

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Розробка інформаційної системи для розпізнавання та ідентифікації
зображень номерів транспортних засобів

Виконав: студент VI курсу, групи СНм-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Соломка Р.Р.

(прізвище та ініціали)

Керівник

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Мацюк О.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Петрик М.Р.

(прізвище та ініціали)

Тернопіль
2022

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Мацюк О.В., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач		

7. Дата видачі завдання 14 листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	14.11.2022-15.11.2022	Виконано
2.	Підбір наукових джерел про алгоритми розпізнавання номерних знаків автотранспорту	16.11.2022-20.11.2022	Виконано
3.	Переклад та опрацювання наукових джерел про методи ідентифікації номерів транспортних засобів	21.11.2022-23.11.2022	Виконано
4.	Виконання дослідження щодо виконання оптичного розпізнавання символів номерного знаку автомобіля	24.11.2022-27.11.2022	Виконано
5.	Оформлення розділу «Аналіз предметної області»	28.11.2022-30.11.2022	Виконано
6.	Оформлення розділу «Аналіз методів та проектування для розпізнавання номерів транспортних засобів»	01.12.2022-04.12.2022	Виконано
7.	Оформлення розділу «Проектування системи для розпізнавання та ідентифікації номерів транспортних засобів»	05.12.2022-07.12.2022	Виконано
8.	Виконання завдання до підрозділу «Охорона праці»	08.12.2022-09.12.2022	Виконано
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	10.12.2022-11.12.2022	Виконано
10.	Оформлення кваліфікаційної роботи	12.12.2022-13.12.2022	Виконано
11.	Нормоконтроль	14.12.2022-15.12.2022	Виконано
12.	Перевірка на плагіат	15.12.2022	Виконано
13.	Попередній захист кваліфікаційної роботи	16.12.2022	Виконано
14.	Захист кваліфікаційної роботи	20.12.2022	

Студент

_____ (підпис)

Соломка Р.Р.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Литвиненко Я.В.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Розробка інформаційної системи для розпізнавання та ідентифікації зображень номерів транспортних засобів // Кваліфікаційна робота освітнього рівня «Магістр» // Соломка Руслан Романович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2022 // С. 71, рис. – 36, додат. – 2, бібліогр. – 50.

Ключові слова: інформаційна система, розпізнавання та ідентифікація, номерний знак транспортного засобу.

Кваліфікаційна робота присвячена розробці інформаційної системи для розпізнавання та ідентифікації зображень номерів транспортних засобів.

В першому розділі кваліфікаційної роботи висвітлено принципи роботи інформаційних систем розпізнавання та ідентифікації зображень номерів транспортних засобів. Описано актуальність даних систем у сучасному світі. Проаналізовано існуючі системи розпізнавання номерних знаків.

В другому розділі кваліфікаційної роботи досліджено найпоширеніші алгоритми для виконання розпізнавання номерних знаків та виконання оптичного розпізнання символів.

В третьому розділі кваліфікаційної роботи описано процес створення інформаційної системи для розпізнавання та ідентифікації зображень номерів транспортних засобів, та спроектовано веб застосунок у якості графічного інтерфейсу.

ANNOTATION

An Information System Development To Recognize and Identify Images Of Vehicles Numbers Plates // Qualification work of the educational level "Master" // Solomka Ruslan Romanovich // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNm-61 group // Ternopil, 2022 // P. 71, fig. - 36, annexes - 2, references. - 50.

Key words: information system, recognition and identification, vehicle number plate.

The qualification work is devoted to the development of an information system for recognition and identification of images of vehicle license plates.

In the first section of the qualification work, the principles of information systems for recognition and identification of images of vehicle license plates are highlighted. The relevance of these systems in the modern world is described. Existing license plate recognition systems were analyzed.

In the second section of the qualification work, the most common algorithms for license plate recognition and optical character recognition were investigated.

In the third section of the qualification work, the process of creating an information system for recognizing and identifying images of vehicle license plates is described, and a web application is designed as a graphical interface.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

OCR (англ. Optical Character Recognition) – оптичне розпізнавання символів.

MSE (англ. Mean Square Error) – середньоквадратична помилка.

HTML (англ. Hyper Text Markup Language) – мова розмітки гіпертексту.

API (англ. Application Programming Interface) – прикладний програмний інтерфейс.

HTTP (англ. Hyper Text Transfer Protocol) – протокол передачі гіпертекстових документів.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Ознайомлення із системами розпізнавання номерних знаків.....	10
1.2 Труднощі при ідентифікації номерних знаків автомобілів.....	12
1.3 Використання систем розпізнавання номерних знаків	14
1.4 Висновок до першого розділу	17
2 АНАЛІЗ МЕТОДІВ ТА ПРОЄКТУВАННЯ ДЛЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ НОМЕРІВ ТРАНСПОРТНИХ ЗАСОБІВ.....	18
2.1 Методи розпізнавання номерних знаків автотранспорту	18
2.2 Методи розробки систем нейронних мереж.....	23
2.3 Методи виконання оптичного розпізнавання символів	28
2.4 Висновок до другого розділу	31
3 ПРОЄКТУВАННЯ СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ТА ІДЕНТИФІКАЦІЇ ЗОБРАЖЕНЬ НОМЕРІВ ТРАНСПОРТНИХ ЗАСОБІВ.....	32
3.1 Структура розроблення інформаційної системи.....	32
3.2 Підготовка тестових даних для інформаційної системи	34
3.3 Попередня обробка тестових даних	39
3.4 Проєктування моделі нейронної мережі	41
3.4.1 Створення та налаштування нейронної мережі	41
3.4.2 Тренування нейронної мережі	43
3.4.3 Тестування нейронної мережі.....	46
3.5 Виконання оптичного розпізнавання символів.....	50
3.6 Веб застосунок для розпізнавання номерів транспортних засобів	51
3.7 Висновок до третього розділу	57
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	58
4.1 Гігієнічна оцінка умов праці програміста.....	58

4.2 Підвищення стійкості роботи об'єктів логістики у воєнний час	61
4.3 Висновки до четвертого розділу	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ДЖЕРЕЛ.....	68
ДОДАТКИ	

ВСТУП

Актуальність теми. У міру збільшення кількості транспортних засобів, зростає також і необхідність в їх автоматичному регулюванні. Можливість розпізнавати та проводити ідентифікацію номерних знаків автотранспорту без прямої участі людини є основою сучасних дорожніх систем. Та незважаючи на велику кількість досліджень, у даній сфері досі присутня необхідність у покращенні технічних та алгоритмічних складових, через складнощі обробки зображень, та варіативності існуючих номерних знаків. Таким чином розроблення інформаційної системи автоматичного розпізнавання та ідентифікації номерів транспортних засобів із використанням сучасних технологій нейронних мереж та машинного навчання є актуальним напрямком наукових досліджень.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є підвищення рівня повноти подання інформації щодо автоматичного розпізнавання номерних знаків автотранспорту. Для досягнення поставленої мети було потрібно виконати наступні завдання:

- Проаналізувати стан досліджень в даній предметній області.
- Дослідити існуючі на даний час алгоритми ідентифікації номерних знаків на зображеннях.
- Проаналізувати методи оптичного розпізнавання символів.
- Розробити модель штучної нейронної мережі, виконати її навчання.
- Спроекувати веб застосунок у якості графічного інтерфейсу для розробленої інформаційної системи.

Об'єкт дослідження. автоматичне розпізнавання номерних знаків автотранспорту та їх ідентифікація.

Предмет дослідження. методи проведення автоматичного розпізнавання та ідентифікації номерів транспортних засобів.

Практичне значення одержаних результатів. Виконано розробку та проектування інформаційної системи автоматичного розпізнавання та ідентифікації номерів транспортних засобів.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на XI міжнародній науково-технічній конференції молодих учених та студентів «АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2022 р.).

Публікації. Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (Див. додатки А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 50 найменувань та 2 додатків. Загальний обсяг кваліфікаційної роботи складає 71 сторінки, з них 47 сторінки основного тексту, який містить 36 рисунків та 16 лістингів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Ознайомлення із системами розпізнавання номерних знаків

Автоматичне розпізнавання та ідентифікація зображень номерів транспортних засобів – це технологія, що використовує оптичне розпізнавання символів на зображеннях для зчитування номерних знаків автотранспорту та створення даних про місцезнаходження автомобіля. У сучасному світі інформаційні системи розпізнання та ідентифікації номерних знаків автомобілів найбільш поширено використовуються правоохоронними силами в усьому світі для адміністративних цілей, зокрема для перевірки того, чи транспортний засіб зареєстровано, або чи має він ліцензію.

На сьогодні існує багато прикладів подібного роду інформаційних систем. Однією з них є Eyedea, що являє собою високоточне програмне забезпечення автоматичного розпізнання номерів автотранспорту на основі функцій штучного інтелекту, спеціально навчених для зчитування номерних знаків. Ця система може виявляти та зчитувати всі номерні знаки, які присутні на зображенні, а також розпізнавати спеціальні символи та коди країн. Дана система підтримує як однорядкові, так і багаторядкові номерні знаки, а також номерні таблички вантажівок. Аналогічно деякі номерні знаки можуть мати різні графічні значки на початку, в середині або в кінці. Деякі можуть містити незвичні шрифти або унікальні формати. Основними перевагами саме цієї системи автоматичного розпізнання номерів автотранспорту є можливість розпізнання широкого діапазону форматів номерних знаків та розпізнавання усіх видимих знаків на зображенні.

Plate Recognizer – це ще один приклад автоматичної системи ідентифікації транспортних засобів. Дана система може налаштувати свій механізм розпізнавання відповідно до конкретних потреб, наприклад

спостереження під певним кутом камери чи умовою освітлення. Такий результат досягається шляхом використання двох різних нейронних мереж. Одна з яких ідентифікує фотографії номерних знаків із зображення, а інша розшифровує кожен символ номерного знаку. Проте ця система може повернути до 5 розшифрованих номерних знаків з одного зображення. Основними перевагами цієї системи є постійні покращення алгоритмів, для роботи з різними факторами, такими як сонячні відблиски, розмиті зображення, швидкісні транспортні засоби, нічний час і багато іншого. Також така система надає можливість роботи в реальному часі.

Першу систему автоматичного розпізнання зображень номерів транспортних засобів було винайдено у 1976 році у відділі наукового розвитку поліції у Великобританії. Прототипи систем працювали до 1979 року, і було укладено контракти на виробництво промислових систем, спочатку в EMI Electronics, а потім у Computer Recognition Systems.

У своїй роботі, системи ідентифікації номерів транспортних засобів використовують низку методів обробки зображень, щоб виявити, нормалізувати та покращити зображення номерного знаку, а потім оптичне розпізнавання символів, щоб витягти текстову інформацію із номерного знаку [6]. Розвиток систем розпізнавання номерних знаків приводив до змін дорожнього регулювання у світі. Прикладом цього може слугувати змінна стилі голландських номерних знаків у 2002 році, одна зі змін стосувалася шрифту, а саме було додано невеликі пропуски в деякі літери (наприклад, P і R), щоб зробити їх більш чіткими та читабельними для систем ідентифікації зображень номерів транспортних засобів. Різні номерні знаки використовують різні розміри шрифтів і розташування тексту, системи автоматичного розпізнання повинні впоратися із такими відмінностями, щоб бути справді ефективними. Більш складні системи можуть працювати із міжнародними варіантами номерних знаків, хоча багато програм індивідуально адаптовані для кожної країни.

Для систем автоматичного розпізнавання та ідентифікації зображень номерів транспортних засобів можуть використовуватися існуючі камери контролю за дотриманням правил дорожнього руху або камери замкнутого телебачення, а також мобільні пристрої, що зазвичай прикріплюються до транспортних засобів. Деякі системи використовують інфрачервоні камери, для того щоб отримати чіткіше зображення номерного знаку.

Встановлення систем ідентифікації номерних знаків вимагає ретельного розташування камер. Використовувати потрібну кількість камер і точно розташувати їх для отримання оптимальних результатів може виявитися складно, враховуючи різноманітні перешкоди та унікальності середовища. Для регулювання дорожнього руху необхідні камери, які охоплюють кілька смуг і здатні зчитувати номерні знаки на високих швидкостях. Для зйомки номерних знаків на припаркованих автомобілях потрібні камери з меншим радіусом дії та меншою фокусною відстанню. Парковки з перпендикулярними місцями для автомобілів часто потребують спеціальної камери з дуже короткою фокусною відстанню. У державах із задніми номерними знаками виникає додаткова проблема, оскільки передня камера неефективна при зустрічному русі. У цьому випадку одна камера може бути повернута назад.

1.2 Труднощі при ідентифікації номерних знаків автомобілів

Власники транспортних засобів використовували різні методи, щоб уникнути системи автоматичного розпізнавання та ідентифікації зображень номерів транспортних засобів та камер контролю за дотриманням правил дорожнього руху в цілому. Один із методів базується на покращенні відбивання світла номерного знаку, тим самим підвищуючи ймовірність того, що система не зможе визначити місцезнаходження номерного знаку або створити достатньо високий рівень контрасту, щоб його можна було

прочитати [8]. Зазвичай це робиться за допомогою прозорої кришки для номеру автотранспорту або специфічного спрею, хоча твердження щодо ефективності останнього заперечуються. Проте у більшості юрисдикцій дані кришки є незаконними та підпадають під чинні закони, тоді як у більшості країн немає закону, який би забороняв використання спреїв.

Загалом існує низка можливих труднощів, з якими програмне забезпечення системи розпізнання номерів транспортних засобів повинно впоратися, таких як:

- погана роздільна здатність файлу, як правило, тому, що номерний знак розташований надто далеко, але іноді через використання камери низької якості;

- розмиті зображення, особливо під час руху;

- погане освітлення та низький контраст через відбиття світла або тіні;

- об'єкти, що закривають номерний знак;

- номерні знаки, що відрізняються спереду та ззаду через використання причепів;

- зміна смуги руху під кутом огляду камери під час зчитування номерного знаку;

- відсутність координації між країнами чи державами. Два автомобіля з різних країн або штатів можуть мати однаковий номер, але різний дизайн таблички.

Незважаючи на те, що деякі з цих проблем можна виправити за допомогою програмного забезпечення, розробка рішень для цих труднощів залишається передусім на апаратній стороні системи. Збільшення висоти камери може уникнути проблеми із перешкодами або зайвими об'єктами (наприклад, іншими транспортними засобами), які закривають номерний знак, але створює та посилює інші проблеми, такі як коригування збільшеного ракурсу номерного знаку.

Якщо система автоматичного розпізнавання та ідентифікації зображень номерів транспортних засобів не здатна прочитати та ідентифікувати номерний знак, вона може позначити зображення для перегляду людиною оператором.

1.3 Використання систем розпізнавання номерних знаків

Автоматичне розпізнавання зображень номерів транспортних засобів може використовуватися для безлічі завдань: від попередження заторів до керування відправленнями, щоб вони прибували вчасно. Це важливий інструмент для поліції, систем паркування, забезпечення безпеки, а віднедавна і для логістики.

Транспортні засоби можна відстежувати в межах певної області на основі їх номерних знаків. Затримки або затори можуть бути випереджені, а плата за проїзд може стягуватися автоматично. Інформаційна система автоматичного розпізнавання та ідентифікації зображень номерів транспортних засобів також дає змогу місцевим органам влади та державним органам, позначати небезпечне водіння, та застарілі водійські права. Аналогічним чином системи ідентифікації номерних знаків можуть використовуватися для відстеження сторонніх відвідувачів, які не повинні перебувати в певному районі.

Для місць із системами платного входу та паркування засоби розпізнавання номерних знаків можна використовувати для відстеження відвідувачів і автоматичного стягнення з них плати. Завдяки цьому людям не доводиться стояти в черзі за квитком, що значно покращує дотримання паркувальних правил і пропущених платежів.

Розширена аналітика відеоданих (на базі машинного навчання) приносить більше застосувань для автоматичної ідентифікації номерів автотранспорту. За допомогою розпізнавання марки та моделі транспортного

засобу можна проаналізувати та порівняти із зареєстрованою маркою та моделлю для певного номерного знаку, щоб швидко ідентифікувати викрадений автомобіль або шахрайство. Транспортний засіб, залишений не в тому місці, може бути автоматично позначено для дослідження. Та про порушення правил проїзду на червоне світло, небезпечне водіння (наприклад, рух у неправильному напрямку) і транспортні засоби, які рухаються неправильною смугою (наприклад, смуга для автобусів), можна сповістити транспортну поліцію для вжиття заходів.

Останні інновації сприяли застосуванню даних систем для програм безпеки периметра та контролю доступу в державних установах. У США у рамках національної безпеки щодо захисту від передбачуваних актів тероризму запровадили застосування систем автоматичного розпізнання зображень номерів транспортних засобів для чутливих об'єктів, таких як посольства, школи, аеропорти, морські порти, військові та федеральні будівлі, правоохоронні та державні установи, а також транспортні центри [9].

