

Київський національний університет імені Тараса Шевченка  
Факультет інформаційних технологій  
Кафедра програмних систем і технологій

УДК 004.056.55

*На правах рукопису*

# **ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА**

Тема: “Нейромережне розпізнавання об’єкта на зображенні (номерних  
знаків автомобілів)”

Спеціальність – 121 “Інженерія програмного забезпечення”

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

БР.ІПЗ – 30.00.00.000

Студент

ІПЗ-41 \_\_\_\_\_ /Артем МУХА/

Науковий керівник

асист. \_\_\_\_\_ /Тетяна КОВАЛЮК/

Консультант

з питань нормоконтролю

фахівець \_\_\_\_\_ /Тамара ЧАПОВСЬКА/

Допускається до захисту

Завідувач кафедри

д.т.н., проф. \_\_\_\_\_ /Олексій БИЧКОВ/

Київ – 2021

Рішенням Екзаменаційної комісії  
випускна кваліфікаційна робота студента

---

захищена з оцінкою

---

Голова Екзаменаційної комісії  
професор, доктор техн. наук Андрій БОНДАРЧУК

**Київський національний університет імені Тараса Шевченка**

Факультет інформаційних технологій

Кафедра програмних систем і технологій

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖЕНО

Зав. кафедри програмних систем і технологій

\_\_\_\_\_ (Олексій БИЧКОВ)

(підпис)

(прізвище та ініціали)

## **ЗАВДАННЯ**

### **НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ**

#### **РОБОТУ СТУДЕНТУ**

Муха Артем В'ячеславович

**1. Тема бакалаврської роботи “ Нейромережне розпізнавання об’єкта на зображенні(номерних знаків автомобілів) ” керівник проекту (роботи) Ковалюк Тетяна Володимирівна**

затверджені наказом вищого навчального закладу від “ 11 ” листопада 2020 р. № 6

**2. Строк подання студентом роботи 30 травня 2021 р.**

**3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних та програмних технологій направлених на нейромережне розпізнавання зображення.**

**4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)**

1. Огляд предметної галузі і постановка задачі дослідження

2. Особливості застосування нейромережного розпізнавання об’єкта на зображенні

3. Реалізація нейромережного модулю розпізнавання символів

4. Тестування та аналіз результатів

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

1. Штучна нейронна мережа з прямим зв’язком(рис. 1.1, ст. 15)

2. Сприйнятливості поле нейрона (рис. 2.2, ст. 36)

3. Схематична ілюстрація архітектури мережі (рис. 3.1, ст. 45)

4. Алгоритм роботи програми (рис. 3.2, ст. 47)

5. Приклад розпізнання номерного знаку програмою (рис. 4.5, ст. 55)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Розділ 1	Ковалюк Т.В.	15.03.2021	16.03.2021
Розділ 2	Ковалюк Т.В.	10.04.2021	12.04.2021
Розділ 3	Ковалюк Т.В.	18.04.2021	20.04.2021
Розділ 4	Ковалюк Т.В.	25.04.2021	27.04.2021

7. Дата видачі завдання 09 жовтня 2020 р.

Керівник \_\_\_\_\_ (Тетяна КОВАЛЮК)

Завдання прийняв до виконання \_\_\_\_\_ (Артем МУХА)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури	25.11.2020	виконано
2	Аналіз концепцій та алгоритмів	12.01.2021	виконано
3	Вивчення концепції нейромережного розпізнавання зображення	15.02.2021	виконано
4	Розробка алгоритмічної моделі	15.03.2021	виконано
5	Опис розробленого алгоритму	25.03.2019	виконано
6	Розробка та тестування програмного забезпечення	05.05.2020	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	28.05.2020	виконано

Студент – бакалавр \_\_\_\_\_ (Артем МУХА)

Керівник роботи \_\_\_\_\_ (Тетяна КОВАЛЮК)

## АНОТАЦІЯ

**Випускна кваліфікаційна бакалаврська робота:** 60 с., 24 рис., 18 джерела.

**Тема:** Нейромережне розпізнавання об'єкта на зображенні (номерних знаків автомобілів).

**Об'єкт дослідження:** процес проектування та навчання нейронної мережі для розпізнавання символів.

**Мета роботи:** аналіз особливостей розробки та навчання нейронної мережі для розпізнавання символів.

**Предмет дослідження:** специфіка розробки та навчання нейронної мережі для розпізнавання номерних знаків автомобілів в системі відеоспостереження.

### **Результати дослідження:**

Досліджено можливості використання штучного інтелекту під час розпізнавання об'єкта на зображенні (номерні знаки автомобілів).

