

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
в.о. завідувача кафедри  
\_\_\_\_\_ Рязанцев О.І.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**МАГІСТЕРСЬКА РОБОТА**

НА ТЕМУ:

**Метод ідентифікації транспортних засобів**

---

---

---

Освітній рівень “Магістр”  
Спеціальність 122 “Комп’ютерні науки”

Науковий керівник роботи:

\_\_\_\_\_  
(підпис)

О.І.Рязанцев

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_  
(підпис)

Я.О.Критська

(ініціали, прізвище)

Студент:

\_\_\_\_\_  
(підпис)

М.А. Запорожський

(ініціали, прізвище)

Група:

КН-19дм

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітній рівень магістр

Напрямок підготовки \_\_\_\_\_

(шифр і назва)

Спеціальність 122 "Комп'ютерні науки"

(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Т.в.о. завідувача кафедри \_\_\_\_\_

В.С.Кардашук

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Запорожському Микиті Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Метод ідентифікації транспортних засобів

керівник проекту (роботи) Рязанцев Олександр Іванович, д.т.н., проф.

(прізвище, м.я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «5» 10 2020 р. № 140/15.15

2. Строк подання студентом роботи 10.01.2021

3. Вихідні дані до роботи Матеріали науково-дослідної практики,

науково-методична література; дані інтернет-мережі; тестові файли для  
завантаження: 3 файла типу jpeg, 3 файла типу png, 1 файла типу bmp

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Розпізнавання образів та його застосування, система контролю і розпізнавання автомобільних номерів, реалізація та аналіз результатів, охорона праці та безпека в надзвичайних ситуаціях, висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Електронні плакати

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 14.10.2020

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Розробка технічного завдання	02.09.2020-15.09.2020	
2	Аналіз літератури з досліджуваної проблеми	16.09.2020-22.09.2020	
3	Аналіз технічних засобів	23.09.2020-25.09.2020	
4	Розробка алгоритмів	26.09.2020-10.10.2020	
4	Програмна реалізація	11.10.2020-06.11.2020	
5	Аналіз результатів дослідження	07.11.2020-25.11.2020	
6	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.11.2020-1.12.2020	
7	Оформлення пояснювальної записки, автореферату та презентації	2.12.2020-09.01.2021	

Студент

\_\_\_\_\_ ( підпис )

Запорожський М.А.

\_\_\_\_\_ (прізвище та ініціали)

Науковий керівник

\_\_\_\_\_ ( підпис )

Рязанцев О.І.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Запорожський М.А. Метод ідентифікації транспортних засобів.

Метою роботи є розробка програмного застосунку, що базується на згортковій нейронній мережі та дозволяє розпізнавати номерні знаки автомобіля на зображеннях.

Об'єктом дослідження є нейромереві методи, що можуть бути використані для розпізнання об'єктів на зображенні, зокрема автомобільних номерних знаків. Робота присвячена вирішенню завдання розпізнавання зображень та ідентифікації автомобільних номерних знаків для подальшого їх застосування у різних сферах.

Був проведений аналіз та обґрунтування отриманих результатів, а також був зроблений висновок щодо доцільності використання даної системи.

У результаті атестаційної роботи було розроблено систему для пошуку номерних знаків на зображенні та детектування типу пересувного засобу. Був здійснений огляд, розбір та аналіз існуючих нейромеревих методів для розпізнання тексту на зображеннях.

**Ключові слова:** розпізнавання зображень, нормалізація зображень, сегментація зображень, методи розпізнавання, нейронна мережа, згорткова нейронна мережа.

## ABSTRACT

Zaporozhsky MA Vehicle identification method.

The aim is to develop an application which is based on Convolutional Neural Network and also allows to recognize car license plates on images.

The object of the research is neural network and methods which can be used for object recognizing on images, in particular car numbers. The work is devoted to the task decision of image recognition and identification of car license plates for their further applying in various spheres.

An analysis and justification of the results were conducted, and a conclusion was made as to the feasibility of using this system.

As a result of the work, the system has been developed for searching car license plates on images and vehicle detection. The review and analysis of existing neural network methods for text recognizing on images were conducted.

**Keywords:** image recognition, image normalization, image segmentation, recognition methods, neural network, convolutional neural network.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ .....	5
ВСТУП.....	6
1 РОЗПІЗНАВАННЯ ОБРАЗІВ ТА ЙОГО ЗАСТОСУВАННЯ .....	7
1.1 Теорія розпізнавання образів.....	7
1.1.1 Теоретичні відомості .....	7
1.1.2 Підходи до розпізнавання образів .....	9
1.2 Сфери застосування.....	10
1.3 Постановка задачі дослідження.....	12
2 СИСТЕМА КОНТРОЛЮ І РОЗПІЗНАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ .....	14
2.1 Система розпізнавання номерів та її характеристики .....	14
2.2 Типові рішення.....	14
2.3 Алгоритми та технології розпізнавання номерних знаків .....	16
2.4 Процедура розпізнавання.....	18
2.4.1 Система розпізнавання .....	18
2.4.2 Найпоширеніші методи розпізнавання .....	20
2.5 Нормалізація та сегментація.....	21
2.6 Використання нейронних мереж.....	26
2.6.1 Навчання нейронних мереж.....	28
2.6.2 Багатошарові нейронні мережі .....	30
2.6.3 Нейронні мережі високого порядку .....	31
2.6.4 Радіально-базисні нейронні мережі.....	32
2.6.5 Персептрон .....	33
2.6.6 Когнітрон .....	34
2.6.7 Неокогнітрон .....	35
2.6.8 Згорткові нейронні мережі .....	36
3 РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ .....	38
3.1 Обґрунтування вибору середовища програмної реалізації .....	38
3.1.1 Java - технології.....	38
3.1.2 Середовище розробки IntelliJ IDEA .....	40
3.1.3 Фреймворк Maven .....	41
3.2 Програмна реалізація.....	42
3.3 Інструкція користувача .....	43
3.4 Тестування розробленої моделі.....	43

	4
4 ОХОРОНА ПРАЦІ .....	53
4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектного об'єкту, що мають вплив на персонал .....	53
4.2 Заходи щодо техніки безпеки .....	54
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці.....	57
4.4 Рекомендації по пожежній безпеці .....	60
ВИСНОВКИ.....	64
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....	66
ДОДАТОК А. Фрагмент лістингу конфігураційного файлу pom.xml .....	69
ДОДАТОК Б. Фрагмент лістингу файлу Recognizer.java.....	71
ДОДАТОК В. Електронні плакати .....	78

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

- ПК – персональний комп’ютер
- НМ – нейронна мережа
- UI – User interface – інтерфейс користувача
- ОС – операційна система
- ПЗ – програмне забезпечення
- ЗНМ – згортова нейронна мережа
- БНМ – багатошарова нейронна мережа
- LPR – License Plate Recognition - система автоматичного розпізнавання номерів
- ANPR – Automatic number plate recognition system – система автоматичного розпізнавання цифрових таблиць
- OCR – Optical Character Recognition – оптичне розпізнавання символів
- CNN – Convolutional Neural Network – згортова нейронна мережа

## ВСТУП

У наш час комп'ютеризація розвивається швидкими темпами і відіграє значну роль в суспільстві. Комп'ютерні технології допомагають автоматизувати широке коло процесів, які в недалекому минулому покладалися на людину. Інформаційні технології використовуються в промисловості, в транспорті, в побуті та ін.

Сьогодні важливим аспектом безпеки дорожнього руху та контролю є ідентифікація автомобілів по їх реєстраційному номерному знаку. Системи розпізнавання номерних знаків мають різні сфери застосування, такі як автотранспортні підприємства, контроль в'їзду на територію підприємства і переміщення транспортних засобів на об'єктах з обмеженим доступом, заправні станції, контроль швидкості руху, автомобільні стоянки.

Основна мета даної роботи – дати користувачеві можливість розпізнати автомобільний номер на зображенні у десктопному застосунку, та отримати інформацію про регіон автомобілю без додаткового пошуку. Застосунок розроблений для операційної системи Windows за допомогою мови програмування Java з використанням нейромережевого підходу. Користувач зможе завантажувати зображення, отримувати автомобільний номер та додаткову інформацію.

Актуальність завдання розпізнавання номерних знаків транспортних засобів полягає в тому, що з кожним днем збільшується потреба автоматизації контролю в'їзду на територію підприємств, контролю швидкості руху і визначення порушення правил дорожнього руху. В даний час існує досить велика кількість систем визначення номерних знаків, але не всі з них є якісною і надійною продукцією. Системи з високою швидкістю і точністю розпізнавання є комерційними, засекреченими і дорогими, що не дозволяє здійснити їх масове впровадження.



# 1 РОЗПІЗНАВАННЯ ОБРАЗІВ ТА ЙОГО ЗАСТОСУВАННЯ

## 1.1 Теорія розпізнавання образів

### 1.1.1 Теоретичні відомості

З розвитком обчислювальної техніки стало можливим вирішити ряд завдань, що виникають в процесі життєдіяльності, полегшити, прискорити, підвищити якість результату. Наприклад, поява роботизованих систем, взаємодія людини з комп'ютером, робота різних систем життєзабезпечення. Проте, слід зазначити, що забезпечити задовільний результат в деяких завданнях (розпізнавання рукописного тексту, розпізнавання об'єктів, що швидко рухаються) в даний час не завжди вдається. Таким чином, у даній роботі обговорюються принципи і методи, що застосовуються в обчислювальній техніці для виконання поставленого завдання.

Теорія розпізнавання образів існує як розділ інформатики та суміжних дисциплін, що розвиває методи класифікації та ідентифікації об'єктів різної природи: сигналів, ситуацій, предметів, що характеризуються кінцевою множиною деяких ознак.

Проблема розпізнавання образів також опиняється у полі міждисциплінарних досліджень – в тому числі в зв'язку з роботою зі створення штучного інтелекту, а також часто використовується при вирішенні практичних завдань галузі комп'ютерного зору.

При постановці класичної задачі розпізнавання образів прийнято використовувати математичну мову, ґрунтуючись на логічних міркуваннях і математичних доказах. В протилежність до цього підходу, існують методи розпізнавання образів з використанням машинного навчання і штучних нейронних мереж, сформовані не такими суворо формалізованими підходами до розпізнавання, але, демонструють не гірший, а в деяких випадках значно перевершує класичні методи результат.

Характер проблематики розпізнавання образів також обмежує сферу застосування різних традиційних методів вузькими спеціалізованими напрямками, в кожному з яких найбільш ефективними виявляються одні методи і неефективними інші. Цей фактор обумовлює складність загального вирішення проблеми індукції і в цій дисципліні.

Завдання пошуку зображення за зразком є частиною більш загальної задачі розпізнавання образів. При ненаправленому і несистематичному пошуку «схожих» об'єктів з множини об'єктів, їх можна перераховувати безкінечно довго і не прийти до завершення із заданою вірогідністю. В окремих випадках об'єкти характеризуються такими ідентифікаційними параметрами, як форма, колір, положення, рухливість, відмінні риси та їх комбінації. Залежно від цих факторів об'єкти піддаються класифікації. Часто

стоїть не глобальне завдання класифікації всіх оточуючих об'єктів, а необхідність виділити в поступаючому відео-поточі об'єкти певного роду.

Розпізнавання образів (об'єктів, ситуацій, сигналів, явищ або процесів) – завдання ідентифікації об'єкта або визначення будь-яких його властивостей по його зображенню (оптичне розпізнавання) або аудіозапису (акустичне розпізнавання).

Одним з базових понять, що не має конкретного формулювання є поняття множини. У комп'ютері множина представляється набором неповторюваних однотипних елементів. Слово «неповторювані» означає, що якийсь елемент присутній в множині, або відсутній. Універсальна множина включає всі можливі для розв'язуваної задачі елементи, порожня множина не містить жодного.

Образ – класифікаційне угруповання в системі класифікації, яка об'єднує певну групу об'єктів за певною ознакою [18]. Образи мають характерну властивість, що виявляється в тому, що ознайомлення з кінцевим числом явищ з одного і того ж безлічі дає можливість дізнаватися як завгодно велике число його представників. Образи мають характерні об'єктивні властивості в тому сенсі, що різні люди, які навчаються на різному матеріалі спостережень, здебільшого однаково і незалежно один від одного класифікують одні і ті ж об'єкти. У класичній постановці задачі розпізнавання універсальна множина розбивається на частини-образи. Кожне відображення будь-якого об'єкта на сприймаючі органи системи, незалежно від його положення щодо цих органів, прийнято називати зображенням об'єкта, а множини таких зображень, об'єднані певними загальними властивостями, являють собою образи.

Методика віднесення елемента до якого-небудь образу називається вирішальним правилом. Ще одне важливе поняття – метрика, спосіб визначення відстані між елементами універсальної множини. Чим менша ця відстань, тим більше схожими є об'єкти – те, що ми розпізнаємо. Зазвичай елементи задаються у вигляді набору чисел, а метрика – у вигляді функції. Від вибору представлення образів і реалізації метрики залежить ефективність програми, бо один алгоритм розпізнавання з різними метриками буде помилятися з різною частотою.

Навчанням зазвичай називають процес вироблення в деякій системі тієї чи іншої реакції на групи зовнішніх ідентичних сигналів шляхом багаторазового впливу на систему зовнішньої коректування. Таке зовнішнє коригування в навчанні прийнято називати «заохоченнями» і «покараннями». Механізм генерації цього коригування практично повністю визначає алгоритм навчання. Самонавчання відрізняється від навчання тим, що в цьому випадку додаткова інформація про правильність реакції системі не повідомляється.

Адаптація – це процес зміни і налаштування параметрів і структури системи, а можливо і керуючих впливів, на основі поточної інформації з метою досягнення певного стану системи при початковій невизначеності і мінливих умовах роботи.

Навчання – це процес, в результаті якого система поступово набуває здатність відповідати потрібним реакціям на певні сукупності зовнішніх впливів.

### **1.1.2 Підходи до розпізнавання образів**

Як перший підхід можна зазначити метод перебору. У цьому випадку проводиться порівняння з базою даних, де для кожного виду об'єктів представлені різноманітні модифікації відображення. Наприклад, для оптичного розпізнавання образів можна застосувати метод перебору виду об'єкта під різними кутами, масштабами, зміщеннями, деформаціями. Для літер потрібно перебирати шрифт, властивості шрифту. У разі розпізнавання звукових образів, відповідно, відбувається порівняння з деякими відомими шаблонами (наприклад, слово, вимовлене кількома людьми).

Другий підхід – знайти контур об'єкта й досліджувати його властивості (зв'язність, наявність кутів). Існує безліч алгоритмів для виділення контурів, такі як: оператор Собеля, оператор Лапласа, оператор Робертса, оператор Прюїтт і оператор Кенні. Детектування контурів необхідно в тому випадку, коли є досить складне зображення і використовуючи малі інструменти бібліотеки комп'ютерного зору необхідно виділити об'єкти на цьому зображенні для подальшої роботи з ними [19].

Одним із важливих підходів являється використання штучних нейронних мереж (багатошарові перцептрони, мапи Кохонена, рекурентні мережі, мережі квантування). Цей метод вимагає спеціальної структури НМ, яка буде враховувати особливості даної задачі або потребує великої кількості прикладів із правильними відповідями (прикладів задач розпізнавання).

Також існує експертний метод, заснований на безперервному навчанні експертної системи в процесі експлуатації.

## 1.2 Сфери застосування

Існуючі задачі розпізнавання образів:

- розпізнавання тексту;
- розпізнавання автомобільних номерів;
- розпізнавання об'єктів на картах;
- розпізнавання облич та інших біометричних даних;
- розпізнавання штрих-кодів;
- розпізнавання мови.

Розпізнавання образів – перспективний напрямок в рекламі і маркетингу. Нейронні мережі дозволяються за лічені години дізнатися речі, для пошуку яких в інших випадках потрібна велика команда професіоналів і тижні, а то й місяці досліджень. Наприклад, сервіс YouScan, система моніторингу соціальних медіа, відстежує згадки брендів в соціальних мережах. Причому робить це не тільки в тексті, але і на фотографіях, а також допомагає зробити певні висновки про продукт. За допомогою розпізнавання образів на фото знаходять певні закономірності що можуть стати в нагоді для таргетування реклами.

Розпізнавання образів на камерах міського відеоспостереження – найперспективніше використання машинного зору. Такі системи розумного відеоспостереження можуть бути використані з метою ідентифікації злочинців в місцях масового скупчення людей, керування доступу до закритих територій, а також використовувати зібрану інформацію для різних досліджень.

Розпізнавання образів вже стало справжнім проривом в медицині – у багатьох випадках комп'ютери помічають речі, які пропускають навіть найдосвідченіші лікарі. Вони виступають своєрідними помічниками, чия «технічна» думка підтверджує гіпотезу лікаря або дає привід для більш глибоких досліджень.

Створена Google нейромережа Inception аналізує мікроскопічне дослідження біопсії лімфатичних вузлів в пошуку ракових клітин в молочних залозах. Для людини це дуже довгий і трудомісткий процес, в ході якого легко помилитися або пропустити щось важливе, так як в деяких випадках розмір зображення становить 100 000 x 100 000 пікселів. Нейромережа Inception забезпечує чутливість близько 92% проти 72% у лікаря. Нейромережа не втратить з уваги усі підозрілі ділянки знімків, хоча і допускаються помилкові спрацьовування, які пізніше відфільтрує лікар.

Розпізнавання об'єктів в автомобілях – це необхідна частина систем безпеки ADAS (Advanced driver-assistance systems). ADAS можуть бути реалізовані як складними

засобами, на зразок радара і інфрачервоних датчиків, так і за допомогою монокулярної камери. Однієї відеокамери цілком достатньо для того, щоб автомобіль в реальному часі зміг розпізнати пішоходів, знаки та світлофори. Однак таке розпізнавання «на льоту» – завдання, що потребує значних витрат ресурсів, для виконання якої потрібен спеціалізований процесор. Toshiba вже протягом декількох років розвиває серію таких процесорів. Вони будують тривимірну модель на основі рухомого зображення з однієї камери, і тим самим помічають невідомі перешкоди на дорозі. Адже якщо неймережа навчена розпізнавати тільки людей, розмітку та знаки, ті перешкоди, що знаходяться на дорозі не будуть розпізнані і розцінені, як небезпека.

У дронах розпізнавання об'єктів використовується як в розважальних, так і в наукових цілях. У 2015 році став досить відомим дрон Lily з автоматичним включенням двигунів при підкиданні і функцією стеження за власником. Lily направляє об'єктів на господаря, незалежно від траєкторії і швидкості його пересування. Хоча до розпізнавання образів ця функція Lily не мала ніякого відношення, так як дрон стежив не стільки за образом людини, скільки за пультом управління, який був надітий на руку власника.

Дрони з розпізнаванням зображень використовуються і для більш серйозних речей. Наприклад, норвезька компанія eSmart Systems розробила інтелектуальні рішення для енергомереж. В рамках одного з їхніх проєктів - Connected Drone - дрони використовуються для пошуку несправностей на лініях електропередач. Навчені розпізнаванню елементів енергомереж, вони перевіряють цілісність проводів, ізоляторів та інших частин ЛЕП (рис. 1.1). Це особливо важливо для швидкої локалізації несправності, коли від лінії залежить електропостачання міста або підприємства. З огляду на те, що часто ЛЕП побудовані в важкодоступних місцях, послати декілька дронів на пошук несправності набагато ефективніше, ніж послати людей.

