображень, що відрізняються за зовнішнім видом від відповідних українських знаків. Тому зображення для цих класів були видалені та замінені на зображення з інших вибірок, що мають такий само вигляд, як і українські аналоги.

Усього кінцева вибірка містить 43 класи дорожніх знаків. Загальна кількість усіх зображень – близько 50000. Зображення також були поділені на групи для навчання та групи для тестування. 39000 зображень були поділені для тренування, 12000 – для тестування.

Зображення у вибірці мають різні характеристики. Вони відрізняються за ступінню зашумленості, освітленості та затемненості, а також розмиття. Такий вибір зображень зумовлений тим, що у житті дорожні зображення від умов погоди, доби та з плином часу, деформується.

Інколи це впливає навіть на його форму, але найчастіше суттєвих змін зазнає колір.

На рисунку 3.1 приведено приклад зображень з навчальної вибірки.



Рисунок 3.1 – Приклад зображень з навчальної вибірки

Після підготовки вибірки наступним важливим етапом є побудування моделі нейронної мережі. В цьому допоможуть методи бібліотек Keras та TensorFlow.

Код побудування нейронної моделі CNN приведено на рисунку 3.2.

```

model = Sequential()

# First Layer
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu', input_shape=X_train.shape[1:]))
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))

# Second Layer
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))

# Dense Layer
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(rate=0.5))
model.add(Dense(43, activation='softmax'))

```

Рисунок 3.2 – Лістинг коду створення CNN моделі

На початку ініціалізуємо моделі за допомогою конструктора *Sequential()*. Модель *Sequential* або послідовна модель є найпоширенішим видом моделі. Вона підходить для простого стеку шарів, де кожен шар має один вхід та вихід. Конструктор *Sequential()* дозволяє створювати модель шар за шаром.

Далі створюються шари моделі.

Для першого шару обираємо 2D згортку. Цей шар створює ядро згортки, яке згортається із входом шару для отримання тензора виходів. Кількість вихідних фільтрів ставимо, як 32. Для перших шарів буде використовуватися менші значення, ніж для глибших шарів.

Зазвичай починають з фільтрів у діапазоні [32, 64, 128] на початку та збільшують до [256, 512, 1024]. Налаштування, звичайно може відрізнятися в залежності від складності вибірки даних та глибини нейронної мережі.

Max pooling на проміжному шарі додаємо для зменшення просторових розмірів вихідного об'єму. У міру того, як вихідний просторовий об'єм зменшується, кількість вивчених фільтрів зростає — це звичайна практика в розробці архітектур CNN.

Також важливим параметром є *kernel_size*. Він вказує на розміри вікна двовимірної згортки Conv2D.

У якості функції активації взята функція ReLu. Вона буде застосовуватися після операції згортки. Функція активації визначає вихідне значення нейрона в залежності від результату виваженої суми входів та порогового значення.

Шар Dropout випадково встановлює вхідні одиниці на 0 на кожному кроці протягом часу навчання, що допомагає запобігти переналаштуванню.

Вхідні дані, для яких не встановлено значення 0, масштабуються, щоб сума всіх вхідних даних не змінювалася.

Другий основний шар буде мати таку ж архітектуру, але значення фільтрів збільшиться на 64, як це вже було зазначено вище.

Останнім основним шаром буде повнозв'язний шар з 256 вузлами. Нарешті, до мережі додається класифікатор softmax. Результатом цього шару є самі прогнозовані значення.

Остаточний вигляд нейронної мережі показано на рисунках 3.3, 3.4.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	2432
conv2d_1 (Conv2D)	(None, 22, 22, 32)	25632
max_pooling2d (MaxPooling2D)	(None, 11, 11, 32)	0
dropout (Dropout)	(None, 11, 11, 32)	0
conv2d_2 (Conv2D)	(None, 9, 9, 64)	18496
conv2d_3 (Conv2D)	(None, 7, 7, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 3, 3, 64)	0
dropout_1 (Dropout)	(None, 3, 3, 64)	0
flatten (Flatten)	(None, 576)	0
dense (Dense)	(None, 256)	147712
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 43)	11051

Рисунок 3.3 – Список шарів створеної нейронної мережі

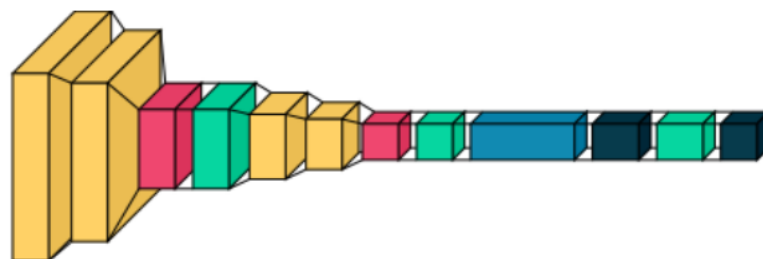


Рисунок 3.4 – Візуалізація шарів нейронної мережі

Після створення та налаштування нейронної мережі йде етап навчання та валідації моделі. Код компіляції моделі приведено на рисунку 3.5.

```

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
epochs = 10
history = model.fit(X_train, y_train, batch_size=64, epochs=epochs, validation_data=(X_test, y_test))
model.save("my_model.h5")

```

Рисунок 3.5 – Лістинг коду компіляції моделі нейронної мережі

Як функцію втрати вибрано *categorical_crossentropy*. Вихідній мітці присвоюється значення кодування однієї категорії у формі 0 та 1. Вихідна мітка, якщо присутня в цілочисельній формі, перетворюється на категоріальне кодування за допомогою бібліотеки Keras.

