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

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«___»_____20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі системи та технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Програмна система для розпізнавання дорожніх знаків з використанням машинного навчання»

Виконав (-ла):
студент (-ка) IV курсу, групи ІС-93
Мулік Рустам Юрійович _____

Керівник:
Доцент кафедри ІСТ, д.т.н., доцент,
Поліщук Михайло Миколайович _____

Рецензент:
Доцент кафедри ТВЛА, к.т.н., доцент,
Горностай Вадим Миколайович _____

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2023 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма – «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___»_____20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Муліку Рустаму Юрійовичу

1. Тема проєкту «Програмна система для розпізнавання дорожніх знаків з використанням машинного навчання», керівник проєкту Поліщук Михайло Миколайович, д.т.н., доцент, затверджені наказом по університету від «31» травня 2023 р. №2101-с.

2. Термін подання студентом проєкту: 12 червня 2023 року.

3. Вихідні дані до проєкту: мова програмування Python, середовище розробки PyCharm, середовище розробки Jupyter Notebook.

4. Зміст пояснювальної записки: аналіз предметної області розпізнавання дорожніх знаків, огляд існуючих сучасних алгоритмів розпізнавання дорожніх знаків, вибір засобів для розробки системи розпізнавання дорожніх знаків, опис програмної реалізації.

5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо): діаграма варіантів використання програмної системи, діаграма алгоритму роботи програмного забезпечення, діаграма алгоритму Віоли-Джонса для розпізнавання дорожніх знаків, діаграма алгоритму опорних векторів для розпізнавання дорожніх знаків.

6. Дата видачі завдання: 28 лютого 2023 року.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Ознайомлення з технічним завданням	21.04.2023	
2	Аналіз предметної області	28.04.2023	
3	Розробка функціональної моделі	05.05.2023	
4	Формування цілей та задач розробки	13.05.2023	
5	Розробка програмного забезпечення	19.05.2023	
6	Виконання графічних документів	21.05.2023	
7	Оформлення ПЗ	21.05.2023	
8	Подання ДП на попередній захист	29.05.2023	
9	Подання ДП на основний захист	12.06.2023	

Студент Рустам МУЛІК

Керівник Михайло ПОЛІЩУК

АНОТАЦІЯ

Мулік Р. Ю. Програмна система для розпізнавання дорожніх знаків з використанням машинного навчання. КПІ ім. Ігоря Сікорського, Київ, 2023.

Проєкт містить 61 с. тексту, 21 рисунків, посилання на 17 літературних джерел та 4 конструкторських документа.

Ключові слова: OpenCV, машинне навчання, комп’ютерний зір, нейронні мережі, немаксимальне придушення, дорожні знаки.

Об'єктом розробки є система розпізнавання дорожніх знаків для автоматизації процесу виявлення та класифікації знаків на дорозі.

Мета розробки полягає у поліпшенні безпеки дорожнього руху шляхом швидкого та точного розпізнавання різних типів дорожніх знаків.

Розроблена система розпізнавання дорожніх знаків надає можливість автоматично виявляти знаки на відео або зображеннях, а потім класифікувати їх відповідно до їхнього значення. Це дозволяє водіям отримувати важливу інформацію про обмеження швидкості, правила перевезення та інші аспекти.

Отримані результати досліджень та розробки системи розпізнавання дорожніх знаків можуть бути корисними для різних застосувань. Вони можуть використовуватись у системах автономного водіння для автоматичного розпізнавання дорожніх знаків, у системах моніторингу дорожнього руху для контролю виконання правил, а також у системах навігації. Система використовує нейронні мережі, зокрема методи згорткових нейронних.

SUMMARY

Mulik R. Y. Software system for recognizing traffic signs using machine learning. Igor Sikorsky KPI, Kyiv, 2023.

The project contains 61 pages. text, 21 figures, references to 17 literary sources and 4 design documents.

Keywords: OpenCV, machine learning, computer vision, neural networks, non-maximal suppression, traffic signs.

The object of development is a road sign recognition system for automating the process of identifying and classifying road signs.

The purpose of the development is to improve road safety by quickly and accurately recognizing different types of road signs.

The developed road sign recognition system provides the ability to automatically detect signs in videos or images and then classify them according to their meaning. This allows drivers to receive important information about speed limits, traffic rules and other aspects.

The obtained results of research and development of the road sign recognition system can be useful for various applications. They can be used in autonomous systems for automatic recognition of road signs, in traffic monitoring systems to control the enforcement of rules, as well as in navigation systems. The system uses neural networks, in particular methods of convolutional neural networks.

Номер	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			Документація загальна			
2						
3			Знову розроблена			
4						
5	А4	ІС93.0150БАК.004 ПЗ	Пояснювальна записка	61		
6	А3	ІС93.0150БАК.004 Д1	Програмна система для розпізнавання	1		
7			дорожніх знаків з використанням			
8			машинного навчання. Діаграма			
9			варіантів використання програмної			
10			системи			
11	А3	ІС93.0150БАК.004 Д2	Програмна система для розпізнавання	1		
12			дорожніх знаків з використанням			
13			машинного навчання. Діаграма			
14			алгоритму роботи програмного			
15			забезпечення			
16	А3	ІС93.0150БАК.004 Д3	Програмна система для розпізнавання	1		
17			дорожніх знаків з використанням			
18			машинного навчання. Діаграма			
19			алгоритму Віоли-Джонса для			
20			розпізнавання дорожніх знаків			
21	А3	ІС93.0150БАК.004 Д4	Програмна система для розпізнавання	1		
22			дорожніх знаків з використанням			
23			машинного навчання. Діаграма			
24			алгоритму опорних векторів для			
25			розпізнавання дорожніх знаків			
26						
27						
28						

Зм.	Аркуш	№ докум.	Підпис	Дата	ІС93.150БАК.004 ТП		
Розроб.		Мулік Р. Ю.			Програмна система для розпізнавання дорожніх знаків з використанням машинного навчання. Відомість проєкту	Літ.	Аркуш 1 / Аркушів 1
Керівн.		Поліщук М. М.				Т	
						КПІ ім. Ігоря Сікорського Група ІС-93	
Затв.							

Пояснювальна записка
до дипломного проєкту
на тему: «Програмна система для розпізнавання дорожніх знаків з використанням машинного навчання»

Київ – 2023

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ 4

ВСТУП..... 5

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ.... 7

1.1 Огляд предметної області дослідження..... 7

1.2 Аналіз існуючих підходів до вирішення задачі розпізнавання дорожніх знаків 8

1.2.1 Метод порівняння з шаблоном..... 14

1.2.2 Метод лінійної дискримінантної класифікації 14

1.2.3 Метод головних компонент 15

1.2.4 Метод структурно-символьної локалізації..... 16

1.2.5 Метод адаптованого полішення 17

1.2.6 Метод прискорених стійких ознак..... 19

1.3 Огляд існуючих рішень 20

1.3.1 Opel Eye Assistant..... 20

1.3.2 Mercedes Speed Limit Assistant..... 21

1.3.3 Ford Speed Sign Detection 22

1.4 Постановка задачі..... 23

2 ОГЛЯД ІСНУЮЧИХ СУЧАСНИХ АЛГОРИТМІВ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ..... 25

2.1 Метод опорних векторів..... 25

2.2 Метод на основі згорткової нейронної мережі 28
2.3 Алгоритм Віоли-Джонса 32
2.4 Алгоритм використання комп’ютерного зору для знаходження та розпізнавання дорожніх знаків 35
3 ВИБІР ЗАСОБІВ ДЛЯ РОЗРОБКИ СИСТЕМИ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ 42
3.1 Мова програмування Python 42
3.2 Бібліотека OpenCV 43
3.3 Бібліотека Tensorflow 44
3.4 Бібліотека Keras 45
3.5 Бібліотека TorchVision 46
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ 47
4.1 Збір навчальних даних та їх нормалізація 47
4.2 Архітектура розробленої нейронної мережі 49
4.3 Налаштування та метрики тестування моделі 53
4.4 Навчання нейронної мережі 54
4.5 Тестування та інтеграція моделі нейронної мережі 55
Висновки до розділу 58
ВИСНОВКИ 59
ПЕРЕЛІК ПОСИЛАНЬ 60

ПЕРЕЛІК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ADAS (Advanced driver-assistance systems) – удосконалена система допомоги водію.

LBP (Local binary pattern) – локальний бінарний шаблон

CNN (Convolution Neural Network) – згорткова нейронна мережа.

GTSRB (German Traffic Sign Recognition Benchmark) – набір даних для тренування моделей розпізнавання дорожніх знаків.

MSER (Maximally stable extremal regions) – максимально стабільні екстремальні області.

ШНМ (Штучні нейронні мережі) – це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин.

LDA (Linear discriminant analysis) – статистичний метод для розв'язку задачі класифікації за допомогою якого будуються лінійні комбінації предикторів, що відділяють області одного класу від іншого.

PCA (Principal component analysis) – метод факторного аналізу в статистиці, який використовує ортогональне перетворення множини спостережень з можливо пов'язаними змінними у множину змінних без лінійної кореляції, які називаються головними компонентами.

SIFT (Scale-invariant feature transform) – це алгоритм комп'ютерного бачення для виявляння, описування та зіставляння локальних ознак.

SURF (Speeded-Up Robust Feature) – це запатентований виявляч та описувач локальних ознак, який частково натхнений описувачем масштабоінваріантного ознакового перетворення (SIFT).

ReLU (Rectified linear unit) – передавальна функція.

RGB (Red, Green, Blue) – колірна модель.

HSL (Hue, Saturation, Lightness) – альтернативна кольорова фільтрація.

ВСТУП

З кожним роком кількість транспортних засобів на дорогах зростає, що призводить до збільшення частоти дорожньо-транспортних пригод. Це підкреслює нагальну потребу вирішити питання безпеки дорожнього руху. Загальновідомо, що у більшості аварій вирішальну роль відіграє «людський фактор». Часто водії не реагують на пішохода, який несподівано переходить дорогу, тому що водій міг пропустити попереджувальний знак або сигнал. Як вирішення цієї проблеми автовиробники все більше наголошують на комп'ютеризації процесів водіння. Комп'ютерні системи зі спеціалізованими функціями розробляються та інтегруються, щоб допомогти водіям у виконанні їх завдань. Ці системи спрямовані на підвищення безпеки водія та зниження ризику аварій на дорозі. Система розпізнавання дорожніх знаків – це технологія, яка використовується великими виробниками автомобілів, такими як Audi, BMW, Ford, Mercedes-Benz, Opel і Volkswagen.

Однак більшість із цих систем мають певні недоліки. Наприклад, деякі системи можуть розпізнавати лише знаки обмеження швидкості, а інші не є універсальними і не можуть бути встановлені в автомобілях інших марок. Крім того, ці системи часто є досить дорогими та обмежено сумісними з іншими транспортними засобами, тому потреба в доступних та універсальних системах розпізнавання дорожніх знаків все ще є гострою. Коротше кажучи, незважаючи на те, що ці системи широко використовуються, вони мають кілька недоліків, через які вони не є ідеальними для багатьох користувачів, і потрібні вдосконалення, щоб зробити їх доступнішими та зручнішими.

Наразі безпілотні автомобілі активно тестуються, і в найближчому майбутньому можна очікувати, що ці автомобілі стануть звичайним явищем на дорогах. Однак для забезпечення безпеки цих автономних транспортних засобів важливо контролювати дотримання правил дорожнього руху з максимальною точністю. У цьому відношенні розробка та інтеграція високоякісних

автоматизованих систем, які можуть ефективно контролювати правила дорожнього руху, є вирішальною.

Незважаючи на значні потенційні переваги таких систем, варто зазначити, що існуючі є дорогими та не мають широкого застосування. Тому існує нагальна потреба в проектуванні та розробці альтернативної системи, яка може бути економічно ефективною та легкодоступною для всіх водіїв.

Метою дослідження є розробка алгоритмів детектування та класифікації дорожніх знаків з використанням нейронних мереж.

Для досягнення поставленої мети були сформульовані наступні завдання:

– проаналізувати предметну області розробки та існуючі аналоги;

– розглянути основні методи та алгоритми розпізнавання дорожніх знаків на зображеннях та відео;

– обрати технології розробки, що задовольняли б усі поставлені вимоги та забезпечували кращий результат;

– розробити програмну систему розпізнавання дорожніх знаків з використанням комбінованого методу класифікації.

Об’єктом дослідження є процес застосування нейромережевих алгоритмів для виявлення та розпізнавання дорожніх знаків.

Предметом дослідження є алгоритми комп’ютерного зору для обробки зображень та системи, створені з використанням нейронних мереж.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

1.1 Огляд предметної області дослідження

Процес розпізнавання дорожніх знаків на перший погляд може здатися простим, але насправді він включає кілька методів і підходів. Як правило, камера знімає поле зору водія, де можуть бути дорожні знаки, а потім ця інформація передається в систему розпізнавання [1].

Система може використовувати кілька методів розпізнавання дорожніх знаків, включаючи, але не обмежуючись, порівняння з попередньо визначеним шаблоном, еластичні стандарти порівняння та класифікацію за допомогою нейронних мереж. Незважаючи на те, що дорожні знаки створені, щоб бути видимими та легко зрозумілими, все ще існують проблеми, які ускладнюють їх виявлення та розпізнавання, особливо за несприятливих погодних умов, таких як дощ, сніг, туман та погане освітлення. Крім того, важливо враховувати, що знак може бути пошкоджений або бути прихованим навколишніми предметами. Таким чином, вибір найбільш ефективного підходу, який ефективно працює за різних обставин, стає вирішальним [2].

Сучасні системи розпізнавання дорожніх знаків мають схожу структуру, що включає відеокамеру, блок управління та екран. Операційний блок, який є процесором або мікроконтролером, виконує всі обчислювальні операції, пов’язані з розпізнаванням дорожніх знаків на фото, знятих камерою. Ці системи функціонують так, що при розпізнаванні знака інформація виводиться на екран і може супроводжуватися звуковим сигналом [2, 4]. Описаний тип системи, який допомагає водіям приймати обґрунтовані рішення на дорозі, зазвичай доступний лише в автомобілях високого класу, таких як BMW 5-ї, 6-ї та 7-ї серії, Mercedes E класу та вище та Opel Insignia. Навіть для цих розкішних автомобілів система зазвичай пропонується як дорогий варіант, який постачається в комплекті з навігаційною системою. Ці системи встановлюються в автомобілі і коштують від

1000 до 4000 євро. Крім того, доступні портативні пристрої, які оснащені можливостями розпізнавання дорожніх знаків, наприклад, комп'ютер-навігатор Blaupunkt TravelPilot 700, який може коштувати від 600 до 700 євро. Загалом можна зробити висновок, що інтелектуальні системи допомоги водієві мають значну вартість, що є одним із основних факторів, що обмежують їх широке впровадження [5].

Хоча виробники в усьому світі досягли значного прогресу в розробці інтелектуальних систем безпеки дорожнього руху, їх здатність розпізнавати дорожні знаки залишається значно обмеженою. Деякі системи можуть точно ідентифікувати невеликий набір знаків обмеження швидкості, збільшення кількості розпізнаних знаків призведе до значних обчислювальних витрат і часу відповіді системи, що знизить ефективність системи. Численні експериментальні випробування [6, 7] показали, що однією з головних проблем у розробці систем технічного зору є точне виявлення об'єктів на складному фоні. Це означає, що системи можуть помилково виявити рекламний щит або інші подібні об'єкти замість дорожнього знака. Таким чином, розробка систем розпізнавання об'єктів, здатних ефективно функціонувати в складних середовищах, залишається ключовим викликом.