Також дрони можуть бути застосовані у військових цілях – для виявлення і супроводу військових об'єктів.



Рисунок 1.1 – Виявлення дронами пошкоджень на ЛЕП

Також досить важливим на даний час є розпізнавання людських облич для захисту інформації, наприклад персональних даних. Одна з таких технологій – автоматичне розпізнавання осіб. Ця технологія аналізує головні особливості особи, створює його математичну репрезентацію, а потім порівнює з базою даних відомих осіб для визначення можливих збігів.

Обробка супутникових зображень є нараз досить актуальним. Сюди відносяться системи дистанційного зондування поверхні землі, різноманітні метеорологічні та системи контролю надзвичайних ситуацій. На сьогоднішній день актуальною задачею також є розробка методів автоматичного розпізнання на екрані радіолокатору градових хмар з послідувачим контролем переміщення цих об'єктів.

### **1.3 Постановка задачі дослідження**

Основною метою магістерської роботи являється аналіз існуючих нейромережових алгоритмів та методів розпізнання символів на зображеннях у реальному масштабі часу та розробка застосунку для реалізації одного з алгоритмів детектування номерних знаків для їх подальшої ідентифікації та використання у різних напрямках. Наприклад, для порівняння з базою даних машин, що знаходяться у розшуку, або для контролю в'їзду на закриті території підприємств. Також отримані дані можна використовувати у цілях збору даних та отримання різних статистичних залежностей.

В процесі виконання даної роботи необхідним є аналіз методів розпізнання та систем, що їх реалізують. У даній роботі будуть детально розглядатися процес розробки програмних продуктів, створених для обробки зображень.

Для досягнення мети необхідно вирішити такі завдання:

- вибрати найбільш актуальний нейромережовий метод для розпізнавання номера на зображенні;
- проаналізувати та вибрати середу розробки застосунку;
- розробити та реалізувати алгоритм для вирішення поставленої задачі у реальному часі;
- дослідити якість програмної реалізації, визначити оптимальні параметри його функціонування;
- реалізувати головні задачі. А саме: завантаження зображення, розпізнання автомобільного номеру, пошук додаткової інформації по номеру;

- розробити застосунок, що у подальшому може інтегруватися на різні типи та версії ОС;

- розробити зручний для користування інтерфейс застосунку;

- протестувати програму на різних версіях ОС.

Визначимо основні завдання, які необхідно вирішити в атестаційній роботі:

- аналіз існуючих методів розпізнання;

- аналіз існуючих нейромережових методів для розпізнавання зображень;

- виявлення переваг та недоліків кожного з нейромережових методів;

- вибір нейромережового методу, що максимально ефективно підійде для вирішення задачі розпізнавання автомобільного номера на зображенні у реальному часі;

- вибір актуального середовища для розробки, опис його переваг та недоліків;

- вибір і аналіз додаткових технологій, що можуть знадобитися для ефективного процесу розробки;

- реалізація;

- детальний розбір реалізованого продукту, проведення тестування, визначення вектору подальшого розвитку.

Таким чином необхідно розробити програмний продукт для розпізнавання номерних знаків на зображенні у реальному часі, що буде використовувати найбільш актуальні та ефективні методи і технології. Також провести аналіз результатів роботи та швидкодії застосунку, визначити існуючі переваги та недоліки, а також зробити висновки щодо доцільності даної системи та зазначити вектор можливого розвитку програмного застосунку.

## **2 СИСТЕМА КОНТРОЛЮ І РОЗПІЗНАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ**

### **2.1 Система розпізнання номерів та її характеристики**

В даний час в світі на дорогах знаходиться понад півмільярда автомобілів. Всі ці транспортні засоби мають унікальний ідентифікаційний номер в якості основного розпізнавального знаку. Ідентифікаційний номер транспортного засобу фактично є реєстраційним номером, що дає законне право на участь у дорожньому русі.

Проблема ідентифікації автомобіля за реєстраційним номерним знаком є важливим аспектом контролю і забезпечення безпеки дорожнього руху. Продукти, здатні вирішувати це завдання, затребувані у самих різних сферах. Прикладом можуть служити автотранспортні підприємства, автомобільні стоянки, гаражні кооперативи, котеджні селища, заправні станції, пункти контролю в'їзду на територію об'єкта тощо.

Коли йдеться про систему автоматичного розпізнання номерів (License Plate Recognition, LPR), мається на увазі програмний або апаратно-програмний комплекс, який реалізує алгоритми автоматичного розпізнавання номерних знаків для реєстрації подій, пов'язаних з переміщенням автомобілів, тобто для автоматизації введення даних і їх подальшої обробки.

Строго кажучи, LPR-система – це пристрій, який реєструє проїзд транспортного засобу, зчитує його реєстраційний номер і виводить його в ASCII-систему обробки даних [20].

В даний час існує досить багато LPR-систем з різним рівнем якості розпізнавання, швидкодії і спектром послуг додаткових функцій. Продукти, що володіють високою швидкістю і точністю розпізнавання, як правило, дуже дорогі. Їх висока вартість не дозволяє здійснити масове впровадження. Розглянемо загальні принципи, покладені в основу розпізнавання автомобільних номерів, щоб зрозуміти причини високої вартості таких систем.

### **2.2 Типові рішення**

Мабуть, одним з наймасовіших і затребуваних застосувань LPR-систем є контроль доступу на територію об'єкта і автостоянки. В рамках даних програм ми вже можемо



виділити основний тип систем розпізнавання номерів з типовою конфігурацією технічного забезпечення і системним обладнанням.

Прикладом систем розпізнавання номерів може послужити система розпізнавання номерів контролю доступу. Варто відзначити, що в більшості випадків система розпізнавання номерів є лише частиною інтегрованої системи контролю доступу.

Автомобіль наближається до воріт зони, що охороняється, на територію якої він хоче проїхати. На контрольно-пропускному пункті встановлений шлагбаум і світлофор, палаючий червоним, інформуючи про те, що проїзд заборонений. Крім того, на в'їзді встановлена CCTV камера відеоспостереження. Шлагбаум, світлофор і камера підключені до комп'ютера, на якому встановлено LPR-систему, яка координує операції системи контролю доступу.

Якщо система реєструє наближення транспортного засобу, LPR-модуль вважає номерний знак в межах зони контролю. Після зчитування номера, LPR-модуль передає розпізнаний номер для подальшого прийняття рішень.

Далі система контролю і управління доступом відправляє номер в модуль бази даних. Модуль бази даних звіряє отриманий номер зі списками прав доступу і повертає прапор «доступ дозволено» або «доступ заборонено». Залежно від типу прапора застосунок контролю доступу відкриває шлагбаум і запалює зелене світло чи ні (рис. 2.1).

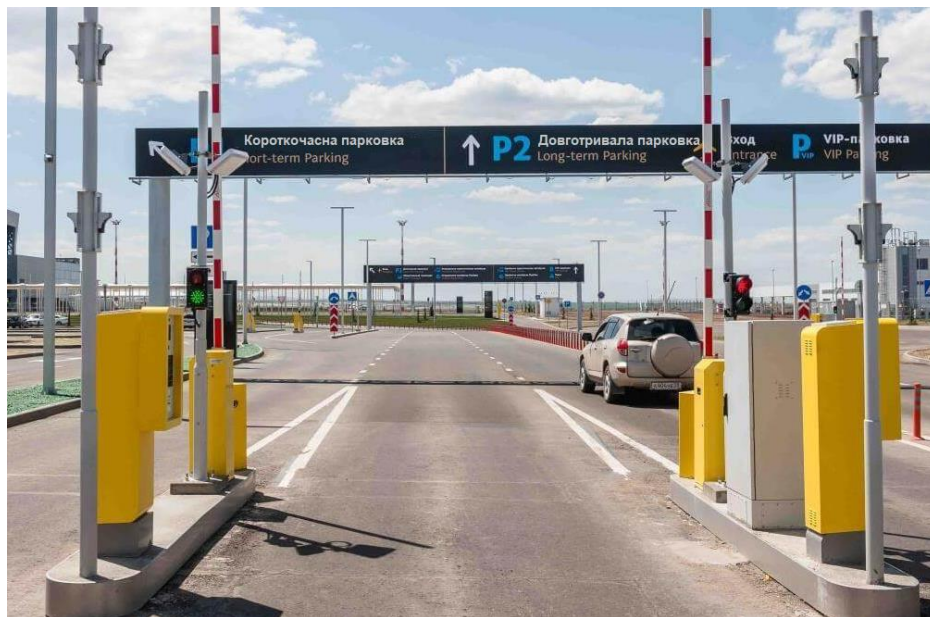


Рисунок 2.1 – Відкриття шлагбауму та запалення зеленого світла після встановлення рішення «доступ дозволено»

Крім того, програма контролю доступу може відправляти модулю бази даних такі додаткові відомості як дата або час доступу для збереження в журналі подій. Після того як автомобіль проїжджає (або їде), система повертається на вихідну позицію і чекає наступного.

### 2.3 Алгоритми та технології розпізнавання номерних знаків

Безперечно, основою будь-якої LPR-системи є використовувані алгоритми розпізнавання. Кваліфікація розробників в області сучасної вищої математики, обробки зображень, програмуванні і технологіях оптимізації програм, а також наявність істотного досвіду роботи – всі ці фактори визначають характеристики LPR-системи, такі як:

- ймовірність розпізнавання;
- швидкість обробки;
- здатність розпізнавати різні типи номерних знаків;
- здатність працювати з зображеннями різної якості.

Розпізнавання державних реєстраційних знаків є нетривіальним завданням з області технічного зору та штучного інтелекту. Використовувані алгоритми локалізації номерного знаку і його розпізнавання, як правило, є комерційною таємницею і, природно, не публікуються. Лише деякі компанії називають їх типи і публікують послідовність дій.

Нижче послідовно наведені ключові етапи розпізнавання автомобільного номера:

- приведення вихідного зображення до виду, який не залежить від умов реєстрації зображення (ступінь освітленості, нерівномірність розподілу яскравості від джерел світла, розмитість, зашумленість тощо);
- виділення на отриманому зображенні областей-кандидатів, потенційно містять пластину з номером;
- проведення детального аналізу областей-кандидатів на основі формального уявлення масштабних характеристик номерний пластини і скорочення простору для подальшого пошуку;
- приведення до стандартного розміру графічного зображення номерний пластини з корекцією якості зображення;
- попереднє визначення типу номерний пластини (в прив'язці до діючих стандартів);

- витяг окремих символів і їх розпізнавання (аналіз символів за ключовими характеристиками, незалежним від масштабу, використовуваного шрифту, геометричних спотворень і розривів);
- уточнення результатів розпізнавання на основі інформації про тип номера і за результатами з попередніх кадрів.

Результатом роботи алгоритму є інформація про проїзд транспортного засобу, що містить рядок з розпізнаним номером, стоп-кадр з найкращим зображенням транспортного засобу, інформацією про час проїзду автомобіля і тому подібне.

З представленої послідовності кроків видно, що вихідні дані для розпізнавання номера не обмежуються тільки візуальним зображенням. У світі існує велика кількість видів номерних знаків, що розрізняються:

- використовуваними шрифтами (знаки з символами різного розміру, латинські, кириличні та інші шрифти);
- кольором фону і символів (чорні символи на світлому тлі або білі символи на темному тлі);
- кількістю рядків у номері (однорядкові, дво- або три- рядкові);
- наявністю або відсутністю коду позначення регіону або спеціальної позначки.

Існуючі типи автомобільних знаків України зображено на рисунку 2.2.



Рисунок 2.2 – Різні види автомобільних реєстраційних знаків

Облік цих відмінностей в номерних знаках дає суттєву перевагу тим розробникам LPR-систем, які використовують цю додаткову інформацію у логіці своїх алгоритмів

розпізнавання. Інформація про структуру знаку та його синтаксисі дозволяє істотно підвищити ймовірність правильного розпізнавання, одночасно знижуючи вимоги до якості алгоритмів вилучення і розпізнавання окремих символів.

## **2.4 Процедура розпізнавання**

Розпізнавання образів – завдання ідентифікації об'єкта, визначення будь-яких його властивостей по його зображенню та віднесення його до множини образів за визначеним правилом. Наприклад, у нас є різне насіння, зібране після врожаю, яке треба поділити на групи. Припустимо, що ми будемо робити це відповідно до розміру насіння. Тобто ми маємо об'єкт «насіння», і згідно зі властивістю «розмір» ми віднесемо насіння або до образу «соняшник», або до образу «просо».

Виділимо найбільш важливі кроки у процедурі розпізнавання:

- сприйняття образу. На цьому етапі проводять отримання значень характеристичних властивостей об'єкта;
- попередня обробка. На цьому етапі проводиться нормалізація. Основними труднощами є спотворення різного роду: геометричні, дія фона та шумів [12];
- виділення характеристик (індексація). На цьому етапі вимірюються характеристичні властивості об'єкта (вимірюємо довжину риби та її колір);
- сегментація. В найпростішому випадку сегментація зводиться до пошуку «однорідних» областей [5];
- класифікація (прийняття рішення).

### **2.4.1 Система розпізнавання**

В даний час існує безліч програмних систем і апаратно-програмних комплексів розпізнавання реєстраційних номерів автомобілів по фотографіях з відеокамер спостереження. Практично в кожній системі розпізнавання процес розпізнавання реєстраційного номера автомобіля складається з наступних основних етапів: попередня обробка, сегментація і розпізнавання. Етап розпізнавання символів є найважливішим і може бути здійснений трьома типами методів – шаблонними, структурними і ознаковими.

В ознакових методах усереднене зображення кожного символу представляється як об'єкт в  $n$ -вимірному просторі ознак. Отриманий  $n$ -мірний вектор порівнюється з еталонними векторами, і зображення відноситься до найбільш підходящому з них. Одним з ознакових методів є адаптивне розпізнавання з використанням нейронних мереж. Цей метод може бути використаний і для розпізнавання елементів реєстраційних номерів автомобілів і полягати в наступному:

- 1) виявлення і виділення горизонтального фрагмента зображення, що імовірно містить реєстраційний номер автомобіля.
- 2) виявлення і виділення з горизонтального фрагмента зображення реєстраційного номеру автомобіля.
- 3) виявлення і виділення елементів реєстраційного номера автомобіля із фрагменту зображення що його містить.
- 4) розпізнавання елементів реєстраційного номера автомобіля.
- 5) корекція елементів реєстраційного номера автомобіля відповідно до його типу.

З огляду на досить велику інформативність одержуваних з відеокамер фотографій автомобілів, необхідно обмежити зону виділення і подальшого розпізнавання їх реєстраційних номерів до аналізованої зони. Для цього необхідно визначити координати області, в межах якої розташовуються реєстраційні номери автомобілів на всіх фотографіях з відеокамери.

Далі необхідно з аналізованої області зображення виділити горизонтальний фрагмент, що імовірно містить автомобільний номер. Це можна зробити, наприклад, наступним чином: обчислити для всіх горизонтальних піксельних ліній їх колірну інтенсивність, тобто визначити колірну інтенсивність вертикальної проекції зображення. Для цієї мети можна скористатися наступним виразом:

$$p_y = \sum_{i=0}^{w-1} f(i, y),$$

де  $w$  – ширина зображення в пікселях,  $f(i, y)$  – колір пікселя з координатами  $(i, y)$ .

Максимуми (піки) колірної інтенсивності вертикальної проекції зображення найімовірніше свідчатимуть про наявність у відповідному фрагменті зображення реєстраційного номера автомобіля. Причина цього полягає в тому, що білий колір його фону має максимальну колірну насиченість. Однак знайти піки колірної інтенсивності

мало, необхідно визначити і їх протяжність, яка визначається передбачуваною висотою номерний пластини автомобіля. Висота пластини стандартизована і це дає підставу відсікти зайві вертикальні піки.

Реалізація наступного етапу – виявлення і виділення з горизонтального фрагмента зображення автомобільного номера може бути здійснено аналогічно попередньому. Виходячи з відомої ширини символу реєстраційного номера автомобіля, можна, як і в попередньому етапі, за допомогою обчислення колірної інтенсивності його горизонтальної проекції, виділити в ньому розташування складових його символів.

Безпосередньо перед етапом виділення у фрагменті зображення символів реєстраційного номера автомобіля може виникнути необхідність в його повороті. Для повороту області зображення використовується алгоритм, заснований на афінному перетворенні координат. Для зменшення спотворень зображення при повороті, пов'язаних з його дискретним характером, може бути застосований метод, заснований на білінійній інтерполяції по найближчих чотирьох пікселів.

#### **2.4.2 Найпоширеніші методи розпізнавання**

Виділяють чотири групи методів розпізнавання:

- нейронні мережі. Для початку вибирається вид нейронної мережі та відбувається налаштування коефіцієнтів. Далі на вхід НМ подається об'єкт для розпізнавання, а група рецепторів мережі відповідає за прийом своєї характеристичної властивості [23];

- структурні та синтаксичні методи. Вони полягають у тому що спочатку відбувається розбір об'єкта на елементи, потім будується правило на основі входження окремих елементів та їх послідовностей;

- статистичні методи. В основі даних методів лежить уявлення про клас розпізнаваних об'єктів як про реалізацію деякої випадкової величини. Цю випадкову величину з визначеними статистичними характеристиками називають статистичною моделлю класу розпізнаваних об'єктів;

- порівняння із зразком. Спершу застосовується геометрична нормалізація і вимірюється відстань до прототипу (метрика). І якщо ми наприклад маємо відскановане зображення символу, а також колекцію зображень усіх букв абетки (зразків), і перед нами стає завдання визначити, якій саме літері алфавіту відповідає зображення символу, то

можна вважати за розв'язок масштабування відсканованого зображення до розмірів зразків і вибір того, відстань до якого мінімальна.

## 2.5 Нормалізація та сегментація

Розглянемо два головних етапи процедури розпізнавання – нормалізацію та сегментацію.

Під нормалізацією у даній роботі розуміється поворот зображення автомобільного номера в площині зображення так, щоб рядок символів розташовувалася горизонтально. Для вирішення цього завдання використовується глобальний аналіз зображення за допомогою перетворення Хафа.

Сегментація символів ґрунтується на використанні моделей розташування символів на номері. В ході зіставлення різних моделей з реальним зображенням визначається модель, що має найкращу відповідність, параметри якої використовуються для отримання координат символів. Такий підхід дозволив також визначати тип номера і приналежність символу до букв або цифрах, що полегшує їх подальше розпізнавання.

Запропонований алгоритм нормалізації і сегментації символів дозволяє використовувати його в системах розпізнавання автомобільних номерів.

Нормалізація зображення номерного знаку проводиться у два етапи. На першому етапі визначається кут повороту номера в площині зображення. На другому - виконується алгоритм отримання нормалізованого зображення номера з вихідного зображення з урахуванням кута його повороту (рис. 2.3).