### **Висновок**

В результаті реалізації інтелектуального модуля, що функціонує у відповідності із запропонованим алгоритмом, розпізнаються символи, які належать до будь-якого алфавіту, цифри й знаки в залежності від об'ємів пам'яті інтелектуальної системи, що містить відповідні еталони із зашумленістю до 40 %. Розроблено програмне забезпечення для розпізнавання номерних знаків автомобілів в системі відеоспостереження.

РОЗПІЗНАННЯ СИМВОЛІВ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, НЕЙРОННІ МЕРЕЖІ, ШТУЧНИЙ ІНТЕЛЕКТ, МЕТОДИ НАВЧАННЯ.

## SUMMARY

**Final qualifying bachelor's thesis:** 60 pages, 24 figures, 18 sources.

**Topic:** Neural network recognition of an object in an image (car license plates).

**Object of research:** the process of designing and training a neural network for character recognition.

**Purpose:** analysis of features of development and training of a neural network for character recognition.

**Subject of research:** specifics of development and training of a neural network for recognition of license plates of cars.

### **Research results:**

The possibilities of using artificial intelligence when recognizing an object in an image (car license plates) have been studied.

### **Conclusion**

As a result of the implementation of the intelligent module, which operates in accordance with the proposed algorithm, recognizes symbols belonging to any alphabet, numbers and symbols depending on the memory of the intelligent system, which contains the appropriate standards with a noise level of up to 40%. Software for recognizing car license plates in the video surveillance system has been developed.

SYMBOL RECOGNITION, SOFTWARE, INTELLECTUAL INFORMATION TECHNOLOGIES, NEURAL NETWORKS, ARTIFICIAL INTELLIGENCE, MECHANICAL INTELLIGENCE.

## АННОТАЦИЯ

**Выпускная квалификационная бакалаврская работа:** 60 с., 24 рис., 18 источника.

**Тема:** нейросети распознавания объекта на изображении (номерных знаков автомобилей).

**Объект исследования:** процесс проектирования и обучения нейронной сети для распознавания символов.

**Цель работы:** анализ особенностей разработки и обучения нейронной сети для распознавания символов.

**Предмет исследования:** специфика разработки и обучения нейронной сети для распознавания номерных знаков автомобилей.

### **Результаты исследования:**

Исследованы возможности использования искусственного интеллекта при распознавании объекта на изображении (номерные знаки автомобилей).

### **Вывод**

В результате реализации интеллектуального модуля функционирует в соответствии с предложенным алгоритмом, распознаются символы, которые относятся к любому алфавиту, цифры и знаки в зависимости от объемов памяти интеллектуальной системы, содержащей соответствующие эталоны с зашумленностью до 40%. Разработано программное обеспечение для распознавания номерных знаков автомобилей в системе видеонаблюдения.

РАСПОЗНАВАНИЯ ТЕКСТА, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, НЕЙРОННЫЕ СЕТИ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, МЕТОДЫ ОБУЧЕНИЯ.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	8
ВСТУП.....	9
<b>РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ</b>	
1.1 Розпізнавання об’єктів. Основні принципи .....	12
1.2 Визначення поняття «нейронної мережі» та його сутність .....	14
1.3 Класифікація нейронних мереж .....	16
1.4 Алгоритми навчання нейронних мереж .....	21
1.5 Постановка задачі дослідження.....	24
Висновки до розділу 1.....	25
<b>РОЗДІЛ 2 ОСОБЛИВОСТІ ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖНОГО РОЗПІЗНАВАННЯ ОБ’ЄКТА НА ЗОБРАЖЕННІ</b>	
2.1 Специфіка застосування нейронних мереж .....	26
2.2 Огляд програмних засобів, що використовують нейронні мережі.....	31
2.3 Загальна модель згорткової нейронної мережі.....	33
Висновки до розділу 2.....	39
<b>РОЗДІЛ 3 РЕАЛІЗАЦІЯ НЕЙРОМЕРЕЖНОГО МОДУЛЮ РОЗПІЗНАВАННЯ СИМВОЛІВ (НА ПРИКЛАДІ НОМЕРНИХ ЗНАКІВ АВТОМОБІЛІВ)</b>	
3.1 Обґрунтування вибору середовища програмної реалізації.....	41
3.1.1 Мова програмування Python.....	41
3.1.2 Середовище розробки PyCharm на основі IntelliJ IDEA .....	42
3.1.3 Модель YOLOv5.....	42
3.1.4 Детектор тексту CRAFT.....	43
3.2 Програмна реалізація .....	45
Висновки до розділу 3.....	49

**РОЗДІЛ 4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ**

4.1 Інструкція користувача .....	50
4.2 Тестування розробленої моделі .....	51
Висновки до розділу 4.....	54
ВИСНОВКИ .....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	57
ДОДАТКИ.....	59

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ЗНМ – згорткова нейронна мережа;

ІБД – інтегрована база даних;

ІІТ – інтелектуальні інформаційні технології;

ЕОД – електронний обмін даними;

НМ – нейронна мережа;

ПК – персональний комп'ютер,

РНМ – рекурентна нейронна мережа;

СРС – система розпізнавання символів,

ШНМ – штучні нейронні мережі.