1.2 Аналіз існуючих підходів до вирішення задачі розпізнавання дорожніх знаків

Наразі існує три основні підходи до виявлення та визначення місцезнаходження дорожніх знаків, а саме використання інформації про колір, інформацію про форму та їх поєднання. Колір знаків допомагає відрізнити їх від навколишнього середовища на дорозі. Щоб виявити ознаки на основі кольору, у колірному просторі RGB часто використовується техніка, яка називається кольоровою фільтрацією, як зазначено в [6]. Однак у певних ситуаціях колірні простори HSI або HSV можуть бути більш придатними. Альтернативним підходом

для розпізнавання дорожніх знаків є використання представлення кольору зображення в просторі HSI. Ця техніка передбачає порівняння відтінку (H - компонент) кожного пікселя з еталонним відтінком і виділення контурів дорожніх знаків на зображенні за допомогою граничних значень. Крім того, для виявлення об’єктів використовуються методи граничних функцій, де ключовим принципом є обчислення відстані між кольором пікселя та еталонним кольором у просторі RGB. Якщо обчислена відстань досить мала, піксель вважається частиною об’єкта, що виявляється.

Підходи на основі кольорової фільтрації для виявлення знаків не потребують значних обчислювальних ресурсів. Однак їхня точність не є задовільною через ряд обмежень. Ці обмеження включають труднощі з виявленням ознак за поганих погодних умов, таких як дощ, сніг або туман. Крім того, такі підходи не в змозі виявити дорожні знаки в ситуаціях недостатнього освітлення, оскільки більшість алгоритмів не розроблені для обробки таких сценаріїв. Також, ці способи неефективні, коли частина вивіски знаходиться в тіні, а інша – яскраво освітлена. Незважаючи на ці обмеження, підходи на основі кольорової фільтрації все ще широко використовуються для виявлення знаків у певних програмах, де висока точність не є критичною [5].

Збільшення обчислювальної потужності сучасних систем дозволило використовувати більш складні алгоритми для виявлення об'єктів у реальному часі. Ці алгоритми часто покладаються на отримання інформації про форму об’єкта, щоб точно його ідентифікувати.

Зазвичай це робиться за допомогою методів виділення контурів, таких як метод Канні або перетворення Хафа. Площа та положення об’єкта, який часто є знаком, аналізуються, щоб виявити його форму. Один із таких алгоритмів базується на обчисленні тангенса нахилу дотичної до об’єкта. Завдяки цьому він здатний визначити, чи це коло, прямокутник чи трикутник, навіть у ситуаціях із сильними шумовими перешкодами.

У багатьох алгоритмах виявлення форма знака є важливим фактором, який слід враховувати, і це робиться шляхом обчислення набору ознак із зображення. Існує кілька способів формування такого набору, і одним із найефективніших методів є алгоритм Scale Invariant Feature Transformation (SIFT). Алгоритм має кілька важливих характеристик, таких як незмінність до змін освітленості, шумів у зображеннях, геометричних змін зображення, а також змін розміру зображення та об’єкта.

Завдяки цим властивостям він добре підходить для процедур автоматичного вибору функцій, особливо коли йдеться про розпізнавання дорожніх знаків. Алгоритм містить чотири основні послідовності: виявлення екстремуму в просторі великомасштабних перетворень, визначення ключових точок, визначення правильної орієнтації зображення та генерування ознак.

Алгоритм SIFT використовує евклідову відстань у векторному просторі ознак, щоб знайти зразок, який відповідає зображенню знака, що аналізується. Цим методом вибирається найбільш близький до поточного зображення знака зразок. Однак, коли вибірок багато або простір ознак має високу розмірність, цей метод стає неефективним. Окрім SIFT, існують інші способи автоматичного створення наборів функцій, наприклад використання коефіцієнтів вейвлет-перетворення Хаара як функцій. Потім для класифікації використовується алгоритм AdaBoost, де кожен наступний класифікатор навчається з використанням векторів, використаних попередніми класифікаторами, які неправильно класифікували певні вектори [7, 8].

Методи визначення дорожніх знаків, засновані на формі, можуть бути дорогими з точки зору обчислень і можуть важко працювати в сценаріях реального часу. Однією з головних проблем із цим підходом є те, що він також може виявляти інші об’єкти зі схожими геометричними формами, як дорожні знаки (наприклад, рекламні щити чи вікна), що призводить до хибних спрацьовувань. Крім того, він також може пропустити пошкоджені або завалені дорожні знаки, які дещо відхиляються від очікуваної форми.

Методи, які одночасно використовують інформацію про колір і форму, широко використовуються в різних програмах. Один із таких зразкових підходів передбачає використання компонентів відтінку та насиченості з колірного простору HSL. Щоб забезпечити оптимальну продуктивність, для гістограм відтінку та насиченості використовуються динамічні пороги. Виконуючи логічне додавання значень відтінку та насиченості, отримують високоефективні бінарні зображення. Крім того, результуюче бінарне зображення сегментується на сім різних типів на основі різних цільових пікселів та їхніх сусідніх пікселів. Потім ці різні типи шаблонів використовуються для точного визначення форми дорожніх знаків.

У процесі виявлення дорожніх знаків система використовує метод, який передбачає аналіз кількох кадрів. Система починає з виявлення зображення, яке потенційно може бути знаком, використовуючи алгоритми, подібні до аналізу оптичного потоку. Якщо межі дорожнього знака не змінюються протягом кількох послідовних кадрів, система вибирає його для класифікації. Щоб відстежувати знак-кандидат, система може використовувати фільтр Калмана, який передбачає положення знака в кожному кадрі.

Завдяки цьому система може зменшити кількість часу, який вона витрачає на аналіз областей кадру, які не містять знака [8].

Базову нейронну мережу також можна використовувати для розпізнавання форми та кольору знака. Вихідні кольорові фотографії спочатку обробляються за допомогою лапласіана фільтра Гаусса. Потім спеціально навчені нейронні мережі використовуються для сегментації об’єктів на основі кольору та форми. Коли знак успішно розпізнається, виконується зіставлення шаблону для розпізнавання. Виходячи з вищевикладеного, можна передбачити, що використання інтегрованої інформації щодо кольору та форми значно підвищить якість та надійність виявлення дорожніх знаків. Нейронні мережі довели свою ефективність у системах безпеки автомобіля, зокрема в розпізнаванні дорожніх знаків. Серед моделей нейронних мереж зазвичай використовуються багатошарові персептрони завдяки

їхній здатності безпосередньо використовувати дані зображення як набір для навчання, усуваючи необхідність попереднього формування векторів ознак. Це істотно скорочує час, необхідний для розпізнавання, оскільки більша частина часу йде на навчання нейронної мережі на початкових етапах. Однак одним із головних недоліків використання нейромережевого підходу для розпізнавання дорожніх знаків є непередбачуваність його результатів, що може бути наслідком великої варіативності помилки розпізнавання [10].

Іншим методом розпізнавання дорожніх знаків є зіставлення за зразком. Цей метод є високоефективним завдяки своїй простоті та швидкості виконання. Однак під час використання цього методу життєво важливо точно вибрати внутрішню область дорожнього знака, щоб порівняти її з еталонним зображенням. Тому стадія виявлення ознак є важливою у всіх існуючих підходах. Метою є не уникнути цього етапу, а підвищити точність, швидкість і надійність. Штучні нейронні мережі (ШНМ) довели свою ефективність у розпізнаванні дорожніх знаків. Однак класичні ШНМ чутливі до спотворень зображення та сильно піксельованих зображень, що може призвести до громіздких структур, що призводить до збільшення часу роботи. Згорткові нейронні мережі (CNN) вважаються найкращим варіантом завдяки їхнім ефективним можливостям обробки. CNN забезпечують багаторівневий підхід для розпізнавання дорожніх знаків, використовуючи спільні ваги та згортки, надаючи швидший і надійний метод розпізнавання дорожніх знаків.

Згорткова нейронна мережа складається з кількох змінних шарів згортки та субдискретизації, які допомагають витягти ключові характеристики із вхідного зображення. На вихідному рівні до архітектури додаються повнозв’язні рівні, щоб робити прогнози на основі вилучених функцій. Щоб виявити дорожні знаки на зображеннях за допомогою CNN, можна використовувати один нейрон у вхідному шарі, який приймає значення в діапазоні від -1 до +1, що відповідає наявності або відсутності дорожнього знаку. На рисунку 1.1 зображено архітектурну структуру CNN, яка використовує цей підхід для виявлення дорожніх знаків на зображеннях.

Згорткові рівні нейронної мережі працюють з використанням наступної математичної формули (1.1):

$$y_k^{(i,j)} = b_k + \sum_{g=1}^{k} \sum_{t=1}^{k} w_{k,s,t} x^{((i-1)+s,(j+t))} \quad (1.1)$$

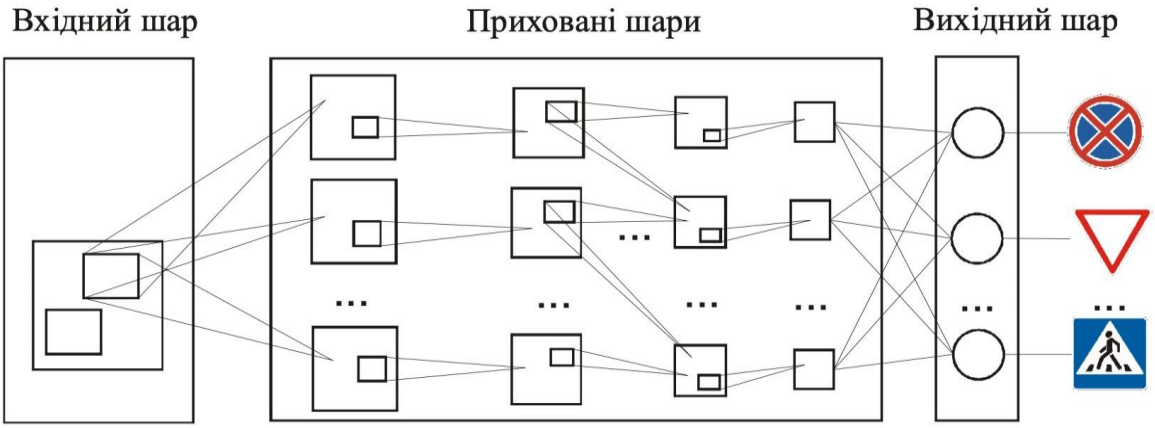

Рисунок 1.1 – Архітектура згорткової нейронної мережі для детектування та класифікації дорожніх знаків

Для навчання згорткової нейронної мережі використовується алгоритм зворотнього поширення помилки (backpropagation) (1.2):

$$w_{i,j}(t+1) = w_{i,j}(t) + \eta \sigma_{pi} o_{pi} \quad (1.2)$$

На основі вищенаведеного аналізу можна зробити висновок, що описана вище нейронна мережа є ефективним рішенням для розробки системи детектування та класифікації дорожніх знаків. Для навчання моделі використовувався сервіс TensorFlow, розроблений Google, який дозволив економічно ефективно навчати невеликі моделі. Для навчання моделі буде використано набір даних із 4000 зображень із набору GTSRB (German Traffic Signs Recognition Benchmark). Цей набір даних було зібрано та класифіковано вручну спеціально з метою навчання та тестування нейронних моделей для детекції та розпізнавання дорожніх знаків. Він

містить широкий спектр дорожніх знаків, що представляють усі класи, які зустрічаються в Європі [9].

1.2.1 Метод порівняння з шаблоном

Ці методи працюють шляхом порівняння кожної частини зображення із заздалегідь визначеним шаблоном для вимірювання взаємної кореляції. Шаблони можуть бути встановлені вручну або встановлені функцією та можуть стосуватися як усього об’єкта, так і його окремих компонентів. Якщо коефіцієнт кореляції перевищує встановлене порогове значення, область аналізу в межах зображення буде позначена як така, що містить дорожній знак. Існують також модифіковані підходи, які використовують кілька шаблонів, що відповідають окремим елементам об’єкта, включаючи області у формі ліній. Ці алгоритми працюють поетапно, на кожному етапі оцінюється надійність виявлених областей. Коли область відповідає будь-якому з існуючих шаблонів, вона позначається як точка інтересу для подальшого аналізу. Цей процес поступово збільшує схожість між областями зображення та їхнім еталонним шаблоном. Згодом ідентифіковані області порівнюються з іншими шаблонами, що призводить до ідентифікації потрібного об’єкта. Ці методи спочатку використовувалися в ранніх дослідженнях щодо класифікації та розпізнавання об’єктів зображення, представляючи початкові експерименти для встановлення особливостей зображення об’єктів.

За останні роки шаблонні методи зазнали значних змін, що призвело до підвищення точності розпізнавання. Однак основним недоліком цих методів є їх низька швидкість роботи, що може перешкоджати загальній ефективності. Ці методи також дуже чутливі до спотворень об’єктів на зображеннях.

1.2.2 Метод лінійної дискримінантної класифікації

Лінійний дискримінантний аналіз (LDA) – це алгоритм класифікації, метою якого є класифікація об’єктів на кілька класів. На відміну від методу головних

компонентів (PCA), LDA не має на меті зменшити простір функцій до найменших розмірів, а скоріше спроектувати функції в простір, де відмінності між класами максимізуються. Це робиться шляхом знаходження підпростору меншої розмірності, який зберігає найважливішу інформацію про різні класи, одночасно мінімізуючи варіації всередині класу та максимізуючи відмінності між класами. Цей метод особливо корисний під час роботи з великими наборами даних зображень, де кожен об’єкт має характерні риси, подібні до характеристик інших об’єктів того ж класу. Проектуючи простір ознак на низьковимірний підпростір, LDA гарантує, що кластери подібних об’єктів утворюють компактні та розділені групи з мінімальним перекриттям між різними класами. Загалом, LDA є потужним інструментом для виділення ознак, який може допомогти зменшити складність великих наборів даних.

1.2.3 Метод головних компонент

Метод головних компонентів (PCA) представлений в 1901 році К. Пірсоном і зараз має різні застосування в таких сферах, як аналіз зображень, розпізнавання образів і стиснення даних. Основна мета методу – зменшити розмірність інформації, що міститься в наборі даних, без втрати надто важливої інформації. Ця техніка має велике значення для аналізу та обробки зображень, де потрібно ефективно обробляти великі обсяги даних. Алгоритм PCA працює шляхом виконання лінійного ортогонального перетворення на багатовимірному вхідному векторі X, що містить корельовані змінні, що призводить до меншого вектора Y. Ці некорельовані змінні називаються головними компонентами, і вони містять важливу інформацію, яка була присутня у вихідному наборі даних. Цей аналіз застосовується до всіх прикладів навчального набору зображень, які складають вектор X. Під час категоризації об’єктів на зображенні процес передбачає обчислення евклідової відстані між основними компонентами зображення, що аналізується, та зваженою комбінацією головних компонентів навчального набору.

Якщо первинні компоненти зображення тісно відповідають основним компонентам будь-якого об'єкта з навчальної множини, то вважається, що зображення містить саме цей об'єкт. Для забезпечення ефективного використання цього методу зображення об'єктів повинні бути зняті в однакових умовах реєстрації. Перевага використання методу PCA полягає в тому, що він зменшує розмірність даних, що виявляється важливим фактором при роботі з великими базами даних зображень. Цей метод спеціально працює шляхом вилучення основної структури даних і визначення шаблонів, які можна використовувати для категоризації нових даних. Метод PCA часто використовується в програмах комп’ютерного зору та обробки зображень і корисний у таких сферах, як розпізнавання обличчя та ідентифікація за відбитками пальців.

1.2.4 Метод структурно-символьної локалізації

Структурні методи стосуються використання символів, проаналізованих на основі їх топології, для отримання інформації про просторове розташування окремих частин символів. Кожен символ може бути представлений у вигляді графіка за допомогою набору характерних властивостей. Процес розпізнавання символу базується на аналізі цих характерних властивостей і виконанні аналізу досліджуваного зображення. Однак на першому етапі цих методів зображення символу має пройти процедуру попередньої обробки, яка передбачає скелетизацію. Ця процедура спрощує структуру зображення символу, щоб полегшити процес розпізнавання. Після процедури скелетонізації символи перетворюються на набори унікальних точок. Потім для цих спеціальних точок оцінюється кілька параметрів, включаючи їх координати, довжину сполучних ребер у межах графіка, нормалізований напрямок входів і виходів цих точок, а також кривизну дуги, створеної з’єднанням кожної спеціальної точки. Характеристики створеного зображення порівнюються з властивостями еталонного графа для кожного символу, і якщо вони збігаються, вершини графа з’єднуються дугами. Потім

визначається найбільша максимальна кліка в створеному графі, а символ на зображенні групується в клас, який має найбільшу кількість вершин у своєму графі.