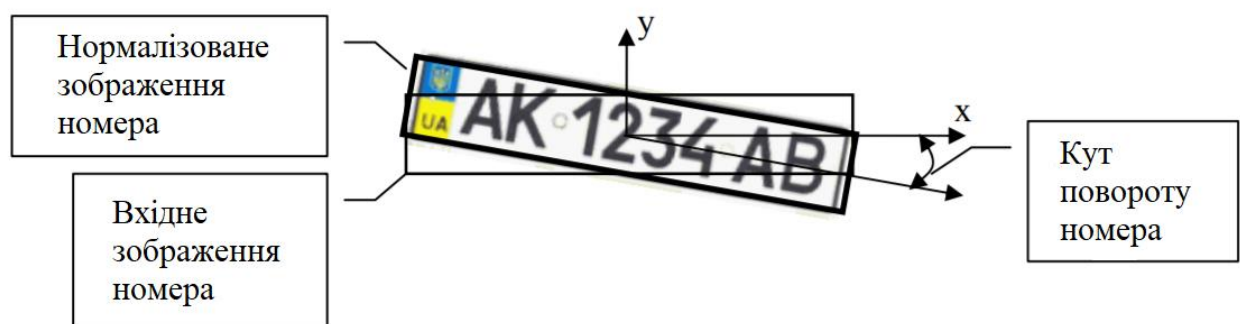


Рисунок 2.3 – Схема отримання нормалізованого зображення номера

Для повороту області зображення, що відповідає нормалізованому номеру, використовується алгоритм, заснований на відповідному афінному перетворенні

координат. Для зменшення спотворень зображення при повороті, пов'язаних з його дискретним характером, використовується метод, заснований на білінійній інтерполяції по найближчих чотирьох пікселях.

Визначення кута повороту зображення номерного знаку виконується з використанням декількох етапів обробки та аналізу зображень.

На першому етапі виконується операція підкреслення меж на зображенні на основі лінійного оператора Собеля для горизонтальних меж, має маску згортки:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

Наведений оператор більш чутливий до напрямів меж, близьким до горизонтального, тому дозволяє добре виділити на зображенні верхню і нижню частину номерного знаку, як показано на рисунках 2.4 – 2.6.



Рисунок 2.4 – Вихідне зображення зі знайденим положенням номера



Рисунок 2.5 – Вирізане зображення номерного знаку



Рисунок 2.6 – Результат підкреслення меж



На другому етапі виконується розрахунок карти щільності знайдених точок меж у просторі коефіцієнтів лінійних залежностей просторових координат згідно перетворенню Хафа. Метою другого етапу є визначення рівняння прямих, що відповідають верхній і нижній межі номерного знаку. Кожна точка карти меж, отриманої на попередньому етапі, породжує ціле сімейство прямих, що проходять через неї:

$$y_{ij} = ax_{ij} + b,$$

що в просторі лінійних коефіцієнтів також відповідає прямій:

$$b = x_{ij} - y_{ij}.$$

Наділяючи прямі в просторі коефіцієнтів вагою  $v(x_{ij}, y_{ij})$ , відповідним значенням яскравості зображення результату підкреслення меж (рис. 2.7), і проводячи їх в просторі коефіцієнтів  $a$  і  $b$  з яскравістю, рівною вазі, отримаємо зображення, подібні до наведеного на рисунку 2.7.

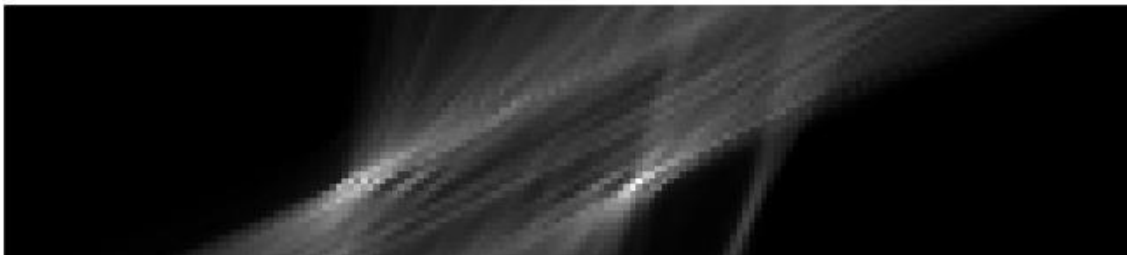


Рисунок 2.7 – Карта результатів перетворення Хафа

Вертикальний напрямок карти, наведеної на рисунку 2.7, відповідає змінам значень коефіцієнта  $a$ , горизонтальне -  $b$ . Діапазон зміни коефіцієнтів розраховується виходячи з допустимих поворотів зображення номера в площині зображення і розмірів знайденого зображення номера відповідно до формул:

$$a \in [-0,2; 0,2] \left( 0,2 \approx \text{tg}(10^\circ) \right),$$

$$b \in [-a_{\max} W; H + a_{\max} W],$$

де  $a_{\max}$  – максимальне значення  $a$ ,  $W$  – ширина зображення номера,  $H$  – висота зображення номера. Найбільш яскраві точки зображення карти результатів перетворення Хафа (рис. 2.7) відповідають найбільш яскравим прямим вихідного зображення кордонів (рис. 2.6), що дозволяє визначити рівняння цих прямих і, відповідно, кути їх нахилу. Так як шуканими є прямі, відповідні до верхньої та нижньої меж номерного знаку, то для визначення найбільш вірогідного кута повороту номера на зображенні карти результатів перетворення Хафа шукаються дві точки, що мають одне значення  $a$  і різні значення  $b$ , з найбільшою сумарною яскравістю. Значення параметра  $a$  знайдених точок повністю визначає кут нахилу зображення номерного знаку ( $a = \text{tg}(\varphi)$ ).

Для сегментації символів на автомобільному номері пропонується використовувати підхід, заснований на підгонці під реальне зображення різних моделей розташування символів на номері. Кожна з моделей відповідає певному стандарту розташування символів. У даній роботі розглядаються дві моделі розташування символів в однорядкових номерних знаках, які можна представити наступними типами: {ЛЛ ЦЦЦЦ ЛЛ} і {ЦЦЦ-ЦЦ ЛЛ}, де Л - літера, Ц - цифра. Кожну модель можна представити у вигляді зображення темних прямокутників, відповідних символів, на світлому тлі, як показано на рисунку 2.8.

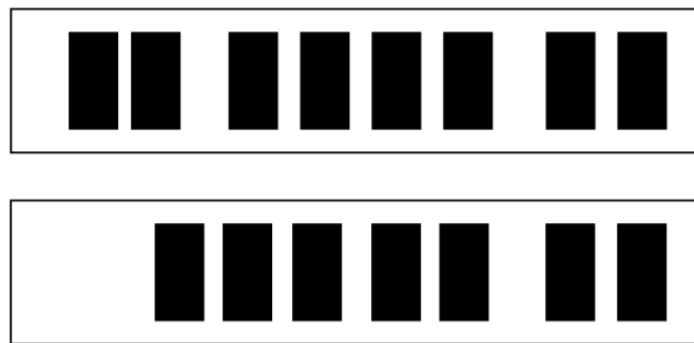


Рисунок 2.8 – Приклади використовуваних моделей розташування символів

Якщо номери мають інший тип, наприклад, білі символи на червоному фоні, або на чорному, то перед зіставленням моделі із зображенням останнім можна інвертувати по яскравості.

Як критерій відповідності моделі зображенню номера використовується величина:

$$K(x, y, W, H) = \frac{S_w - S_b}{\sigma_w},$$

де  $S_w$  – середня яскравість зображення під світлою областю,  $S_b$  – середня яскравість зображення під чорною областю,  $\sigma_w$  – середній квадратичний розкид яскравості зображення під світлою областю,  $x$  і  $y$  – координати моделі всередині зображення номера,  $W$  і  $H$  – горизонтальний і вертикальний розміри моделі. Чим більше значення критерію  $K(x, y, W, H)$  тим більше модель відповідає зображенню номера. Підгонка моделі під зображення полягає у виборі згідно з прийнятим критерієм найкращого положення і розмірів моделі. Після цього, згідно з тим ж критерієм, приймається рішення про найкращий тип моделі для поточного зображення номера. Якщо враховувати можливі похибки при нормалізації номера, на основі запропонованого критерію можна незалежно визначити більш точне положення для кожного символу окремо поблизу знайденого його положення з використанням всієї моделі.

Обчислення значення критерію  $K(x, y, W, H)$  в кожному можливому положенні моделі всередині зображення номерного знаку може бути ефективно виконано на основі використання інтегральних зображень для вихідного зображення номера і зображення квадратів значень яскравості його пікселів.

Приклади спільної роботи послідовних етапів нормалізації зображення номера і сегментації символів наведено на рисунку 2.9. У верхньому рядку рисунку 2.9 показані вихідні зображення номерів, у другому рядку – результат їх нормалізації, в нижньому рядку – результат сегментації символів на номері (кожен символ являє собою окреме зображення).



Рисунок 2.9 – Приклади роботи алгоритмів нормалізації зображення номера і сегментації символів

Отримані після сегментації зображення символів дозволяють використовувати їх для вирішення задачі розпізнавання. Запропонований алгоритм сегментації дозволяє також визначити тип номерного знаку, а отже, з'ясувати, чи є кожен із символів буквою або цифрою, що дозволить полегшити аналіз зображень на наступному етапі

розпізнавання. Запропоновані досить прості моделі дозволяють виконувати операцію сегментації з високою ефективністю за якістю і швидкості, а також дозволяють в разі потреби без значних додаткових зусиль розширити число використовуваних моделей номерного знаку шляхом введення в розгляд інших можливих розташувань символів.

## 2.6 Використання нейронних мереж

Штучна нейронна мережа – це математична модель, а також її програмна реалізація, яка будується за принципом функціонування нервових систем живих об'єктів. З точки зору машинного навчання, НМ являє собою окремий метод розпізнавання образів або класифікації.

Найпростіша модель нейронної мережі зображена на рисунку 2.10.

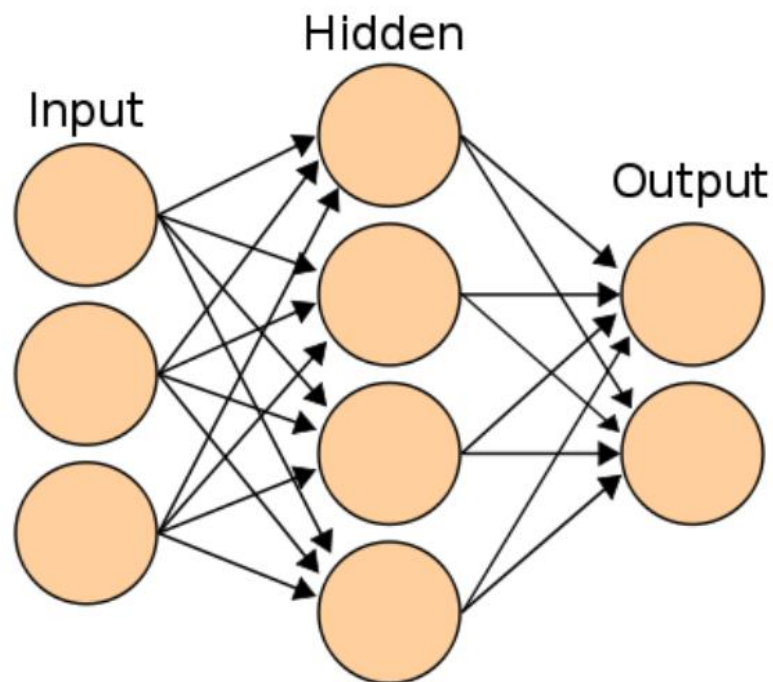


Рисунок 2.10 – Приклад найпростішої нейронної мережі

Нейронні мережі навчаються – це їхня відмінність від більшості інших алгоритмів. Технічно, навчання – це процес знаходження коефіцієнтів зв'язків між нейронами. Завдяки узагальненню, НМ може зробити висновок про об'єкт, який не був присутній в навчальній вибірці. Крім цього, проблема шуму також не є суттєвою. Нейронні мережі ідеально підходять для задачі розпізнавання автомобільних номерів.

На даний момент побудова НМ не є чимось надзвичайно складним. Існує величезна кількість бібліотек, що спрощують ті або інші аспекти створення. Наприклад, для мови Python існує бібліотека keras, за допомогою якої за кількох команд можна створити на пристрої згорткову нейронну мережу. Також у задачах обробки зображень, зв'язаних з сегментацією, кластеризацією, стисненням, виділенням ознак тощо, широкого поширення набули штучні нейронні мережі з латеральними зв'язками [3].

Одна з основних переваг НМ полягає в можливості паралельного функціонування її елементів, що істотно підвищує ефективність вирішення завдання. Навчання НМ спрощує вибір ключових ознак, їх вагових коефіцієнтів і зв'язків між ними.

При використанні нейронних мереж для розпізнавання об'єктів, у якості образів можуть виступати різні за своєю природою об'єкти: символи тексту, зображення, зразки звуків тощо. При навчанні мережі пропонуються різні зразки образів із зазначенням того, до якого класу вони відносяться. Зразок, як правило, представляється як вектор значень ознак. При цьому сукупність усіх ознак повинна однозначно визначати клас, до якого належить зразок. У разі, якщо ознак недостатньо, мережа може співвіднести один і той же зразок з декількома класами, що являється помилкою. Після закінчення навчання мережі їй можна пред'являти невідомі раніше образи і отримувати відповідь про належність до певного класу.

Топологія такої мережі характеризується тим, що кількість нейронів у вихідному шарі, як правило, дорівнює кількості визначених класів [2]. При цьому встановлюється відповідність між виходом нейронної мережі і класом, який він представляє. Коли мережі пред'являється якийсь образ, на одному з її виходів повинен з'явитися ознака того, що образ належить цьому класу. У той же час на інших виходах повинна бути ознака того, що образ до даного класу не належить. Якщо на двох або більше виходах є ознака приналежності до класу, вважається, що мережа «не впевнена» в своїй відповіді.

Найпоширеніші архітектури нейронних мереж для розпізнавання образів:

- багат шарові нейронні мережі;
- нейронні мережі високого порядку;
- радіально-базисні нейронні мережі;
- персептрон;
- когнітрон;
- неокогнітрон;
- згорткові нейронні мережі.

### 2.6.1 Навчання нейронних мереж

Процес навчання ІНС розглядається як налаштування архітектури і ваг зв'язків між нейронами для ефективного виконання поставлених перед НМ завдань. Існує два великих класу навчання НМ:

- клас детерменірованих методів;
- клас стохастичних методів.

В клас детерменірованих методів входять методи, в основі яких лежить ітеративна корекція параметрів мережі, в ході поточної ітерації яка ґрунтується на поточні параметри [22]. Основним детерменірованим методом і найпоширенішим методом навчання НМ сьогодні взагалі є метод зворотного поширення помилки.

До класу стохастичних методів входять методи, що змінюють параметри мережі випадковим чином і зберігають тільки ті зміни параметрів, які привели до поліпшення результатів. Стохастичні алгоритми навчання реалізуються за допомогою порівняння помилок і деякі з них пов'язані з проблемою «пастки локального мінімуму», що розв'язується за допомогою деяких ускладнень стохастичних алгоритмів.

Розглянемо окремі методи навчання НМ більш детально.

#### 1) Навчання з вчителем.

Під час навчання НМ з вчителем кожному прикладу з навчальної вибірки відповідає вектор, що характеризує однозначний правильну відповідь, що подається відразу на вихід мережі в обхід всієї її архітектури. Після отримання власного результату мережі, алгоритм порівнює результуючий вектор з правильною відповіддю, на основі чого відбувається корекція подальшої помилки. Правило корекції за помилкою, на якому заснований метод зворотного поширення помилки, є класичним прикладом навчання з вчителем.

#### 2) Навчання без вчителя.

Навчання без вчителя реалізується природним чином в процесі навчання, коли автоматичне налаштування параметрів мережею призводить до появи однакових результатів її функціонування при досить близьких вхідних значеннях, що на практиці можна порівняти зі зниженням розмірності даних у результаті ітераційного методу головних компонент. Правило Хеба, засноване на гіпотезі про посилення зв'язків між біологічними нейронами в разі їх одночасного збудження, і методи навчання при змаганні нейронів шляхом порівняння інтенсивності їх реакції є класичними прикладами методів навчання без учителя.

### 3) Навчання з підкріпленням.

У навчанні з підкріпленням існує агент (agent), що взаємодіє з навколишнім середовищем (environment), та виконує дії (actions). Навколишнє середовище дає нагороду (reward) за ці дії, а агент продовжує їх робити. Такий спосіб застосовується тоді, коли ми можемо ґрунтуючись на результатах отриманих від НМ, дати їй оцінку.

При навчанні з підкріпленням, на відміну від навчання з учителем, не надаються вірні пари "вхідні дані – відповідь", а прийняття субоптимальних рішень не обмежується явно. Навчання з підкріпленням намагається знайти компроміс між дослідженням невивчених областей і застосуванням наявних знань.

### 4) Метод зворотного поширення помилки (Backprop).

Метод є класичним методом навчання з учителем, заснований на правилі корекції за помилкою і був розроблений як метод навчання багатошарового перцептрона. Основна ідея методу полягає в поширенні сигналів помилки після її обчислення на виході мережі в напрямку, зворотному прямому поширенню сигналів під час звичайного обчислювального процесу, від виходів мережі до її входів [23]. При зворотному проході синаптичні ваги налаштовуються з метою мінімізації помилки

Фактично, реалізується стохастический градієнтний спуск, відбувається рух в багатовимірному просторі вагів до мінімуму помилки в сторону, протилежну градієнту. Слід зазначити, що градієнтний алгоритм має низьку швидкість сходимості та не враховує неоднорідності архітектури мережі [4].

В процесі навчання циклічно вирішуються однокритеріальні завдання оптимізації. Для можливості реалізації методу передавальна функція нейронів повинна бути диференційована. Оскільки метод є модифікацією класичного методу градієнтного спуску, то може розглядатися як градієнтний спуск по поверхні помилки.

Незважаючи на широке успішне застосування методу, він не позбавлений ряду недоліків, таких як вкрай низька швидкість навчання і принципова невдача навчання мережі (наприклад, нескінченна навчання), що можлива по ряду причин, серед яких параліч мережі, проблема локальних мінімумів і проблема розміру кроку. Ряд цих недоліків усувається різними модифікаціями методу, наприклад, модифікація RProp (Resilient Propagation) значно прискорює процес навчання за рахунок обчислення тільки знаків часткових похідних для підстроювання вагових коефіцієнтів, замість обчислення повних похідних. RProp використовує так званий метод навчання по епохах. Також існують методи QuickProp, метод сполучених градієнтів, метод Левенберга-Маркара, що прискорюють процес навчання.

### 5) Генетичні алгоритми.

Генетичні алгоритми є різновидом еволюційних обчислень, та розвиваються як евристичні алгоритми пошуку. Подібні методи використовуються для вирішення задач моделювання та оптимізації шляхом випадкового підбору, алгоритмічно реалізуючи базові принципи теорії еволюції. Генетичні алгоритми можна використовувати для навчання нейронних мереж в рідкісних випадках, де складно або неможливо сформулювати завдання у вигляді, придатному для значно більш швидких алгоритмів локальної оптимізації, або у випадках багатоекстремальної оптимізації або оптимізації недиференційованої функції.

### 2.6.2 Багат шарові нейронні мережі

Багат шарові нейронні мережі (БНМ) складаються з послідовно з'єднаних шарів, нейрон кожного з яких своїми входами пов'язаний з усіма нейронами попереднього шару, а виходами - наступного. Для активації таких нейронів служать різновиди лінійних, порогових та сігмоїдних функцій [27]. На рисунку 2.11 представлена архітектура багат шарової нейронної мережі для розпізнавання зображень. Нейрон з максимальною активністю вказує на приналежність до розпізнаного класу.