Окрім державних установ, багато галузей приватного сектора, які мають проблеми з безпекою об'єктів, починають впроваджувати рішення систем автоматичної ідентифікації транспортних засобів. Приклади включають казино, лікарні, музеї, автостоянки та курорти. Приватні заклади, як правило, не мають доступу до державних чи поліцейських списків спостереження, але можуть розробляти та порівнювати зі своїми власними базами даних для клієнтів, VIP-персон, критично важливого персоналу чи списків заборонених осіб. На додаток до забезпечення безпеки периметра, приватна система розпізнання автотранспорту має сервісні додатки для керування паркуванням та логістики (відстеження транспортних засобів постачальників і служб підтримки).

У багатьох містах і районах розроблені системи контролю дорожнього руху, які допомагають контролювати потік транспортних засобів навколо

дорожньої мережі. Зазвичай це включало перегляд історичних даних, спостережень і статистичних даних, таких як:

- користування автостоянкою;
- використання пішохідного переходу;
- кількість транспортних засобів на дорозі;
- ділянки низької та високої завантаженості;
- періодичність, місце та причина дорожніх робіт.

Камери відеоспостереження можуть використовуватися, щоб допомогти центрам керування дорожнім рухом, надаючи їм дані в реальному часі, дозволяючи приймати рішення щодо управління дорожнім рухом у режимі реального часу. Використовуючи системи автоматичного розпізнання зображень номерів автотранспорту на цьому відеоматеріалі, можна стежити за рухом окремих транспортних засобів, автоматично надаючи інформацію про швидкість і прохідність на різних маршрутах. Ці деталі можуть висвітлити проблемні області, коли вони виникають, і допомогти центру приймати обґрунтовані рішення щодо управління інцидентами.

Деякі округи Великобританії працювали з Siemens Traffic над розробкою систем моніторингу дорожнього руху для власних центрів управління. Такі проекти, як ROMANSE, надають інтерактивний веб-сайт у режимі реального часу, який показує подробиці про дорожній рух у місті. Веб-сайт показує інформацію про автостоянки, поточні дорожні роботи, спеціальні події та записи, зняті з камер відеоспостереження. Системи ідентифікації номерних знаків автомобілів можна використовувати для визначення середнього часу в дорозі від точки до точки за певними маршрутами, що може відобразитися на інформаційній панелі, тим самим даючи водіям можливість планувати свій маршрут. ROMANSE також дозволяє мандрівникам бачити поточну ситуацію за допомогою мобільного пристрою з підключенням до інтернету, дозволяючи їм переглядати зображення камер відеоспостереження.

Британська компанія Trafficmaster використовує системи автоматичного розпізнання зображень номерів транспортних засобів із 1998 року для оцінки середньої швидкості руху по авто-магістральних дорогах без спотворення результатів місцевими коливаннями, спричиненими світлофорами.

Системи ідентифікації автотранспорту також можуть використовуватися для:

- перетину кордону;
- вилучення автомобілів;
- вимірювання середньої швидкості транспортного засобу на великих відстанях;
- забезпечення виконання законів про переїзд транспортних засобів екстреної допомоги.

1.4 Висновок до першого розділу

В першому розділі кваліфікаційної роботи наведено короткий розвиток інформаційних систем розпізнавання та ідентифікації зображень номерів транспортних засобів.

Розглянуто принципи роботи систем ідентифікації автотранспорту як і поширені труднощі при виконанні ідентифікації номерних знаків. Та проведено огляд існуючих систем розпізнавання номерних знаків автотранспорту.

Відповідно було наведено доречність використання інформаційної системи автоматичного розпізнавання та ідентифікації зображень номерів транспортних засобів у сферах регулювання правил дорожнього руху, нагляду за чутливими об'єктами, та діяльності приватних підприємств.

2 АНАЛІЗ МЕТОДІВ ТА ПРОЄКТУВАННЯ ДЛЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ НОМЕРІВ ТРАНСПОРТНИХ ЗАСОБІВ

2.1 Методи розпізнавання номерних знаків автотранспорту

Розпізнавання номерних знаків – це передова технологія машинного бачення, яка використовується для ідентифікації транспортних засобів за їх номерними знаками без прямого втручання людини. Цей розвиток інтелектуальної транспортної системи надає дані номери транспортних засобів, які можуть бути використані для подальшого аналізу та моніторингу.

В останні роки були розроблені різні алгоритми виявлення номерних знаків. Процес розпізнавання та ідентифікації зображень номерів транспортних засобів складається із трьох частин:

- попередня обробка вхідного зображення;
- виявлення номерного знаку;
- вилучення номерного знаку та перевірка номера.

Загалом можна виділити шість основних алгоритмів що необхідні для ідентифікації номерного знаку автотранспорту:

- локалізація номерного знаку, відповідає за пошук і виділення знаку на зображенні транспортного засобу;
- визначення орієнтації та розміру номерного знаку, компенсує перекид знаку та регулює розміри до необхідного формату;
- нормалізація, регулює яскравість і контрастність зображення;
- сегментація символів, знаходить окремі символи номерного знаку;
- оптичне розпізнавання символів;
- синтаксичний аналіз, перевірка символів та позицій відповідно до правил країни.

Ефективність кожного з цих підрозділів системи ідентифікації номерних знаків визначає точність системи. Також для покращення точності

роботи інформаційної системи, під час нормалізації можна використовувати методи визначення країв номерного знаку, та зменшення візуального шуму на зображенні.

У дослідженні [14] запропоновано метод розпізнавання номерних знаків автотранспорту на основі виявлення країв. Сам алгоритмом складається із двох частин для виявлення номеру транспортного засобу. У ньому використовується групування ліній зображень і використання щільності країв.

Перша частина виділяє відрізки ліній і групує їх на основі набору геометричних умов. Це точно визначає прямокутник на межі номерного знаку.



Рисунок 2.1 – Розпізнавання номерного знаку на основі виявлення країв

При умові коли прямокутник не можливо було утворити, зображення позначається як відхилене і вступає у дію друга частина алгоритму. Друга частина знаходить області номерного знаку у яких вертикальні краї

виглядають найщільнішими. Процес перевірки виконується за допомогою модуля сегментації символів. Використовуючи зображення із реальної бази даних зібрані камерою контролю швидкості, даний метод показав високий успіх в розпізнаванні номерних знаків транспортних засобів.

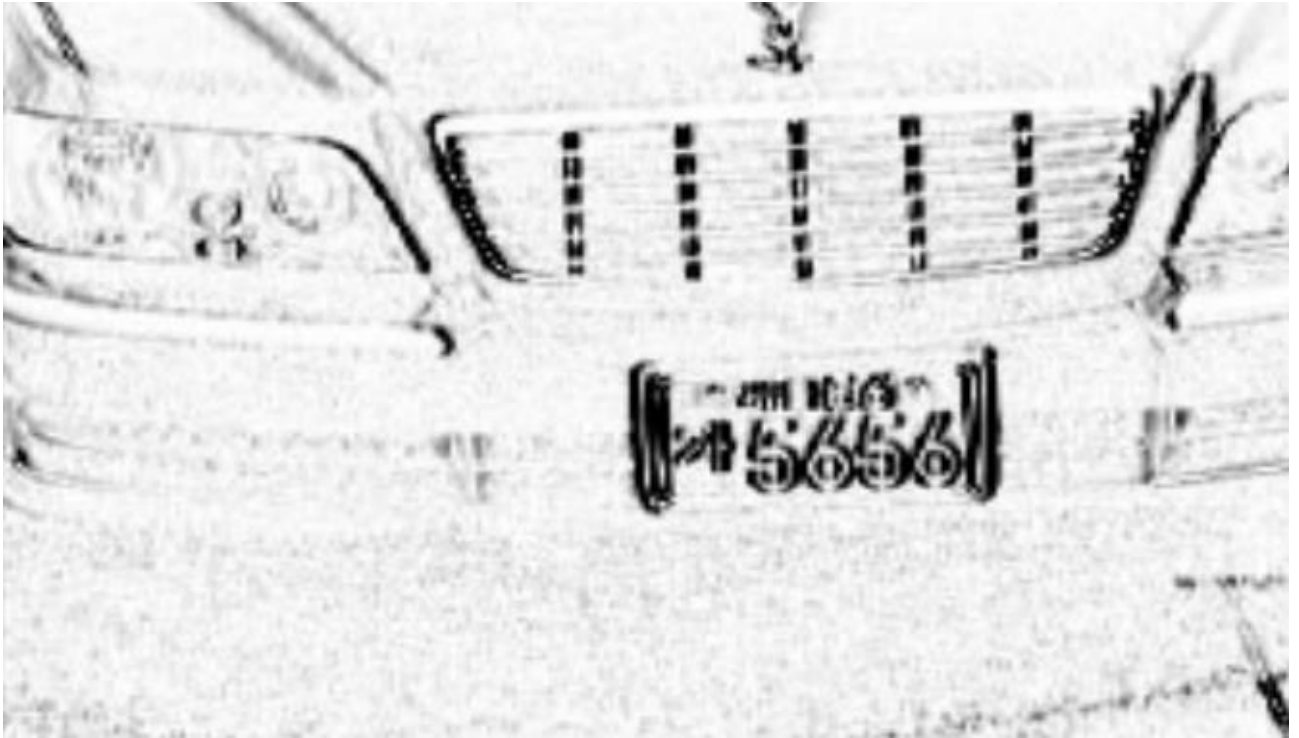


Рисунок 2.2 – Перевірка щільності вертикальних країв

Що стосується виділення області номерного знаку, методи засновані на поєднанні статистики країв і математичної морфології загалом показують дуже хороші результати. У цих методах обчислюється величина градієнта та локальна дисперсія вхідного зображення. Вони засновані на тій властивості, що зміна яскравості в області номерного знаку є більш помітною та частішою, ніж у інших частинах зображення. Потім області з високою дисперсією країв ідентифікуються як можливі області номерного знаку. Оскільки цей метод не так сильно залежить від контурів номерного знаку, його можна застосувати до зображення з нечіткими межами. Недоліком є те, що лише методи на основі країв навряд чи можна застосувати до складних зображень, оскільки вони надто чутливі до небажаних країв, які також

можуть показувати високу величину дисперсії (наприклад, область радіатора на виді спереду автомобіля). Незважаючи на це, у поєднанні з морфологічними етапами, які усувають небажані краї в оброблених зображеннях, швидкість вилучення номерних знаків є відносно високою та швидкою порівняно з іншими методами.

У літературі також для визначення номерних знаків запропоновано методи обробки на основі кольору або шкали сірого. Вирішальним для успіху методу на основі кольору (або рівня сірого) є етап сегментації кольору. З іншого боку, доступні наразі рішення не забезпечують високого ступеня точності для природних середовищ, оскільки колір є нестабільним при зміні умов освітлення. Враховуючи що ці методи, як правило, засновані на кольорах, вони не можуть виявляти номерні знаки різних кольорів. Хоча обробка кольорів показує кращу продуктивність, вона все ще має труднощі з розпізнаванням зображення автомобіля, якщо зображення містить багато подібних частин із кольорами області номера.

Розширений метод на основі кольорової текстури для виявлення номерного знаку транспортних засобів на зображеннях був представлений в роботі [15]. Система аналізує колірні та текстурні властивості номерних знаків на зображеннях за допомогою опорних векторів і визначає їх обмежувальні рамки, застосовуючи алгоритм постійного адаптивного середнього зсуву. Поєднання даного алгоритму і опорних векторів забезпечило ефективне виявлення номерних знаків автотранспорту, оскільки трудомісткий аналіз кольорової текстури для менш релевантних пікселів був обмежений, залишаючи для аналізу лише невелику частину вхідного зображення. Проте запропонований метод все ще стикався з проблемами, коли зображення було надзвичайно розмитим або досить складним за кольором.

У методі з використанням перетворення Nough (НТ) першими визначаються краї номерного знаку у вхідному зображенні. У роботі [17]

автори визнають, що виконання такого перетворення потребує занадто багато обчислень, коли він застосовується до бінарного зображення з великою кількістю пікселів. У результаті алгоритм, який вони використовували, являв собою комбінацію перетворення НТ та контурного алгоритму, що забезпечило вищу точність і швидкість, щоб його можна було застосувати до систем реального часу. Однак, оскільки НТ дуже чутливий до деформації меж, цей підхід викликає труднощі з виділенням області номерного знаку, коли його межі нечіткі або спотворені, чи зображення містять багато вертикальних і горизонтальних країв навколо решіток радіатора. Однак цей метод досяг дуже хороших результатів при застосуванні для близьких знімків автомобіля.

Симетрія також використовується як метод для ідентифікації автомобільних номерних знаків. Узагальнене перетворення симетрії (GST) створює безперервні характеристики симетрії між двома точками шляхом поєднання обмеження локальності та показником симетрії. Цей процес зазвичай займає багато часу, оскільки кількість можливих симетричних пікселів у зображенні величезна. Крім того, перевернуте або спотворене зображення номерного знаку автомобіля виявити практично неможливо. У роботі [18] автори пропонують метод декомпозиції сканування лінії розрахунку GST з метою досягнення значного зменшення обчислювального навантаження. Результат справді обнадійливий, що стосується часу обчислення. Але, оскільки GST на основі лінії сканування оцінює симетрію між парою крайових пікселів уздовж ліній сканування, час виконання збільшується лінійно щодо радіуса області пошуку. Таким чином, алгоритм встановлює обмеження на ефективну відстань, оскільки ближчий огляд потенційних номерних знаків призводить до збільшення часу обробки. Крім того, такого підходу недостатньо, коли з'являються повернуті або деформовані номерні знаки транспортних засобів.

2.2 Методи розробки систем нейронних мереж

Нейронні мережі, також відомі як штучні нейронні мережі, є основою машинного навчання та основою алгоритмів для глибокого навчання інформаційної системи. Їх назва та структура, імітують спосіб, яким біологічні нейрони передають сигнали один одному.

У більш практичному сенсі нейронні мережі – це інструменти нелінійного моделювання статистичних даних або прийняття рішень. Їх можна використовувати для моделювання складних зв'язків між входами та виходами або для пошуку шаблонів у даних.

Штучні нейронні мережі складаються з вузлових шарів, що містять вхідний рівень, один або більше прихованих шарів і вихідний рівень. Кожен вузол, або штучний нейрон, з'єднується з іншим і має відповідну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі.

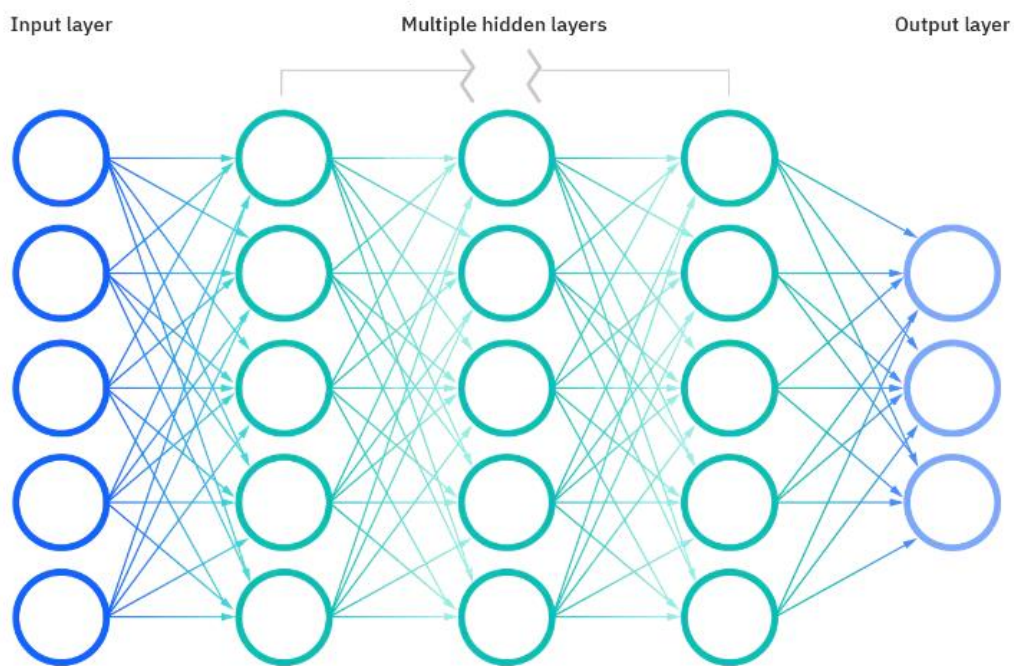


Рисунок 2.3 – Структура нейронної мережі

Нейронні мережі покладаються на навчальні дані, щоб навчатися та підвищувати свою точність з часом. Однак, коли ці алгоритми навчання налаштовані на точність, вони стають потужними інструментами в інформатиці та штучному інтелекті, що дозволяє класифікувати та кластеризувати дані з високою швидкістю. Завдання з розпізнавання мовлення або розпізнавання зображень можуть тривати хвилини чи години порівняно з ручною ідентифікацією експертів.