1.2.5 Метод адаптованого полішення

Алгоритм AdaBoost – це популярна техніка машинного навчання, яка використовується для класифікації об’єктів у зображеннях і відеорядах. Цей алгоритм використовує адаптивний підхід і має каскадну структуру, що складається зі слабких класифікаторів. Кожен слабкий класифікатор зосереджується на визначенні певних характеристик і може дати «істинну» або «хибну» відповідь, навчаючись на помилках попереднього класифікатора. Сума значень слабких класифікаторів обчислюється в кожному каскаді, і результат порівнюється із заздалегідь визначеним порогом. У результаті каскадний класифікатор допомагає усунути області зображення, які не містять бажаного об’єкта на основі накопичених знань про слабкі класифікатори. Ця методика гарантує, що розглядаються лише найбільш ймовірні регіони з об’єктом інтересу. Щоб сформувати надійний класифікатор, AdaBoost поєднує в собі характерні риси слабких класифікаторів. Як правило, фундаментальний алгоритм навчання відомий як слабкий класифікатор. Однією з головних переваг AdaBoost є те, що це не тільки швидкий класифікатор, але й техніка, яка використовується для вибору функцій. Процес генерації прототипів векторів ознак для різних категорій об’єктів досягається шляхом вибору медіани з усіх відповідних елементів у векторах ознак усього навчального набору послідовностей виразів обличчя. Специфіка цього процесу детально обговорена в роботі [15]. На рішення використовувати медіану замість середнього впливає висока стійкість до дисперсії. Використовуючи медіану, на вектори ознак прототипу не впливають жодні наявні викиди. У 2001 році комп’ютерні вчені Пол Віола та Майкл Джонс представили більш просунутий алгоритм під назвою алгоритм Віоли-Джонса, який є реалізацією алгоритму AdaBoost. Цей передовий алгоритм дозволяє виявляти різні об’єкти на зображеннях

за допомогою аналізу характеристик Хаара. Особливості Хаара стосуються прямокутних областей на зображенні, які складаються з кількох суміжних частин. Кожен клас об'єктів має свою унікальну ідентичність, яку можна виразити через розподіл їхніх конкретних характеристик Хаара. У кожному полі зображення розраховуються тисячі варіантів розташування об'єктів залежно від масштабу та їх статистичної поведінки. Чорні та білі області функції Хаара підсумовуються, і різниця між цими сумами вимірюється для відповідного рівня інтенсивності пікселів. На рисунку 1.3 представлена блок-схема алгоритму Adaboost.

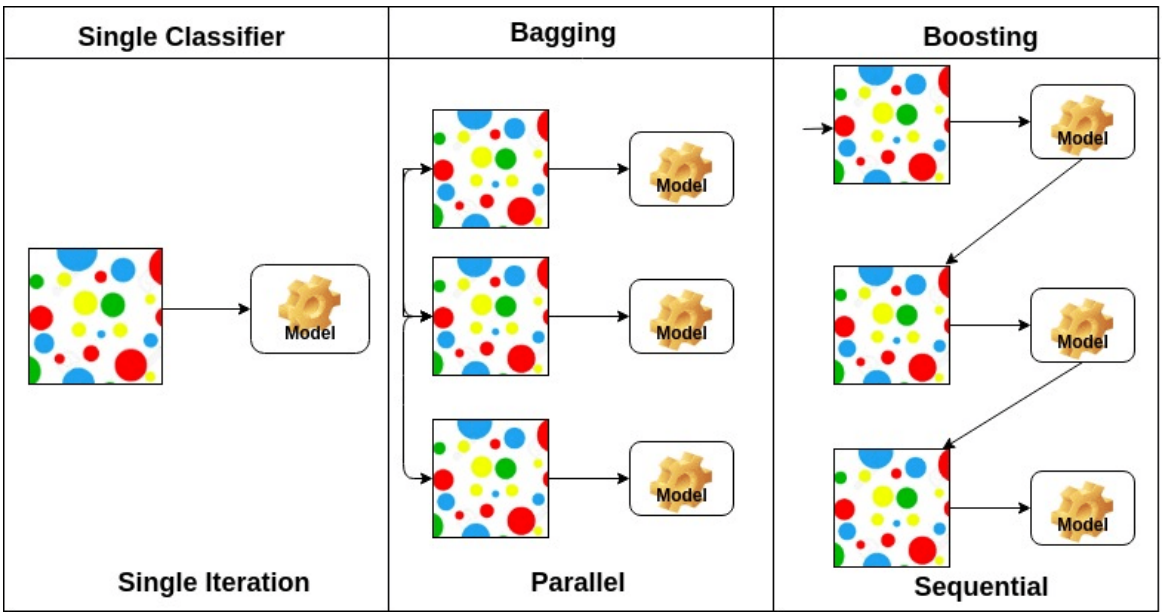

Рисунок 1.2 – Блок-схема роботи алгоритму AdaBoost [12]

Ефективність алгоритму навчання значною мірою залежить від типу проблеми, яку він намагається вирішити, а сам алгоритм може мати різні параметри та конфігурації, які потрібно налаштувати для досягнення оптимальної продуктивності набору даних.

AdaBoost, який використовує дерева рішень як слабкі навички, часто вважається найкращим попередньо створеним класифікатором. Коли цей алгоритм використовується з навчанням дерева рішень, він збирає інформацію про стійкість кожного навчального зразка на кожному етапі процесу AdaBoost, а потім передає

ці дані в алгоритм вирощування дерева. У результаті наступні дерева зосереджуються на прикладах, які важче класифікувати.

1.2.6 Метод прискорених стійких ознак

Прискорені стійкі функції (SURF) є детектором і дескриптором, який використовується в програмах комп'ютерного зору, таких як розпізнавання об'єктів, зіставлення зображень, класифікація та об'ємна реконструкція. Він був розроблений на основі дескриптора Scale Invariant Feature Transform (SIFT) і розроблений, щоб бути більш ефективним і надійним, ніж SIFT, щодо різноманітних перетворень зображень. SURF виявляє особливі точки за допомогою цілочисельних наближень спекл-детектора через визначник Гессе, який можна обчислити за допомогою трьох цілочисельних операцій із попередньо обчисленим інтегральним зображенням. Його дескриптор ознак базується на сумі вейвлет-відповідей Гаара навколо сингулярної точки, яку також можна обчислити за допомогою інтегрального зображення. Стандартна версія SURF набагато швидша, ніж SIFT, і її автори визнають надзвичайно надійною. Для перетворення зображень у координати використовується техніка піраміди з різною роздільною здатністю, яка повторює вихідне зображення шляхом створення піраміди Гауса чи Лапласа зі зменшеною пропускною здатністю. Отримане зображення має той самий розмір, що й оригінал, але зі спеціальним розмиттям зображення, відомим як масштабний простір. Простір відповідає за забезпечення незмінності особливих точок, які є точками з унікальними характеристиками, які залишаються впізнаваними в різних масштабах. Для обчислення дескриптора в комп'ютерному зорі спочатку вибирається прямокутна область навколо конкретної точки інтересу. Потім область вирівнюється таким чином, щоб її орієнтація відповідала орієнтації об'єкта. Дескриптор отримано в результаті аналізу 16 квадрантів, що оточують цю точку інтересу. Вихідне зображення не перевертається під час обчислення, а фільтр оцінюється в стандартній орієнтації.

Отримані координати градієнта налаштовуються відповідно до попередньо визначеної орієнтації. Алгоритм має три складові: виявляння особливих точок, описування локального околу, та зіставляння.

1.3 Огляд існуючих рішень

Розглянемо сучасні системи ідентифікації дорожніх знаків, серед яких: Open Eye Assistant, Mercedes Speed Limit Assistant, Ford Speed Sign Detection.

1.3.1 Opel Eye Assistant

Система Opel Eye – це інноваційна технологія, яка дозволяє розпізнавати дорожні знаки. Система працює за допомогою відеокамери високої роздільної здатності, яка встановлена біля лобового скла автомобіля, розташована між вітровим склом і дзеркалом заднього виду. Ця камера здатна фіксувати велику кількість інформації про дорогу попереду завдяки широкому куту огляду та високій частоті кадрів до тридцяти кадрів на секунду, які обробляються двома процесорами обробки сигналів. Серед дорожніх знаків, які можна розпізнати, є заборона обгону, обмеження швидкості та інші. На рисунку 1.2 представлена візуалізація роботи системи Opel Eye Assistant.

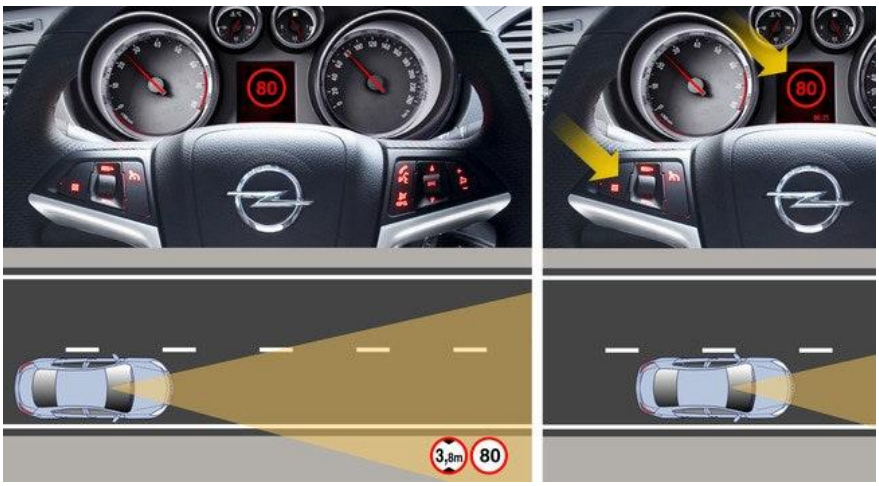

Рисунок 1.2 – Приклад роботи системи Opel Eye Assistant

Система має базу даних дорожніх знаків, збережену в пам’яті, яку вона використовує, щоб зіставити зроблені зображення з існуючими знаками. У разі збігу на панелі приладів відображається відповідний знак, який інформує водія про конкретний дорожній знак, який розпізнала система.

Система Opel Eye розроблена, щоб допомогти водієві виявляти дорожні знаки, навіть якщо водій їх пропускає. Це досягається відображенням пропущеного знака на панелі приладів. На відміну від інших навігаційних систем, Opel Eye оцінює ситуацію на дорозі в реальному часі, забезпечуючи таким чином кращу точність розпізнавання стану поточної дорожньої обстановки. Однак точність розпізнавання значно знижується, якщо знак нахилений, частково закритий або забруднений. Система також має навігаційну програму, яка відправляє спеціальні повідомлення щодо заторів на певних ділянках дороги. Точність системи Opel Eye становить близько 75%.

1.3.2 Mercedes Speed Limit Assistant

Найновіші автомобілі S-, CL-, CLS- і E-класу оснащені інноваційною технологією під назвою Speed Limit Assist. Його призначення – допомогти водієві дотримуватися встановленої швидкості на певній ділянці дороги. Приклад роботи системи Mercedes Speed Limit представлений на рисунку 1.3.

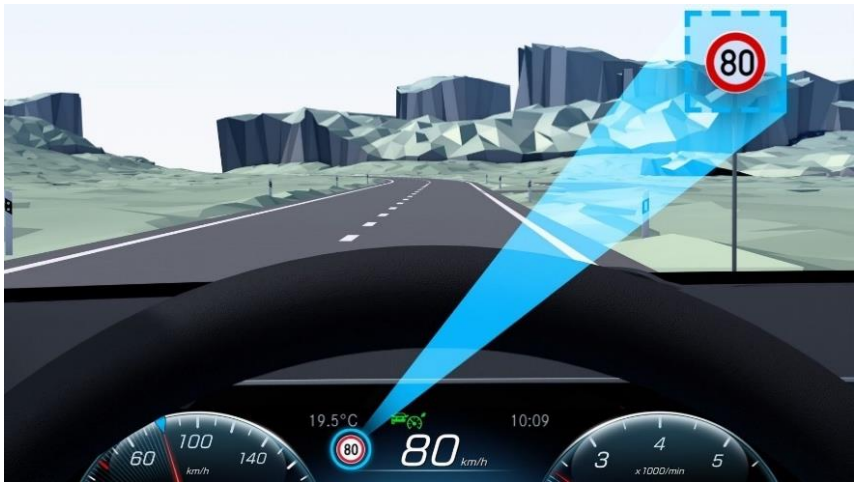

Рисунок 1.3 – Приклад роботи системи Mercedes Speed Limit Assistant

Остання версія Speed Limit Assist зазнала значних удосконалень, дозволяючи їй не лише розпізнавати дорожні знаки, але й оцінювати, чи відповідають отримані дані вантажівкам. Крім того, система здатна розпізнавати кінець знаків обмеження швидкості. Якщо знак обмеження швидкості не було знято протягом певного періоду часу, навігаційна система відобразить дані про обмеження швидкості, записані раніше для цієї конкретної ділянки дороги. Ця передова технологія аналізує зображення, передані камерою, встановленою на лобовому склі, у режимі реального часу, що дозволяє швидко розпізнавати знак незалежно від місця розташування. Завдяки цим удосконаленням система може аналізувати дані протягом секунди та негайно передавати результати на дисплей. Крім того, на думку німецьких експертів, ця інноваційна система, розроблена для допомоги водіям у дотриманні обмежень швидкості, значно зменшить частоту аварій [10].

Згідно зі статистичними даними, велика кількість ДТП відбувається через перевищення водіями дозволеної швидкості на дорогах. На основі досліджень було підраховано, що загальна точність визначення дорожніх знаків системою Mercedes Speed Assistant становить близько 70 відсотків [10].

1.3.3 Ford Speed Sign Detection

Система розпізнавання дорожніх знаків Ford – це інноваційна технологія, розроблена для виявлення дорожніх знаків і відображення їх на приладовій панелі автомобіля, повідомляючи водія про будь-які важливі знаки, які потребують його уваги під час руху. Це особливо корисно для водіїв, яким важко стежити за важливими знаками на дорозі, оскільки забезпечує візуальне відображення відповідних дорожніх знаків і повідомлень. Крім того, технологія запрограмована на блимання попередження, коли водій перевищує обмеження швидкості, що сприяє безпечному та законному водінню, мінімізуючи ризик переслідування. Це стало можливим завдяки передній відеокамері, встановленій за дзеркалом заднього виду, яка постійно контролює зустрічні дорожні знаки та відображає їх на

кольоровому екрані. Система може навіть показувати два знаки одночасно, забезпечуючи водієві чітке бачення всіх відповідних інструкцій. Однією з найбільш привабливих особливостей системи розпізнавання дорожніх знаків Ford є її гнучкість.

Водій може вибрати, коли отримувати попередження про порушення, з опціями в діапазоні від 5, 10 або 15 миль/год вище дозволеної швидкості.

Приклад роботи розпізнавання знаків системою Ford Speed Sign Detection представлено на рисунку 1.4.

Рисунок 1.4 – Приклад роботи системи Ford Speed Sign Detection

1.4 Постановка задачі

Метою дослідження є розробка системи детектування та класифікації дорожніх знаків. На основі вищезазначеного можна виділити наступні функціональні можливості системи:

– виконання попередньої обробки цифрових зображень дорожніх знаків для здійснення подальших операцій з ними;

– можливість налаштування параметрів класифікаторів;

– здійснення нормалізації зображень символів;

– розпізнавання символів на зображеннях дорожніх знаків.