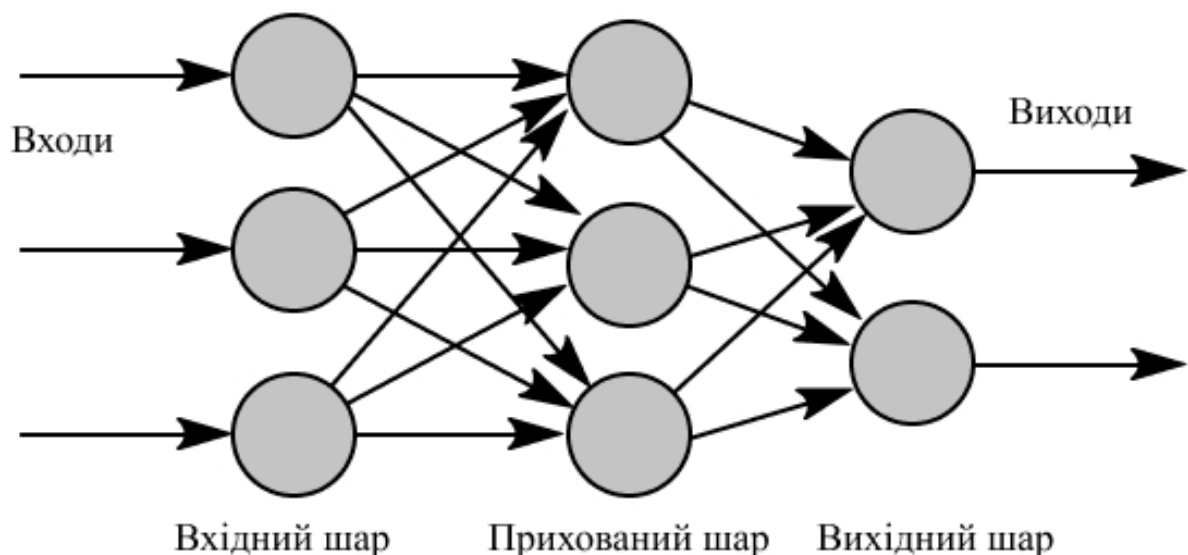


Рисунок 2.11 – Архітектура багат шарової нейронної мережі

НМ з одним вирішальним шаром здатна формувати лінійні розділяючі поверхні, що значно звужує коло вирішуваних завдань, зокрема, така мережа не зможе вирішити



завдання типу "виключне або". НМ з нелінійною функцією активації і двома вирішальними шарами дозволяє формувати будь-які опуклі області в просторі рішень, а з трьома вирішальними шарами - області будь-якої складності, в тому числі і неопуклого. Навчання БНМ здійснюється за допомогою алгоритму зворотного поширення помилки.

Оскільки еталонні значення виходів відомі, такий алгоритм відноситься до класу методів навчання з учителем. Що стосується вилучення ключових ознак, коли відбувається навчання мережі реконструкції поданого на вхід зображення, на прихованих нейронах мережі формується стислий уявлення такого зображення, що може бути віднесено до класу методів самонавчання.

Ініціалізація БНМ перед початком навчання проводиться випадковим вибором вагових коефіцієнтів. Тому дві різні навчені НМ, що забезпечують однакові значення помилки, часто можуть бути представлені різними розділяючими поверхнями, що не зводяться одна до одної. Узагальнене, отримане таким методом рішення точніше і надійніше, ніж рішення єдиною нейронною мережею.

### 2.6.3 Нейронні мережі високого порядку

Нейронні мережі високого порядку (НМВП) відрізняються від БНМ наявністю одного шару, на входи нейронів надходять також сукупності сигналів, які можуть розглядатися як терми високого порядку, що є добутком двох або більше компонентів вхідного вектору. Наприклад, для мереж другого порядку такий вектор забезпечує функціонування відповідно до виразу:

$$S = \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j - T.$$

Такі мережі можуть формувати складні розділяючі поверхні, додавши компоненти вхідного вектору в добуток, отримаємо клас поліноміальних розділяючих поверхонь. НМ також можна навчати за методом зворотного поширення. Використання БНМ в загальному випадку ефективніше, але існує ряд застосунків, в яких мережі високого порядку краще.

Особливість НМВП полягає в тому, що для розпізнавання певного класу досить пред'явити узагальнений образ зображення без варіацій масштабів і поворотів. Після

навчання мережа буде розпізнавати відомі класи інваріантної до масштабу і поворотів зображення. Така мережа не є повнозв'язною, вона характеризується високою швидкістю навчання і швидкодією. Точність класифікації такою мережею зображень, що розрізняються масштабом і кутом повороту вище в порівнянні з БНМ.

#### 2.6.4 Радіально-базисні нейронні мережі

Радіально-базисні нейронні мережі (РБНМ) складаються з двох шарів. Перший шар описується радіально-базисною активаційною функцією:

$$y = \exp\left(\frac{-S^2}{2\sigma^2}\right),$$

де  $\sigma$  – середньоквадратичне відхилення, що визначає розмір кластера,  $S$  – відстань між вектором вхідних сигналів і сформованим вектором вагових коефіцієнтів  $W$ :

$$S^2 = |X - W|^2 = \sum_i (x_i - w_i)^2.$$

Значення  $S$  визначає відстань до центру кластера від вихідного зображення на вході конкретного нейрона. Другий (прихований) шар являє собою набір кластерів в просторі образів і реалізує перший етап кластеризації вхідного образу – значення активаційної функції кожного нейрона швидко зменшується з віддаленням від центру кластера. Наступний шар нейронів може бути описаний лінійною активаційною функцією, він реалізує другий етап кластеризації – розподіляє кластери по класах.

РБНМ дозволяють будувати погано поділяючихся областей та апроксимувати багатомірні функції. У порівнянні з багатошаровою нейронною мережею РБНМ навчається на порядок швидше, проте володіє набагато гіршою екстраполюючою здатністю, тобто не спроможна розпізнавати образи, що значно відрізняються від образів-еталонів. Розмірність РБНМ більше, ніж БНМ, що призначені для вирішення аналогічних завдань, тому ефективність РБНМ зменшується з ростом розмірності вхідних даних.

Навчання такої мережі відбувається в два етапи: на першому – без вчителя: перший шар виділяє компактно розташовані групи кластерів, при цьому відбувається корекція

центрів кластерів. На другому етапі другий шар вчиться розподіляти за класами вхідні образи, пропущені через перший шар. Якщо відомі еталонні значення виходів, навчання забезпечується матричними методами або алгоритмом зворотного поширення помилки. Розглянуті типи нейронних мереж – БНМ і РБНМ – дозволяють врахувати топологію простору зображення. Їх принципи роботи ґрунтуються на розбитті зображення на локальні області та ієрархічному зіставленні як їх взаємного розташування, так і змісту. Такі мережі найбільш перспективні для розпізнавання зображень.

### 2.6.5 Персептрон

Елементарний персептрон організовується на основі сенсорних даних на вході –  $S$ -елементів, асоціативних елементів –  $A$ -елементів, і реагуючих елементів на виході –  $R$ -елементів. Набір  $S$ -елементів, що пов'язаний з  $A$ -елементом утворює асоціацію, і  $A$ -елемент активізується після того як досягнуто певне число сигналів від  $S$ -елементів.  $A$ -елемент передає зважений сигнал на сумуючий  $R$ -елемент, і в залежності від того, чи перевищує зважена сума деякий поріг,  $R$ -елемент видає результат роботи персептрону (рис. 2.12).

Багат шаровий персептрон організовується з додатковими прихованими шарами  $A$ -елементів, розташованими між  $S$ -елементами і  $R$ -елементами. Принципова складність завдань, що вирішуються багат шаровим персептроном, є найвищою для класу персептронів [26].

Навчання елементарного і багат шарового персептрона полягає в зміні вагових коефіцієнтів зв'язків  $A - R$ . Персептрон здатний працювати в режимі розпізнавання або узагальнення.

В одношаровому персептроні вхідні елементи безпосередньо пов'язані з вихідними за допомогою системи вагів, зв'язки  $S - A$  організовані за принципом однозначної відповідності. Одношаровий персептрон є окремим випадком класичного елементарного персептрона, найпростішої мережею прямого поширення – лінійним класифікатором, і має безліч принципових обмежень, таких як неможливість реалізації функції XOR.

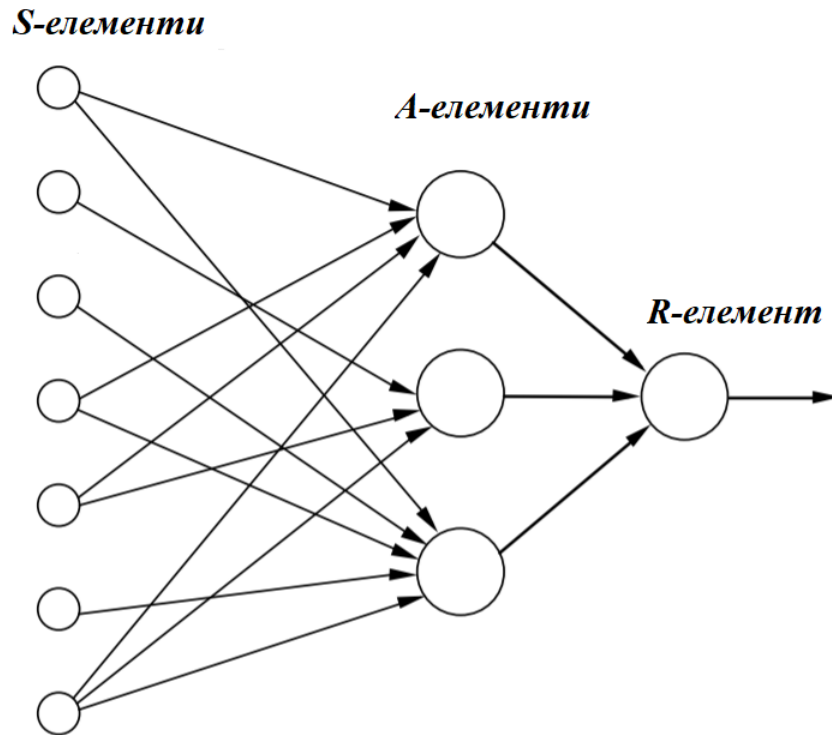


Рисунок 2.12 – Архітектура персеPTRону

### 2.6.6 Когнітрон

В основу функціонування когнітрону покладена модель зорової кори мозку. Кожен шар мозку реалізує різні рівні узагальнення – вхідний шар чутливий до простих образів, таких як лінії різної орієнтації в площині, в той час як інші шари дозволяють формувати більш складні, абстрактні і незалежні від положення образи.

Кожен нейрон когнітрону зв'язаний тільки з локальною областю попереднього шару, причому області можуть взаємно перекриватися. Шарів в когнітроні зазвичай більше, ніж в мережах інших типів, таким чином досягається ієрархічна організація. Вищі шари когнітрону реагують на виділені абстрактні признаки, тому на розпізнавання в меншій мірі впливають зміщення і спотворення вхідного образу.

Когнітрон складається з ієрархічно пов'язаних шарів нейронів двох типів – гальмуючих і збудливих. Стан збудження кожного нейрона визначається співвідношенням його гальмуючих і збуджуючих входів. Синаптичні зв'язки йдуть від нейронів одного шару (далі шару 1) до наступного (шару 2). Відносно даного синаптичного зв'язку відповідний нейрон шару 1 є пресинаптичним, а нейрон другого шару - постсинаптичним. Постсинаптичні нейрони зв'язані не з усіма нейронами 1-го шару, а лише з тими, які

належать їх локальній області зв'язків (рис. 2.13). Области зв'язків близьких один до одного постсинаптичних нейронів перекриваються, тому активність даного пресинаптичного нейрона буде позначатися на усі більш поширючі області постсинаптичних нейронів наступних шарів ієрархії.

Когнітрон конструюється у вигляді шарів нейронів, з'єднаних синапсами. Пресинаптичний нейрон в одному шарі пов'язаний з постсинаптичним нейроном в наступному шарі. Є два типи нейронів: збуджуючі вузли, які прагнуть викликати збудження постсинаптичного вузла, і гальмуючі вузли, які гальмують це збудження. Збудження нейрона визначається зваженою сумою його збуджуючих і гальмуючих входів, однак в дійсності механізм є більш складним, ніж просте підсумовування.

Дана нейронна мережа одночасно є як моделлю процесів сприйняття на мікрорівні, так і обчислювальної системою, що застосовується для технічних завдань розпізнавання образів.

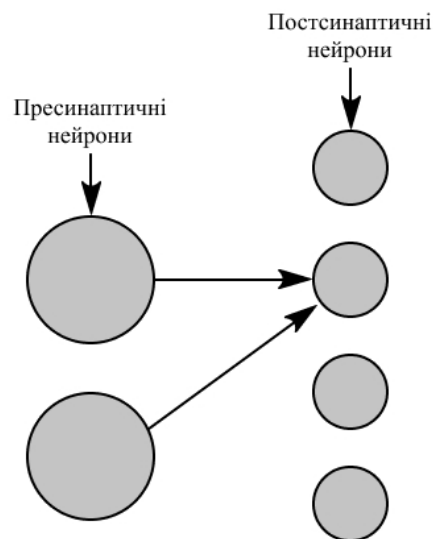


Рисунок 2.13 – Пресинаптичні та постсинаптичні нейрони когнітрона

### 2.6.7 Неокогнітрон

У зоровій корі мозку людини були виявлені групи нейронів (вузли), що реагують на такі елементи, як лінії і кути певної орієнтації. На більш високому рівні вузли реагують на більш складні і абстрактні образи –окружності, трикутники і прямокутники. Зі збільшенням рівня ступінь абстракції зростає до тих пір, поки не сформуються вузли, що реагують на цифри та інші складні об'єкти. У загальному випадку вузли наступних рівнів отримують на вхід результати обробки групи низькорівневих вузлів і, отже, реагують на

більш широкую область візуального поля. Реакції високорівневих вузлів більш стійкі до спотворень вихідного образу.

Неокогнітрон більш точно, у порівнянні з когнітроном, відображає будову зорової кори і дозволяє розпізнавати образи незалежно від їх перетворень: зміщення, обертання, зміни масштабу і спотворення. Неокогнітрон може як самонавчатися, так і навчатися з учителем. На вхід неокогнітрона надходять двовимірні образи, аналогічні зображенням, що формуються сітчастою оболонкою ока, і обробляються аналогічно зоровій корі мозку людини.

Головна відмінність неокогнітрона від когнітрону – двовимірна організація локальних ділянок у вигляді ієрархічної структури, що складається з площин.

Шари складаються з простих і складних площин. Кожен нейрон простої площини пов'язаний з локальною двовимірною ділянкою площин попереднього шару, значення вагових коефіцієнтів всіх нейронів в межах однієї площини однакові, і таким чином площина реагує на певний образ, що знаходиться в ділянці зображення. Місцезнаходження активованого нейрона у простій площині визначає ділянку, в якому знайдений цей образ, незалежно від його спотворення.

Класичний неокогнітрон є потужним засобом розпізнавання зображень, проте вимагає великих, на сьогоднішній день важкодосяжних, обчислювальних витрат.

### 2.6.8 Згорткові нейронні мережі

У класичній багатосарової нейронній мережі міжшарові нейрони з'єднання є повнозв'язними, зображення представляється у вигляді  $n$ -мірного вектора, який не враховує ні двовимірної локальної організації пікселів, ні можливостей деформації образу. Архітектура згорткової НМ (рис. 2.14) дозволяє подолати ці недоліки, в ній реалізовані принципи архітектури неокогнітрона, спрощеного і доповненого алгоритмом навчання зі зворотним поширенням помилки.

У ЗНМ використовуються локальні рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів), загальні вагові коефіцієнти (забезпечують детектування номеру, що знаходяться в будь-якому фрагменті зображення) і ієрархічна організація з просторовими підвибірками (Spatial subsampling).

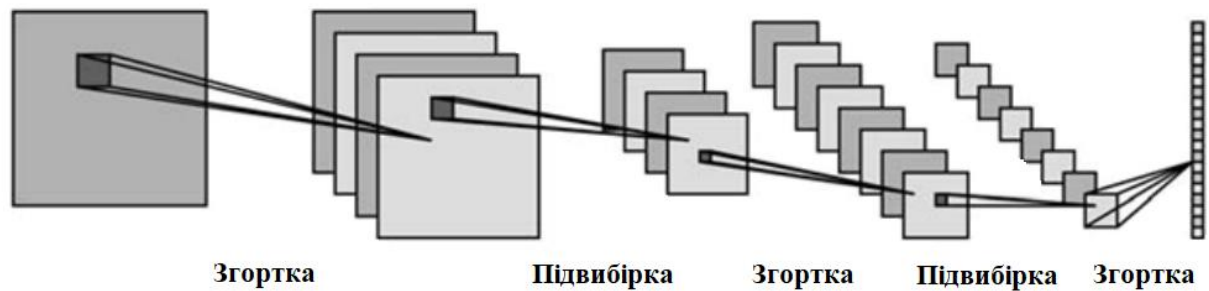


Рисунок 2.14 – Архітектура згорткової нейронної мережі

ЗНМ забезпечує часткову стійкість до змін масштабу, зсувів, поворотам, зміні ракурсу і іншим спотворень. Архітектура ЗНМ багат шарова. Шари поділяються на два типи: згорткові (Convolutional) і підвибіркові (Subsampling), що чергуються один з одним. У кожному шарі є набір з декількох площин, причому нейрони однієї площини мають однакові вагові коефіцієнти, що надходять до всіх локальних ділянок попереднього шару (як в зоровій корі людини), зображення попереднього шару "сканується" невеликим вікном і "зважується" набором вагових коефіцієнтів, а результат відображається на відповідний нейрон поточного шару. Таким чином, площини називаються картами характеристик (feature maps), кожна з них виділяє "свої" ділянки зображення в будь-якому місці попереднього шару. Наступний за згортковим, підвибірковий шар зменшує масштаб площин за рахунок локального усереднення значень реакції шару на виходах нейронів, таким чином досягається ієрархічна організація ЗНМ. Наступні шари витягують найбільш загальні властивості, менше залежать від спотворень зображення [24].

Навчається ЗНМ стандартним методом зворотного поширення помилки. Порівняння БНМ і ЗНМ показує істотні переваги останньої як за швидкістю, так і по надійності класифікації. Корисною властивістю ЗНМ є і те, що характеристики, які формуються на виходах верхніх шарів структури, можуть застосовуватися для класифікації за методом найближчого сусіда (наприклад, при обчисленні евклидової відстані), причому ЗНМ може успішно використовувати такі характеристики і для образів, відсутніх в навчальному наборі. Для ЗНМ характерні висока швидкість навчання і швидкодія.

На даний момент згорткова нейронна мережа та її модифікації вважаються кращими по точності і швидкості алгоритмами знаходження об'єктів на зображенні. Починаючи з 2012 року, нейромережеві методи займають перші місця на відомому міжнародному конкурсі з розпізнавання образів ImageNet.

## 3 РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

### 3.1 Обґрунтування вибору середовища програмної реалізації

В даній роботі для написання програми було обрано мову програмування Java, середовище розробки IntelliJ IDEA, та фреймворк збірки проектів Maven, що полегшив процес встановлення зв'язку з нейронною мережею.