Як оптимізатор навчання нейронної мережі обрано оптимізатор *Adam*, що є одним з найефективніших з існуючих на даний момент. Він базується на використанні алгоритму градієнтного спуску.

Налаштована модель проходить етап навчання. Процес навчання та пройдені епохи показані на рисунку 3.6.

```

Epoch 1/20
491/491 [=====] - 33s 67ms/step - loss: 5.0469 - accuracy: 0.0412 - val_loss: 3.7701 - val_accuracy: 0.3359
Epoch 2/20
491/491 [=====] - 34s 69ms/step - loss: 3.8104 - accuracy: 0.2786 - val_loss: 2.4422 - val_accuracy: 0.5539
Epoch 3/20
491/491 [=====] - 34s 68ms/step - loss: 2.9014 - accuracy: 0.4360 - val_loss: 1.9032 - val_accuracy: 0.6463
Epoch 4/20
491/491 [=====] - 34s 69ms/step - loss: 2.4352 - accuracy: 0.5235 - val_loss: 1.6446 - val_accuracy: 0.6968
Epoch 5/20
491/491 [=====] - 37s 75ms/step - loss: 2.1539 - accuracy: 0.5820 - val_loss: 1.4584 - val_accuracy: 0.7343
Epoch 6/20
491/491 [=====] - 36s 74ms/step - loss: 1.9443 - accuracy: 0.6229 - val_loss: 1.3502 - val_accuracy: 0.7540
Epoch 7/20
491/491 [=====] - 38s 78ms/step - loss: 1.7936 - accuracy: 0.6541 - val_loss: 1.2665 - val_accuracy: 0.7685
Epoch 8/20
491/491 [=====] - 37s 76ms/step - loss: 1.2734 - accuracy: 0.6196 - val_loss: 0.6696 - val_accuracy: 0.8046
Epoch 9/20
491/491 [=====] - 39s 79ms/step - loss: 0.9789 - accuracy: 0.6972 - val_loss: 0.4302 - val_accuracy: 0.8639
Epoch 10/20
491/491 [=====] - 39s 80ms/step - loss: 0.7710 - accuracy: 0.7607 - val_loss: 0.3750 - val_accuracy: 0.8877

```

Рисунок 3.6 – Процес навчання створеної нейронної мережі

Модель успішно пройшла 10 епох навчання з 10. Точність валідації моделі сягнула 0,7607.

Результати точності тренування та валідації, а також розрахунки втрат будуть наведені у наступному пункті.

Після успішного навчання моделі нейронної мережі настає етап створення простого графічного інтерфейсу для зручної взаємодії. Для цього

використовуємо існуючі методи створення графічного інтерфейсу мови Python. Код створення наведено на рисунках 3.7, 3.8.

```

top=tk.Tk()
top.geometry('800x600')
top.title('Traffic sign classification')
top.configure(background='#d1e1d9')

label=Label(top,background='#CDCDCD', font=('arial',15,'bold'))
sign_image = Label(top)

def classify(file_path):
    global label_packed
    image = Image.open(file_path)
    image = image.resize((30,30))
    image = numpy.expand_dims(image, axis=0)
    image = numpy.array(image)
    pred = model.predict([image])
    cl = np.argmax(pred,axis=1)[0]
    print(pred)
    print(cl)
    sign = classes[cl+1]
    print(sign)
    label.configure(foreground='#011638', text=sign)

def show_classify_button(file_path):
    classify_b=Button(top,text="Classify Image",command=lambda: classify(file_path),padx=10,pady=5)
    classify_b.configure(background='#364156', foreground='white',font=('arial',10,'bold'))
    classify_b.place(relx=0.79, rely=0.46)

def upload_image():
    try:
        file_path=filedialog.askopenfilename()
        uploaded=Image.open(file_path)
        uploaded.thumbnail(((top.winfo_width()/4.25),(top.winfo_height()/4.25)))
        im=ImageTk.PhotoImage(uploaded)

        sign_image.configure(image=im)
        sign_image.image=im
        label.configure(text='')
        show_classify_button(file_path)
    except:
        pass

```

Рисунок 3.7 – Лістинг коду створення графічного інтерфейсу для взаємодії з моделлю нейронної мережі (початок лістингу)

```

upload=Button(top,text="Upload an image",command=upload_image,padx=30,pady=10)
upload.configure(background='#364156', foreground='white',font=('arial',10,'bold'))

upload.pack(side=BOTTOM,pady=50)
sign_image.pack(side=BOTTOM,expand=True)
label.pack(side=BOTTOM,expand=True)
heading = Label(top, text="Traffic Sign Recognition",pady=20, font=('arial',20,'bold'))
heading.configure(background='#d1e1d9',foreground='#364156')
heading.pack()
top.mainloop()

```

Рисунок 3.8 – Лістинг коду створення графічного інтерфейсу для взаємодії з моделлю нейронної мережі (кінець лістингу)

На цьому робота над першим методом розпізнавання дорожніх знаків завершена.