Спираючись на сучасні вимоги до якості програмного забезпечення, що розробляється, програмна система розпізнавання дорожніх знаків повинна мати наступні характеристики:

– стійкість програмної системи до різних помилок, збоїв та дій, що виходять за рамки її функціонального призначення;

– програмна система повинна володіти модульною архітектурою, де кожен модуль призначений для вирішення певних завдань;

– коректна робота системи в вирішенні поставлених завдань;

– можливість розширювати, змінювати і доповнювати функціональність розроблених бібліотек. Для забезпечення цього пункту необхідно слідувати принципам децентралізації архітектури ПЗ.

2 ОГЛЯД ІСНУЮЧИХ СУЧАСНИХ АЛГОРИТМІВ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

2.1 Метод опорних векторів

Метод опорних векторів (англ. Supportt Vector Machine) — це метод лінійної класифікації, який використовується для вирішення завдань, пов’язаних із класифікацією та регресійним аналізом. Зокрема, коли йдеться про класифікацію об’єктів на зображеннях, метод SVM працює, беручи набір точок у просторі, згрупованих у два класи, і знаходить оптимальну гіперплощину, яка може точно розділити точки на ці класи. Іншими словами, цей метод вимагає накреслення лінії в просторі, яка може розділити об’єкти на дві категорії. Щоб досягти точної класифікації, метод SVM шукає гіперплощину, яка може максимізувати відстань до кожного класу. Це означає, що чим більша відстань між гіперплощиною та об’єктами, що належать до кожного класу, тим менша помилка класифікації. Крім того, опорні вектори, які розташовані найближче до гіперплощини, мають вирішальне значення для оптимізації точності класифікації. Щоб класифікувати вектори з різних класів, важливо знайти функцію класифікації f(x), яка виводить різні значення для кожного класу. Щоб визначити цю функцію, ми використовуємо навчальний набір даних, що складається з пар (x_1, y_1), ..., (xm, ym). Оптимальною функцією є та, яка дає мінімальну очікувану помилку класифікації.

Нижче представлено вигляд функції класифікації (2.1):

$$f(x) = sign\left((w, x) + b\right) \quad (2.1)$$

де x – об'єкт, що належить простору; w – вектор нормалі до гіперплощини; b – допоміжний параметр.

У задачах класифікації об’єкти, які дають значення 1 або -1 на основі функції $f(x)$, призначаються до різних класів. Вирішальним стає ретельний вибір параметрів «w» і «b» таким чином, щоб відстань від цих об’єктів до кожного класу

була максимальною. У сценаріях реального світу існують певні ситуації, коли лінійний поділ даних стає неможливим. Для вирішення такої умови початковий простір 'x' перетворюється на простір вищого виміру за допомогою унікальної функції відображення φ: $Rn \rightarrow X$.

До цієї трансформації простору підходять у спосіб, який забезпечує лінійну невіддільність у результуючому просторі порівняно з початковим простором. Однак слід мати на увазі, що зі збільшенням розмірності простору процес класифікації також стає складнішим.

У реальному світі незвично стикатися з проблемами розпізнавання, коли дані можна розділяти лінійно. Для вирішення таких випадків використовується техніка використання ядер, яка була запропонована Б. Бозером, І. Гуйоном та В. Вапнікою. У цьому методі використовується спеціальна функція відображення φ: $Rn \rightarrow x'$ для вбудовування елементів вибірки у більший розмірний простір x', де вибірка може бути розділена лінійно. Ядро класифікатора представлено виразом $x\,(x, x') = (\varphi\,(x), \varphi\,(x'))$, який визначає відображення вибірки в новий простір, і його роль може виконувати будь-який позитивний певна симетрична функція двох змінних. Іншими словами, метод ядра перетворює вибірку в новий простір, де її стає набагато простіше лінійно відокремити, і це перетворення здійснюється через застосування спеціальної функції відображення. На практиці зустрічаються такі ядра: радіальна базисна функція, гауссова базисна функція, сигмоїда [11].

Під час навчання моделі за допомогою методу опорного вектора зазвичай використовуються методи квадратичного програмування, такі як послідовна мінімальна оптимізація. Однак у використанні цього методу є плюси і мінуси порівняно з використанням багатошарових персептронів. Багатошарова модель персептрона має численні приховані параметри, які змінюються залежно від кількості нейронів у мережі. Ця параметризована модель має здатність охоплювати більш складні та складні функції, але з недоліком, що вимагає більш тривалого часу навчання та більших обчислювальних ресурсів для точного налаштування параметрів.

Метод опорних векторів, з іншого боку, вибирає вектори з навчальної вибірки та має обмежені параметри, які визначаються розміром вибірки.

У сфері алгоритмів розпізнавання та машинного навчання використання методу опорного вектора зараз переважає завдяки його здатності скорочувати час навчання та уникати потрапляння в локальні мінімуми, що підвищує його загальну стабільність. Метод опорного вектора, який в основному використовується для програм розпізнавання зображень, як-от ідентифікація людських облич, має високий рівень успіху з точністю 80-85% для вирівняних зразків. Однак розпізнавання зображень становить унікальну проблему, оскільки візуальні сигнали фактично мають низьку інформаційну ємність і лише кілька точок на растровому зображенні (наприклад, ті, що відповідають однорідному фону) несуть важливу інформацію для цілей розпізнавання [11].

Класичні методи розпізнавання образів зазвичай включають пряму кореляцію між кількістю параметрів у вибіркових даних і необхідним часом навчання, а також сигнали конвергенції під час фази оптимізації моделі. Однак наявність великої кількості параметрів, де більшість із них фактично не містить необхідної інформації для розпізнавання, може негативно вплинути на продуктивність моделі. Крім того, такі моделі вимагають значно більших обчислювальних ресурсів і призводять до явища, відомого як перетренування, коли функція розпізнавання, яка апроксимується моделлю, успішно класифікує навчальну вибірку, але не може узагальнити та демонструє низький рівень точності для тестової вибірки. Щоб подолати цю проблему, використовується більш ефективний метод відбору ознак для ідентифікації стислого представлення зображення, що містить лише обмежену кількість узагальнених ознак, які фіксують необхідну інформацію для розпізнавання.

Метод SVM широко застосовується для ідентифікації різних об'єктів, наприклад дорожніх знаків. З іншого боку, під час процесу класифікації цей метод враховує лише окрему частину даних, розташовану в основному на межах набору, ігноруючи частину, що залишилася.

Незважаючи на це обмеження, метод SVM має перевагу перед іншими методами, оскільки для точної класифікації потрібен невеликий набір даних.

2.2 Метод на основі згорткової нейронної мережі

Типом архітектури штучної нейронної мережі, яка зазвичай використовується в задачах розпізнавання зображень, є згорточна нейронна мережа. Американські нейрофізіологи ще в 1962 році провели дослідження, яке показало, що нервова клітина в мозку реагує на зорове сприйняття кордону в певному напрямку. Кожен нейрон відповідає за виявлення певних візуальних особливостей, таких як горизонтальні, вертикальні або діагональні межі, і ці нейрони розташовані у формі стрижня, який поєднується, щоб сформувати остаточне візуальне сприйняття. Використовуючи алгоритмічний підхід для імітації цього процесу, архітектура згорткової нейронної мережі може успішно класифікувати та ідентифікувати конкретні особливості зображення. Ця архітектура виявилася досить ефективною в задачах розпізнавання зображень.

Згорткова нейронна мережа (CNN) – це тип нейронної мережі, яка приймає цифрові матриці як вхідні дані. Для зображень у відтінках сірого використовується одна матриця, тоді як багатошарові зображення RGB представлені за допомогою трьох матриць. Мережа передбачає використання фільтра для ідентифікації конкретних особливостей або характеристик вхідного зображення, яке також представлено матрицею із заздалегідь визначеними значеннями. Ефективний дизайн архітектури для розпізнавання активності за допомогою CNN вимагає оцінки складності шарів обробки, кількості шарів і швидкості виведення. Кількість рівнів у CNN залежить від дизайну мережі та проблеми, яку вона намагається вирішити. Перший рівень у CNN завжди згортковий, що дає назву мережі.

Матриця результату є результатом операції згортання матриці фільтра та частини матриці зображення, і її розмір можна порівняти з матрицею фільтра (рис. 2.1). За допомогою наступного рівняння кожен елемент результату обчислюється

як скалярний добуток матриці фільтра та підматриці того самого розміру (частина зображення):

$$C_{i,j} = \sum_{u=0}^{m_x-1} \sum_{v=0}^{m_y-1} A_{i+u,j+v} B_{u,v'} \quad (2.2)$$

де A, B – матриці з розмірністю $n_x \times n_y$ та $m_x \times m_y$ відповідно, C – вихідна матриця з розмірністю $(n_x - m_x + 1) \times (n_y - m_y + 1)$.

Міра подібності між початковим зображенням і застосованим фільтром вказується значенням у матриці результатів, як показано на рис. 2.1. Важливо зауважити, що кількість каналів у фільтрі має відповідати кількості каналів у вхідному зображенні. Наприклад, при обробці трьох матриць зображення кожен канал повинен мати три матриці. Кількість карток (або фільтрів), які будуть використовуватися, визначається вимогами роботи. Збільшення кількості фільтрів покращує точність ідентифікації, але також призводить до більш складних обчислень, як зазначено в [16]. Загалом рекомендоване співвідношення один до двох, за якого кожне зображення попереднього шару (наприклад, першого згорткового шару, причому попередній шар є вхідним) з'єднується з двома зображеннями згорткового шару нейронної мережі.

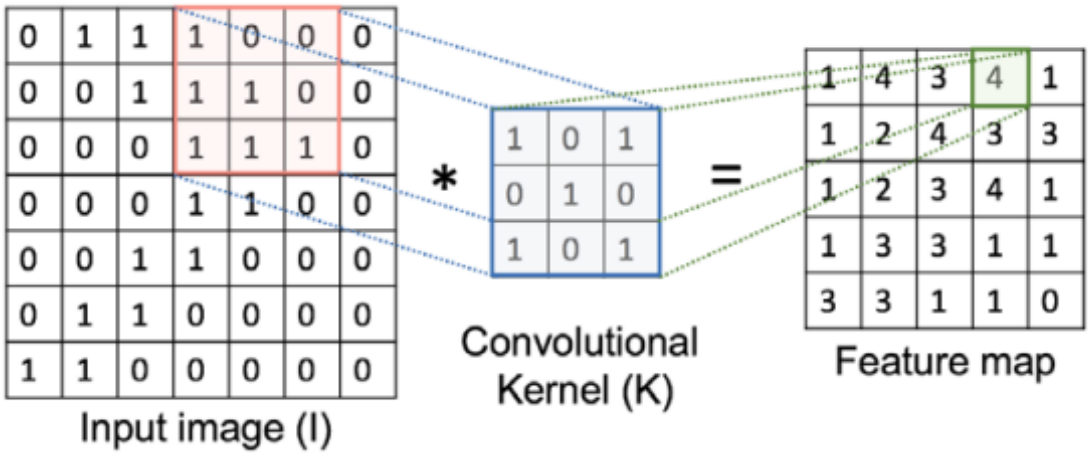

Рисунок 2.1 – Приклад згортки матриць розмірністю 7×7 та 3×3 [8]

У контексті згорткових нейронних мереж карти активації, створені під час проходження зображень через послідовні згорточні шари та підвибірки, стають дедалі складнішими, визначаючи більш складні характеристики. Метою шарів підвибірки є зменшення розмірності карт активації попереднього згорткового шару. Це служить двом основним цілям: усунення непотрібних деталей, які вже були оброблені на попередньому рівні, і запобігання перенавчанню шляхом фільтрації зайвої інформації. Зменшення розмірності карт активації через підвибірку допомагає підтримувати обчислювальну ефективність, зберігаючи відповідну інформацію [10].

Під час процесу підвибірки в нейронній мережі ядро, яке використовується шаром підвибірки, не перекривається, на відміну від ядра, що використовується в згортковому рівні. Карта шару підвибірки зазвичай має ядро, яке вдвічі менше за розмір карти попереднього шару. Потім уся карта функцій ділиться на менші клітинки, і з кожної клітинки вибирається максимальне значення. Крім того, рівень підвибірки часто включає функцію активації Rectified Linear Unit (ReLU), яка підвищує точність оцінки нейронної мережі. Важливість підвибірки в процесі вилучення характеристик із нейронної мережі не можна недооцінювати. Використовуючи підвибірку, нейронна мережа може ідентифікувати та виділяти певні характеристики.

Згорткова нейронна мережа складається з кількох шарів, причому повністю зв’язаний шар є останнім типом шару. Зазвичай він реалізується за допомогою багатошарового персептрона та відіграє вирішальну роль у класифікації, використовуючи складні нелінійні функції та оптимізуючи їх для покращення якості розпізнавання. Архітектура CNN передбачає багаторазове згортання зображення з подальшим ущільненням за допомогою підвибірки, що генерує абстрактні карти ознак. Наступні шари, як правило, збільшують кількість каналів і зменшують розмірність зображення в кожному каналі. Отже, створюється набір із кількох каналів, у кожному з яких зберігаються невеликі обсяги даних (наприклад, окремий параметр), які інтерпретуються як абстрактні поняття, присутні на

початковому зображенні. Отже, повнозв'язний рівень у CNN відповідає за ідентифікацію складних нелінійних функцій, які сприяють покращенню якості розпізнавання системи. Коли зображення проходить через шари мережі, канали збільшуються, а розмірність зображення зменшується, що призводить до появи безлічі абстрактних карт функцій, доступних для інтерпретації. Зрештою, повнозв'язний рівень зберігає кілька каналів, кожен з яких містить невелику кількість даних, інтерпретованих як концепції, які допомагають у процесі класифікації [7, 10].

Процес передачі даних у нейронних мережах передбачає об'єднання даних, які потім передаються до повноцінної нейронної мережі. Цей тип нейронної мережі може мати кілька рівнів, що дозволяє проводити більш складний аналіз даних. Однак у результаті повного з'єднання шару просторова структура пікселів втрачається, і мережа має менший розмір у порівнянні з вихідним зображенням. На рисунку 2.2 представлено узагальнене зображення нейронної мережі з використанням багатошарового персептрона.

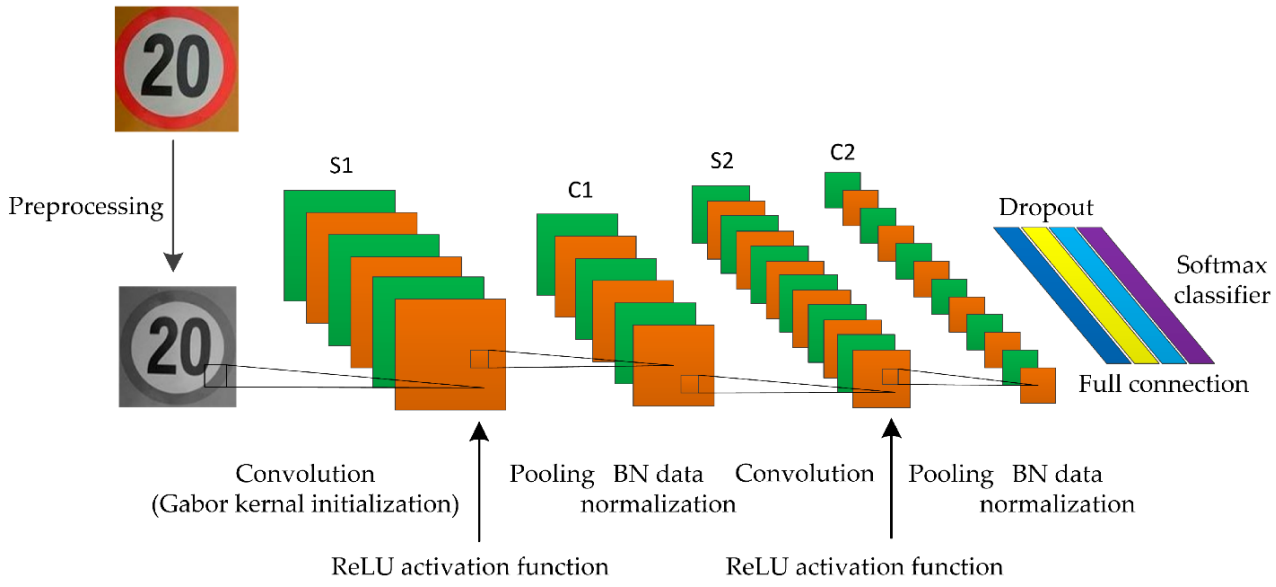

Рисунок 2.2 – Узагальнена архітектура згорткової нейронної мережі [14]

У нейронній мережі вихідний рівень відіграє вирішальну роль в оцінці ймовірності належності даного вхідного зображення до певного класу з певним

рівнем достовірності. Щоб зробити це визначення, кількість нейронів у вихідному шарі має відповідати кількості класів ознак у наборі даних. Потім до сигналу, створеного вихідним шаром, застосовується функція активації для подальшого уточнення результатів. Для навчання моделі використовується популярна і широко використовувана техніка зворотного поширення помилки.