#### 3.1.1 Java - технології

Java - строго типізований об'єктно-орієнтована мова програмування, розроблений компанією Sun Microsystems (в подальшому придбаній компанією Oracle). Розробка ведеться співтовариством, організованим через Java Community Process, мова і основні реалізуючі його технології поширюються за ліцензією GPL. Права на торговельну марку належать корпорації Oracle.

Дата офіційного випуску – 23 травня 1995 року. Спочатку мова називалася Oak («Дуб»), розроблявся Джеймсом Гослінгом для програмування побутових електронних пристроїв. Через те, що мова з такою назвою вже існувала, незабаром Oak був перейменований в Java. Названий на честь марки кави Java, яка, в свою чергу, отримала найменування однойменного острова (Ява), тому на офіційній емблемі мови зображена чашка з гарячою кавою. Існує й інша версія походження назви мови, пов'язана з алюзією на каву-машину як приклад побутового пристрою, для програмування якого спочатку мова створювався.

В результаті роботи проекту світ побачив принципово новий пристрій, який випередив свій час більш ніж на 10 років. Пристрій одержав назву Star7, але через велику вартість в 50 доларів не зміг зробити переворот в світі технологій і поступово був забутий.

Star7 не користувався популярністю на відміну від мови програмування Java і його оточення. Наступним етапом життя мови стала розробка інтерактивного телебачення. У 1994 році стало очевидним, що інтерактивне телебачення було помилкою.

З середини 1990-х років мова стала широко використовуватися для написання клієнтських застосунків і серверного програмного забезпечення. Тоді ж певного поширення набула технологія Java-апплетів – графічних Java-застосунків, вбудованих в веб-сторінки, однак з розвитком можливостей динамічних веб-сторінок в 2000-і роки технологія стала застосовуватися рідко.



Програми Java зазвичай транслюються в спеціальний байт-код, тому вони можуть працювати на будь-якій комп'ютерній архітектурі за допомогою віртуальної Java-машини (JVM). JVM (Java Virtual Machine) – програма, що обробляє байтовий код і передає інструкції обладнанню як інтерпретатор.

Перевагою подібного способу виконання програм є повна незалежність байт-коду від операційної системи і обладнання, що дозволяє виконувати Java-застосунки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки, в рамках якої виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером), викликають негайне переривання.

На 2019 рік Java – один з найпопулярніших мов програмування. Java-технології мають багато особливостей, що відрізняють їх від інших технологій розробки програмного забезпечення. Головні з особливостей наведені нижче.

**Переносимість.** Програми, написані на мові Java, після одноразової трансляції в байт-код можуть бути виконані на будь-якій платформі, для якої реалізована віртуальна Java-машина. Найбільш ефективно можливості реального комп'ютера може використовувати тільки програма, написана з використанням «рідного» машинного коду.

**Безпека.** Функціонування програми повністю визначається (і обмежується) віртуальною Java-машиною. Відсутні покажчики та інші механізми для безпосередньої роботи з фізичної пам'яттю і іншим апаратним забезпеченням комп'ютера. Хоча й додаткові обмеження знижують можливість написання ефективних Java-програм.

**Надійність.** У мові Java відсутні механізми, що потенційно призводять до помилок: арифметика покажчиків, неявне перетворення типів з втратою точності. Присутній суворий контроль типів, обов'язковий контроль виняткових ситуацій. Велика кількість логічних помилок виявляється ще на етапі компіляції. Однак, наявність додаткових перевірок знижує ефективність виконання Java-програм.

**Збирач сміття.** Звільнення пам'яті при роботі програми здійснюється автоматично за допомогою «збирача сміття», тому програмувати з використанням динамічно розподіленої пам'яті простіше і надійніше. Але треба пам'ятати, що при інтенсивній роботі з динамічно розподіленою пам'яттю можливі помилки через те, що «збирач сміття» не встиг звільнити невикористовувані області пам'яті.

**Стандартні бібліотеки.** Велика кількість завдань, що зустрічаються при розробці програмного забезпечення, вже вирішені в рамках стандартних бібліотек. Використання об'єктно-орієнтованого підходу дозволяє легко використовувати готові об'єкти в своїх

програмах. Для запуску програми необхідна установка JRE, що містить повний набір бібліотек, навіть якщо всі вони не використовуються у застосунку. Тож відсутність бібліотеки необхідною версії може перешкодити запуску застосунку.

Самодокументуємий код. У розпорядженні Java є механізм автоматичного генерування документації на основі коментарів, розміщених в тексті програм.

Різноманіття типів застосунків. Мовою Java можливо реалізувати абсолютно різні за способом функціонування та сфері використання програми.

### 3.1.2 Середовище розробки IntelliJ IDEA

IntelliJ IDEA – інтегроване середовище розробки програмного забезпечення для багатьох мов програмування, зокрема Java, JavaScript, Python, розроблена компанією JetBrains.

Перша версія з'явилася в січні 2001 року і швидко набула популярності як перша среда для Java з широким набором інтегрованих інструментів для рефакторингу, які дозволяли програмістам швидко реорганізувати вихідні тексти програм. Дизайн середовища орієнтований на продуктивність роботи програмістів, дозволяючи сконцентруватися на функціональних завданнях, в той час як IntelliJ IDEA бере на себе виконання рутинних операцій.

Починаючи з шостої версії продукту IntelliJ IDEA надає інтегрований інструментарій для розробки графічного інтерфейсу користувача. Серед інших можливостей, среда добре сумісна з багатьма популярними вільними інструментами розробників, такими як CVS, Subversion, Apache Ant, Maven і JUnit. У лютому 2007 року розробники IntelliJ анонсували ранню версію плагіна для підтримки програмування на мові Ruby.

Починаючи з версії 9.0, среда доступна в двох редакціях: Community Edition і Ultimate Edition. Community Edition є повністю вільною версією, доступною під ліцензією Apache 2.0, в ній реалізована повна підтримка Java SE, Kotlin, Groovy, Scala, а також інтеграція з найбільш популярними системами управління версіями. В редакції Ultimate Edition, доступною під комерційною ліцензією, реалізована підтримка Java EE, UML-діаграм, підрахунок покриття коду, а також підтримка інших систем управління версіями, мов та фреймворків.

### 3.1.3 Фреймворк Maven

Apache Maven - фреймворк для автоматизації збірки проектів на основі опису їх структури в файлах на мові POM (Project Object Model), що є підмножиною XML. Проект Maven видається співтовариством Apache Software Foundation, де формально є частиною Jakarta Project.

Назва системи є словом з мови ідиш, сенс якого можна приблизно висловити як «збирач знання».

Maven забезпечує декларативну, а не імперативну (на відміну від засобу автоматизації збирання Apache Ant) збірку проекту. У файлах опису проекту міститься його специфікація, а не окремі команди виконання. Всі завдання по обробці файлів, описані в специфікації, Maven виконує за допомогою їх обробки послідовністю вбудованих і зовнішніх плагінів.

Maven використовується для побудови і управління проектами, написаними на Java, C #, Ruby, Scala, та іншими мовами.

Серед примітних альтернатив – система автоматичного складання Gradle, побудована на принципах Apache Ant і Maven, але використовує спеціалізований DSL на Groovy замість POM-конфігурації.

Інформація для складання проекту, підтримуваного Apache Maven, міститься в XML-файлі з назвою pom.xml. При запуску Maven перевіряє, чи містить конфігураційний файл всі необхідні дані і всі дані синтаксично правильно записані.

Мінімальна конфігурація включає версію конфігураційного файлу, ім'я проекту, його автора і версію. За допомогою pom.xml конфігуруються залежності від інших проектів, індивідуальні фази процесу побудови проекту (build process), список плагінів, що реалізують порядок складання.

Великі проекти можуть бути поділені на кілька модулів, або підпроектів, кожен зі своїм власним POM. Операції над модулями можуть виконуватися через загальний корневий POM єдиною командою.

POM-файли підпроектів можуть успадковувати конфігурацію від інших файлів конфігурації. У той же час всі файли конфігурації обов'язково успадковуються від «Супер POM» файлу за замовчуванням. Супер POM забезпечує конфігурацію за замовчуванням, наприклад, стандартна структура каталогів, плагіни що використовуються за замовчуванням, прив'язка до фаз життєвого циклу та інше.

### 3.2 Програмна реалізація

Програмний застосунок у даній роботі був реалізований з використанням вже побудованої нейронної мережі Plate Recognizer. Зв'язок з нею був встановлений за допомогою Plate Recognizer API, що потребує встановлення залежностей у проекті Maven, у конфігураційному файлі pom.xml (Додаток А), та зазначення персонального з'єднувального ключа API key. Plate Recognizer API містить у собі розгорнуту навчену нейронну мережу, побудовану за принципом ЗНМ.

При завантаженні зображення зі сторони клієнта, до серверу API надсилається запит, що містить завантажене зображення. На сервері за допомогою вбудованої ЗНМ (Додаток Б) зображення оброблюється, та проходить процес розпізнавання (Додаток Б, клас OCRReader).

Після процесу розпізнавання до сторони клієнта відправляються дані у форматі JSON. Вони містять масив визначених автомобільних номерів та додаткові коефіцієнти. Далі проводиться операція розбиття JSONa на рядки з необхідною інформацією. Після отримання розпізнаного номерного знаку у вигляді символного поля проводиться операція ідентифікації для визначення регіону номера автомобілю.

На рисунку 3.1 зазначено регіони та відповідні їм серії автомобільних номерів.

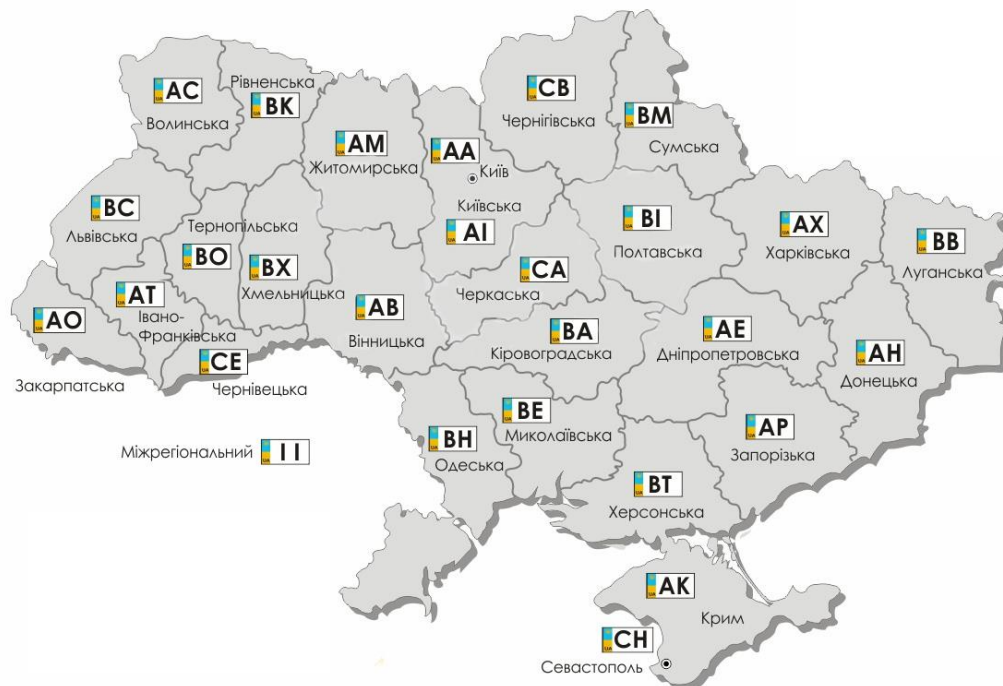


Рисунок 3.1 – Серії автомобільних номерів згідно регіону

### 3.3 Інструкція користувача

У даній роботі було розроблено застосунок для операційних системи Windows та MacOS.

Для запуску програми користувач повинен мати комп'ютер або ноутбук на базі операційних систем, що вказані вище, мати встановлену версію JVM не менше 1.8, та мати розмір оперативної пам'яті не менше 4 ГБ для успішного запуску та роботи програми

Для запуску розробленого застосунку користувачу потрібно завантажити на комп'ютер файл програми (файл з розширенням .jar) і запустити його. Вказаний файл міститься на диску або флеш-карті.

Програма в цілому і функції, які необхідно реалізувати, має на увазі, що користувач має стабільне підключення для виходу в мережу Інтернет для підтримки зв'язку з нейронною мережею.

Головна проблема, з якою стикаються користувачі в початковій версії системи - неточність розпізнання тексту, бо іноді значення цифр плутається з буквами («О» та 0, «I» та 1). Надалі буде збільшена кількість тестових зображень для кращого навчання нейронної мережі, а також розширено спектр можливих причин відмови застосунку.

Також планується офіційне оформлення логотипу та назви продукту, розміщення програми на порталі для більш зручного завантаження та інсталювання.

### 3.4 Тестування розробленої моделі

Тестування проводиться на ПК з операційною системою Windows 10, обсягом оперативної пам'яті 8,00 ГБ, та встановленою версією JVM 1.8.0\_221.

Для запуску застосунку було відкрито папку «Diploma» та запущено файл diploma.jar. Початковий екран при запуску застосунку зображений на рисунку 3.2.

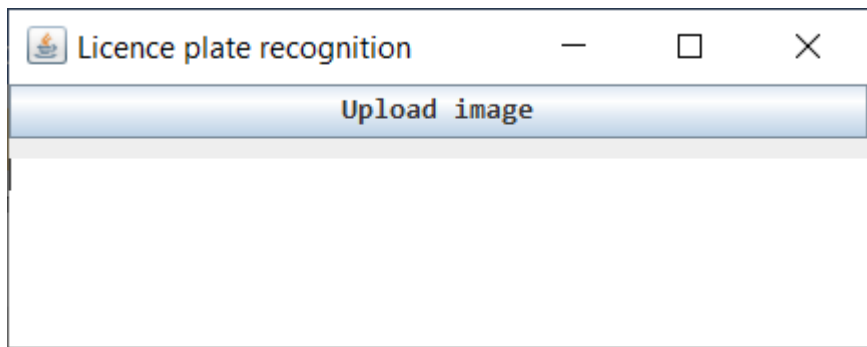


Рисунок 3.2 – Діалогове вікно провідника для вибору файлу

Після запуску програми користувачу надається можливість завантажити файл. Після натискання на кнопку «Upload image» відкривається діалогове вікно Провідника (рис. 3.3). Після вибору файлу з зображенням, автоматично запускається процес розпізнавання.

Для тестування програми були використані різноманітні варіації типів файлів, розміру та чіткості автомобільного номеру на зображеннях. Було протестовано розпізнавання на файлах типу .jpg, .bmp, .png.

Завантаження файлу іншого типу є неможливим так як у Провіднику, будуть відображені тільки файли відповідних типів.

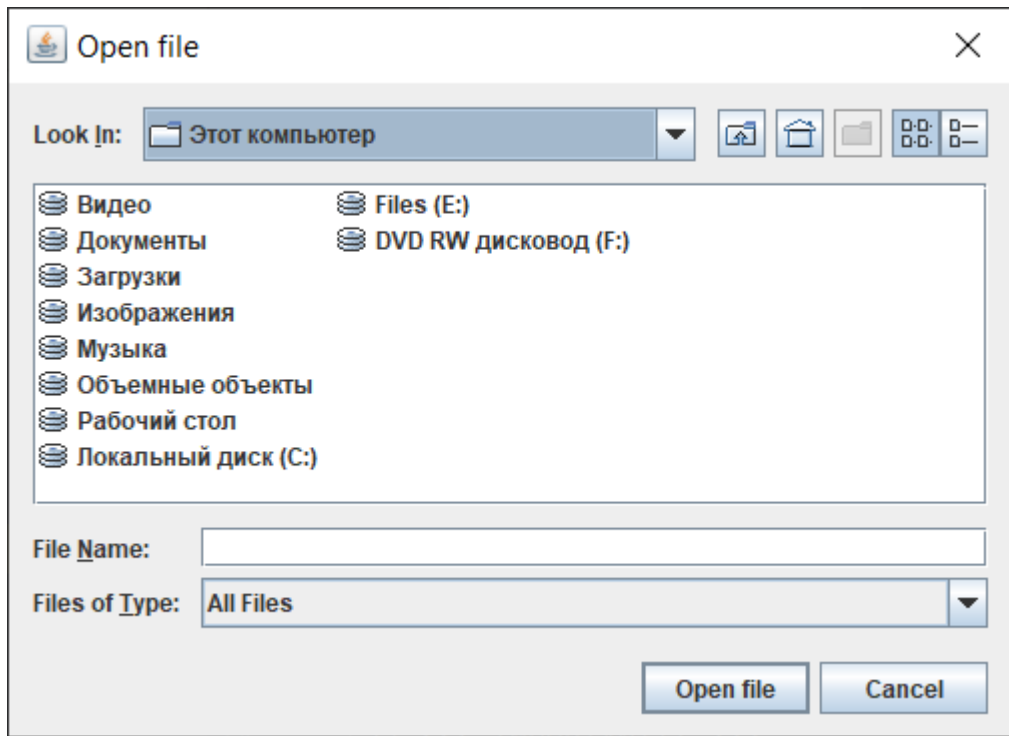


Рисунок 3.3 – Діалогове вікно провідника для вибору файлу

Якщо розпізнаний номер автомобіля не відповідає сьогоднішньому формату, замість регіону виводиться повідомлення «Region is not found!» (рис. 3.4).



Рисунок 3.4 – Застарілий формат автомобільного номера

Для визначення можливостей розпізнавання на тому чи іншому зображенні було знайдено два коефіцієнта:

DScore (Detection score) – коефіцієнт виявлення, або знаходження області, де знаходиться автомобільний номер;

RScore (Recognition score) – коефіцієнт розпізнавання автомобільного номеру на визначеній області.

Обидва коефіцієнта мають значення у рамках  $[0,1]$ . Значення DScore залежить від типу зображення, яке ми обробляємо. Чим краща якість зображення, та чим рівніше розташований автомобіль на зображенні, тим більше значення коефіцієнта DScore, що й було доведено тестуванням програми (рис. 3.5 – 3.6).