Другий метод, що використовує модель YOLOv3, буде використовувати той самий графічний інтерфейс. Змінюється лише основа – нейронна мережа та дещо змінюється робота з вибіркою даних.

Через те, що YOLO є вже готовою навченою моделлю, нам не треба самостійно налагоджувати її архітектуру. Все, що потрібно – це завантажити модель з відкритого джерела.

Для запуску моделі необхідно створити конфігураційний файл. Для навчання YOLO потрібна велика кількість зображень. До кожному зображенню додається текстовий файл з розміченими регіонами тренованого класу об'єктів. Також необхідний файл з початковими вагами та файл з системною інформацією, в якій вказані шляхи до зображень та шлях, яким будуть записувати точки відновлення. Точки відновлення являють собою файли wag на певному етапі навчання.

Формат даних, що має записатися й конфігураційний тестовий файл повинен мати такий вигляд: *[object-class] [x]/[image_width] [y]/[image_height] [width]/[image_width] [height]/[image_height]*.

Тож, створюємо конфігураційний текстовий файл (рис. 3.9, 3.10).

```
def compare(v1, v2):
    if v1 > v2:
        vmax, vmin = v1, v2
        return vmax, vmin
    else:
        vmax, vmin = v2, v1
        return vmax, vmin
def convert_labels(z):
    x1 = z['Roi.X1']
    y1 = z['Roi.Y1']
    x2 = z['Roi.X2']
    y2 = z['Roi.Y2']
    size = [z['Height'], z['Width']]
    xmax, xmin = compare(x1, x2)
    ymax, ymin = compare(y1, y2)
    dw = 1./size[1]
    dh = 1./size[0]
    x = (xmin + xmax)/2.0
    y = (ymin + ymax)/2.0
    w = xmax - xmin
    h = ymax - ymin
    x = x*dw
    w = w*dw
    y = y*dh
    h = h*dh
    return x, y, w, h
```

Рисунок 3.9 – Лістинг коду отримання значень для конфігураційного файлу

```

for index, x in train_csv.iterrows():
    name = x['Filename'].replace('.ppm', '.txt')
    labels = convert_labels(x)
    cls = x['ClassId']
    f = open(f'{name}', 'w+')
    f.write(f'{cls} {labels[0]} {labels[1]} {labels[2]} {labels[3]}')
    f.close()

```

Рисунок 3.10 – Лістинг коду створення конфігураційного файлу та занесення в нього отримані значення

Далі кроки реалізації не відрізняються від тих, що були реалізовані при створенні моделі звичайної згорткової мережі.

Слід лише зупинитися на особливостях роботи моделі YOLO.

YOLOv3 ділить вхідне зображення на сітку $S \times S$. Кожна клітинка сітки передбачає лише один об'єкт. YOLO при цьому може виявляти дублікати для одного й того ж самого об'єкта.

Щоб виправити це, YOLO застосовує немаксимальне придушення, щоб видалити дублікати з меншою впевненістю. Блок виявлення з найвищим балом, що перевищує порогове значення, зберігається, а інші блоки видаляються.

YOLOv3 використовує Darknet-53 як функцію-екстрактор.

Тут використовується ідея пропуску з'єднань, щоб допомогти активаціям поширюватися через більш глибокі шари без зменшення градієнта.

Оскільки YOLO розроблений, як багатомасштабний детектор, тому також потрібні функції з кількох масштабів. Таким чином, всі ознаки з останніх трьох залишкових блоків використовуються в подальшому виявленні.

Вектор ознак, отриманий з darknet-53, подається в багатомасштабний детектор. Архітектура darknet-53 складається з рівня conv, який присутній у функції darknetConv, за яким слідує залишкові блоки.

І рівень conv, і залишкові блоки об'єднуються, щоб створити darknet блок, залишкові шари повторюються, як зазначено в архітектурі.

Використовуються з'єднання пропуску, а рівень x_{36} , рівень x_{61} і останній рівень є векторами ознак, які повертаються як вихідні дані darknet мережі.

Функція втрат моделі YOLO була розглянута в попередніх розділах.

Архітектура моделі приведена на рисунку 3.11.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Рисунок 3.11 – Структура моделі YOLOv3

Також необхідно зазначити, що YOLO є доволі вимогливою до оперативної пам'яті та графічного процесору. Порівняно з моделлю звичайної згорткової мережі, модель YOLO навчалася набагато довше та потребувала доволі багато ресурсів (рис. 3.12).