Можна представити кожен окремий вузол як власну модель лінійної регресії, що складається з вхідних даних, вагових коефіцієнтів, зміщення (або порогу) і вихідних даних. Після визначення вхідного рівня призначаються ваги. Ці вагові коефіцієнти допомагають визначити важливість будь-якої заданої змінної, причому більші з них мають більший внесок у результат порівняно з іншими вхідними параметрами. Потім усі вхідні дані множаться на відповідні ваги, та підсумовуються. Після цього вихідні дані проходять через функцію активації. Якщо цей вихід перевищує заданий поріг, він активує вузол, передаючи дані на наступний рівень у мережі. Це призводить до того, що вихід одного вузла стає входом наступного вузла. Цей процес передачі даних з одного рівня на наступний визначає цю нейронну мережу як мережу прямого зв'язку.

Маючи справу із більш практичними випадками використання нейронних мереж, такими як розпізнавання або класифікація зображень, використовується контрольоване навчання або мічені набори даних для навчання. Під час навчання моделі також оцінюється її точність за допомогою функції визначення помилок (або втрат). Зазвичай використовується функція середньоквадратичної помилки.

Зрештою, мета проведення навчання нейронної мережі полягає в тому, щоб мінімізувати значення функції витрат, для того щоб забезпечити правильність відповідності для будь-якого даного спостереження. Коли модель коригує свої ваги, вона використовує функцію вартості, щоб досягти

точки збіжності або локального мінімуму. Процес, у якому алгоритм коригує свої вагові коефіцієнти, здійснюється за допомогою градієнтного спуску, що дозволяє моделі визначити напрямок для зменшення помилок (або мінімізації функції витрат). З кожним навчальним прикладом параметри моделі налаштовуються для поступового зближення до мінімуму.

Більшість глибоких нейронних мереж є прямими, тобто вони рухаються лише в одному напрямку, від входу до виходу. Однак також можливо навчити модель за допомогою зворотного поширення, тобто рухатися в протилежному напрямку від виходу до входу. Зворотне розповсюдження дозволяє обчислювати та визначати помилку, відносно кожного нейрону, дозволяючи належним чином налаштувати та підігнати параметри моделі нейронної мережі. Нейронні мережі можна класифікувати на різні типи, які використовуються для різних цілей.

Персептрон – найстаріша нейронна мережа, створена у 1958 році. Вона складається з одного нейрона і є найпростішою формою нейронної мережі.

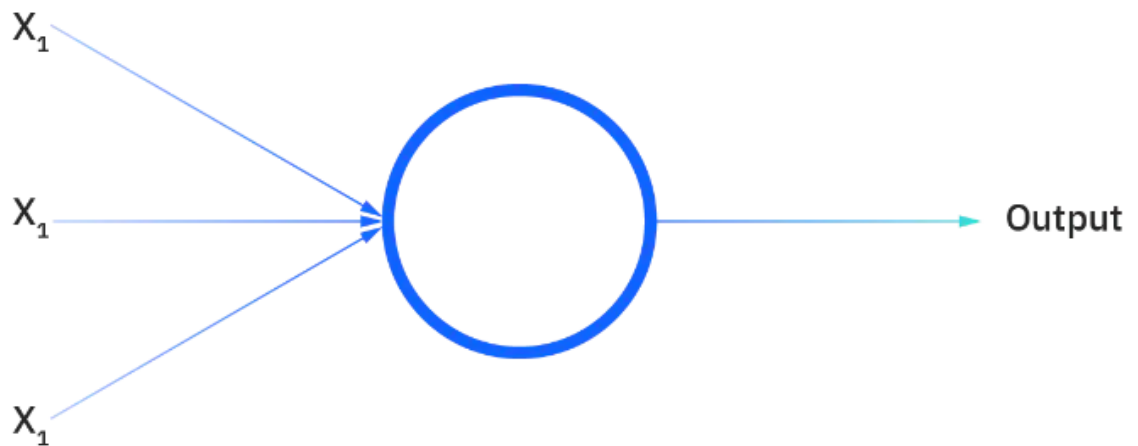


Рисунок 2.4 – Структура найпростішої нейронної мережі

Нейронні мережі прямого зв'язку, або багатошарові персептрони – це найпоширеніший вид моделі нейронної мережі. Вони складаються з вхідного шару, прихованого шару або шарів і вихідного шару. Важливо зазначити, що

насправді вони складаються з сигмовидних нейронів, а не з персептронів, оскільки більшість проблем реального світу є нелінійними. Дані зазвичай подаються в ці моделі для їх навчання, і вони є основою для комп'ютерного бачення, та інших нейронних мереж.

Згорткові нейронні мережі схожі на мережі прямого зв'язку, але вони зазвичай використовуються для розпізнавання зображень, об'єктів та образів. Ці мережі використовують принципи лінійної алгебри, зокрема множення матриць, щоб ідентифікувати шаблони в зображенні.

Повторювані нейронні мережі ідентифікуються за їх петлями зворотного зв'язку. Ці алгоритми навчання в основному використовуються під час використання даних часових рядів для прогнозування майбутніх результатів, наприклад прогнозування фондового ринку чи прогнозування продажів.

Корисність моделей штучних нейронних мереж полягає в тому, що їх можна використовувати для виведення різних функції обробки даних, а також для її використання. Неконтрольовані нейронні мережі можна використовувати для вивчення вхідних даних, та фіксування основних характеристик їх розподілу. Навчання в нейронних мережах особливо корисно в програмах, де складність даних або завдання робить розробку таких функцій вручну непрактичним.

Нейронні мережі можуть навчатися за допомогою контрольованого, неконтрольованого та навчання з підкріпленням. Для контрольованого навчання мережі надаються позначені вхідні дані та цільові приклади, тоді як у неконтрольованому та навчанні з підкріпленням, мережі будуть надані вхідні дані, і вона повинна буде знайти правильний результат.

Одним із найбільш типових алгоритмів контрольованого навчання є зворотне поширення. Зворотне поширення має на меті змінити параметри нейронних мереж відповідно до певної цілі або критеріїв. Зворотне

розповсюдження базується на локальній оптимізації градієнтного спуску, яка робить похідну функції помилки рівною нулю.

У неконтрольованому навчанні немає зворотного зв'язку з мережею та вхідними даними. У цьому виді навчання нейронна мережа повинна буде згрупувати або класифікувати вхідні дані на основі їхніх статистичних характеристик, таких як середнє значення дисперсії та стандартне відхилення. Найпоширенішим типом неконтрольованого навчання є кластерний аналіз. Самоорганізуючі карти є одним із типів неконтрольованого навчання, яке базується на конкурентному навчанні. У конкурентному навчанні вихідні нейрони конкурують між собою. Щоб нейрон став переможцем, його вектор ваги має бути найбільш схожим на вхідний шаблон. Зазвичай подібність вимірюється на основі відстані між вагою нейрона та вхідним шаблоном.

Навчання з підкріпленням є різновидом неконтрольованого навчання, яке спирається на навчання від взаємодії з навколишнім середовищем. Ідея полягає в тому, що нейронна мережа буде виконувати певні дії, та після цього вона спостерігатиме за ефектом дії, а потім на основі результату дії мережа буде або нагороджена, або покарана. Винагороду або штраф можна використовувати для налаштування параметрів нейронної мережі.

Вага – це параметр нейронної мережі, який впливає на перетворення вхідних даних в прихованих шарах мережі.

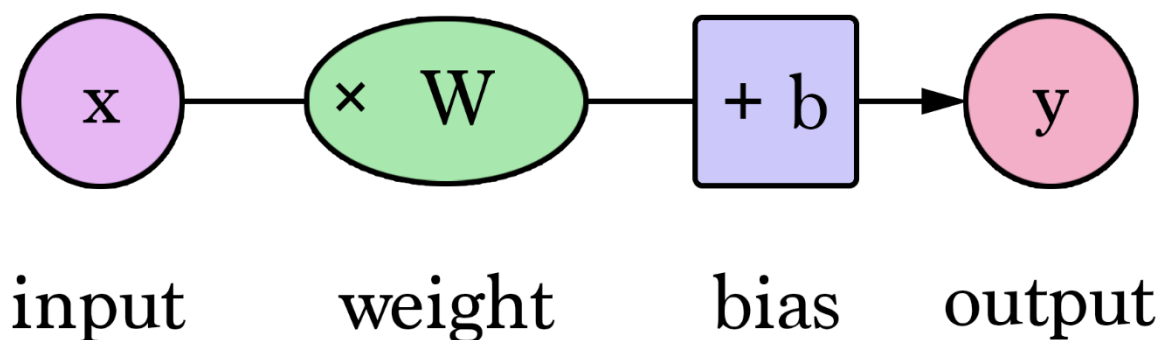


Рисунок 2.5 – Принцип роботи нейронної мережі

Нейронна мережа містить ряд прихованих шарів, які застосовують перетворення до вхідних даних. Саме в вузлах прихованих шарів застосовуються ваги. Наприклад, один вузол може взяти вхідні дані та помножити їх на значення ваги, а потім додати зміщення перед передачею даних на наступний рівень. Останній рівень нейронної мережі також відомий як вихідний рівень. Вихідний рівень часто налаштовує вхідні дані з прихованих шарів, щоб виробляти потрібні числа в заданому діапазоні.

Вага та зміщення є параметрами, які використовуються під час процесу навчання мережі. Нейронна мережа, рандомізує значення ваги та зміщення перед початком навчання. Ці два параметри відрізняються ступенем їхнього впливу на вхідні дані. Простіше кажучи, вага показує наскільки далекі прогнози від їх очікуваного значення а зміщення становлять різницю між результатом функції та її наміченим результатом. Низьке зміщення означає, що мережа робить більше припущень щодо форми виходу, тоді як високе значення зміщення робить менше припущень. З іншого боку, вагу можна розглядати як силу зв'язку. Вага впливає на ступінь впливу зміни вхідних даних на вихід. Низьке значення ваги не матиме змін на вхідні дані, як альтернатива, більше значення ваги більш суттєво змінить вихідні значення.

2.3 Методи виконання оптичного розпізнавання символів

Оптичне розпізнавання символів – це електронне перетворення, рукописних або друкованих текстових зображень у текстовий формат. Для проведення операцій автоматичного розпізнавання символів можуть використовуватися ряд різних методів:

- зіставлення матриць;
- нечітка логіка;
- витяг характеристик;
- структурний аналіз;

– використання нейронних мереж.

Зіставлення матриць перетворює кожен символ на шаблон у матриці, а потім порівнює даний шаблон із індексами відомих символів. Такий метод найкраще працює для однотипних зображень, та друкавоних текстів.



Рисунок 2.6 – Матричне зіставлення символів

Метод нечіткої логіки зазвичай являє собою комп'ютерні обчислення що дозволяють визначати проміжні значення між звичайними оцінками, таких як так чи ні, чорний або білий [21]. Робиться спроба приписати більш схожий на людину спосіб логічного мислення в програмуванні комп'ютерів. Даний метод використовується, коли відповіді не мають чітких істинних чи хибних значень і є невизначеними.

Витяг характеристик, це метод що визначає кожен символ за наявністю або відсутністю ключових характеристик, включаючи висоту, ширину, щільність, петлі, лінії, та інші риси відомих символів. Вилучення

характеристик є ідеальним підходом для оптичного розпізнання символів журналів, лазерного друку та високоякісних зображень.



Рисунок 2.7 – Виділення характеристик символів

Метод структурного аналізу ідентифікує символи, досліджуючи форми їхніх характеристик зображення використовуючи вертикальні та горизонтальні гістограми [25]. Його здатність відновлювати символи чудово підходить для тексту низької якості.

Популярним методом є використання нейронних мереж. Ця стратегія моделює роботу нервової системи людини, яка робить вибірку пікселів у кожному зображенні та зіставляє їх із відомим індексом шаблонів символів. Здатність розпізнавати символи через абстракцію чудово підходить для розпізнання тексту номерних знаків транспортних засобів.

Ідентифікація номерного знаку та його розпізнавання повністю залежить від сегментації символів. Сегментація – це процес, який визначає складові зображення. Тобто знаходження області номерного знаку, де надруковано дані, та відрізнити їх від малюнків і графіки. При неефективності сегментації символів, система для автоматичного розпізнавання та ідентифікації зображень номерів транспортних засобів не зможе правильно працювати, оскільки неможливо буде визначити окремі символи номерного знаку.



Рисунок 2.8 – Оптичне розпізнавання символів

Для покращення ефективності оптичного розпізнавання символів використовується також попередня обробка зображень, оскільки вони можуть містити певну кількість шуму залежно від роздільної здатності камери та успішності застосованої техніки порогового значення, символи можуть бути спотвореними або заламаними. Деякі з цих дефектів, які пізніше можуть спричинити низьку швидкість розпізнавання, можна усунути за допомогою обробки зображення для згладжування символів.

2.4 Висновок до другого розділу

У другому розділі кваліфікаційної роботи наведено основні методи проектування системи розпізнавання та ідентифікації номерів транспортних засобів, так само як найпоширеніші алгоритми для виконання розпізнавання номерних знаків.

Розглянуто загальну інформацію про побудову систем нейронних мереж та методи проведення навчання штучної нейронної мережі. Також було подано загальні принципи виконання оптичного розпізнавання символів.

3 ПРОЕКТУВАННЯ СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ТА ІДЕНТИФІКАЦІЇ ЗОБРАЖЕНЬ НОМЕРІВ ТРАНСПОРТНИХ ЗАСОБІВ

3.1 Структура розроблення інформаційної системи

Розробка інформаційної системи для розпізнавання та ідентифікації зображень номерів транспортних засобів відбувалася на мові програмування Python, у середовищі Jupyter Notebook, що є клієнт-серверною програмою, для редагування та запуску коду через веб-браузер.

Розпізнавання та ідентифікація зображень номерних знаків транспортних засобів проводилося за допомогою моделі нейронної мережі. Також для роботи із машинним навчанням, та побудови нейронної мережі використовувалися бібліотеки із відкритим доступом OpenCV та TensorFlow, що містять набір оптимізованих алгоритмів для виявлення та розпізнавання об'єктів.

Для виконання ідентифікації номерів транспортних засобів у даній кваліфікаційній роботі використовується технологія PyTesseract, що є інструментом оптичного розпізнавання символів.

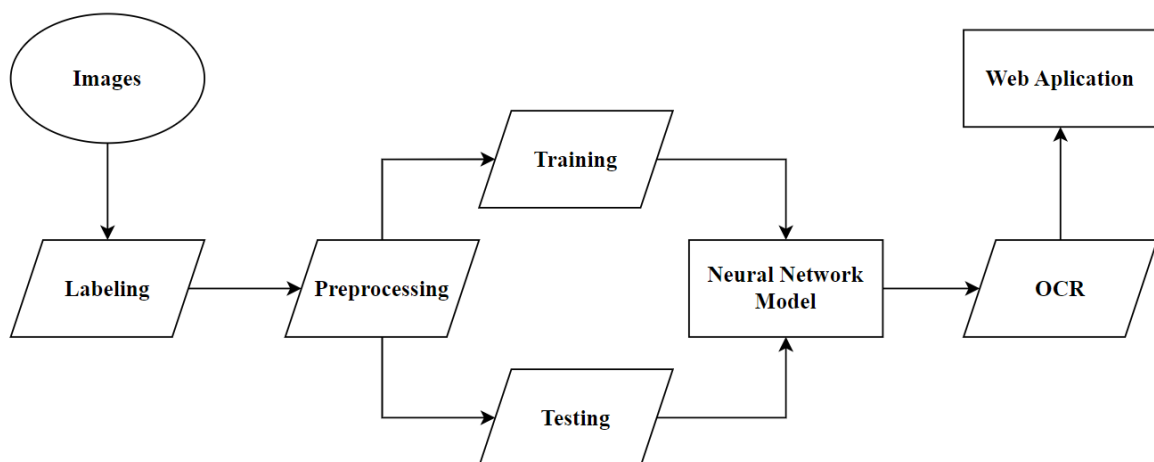


Рисунок 3.1 – Архітектура розробки системи розпізнавання номерних знаків

Першим етапом розробки моделі нейронної мережі є збір даних. Для завдання розпізнання номерних знаків транспортних засобів, вхідними даними будуть власне зображення автотранспорту. Використовуючи програмне забезпечення для анотації рисунків, на кожному зображенні заздалегідь позначається місце розташування номерного знаку, таким чином модель нейронної мережі зможе провести навчання та самостійно визначити значення помилки власних передбачень. Цей процес займає досить багато часу, оскільки це дії що потрібно самостійно виконати для кожного зображення, проте він є дуже важливим, так як весь подальший процес буде залежати від точності виконання позначень.