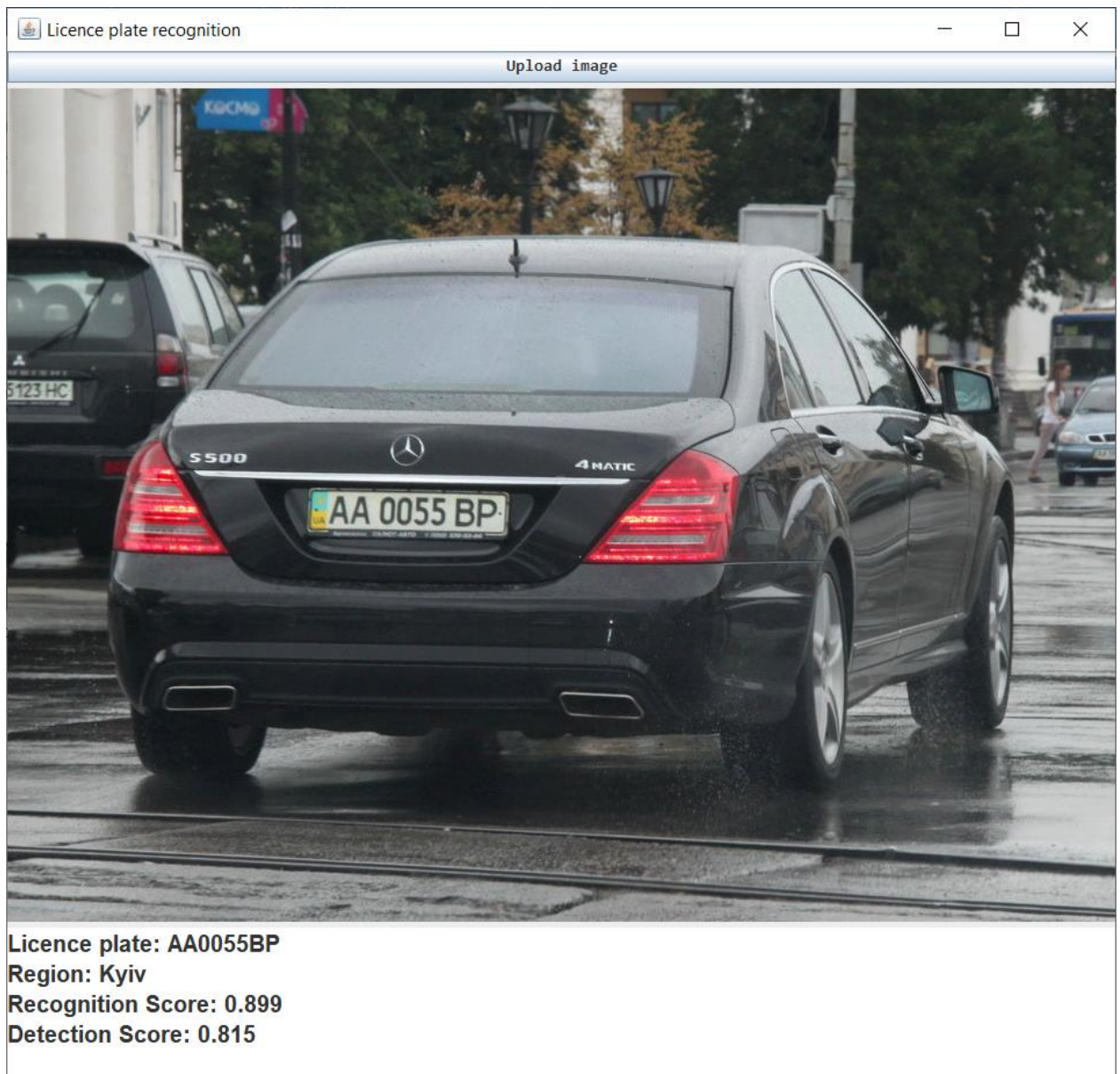


Рисунок 3.5 – Автомобільний номер на зображенні з достатньою якістю



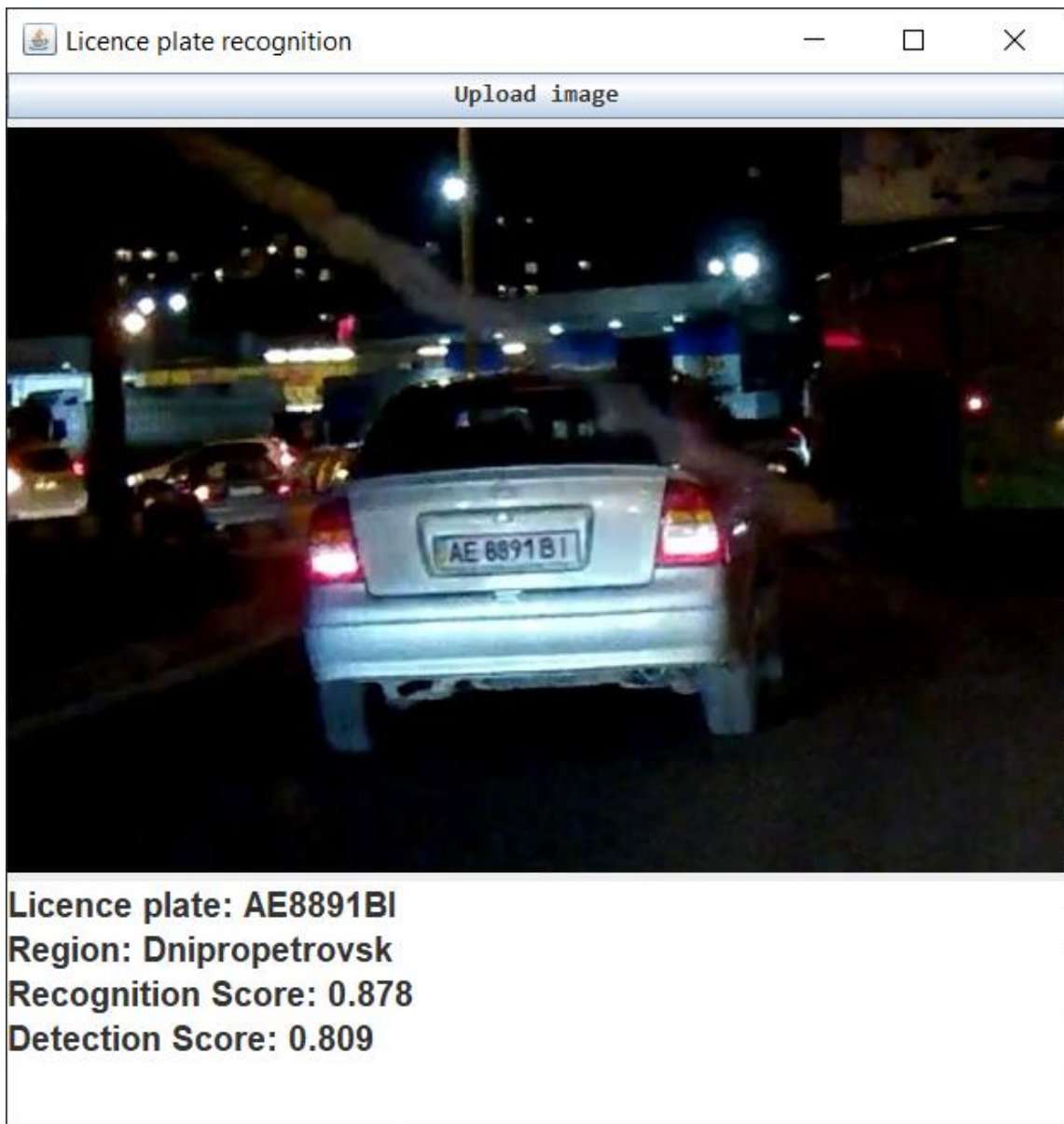


Рисунок 3.6 – Низька якість зображення

Для RScore це також має значення, але важливішим є той факт, щоб автомобільний номер на визначеній області був не спотворений (замилений, покритий брудом, розташований під нахилом). Результати даного тестування представлені на рисунках 3.7 – 3.8. Важливо також зазначити що автомобільний номер повинен бути добре освітлений та не закритий іншими об'єктами (наприклад іншим авто).

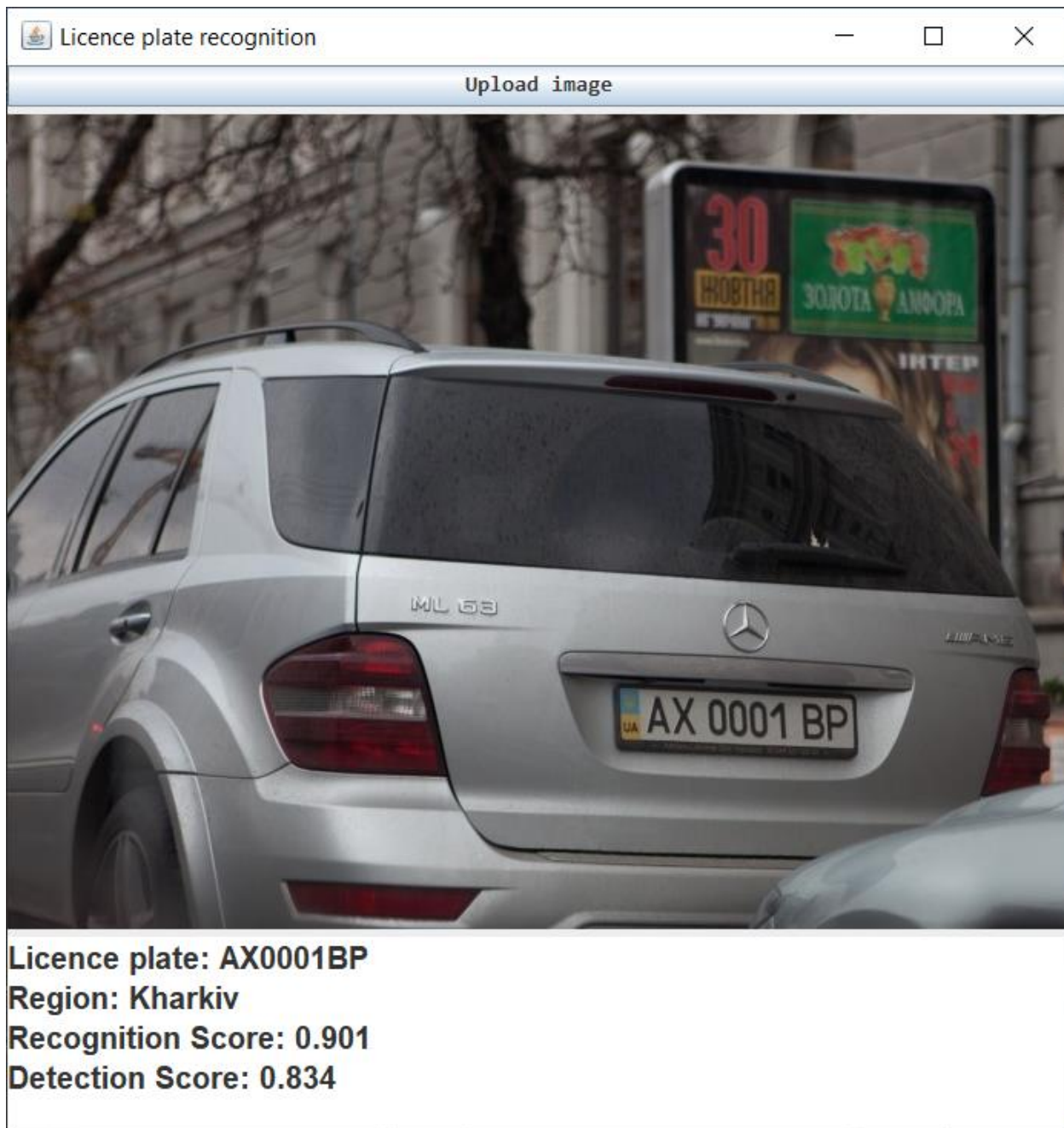


Рисунок 3.7 – Автомобільний номер без спотворень

Для даного застосунку реалізовано детектування лише регіонів України. Також це працює лише для автомобілів з номерними знаками не пізніше 2004, у яких номерний знак визначений у форматі «AA0000AA». Тож регіон автомобіля визначається по першим двом літерам у номерному знаку.

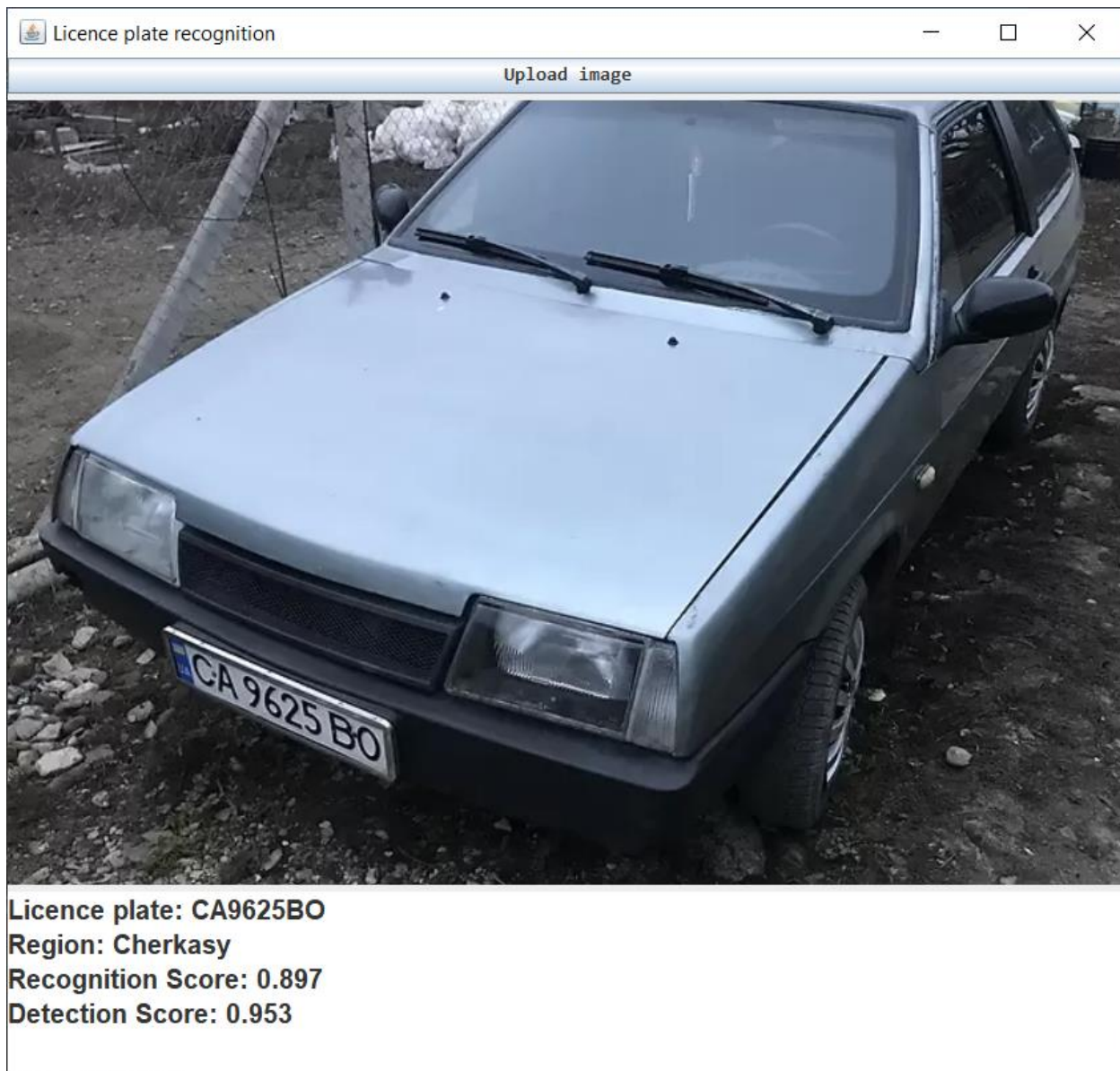


Рисунок 3.8 – Автомобільний номер під нахилом

Слід відзначити що існує проблема для спотворених зображень: іноді літера “O” сприймається як число 0 (нуль). Особливо якщо автомобіль був сфотографований збоку, тож автомобільний номер розташований не прямо (рис. 3.9). Саме тому й регіон автомобіля неможливо визначити, бо у аббревіатурі регіону не може міститися цифра.

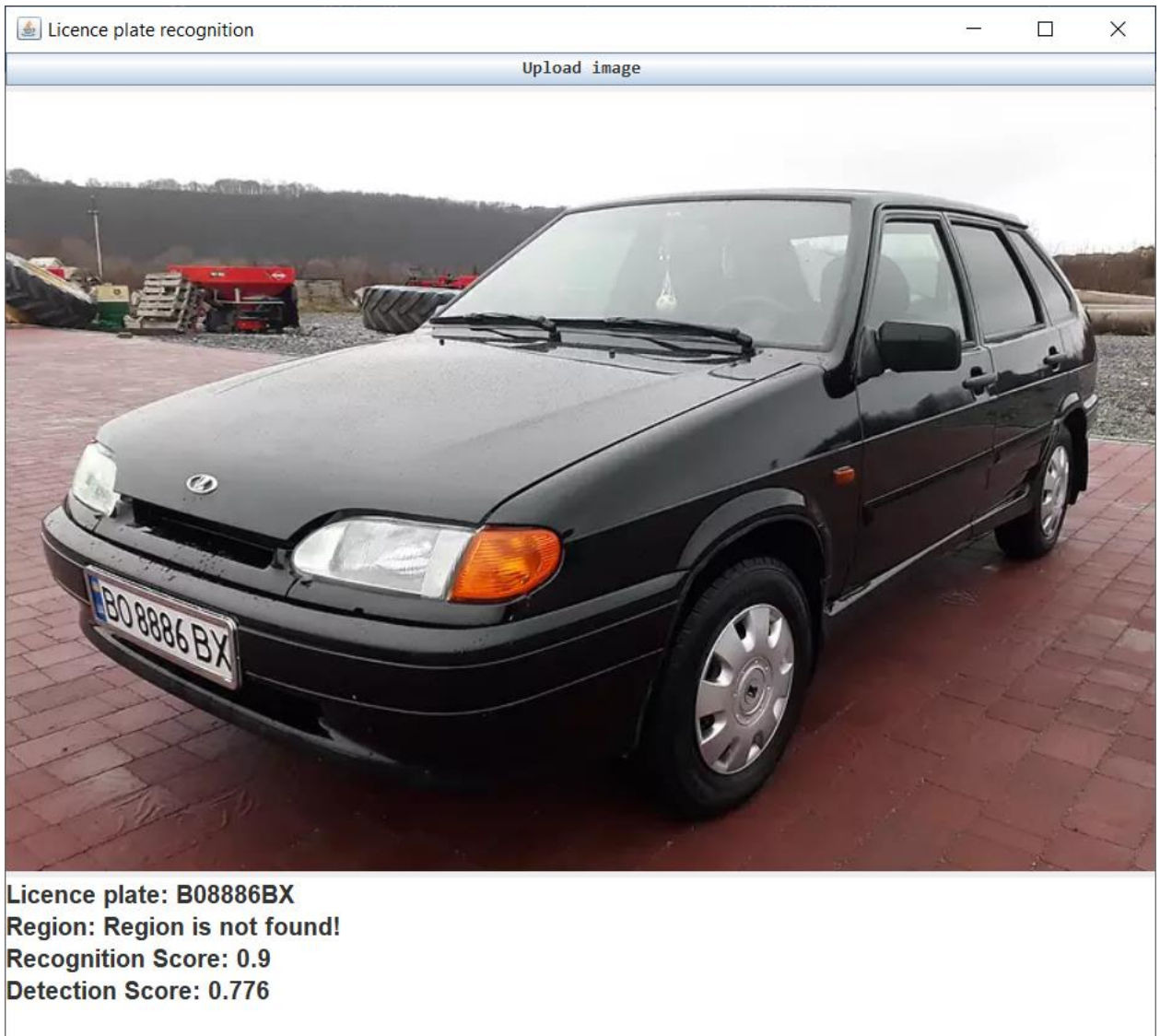


Рисунок 3.9 – На викривленому зображенні зображенні літера “О” сприймається як число 0 (нуль)

Однак є програмне вирішення. Оскільки нам відомо що літери у українських номерах присутні лише у чотирьох позиціях (дві попереду та дві у кінці), то якщо програма знаходить цифру 0 в будь-якій іншій позиції, вона автоматично замінюється на літеру “О”. Дане рішення зображене на рисунку 3.10. Аналогічна проблема існує для літер “А”, “В” та “Г”, що плутаються з цифрами 4, 8 та 1 відповідно. Ця проблема була також вирішена програмним шляхом.