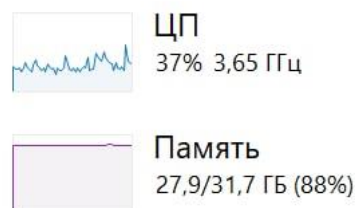


Рисунок 3.12 – Використання ресурсів обчислювальної машини під час процесу тренування моделі YOLO

Модель пройшла всі ітерації процесу навчання у загальній кількості у шість з половиною годин.

3.3 Тестування розроблених застосунків та аналіз результатів

Спершу проаналізуємо роботу першої моделі, яка базується на використанні звичайної згорткової мережі.

Графіки точності тренування та валідації моделі, а також графік втрат моделі наведені на рисунках 3.13, 3.14.

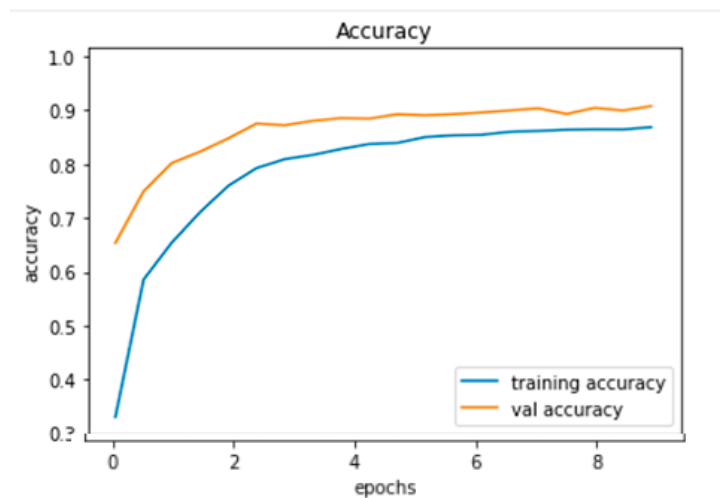


Рисунок 3.13 – Графік точності тренування та валідації створеної моделі

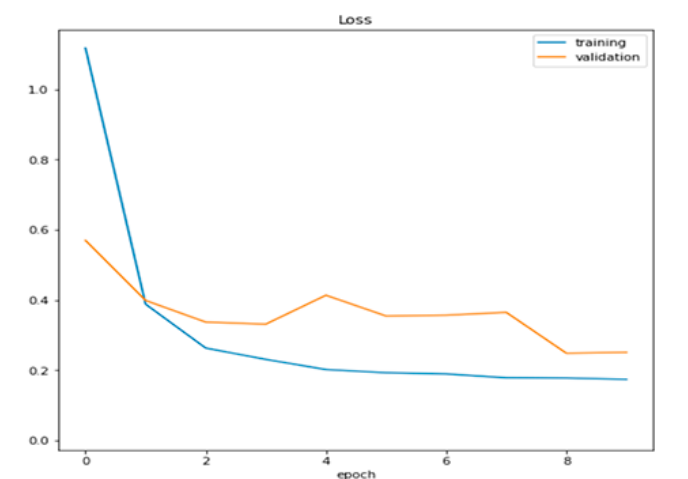


Рисунок 3.14 – Графік втрат тренування та валідації створеної моделі

Під час тестування також були отримані значення точності та втрат моделі. Точність моделі складає 0,7607.

Також візуалізуємо матрицю невідповідностей або матрицю помилок, що дасть можливість унаочнити продуктивність моделі (рис. 3.15).

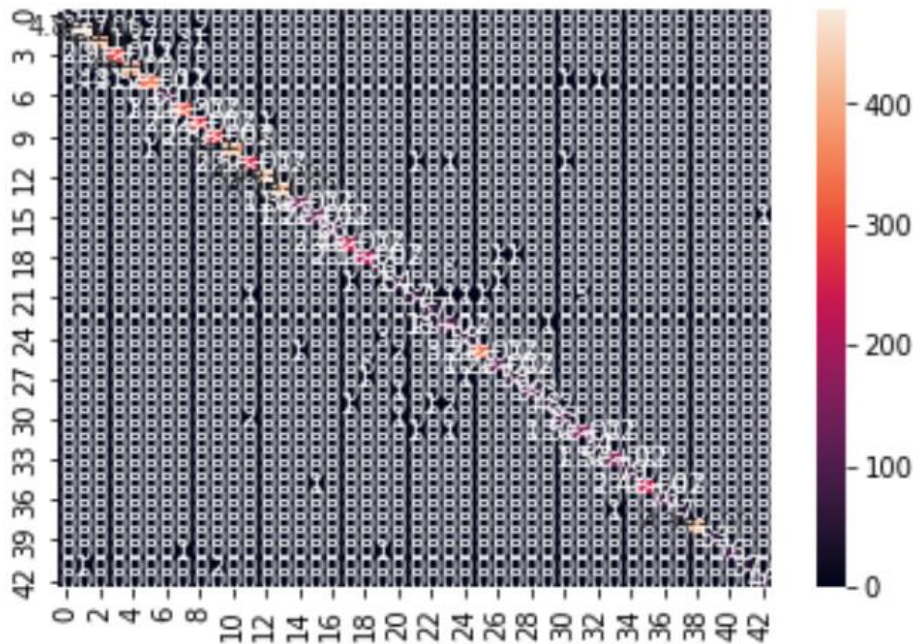


Рисунок 3.15 – Матриця помилок моделі згорткової мережі

Слід зазначити, що хоча точність у 76% – це досить непоганий результат, але для задачі розпізнавання дорожніх знаків, де точність та швидкість відіграють критичну роль, цього результату буде замало для практичного використання.