## ВСТУП

**Актуальність дослідження.** В даний час комп'ютеризація в нашому суспільстві рухається дуже швидкими темпами і відіграє величезну роль в житті людини. За допомогою комп'ютерних технологій автоматизується широке коло процесів, які ще в недалекому минулому були сферою компетенції людини. Інформаційні технології використовуються всюди: в промисловості, в транспорті, в побуті тощо. Програмісти всього світу розробляють нові і вдосконалюють вже існуючі алгоритми автоматизації.

Рішення проблеми ідентифікації автомобіля за реєстраційним номерним знаком є важливим аспектом безпеки та контролю. Використовувати такий продукт можна в різних сферах застосування, що стосуються автотранспорту. Прикладом можуть служити автотранспортні підприємства, заправні станції, контроль швидкості руху, автомобільні стоянки, контроль в'їзду на територію підприємства тощо.

Варто акцентувати, що наразі існує багато підходів стосовно вирішення проблеми розпізнавання символів, але більшість із них або вузько спрямовані на певну сферу розпізнавання (вони показують високі результати розпізнавання, але не є універсальними), або якість розпізнавання є дуже низькою, й сам метод працює незадовільно. Тому для вирішення задач оптимізації СРС часто застосовуються високоінтелектуальні системи на основі ШНМ. Однак штучні нейронні мережі не є інструментом для вирішення задач будь-якого типу. Вони є непридатними для виконання таких задач як нарахування зарплатні, проте вони мають перевагу під час задач розпізнавання символів, з якими погано або взагалі не справляються звичайні ПК. Так, наприклад, системи відеоспостереження, що використовують штучний інтелект та принцип нейромережного розпізнання об'єкта, наразі активно розвиваються та з успіхом починають застосовуватись на масштабному рівні. Покращується як наукова база ШНМ, так і обчислювальні потужності відповідного технічного обладнання.

Завдання ідентифікації автомобіля можна умовно розділити на дві підзадачі: локалізація номерної пластини і розпізнавання символів. Дана робота присвячена

розробленню і реалізації алгоритму розпізнавання номерного знаку. У загальному випадку розпізнавання реалізується в три етапи: попередня обробка зображення, сегментація і саме розпізнавання символів.

Попередня обробка зображення полягає у виділенні номерної пластини і обробці отриманого зображення різними фільтрами з метою поліпшення якості. На етапі сегментації виділяються символи, які потім ідентифікуються оптичним розпізнаванням символів

Таким чином, постає необхідність у застосуванні сучасного підходу до проектування автоматизованих систем розпізнавання номерних знаків за допомогою використання комп'ютерного зору, а це, в свою чергу, вимагає залучення новітніх способів обробки інформаційних ресурсів. Для цього використовуються методи, які засновані на розпізнаванні образів, що сприяють вирішенню наступних завдань:

- завдання ідентифікації;
- завдання кластеризації;
- завдання класифікації процесів, сигналів, символів.

Не секрет, що використання методу розпізнавання об'єктів на зображенні, підвищує ефективність розробки автоматизованого процесу, зменшує матеріальні та часові витрати, сприяє отриманню об'єктивних та оперативних даних в режимі он-лайн. Виходячи з вищенаведеного, наше дослідження особливостей розробки та практичного застосування нейромережного розпізнавання об'єкта на зображенні в системі відеоспостереження є актуальним.

**Мета та завдання дослідження.** *Метою роботи* було проаналізувати особливості розробки та навчання нейронної мережі для розпізнавання символів.

Для досягнення мети були поставлені наступні *завдання*:

- розглянути теоретичні засади навчання нейронних мереж;
- дослідити класифікацію нейронних мереж;
- визначити алгоритми навчання нейронних мереж;
- проаналізувати особливості застосування нейромережного розпізнавання об'єкта на зображенні;

– розробити алгоритм ШНМ для розпізнавання номерних знаків автомобілів та його програмну реалізацію.

**Об’єкт дослідження** – процес проектування та навчання нейронної мережі для розпізнавання символів.

**Предмет дослідження** – специфіка розробки та навчання нейронної мережі для розпізнавання номерних знаків автомобілів на зображенні.

У роботі застосовуються такі загальнонаукові **методи дослідження** як

- методи системного аналізу,
- аналіз наукової літератури,
- спостереження,
- абстрагування,
- узагальнення.