Наступним кроком є проведення попередньої обробки вхідних даних. Це може включати в себе трансформацію зображень та нормалізацію даних, для того щоб досягти єдиного формату даних із мінімальною варіативністю.

Використовуючи оброблені зображення, та після їх розділення на тренувальну і тестову вибірки, проводиться тренування системи нейронної мережі з потужною моделлю для розпізнавання номерних знаків. Після завершення навчання моделі нейронної мережі, та перевірки коректності її передбачень стосовно розташування номерних знаків на тестовому зображенні автомобіля, вона зберігається для подальшого використання.

Після цього підготовка системи розпізнання автомобільних номерних знаків буде завершеною. Таким чином модель нейронної мережі отримавши довільне зображення автомобіля зможе зробити передбачення місця розміщення номерного знаку, що буде так званим ROI (Reason of Interest). І наостанок використовуючи технології оптичного розпізнавання символів, розроблена система ідентифікує дані номерного знаку, та поверне номер автотранспорту у текстовому форматі.

Кінцевим етапом є створення графічного інтерфейсу для взаємодії розробленої системи розпізнання номерних знаків із користувачами. Для цього буде створений веб застосунок, на якому користувачі зможуть

завантажувати власні зображення автомобілів, та отримувати усі результати роботи створеної моделі нейронної мережі, такі як зображення із передбаченням розміщення номерного знаку, та ідентифікований номер автотранспорту у текстовому форматі.

3.2 Підготовка тестових даних для інформаційної системи

Першим кроком при розробці нейронної мережі для розпізнавання номерних знаків транспортних засобів є збір тестових даних. У даному випадку це є зображення передньої та задньої частин автотранспорту, де чітко видно номерний знак. Загалом було відібрано 225 зображень із мережі інтернет.

Проте для навчання нейронної мережі одних зображень буде недостатньо. Необхідно також вказати де саме на цих зображення розташовується номерний знак. Таким чином система зможе на прикладах розуміти закономірності розташування необхідних нам об'єктів, та зможе згодом робити передбачення. Для надання такої інформації використовується програма LabelImg, що перебуває у відкритому доступі.

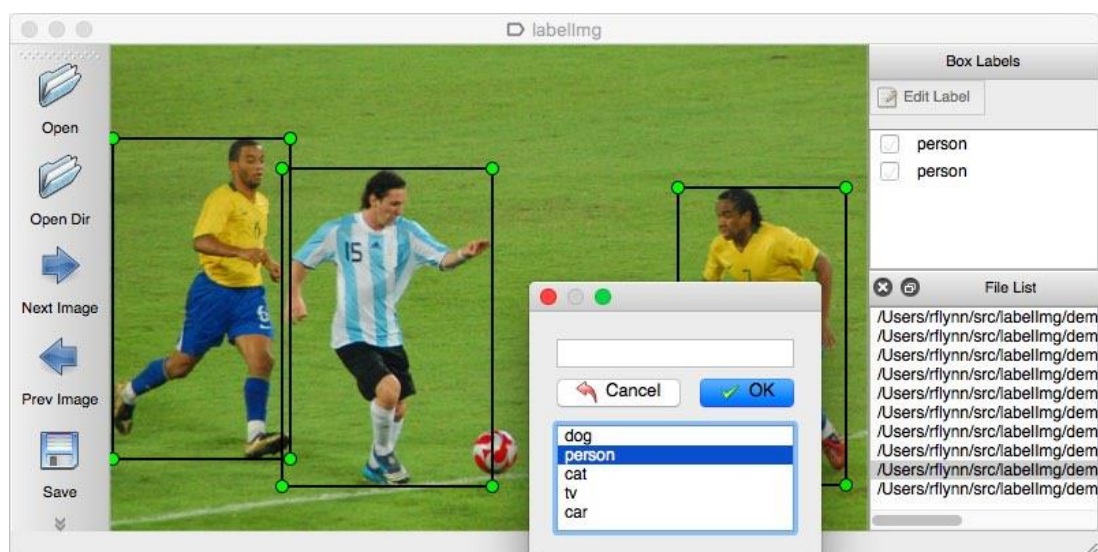


Рисунок 3.2 – Інтерфейс програмного забезпечення для позначення зображень

За допомогою LableImg можна додати анотацію до зображення з інформацією про розміщення номерного знаку, та зберегти ці дані у відповідному XML файлі.

Для цього необхідно відкрити зображення за допомогою даної програми, та використовуючи інструмент “Create RectBox” намалювати прямокутну рамку на об’єкті, який нас цікавить, в даному випадку це номерний знак автомобіля, та відповідно позначити цю рамку.

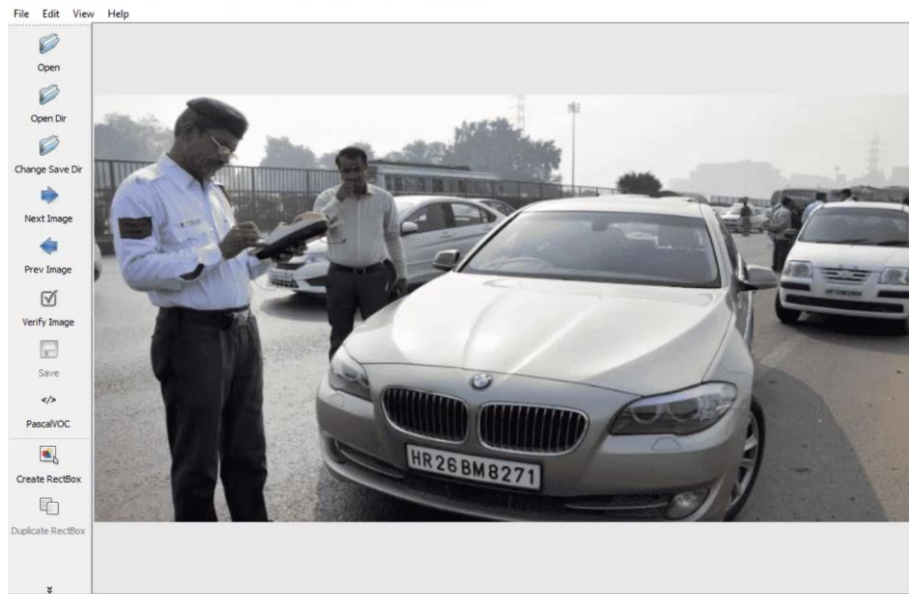


Рисунок 3.3 – Завантаження тестового зображення автомобіля



Рисунок 3.4 – Додавання анотації про положення номерного знаку

Після збереження усіх змін, анотація зберігається в окремому XML файлі. Таку процедуру необхідно повторити для кожного зображення, та в кінцевому результаті було створено 225 різних XML файлів, відповідно до кожного рисунку.

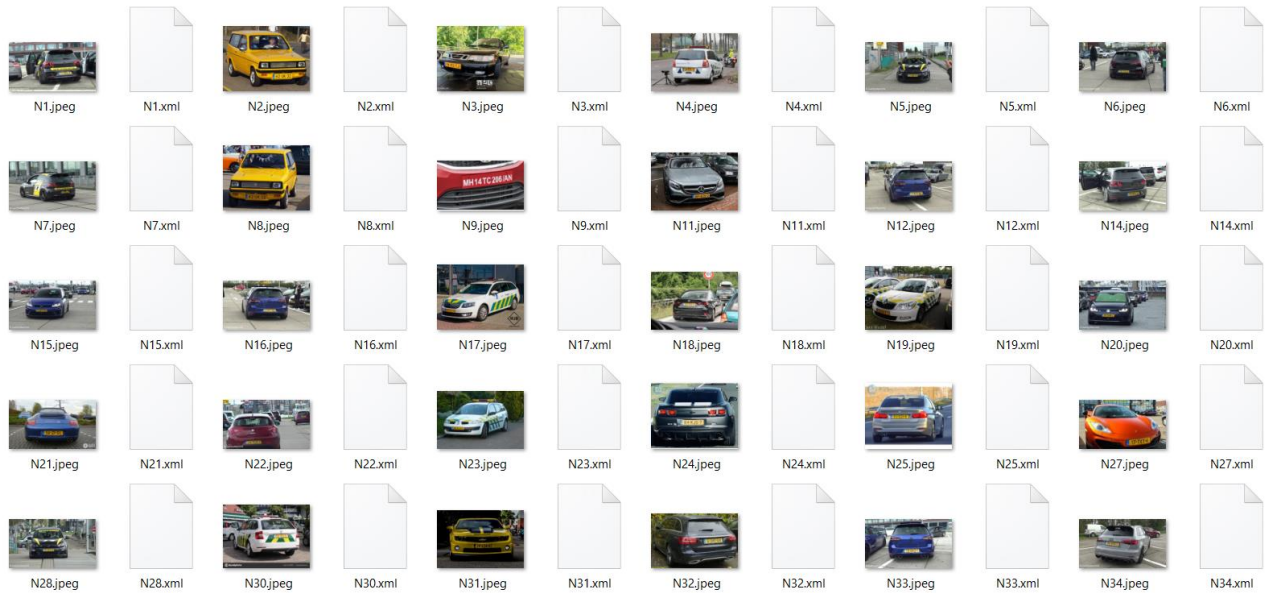


Рисунок 3.5 – Результати анотації усіх тестових зображень

```

E:\TNTU\DUPL0\APP\imag... x
<?xml version="1.0"?>
- <annotation>
  <folder>E</folder>
  <filename>N1.jpeg</filename>
  <path>E:/TNTU/DUPL0/APP/images/N1.jpeg</path>
  - <source>
    <database>Unknown</database>
  </source>
  - <size>
    <width>1920</width>
    <height>1080</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  - <object>
    <name>number_plate</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>1093</xmin>
      <ymin>645</ymin>
      <xmax>1396</xmax>
      <ymax>727</ymax>
    </bndbox>
  </object>
</annotation>
  
```

Рисунок 3.6 – Вміст XML файлу із даними про розміщення номерного знаку

Кожен XML файл зберігає інформацію про відповідне зображення, а саме його назву та розміщення на комп'ютері. Також у файлі присутні дані про позначений прямокутник, та саме ця інформація є ключовою для навчання нейронної мережі.

Лістинг 3.1 – XML дані про положення номерного знаку

```
<bndbox>
  <xmin>1093</xmin>
  <ymin>645</ymin>
  <xmax>1396</xmax>
  <ymax>727</ymax>
</bndbox>
```

З лістингу 3.1 видно що ми зберігаємо дані про дві діагональні точки прямокутника, а саме їх координати на площині зображення. Наступним кроком буде поєднання цих координат з усіх XML файлів у єдину таблицю формату CSV. Таким чином буде всього один файл із інформацією про усі номерні знаки зображень, що полегшить процес навчання нейронної мережі. Для виконання цього завдання було створено новий Python скрипт в середовищі Jupyter Notebook.

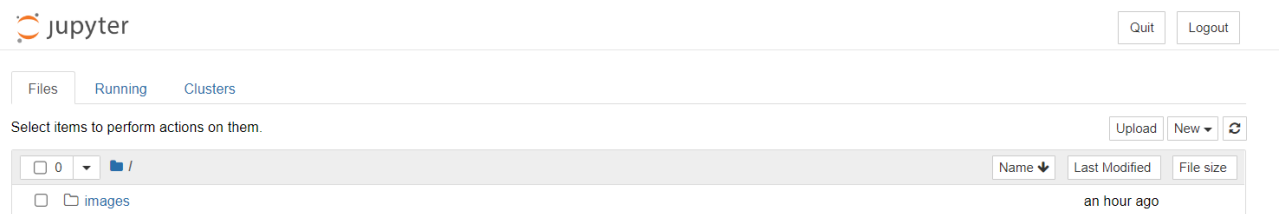


Рисунок 3.7 – Середовище розробки Jupyter Notebook

Даний скрипт складається із трьох частин, завантаження усіх XML файлів із вказаної папки, прохід циклом по кожному файлі для зчитування значень координат номерного знаку транспортного засобу, та поєднання цих значень із шляхом до XML файлу і внесення у CSV файл як окремий рядок даних.

Лістинг 3.2 – Перетворення XML даних у CSV таблицю

```

path = glob('./images/*.xml')
labels_dict = dict(filepath=[], xmin=[], xmax=[], ymin=[], ymax=[])

for filename in path:
    info = xet.parse(filename)
    root = info.getroot()
    member_object = root.find('object')
    labels_info = member_object.find('bndbox')
    xmin = int(labels_info.find('xmin').text)
    xmax = int(labels_info.find('xmax').text)
    ymin = int(labels_info.find('ymin').text)
    ymax = int(labels_info.find('ymax').text)
    labels_dict['filepath'].append(filename)
    labels_dict['xmin'].append(xmin)
    labels_dict['xmax'].append(xmax)
    labels_dict['ymin'].append(ymin)
    labels_dict['ymax'].append(ymax)
df = pd.DataFrame(labels_dict)
df.to_csv('labels.csv', index=False)

```

В результаті можна спостерігати, що утворився новий CSV файл із усією інформацією про позначення номерних знаків транспортних засобів.



Рисунок 3.8 – Створення CSV файлу

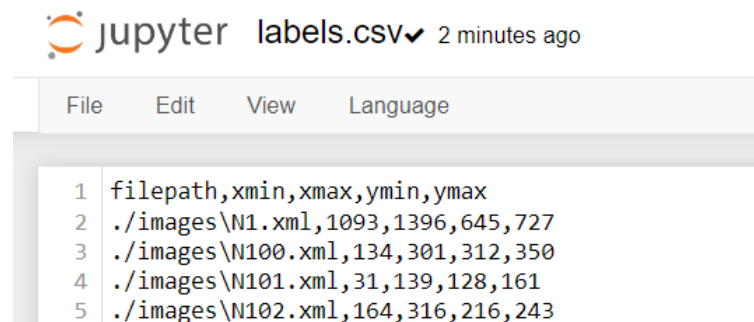


Рисунок 3.9 – Вміст CSV файлу із даними про номерні знаки усіх зображень

В подальших розділах ця інформація буде використана для проведення тренування нейронної мережі та розпізнавання номерних знаків транспортних засобів.

3.3 Попередня обробка тестових даних

Наступним етапом у процесі роботи системи автоматичного розпізнавання та ідентифікації номерів транспортних засобів є виконання попередньої обробки даних. Адже різні зображення можуть мати різний розмір, що негативно вплине на навчання нейромережі. Така попередня обробка даних виконується за допомогою нового скрипту в середовищі Jupyter Notebook, у який спочатку потрібно завантажити CSV таблицю, що містить дані про кожний XML файл.

```
In [2]: df = pd.read_csv('labels.csv')
df.head()
```

Out[2]:

	filepath	xmin	xmax	ymin	ymax
0	./images\N1.xml	1093	1396	645	727
1	./images\N100.xml	134	301	312	350
2	./images\N101.xml	31	139	128	161
3	./images\N102.xml	164	316	216	243
4	./images\N103.xml	813	1067	665	724

Рисунок 3.10 – Завантаження CSV таблиці в середовищі Jupyter Notebook

Для опрацювання даних зображення необхідно також завантажити саме зображення. Для цього потрібно отримати повне ім'я цього файлу, або відносний шлях до нього. Ім'я файлу зображення можна знайти у відповідному корені кожного XML файлу. Також будується відносний шлях

до цього файлу, додаючи назву папки в якій знаходяться усі зображення. Для виконання такої задачі доцільно написати просту функцію, та застосувати її для кожного рядка CSV таблиці.