Крім того, існує також відмінність у якості розпізнавання зображень зблизу та у віддаленні.

Якщо на вхід до моделі подаються зображення дорожніх знаків зблизька, то модель у більшості випадків повертає правильний результат розпізнавання. Але по мірі віддаленості точність стає гіршою.

Також, звісно, точність зменшувалася, якщо знак був прикритий елементами ландшафту, або мав погану освітленість.

Великим випробуванням також ставали зображення, де знак мав деформування, або був сфотографований збоку (рис. 3.16).



Рисунок 3.16 – Приклад хибного розпізнавання знаку, повернутого боком

Модель зовсім не впоралася з розпізнаванням на зображенні з декількома знаками та замість класу хоча б одного зі знаків, повернула помилку (рис. 3.17).



Рисунок 3.17 – Робота моделі згорткової нейронної мережі при наявності декількох знаків на одному зображенні

Якщо ж говорити про роботу моделі YOLOv3, то попри надто довгий процес навчання та велике навантаження на можливості обчислювальної машини, мережа показала дещо кращий результат у точності. Модель змогла показати точність у 86%, а також швидше розпізнавання. Усі тести проводилася на однакових зображеннях. Результати порівняння швидкості розпізнавання наведені у таблицях 3.1, 3.2.

Таблиця 3.1 – Результати розпізнавання моделі звичайної згорткової мережі

Клас вхідного зображення	Швидкість розпізнавання	Валідність розпізнавання
Yild	0,63 мс	вірно
Speed limit (20km/h)	0,15 мс	невірно
Speed limit (120km/h)	0,21 мс	вірно
Priority road	0,20 мс	невірно
Roundabout mandatory	0,94 мс	вірно
Road work	0,55 мс	вірно
Turn right ahead	0,69 мс	невірно

Таблиця 3.2 – Результати розпізнавання моделі YOLO

Клас вхідного зображення	Швидкість розпізнавання	Валідність розпізнавання
Yild	0,33 мс	вірно
Speed limit (20km/h)	0,21 мс	вірно
Speed limit (120km/h)	0,13 мс	вірно
Priority road	0,37 мс	невірно
Roundabout mandatory	0,46 мс	вірно
Road work	0,35 мс	вірно
Turn right ahead	0,63 мс	вірно

Як видно з результатів, наведених на таблицях вище, модель YOLOv3 мала вищу точність розпізнавання та в деяких випадках відпрацьовувала значно швидше. Ці дві характеристики, швидкість та точність, є вкрай важливими для розпізнавання дорожніх знаків під час практичного використання моделей.

3.4 Перспективи подальшої роботи

Наразі простежуються декілька перспективних ідей для подальшої роботи над проблемою:

- покращити вибірку даних дорожніх знаків, урізноманітнити зображення дорожніх знаків з різними ступіннями віддаленості, деформації та ракурсами зйомки [56–65];

- розширити кількість класів вибірки, щоб наблизити до актуальної кількості видів дорожніх знаків України;

- продовжити роботу з моделлю YOLO, розглянути більш нові версії, які працюють з розпізнаванням у реальному часі;

- сфокусувати роботу на розпізнавання дорожніх знаків у реальному часі та наблизити роботу до можливості практичного використання [66–79].

ВИСНОВКИ

У рамках кваліфікаційної роботи було досліджено та практично реалізовано методи розпізнавання дорожніх знаків України: метод з використанням згорткових мереж та метод з використання моделі YOLOv3, що попередньо навчена на інших даних та використовує transfer learning.

У ході роботи були вирішені наступні завдання:

- проаналізовано сучасний стан систем розпізнавання дорожніх знаків;
- проведено класифікацію та аналіз існуючих методів розпізнавання дорожніх знаків;
- проаналізовано особливості дорожніх знаків України;
- проведено аналіз літературних джерел щодо апробації результатів розпізнавання дорожніх знаків;
- розглянуто механізм оброблення зображень дорожніх знаків;
- досліджено принципи розпізнавання складових дорожніх знаків;
- розроблено методику розпізнавання дорожніх знаків;
- проведено вибір інструментальних засобів для реалізації методів розпізнавання дорожніх знаків;
- наведено детальний опис етапів програмної реалізації методів розпізнавання дорожніх знаків;
- проведено тестування розроблених застосунків та проаналізовано результати;
- виявлено перспективи подальшої роботи;
- зроблено висновки щодо виконаної роботи.

Мета роботи досягнута.