2.3 Алгоритм Віоли-Джонса

Алгоритм Віоли-Джонса широко визнаний як ефективний підхід до вирішення проблеми розпізнавання дорожніх знаків завдяки його чудовій здатності виявляти об’єкти на зображеннях. Спочатку розроблений Полом Віолою та Майклом Джонсом у 2001 році, цей алгоритм ґрунтується на використанні двох життєво важливих функцій, а саме техніки ковзного вікна та каскадного протоколу Хаара. Техніка розсувного вікна передбачає створення рамки, яка охоплює зображення, її розміри ретельно відрегульовані для інкапсуляції важливих особливостей зображення. Після завершення етапу навчання алгоритм використовує каскадний алгоритм Хаара для класифікації вхідного зображення на дві категорії залежно від того, містить воно визначений об’єкт чи ні. Алгоритм Віоли-Джонса вважається одним із найефективніших методів у сфері розпізнавання об’єктів завдяки його здатності забезпечувати високу швидкість обробки даних і якість розпізнавання в порівнянні з іншими сучасними методами. Крім того, цей метод має відносно низьку ймовірність отримання хибнопозитивних результатів під час процесу класифікації. Алгоритм перетворює вхідне зображення в градації сірого для спрощення обробки шляхом зменшення обсягу даних перед початком будь-яких методів розпізнавання об’єктів [9].

Нижче розглянемо принцип роботи алгоритму Віоли-Джонса [12]:

Етап 1. Передбачає перетворення RGB-зображення, яке складається з трьох кольорових каналів, на зображення у градаціях сірого з $2^8 = 256$ рівнями яскравості.

Потім це зображення у градаціях сірого перетворюється на інтегральне представлення, відоме як матриця яскравості.

Техніка матриці яскравості використовується для обчислення загальної яскравості будь-якої прямокутної області в межах зображення, зберігаючи незмінний час обчислення незалежно від розміру прямокутника. Інтегроване представлення цієї техніки приймає форму матриці, яка відображає пропорції вихідного зображення. Ця матриця формується шляхом додавання значень яскравості всіх пікселів ліворуч і над кожним елементом матриці. Потім сума яскравості зберігається в кожному елементі за формулою, наведеною нижче (2.3):

$$L(x,y) = \sum_{i=0,j=0}^{i\le x, j\le y} I(i,j) \qquad (2.3)$$

де $I(i,j)$ – яскравість пікселів вихідного зображення.

Матриця $L(x,\ y)$ охоплює елементи, які позначають суму пікселів у прямокутнику від (0, 0) до (x,y). Кожен піксель (x,y) має значення, що дорівнює сумі значень усіх пікселів, розташованих ліворуч і вгору від нього. Час, необхідний для розрахунку зображення, є лінійним і прямо пропорційним кількості пікселів, присутніх на зображенні, тобто повне зображення можна отримати за один прохід.

На другому кроці процесу до матриці яскравості, отриманої на попередньому кроці, застосовується квадратна маска, яка називається ознакою Хаара. Ознака Хаара призначений для покриття зображення різними прямокутними областями, які прилягають одна до одної та розділені на дві групи. Ці області відрізняються своєю шириною та висотою, і вони можуть бути розташовані в різних положеннях на зображенні. Маска систематично рухається по зображенню, змінюючи свій розмір і положення з кожним кроком і охоплюючи різні частини зображення в різний час. Метою цього кроку є застосування маски ознаки Хаара до матриці яскравості для виділення особливостей зображення. Маска знака Хаара містить прямокутні області різного розміру та висоти, які прилягають одна до одної та розділені на дві групи. Його положення та розмір систематично змінюються під час

руху, охоплюючи різні частини зображення в різний час. Цей метод призначений для захоплення всіх можливих особливостей зображення шляхом дослідження різних конфігурацій прямокутних областей у межах маски в різних положеннях.

Алгоритм Віоли-Джонса заснований на каскаді посилених функцій Хаара. Посилення здійснюється за допомогою AdaBoost, алгоритму машинного навчання, який поєднує слабкі класифікатори, створені на основі функцій Хаара, у потужний класифікатор. За наявності достатньої кількості різних слабких класифікаторів AdaBoost створить потужний класифікатор із довільною точністю. Рішення про те, чи виявлено об'єкт, приймається шляхом голосування слабких класифікаторів, кожен відповідно до своєї ваги. Каскадні класифікатори значно пришвидшують цей процес, оскільки першими голосують більш важливі слабкі класифікатори: якщо їхнє рішення негативне, зображення відхиляється, а інші менш важливі класифікатори взагалі не голосують. Об'єкт класифікується позитивно, лише якщо він успішно проходить через каскад, позитивно класифікований на кожному етапі.

На кроці 3 процесу виявлення дорожніх знаків значення функції для певної області зображення обчислюється шляхом віднімання суми значень пікселів у чорному прямокутнику від суми значень пікселів у білому прямокутнику. У цьому обчисленні використовуються інтегральні зображення, які передбачають підсумовування цифрових значень яскравості групи пікселів у матриці. Якщо різниця між значеннями яскравості в білих і чорних прямокутниках перевищує визначений поріг, тоді виконується додаткова обробка області зображення. Якщо порогове значення не досягнуто, то область розпізнавання не містить дорожніх знаків.

На кроці 4 процесу повторюється крок 2 із використанням нової маски, але лише для області, яка пройшла перший пороговий тест. Будь-яка ділянка, яка пройшла всі попередні етапи процесу, підтверджується як дорожній знак.

2.4 Алгоритм використання комп’ютерного зору для знаходження та розпізнавання дорожніх знаків

Процес виявлення дорожніх знаків за допомогою комп’ютерного зору вимагає послідовного виконання кількох етапів. Ці етапи включають попередню обробку зображення, сегментацію кольорів, фільтрацію шуму та пошук відповідного зображення. Розглянемо етапи детальніше на прикладі. На рисунку 2.3 представлене зображення до виконання етапів обробки.

Рисунок 2.3 – Зображення до виконання етапів попередньої оброки та виявлення дорожніх знаків

На початковому етапі здійснюється попередня обробка зображення. Цей процес включає прості, але ефективні прийоми для покращення якості зображення. Зображення RGB спочатку перетворюється на кольорову схему HSV. Колірний простір HSV складається з трьох каналів: тон, насиченість і значення. Канал Hue відповідає за кодування інформації про колір, а кут відображає тип кольору – від

червоного на 0 градусів, зеленого на 120 градусів і синього на 240 градусів. Канал Saturation кодує ступінь чистоти або інтенсивності кольору – наприклад, рожевий менш насичений порівняно з червоним. Нарешті, канал Value представляє яскравість кольору та відповідає за захоплення аспектів тіні та блиску зображення. На відміну від RGB, який відноситься до основних кольорів, HSV – це колірний простір, який визначається на основі того, як люди сприймають колір. У цьому колірному просторі колір, відтінок і довжина хвилі представлені лише компонентом відтінку, що є головною перевагою використання колірного простору HSV у задачі розпізнавання знаків. Це представлення полегшує роботу з кольорами та сприйняття кольорів, а також дозволяє налаштовувати значення кольорів.

Наступний етап – кольорова сегментація зображення. Існують різні способи сегментації кольорів під час обробки зображень, але для цієї статті було обрано метод, розроблений Флейхом [4]. Цей підхід використовує техніку сегментації регіону зростання. Спочатку затінене зображення відокремлюється за кольором дорожнього знака. Як правило, діапазон кутів відтінку визначається там, де можна знайти колірний спектр дорожнього знака. Результатом цього кроку є бінарне зображення, що виділяє потенційних кандидатів для подальшої обробки. Це бінарне зображення далі використовується для визначення початкових точок для зображення насиченості. Ці початкові точки створюються шляхом поділу бінарного зображення на підобласті 16x16 пікселів і встановлення початкового значення в центрі кожної підобласті, якщо в цій області достатньо пікселів відтінку. Кількість необхідних пікселів відтінку зазвичай визначається шляхом взяття однієї третини загальної площі кожної підобласті. На наступному кроці насіння, створене на попередньому кроці разом із зображенням насиченості, використовуються як вхідні дані для алгоритму вирощування області.

Тут зображення насиченості сегментується з використанням цих початкових точок для створення іншого бінарного зображення, яке представляє потенційні об’єкти дорожніх знаків. Останній крок цього процесу реалізується шляхом застосування логічного І двійкових зображень відтінку та насиченості для

створення єдиного бінарного зображення. На рисунку 2.4 представлений результат кольорової сегментації первинного зображення. В результаті сформовано зображення з сегментацією, яке містить дорожній знак.

Рисунок 2.4 – Результат кольорової сегментації зображення

Наступний етап розпізнавання – фільтрація шуму на зображенні. На попередньому кроці процесу отримано зображення з дорожнім знаком, який виділено з точністю та деяким небажаним шумом. Основна мета цього поточного кроку полягає в тому, щоб підвищити якість алгоритму виявлення шляхом усунення будь-яких шумів, присутніх на зображенні. Процес фільтрації шуму здійснюється в два основних етапи, які включають використання функції floodFill() і техніки морфологічного перетворення.

Функції floodFill використовуються для заповнення підключеного компонента певним кольором, починаючи з початкової точки. Близькість кольорів або яскравості сусідніх пікселів використовується для визначення зв’язку між ними. Щоб піксель був доданий до підключеного компонента, його колір або яскравість має бути достатньо близьким або до одного з його сусідів, які вже

належать компоненту (у випадку плаваючого діапазону), або до кольору або яскравості вихідної точки (у випадку встановленого діапазону). Ці функції дозволяють користувачеві або позначити підключений компонент певним кольором, створити маску, а потім окреслити її, або скопіювати область на інше зображення, серед іншого. Бібліотека Scikit-Learn для Python надає можливості використання функції floodFill(). У обробці зображень другим кроком є морфологічне перетворення, де кілька простих операцій виконуються над бінарними зображеннями на основі форми зображення. Процес передбачає використання структурного елемента або ядра, яке визначає тип операції, яка буде виконана над вихідним зображенням. У більшості випадків морфологічні перетворення здійснюються на бінарних зображеннях. Двома основними операторами морфологічного перетворення є ерозія та розширення, і вони доповнюються вторинними операторами, такими як оператори відкриття, закриття та градієнта. Ерозія, наприклад, дуже схожа на ерозію ґрунту, і вона діє шляхом зменшення помітності контурів об’єкта переднього плану. Тут ядро переміщується по зображенню за допомогою двовимірної згортки. Якщо піксель на зображенні дорівнює 1 або 0, цей піксель вважатиметься 1 лише тоді, коли всі пікселі в ядрі дорівнюють 1, інакше він буде видалений або обнулений. На практиці процес розмивання призводить до видалення небажаного білого шуму, кращої сегментації тісно пов’язаних об’єктів і зменшення розміру. Розрідження та ерозія – це дві протилежні концепції, які використовуються в обробці зображень для керування розміром і формою об’єктів на зображенні. Розведення розширює білі області на зображенні, встановлюючи піксельному елементу значення «1», якщо принаймні один піксель під ним також має значення «1». Це збільшує розмір об’єкта на передньому плані, і це корисно для з’єднання зламаних частин об’єкта. Однак це також може призвести до розрідження, яке відбувається, коли площа об’єкта збільшується, але його форма стає менш чіткою. З іншого боку, ерозія зменшує розмір об’єктів переднього плану, видаляючи фонові пікселі, які вважаються шумом. Це може призвести до того, що об’єкт стане меншим і чіткішим. Однак

ерозія також може призвести до втрати важливих деталей і особливостей об’єкта. Як правило, ерозія використовується в поєднанні з розширенням для усунення цих недоліків. Розкриття – це операція, яка починається з ерозії, а потім розширення. Це корисно для зменшення фонового шуму на зображенні, зберігаючи основні характеристики об’єкта. І навпаки, закриття починається з розширення, а потім ерозії, і це використовується для заповнення невеликих дірок або прогалин в об’єктах переднього плану або видалення чорних крапок з них. На рисунку 2.5 представлено зображення після виконання фільтрації шуму.

Рисунок 2.5 – Результат фільтрації шуму на зображенні

Щоб виявити потенційні дорожні знаки на зображенні, запропонований метод використовує той факт, що більшість дорожніх знаків мають круглу або квадратну форму. Цей метод включає виявлення форм, які дуже схожі на ці звичайні форми дорожніх знаків, а потім аналіз відповідних символів у цих формах з метою виявлення. Процес зіставлення шаблонів використовується для визначення регіонів у зображенні, які відповідають певному шаблону або зображенню. Для

цього процесу потрібні два основні компоненти: вихідне зображення, яке є зображенням, яке шукається на відповідність, і шаблонне зображення, яке є конкретним зображенням латки, яке порівнюється з вихідним зображенням. Мета полягає в тому, щоб визначити найбільш релевантну область зображення, яка відповідає зображенню шаблону. Це досягається шляхом порівняння зображення шаблону з вихідним зображенням у різних місцях, зсуваючи його на один піксель за раз ковзаючи зліва направо та зверху вниз. У кожному місці зображення шаблону в межах вихідного зображення обчислюється оцінка, яка вказує, наскільки патч схожий на цю конкретну область вихідного зображення. Потім ці оцінки зберігаються в матриці результатів для аналізу. Показник відповідності може бути представлений наступною математичною формулою (2.4):

$$R(x,y) = \frac{\sum_{x',y'}(T(x',y') * I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'}(T(x',y')^2 * \sum_{x',y'} I(x+x',y+y')^2}} \quad (2.4)$$

Вищезазначена метрика декларується як TM_CCORR_NORMED в ресурсах бібліотеки OpenCV [26]. На рисунку 2.6 представлено зображення після пошуку за шаблонами з використанням вищезазначної метрики.

Рисунок 2.6 – Результат пошуку дорожніх знаків за шаблонами

На рисунку 2.6 дорожній знак інкапсульовано обмежувальною рамкою. Незважаючи на точне розташування на зображенні, область перекрита безліччю подібних об'єктів. На цьому етапі алгоритму важливий лише обмежувальний квадрат, що охоплює знак. Вибір відповідного обмежувального вікна є останнім аспектом виявлення об’єкта на зображенні.

NMS працює наступним чином: спочатку відкидаються комірки з ймовірністю присутності об'єкта, меншою або рівною 0,6. По-друге, вибирається комірка з найвищою ймовірністю серед кандидатів на об’єкт, і, нарешті, усі комірки з перетином більше 0,5 і об’єднанням, що перевищує значення об’єкта, відкидаються. Цей процес забезпечує вибір оптимальної комірки як місця розташування об’єкта. Метою NMS є зменшення помилкових спрацьовувань, і це звичайний крок у різних алгоритмах виявлення об’єктів.