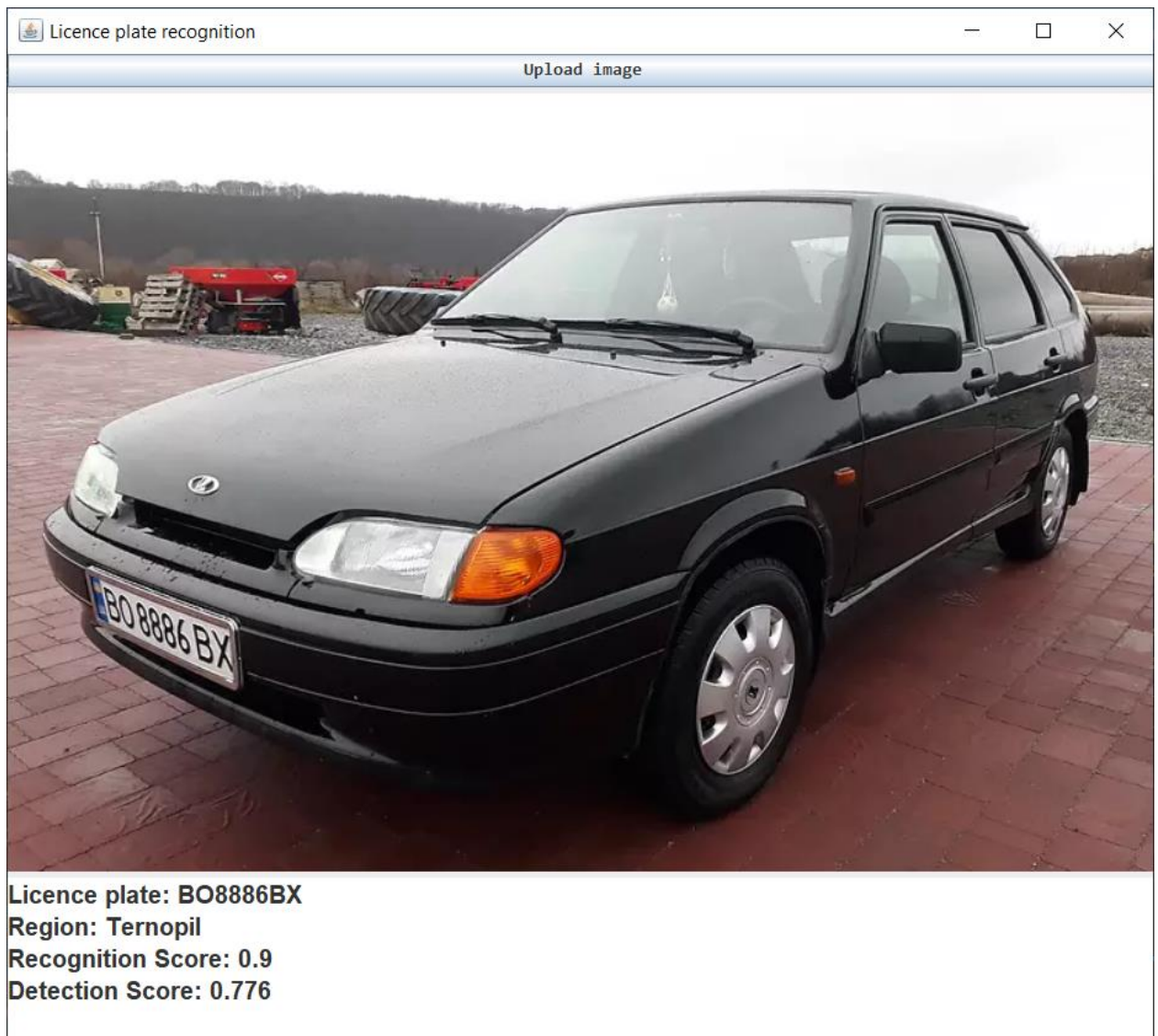



Рисунок 3.10 – Заміна числа 0 (нуль) літерою “O” для визначення регіону

Визначення декількох номерів на одному зображенні також є можливим. В цьому випадку для обох номерів проводиться пошук по регіону та визначаються коефіцієнти RScore та Dscore (рис. 3.11). На цьому фото досить чітко видно різницю у коефіцієнтах. Так для ближчого автомобілю Dscore буде нижчий тому що номер розташований під нахилом, на відміну від дальнього авто. Тоді як RScore для нього більший, тому що сам номер на визначеній області видно чіткіше ніж для другого автомобілю.

Licence plate recognition — □ ×

Upload image



Licence plate: AA0001MP  
Region: Kyiv  
Recognition Score: 0.906  
Detection Score: 0.777

Licence plate: AA0001MX  
Region: Kyiv  
Recognition Score: 0.9  
Detection Score: 0.809

Рисунок 3.11 – Розпізнавання декількох номерів на одному зображенні

## 4 ОХОРОНА ПРАЦІ

### 4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення.

Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу ІВМ РС мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до ДСН 3.3.6.042-99 [21] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до НПАОП 0.00-7.15-18 [22], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

#### **4.2 Заходи щодо техніки безпеки**

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.



Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення мережі;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує години, протягом яких людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_T}{3}}, \quad (4.1)$$

де  $U_\phi$  - номінальна фазна напруга мережі, В;

$Z_\Pi$  - повний опір петлі, створене фазними і нульовими дротами, Ом;

$Z_T$  - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [23]:  $Z_T / 3 = 0,1$  Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_\Pi = \sqrt{R_\Pi^2 + X_\Pi^2}, \quad (4.2)$$

де  $R_\Pi = R_\phi + R_0$  - сумарний активний опір фазного  $R_\phi$  і нульового  $R_0$  дротів, Ом;

$X_\Pi$  - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту  $S = 2.5$  мм, тоді згідно таблицям 5 і 6 [23], має такий опір:

$X_\Pi = 0,11$  Ом;

$R_\phi = 7,55$  Ом;

$R_0 = 7,55$  Ом.

Отже,  $R_{\Pi} = 7,55 + 7,55 = 15,1 \text{ Ом}$ .

Тоді по формулі (4.2) знаходимо повний опір петлі :

$$Z_{\Pi} = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n, \quad (4.3)$$

де  $I_n$  - номінальний струм спрацьовування плавкої вставки, А;

$k$  - коефіцієнт кратності нелінійного струму  $I_n$ , А.

Коефіцієнт кратності нелінійного струму  $I_n$  розраховується по формулі (4.4.) :

$$I_n = P / U, \quad (4.4)$$

де  $P = 220 \text{ Вт}$  - споживана потужність;

$U = 220 \text{ В}$  - робоча напруга;

$k = 3 \text{ А}$  - для плавких вставок.

Отже,  $I_n = 220 / 220 = 1 \text{ А}$ .

Підставивши значення у вираз (4.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

### 4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються ДСН 3.3.6.042-99[21], ДБН, відповідними ГОСТами і ОСТами з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до ДСН 3.3.6.042-99 [21] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу(категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й установленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в ДБН (30 м<sup>3</sup> в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму,

використовують наступні методи. При покупці принтера слід вибрати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повісті, поролону, пеноплону).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту ТСО'03.

ТСО'03 пред'являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання (електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [24].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до ДБН В.2.5-28-2006 [27] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(Ен) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛПО2П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки  $a=12$  м, шириною  $b=10$  м, заввишки  $z=4$  м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де  $F$  - світловий потік = 3120 лм;

$E$  - максимально допустима освітленість робочих поверхонь = 200 лк;

$S$  - площа підлоги = 120 м<sup>2</sup>;

$Z$  - поправочний коефіцієнт світильника = 1,2;

$k$  - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

$n$  - кількість світильників;

$U$  - коефіцієнт використання освітлювальної установки = 0,6;

$M$  - кількість ламп у світильнику = 2.

З формули (4.5) виразимо  $n$  (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.6)$$

$$\text{Отже, } n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$$

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

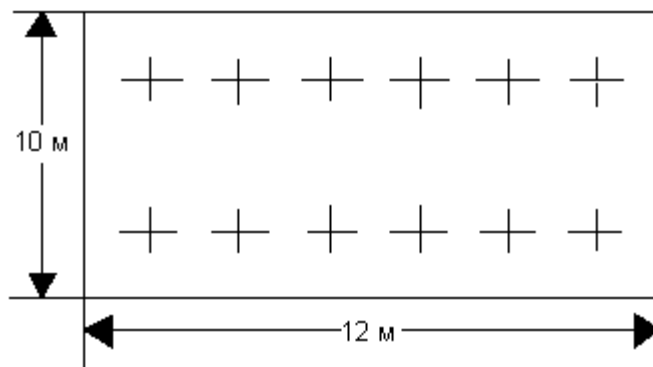


Рисунок 4.1 - Схема розташування світильників

#### 4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 °С ;
- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;
- стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;
- деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 °С, схильна до самозаймання.

Згідно ДСТУ Б В.1.1-36-2016 [28] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегріву від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акролеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигазу з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується і забезпечується системами запобігання пожежам і протипожежному захисту[8]. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% як найменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізольоване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;
- механізація, автоматизація та справність(потокова) виробництва;
- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;
- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- ліквідація можливості самозаймання речовин і матеріалів .

Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;
- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до правил пожежної безпеки для промислових підприємств приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар (сокири, ломи, багри, шерстяну або азбестову ковдри). Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

#### **4.5 Вплив на навколишнє природне середовище**

Діяльність за темою магістерської роботи в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища»[31], Законом



України «Про забезпечення санітарного та епідемічного благополуччя населення»[33], Законом України «Про відходи»[32].

В процесі діяльності виконанням дипломного проектування виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- відпрацьовані люмінесцентні лампи - I клас небезпеки;
- змінні носії інформації - IV клас небезпеки;
- відпрацьовані вогнегасники - IV клас небезпеки;
- макулатура - IV клас небезпеки.

## ВИСНОВКИ

В магістерській роботі були проведені дослідження щодо актуальності, доцільності та практичності систем розпізнання образів, а саме номерних знаків на зображеннях. Був проведений аналіз технологій для реалізації методу детектування даних із зображення, а також технологій для проектування десктопних застосунків.

Був проведений аналіз методів розпізнавання зображень і було виявлено що нейромережеві методи на даний час є найефективнішими, витрачають менше часу на обробку інформації та більш якісно проводять процес розпізнавання, хоча й для їх навчання потребується достатня кількість часу та ресурсів. Завдяки узагальненню, нейронна мережа може зробити висновок про об'єкт, який не був присутній в навчальній вибірці, саме тому було зроблено висновок, що нейронні мережі найбільше підходять для задачі розпізнавання автомобільних номерів. Також слід зазначити, що основною перевагою нейронних мереж є той факт, що вони можуть розвиватися та вдосконалюватися при збільшенні об'ємів даних, що подаються на вхід.

Зокрема було проведений аналіз переваг та недоліків найпопулярніших нейромережевих методів обробки і для розробки було обрано згорткову нейронну мережу, адже на даний момент ЗНМ та її модифікації вважаються кращими по точності і швидкості алгоритмами знаходження об'єктів на зображенні. В результаті були зроблені висновки, щодо можливостей її використання для рішення різних типів задач.

У ході виконання атестаційної роботи був розроблений застосунок і реалізований метод розпізнавання номерних знаків на зображеннях. Як середовище розробки було обрано IntelliJ IDEA сумісно з фреймворком Maven, а сам програмний код був написаний на мові Java. Вихідний продукт призначений для оптимізації та зручності пошуку інформації, орієнтований на використання у вигляді десктопного застосунку. Був проведений аналіз результатів та зроблені висновки щодо доцільності та актуальності теми.

У якості подальшого розвитку теми можна зазначити використання більшої кількості тестових зображень для покращення результатів розпізнавання, а також вдосконалення програмного коду для обробки більшої кількості виняткових ситуацій. Також можливе тестування та навчання нейронної мережі на зображеннях, в яких присутній текст, що схожий за форматом на автомобільний номер.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки,

заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці. А також визначені основні екологічні аспекти впливу на навколишнє природне середовище та зазначені заходи щодо поводження з ними.

Результати атестаційної роботи були апробовані на міжнародній конференції «Майбутній науковець 2020».

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Дэвид А. Форсайт, Джин Понс [Computer Vision: A Modern Approach Компьютерное зрение. Современный подход]. — М.: «Вильямс», 2004. — 928 с.
- 2) Джордж Стокман, Линда Шапиро [Computer Vision Компьютерное зрение]. — М.: Бином. Лаборатория знаний, 2006. — 752 с.
- 3) Дуда Р., Харт П. Распознавание образов и анализ сцен. — М.: Мир, 1976. 2. Фукунага К. Введение в статистическую теорию распознавания образов. — М.: Наука, 1979.
- 4) Главач В., Шлезингер М.И. Десять лекций по статистическому и структурному распознаванию образов. К.: Наукова думка, 2004. [www.irtc.org.ua/image/Files/Schles/esh10\\_full.pdf](http://www.irtc.org.ua/image/Files/Schles/esh10_full.pdf).
- 5) Фисенко, В. Т., & Фисенко, Т. Ю. (2008). Компьютерная обработка и распознавание изображений: учеб. пособие. СПб: СПбГУ ИТМО, 192, 5.
- 6) Анисимов, Б. В., Курганов, В. Д., & Злобин, В. К. (1983). Распознавание и цифровая обработка изображений: Учеб. пособие для вузов. Высш. шк..
- 7) Ту, Д., & Гонсалес, Р. (1978). Принципы распознавания образов: Пер. с англ. Мир.
- 8) Васильев, В. И. (1969). Распознающие системы: справочник. Наукова думка.
- 9) Дорофеюк, А. А. (1971). Алгоритмы автоматической классификации (обзор). Автоматика и телемеханика, 12, 78-113.
- 10) Каллан, Р. (2003). Основные концепции нейронных сетей. Вильямс.
- 11) Хайкин, С. (2008). Нейронные сети: полный курс, 2-е издание. Издательский дом Вильямс.
- 12) Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks, 8(1), 98-113.
- 13) Porikli, F., & Kocak, T. (2006, November). Robust license plate detection using covariance descriptor in a neural network framework. In 2006 IEEE International Conference on Video and Signal Based Surveillance (pp. 107-107). IEEE.
- 14) Уоссермен, Ф. (1992). Нейрокомпьютерная техника: Теория и практика.
- 15) Головкин, В. А., & Галушкин, А. И. (2001). Нейронные сети: обучение, организация и применение. Нейрокомпьютеры и их применение.
- 16) Fu, L., & Kara, L. B. (2011). Neural network-based symbol recognition using a few labeled samples. Computers & Graphics, 35(5), 955-966.

- 17) Флах, П. (2019). Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. Litres.
- 18) Foltyniewicz, R. (1995, September). Efficient high order neural network for rotation, translation and distance invariant recognition of gray scale images. In International Conference on Computer Analysis of Images and Patterns (pp. 424-431). Springer, Berlin, Heidelberg.
- 19) Горелик А.Л., Скрипкин В.А. Методы распознавания. /Учебное пособие.- М.:Высш.шк.,1984. — 208с.
- 20) Кутовецький В.Я, Розпізнання образів: Навчальний посібник. — Миколаїв: Вид-во МДГУ ім. П. Могили, 2003. — 196 с.
- 21) ДСН 3.3.6.042-99 Державні санітарні норми мікроклімату виробничих приміщень. Режим доступу: www. URL: <https://zakon.rada.gov.ua/rada/show/va042282-99>
- 22) НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. Режим доступу: www. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18>
- 23) ДСТУ 7237:2011 Національний стандарт України. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та
- 24) Номенклатура видів захисту. Режим доступу: www. URL: <https://zakon.rada.gov.ua/rada/show/ru/v0037831-11>
- 25) ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Режим доступу: www. URL: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
- 26) ДБН В.2.5-67:2013. Опалення вентиляція та кондиціонування. Режим доступу: www. URL: <https://zakon.rada.gov.ua/rada/show/v0024858-13>
- 27) ДБН В.2.5-28-2006. Природне і штучне освітлення. Режим доступу: www. URL: <https://zakon.rada.gov.ua/rada/show/v0168667-06>
- 28) ДСТУ Б В.1.1-36-2016. Визначення категорії приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою. Режим доступу: www. URL: [http://online.budstandart.com/ua/catalog/doc-page.html?id\\_doc=65419](http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65419)
- 29) ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів Режим доступу: www. URL: <https://zakon.rada.gov.ua/laws/show/z0379-96>
- 30) Симметрон. Электронные компоненты. Каталог 2002, 2002г. – 192с.
- 31) Закон України «Про охорону навколишнього природного середовища» . Вводиться в дію Постановою ВР № 1268-ХІІ від 26.06.91, ВВР, 1991, № 41, ст.547. Режим доступу: www. URL: <https://zakon.rada.gov.ua/laws/show/1264-12>

32) Закон України «Про відходи». Відомості Верховної Ради України (ВВР), 1998, № 36-37, ст.242. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/187/98-вр](http://www.zakon.rada.gov.ua/laws/show/187/98-вр)

33) Закон України «Про забезпечення санітарного та епідемічного благополуччя населення». Відомості Верховної Ради України (ВВР), 1994, № 27, ст.218. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/4004-12](http://www.zakon.rada.gov.ua/laws/show/4004-12)

## ДОДАТОК А.

## Фрагмент лістингу конфігураційного файлу pom.xml

```

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.1</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <debug>>false</debug>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-javadoc-plugin</artifactId>
      <version>3.1.0</version>
      <configuration>
        <detectJavaApiLink>>false</detectJavaApiLink>
        <offlineLinks>
          <offlineLink>
            <url>https://docs.oracle.com/en/java/javase/12/docs/api/</url>
            </offlineLink>
          </offlineLinks>
        </configuration>
      </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-resources-plugin</artifactId>
      <version>2.7</version>
      <configuration>
        <includeEmptyDirs>>true</includeEmptyDirs>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>3.1.1</version>
      <configuration>
        <excludes>
          <exclude>*/logback.xml</exclude>
        </excludes>
        <archive>
          <addMavenDescriptor>>true</addMavenDescriptor>
          <compress>>true</compress>
          <forced>>true</forced>
          <index>>true</index>
          <manifest>
            <addDefaultImplementationEntries>>true</addDefaultImplementationEntries>
            <addDefaultSpecificationEntries>>true</addDefaultSpecificationEntries>
          </manifest>
          <manifestEntries>
            <Implementation-
URL>${project.url}</Implementation-URL>
          </manifestEntries>
        </manifestSections>
      </configuration>
    </plugin>
  </plugins>
</build>

```

```

        <manifestSection>
            <name>Build-Information</name>
            <manifestEntries>
                <Java-Version>${java.version}</Java-
Version>
                <Java-Vendor>${java.vendor}</Java-Vendor>
                <Os-Name>${os.name}</Os-Name>
            </manifestEntries>
        </manifestSection>
    </manifestSections>
</archive>
</configuration>
</plugin>
<plugin>
    <groupId>org.sonatype.plugins</groupId>
    <artifactId>nexus-staging-maven-plugin</artifactId>
    <version>1.6.8</version>
    <extensions>>true</extensions>
    <configuration>
        <serverId>nexus</serverId>
        <nexusUrl>https://oss.sonatype.org/</nexusUrl>
    </configuration>
</plugin>
</plugins>
</build>