Результати роботи апробовано у вигляді 2 тез доповідей під час XXX Міжнародної науково-практичної конференції MicroCAD-2022 [80], XXXVI Міжнародної науково-практичної «The main prospects for the development of science in modern life» [81].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Keller, C. G., Sprunk, C., Bahlmann, C., Giebel, J., & Baratoff, G. (2008, June). Real-time recognition of US speed signs. In 2008 IEEE Intelligent Vehicles Symposium (pp. 518-523). IEEE.
2. Nguwi, Y. Y., & Kouzani, A. Z. (2016, June). A study on automatic recognition of road signs. In 2016 IEEE conference on cybernetics and intelligent systems (pp. 1-6). IEEE.
3. Лубко, Д. В., Лубко, Д. В., Шаров, С. В., & Шаров, С. В. (2019). Методи та системи штучного інтелекту: навчальний посібник. Мелітополь: ФОП Однорог ТВ.
4. Shaha, P., Sharma, U., & Pawar, K. Face Recognition Technology.
5. Vitabile, S., Pollaccia, G., Pilato, G., & Sorbello, F. (2005, September). Road signs recognition using a dynamic pixel aggregation technique in the HSV color space. In Proceedings 11th International Conference on Image Analysis and Processing (pp. 572-577). IEEE.
6. Автопілот компанії Tesla. ІТС.UA: a web-site. URL: <https://itc.ua/blogs/avtopilot-tesla-vyhodit-na-novyj-uroven-nachalos-testirovanie-proaktivnoj-funkczii-reagirovaniya-na-signaly-svetoforov-i-dorozhnye-znaki/>(дата звернення:15.10.2022)
7. Koschan, A., & Abidi, M. (2008). Digital color image processing. John Wiley & Sons.
8. Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and trends® in signal processing, 7(3–4), 197-387.
9. Прохоров, В. Г. (2018). Использование сверточных сетей для распознавания рукописных символов.
10. Wu, Y., Qin, X., Pan, Y., & Yuan, C. (2018, July). Convolution neural network based transfer learning for classification of flowers. In 2018 IEEE 3rd international conference on signal and image processing (ICSIP) (pp. 562-566). IEEE.

11. Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded object detection in video. arXiv preprint arXiv:1709.05943.
12. Yan, B., Fan, P., Lei, X., Liu, Z., & Yang, F. (2021). A real-time apple targets detection method for picking robot based on improved YOLOv5. *Remote Sensing*, 13(9), 1619.
13. Zhang, J., Hui, L., Lu, J., & Zhu, Y. (2018, August). Attention-based neural network for traffic sign detection. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 1839-1844). IEEE.
14. Xiaoping, Z., Jiahui, J., Li, Zhonghe, H., & Shida, L. (2021). People's fast moving detection method in buses based on YOLOv5. *Int. J. Sens. Sens. Netw*, 9, 30.
15. Huang, R., Pedoeem, J., & Chen, C. (2018, December). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2503-2510). IEEE.
16. Malik, R., Khurshid, J., & Ahmad, S. N. (2007, August). Road sign detection and recognition using colour segmentation, shape analysis and template matching. In 2007 international conference on machine learning and cybernetics (Vol. 6, pp. 3556-3560). IEEE.
17. Travis, D. (1991). *Effective color displays: Theory and practice* (Vol. 2015). London: Academic press.
18. Liu, Y. H. (2018, September). Feature extraction and image recognition with convolutional neural networks. In *Journal of Physics: Conference Series* (Vol. 1087, No. 6, p. 062032). IOP Publishing.
19. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.

20. Prieto, M. S., & Allen, A. R. (2009). Using self-organising maps in the detection and recognition of road signs. *Image and Vision Computing*, 27(6), 673-683.
21. Gao, X. W., Podladchikova, L., & Shaposhnikov, D. (2003). Application of vision models to traffic sign recognition. In *Artificial neural networks and neural information processing—ICANN/ICONIP 2003* (pp. 1100-1105). Springer, Berlin, Heidelberg.
22. Ellahyani, A., El Ansari, M., Lahmyed, R., & Trémeau, A. (2018). Traffic sign recognition method for intelligent vehicles. *JOSA A*, 35(11), 1907-1914.
23. Chen, Q., Huang, N., Zhou, J., & Tan, Z. (2018, July). An SSD algorithm based on vehicle counting method. In *2018 37th Chinese Control Conference (CCC)* (pp. 7673-7677). IEEE.
24. Bousarhane, B., Bensiali, S., & Bouzidi, D. (2021). Road signs recognition: state-of-the-art and perspectives. *International Journal of Data Analysis Techniques and Strategies*, 13(1-2), 128-150.
25. Tvoroshenko I., and Tkachenko D. (2020) Mechanisms of image classification based on descriptors of local features, Abstracts of IV International Scientific and Practical Conference «Integration of scientific bases into practice» (October 12-16, 2020). Stockholm, Sweden, pp. 443-448.
26. Якимов, П. Ю. (2013). Предварительная обработка цифровых изображений в системах локализации и распознавания дорожных знаков. *Компьютерная оптика*, 37(3), 401-406.
27. Dubey, A. R., Shukla, N., & Kumar, D. (2020). Detection and classification of road signs using HOG-SVM method. In *Smart Computing Paradigms: New Progresses and Challenges* (pp. 49-56). Springer, Singapore.
28. Adam, A., & Ioannidis, C. (2014). AUTOMATIC ROAD-SIGN DETECTION AND CLASSIFICATION BASED ON SUPPORT VECTOR MACHINES AND HOG DESCRIPTORS. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2(5).