На рисунку 2.7 представлений результат розпізнавання дорожнього знака на зображенні після виконання всіх вищезазначених етапів.

Рисунок 2.7 – Результат розпізнавання дорожнього знаку на зображенні

3 ВИБІР ЗАСОБІВ ДЛЯ РОЗРОБКИ СИСТЕМИ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ

Для розробки системи розпізнавання дорожніх знаків були використані наступні технології: Python, OpenCV, Keras, Tensorflow, TorchVision. В наступних підрозділах розглянуто технічні характеристики вищеперерахованих технологій.

3.1 Мова програмування Python

Python – це мова програмування, яка широко відома своєю простою та зрозумілою структурою. Вона надає різноманітну колекцію інструментів і бібліотек, що робить його надзвичайно універсальним для різних застосувань, включаючи розпізнавання дорожніх знаків. Python можна використовувати для розпізнавання дорожніх знаків за допомогою попередньої обробки зображень, яка передбачає маніпулювання зображеннями за допомогою бібліотек, таких як OpenCV і Pillow, для полегшення зміни розміру зображення, фільтрації та видалення шуму. Крім того, Python надає кілька бібліотек обробки зображень, таких як scikit-image, оснащених такими методами вилучення, як HOG, SIFT або SURF, які допомагають ідентифікувати відмінні риси з попередньо оброблених зображень. Також, Python містить бібліотеки машинного навчання, такі як Scikit-Learn і TensorFlow, які допомагають створювати та навчати моделі класифікації для зображень дорожніх знаків. Шляхом навчання моделей на об’єктах, отриманих із зображень дорожніх знаків, такі алгоритми, як SVM або CNN можуть бути використані для допомоги в класифікації дорожніх знаків. Python оснащено великою бібліотекою модулів і функцій для різних цілей. Бібліотеки Python NumPy і Pandas є потужними інструментами обробки даних і аналізу. NumPy особливо корисний для числових обчислень, необхідних у машинному навчанні, завдяки його ефективній обробці багатовимірних масивів. У той же час Pandas пропонує безліч структур даних і функцій для редагування та аналізу структурованих даних,

таких як можливість завантажувати набори даних, фільтрувати відсутні значення, очищати дані та виконувати пошуковий аналіз даних. Python легко поєднується з іншими інструментами, що використовуються в процесах машинного навчання, що робить її цінним ресурсом для будь-якого спеціаліста з обробки даних.

Одним із таких інструментів є Jupyter Notebook, який забезпечує зручне інтерактивне середовище для аналізу та експериментування з даними. Python також має можливість інтеграції з бібліотеками візуалізації, такими як Matplotlib і Seaborn, які допомагають створювати глибокі та інформативні візуалізації даних.

3.2 Бібліотека OpenCV

OpenCV, що розшифровується як Open Source Computer Vision Library, це бібліотека обробки зображень і комп'ютерного зору з відкритим кодом, яка була розроблена та підтримується спільнотою OpenCV. OpenCV написана на C++, але має прив'язки до кількох інших мов програмування, включаючи Python. OpenCV містить величезну колекцію функцій і алгоритмів, починаючи від виявлення об'єктів і відстеження до машинного навчання та геометричних перетворень. Крім того, OpenCV пропонує ефективну продуктивність завдяки своїм оптимізованим алгоритмам і можливості використовувати багатопотоковість і прискорення GPU. Така ефективність продуктивності робить OpenCV надзвичайно корисним для програм реального часу та обробки великомасштабних наборів даних.

Для застосування OpenCV у практичних програмах потрібне глибоке розуміння концепцій комп'ютерного зору та алгоритмів. Крім того, основна увага OpenCV – це низькорівнева обробка зображень і операції комп'ютерного зору, де відсутні високорівневі абстракції, які зазвичай зустрічаються в бібліотеках машинного навчання. Це може вимагати інтеграції OpenCV з іншими бібліотеками для впровадження наскрізних конвеєрів машинного навчання. Іншим потенційним недоліком OpenCV є його документація та API, які, незважаючи на надання прикладів, можуть не мати деталей чи чітких інструкцій щодо використання. Це

може ускладнити для досвідчених користувачів навігацію та розуміння різних функцій та їхніх параметрів. В результаті, хоча OpenCV забезпечує детальний контроль над завданнями обробки зображень, його акцент на низькорівневих операціях може збільшити складність обробки зображень, потенційно призводячи до потреби більше зусиль, ніж фреймворки, які абстрагуються від складності.

3.3 Бібліотека Tensorflow

TensorFlow – популярна платформа машинного навчання з відкритим кодом, розроблена Google. Завдяки TensorFlow користувачі мають універсальну платформу, яка може з легкістю виконувати масштабні розподілені обчислення. Фреймворк широко використовується в багатьох дисциплінах, таких як глибоке навчання, обробка природної мови, комп'ютерний зір та багато іншого. TensorFlow надає можливість будувати діаграми потоку даних, які зображують рух даних через граф, що складається з різних вузлів обробки, кожен з яких представляє математичну операцію. Ці вузли з'єднані один з одним через багатовимірні масиви даних або тензори. Програми на основі TensorFlow можна запускати на різноманітних цілях, таких як локальні машини, хмарні кластери або мобільні пристрої, які використовують платформи iOS та Android, або процесори та графічні процесори. Використовуючи Google Cloud, розробники можуть використовувати спеціальний процесор TensorFlow (TPU) для максимального прискорення. Незважаючи на те, що моделі, згенеровані TensorFlow, можна розгорнути на більшості пристроїв для надання прогнозів, його найбільша перевага для розробки машинного навчання полягає в тому, що він абстрагується від складних деталей, пов'язаних із реалізацією алгоритму та зв'язуванням функцій. Це дозволяє розробникам зосередитися на логіці програми вищого рівня, залишаючи TensorFlow зосередитися на обробці базових технічних деталей. Крім того, бібліотека надає функції для налагодження та отримання інформації про додатки TensorFlow, у яких кожну операцію з графіком можна виокремити та ретельно

перевірити окремо, а не через один об’єкт. Ця функція, яка називається «режим швидкого виконання» тепер є стандартною у нових версіях Tensorflow.

Одним з недоліків TensorFlow є його докладний синтаксис, особливо з API нижчого рівня. Це може призвести до написання більшої кількості рядків коду порівняно з платформами вищого рівня, що призведе до більшої та потенційно складнішої кодової бази. Крім того, побудова та виконання обчислювальних графіків може передбачати написання більш шаблонного коду, що може зайняти багато часу та трудомісткість. Налагодження моделей TensorFlow також може бути складним через складний характер фреймворку. Виявлення та вирішення таких проблем, як неправильні форми тензорів або неправильне використання операцій, може бути складним і займати багато часу. Незважаючи на це, TensorFlow надає користувачам інструменти візуалізації та інструменти для налагодження моделей.

3.4 Бібліотека Keras

Keras – це потужна бібліотека глибокого навчання, яка доступна як програмне забезпечення з відкритим кодом і написана на Python. Вона спеціально розроблений, щоб бути зручним і модульним, з можливістю розширення, що дозволяє розробникам швидко створювати прототипи та розробляти моделі глибокого навчання. Keras надає високорівневий інтерфейс прикладного програмування (API), який абстрагує складності, пов’язані на нижчих рівнях із фреймворками глибокого навчання (наприклад, TensorFlow, Theano тощо), що полегшує людям, які лише починають глибоке навчання. почати свій навчальний процес. Завдяки широкому спектру попередньо визначених шарів, функцій активації та алгоритмів оптимізації Keras дозволяє легко будувати складні мережі шляхом укладання та з’єднання модульних будівельних блоків, які називаються шарами. Ця модульність також дозволяє користувачам створювати власні шари, функції втрат і показники, що робить Keras надзвичайно розширюваним. Крім того, Keras пропонує гнучкість бекенда, дозволяючи користувачам вибрати найкращий

варіант бекенда для своїх потреб із TensorFlow, Theano та Microsoft Cognitive Toolkit (CNTK). Загалом Keras є ідеальним вибором для дослідників, які шукають універсальну структуру глибокого навчання, яка поєднує простоту та потужність.

3.5 Бібліотека TorchVision

PyTorch – це популярна платформа глибокого навчання з відкритим вихідним кодом, яка була розроблена дослідницькою лабораторією штучного інтелекту Facebook як гнучка, ефективна та проста у використанні. Створена на основі бібліотеки Torch, спочатку створеної для мови Lua, дослідники та розробники з усього світу можуть отримати доступ до PyTorch, оскільки вона пропонує зручний інтерфейс Python. Вона забезпечує динамічні обчислювальні графіки, які будуються в режимі реального часу під час виконання. Ця функція є унікальною для PyTorch і робить його більш гнучким і динамічним порівняно з іншими фреймворками. Динамічний графік дозволяє вносити зміни в архітектуру моделі, що допомагає розробникам легко налагоджувати свій код.

Ще одна особливість PyTorch полягає в тому, що він має повну підтримку прискорення графічного процесора, що дозволяє швидше навчатися та робити висновки щодо графічних процесорів. Ця функція призводить до значного підвищення швидкості завдань глибокого навчання, оскільки PyTorch використовує CUDA, платформу паралельних обчислень, щоб використовувати обчислювальну потужність графічних процесорів. Крім того, PyTorch легше налагоджувати та контролювати порівняно з іншими фреймворками. Розробники можуть перевіряти тензори, аналізувати значення градієнтів і налагоджувати свій код за допомогою стандартних інструментів налагодження Python.

Додатково, PyTorch надає інструменти візуалізації, такі як TensorBoardX, які дозволяють розробникам візуалізувати продуктивність моделі, прогрес навчання та інші аспекти розробки моделі за допомогою інтерактивних графіків і показників.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Збір навчальних даних та їх нормалізація

Для забезпечення успішних і точних результатів за допомогою нейронних мереж у машинному навчанні вкрай важливо ефективно підготувати навчальні дані. Якість вхідних даних має значний вплив на вихідну модель, і оптимізація даних може бути важливішою, ніж оптимізація самої моделі. Сучасні моделі нейронних мереж здатні самостійно ідентифікувати нелінійні шаблони, але для правильної ідентифікації особливостей все ще потрібна значна кількість позначених даних. Таким чином, життєво важливо забезпечити алгоритм актуальною та корисною інформацією, щоб збільшити шанси на успіх під час навчання моделі нейронної мережі. Правильна підготовка не тільки покращує якість моделі, але й економить багато часу в довгостроковій перспективі, а також дозволяє уникнути помилок, які можуть виникнути пізніше.

Для розробки класифікатора системи розпізнавання дорожніх знаків використовувася набір даних GTSRB. Набір даних містить зображення 43 різних класів дорожніх знаків, включаючи знаки обмеження швидкості, знаки зупинки, знаки пріоритету, знаки небезпеки та інші. Кожен клас має унікальний ідентифікатор. Набір даних містить близько 50 000 зображень, розділених на тренувальну та тестову вибірки. Тренувальна вибірка містить близько 39 000 зображень, а тестова - близько 12 000 зображень. Зображення дорожніх знаків у наборі даних GTSRB мають різні розміри, але зазвичай вони знаходяться у діапазоні від 15x15 до 250x250 пікселів. Кожне зображення супроводжується анотацією, яка містить клас дорожнього знаку та іншу інформацію, таку як позиція та розмір знаку на зображенні.

При роботі з великим набором даних, важливо переконатися, що дані нормалізовані. Нормалізація стосується процесу масштабування значень вхідних характеристик до діапазону від 0 до 1. Це вкрай важливо зробити, оскільки діапазон значень ознак у вихідному наборі даних може змінюватися від 0 до 255, що може

призвести до дисбалансу між функціями під час надання даних до нейронної мережі. Цей дисбаланс може суттєво вплинути на продуктивність моделі, що призведе до недосконалих результатів класифікації у майбутньому.

Розподіл зображень за класами представлений на рисунку 4.1.

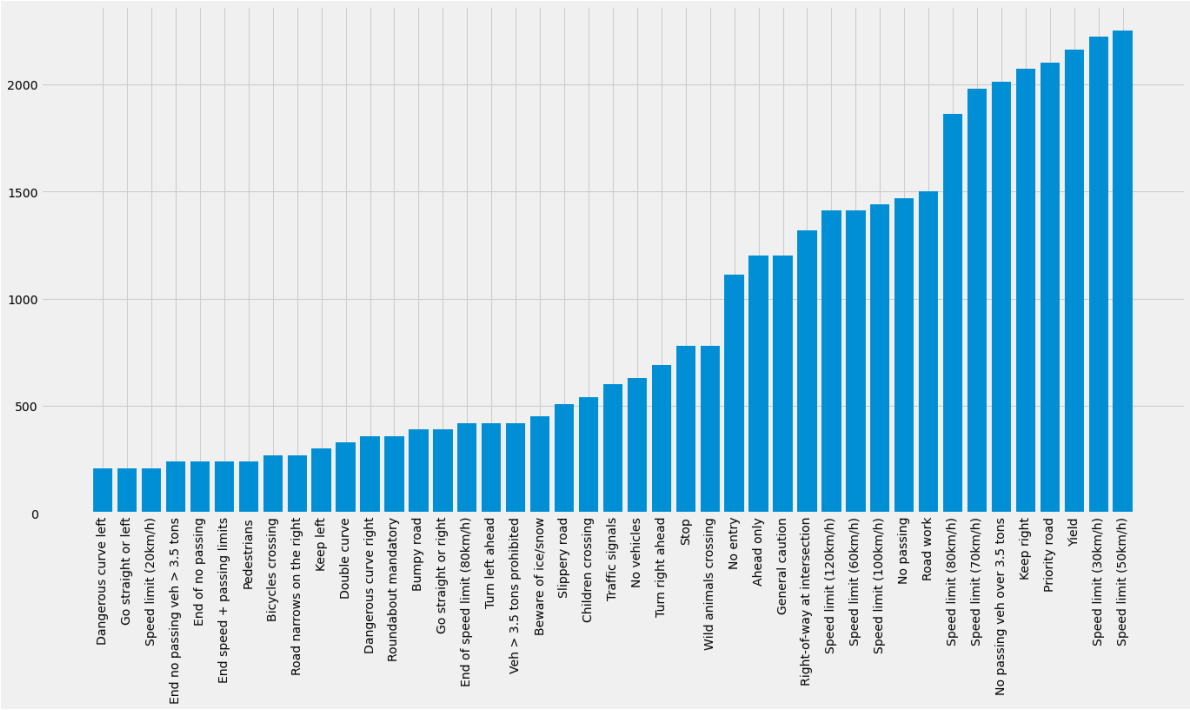

Рисунок 4.1 – Розподіл зображень за класами в наборі даних GTSRB

На рисунку 4.2 представлений приклад виборки з 10 випадкових зображень з тренувального набору даних.