Лістинг 3.3 – Функція для отримання розташування зображення

```
def getFilename(name):
    file_image = xet.parse(name).getroot().find('filename').text
    filepath_image = os.path.join('./images',file_image)
    return filepath_image
```

```
In [7]: image_path = list(df['filepath'].apply(getFilename))
        image_path

Out[7]: ['./images\\N1.jpeg',
          './images\\N100.jpeg',
          './images\\N101.jpeg',
          './images\\N102.jpeg',
          './images\\N103.jpeg',
          './images\\N104.jpeg',
          './images\\N105.jpeg',
          './images\\N106.jpeg',
          './images\\N107.jpeg',
          './images\\N108.jpeg',
          './images\\N109.jpeg',
          './images\\N11.jpeg',
          './images\\N110.jpeg',
          './images\\N111.jpeg',
          './images\\N112.jpeg',
          './images\\N113.jpeg',
          './images\\N114.jpeg',
          './images\\N116.jpeg',
          './images\\N117.jpeg',
          './images\\N118.jpeg',
          './images\\N119.jpeg',
          './images\\N12.jpeg',
          './images\\N120.jpeg',
          './images\\N121.jpeg',
          './images\\N122.jpeg',
          './images\\N123.jpeg',
          './images\\N124.jpeg',
          './images\\N125.jpeg',
          './images\\N126.jpeg',
          './images\\N127.jpeg',
          './images\\N128.jpeg',
          './images\\N129.jpeg',
          './images\\N13.jpeg',
          './images\\N130.jpeg',
          './images\\N131.jpeg',
          './images\\N132.jpeg',
          './images\\N133.jpeg',
          './images\\N134.jpeg',
          './images\\N135.jpeg',
          './images\\N136.jpeg',
          './images\\N137.jpeg',
          './images\\N138.jpeg',
          './images\\N139.jpeg',
          './images\\N14.jpeg',
          './images\\N140.jpeg',
          './images\\N141.jpeg',
          './images\\N142.jpeg',
          './images\\N143.jpeg',
          './images\\N144.jpeg',
          './images\\N145.jpeg',
          './images\\N146.jpeg',
          './images\\N147.jpeg',
          './images\\N148.jpeg',
          './images\\N149.jpeg',
          './images\\N15.jpeg',
          './images\\N150.jpeg',
          './images\\N151.jpeg',
          './images\\N152.jpeg',
          './images\\N153.jpeg',
          './images\\N154.jpeg',
          './images\\N155.jpeg',
          './images\\N156.jpeg',
          './images\\N157.jpeg',
          './images\\N158.jpeg',
          './images\\N159.jpeg',
          './images\\N16.jpeg',
          './images\\N160.jpeg',
          './images\\N161.jpeg',
          './images\\N162.jpeg',
          './images\\N163.jpeg',
          './images\\N164.jpeg',
          './images\\N165.jpeg',
          './images\\N166.jpeg',
          './images\\N167.jpeg',
          './images\\N168.jpeg',
          './images\\N169.jpeg',
          './images\\N17.jpeg',
          './images\\N170.jpeg',
          './images\\N171.jpeg',
          './images\\N172.jpeg',
          './images\\N173.jpeg',
          './images\\N174.jpeg',
          './images\\N175.jpeg',
          './images\\N176.jpeg',
          './images\\N177.jpeg',
          './images\\N178.jpeg',
          './images\\N179.jpeg',
          './images\\N18.jpeg',
          './images\\N180.jpeg',
          './images\\N181.jpeg',
          './images\\N182.jpeg',
          './images\\N183.jpeg',
          './images\\N184.jpeg',
          './images\\N185.jpeg',
          './images\\N186.jpeg',
          './images\\N187.jpeg',
          './images\\N188.jpeg',
          './images\\N189.jpeg',
          './images\\N19.jpeg',
          './images\\N190.jpeg',
          './images\\N191.jpeg',
          './images\\N192.jpeg',
          './images\\N193.jpeg',
          './images\\N194.jpeg',
          './images\\N195.jpeg',
          './images\\N196.jpeg',
          './images\\N197.jpeg',
          './images\\N198.jpeg',
          './images\\N199.jpeg']
```

Рисунок 3.11 – Масив з розташуванням усіх тестових зображень

Кожен елемент даного масиву використовується для завантаження зображення за допомогою модуля `load_img`. Таким чином можна завантажити зображення у форматі PIL об'єкта та задати єдиний розмір що буде використовуватися для усіх рисунків. Отже модель нейронної мережі завжди буде навчатися із зображеннями одного розміру. Проте необхідно також перетворити кожне зображення із PIL об'єкта, на масив чисел. Для цього використовується модуль `img_to_array`, що в результаті повертає тривимірний масив чисел.

Лістинг 3.4 – Виконання попередньої обробки даних

```
from tensorflow.preprocessing.image import load_img,
img_to_array
labels = df.iloc[:,1:].values
```

```

data = []
output = []
for ind in range(len(image_path)):
    image = image_path[ind]
    img_arr = cv2.imread(image)
    h,w,d = img_arr.shape

    # preppocesing
    load_image = load_img(image,target_size=(224,224))
    load_image_arr = img_to_array(load_image)

    # normalization
    norm_load_image_arr = load_image_arr/255.0
    xmin,xmax,ymin,ymax = labels[ind]
    nxmin,nxmax = xmin/w,xmax/w
    nymin,nymax = ymin/h,ymax/h
    label_norm = (nxmin,nxmax,nymin,nymax)

    data.append(norm_load_image_arr)
    output.append(label_norm)

```

Не менш важливим етапом для попередньої обробки даних є їх нормалізація. На даний момент зображення завантажуються та трансформуються у масив чисел, проте варіація цих чисел доволі велика. Щоб зменшити розподіл цих даних використовується нормалізація даних. Таким чином матриці зображень діляться на 255 для того щоб усі значення в них були у проміжку від нуля до одиниці. Аналогічна операція виконується і для координат прямокутника номерного знаку, проте вони діляться на ширину та висоту оригінального зображення. Така обробка даних відбувається у циклі для кожного рядка в CSV файлі.

3.4 Проектування моделі нейронної мережі

3.4.1 Створення та налаштування нейронної мережі

Технології нейронних мереж значно розвинулися за останні роки. Тому існує багато різних моделей для побудови таких систем. У даній роботі для створення системи нейронної мережі використовується модель

InceptionResNetV2. Для її ініціалізації вказується розмір вхідних параметрів та основні налаштування, як порядок визначення ваг, чи додаткові шари для навчання.

Лістинг 3.5 – Створення моделі нейронної мережі

```
inception_resnet =
InceptionResNetV2(weights="imagenet",include_top=False,
input_tensor=Input(shape=(224,224,3)))
inception_resnet.trainable=True

headmodel = inception_resnet.output
headmodel = Flatten()(headmodel)
headmodel = Dense(500,activation="relu")(headmodel)
headmodel = Dense(250,activation="relu")(headmodel)
headmodel = Dense(4,activation='sigmoid')(headmodel)

model = Model(inputs=inception_resnet.input,outputs=headmodel)
```

Після налаштування, модель необхідно скомпілювати, для цього вказується метод обрахунку помилки та метод оптимізації з вказаним показником швидкості навчання. Оскільки це модель регресії, то помилка буде обраховуватися методом MSE (Mean Square Error).

```
In [19]: # compile model
model.compile(loss='mse',optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4))
model.summary()

Model: "functional_1"

```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	
conv2d (Conv2D)	(None, 111, 111, 32)	864	input_1[0][0]
batch_normalization (BatchNormaliza	(None, 111, 111, 32)	96	conv2d[0][0]
activation (Activation)	(None, 111, 111, 32)	0	batch_normalization[0][0]
conv2d_1 (Conv2D)	(None, 109, 109, 32)	9216	activation[0][0]
batch_normalization_1 (BatchNor	(None, 109, 109, 32)	96	conv2d_1[0][0]
activation_1 (Activation)	(None, 109, 109, 32)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 109, 109, 64)	18432	activation_1[0][0]
batch_normalization_2 (BatchNor	(None, 109, 109, 64)	192	conv2d_2[0][0]
activation_2 (Activation)	(None, 109, 109, 64)	0	batch_normalization_2[0][0]
max_pooling2d (MaxPooling2D)	(None, 54, 54, 64)	0	activation_2[0][0]
conv2d_3 (Conv2D)	(None, 54, 54, 80)	5120	max_pooling2d[0][0]
batch_normalization_3 (BatchNor	(None, 54, 54, 80)	240	conv2d_3[0][0]

Рисунок 3.12 – Початок компіляції моделі нейронної мережі

block8_10_conv (Conv2D)	(None, 5, 5, 2080)	933920	block8_10_mixed[0][0]
block8_10 (Lambda)	(None, 5, 5, 2080)	0	block8_9_ac[0][0] block8_10_conv[0][0]
conv_7b (Conv2D)	(None, 5, 5, 1536)	3194880	block8_10[0][0]
conv_7b_bn (BatchNormalization)	(None, 5, 5, 1536)	4608	conv_7b[0][0]
conv_7b_ac (Activation)	(None, 5, 5, 1536)	0	conv_7b_bn[0][0]
flatten (Flatten)	(None, 38400)	0	conv_7b_ac[0][0]
dense (Dense)	(None, 500)	19200500	flatten[0][0]
dense_1 (Dense)	(None, 250)	125250	dense[0][0]
dense_2 (Dense)	(None, 4)	1004	dense_1[0][0]
=====			
Total params: 73,663,490			
Trainable params: 19,326,754			
Non-trainable params: 54,336,736			

Рисунок 3.13 – Завершення компіляції моделі нейронної мережі

Після завершення компіляції, модель нейронної мережі готова до проведення навчання. Також на рисунку 3.13 можна побачити що в моделі присутні понад сімдесяти мільйонів параметрів, з яких близько двадцяти мільйонів є тренувальними.

3.4.2 Тренування нейронної мережі

Для проведення процедури тренування нейронної мережі необхідно розділити усі дані на дві вибірки, для тренувань, що складатиме близько вісімдесяти відсотків усіх зображень, та для тестування коректності роботи системи.

Лістинг 3.6 – Розділення зображень на навчальну та тестову вибірки

```
X = np.array(data, dtype=np.float32)
y = np.array(output, dtype=np.float32)
x_trn, x_test, y_trn, y_test = train_test_split(X, y, train_size=0.8)
```

Після цього ми можемо виконувати тренування створеної нейронної системи. Для цього потрібно вказати вибірки тренувальних даних, кількість епох навчання та вибірки з даними для тестування.

Лістинг 3.7 – Запуск навчання моделі нейронної мережі

```

history = model.fit(
    x=x_trn,
    y=y_trn,
    batch_size=10,
    epochs=100,
    validation_data=(x_test,y_test),
    callbacks=[tfb]
)

```

```

Epoch 1/100
1/18 [>.....] - ETA: 0s - loss: 0.1285WARNING:tensorflow:From C:\Users\Srikanth\anaconda3\lib\site-pac
kages\tensorflow\python\ops\summary_ops_v2.py:1277: stop (from tensorflow.python.eager.profiler) is deprecated and will be remo
ved after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
18/18 [=====] - 106s 6s/step - loss: 0.1078 - val_loss: 0.0965
Epoch 2/100
18/18 [=====] - 88s 5s/step - loss: 0.0977 - val_loss: 0.0802
Epoch 3/100
18/18 [=====] - 56s 3s/step - loss: 0.0967 - val_loss: 0.0894
Epoch 4/100
18/18 [=====] - 55s 3s/step - loss: 0.0945 - val_loss: 0.0805
Epoch 5/100
18/18 [=====] - 63s 3s/step - loss: 0.0866 - val_loss: 0.0891
Epoch 6/100
18/18 [=====] - 61s 3s/step - loss: 0.0824 - val_loss: 0.0746
Epoch 7/100
18/18 [=====] - 59s 3s/step - loss: 0.0798 - val_loss: 0.0680
Epoch 8/100
18/18 [=====] - 59s 3s/step - loss: 0.0773 - val_loss: 0.0671
Epoch 9/100
18/18 [=====] - 67s 4s/step - loss: 0.0753 - val_loss: 0.0730
Epoch 10/100
18/18 [=====] - 62s 3s/step - loss: 0.0744 - val_loss: 0.0700

```

Рисунок 3.14 – Початок процесу навчання моделі нейронної мережі

```

Epoch 90/100
18/18 [=====] - 19s 1s/step - loss: 0.0475 - val_loss: 0.0376
Epoch 91/100
18/18 [=====] - 20s 1s/step - loss: 0.0428 - val_loss: 0.0338
Epoch 92/100
18/18 [=====] - 19s 1s/step - loss: 0.0265 - val_loss: 0.0125
Epoch 93/100
18/18 [=====] - 19s 1s/step - loss: 0.0068 - val_loss: 0.0077
Epoch 94/100
18/18 [=====] - 19s 1s/step - loss: 0.0039 - val_loss: 0.0065
Epoch 95/100
18/18 [=====] - 19s 1s/step - loss: 0.0023 - val_loss: 0.0067
Epoch 96/100
18/18 [=====] - 22s 1s/step - loss: 0.0018 - val_loss: 0.0060
Epoch 97/100
18/18 [=====] - 19s 1s/step - loss: 0.0011 - val_loss: 0.0061
Epoch 98/100
18/18 [=====] - 19s 1s/step - loss: 6.2169e-04 - val_loss: 0.0061
Epoch 99/100
18/18 [=====] - 19s 1s/step - loss: 4.7802e-04 - val_loss: 0.0060
Epoch 100/100
18/18 [=====] - 19s 1s/step - loss: 3.6833e-04 - val_loss: 0.0060

```

Рисунок 3.15 – Завершення процесу навчання моделі нейронної мережі

Дана процедура займає досить довгий час, та після її виконання можна побачити що між першою епохою навчання та останньою обчислювальне значення помилки значно зменшилося, та вкінці було рівним 0,006.

Було також проведено друге тестування цієї ж моделі нейронної мережі із вказання початкової епохи для навчання рівній 101.

Лістинг 3.8 – Другий запуск навчання моделі нейронної мережі

```
history = model.fit(
    x=x_train,
    y=y_train,
    batch_size=10,
    epochs=200, validation_data=(x_test,y_test),
    callbacks=[tfb],
    initial_epoch=101
)
```

```
Epoch 102/200
 2/18 [=>.....] - ETA: 1:03 - loss: 3.1036e-04WARNING:tensorflow:Callbacks method `on_train_batch_end` is
s slow compared to the batch time (batch time: 1.2072s vs `on_train_batch_end` time: 6.6735s). Check your callbacks.
18/18 [=====] - 34s 2s/step - loss: 2.9122e-04 - val_loss: 0.0063
Epoch 103/200
18/18 [=====] - 21s 1s/step - loss: 1.9947e-04 - val_loss: 0.0059
Epoch 104/200
18/18 [=====] - 21s 1s/step - loss: 2.0441e-04 - val_loss: 0.0061
Epoch 105/200
18/18 [=====] - 19s 1s/step - loss: 1.8242e-04 - val_loss: 0.0059
Epoch 106/200
18/18 [=====] - 19s 1s/step - loss: 1.5497e-04 - val_loss: 0.0059
Epoch 107/200
18/18 [=====] - 21s 1s/step - loss: 2.8745e-04 - val_loss: 0.0060
Epoch 108/200
18/18 [=====] - 20s 1s/step - loss: 3.3103e-04 - val_loss: 0.0063
Epoch 109/200
18/18 [=====] - 21s 1s/step - loss: 2.3359e-04 - val_loss: 0.0059
Epoch 110/200
18/18 [=====] - 22s 1s/step - loss: 1.7057e-04 - val_loss: 0.0061
```

Рисунок 3.16 – Початок процесу навчання моделі нейронної мережі

```
18/18 [=====] - 21s 1s/step - loss: 2.2735e-04 - val_loss: 0.0061
Epoch 195/200
18/18 [=====] - 20s 1s/step - loss: 1.9882e-04 - val_loss: 0.0061
Epoch 196/200
18/18 [=====] - 20s 1s/step - loss: 1.7346e-04 - val_loss: 0.0063
Epoch 197/200
18/18 [=====] - 22s 1s/step - loss: 1.5138e-04 - val_loss: 0.0060
Epoch 198/200
18/18 [=====] - 22s 1s/step - loss: 2.5786e-04 - val_loss: 0.0060
Epoch 199/200
18/18 [=====] - 20s 1s/step - loss: 2.7735e-04 - val_loss: 0.0061
Epoch 200/200
18/18 [=====] - 21s 1s/step - loss: 2.2967e-04 - val_loss: 0.0063
```

Рисунок 3.17 – Завершення процесу навчання моделі нейронної мережі

Jupyter Notebook, та випробовується на тестовому зображенні транспортного засобу.

Лістинг 3.9 – Відображення рисунку автомобіля

```
model =  
tf.keras.models.load_model('./models/object_detection.h5')  
path = './test_images/N207.jpeg'  
image = load_img(path)  
image = np.array(image, dtype=np.uint8)  
plt.figure(figsize=(10, 8))  
plt.imshow(image)  
plt.show()
```



Рисунок 3.19 – Відображення рисунку автомобіля

Проте, щоб отримати передбачення розміщення номерного знаку використовуючи дану модель нейронної мережі, вхідне зображення необхідно нормалізувати до такого формату, на якому відбувався процес

навчання системи. Тобто зображення необхідно перевести у масив чисел від нуля до одиниці.