```



## ДОДАТОК Б.

## Фрагмент лістингу файлу Recognizer.java

```

abstract class pRes {
    public static List<String> choosedFilePathList;
}

interface EventInjector {
    public void injectView();
    public void injectEvent();
    public void clickAddBt(Stage stage);
    public void clickRemoveBt();
    public void clickStartBt();
    public void clickListMenu();
}

class ROIExtractor {
    public Mat[] getROI(Mat raw) {
        ArrayList<Rect> pureBoundRectList = getPureBoundRectList(raw);
        Mat view = raw.clone();
        for (int i = 0; i < pureBoundRectList.size(); ++i) {
            Imgproc.rectangle(view, pureBoundRectList.get(i), new Scalar(0,
255, 0), 1, 8, 0);
        }

        Imgproc.cvtColor(raw, raw, Imgproc.COLOR_BGR2GRAY);
        Rect[] cutPointRects = getCutPointRects(pureBoundRectList);
        Rect roiRect = getRoiRect(cutPointRects);
        Mat roi = raw.submat(roiRect);
        view = view.submat(roiRect);
        return new Mat[] { roi, view };
    }

    private ArrayList<Rect> getPureBoundRectList(Mat raw) {
        Mat processed = new Mat();
        Imgproc.blur(raw, processed, new Size(2, 2));
        Imgproc.cvtColor(processed, processed, Imgproc.COLOR_BGR2GRAY);
        Imgproc.Canny(processed, processed, 200, 300);
        ArrayList<MatOfPoint> contourList = new ArrayList<>();
        Imgproc.findContours(processed, contourList, new Mat(),
Imgproc.RETR_TREE, Imgproc.CHAIN_APPROX_SIMPLE);

        int contourListSize = contourList.size();
        Rect[] allBoundRects = new Rect[contourListSize];
        for (int i = 0; i < contourListSize; ++i) {
            allBoundRects[i] = new Rect();
        }

        ArrayList<Rect> pureBoundRectList = new ArrayList<>();
        MatOfPoint2f curContourPoly = new MatOfPoint2f();
        for (int i = 0; i < contourListSize; ++i) {
            Imgproc.approxPolyDP(new
MatOfPoint2f(contourList.get(i).toArray()), curContourPoly, 1, true);
            allBoundRects[i] = Imgproc.boundingRect(curContourPoly);
        }

        double ratio = -1;
        for (int i = 0; i < contourListSize; ++i) {
            ratio = (double) allBoundRects[i].height /
allBoundRects[i].width;

```

```

        if ((0.5 <= ratio) && (ratio <= 2.5)) {
            if ((150 <= allBoundRects[i].area()) &&
(allBoundRects[i].area() <= 1200)) {
                pureBoundRectList.add(allBoundRects[i]);
            }
        }
    }

    return pureBoundRectList;
}

private Rect[] getCutPointRects(ArrayList<Rect> pureBoundRectList) {
    pureBoundRectList.sort((prev, next) -> {
        return Double.compare(prev.tl().x, next.tl().x);
    });

    int maxCount = 0;
    double deltaX = 0;
    double deltaY = 0;
    double gradient = 0;
    final double toleranceAreaRatio = 0.25;
    Rect maxCountRect = null;
    Rect endRectOfMaxCountRect = null;
    for (int i = 0; i < pureBoundRectList.size(); ++i) {
        int curCount = 0;
        double curStdArea = pureBoundRectList.get(i).area();
        double tolerance = curStdArea * toleranceAreaRatio;
        double minArea = curStdArea - tolerance;
        double maxArea = curStdArea + tolerance;

        for (int j = i + 1; j < pureBoundRectList.size(); ++j) {
            deltaX = Math.abs(pureBoundRectList.get(j).tl().x -
pureBoundRectList.get(i).tl().x);
            if (150 < deltaX) {
                break;
            }

            deltaY = Math.abs(pureBoundRectList.get(j).tl().y -
pureBoundRectList.get(i).tl().y);
            if (deltaX == 0) {
                deltaX = 1;
            }

            if (deltaY == 0) {
                deltaY = 1;
            }

            gradient = deltaY / deltaX;
            double curArea = pureBoundRectList.get(j).area();
            if (gradient < 0.12 && minArea <= curArea && curArea <=
maxArea) {
                ++curCount;
                if (maxCount <= curCount) {
                    endRectOfMaxCountRect = pureBoundRectList.get(j);
                }
            }
        }

        if (maxCount < curCount) {
            maxCount = curCount;
            maxCountRect = pureBoundRectList.get(i);
        }
    }
}

```

```

        return new Rect[] { maxCountRect, endRectOfMaxCountRect };
    }

    private Rect getRoiRect(Rect[] cutPointRects) {
        Rect startRect = cutPointRects[0];
        Rect endRect = cutPointRects[1];
        double ltlx = startRect.tl().x;
        double ltly = startRect.tl().y;
        double lbry = startRect.br().y;
        double rtly = endRect.tl().y;
        double rbrx = endRect.br().x;
        double rbry = endRect.br().y;
        double widthVariation = 1;
        double heightVariation = 1;

        Point roiStartPoint = null;
        Point roiEndPoint = null;
        if (rtly <= ltly) {
            roiStartPoint = new Point(ltlx - widthVariation, rtly -
heightVariation);
            roiEndPoint = new Point(rbrx + widthVariation, lbry +
heightVariation);
        } else {
            roiStartPoint = new Point(ltlx - widthVariation, ltly -
heightVariation);
            roiEndPoint = new Point(rbrx + widthVariation, rbry +
heightVariation);
        }

        return new Rect(roiStartPoint, roiEndPoint);
    }
}

class OCRReader {

    private Tesseract tesseract;

    public OCRReader() {
        tesseract = new Tesseract();
        tesseract.setDatapath("src/main/resources/tessdata");
        tesseract.setLanguage("eng");
    }

    public String getOcrResult(Mat roi) {
        Imgcodecs.imwrite("tmp.jpg", roi);
        String tmpOcrResult = "";
        try {
            tmpOcrResult = tesseract.doOCR(new File("tmp.jpg"));
        } catch (TesseractException e) {
            e.printStackTrace();
        }

        String ocrResult = "";
        char[] iso = tmpOcrResult.toCharArray();
        for (char c : iso) {
            if (isPlateResult(c)) {
                ocrResult += c;
            }
        }

        return ocrResult;
    }

    private boolean isPlateResult(char c) {

```

```

        return ('0' <= c && c <= '9') || ('A' <= c && c <= 'Z');
    }
}

class ValidExtensionChecker {

    private String[] validExtension = { "png", "jpg", "bmp", "jpeg", "PNG",
    "JPG", "BMP", "JPEG" };
    public boolean isValidExtension(String absPath) {
        int dotIdx = -1;
        for (int i = absPath.length() - 1; i >= 0; --i) {
            if (absPath.charAt(i) == '.') {
                dotIdx = i;
                break;
            }
        }

        if (dotIdx == -1) {
            return false;
        }

        String extension = absPath.substring(dotIdx + 1);
        System.out.println(extension);
        for (String curValidExtension : this.validExtension) {
            if (extension.equals(curValidExtension)) {
                return true;
            }
        }

        return false;
    }
}

abstract class View {

    public static ImageView imgView, roiView;
    public static ListView<String> imgList;
    public static Button addBt, removeBt, startBt;
    public static Label resultLb;
}

public class Recognizer extends Application implements Initializable,
EventInjector {

    static {
        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
    }

    private ValidExtensionChecker validExtensionChecker;

    private ROIExtractor roiExtractor;

    private OCRReader ocrReader;

    private Stage stage;

    @FXML
    private ImageView imgView;

    @FXML
    private ImageView roiView;

    @FXML
    private ListView<String> imgList;
}

```

```

@FXML
private Button addBt, removeBt, startBt;

@FXML
private Label resultLb;

@Override
public void injectView() {
    View.imgView = this.imgView;
    View.roiView = this.roiView;
    View.imgList = this.imgList;
    View.addBt = this.addBt;
    View.removeBt = this.removeBt;
    View.startBt = this.startBt;
    View.resultLb = this.resultLb;
}

@Override
public void injectEvent() {
    View.addBt.setOnMouseClicked(event -> this.clickAddBt(stage));
    View.removeBt.setOnMouseClicked(event -> this.clickRemoveBt());
    View.startBt.setOnMouseClicked(event -> this.clickStartBt());
    View.imgList.setOnMouseClicked(event -> this.clickListMenu());
}

@Override
public void clickAddBt(Stage stage) {
    FileChooser fileChooser = new FileChooser();
    List<File> fileList = fileChooser.showOpenMultipleDialog(stage);
    if (fileList == null) {
        return;
    }

    String absPath = null;
    String invalidPath = "";
    for (File curFile : fileList) {
        absPath = curFile.getAbsolutePath();
        if (!validExtensionChecker.isValidExtension(absPath)) {
            invalidPath += absPath +
System.getProperty("line.separator");
            continue;
        }

        pRes.choosedFilePathList.add(absPath);
        View.imgList.getItems().add(curFile.getName());
    }

    if (!invalidPath.equals("")) {
        Alert alert = new Alert(AlertType.ERROR);
        alert.setTitle("Error");
        alert.setContentText(invalidPath);
        alert.showAndWait();
    }
}

@Override
public void clickRemoveBt() {
    int removeIdx = View.imgList.getFocusModel().getFocusedIndex();
    if (removeIdx == -1) {
        return;
    }

    View.imgList.getItems().remove(removeIdx);
}

```

```

pRes.choosedFilePathList.remove(removeIdx);
if (pRes.choosedFilePathList.size() == 0) {
    View.imgView.setImage(null);
    View.roiView.setImage(null);
    View.resultLb.setText(null);
}

for (String curPath : pRes.choosedFilePathList) {
    System.out.println(curPath);
}
}

@Override
public void clickStartBt() {
    Platform.runLater(() -> {
        new Thread(() -> {
            for (int i = 0; i < pRes.choosedFilePathList.size(); ++i) {
                View.imgList.getSelectionModel().select(i);
                clickListMenu();
                Mat curRaw =
Imgcodecs.imread(pRes.choosedFilePathList.get(i));
                Mat[] roi = roiExtractor.getROI(curRaw);
                String viewName = "tmpView";
                String viewPath = "tmp/" + viewName + ".jpg";
                Imgcodecs.imwrite(viewPath, roi[1]);
                try {
FileInputStream(viewPath));
                    Image image = new Image(new
View.roiView.setImage(null);
                    View.roiView.setImage(image);
                    setImageCenter(View.roiView, image);
                } catch (FileNotFoundException e1) {
                    e1.printStackTrace();
                }

                Platform.runLater(() -> {
                    String ocrResult =
ocrReader.getOcrResult(roi[0]);
                    System.out.println(ocrResult);
                    View.resultLb.setText(ocrResult);
                });

                try {
                    Thread.sleep(2000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }).start();
    });
}

@Override
public void clickListMenu() {
    int clickedIdx = View.imgList.getFocusModel().getFocusedIndex();
    if (clickedIdx == -1) {
        return;
    }

    String absPath = pRes.choosedFilePathList.get(clickedIdx);
    Image image = null;
    try {
        image = new Image(new FileInputStream(absPath));
        View.imgList.scrollTo(clickedIdx);
    }
}

```

```

    } catch (FileNotFoundException e) {
        System.out.println("file not found");
        return;
    }

    View.imgView.setImage(image);
    setImageCenter(View.imgView, image);
}

private void setImageCenter(ImageView imageView, Image draggedImage) {
    double ratioX = imageView.getFitWidth() / draggedImage.getWidth();
    double ratioY = imageView.getFitHeight() / draggedImage.getHeight();

    double reducCoeff;
    if (ratioX >= ratioY) {
        reducCoeff = ratioY;
    } else {
        reducCoeff = ratioX;
    }

    double w = draggedImage.getWidth() * reducCoeff;
    double h = draggedImage.getHeight() * reducCoeff;

    imageView.setX((imageView.getFitWidth() - w) / 2);
    imageView.setY((imageView.getFitHeight() - h) / 2);
}

@Override
public void initialize(URL location, ResourceBundle resources) {
    validExtensionChecker = new ValidExtensionChecker();
    roiExtractor = new ROIExtractor();
    ocrReader = new OCRReader();
    injectView();
    injectEvent();
}

@Override
public void start(Stage primaryStage) throws Exception {
    Parent fxml = FXMLLoader.load(getClass().getResource("gui.fxml"));
    primaryStage.setScene(new Scene(fxml));
    primaryStage.setTitle("Number plate recognizer by inzapp");
    primaryStage.setResizable(false);
    primaryStage.setOnCloseRequest(event -> {
        new File("tmp.jpg").delete();
        System.exit(0);
    });

    primaryStage.show();
    this.stage = primaryStage;
    pRes.choosedFilePathList = new ArrayList<>();
}

public static void main(String[] args) {
    launch(args);
}
}

```

## ДОДАТОК В. Електронні плакати

*Міністерство освіти і науки України  
Східноукраїнський національний  
університет ім. В.Даля  
Кафедра комп'ютерних наук та інженерії*

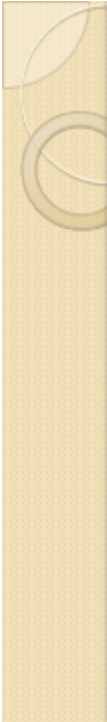
*Магістерська робота  
«Методи аналізу та прогнозування  
бізнес даних»*

*ст.гр. КН-19дм Запорозьський М.А.  
Керівник: Рязанцев О.І.*

### Актуальність роботи

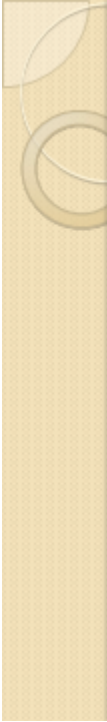
- **Метою роботи** є розробка програмного застосунку, що базується на згортковій нейронній мережі та дозволяє розпізнавати номерні знаки автомобіля на зображеннях.
- **Об'єктом дослідження** є нейромережеві методи, що можуть бути використані для розпізнання об'єктів на зображенні, зокрема автомобільних номерних знаків.
- **Робота присвячена** вирішенню завдання розпізнавання зображень та ідентифікації автомобільних номерних знаків для подальшого їх застосування у різних сферах.





## Сфери застосування розпізнавання образів

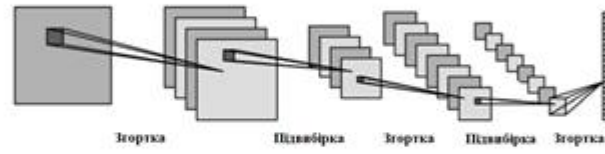
- розпізнавання тексту;
- розпізнавання автомобільних номерів;
- розпізнавання об'єктів на картах;
- розпізнавання облич та інших біометричних даних;
- розпізнавання штрих-кодів;
- розпізнавання мови.



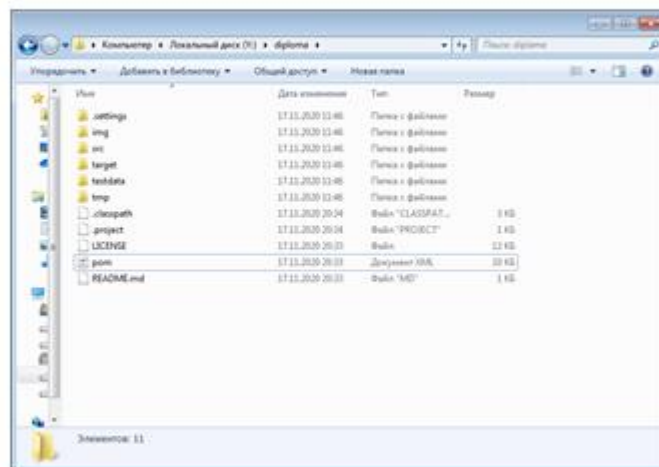
## Алгоритми та технології розпізнавання номерних знаків

- приведення вихідного зображення до виду, який не залежить від умов реєстрації зображення (ступінь освітленості, нерівномірність розподілу яскравості від джерел світла, розмитість, зашумленість тощо);
- виділення на отриманому зображенні областей-кандидатів, потенційно містять пластину з номером;
- проведення детального аналізу областей-кандидатів на основі формального уявлення масштабних характеристик номерної пластини і скорочення простору для подальшого пошуку;
- приведення до стандартного розміру графічного зображення номерної пластини з корекцією якості зображення;
- попереднє визначення типу номерної пластини (в прив'язці до діючих стандартів);
- витяг окремих символів і їх розпізнавання (аналіз символів за ключовими характеристиками, незалежним від масштабу, використовуваного шрифту, геометричних спотворень і розривів);
- уточнення результатів розпізнавання на основі інформації про тип номера і за результатами з попередніх кадрів.

## Архітектура згорткової нейронної мережі



## Програмна реалізація



## Програмна реалізація



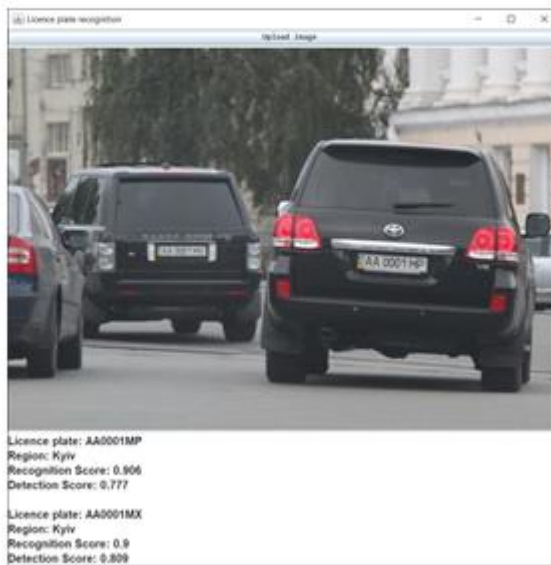
## Програмна реалізація



## Програмна реалізація



## Програмна реалізація



## ВИСНОВКИ

Був проведений аналіз методів розпізнавання зображень і було виявлено що нейромережеві методи на даний час є найефективнішими, затрачають менше часу на обробку інформації та більш якісно проводять процес розпізнавання, хоча й для їх навчання потребується достатня кількість часу та ресурсів.

У ході виконання атестаційної роботи був розроблений застосунок і реалізований метод розпізнавання номерних знаків на зображеннях. Як середовище розробки було обрано IntelliJ IDEA сумісно з фреймворком Maven, а сам програмний код був написаний на мові Java. Вихідний продукт призначений для оптимізації та зручності пошуку інформації, орієнтований на використання у вигляді десктопного застосунку. Був проведений аналіз результатів та зроблені висновки щодо доцільності та актуальності теми.

У якості подальшого розвитку теми можна зазначити використання більшої кількості тестових зображень для покращення результатів розпізнавання, а також вдосконалення програмного коду для обробки більшої кількості виняткових ситуацій. Також можливе тестування та навчання нейронної мережі на зображеннях, в яких присутній текст, що схожий за форматом на автомобільний номер.