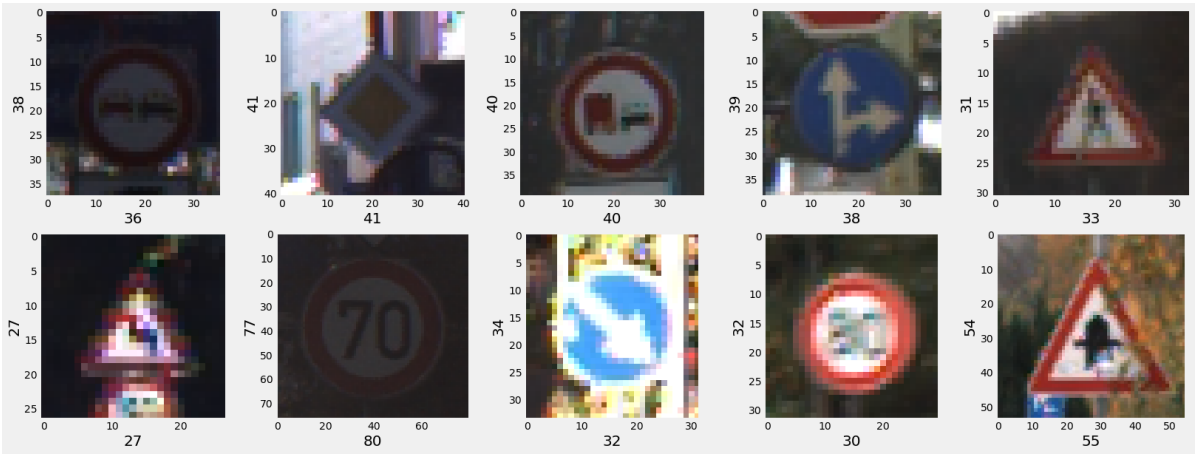

Рисунок 4.2 – Набір випадкових зображень з тренувального набору даних

4.2 Архітектура розробленої нейронної мережі

Розроблена архітектура нейронної мережі складається з кількох рівнів, які разом виконують аналіз зображень. Зокрема, існує 3 шари згортки, 2 рівні максимального об’єднання, рівень підсумовування та 2 повністю зв’язані рівні. Перший згортковий рівень відповідає за згортання вхідного зображення з 30 різними згортковими ядрами. Ці ядра відіграють вирішальну роль у виборі відповідних функцій із вхідного зображення, які потім використовуються для створення набору карт функцій. Розмір цих ядер є важливим фактором у визначенні загальної продуктивності нейронної мережі. Архітектура нейронної мережі використовує загальний розмір ядра 3x3 для першого згорткового рівня. Отримані карти функцій потім передаються на наступний шар для подальшої обробки. Таким чином, нейронна мережа може вивчати особливості вхідного зображення та виявляти закономірності для прогнозів.

Під час роботи зі згортковою нейронною мережею визначення розміру вхідних даних є вирішальним кроком. Процес визначення розміру починається з встановлення розміру зображення, як правило, 30 на 30 пікселів, із глибиною кольору 3 для першого згорткового шару. У міру просування мережі наступні розміри вхідних даних визначаються автоматично на основі вихідних вимірювань попередніх шарів. Важливо брати до уваги три основні параметри, коли йдеться про вхідний розмір у згорткових шарах: кількість рядків і стовпців у матриці зображення та кількість кольорових каналів, які вона містить. Варто зазначити, що навіть для зображень у відтінках сірого кількість кольорових каналів дорівнює одному, а не зовсім відсутні. Під час операції стиснення, заповнення має значення для збереження розміру зображення.

У сфері навчання нейронної мережі існує певний ступінь пакетної нормалізації, яка використовується для нормалізації вхідних даних. Під час навчання вага шару може змінюватися, таким чином змінюючи його розподіл балів і згодом змушуючи наступний шар проходити подальше навчання. Це явище

називається змінним зміщенням і може бути виправлено за допомогою окремої нормалізації кожного вхідного сигналу, що виконується за допомогою пакетів зображень, а не всього набору даних. Крім того, нові параметри, відомі як зсув і розтягування, проходять навчання за допомогою градієнтного спуску.

Функція активації – це математична операція, яка використовується в штучних нейронних мережах для перетворення виходу суматора у відповідний вихід, який вирішує конкретну задачу. Вона бере загальну суму зважених вхідних даних і застосовує математичне перетворення для отримання результату. Концепція функції активації емулює реальний нейрон, тому що коли функція підсумовування повертає мале значення, функція активації може створити нульовий результат, який можна порівняти з відсутністю збудження біологічних нейронів. Функція активації Rectified Linear Unit (ReLU) повертає вхідне значення, якщо воно позитивне, і повертає нуль, якщо воно негативне. Нижче представлений математичний вигляд функції активації (4.1):

$$f(x) = \max(0, x) \qquad (4.1)$$

У процесі навчання функція ReLU (Rectified Linear Unit) має перевагу над сигмоподібними та гіперболічними функціями, оскільки призводить до створення розрідженості в активації нейронів, що означає, що не всі нейрони активуються, що зменшує складність обчислень.

У нейронних мережах низька щільність часто пояснюється відсутністю негативних значень. Однак Rectified Linear Unit (ReLU) може ефективно підвищити швидкість конвергенції градієнтного спуску порівняно з іншими функціями активації. Ще один спосіб підвищити обчислювальну ефективність – включити рівень Maxpooling у блок згортки, який зазвичай встановлюється на розмір за замовчуванням 2*2, але може бути налаштований відповідно до вимог користувача.

Після рівня Maxpooling використовується ядро згортки 3*3 із 64 фільтрами, що використовує ReLU як функцію активації.

На рисунку 4.3 представлена архітектура розробленої згорткової нейронної мережі для розпізнавання дорожніх знаків.

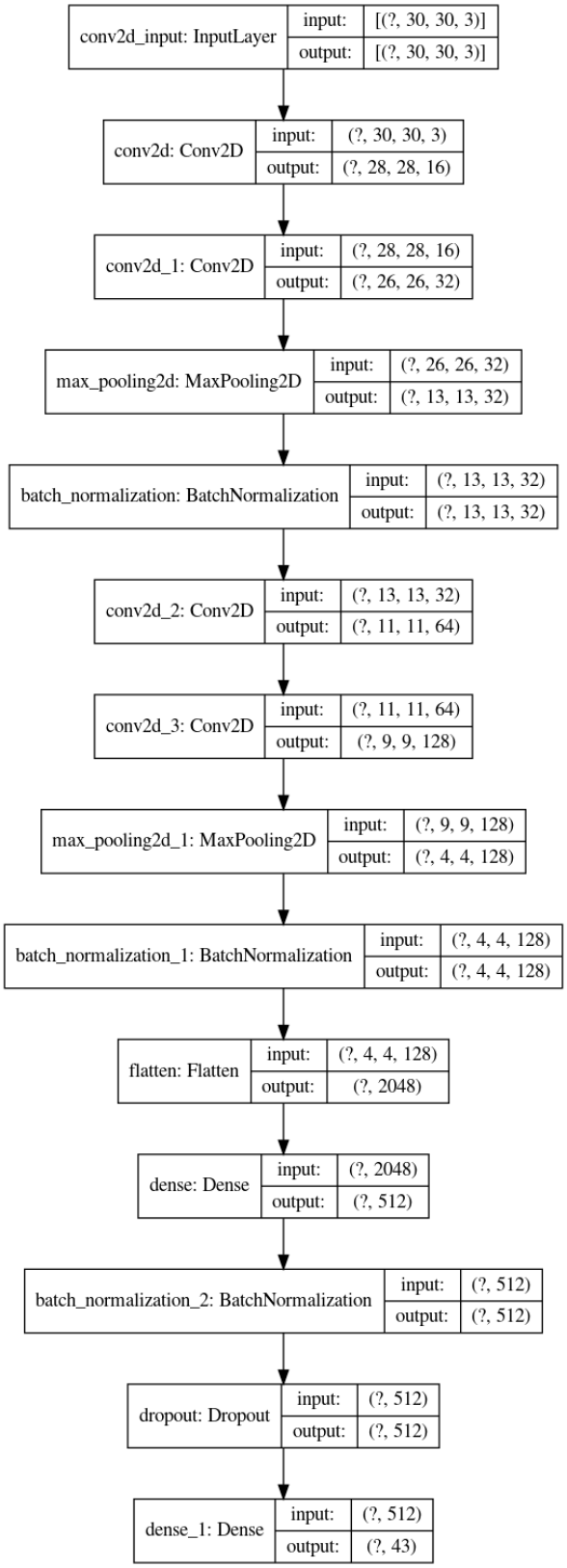

Рисунок 4.3 – Архітектура розробленої нейронної мережі

У нейронній мережі шар Flatten йде після двох блоків згортки. Цей рівень не має параметрів, і його основне призначення – перетворення багатовимірних даних в одновимірний вектор. Цей вектор можна використовувати як вхідні дані для повнозв'язного шару. Важливо відзначити, що згорткові блоки не здатні самостійно класифікувати зображення. Повнозв'язний рівень виконує це завдання. Архітектура містить два повністю зв'язані рівні, причому перший рівень має 132 входи на нейрон, тоді як другий рівень має 7 входів, що відповідає кількості класів, які потрібно передбачити. Загалом, зведений вектор із шару Flatten використовується повністю підключеним шаром для класифікації зображення за певними категоріями. На графічному матеріалі ІС93.150БАК.004 Д2 представлений алгоритм розпізнавання дорожніх знаків з використанням розробленої згорткової нейронної мережі та класифікаторів.

Запропонований підхід для ідентифікації дорожніх знаків на зображеннях є злиттям двох різних методів: опоних векторів (SVM) і згорткової нейронних мереж (CNN). Комбінація працює таким чином: спочатку метод SVM перевіряє, чи присутній на зображенні дорожній знак чи ні. Якщо виявлено дорожній знак, зображення потім передається на CNN для прийняття остаточного рішення щодо його ідентифікації. Цей синтез методів SVM і CNN забезпечує точні показники розпізнавання, навіть якщо зображення спотворені, зроблені під різними кутами або за несприятливих погодних умов. Алгоритм розпізнавання дорожніх знаків складається з двох етапів: навчання та розпізнавання. На першому етапі процес навчання включає кілька кроків, а саме вибір компонентів зображень дорожніх знаків у навчальному наборі даних за допомогою SVM, зменшення розмірності вибраних векторів ознак за допомогою опорних векторів і навчання CNN на згенерованих векторах ознак. На другому етапі CNN перевіряє наявність дорожнього знака на зображенні, щоб усунути помилки розпізнавання.

4.3 Налаштування та метрики тестування моделі

Після побудови моделі важливо точно налаштувати її за допомогою спеціальних інструментів, таких як показники якості, оптимізатори та функція втрат. У контексті класифікації трьох або більше груп найбільш прийнятною функцією втрат часто є категоріальна крос-ентропія. Ця конкретна функція полегшує визначення дисперсії між двома розподілами ймовірностей і вимірює ступінь точності, який існує між очікуваним і прогнозованим розподілами. Математична формула для розрахунку представлена нижче (4.2):

$$Cross - Entropy\ Loss = \sum_{i=1}^{\text{розмір виводу}} y_i * \log \widehat{y_i} \qquad (4.2)$$

де $\widehat{y_i}$ – скалярне значення вектору на виході моделі, y_i – цільове значення, а розмір виводу – кількість скалярних значень на виході моделі.

Adam – це широко використовуваний оптимізатор для моделей глибокого навчання, який ефективно регулює вагу цих моделей шляхом поєднання концепцій двох інших оптимізаторів, RMSprop і Momentum. Щоб забезпечити ефективний процес навчання, Адам використовує дві основні концепції: експоненціальне ковзне середнє градієнтів (EMA) і експоненціальне ковзне середнє квадратів (EMA). Оптимізатор Adam проходить кілька етапів, включаючи ініціалізацію, обчислення градієнтів, оновлення швидкості навчання та експоненціальне оновлення середнього. У процесі оптимізації оптимізатор використовує адаптивну швидкість навчання, яка залежить від середньоквадратичних градієнтів.

У сфері категоризації індикатором якості, який зазвичай використовується, є показник точності. Цей показник показує частоту, з якою прогнози моделі є правильними. Простіше кажучи, точність – це частка точних прогнозів, зроблених моделлю, порівняно з усіма іншими зробленими прогнозами.

Формула представлена нижче (4.3):

$$\text{Точність} = \frac{\text{Кількість правильних прогнозів}}{\text{Загальна кількість прогнозів}} \quad (4.3)$$

4.4 Навчання нейронної мережі

Коли йдеться про навчання нейронної мережі, процес передбачає використання певного набору даних, який може допомогти підвищити точність прогнозів. Але, щоб переконатися, що навчена мережа працює ефективно, важливо протестувати її за допомогою окремого набору даних. Однак надання всієї вибірки даних одночасно може бути неефективним для розпізнавання зображення. Таким чином, рішення полягає в тому, щоб розділити дані на сегменти, кожен з яких містить 64 елементи, і навчити мережу за допомогою кількох ітерацій кожного сегмента. Зі збільшенням розміру вибірки для її обробки потрібно більше пам’яті комп’ютера. Модель тренується протягом 30 епох для досягнення оптимальних рівнів продуктивності, і дуже важливо уникати перетренування. Перенавчання виникає, коли модель стає надто зосередженою на розпізнаванні прикладів із навчального набору та не в змозі розпізнати інші приклади з тестового набору.

На рисунку 4.4 представлені результати метрики точності розробленої нейронної мережі на тестових та навчальних даних з датасету GTSRB.

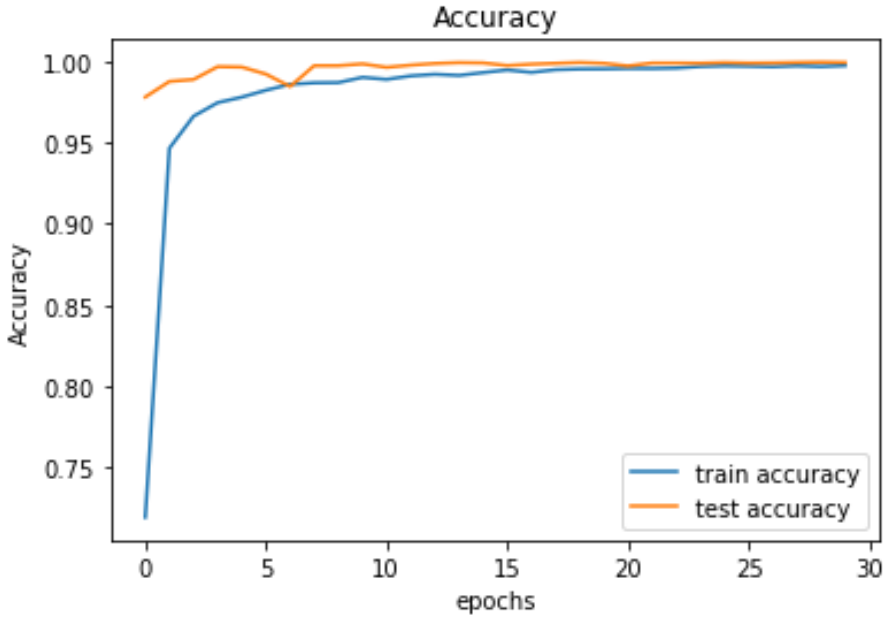

Рисунок 4.4 – Метрика точності розробленої моделі нейронної мережі

На рисунку 4.5 представлені результати метрики функції втрат розроблено моделі нейронної мережі на тестових та навчальних даних з датасету GTSRB.