Лістинг 3.10 – Отримання координат потенційного номерного знаку

```
image1 = load_img(path, target_size=(224, 224))
image_arr_224 = img_to_array(image1)/255.0
coords = model.predict(test_arr_224)
coords
```

```
In [9]: # make predictions
        coords = model.predict(test_arr)
        coords
```

```
Out[9]: array([[0.40387744, 0.6220704 , 0.67173195, 0.73575115]], dtype=float32)
```

Рисунок 3.20 – Отримання координат потенційного номерного знаку

Результатом є нормалізовані координати потенційного номерного знаку автотранспорту. Наступним кроком є виконання де-нормалізації цих координат для того щоб знати точні координати оригінального зображення. Для цього необхідно виконати множення нормалізованих координат на оригінальні ширину та висоту зображення. Перші два значення множаться на довжину та останні два на висоту.

```
In [10]: # denormalize the values
         denorm = np.array([w,w,h,h])
         coords = coords * denorm
         coords
```

```
Out[10]: array([[218.49769348, 336.54007131, 284.14261436, 311.22273731]])
```

```
In [11]: coords = coords.astype(np.int32)
         coords
```

```
Out[11]: array([[218, 336, 284, 311]])
```

Рисунок 3.21 – Обробка координат потенційного номерного знаку

Маючи ці координати, що є місцями розміщення двох діагональних точок прямокутника, легко можна зобразити дану фігуру на оригінальному зображенні для наглядної перевірки коректності передбачень моделі нейронної мережі.

Лістинг 3.11 – Відображення передбачення розміщення номерного знаку

```
xmin, xmax, ymin, ymax = coords[0]
pt1 = (xmin, ymin)
pt2 = (xmax, ymax)
cv2.rectangle(image, pt1, pt2, (0, 255, 0), 3)

plt.figure(figsize=(10, 8))
plt.imshow(image)
plt.show()
```



Рисунок 3.22 – Передбачення системи розпізнавання номерних знаків

Із наведеного рисунку можна зробити висновок що розроблена модель нейронної мережі успішно розпізнає номерний знак на зображенні автомобіля,

проте можуть виникати труднощі із точним визначенням країв номерного знаку. Такі неточності можна виправити додатковим тренуванням моделі використовуючи більшу вибірку зображень автотранспорту.

3.5 Виконання оптичного розпізнавання символів

Оптичне розпізнавання символів або OCR (Optical Character Recognition) – це процес виділення власне текстової інформації із зображення на якому присутній текст. Для розпізнавання тексту номерних знаків автомобіля використовується безкоштовна технологія від Google, Tesseract, а саме її Python версія – PyTesseract.

Отже для того щоб успішно ідентифікувати та зчитати текстову інформацію із номерного знаку необхідно відділити його від загального зображення.

Оскільки координати розміщення номерного знаку автомобіля вже є відомі, що можна бачити на рисунку 3.22. Легко можна обрізати оригінальне зображення по цим координатам, створивши рисунок виключно номерного знаку.

Лістинг 3.12 – Відображення рисунку номеру транспортного засобу

```
path = './test_images/N207.jpeg'
img = np.array(load_img(path))
cords = object_detection(path)
xmin, xmax, ymin, ymax = cords[0]
roi = img[ymin:ymax, xmin:xmax]
plt.imshow(roi)
plt.show()
```



Рисунок 3.23 – Вирізаний номер транспортного засобу

Після цього використовуючи PyTesseract, за допомогою вбудованого модуля `image_to_string` можна зчитати текстову інформацію із рисунка та вивести її на екран для перевірки.

```
In [23]: # extract text from image
text = pt.image_to_string(roi)
print(text)
```

MN 20 EE D943

Рисунок 3.24 – Зчитування тексту із зображення номерного знаку

У підсумку розроблена модель нейронної мережі успішно розпізнала номерний знак із зображення автомобіля, та змогла ідентифікувати текстову інформацію з нього. Оскільки Tesseract є однією із найдоступніших технологій оптичного розпізнання символів, в ній присутній ряд обмежень, таким чином при низькій якості зображень інколи може виникати неточності при зчитуванні тексту.

3.6 Веб застосунок для розпізнавання номерів транспортних засобів

З метою надання зручного графічного інтерфейсу для розробленої системи розпізнання номерних знаків транспортних засобів, було створено веб застосунок на мові програмування Python, із використанням технології Flask, у середовищі Visual Studio Code. Структура веб застосунку наведена нижче на рисунку 3.25.

У веб застосунку присутні два файли із HTML розміткою, так само як і папка із статичними даними. Тут і зберігається раніше натренована модель нейронної мережі та різні папки для зображень. Коли користувач надсилає на сервер зображення, воно зберігається в папці `upload`. Після того як нейронна

мережа зробила передбачення стосовно розташування номерного знаку, файл із додатковим позначенням зберігається у папці `predict` із оригінальним ім'ям.

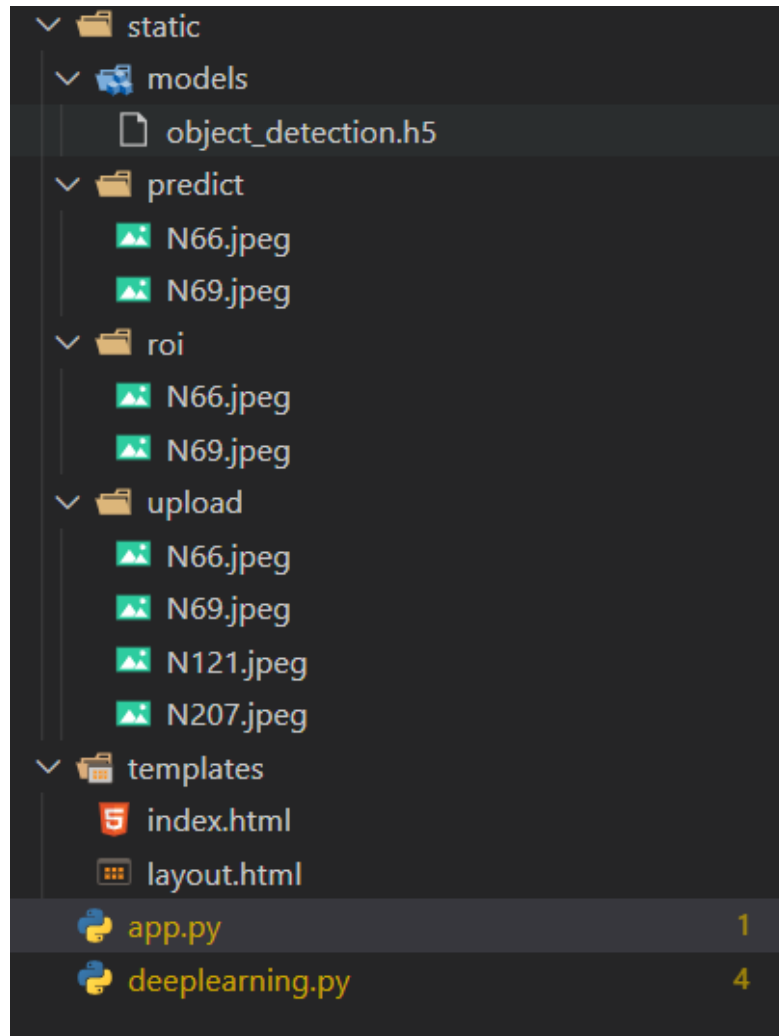


Рисунок 3.25 – Структура розробленого веб застосунку

Також фрагмент із номерним знаком окремо зберігається у папку `roi`. Таким чином достатньо знати лише назву оригінального файлу щоб завантажити усі згенеровані застосунком зображення із вже відомої структури директорій.

Головним файлом то коренем застосунку є файл `app.py` у якому запускається програма та створюється веб застосунок.

Серверна частина веб застосунку має можливість приймати GET та POST запити від клієнтів. При отриманні GET запиту, у відповідь

відправляється HTML розмітка головної сторінки без додаткової інформації про номерний знак, чи його текст. Такий сценарій відбувається коли користувач вперше запустив веб застосунок та ще не обрав зображення для ідентифікації.

Лістинг 3.13 – Обробка вхідних HTTP запитів із файлу app.py

```
app = Flask(__name__)

BASE_PATH = os.getcwd()
UPLOAD_PATH = os.path.join(BASE_PATH, 'static/upload/')

@app.route('/', methods=['POST', 'GET'])
def index():
    if request.method == 'POST':
        upload_file = request.files['image_name']
        filename = upload_file.filename
        path_save = os.path.join(UPLOAD_PATH, filename)
        upload_file.save(path_save)
        text = OCR(path_save, filename)

        return
    render_template('index.html', upload=True, upload_image=filename, text=text)

    return render_template('index.html', upload=False)

if __name__ == "__main__":
    app.run(debug=True)
```

У разі коли сервер отримує POST запит, надіслане зображення береться із тіла запиту, та використовуючи раніше створену модель нейронної мережі робиться передбачення про положення номерного знаку та ідентифікація текстової інформації, яка згодом передається до HTML розмітки головної сторінки, та надсилається як відповідь клієнту.

Уся логіка стосовно завантаження моделі нейронної мережі, виконання передбачення розміщення номерного знаку та виділення тексту відбувається у файлі *deeplearning.py* що складається із коду який використовувався у скриптах Jupyter Notebook, який згруповано у дві функції.

Лістинг 3.14 – Виконання розпізнавання та ідентифікації номерного знаку

```

model =
tf.keras.models.load_model('./static/models/object_detection.h5'
)

def object_detection(path, filename):
    image = load_img(path)
    image = np.array(image, dtype=np.uint8)
    image1 = load_img(path, target_size=(224, 224))
    image_arr_224 = img_to_array(image1)/255.0
    h, w, d = image.shape
    test_arr = image_arr_224.reshape(1, 224, 224, 3)
    coords = model.predict(test_arr)
    denorm = np.array([w, w, h, h])
    coords = coords * denorm
    coords = coords.astype(np.int32)
    xmin, xmax, ymin, ymax = coords[0]
    pt1 = (xmin, ymin)
    pt2 = (xmax, ymax)
    cv2.rectangle(image, pt1, pt2, (0, 255, 0), 3)
    image_bgr = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    cv2.imwrite('./static/predict/{}'.format(filename), image_bgr)
    return coords

def OCR(path, filename):
    img = np.array(load_img(path))
    cods = object_detection(path, filename)
    xmin, xmax, ymin, ymax = cods[0]
    roi = img[ymin:ymax, xmin:xmax]
    roi_bgr = cv2.cvtColor(roi, cv2.COLOR_RGB2BGR)
    cv2.imwrite('./static/roi/{}'.format(filename), roi_bgr)
    text = pt.image_to_string(roi)
    return text

```

Розроблений застосунок має лише одну сторінку, на якій завжди присутня форма для завантаження файлів користувачів.

Лістинг 3.15 – Форма для надсилання файлу транспортного засобу

```

<form action="#" method="POST" enctype="multipart/form-data">
  <div class="input-group">
    <input type="file" class="form-control" name="image_name"
required>
    <input type="submit" value="Upload" class="btn btn-outline-
secondary">
  </div>
</form>

```

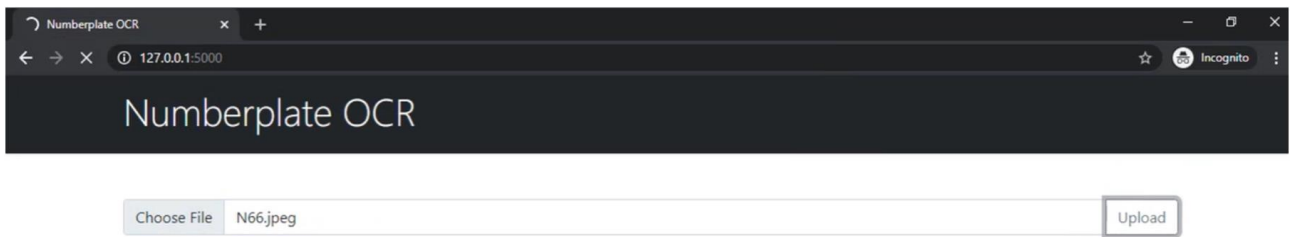


Рисунок 3.26 – Форма для надсилання файлу транспортного засобу

Та при умові що до сервера надійшов POST запит, у розмітці присутні також таблиці із оригінальним зображенням, зображенням з передбаченням місця знаходження номерного знаку, власне рисунок номерного знаку та ідентифікований текст номерного знаку транспортного засобу.

Лістинг 3.16 – Розмітка із результатами роботи інформаційної системи для розпізнавання та ідентифікації зображень номерів транспортних засобів

```
{% extends 'layout.html' %}

{% block body %}
  {% if upload %}
    <div class="container">
      <table>
        <tr>
          <td>
            
          </td>
          <td>
            
          </td>
        </tr>
      </table>
      <table style="border: solid black; width: 100%;">
        <tr style="border: solid black;">
          <th>Cropped Licence Plate Image</th>
          <th>Text</th>
        </tr>
        <tr style="border: solid black;">
          <td>
            
          </td>
          <td style="background-color: greenyellow;">
            <h1 class="display-8"> {{ text }}</h1>
          </td>
        </tr>
      </table>
    </div>
  </if>
{% endblock %}
```



```

    </tr>
  </table>
</div>
{% endif %}
{% endblock %}

```

Отже усі обрахунки та збереження файлів відбувається на сервері, який в свою чергу передає HTML розмітці дані про назву зображення яке необхідно відобразити, разом із ідентифікованим текстом номерного знаку.

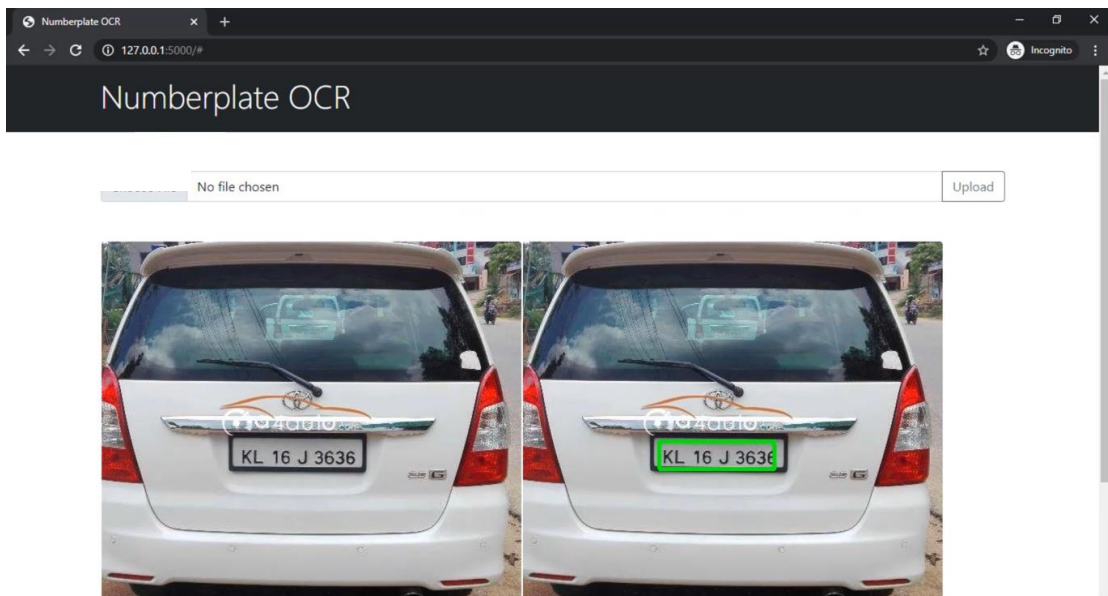


Рисунок 3.27 – Розпізнавання зображення номерного знаку



Рисунок 3.28 – Розпізнавання зображення номерного знаку

Таким чином при завантаженні користувачем довільного зображення автомобіля, веб застосунок відображає результати роботи інформаційної системи розпізнавання та ідентифікації зображень номерів транспортних засобів, виділяючи із зображення номерний знак автотранспорту, та надає дані про вміст конкретного номерного знаку у текстовому форматі, які надалі можуть використовуватися для безлічі завдань та потреб систем дорожнього управління.

3.7 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи наведено весь процес проектування інформаційної системи для розпізнавання та ідентифікації зображень номерів транспортних засобів. Було створено модель нейронної мережі, проведено навчання моделі на підготовлених тестових зображеннях, та перевірено коректність результатів розробленої системи.

Також було спроектовано графічний інтерфейс для взаємодії користувачів із системою розпізнавання та ідентифікації зображень номерів транспортних засобів, у вигляді веб застосунку, із можливістю завантаження власних зображень транспортних засобів для розпізнання та ідентифікації номерного знаку.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Гігієнічна оцінка умов праці програміста

Державні санітарні норми та правила «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу» розподіляє умови праці на 4 класи.

- 1 клас, оптимальні умови праці;
- 2 клас, допустимі умови праці;
- 3 клас, шкідливі умови праці;
- 4 клас, небезпечні умови праці.