29. Yao, C., Wu, F., Chen, H. J., Hao, X. L., & Shen, Y. (2014, October). Traffic sign recognition using HOG-SVM and grid search. In 2014 12th International Conference on Signal Processing (ICSP) (pp. 962-965). IEEE.
30. Overett, G., & Petersson, L. (2011, June). Large scale sign detection using HOG feature variants. In 2011 IEEE intelligent vehicles symposium (IV) (pp. 326-331). IEEE.
31. Hachour O. Pathplanning of Autonomous Mobile robot. International Journal of Systems Applications, Engineering and Development. 2009. 4. P. 178-190.
32. Lian, J., Yin, Y., Li, L., Wang, Z., & Zhou, Y. (2021). Small object detection in traffic scenes based on attention feature fusion. *Sensors*, 21(9), 3031.
33. Bouti, A., Mahraz, M. A., Riffi, J., & Tairi, H. (2020). A robust system for road sign detection and classification using LeNet architecture based on convolutional neural network. *Soft Computing*, 24(9), 6721-6733.
34. Chen, J., Jia, K., Chen, W., Lv, Z., & Zhang, R. (2022). A real-time and high-precision method for small traffic-signs recognition. *Neural Computing and Applications*, 34(3), 2233-2245.
35. Dagan, E., Mano, O., Stein, G. P., & Shashua, A. (2004, June). Forward collision warning with a single camera. In *IEEE Intelligent Vehicles Symposium, 2004* (pp. 37-42). IEEE.
36. Romdhane, N. B., Mliki, H., & Hammami, M. (2016, June). An improved traffic signs recognition and tracking method for driver assistance system. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)* (pp. 1-6). IEEE.
37. Daradkeh Y.I., and Tvoroshenko I. (2020) Technologies for Making Reliable Decisions on a Variety of Effective Factors using Fuzzy Logic, *International Journal of Advanced Computer Science and Applications*, 11(5), pp. 43-50.

38. Yakimov, P. (2015, July). Traffic signs detection using tracking with prediction. In *International Conference on E-Business and Telecommunications* (pp. 454-467). Springer, Cham.
39. Huang, H., Guo, W., & Zhang, Y. (2008, December). Detection of copy-move forgery in digital images using SIFT algorithm. In *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application* (Vol. 2, pp. 272-276). IEEE.
40. Minetto, R., Thome, N., Cord, M., Leite, N. J., & Stolfi, J. (2013). T-HOG: An effective gradient-based descriptor for single line text regions. *Pattern recognition*, 46(3), 1078-1090.
41. Bay, H., Tuytelaars, T., & Gool, L. V. (2006, May). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.
42. Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10), 1615-1630.
43. Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data—with application to face recognition. *Pattern recognition*, 34(10), 2067-2070.
44. Solomatine, D. P., & Shrestha, D. L. (2004, July). AdaBoost. RT: a boosting algorithm for regression problems. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)* (Vol. 2, pp. 1163-1168). IEEE.
45. Ma, S., & Bai, L. (2016, August). A face detection algorithm based on Adaboost and new Haar-Like feature. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (pp. 651-654). IEEE.
46. Wang, Y. Q. (2014). An analysis of the Viola-Jones face detection algorithm. *Image Processing On Line*, 4, 128-148.
47. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, Al-Shamma, O., Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, 8(1), 1-74.

48. Творошенко, И. С. (2010). Анализ процессов принятия решений в интеллектуальных системах. Системи обробки інформації, (2), 248-253.
49. Duan, D., Xie, M., Mo, Q., Han, Z., & Wan, Y. (2010, October). An improved Hough transform for line detection. In 2010 International Conference on Computer Application and System Modeling (ICCSAM 2010) (Vol. 2, pp. V2-354). IEEE.
50. Matas, J., Galambos, C., & Kittler, J. (2000). Robust detection of lines using the progressive probabilistic hough transform. *Computer vision and image understanding*, 78(1), 119-137.
51. Pandey, P., & Kulkarni, R. (2018, August). Traffic sign detection using Template Matching technique. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCCUBEA) (pp. 1-6). IEEE.
52. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання. Міжнародний науковий симпозіум «Інтелектуальні рішення-С». Обчислювальний інтелект (результати, проблеми, перспективи). Теорія прийняття рішень: праці міжн. наук. симпозіуму (Вересень 29, 2021). Київ – Ужгород, С. 44-45.
53. De Smedt, T., & Daelemans, W. (2012). Pattern for python. *The Journal of Machine Learning Research*, 13(1), 2063-2067.
54. Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45(2), 227-248.
55. Ketkar, N. (2017). Introduction to keras. In *Deep learning with Python* (pp. 97-111). Apress, Berkeley, CA.
56. Tvoroshenko, I., & Dziubenko, M. (2020). Modern methods of analysis of the movement scheme using video detection of vehicles.
57. Gorokhovatskyi, V. A., Rusakova, N., & Tvoroshenko, I. S. (2020). The application of image analysis methods and predicate logic in applied problems of magnetic monitoring. *Telecommunications and Radio Engineering*, 79(20).

58. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.

59. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Al-Dhaifallah, M. (2022). Classification of Images Based on a System of Hierarchical Features. *Computers, Materials & Continua*, 72(1), 1785-1797.

60. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ. 124.

61. Ahmad M.A., Gorokhovatskyi V., Tvoroshenko I., Vlasenko N., and Mustafa S.K. (2021) The Research of Image Classification Methods Based on the Introducing Cluster Representation Parameters for the Structural Description, *International Journal of Engineering Trends and Technology*, 69(10), pp. 186-192.

62. Tvoroshenko I., and Zarivchatskyi R. (2020) Analysis of existing methods for searching object in the video stream, Abstracts of VI International Scientific and Practical Conference «About the problems of science and practice, tasks and ways to solve them» (October 26-30, 2020). Milan, Italy, pp. 500-505.

63. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. *Сучасні інформаційні системи*, 6(3), С.5–12.

64. Tvoroshenko, I., & Babochkin, O. (2021). Object identification method based on image keypoint descriptors.

65. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73 (3), pp. 6069–6084.

66. Lyashenko V., Mustafa S.K., Tvoroshenko I., and Ahmad M.A. (2020) Methods of Using Fuzzy Interval Logic During Processing of Space States of Complex Biophysical Objects, *International Journal of Emerging Trends in Engineering Research*, 8(2), pp. 372-377.

67. Matarneh Rami, Tvoroshenko Irina, and Lyashenko Vyacheslav (2019) Improving Fuzzy Network Models For the Analysis of Dynamic Interacting

Processes in the State Space, *International Journal of Recent Technology and Engineering*, 8(4), pp. 1687-1693.

68. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Modification of the branch and bound method to determine the extremes of membership functions in fuzzy intelligent systems, *Telecommunications and Radio Engineering*, 78(20), pp. 1857-1868.

69. Tvoroshenko, I. (2019). Development of models of spatial analysis of status of interactive processes of complex systems.

70. Tvoroshenko, I. S., & Kramarenko, O. O. (2019). Software determination of the optimal route by geoinformation technologies. *Radio Electronics, Computer Science, Control*, (3), 131-142.

71. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.

72. Кучеренко, Є. І., Творошенко, І. С., Анопрієнко, Т. В. (2016) Моделювання та оцінювання станів складних об'єктів із застосуванням формальної логіки. *Системи обробки інформації*, (2), 76-82.

73. Кучеренко, Е. И., Филатов, В. А., Творошенко, И. С., & Байдан, Р. Н. (2005). Интеллектуальные технологии в задачах принятия решений технологических комплексов на основе нечеткой интервальной логики. *Восточно-Европейский журнал передовых технологий*, (2), 92-96.

74. Творошенко, І. С., & Табашник, В. А. (2018). Розробка просторової моделі геоінформаційної підтримки людей з обмеженими можливостями, що пересуваються на інвалідних колясках, у місті Харків. *Збірник наукових праць Харківського національного університету Повітряних Сил*, (1), 122-128.

75. Кучеренко, Е. И., Творошенко, И. С. (2010) Прикладные аспекты моделирования нечетких процессов в сложных системах. *Збірник наукових праць Харківського університету Повітряних сил*, (1), 127-131.

76. Кучеренко, Е. И., Корниловский, А. В., Творошенко, И. С. (2010) О методах настройки функций принадлежности в нечетких системах. *Системы управления, навигации и связи*, (1), 13.

77. Творошенко, І. С. (2018). Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем. *Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland*, 118-121.

78. Творошенко, І. С., Шевченко, А. Р. (2018) Удосконалення просторової мережі навчальних закладів міста Сєвєродонецька на основі геоінформаційного аналізу. *Системи обробки інформації*, (1), 46-52.

79. Asaad Ma. Babker, Abd Elgadir A. Altoum, Irina Tvoroshenko, and Vyacheslav Lyashenko (2019) Information Technologies of the Processing of the Spaces of the States of a Complex Biophysical Object in the Intellectual Medical System HEALTH, *International Journal of Advanced Trends in Computer Science and Engineering*, 8(6), pp. 3221-3227.

80. Темчур, К. (2022). Development of a road sign recognition system in Ukraine.

81. Темчур, К. (2022). Аналіз існуючих методів розпізнавання дорожніх знаків. THE MAIN PROSPECTS FOR THE DEVELOPMENT OF SCIENCE IN MODERN LIFE, 36, 367.