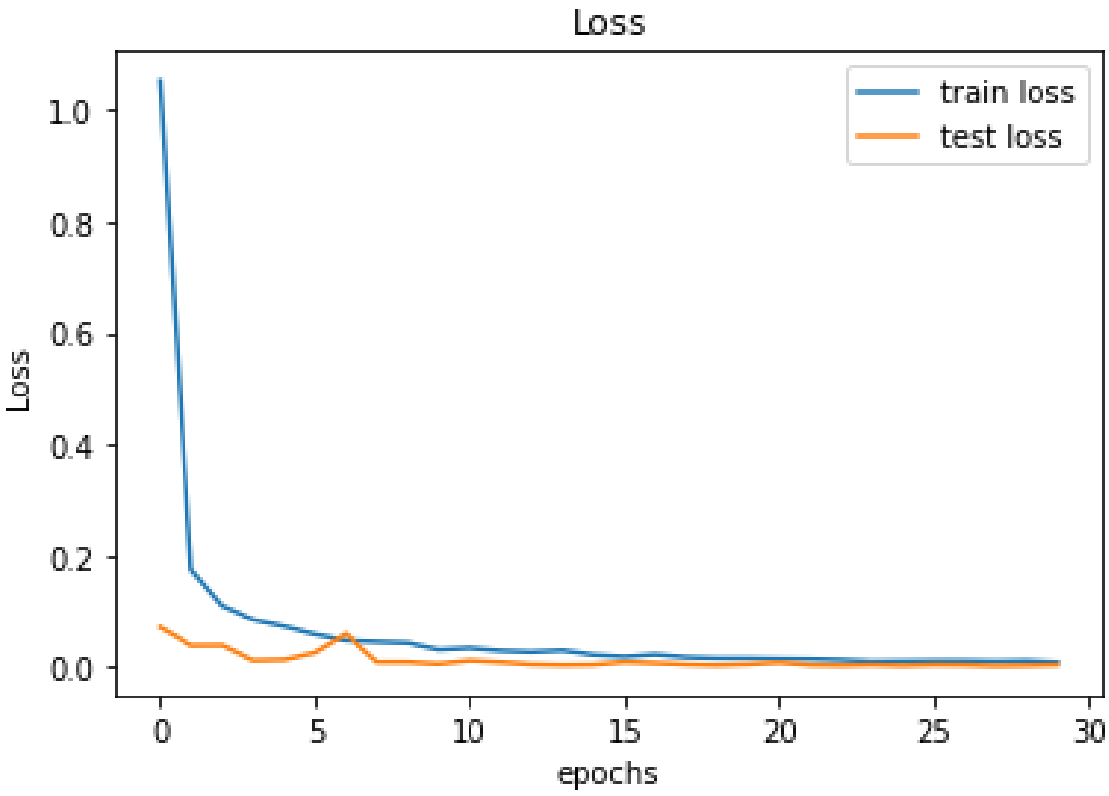

Рисунок 4.5 – Метрика втрат розробленої моделі нейронної мережі

З рисунку 4.4 видно, що модель нейронної мережі на тестовому сегменті навчальних даних з перших епох навчання демонструє точність 97%, що підтвердує ефективність обраної архітектури згорткової нейронної мережі.

4.5 Тестування та інтеграція моделі нейронної мережі

Після проведення тестування нейронної мережі на тестових та валідаційних даних, проведемо тестування розпізнавання дорожніх знаків. На рисунку 4.5 представлений результат розпізнавання 15 зображень з набору даних GTSRB з використанням розробленого класифікатора.

Рисунок 4.6 – Результат розпізнавання тестових зображень

Як видно з рисунку 4.6, модель розпізнала всі дорожні знаки з тестового набору даних без помилок, що говорить про високу ефективність та точність моделі нейронної мережі у розпізнаванні дорожних знаків. Це підтверджує високу якість навчання моделі та її здатність до узагальнення на нових даних

Далі, протестуємо модель на відеофрагменті, який містить 3 знаки для розпізнавання. Результати класифікації представлені на рисунках 4.7 – 4.10.

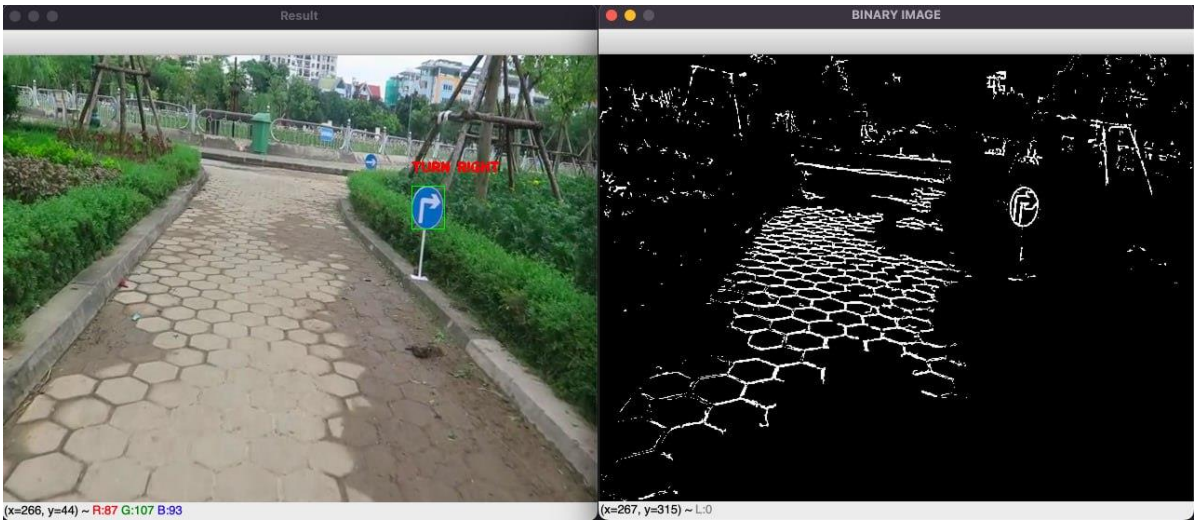

Рисунок 4.7 – Результат класифікації знаку “Поворот праворуч”

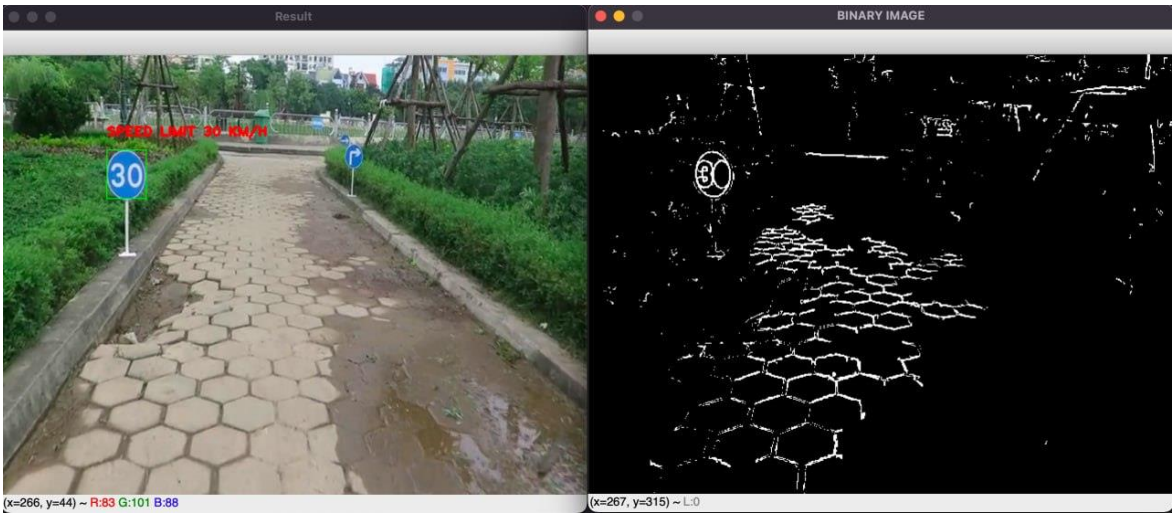

Рисунок 4.8 – Результат класифікації знаку “Мінімальна швидкість 30 км/год”

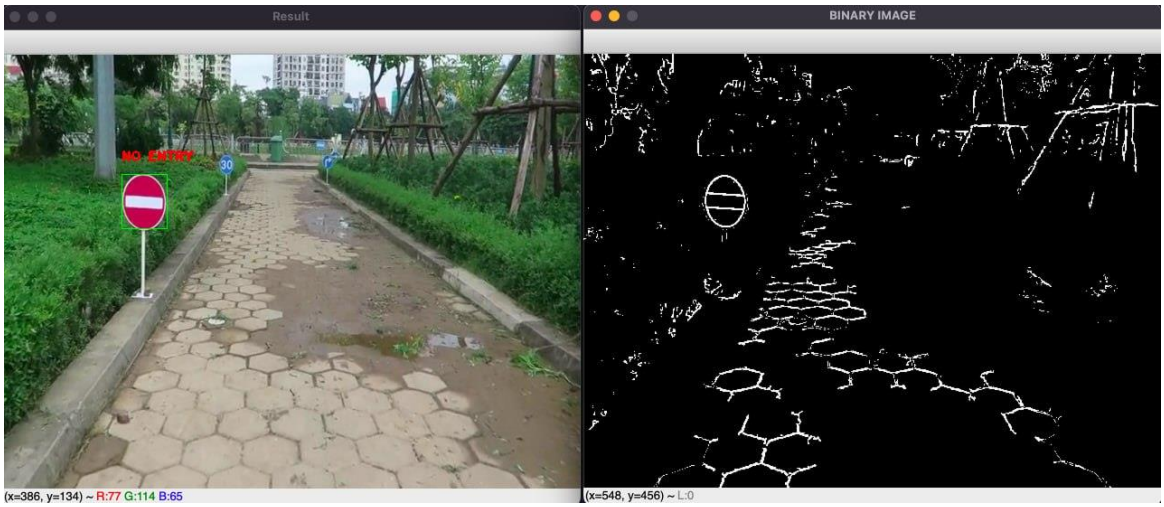

Рисунок 4.10 – Результат класифікації знаку “Проїзду немає”

З рисунків 4.7 – 4.10 видно, що розроблений підхід з використаннями методу SVM та методу згорткових нейронних мереж виявився ефективним рішенням для розпізнавання дорожніх знаків. Модель згорткової нейронної мережі добре справляється з розпізнаванням знаків та присвоює вірні класи. Використання комбінації методу SVM та методу CNN дозволяє отримати кращі результати в розпізнаванні дорожніх знаків. SVM виконує ефективну класифікацію, використовуючи розміщення в ознаковому просторі, тоді як CNN дозволяє автоматично витягти ознаки зі зображень.

Висновки до розділу

В даному розділі було наведено опис програмної реалізації системи розпізнавання дорожніх знаків з використанням комбінованого підходу на основі SVM та згорткових нейронних мереж. Метод SVM використовується для розділення класів дорожніх знаків у просторі ознак, в той час як CNN використовується для автоматичного витягування корисних ознак з вхідних зображень. Проведено тестування розробленої моделі нейронної мережі на тестових зображенях та загальної програмної системи на відеофрагментах.

ВИСНОВКИ

У ході виконання дипломного проєкту було розроблено програмну систему для розпізнавання дорожніх знаків з використанням машинного навчання. Метою проєкту було поліпшення ефективності та точності розпізнавання знаків на дорозі, забезпечуючи безпеку дорожнього руху.

У ході розробки було проведено аналіз предметної області та існуючих методів розпізнавання дорожніх знаків. Були визначені функціональні та нефункціональні вимоги до системи, обрані необхідні технології та методи. Після спроектування системи було реалізовано програмний функціонал, який дозволяє виявляти та класифікувати дорожні знаки на основі навченої моделі.

Розроблена програмна система використовує методи машинного навчання, зокрема згорткові нейронні мережі, для автоматичного розпізнавання дорожніх знаків на зображеннях або відео. Система навчається на великому наборі даних, який містить різноманітні знаки та їх класифікацію.

У результаті виконання дипломного проєкту було розроблено програмну систему для розпізнавання дорожніх знаків з використанням машинного навчання. Розроблена система має потенціал для подальшого розширення та використання в різних сферах, пов'язаних з безпекою дорожнього руху, а також спрощення інформування водіїв про дорожні знаки.

Для можливого подальшого розвитку та популяризації програмної системи для розпізнавання дорожніх знаків з використанням машинного навчання пропонуються наступні дії та покращення:

– Додавання додаткового функціоналу до системи, наприклад, розпізнавання та аналіз тексту на дорожніх знаках, виявлення порушень правил дорожнього руху або надання рекомендацій.

– Подальше удосконалення алгоритмів розпізнавання дорожніх знаків з метою підвищення швидкості та точності системи. Використання новітніх методів машинного навчання покращить продуктивність системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Traffic Sign Recognition. URL: https://www.sygic.com/what-is/sign-recognition (дата звернення: 17.05.2023).

2. B. Smitha Shekar and G. Harish. A Machine Learning Model For Detection and Recognition Of Traffic Signs. *Conference on Intelligent Technologies*. 2021. pp. 1-4. URL: https://ieeexplore.ieee.org/document/9498270 (дата звернення: 17.05.2023).

3. Understanding ADAS: Traffic Sign Recognition. URL: https://caradas.com/adas-traffic-sign-recognition/ (дата звернення: 17.05.2023).

4. Eichner M. Integrated speed limit detection and recognition from real-time video. *IEEE International Intelligent Vehicles Symposium*. 2009, № 8, P. 626-631.

5. A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda and P. Hanagal, "Traffic Sign Detection and Recognition using a CNN Ensemble. 2019 IEEE International Conference on Consumer Electronics (ICCE). 2019. pp. 1-4. URL: https://ieeexplore.ieee.org/document/8662019 (дата звернення: 17.05.2023).

6. A. Barodi, A. Bajit, M. Benbrahim and A. Tamtaoui,. An Enhanced Approach in Detecting Object Applied to Automotive Traffic Roads Signs. IEEE 6th International Conference on Optimization and Applications (ICOA). 2020. pp. 1-6. URL: https://ieeexplore.ieee.org/document/9094457 (дата звернення: 20.05.2023).

7. R. Kukreja, S. Rinchen, B. Vaidya and H. T. Mouftah, "Evaluating Traffic Signs Detection using Faster R-CNN for Autonomous driving. *IEEE 25th International Workshop on Computer Modeling and Design of Communication Networks.* 2020. pp. 1-6. URL: https://ieeexplore.ieee.org/document/9209289 (дата звернення: 20.05.2023).

8. X. Changzhen, W. Cong, M. Weixin and S. Yanmei. A traffic sign detection algorithm based on deep convolutional neural network. *IEEE International Conference Image Processing (ICSIP), Beijing, China*. 2016. pp. 676-679. URL: https://ieeexplore.ieee.org/document/7888348 (дата звернення: 17.05.2023).

9. GTSRB – German Traffic Sign Recognition Benchmark. URL: https://paperswithcode.com/dataset/gtsrb/analysis (дата звернення: 20.05.2023).

10. N. Bhatt, P. Laldas and V. B. Lobo. A Real-Time Traffic Sign Detection and Recognition System on Hybrid Dataset using CNN. *2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India*. 2022, pp. 1354-1358, doi: 10.1109/ICCES54183.2022.9835954.

11. C. Liu, F. Chang and Z. Chen. High performance traffic sign recognition based on sparse representation and SVM classification. *2014 10th International Conference on Natural Computation (ICNC), Xiamen, China*. 2014, pp. 108-112.

12. K. Sukhani, R. Shankarmani, J. Shah and K. Shah. Traffic Sign Board Recognition and Voice Alert System using Convolutional Neural Network. *International Conference for Emerging Technology*. 2021. pp. 1-5. URL: https://ieeexplore.ieee.org/document/9456302 (дата звернення: 22.05.2023).

13. V. A. Prisacariu, R. Timofte, K. Zimmermann, I. Reid and L. Van Gool. Integrating Object Detection with 3D Tracking Towards a Better Driver Assistance System. *International Conference on Pattern Recognition*. 2010. pp. 3344-3347. URL: https://ieeexplore.ieee.org/document/5597510 (дата звернення: 22.05.2023).

14. W. Yang and W. Zhang. Real-time Traffic Signs Detection Based on YOLO Network Model. *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. 2020. pp. 354-357. URL: https://ieeexplore.ieee.org/document/9329389 (дата звернення: 22.05.2023).

15. P. Tumuluru, L. R. Burra, N. Sunanda, S. S. Hussain, D. Madhu and H. V. Varma. SMS: SIGNS MAY SAVE – Traffic Sign Recognition and Detection using CNN. *International Conference on Electronics*. 2022. pp. 1272-1277. URL: https://ieeexplore.ieee.org/document/10009638 (дата звернення: 26.05.2023).

16. Traffic Signs Recognition using CNN and Keras in Python. URL: https://www.geeksforgeeks.org/tsr-using-cnn (дата звернення: 26.05.2023).

17. M. Swathi and K. V. Suresh. Automatic traffic sign detection and recognition in video sequences. *IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. 2017. pp. 476-480. URL: https://ieeexplore.ieee.org/document/8256642 (дата звернення: 26.05.2023).