При чому третій клас за рівнем перевищення гігієнічних нормативів та вираженості можливих змін в організмі працівників поділяється на 4 ступеня.

Якщо на робочому місці фактичні значення рівнів шкідливих факторів знаходяться в межах оптимальних або допустимих рівнів, умови праці на цьому робочому місці відповідають гігієнічним вимогам і відносяться відповідно до 1 або 2 класу. Якщо рівень хоча б одного фактора перевищує допустиму величину, то умови праці на такому робочому місці (залежно від величини перевищення та відповідно до гігієнічних критеріїв цієї гігієнічної класифікації праці) як за окремим фактором, так і при їх поєднаній дії, можуть бути віднесені до 1-4 ступенів 3 класу шкідливих або 4 класу небезпечних умов праці.

Шкідливими умовами праці можна вважати такі умови, за яких рівень впливу одного або більше факторів виробничого середовища та/або трудового процесу перевищує допустимий. Шкідливі виробничі фактори поділяються на:

- фізичні фактори;
- хімічні фактори;

- біологічні фактори;
- фактори трудового процесу.

Розглядаючи професійну діяльність програмістів, можна сказати що хімічні та біологічні фактори не будуть відігравати великої ролі при оцінюванні гігієнічних умов праці. До шкідливих факторів трудового процесу відносяться важкість та напруженість праці [47]. Важкість праці характеризується рівнем загальних енергозатрат організму або фізичним динамічним навантаженням. Процес роботи програміста можна віднести до класу легкої роботи за важкістю праці, оскільки він не потребує фізичних навантажень чи вимушених переміщень.

Напруженість праці – характеристика трудового процесу, що відображає навантаження переважно на центральну нервову систему, органи чуттів, емоційну сферу працівника. Напруженість праці визначаються за основними та допоміжними показниками, що є характерними для конкретного робочого місця. Гігієнічна оцінка напруженості праці проводиться шляхом додавання відношень вимірних або розрахованих показників до їх допустимих рівнів, помножених на коефіцієнт значимості показника.

Для праці програміста присутні ряд показників які можуть вказувати на шкідливість умов праці, та напружений вид діяльності, до таким відносяться:

- спостереження за екранами відеотерміналів, від 4 до 6 годин на зміну;
- робота в умовах дефіциту час;
- рішення складних завдань з вибором за алгоритмом.

При виконанні робіт за із персональним комп'ютером до основних фізичних факторів можна віднести: освітлення, виробничий шум та мікроклімат.

Гігієнічна оцінка світлового середовища здійснюється за показниками природного та штучного освітлення. Для допустимих умов праці коефіцієнт природного освітлення має бути не меншим за 0,6 відсотків, згідно з ДБН

В.2.5-28-2006. Штучне освітлення оцінюється за рядом показників (освітленість, прямий відблиск, коефіцієнт пульсації освітлення тощо). Загальна гігієнічна оцінка умов праці за показниками світлового середовища здійснюється на підставі оцінок показників із «природного» та «штучного» освітлення шляхом вибору показника з найвищим ступенем шкідливості.

Ступінь шкідливості умов праці при дії на працівників шуму встановлюється відповідно від величин перевищення нормативів, та з урахуванням їх часових характеристик (постійний, непостійний шум). Рівень звуку еквівалентний звуку до 85 дБА шкідливим вважатися не буде, згідно ДСН 3.3.6.037-99. Для забезпечення допустимих рівнів шуму на робочих місцях програмістів слід застосовувати засоби звукопоглинання.

Нормування параметрів мікроклімату проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Таким чином узагальненими оптимальними параметрами мікроклімату можна вважати:

- температура повітря в приміщенні 22 – 24°C;
- відносна вологість 40 – 60%;
- швидкість руху повітря 0,1 – 0,2 м/с.

Таким чином ступінь шкідливості умов праці через мікроклімат встановлюється відповідно від величин перевищення нормативів, згідно ДСН 3.3.6.042-99.

При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності програміста та можуть виникнути професійні захворювання органів зору, руху чи нервової системи [48]. Таким чином, вивчення та оцінка умов праці програміста є необхідним для запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Також можна зробити висновки що умови праці програміста можуть відноситися до шкідливих через потенційно велику напруженість праці.

4.2 Підвищення стійкості роботи об'єктів логістики у воєнний час

Логістика об'єднує постачання, зберігання, виробництво та збут в один процес. При цьому функції логістики на підприємстві включають в себе аналіз потреби в матеріальних ресурсах для забезпечення потреб виробництва, прогнозування попиту на продукцію, збір даних про потреби ринку. Логістика на підприємстві також має розглядати питання оперативного планування, що має на увазі скорочення запасів при належному рівні виробничої і збутової ефективності підприємства. В Україні і за кордоном існують різні інститути, мета яких – застосування наукового потенціалу логістики на практиці.

Логістична діяльність підприємства – це частина управління ланцюгом поставок, яка планує, впроваджує та контролює ефективний прямий та зворотний потік зберігання товарів, послуг та пов'язаної з ними інформації між точкою походження та точкою споживання з метою задоволення потреб клієнтів. Логістичну діяльність підприємства можна також розглядати і з точки зору реалізації основних логістичних процесів – діяльності з оперативно-календарним планом постачання сировини, матеріалів, транспортні роботи з предметами постачання, діяльності з планом фізичного розподілу під час виробництва продукції, внутрішньозаводським переміщенням матеріалів, сировини, запчастин та готової продукції [49].

У воєнний час значно ускладнюється виконання практично усіх аспектів логістики, що негативно впливає на підприємницьку діяльність. Перша та найголовніша зміна, що сталася в українській логістиці – це усунення «центру тяжіння». Як відомо, найбільший складський хаб в Україні, де зосереджено 70-80% усіх професійних складських площ, був у Київській області. Тут було зосереджено 1,8-2,2 мільйонів квадратних метрів професійних складів.

З початком війни великі компанії, а за ними середні та дрібні, були змушені перевезти свої складські залишки та товари на захід України. Відбувся колосальний відтік до Львівської, Тернопільської, Івано-Франківської областей, де такого обсягу складських площ не існувало в принципі. Приміщень на всіх не вистачало, а ті, які компаніям вдавалося зайняти, на порядок поступалися колишнім площею, організацією простору та рівнем обслуговування.

Таким чином, бізнес та підприємства були змушені змінити складський ланцюжок і тим самим збільшити складність та вартість цих операцій. Можна назвати три основні фактори, що вплинули на бізнес-процеси, пов'язані з логістикою. Якщо раніше товар міг довгий час перебувати на складах, звідки йшло відвантаження, то зараз бізнес почав відвантажувати «з коліс», намагаючись мінімально накопичувати залишки, щоб у разі можливої атаки на склади не було втрат товару.

Зазвичай запуск складу забирає приблизно три місяці: переїзд складу, розгортання ІТ-системи та ІТ-інтеграція, налаштування систем безпеки, відеоспостереження. Зрозуміло, що стільки часу в компаній не було, вони були змушені мігрувати за лічені тижні, а то й дні на невідомі площі. Логістиці, особливо складській, було, м'яко кажучи, складно, але при цьому компанії впоралися досить добре. Включився режим запуску з нуля, закривалися базові потреби, такі як пошуки складів, водіїв, складського персоналу.

Ще однією додатковою проблемою що виникла через воєнний стан є величезна кількість блокпостів та оглядів. Також відсутність чітких правил пересування під час комендантської години, яким логістам можна було їхати вночі, а яким не можна, дії з боку тероборони, нічні пересування транспорту, що доставляє продукти.

У принципі, логістика як така в мирний та воєнний час не дуже відрізняється. Якщо говорити про звичайні продовольчі товари, то і зараз все

йде своєю чергою, все доставляється. Головне – це забезпечення безпеки зон та точок відвантаження, пунктів доставки, транспорту.

Зміни відбулися, але локально чи в рамках окремих бізнес-процесів. Була низка проблем з імпортом, які вже вирішені. Є затримки доставки товарів в окремі райони через окупацію або високі ризики. Відповідно нульова авіаційна та воднотранспортна логістика. Є певні складності в'їзду та виїзду залізницею через відсутність залізничних вагонів для експорту з України товарів аграрної групи та відсутність необхідного обсягу залізничних вагонів по імпорту палива.

Підвищення стійкості роботи об'єктів логістики досягається за рахунок координації дій їх компонентів (внутрішньої координації) та взаємодії із зовнішніми контрагентами (зовнішньої координації), тому для ефективного функціонування логістики у воєнний час необхідне створення координаційного механізму. Виділяють два напрямки досягнення необхідного рівня координації в управлінні логістичними потоками:

- посилення взаємодії між різними функціональними ланками (службами);
- організаційні перетворення в структурі підприємства.

На практиці ці напрямки доповнюють один одного, при цьому використовуються різні методи координації за допомогою розроблених процедур, які регламентують дії менеджерів з управління потоками. Також з цією метою широко використовуються спеціалізовані інформаційні системи, що дозволяють оперативно погоджувати плани постачання, виробництва та збуту у довгостроковій та короткостроковій перспективі і забезпечувати збалансоване поточне регулювання та контроль матеріальних та інших ресурсів з урахуванням змін.

Основою планування та ефективного функціонування підсистеми логістичного обслуговування у воєнний час є розробка процедур обслуговування. Під процедурами розуміють способи та час обслуговування,

вони містять опис суті кожної послуги, процесів її надання та контролю якості і розробляються таким чином, щоб забезпечити достатній рівень якості обслуговування при мінімальних витратах. Задокументовані та затверджені процедури є базою стандартів логістичного обслуговування на підприємстві.

Для покращення роботи об'єктів логістики застосовуються певні засоби вдосконалення логістичних ланцюгів та операцій. Прискорення виконання логістичних операцій та зниження загальних витрат досягається такими засобами:

- відтермінування операцій;
- консолідація відправлень;
- оптимізація маршрутів.

Відтермінування операцій зменшує ризик, що пов'язаний із помилками у прогнозуванні, тому рішення про виробництво або розміщенні запасів відкладається до отримання замовлення. Розрізняють відтермінування виробничих операцій та логістичних операцій [50].

При відтермінуванні виробничих операцій виготовляється стандартна базова продукція у великій кількості, для отримання економії на масштабах, а завершуючі операції (обробка, складання, комплектація, пакування виробу) відкладаються до тих пір, поки не надійде конкретне замовлення. Наприклад, фарбування готових виробів може проводитись після доставки в конкретний регіон з урахуванням вимог споживачів, в роздрібних магазинах є запаси тільки білої фарби та пропонуються різні варіанти тонування фарби для отримання відтінку по вибору конкретного замовника. Такий підхід дозволяє поєднати переваги масового виробництва та гнучке пристосування до індивідуальних запитів споживачів, відповідно зменшується ризик помилок. Коли операції, що завершують технологічний цикл не складні їх доцільно переносити у логістичному ланцюгу ближче до кінцевих ринків збуту і кінцеву обробку продукції, у такому випадку, можна проводити на складах. Це забезпечує високий рівень обслуговування з меншими інвестиціями у

запаси. Можливість відстрочення логістичних операцій створюється завдяки сучасним інформаційним технологіям, які дозволяють з великою точністю та швидкістю виконувати замовлення.

Також для підвищення стійкості роботи об'єктів логістики у воєнний час менеджери мають шукати способи оптимальної консолідації відправлень для забезпечення повного завантаження транспортних засобів, що буде спрямоване на скорочення транспортних витрат. Однак для сучасних логістичних систем, що працюють у відповідь на попит, характерні невеликі та нерегулярні відправлення вантажів. Існує три способи забезпечити ефективну консолідацію відправлень для підвищення ефективності логістики підприємств:

- за цільовим ринком (поєднання невеликих партій відправлень для різних споживачів, що розташовані в одній ринковій зоні);
- за графіком поставок (вантажі відправляються на конкретний ринок тільки за певним графіком);
- за групами вантажовідправників, що обслуговують певну ринкову зону.

Саме тому предметом логістичного аналізу та планування є транспортування вантажів за певним маршрутом між пунктами відправлення та призначення. Аналіз маршрутів має охоплювати як обсяги перевезень, так і число рейсів між пунктами, завантаженість транспорту в обох напрямках. Необхідно різними способами постійно проводити:

- балансування між перевантаженими та недовантаженими рейсами;
- змінювати у разі потреби перевізника або вид транспорту;
- перерозподіляти перевезення вантажів між власним та залученим транспортом;
- збільшувати перевезень зворотними рейсами.

Наведені технології та методичні підходи та засоби використовуються як в стратегічному, так і в оперативному плануванні логістичної діяльності. В

оперативному плануванні зусилля менеджменту зосереджуються на конкретних виконавчих діях, наприклад, на процесах розподілу.

Ще одним із сучасних методів для підвищення стійкості об'єктів логістики є задіяння хмарних технологій та логістичні платформи ланцюгів постачань. Натепер хмарна логістика набуває все більшої популярності. Більше половини постачальників логістичних послуг вже використовують хмарні сервіси. Все більше послуг логістики стає доступними малим підприємствам, яким не потрібно витратитися на пошук ефективних комплексних IT-рішень, натомість можна платити лише за конкретну послугу, яка буде надана у конкретний період часу.

4.3 Висновки до четвертого розділу

У розділі охорона праці та безпека у надзвичайній ситуації кваліфікаційної роботи наведено основні методи проведення гігієнічної оцінки умов праці програміста, та подано оптимальні показники рівнів освітленості, шуму і мікрокліматичних параметрів при роботі за персональним комп'ютером.

Розглянуто перешкоди розвитку логістики для підприємства під час воєнного стану, а також досліджено сучасні рішення щодо підвищення стійкості роботи об'єктів логістики у воєнний час.

ВИСНОВКИ

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр»:

- Висвітлено принципи роботи інформаційних систем розпізнавання та ідентифікації зображень номерів транспортних засобів.
- Описано актуальність даних систем у сучасному світі, та головні сфери у яких вони задіянні.
- Проаналізовано існуючі системи розпізнавання номерних знаків.
- Розглянуто основні труднощі при виконанні ідентифікації номерів автотранспорту.

В другому розділі кваліфікаційної роботи:

- Описано методи проектування системи розпізнавання та ідентифікації номерів транспортних засобів.
- Досліджено найпоширеніші алгоритми для виконання розпізнавання номерних знаків.
- Розглянуто інформацію про побудову систем нейронних мереж.
- Наведено основні методи проведення навчання нейронної мережі.
- Подано загальні принципи виконання оптичного розпізнавання символів.

В третьому розділі кваліфікаційної роботи:

- Описано процес проектування інформаційної системи для розпізнавання та ідентифікації зображень номерів транспортних засобів.
- Створено модель штучної нейронної мережі та виконано її навчання.
- Проведено перевірку коректності результатів розробленої системи.
- Спроековано веб застосунок у якості графічного інтерфейсу.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» проаналізовано небезпечні фактори при виконанні робіт за комп'ютером. Описано методи оцінювання гігієнічних умов праці програміста. Досліджено рішення щодо підвищення стійкості роботи об'єктів логістики у воєнний час.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 E. Ercelebi, Automatic vehicle identification by plate recognition; Transactions on Engineering, Computing and Technology, Vol 9, 2018. 222-225 p.
- 2 Chen, Y. Chung and S. Chen, Automatic license plate recognition; Transactions on Intelligent Transportation Systems, Vol 5, 2017. 42-53 p.
- 3 Фукунага, К. Введення в статичну теорію розпізнавання образів / К. Фукунага. - М.: Головна редакція "Наука", 2015. - 368 с.
- 4 Parker, P. Federl, An approach to license plate recognition; Computer Science Technical Reports, University of Calgary 2019.
- 5 Ткаченко Р. Ю. Теоретичні та прикладні аспекти розробки комп'ютерних систем //Технології проектування і розробки інформаційних систем, 2018 – С. 234.
- 6 Палієнко В. Г. дослідження методів класифікації веб-документів на основі machine learning //ББК 32 I 74. 2017. – С. 146.
- 7 R.E. Woods, S.L. Eddins, Digital image processing using Matlab, Prentice Hall, ISBN 2017. 81-297 p.
- 8 Shashank Arokar, Visual Character Recognition using Artificial Neural; MGM's College of Engineering and Technology, University of Mumbai, India 2014.
- 9 Y. Lecun, B. Boser, J. S. Denker, D. Henderson, Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 2017. 541-551p.
- 10 T. Xiao, J. Zhang, Error-driven incremental learning in convolutional neural network; International Conference on Multimedia, no. 22. ACM, 2016. 77– 86 p.
- 11 Qiang Wu, Wenjing Jia, Samali, M., Computer and Information Technology; 8th IEEE International Conference 2019. 125 – 132 p.
- 12 Dai Yan, Liu Jilin, Intelligent Transportation Systems; Proceedings. IEEE 25-29 Aug. 2015. 325 - 329 p.

- 13 Wu; Lei Chan On, Chan Hon Weng, Machine Learning and Cybernetics, 2015.
- 14 A Method for Chinese License Plate Recognition, Xiaojun Chi, Junyu Dong, Aihua Liu, International Conference on Volume 3, 25-28 June 2016.
- 15 Lopez, J.M., Gonzalez, J., Galindo, C., Cabello, J., Signal Processing and Its Applications; ISSPA 2007. 9th International Symposium. 2017. 1-4 p.
- 16 An Algorithm for License Plate Recognition Using Radial Basis Function Neural Network Bo Li, Zhi-yuan Zeng, 2018. 569 - 572 p.
- 17 Chan Hon, Plate detection and recognition by using color information; Electronic Engineering Department, Ankara University, ANKARA, 2017.
- 18 Distance and Color Invariant Automatic License Plate Recognition System Khan, N.Y.; Imran, A.S.; Ali, N.; Emerging Technologies, 2017. 232-237 p.
- 19 Car license plate recognition through Hausdorff distance technique Juntanasub, R.; Sureerattanan, N.; Tools with Artificial Intelligence, 2015. 5 p.
- 20 Character recognition using parallel BP neural network Feng Yang, Fan Yang; ICALIP 2008. International Conference 2018. 1595-1599 p.
- 21 Chinese License Plate Recognition Using a Convolutional Neural Network Zhihong Zhao, Shaopu Yang; Computational Intelligence and Industrial Application, 2018. 27-30 p.
- 22 Color edge enhancement based fuzzy segmentation of license plates Syed, Y.A., Sarfraz, M.; Information Visualisation, 2015. 227-232 p.
- 23 Розробка адапційного веб-сайту та макету десктопного додатку. [Електронний ресурс] / Ольховатенко Д. С // DSpace. – 2020. – Режим доступу до ресурсу: <https://krs.chmnu.edu.ua/jspui/handle/123456789/1497>.
- 24 Comparison of Feature Extractors in License Plate Recognition, Siti Norul Huda Sheikh, Khalid, Marzuki; Modelling & Simulation, 2017. 502–506 p.
- 25 Efficient Embedded Neural-Network-Based License Plate Recognition System H. Caner, S. Alkar, Vehicular Technology, 2018. 2675 – 2683 p.

26 Graphical Models for Joint Segmentation and Recognition of License Plate Characters Xin Fan, Guoliang Fan; *Signal Processing Letters, IEEE* Volume 16, Issue 1, 2019. 10 – 13 p.

27 License plate character recognition algorithm based on filled function method, Yingtao Xu; *Control and Decision Conference*, 2018. 3886-3891 p.

28 Generative Models for License Plate Recognition by using a Limited Number of Training Samples, Tommaso, C.; *Image Processing*, 2016. 276 – 279 p.

29 License plate recognition system based on morphology, Haiyan Zhao, Haili Zhao, Shizheng Zhang; *Granular Computing*, 2018. 826 - 829 p.

30 New approaches for license plate recognition system, Ramkumar, J., Kamakshi Priya, S.; *Intelligent Information Processing*, 2014. 149 - 152 p.

31 Research of Pattern Matching in License Plate Recognition Ping Wang, Wei Zhang; *Intelligent Information Application Workshops*, 2018. 1089 – 1092 p.

32 Robust license plate recognition based on dynamic projection warping, Feng-Chou Ni; Yon-Ping Chen; *Networking, Sensing and Control*, 2014. 84-88 p.

33 Segmentation of characters on car license plates, Lihong Zheng, Qiang Wu, Wenjing Jia; *Multimedia Signal Processing*, 2018. 399-402 p.

34 The Performance Improvement of License Plate Recognition System, Hye-Youn Lim, Young-Do Lim; *Future Generation Networking*, 2018. 82-85 p.

35 Asif M. Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition; 2019 International Conference on Education Technology and Computer, Singapore.

36 Divya KN, Danti A. Retrieval of Vehicle Owner's Registration Details; *International Journal of Innovative Research*. 2017. 61-66 p.

37 Y. Zhang, D. Irfan, Vehicle Number Plate Recognition Using Mathematical Morphology and Neural Networks; *WSEAS Trans Computing*. 2018. 781–790 p.

38 Automatic Number-Plate Recognition: Neural Network Approach; *Vehicle Navigation and Information Systems Conference*, Yokohama, Japan, 2017.

39 Wei W, Wang M, Huang Z. An Automatic Method of Location for Number-Plate Using Color Features. 2018. 245-266 p.

40 Hsieh CT, Juan YS, Hung KM. Multiple License Plate Detection for Complex Background. Volume 1, Taipei, Taiwan. 28-30 Mar 2015.

41 Kamat V, Ganesan S. An Efficient Implementation of the Hough Transforms for Detecting Vehicle License Plates Using DSP'S; Real-Time Technology and Applications Symposium, Chicago, IL, USA. 15-17 May 2016.

42 S. Mahajan, A Review on Automatic Number Plate Recognition System. Computing Via Pattern Recognition. 2015. 1-6 p.

43 A. Banerjee, K. Roy, A Morphology-Based Approach for Car License Plate Extraction; Annual IEEE India Conference, Chennai, India. 11-13 Dec 2015.

44 J. Hsieh, S. Yu, Morphology Based License Plate Detection from Complex Scenes; Object Recognition Supported by User Interaction, Quebec, Canada. 2017.

45 Жидецький В.Ц. Основи охорони праці / В.Ц. Жидецький – Львів Афіша, 2016 – 320 с.

46 Директива 90/270/ЄЕС Європейського Парламенту та Ради від 29 травня 2010 року про вимоги безпеки при роботі з екранними пристроями [Електронний ресурс]: http://zakon5.rada.gov.ua/laws/show/994_b23

47 Директива 89/391/ЄЕС Європейського Парламенту та Ради від 12 червня 2010 року про впровадження заходів для заохочення вдосконалень у сфері безпеки і захисту здоров'я працівників під час роботи [Електронний ресурс]: http://zakon5.rada.gov.ua/laws/show/994_b23.

48 Геврик Є. О. Охорона праці / Є. О. Геврик . – К.: Ельга, Ніка-Центр, 2017 – 280 с.

49 Крикавський Є. В. Логістика. Основи теорії: підручник для студ. вищ. навч. закладів / Є. В. Крикавський. – Л.: Національний університет "Львівська політехніка", Інтеллект-Захід, 2018. – 416 с.

50 Глушкова Ю. О., Пахомова А. В, «Формування і управління потоками економічної діяльності підприємства.» / Вісник. – Львів, 2019, 7 с.

ДОДАТКИ

Тези конференції

УДК 004.93

Руслан Соломка, Ярослав Литвиненко д.т.н., проф.

Тернопільський національний технічний університет імені Івана Пулюя

ДОСЛІДЖЕННЯ СИСТЕМ РОЗПІЗНАВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ

Ruslan Solomka, Iaroslav Lytvynenko Doctor of Sc., Professor.

RESEARCH OF VEHICLE RECOGNITION SYSTEMS

У країнах, що швидко розвиваються, відповідно разом із технологіями, зростає кількість транспортних засобів. Паралельно також росте потреба в автоматичному розпізнаванні даних транспортних засобів. Відповідно на цю тему було проведено багато досліджень. В працях [1, 2] запропоновано різні методи для досягнення цієї мети.

Ця доповідь стосується аналізу сучасних досліджень у сфері розпізнавання автотранспорту. Загалом можливо розділити такі дослідження на дві категорії – активні системи, та пасивні системи.

Активні системи використовують лазерні, радіочастотні та інші технології для розпізнавання транспортного засобу. У лазерних системах на автомобіль наноситься штрих-код. При використанні радіочастотних методів на транспортному засобі розміщується певний ідентифікатор в який можна додати більш детальну інформацію разом із номерним знаком транспортного засобу. Зображення транспорту даних системі непотрібне, таким чином вона працює практично без помилок та є досить надійною. В останні роки такі радіочастотні станції можуть розташовуватися там, де інші рішення неможливі та різного виду помилки є неприпустимі.

У пасивних системах розпізнавання транспортних засобів відбувається шляхом ідентифікації та обробки зображення номерного знаку автомобіля. Відповідно при великій кількості шумів в зображенні, можуть виникнути невірні результати.

Систему ідентифікації номерних знаків можна використовувати у багатьох напрямках, наприклад стоянки без людського регулювання, контроль безпеки заборонених зон, дотримання правил дорожнього руху та автоматичний збір плати за проїзд. Така система фіксує зображення транспортних засобів та ідентифікує номерні знаки автоматично. Стандартний процес розпізнавання номерних знаків складається з трьох частин:

- визначення місця розташування номерного знаку;
- сегментація символів із зображень номерного знаку;
- розпізнавання сегментованих символів.

З цього можна зробити висновок що активні системи більш надійні для розпізнавання транспортних засобів, адже сонячне світло, тінь, бруд та інші фактори на зображенні номерного знаку ускладнюють його розпізнавання в пасивних системах. В активних системах зображення номерного знаку не використовується, таким чином, ці фактори не впливають на розпізнавання транспортних засобів. Проте пасивні системи є вигіднішими у плані простоти використання та масштабованості.

Література

1. Plate detection and recognition by using color information and ANN, Ziya TELATAR, Emre, Electronic Engineering Department, Ankara University, ANKARA, 2007.

2. A Macao license plate recognition system Cheokman Wu; Lei Chan On; Chan Hon Weng; Tong Sio Kuan; Machine Learning and Cybernetics, 2005. Page(s):4506 – 4510.

УДК 004.93

Руслан Соломка, Ярослав Литвиненко д.т.н., проф.

Тернопільський національний технічний університет імені Івана Пулюя

АНАЛІЗ МЕТОДІВ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ

Ruslan Solomka, Yaroslav Lytvynenko Doctor of Sc., Professor.

ANALYSIS OF NUMBER PLATES RECOGNITION METHODS

Ідентифікація транспортних засобів – це область досліджень, у якій використовуються методи обробки зображень шляхом виявлення та розпізнавання номерних знаків. Типовий процес для ідентифікації автомобіля складається з трьох основних етапів. Це відстеження транспортного засобу, визначення номерного знаку та ідентифікація (розпізнавання) символів номерного знаку.

Ця доповідь стосується дослідження різних методів розпізнавання номерних знаків. Загалом пасивні методи розпізнавання можна розділити на дві групи, ті що використовують штучні нейронні мережі та ті що використовують методи відмінні від нейронних мереж. Для системи на основі нейронної мережі необхідно її навчити на основі тренувальних даних. Система вивчає ці символи, та після тренувань порівнює вхідні дані із вже відомими символами. Відповідно на даний час існує велика кількість методів для розпізнавання номерних знаків.

Метод описаний в роботі [1] найчастіше застосовується у камерах, що перебувають в русі. Умови освітлення, фон і положення транспортного засобу не так важливі для цього розпізнавання символів. Цей метод може локалізувати різні розміри номерних знаків із зображення. Після локалізації номерного знаку (встановлення ділянки кадру), його символи сегментуються. Для розпізнавання символів використовується декілька нейронних мереж.

Інший метод, що описаний в роботі [2] використовується для розпізнавання китайських номерних знаків. Зображення знаків перетворюється на бінарне представлення, з якого потім видаляється шум. Скелет що залишився використовується для виділення особливостей символу. Тоді усі символи нормалізують до розміру 8*16 пікселів. Далі зображення обробляється за допомогою нейронної мережі для розпізнавання окремих символів.

У статті [3] використовується інший алгоритм, розроблений для італійської мови дорожньою компанією. Їх алгоритм класифікації має два етапи. Один з них це фаза налаштування параметрів, а інший – класифікація символів за вбудованими генеративними моделями з використанням коваріаційної матриці. Вони перевірили 7000 різних зображень номерних знаків, та у цьому дослідженні правильна класифікація становить близько 98,1 відсотків.

Проаналізувавши наведені алгоритми можна зробити висновок що використання нейронних мереж наразі є домінуючим методом при вирішенні завдань розпізнавання номерних знаків автотранспорту. Тому розвиток альтернативних методів є доволі актуальним.

Література

1. A versatile low-cost car plate recognition system Lopez, J.M.; Gonzalez, J.; Galindo, C.; Cabello, J.; Signal Processing and Its Applications, 2007. Page(s):1 – 4.
2. Character recognition using parallel BP neural network Feng Yang; Fan Yang; 2008. ICALIP 2008. International Conference on 7-9 July 2008 Page(s):1595 – 1599.
3. Generative Models for License Plate Recognition by using a Limited Number of Training Samples Tommaso, C.; Image Processing, 2006 Page(s):2769 – 2772.

Лістинг 1 – ф

```

import pandas as pd
import xml.etree.ElementTree as xet
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
import os
import cv2
import xml.etree.ElementTree as xet

path = glob('./images/*.xml')

labels_dict = dict(filepath=[], xmin=[], xmax=[], ymin=[], ymax=[])
for filename in path:
    #filename = path[0]
    info = xet.parse(filename)
    root = info.getroot()
    member_object = root.find('object')
    labels_info = member_object.find('bndbox')
    xmin = int(labels_info.find('xmin').text)
    xmax = int(labels_info.find('xmax').text)
    ymin = int(labels_info.find('ymin').text)
    ymax = int(labels_info.find('ymax').text)
    #print(xmin, xmax, ymin, ymax)
    labels_dict['filepath'].append(filename)
    labels_dict['xmin'].append(xmin)
    labels_dict['xmax'].append(xmax)
    labels_dict['ymin'].append(ymin)
    labels_dict['ymax'].append(ymax)

df = pd.DataFrame(labels_dict)
df.to_csv('labels.csv', index=False)

df = pd.read_csv('labels.csv')
filename = df['filepath'][0]

def getFilename(filename):
    filename_image =
    xet.parse(filename).getroot().find('filename').text
    filepath_image = os.path.join('./images', filename_image)
    return filepath_image

image_path = list(df['filepath'].apply(getFilename))
file_path = image_path[0]
img = cv2.imread(file_path)

cv2.namedWindow('example', cv2.WINDOW_NORMAL)
cv2.imshow('example', img)

```

```

cv2.waitKey(0)
cv2.destroyAllWindows()
cv2.rectangle(img, (1093, 645), (1396, 727), (0, 255, 0), 3)
cv2.namedWindow('example', cv2.WINDOW_NORMAL)
cv2.imshow('example', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import load_img,
img_to_array

labels = df.iloc[:,1:].values

data = []
output = []
for ind in range(len(image_path)):
    image = image_path[ind]
    img_arr = cv2.imread(image)
    h,w,d = img_arr.shape
    # preppocessing
    load_image = load_img(image, target_size=(224, 224))
    load_image_arr = img_to_array(load_image)
    norm_load_image_arr = load_image_arr/255.0 # normalization
    # normalization to labels
    xmin,xmax,ymin,ymax = labels[ind]
    nxmin,nxmax = xmin/w, xmax/w
    nymin,nymax = ymin/h, ymax/h
    label_norm = (nxmin,nxmax,nymin,nymax) # normalized output
    # ----- append
    data.append(norm_load_image_arr)
    output.append(label_norm)

X = np.array(data, dtype=np.float32)
y = np.array(output, dtype=np.float32)
x_train,x_test,y_train,y_test =
    train_test_split(X,y,train_size=0.8,random_state=0)
x_train.shape,x_test.shape,y_train.shape,y_test.shape

from tensorflow.keras.applications import MobileNetV2,
InceptionV3, InceptionResNetV2
from tensorflow.keras.layers import Dense, Dropout, Flatten,
Input
from tensorflow.keras.models import Model
import tensorflow as tf

inception_resnet =
    InceptionResNetV2(weights="imagenet",include_top=False,

        input_tensor=Input(shape=(224,224,3)))
inception_resnet.trainable=False
# -----

```

```

headmodel = inception_resnet.output
headmodel = Flatten()(headmodel)
headmodel = Dense(500,activation="relu")(headmodel)
headmodel = Dense(250,activation="relu")(headmodel)
headmodel = Dense(4,activation='sigmoid')(headmodel)
# ----- model
model = Model(inputs=inception_resnet.input,outputs=headmodel)
model.compile(loss='mse',optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4))

from tensorflow.keras.callbacks import TensorBoard
tfb = TensorBoard('object_detection')

history =
    model.fit(x=x_train,y=y_train,batch_size=10,epochs=100,
             validation_data=(x_test,y_test),callbacks=[tfb])

history =
    model.fit(x=x_train,y=y_train,batch_size=10,epochs=200,
             validation_data=(x_test,y_test),callbacks=[tfb],initial_epoch
             =101)

model.save('./models/object_detection.h5')

```