**Джерела дослідження** представлені науковими працями таких науковців, як Г. Аваліані, Ю. Борисов, Г. Вороновський, О. Галушкін, Б. Головкін, В. Комашинський, В. Круглов, В. Месюра, Д. Поліщук, Б. Советов, Ф. Уосерман, Л. Ясницький та інші.

**Наукова новизна одержаних результатів.** Результати дослідження пропонують альтернативний метод застосування штучного інтелекту під час розпізнавання символів на номерних знаках автомобілів.

**Практичне значення одержаних результатів** полягає в тому, що дослідження ґрунтується на результатах поглибленого вивчення особливостей застосування перенесення навчання нейронних мереж, а також розроблено модуль розпізнавання номерних знаків автомобілів.

**Апробація результатів дослідження.** Результати дослідження були апробовані на кафедрі програмних систем і технологій КНУ ім. Т. Шевченка.

**Публікації.** За результатами наукового дослідження опубліковано наукову статтю.

**Структура роботи.** Дипломна робота складається зі вступу, трьох розділів, висновків, списку джерел посилання, додатків. Основний зміст роботи викладено на 60 сторінках. Список використаних джерел складається з 18 найменувань.

## РОЗДІЛ 1

### ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

#### 1.1 Розпізнавання об'єктів. Основні принципи

Розпізнавання об'єктів - це технологія комп'ютерного зору, яка використовується для ідентифікації об'єктів на зображеннях чи відео. Розпізнавання об'єктів це важливий результат алгоритмів глибокого та машинного навчання. Коли люди переглядають фотографію чи відео, вони можуть легко бачити та впізнавати інших людей, предмети, сцени та візуальні деталі. Задача заключається в тому, щоб навчити програму робити те, що дано людині від природи – розрізняти предмети на зображенні.

Є багато підходів, які використовуються різні у сфері розпізнавання об'єктів. Останнім часом популярним підходом для розпізнавання об'єктів стали методи машинного навчання та глибокого навчання. Обидві методика вчаться ідентифікувати предмети на зображеннях, але відрізняються за своїми алгоритмами.

Останнім часом технології глибокого навчання стають все більш популярними. Вони використовуються для автоматичного вивчення властивостей об'єкта з метою його ідентифікації. Найпопулярнішими представниками є Convolutional NM і CNN. Наприклад, CNN може навчитися визначати різницю між котами та собаками, аналізуючи тисячі зображень та вивчаючи особливості, які роблять цих тварин різними для комп'ютерного зору.

Глибоке навчання використовує два підходи для розпізнання об'єктів:

Вивчення моделі з нуля: щоб навчити мережу глибокого навчання з нуля, вам потрібно зібрати дуже великий набір міток даних та розробити мережеву архітектуру, яка вивчає функції та будує модель. Такий підхід вимагає великої кількості навчальних даних, і вам потрібно відрегулювати шари та ваги в CNN (згорткові нейронні мережі), але точність розпізнавання перебиває всі труднощі.

Використання вже вивченої моделі глибокого навчання: більшість програм глибокого навчання користуються підходом передачі навчання, процес, що передбачає точне налаштування перевіреної моделі. Береться за основу вже частково

навчена модель, така як AlexNet або GoogLeNet, і проходить процес навчання, але вже новими даними і раніше невідомими класами. Цей метод підходить якщо ви шукаєте швидкого результату, без докладання зусиль, оскільки вже навчена тисячами зображень.

З використанням стандартного методу для машинного навчання, береться колекція зображень (або відео) та вибирається відповідна функція для кожного зображення. Розглянемо на прикладі алгоритма вилучення можливостей, який вилучає кутові або крайові функції, які потім використовують для розмежування класів.

Ці функції додаються до моделі машинного навчання, яка розділяє ці функції на окремі категорії, а потім використовує цю інформацію для аналізу та класифікації нових об'єктів.

Швидше за все, буде потрібно використовувати різноманітні алгоритми машинного навчання та методи вилучення, пропонуючи безліч комбінацій, для побудови точної моделі розпізнавання об'єктів.

Використання машинного навчання для розпізнавання об'єктів забезпечує гнучкість вибору найкращого поєднання функцій та класифікаторів для навчання. Це дозволяє отримати точні результати з мінімальними даними.

*Машинне навчання проти глибокого навчання для розпізнавання об'єктів:*

Визначення найкращого підходу до виявлення об'єктів залежить від програми та вирішуваної проблеми. У багатьох випадках машинне навчання може бути ефективною технікою, особливо якщо ви знаєте, які особливості чи характеристики зображення найкраще використовувати для розрізнення класів предметів.

Вибираючи між машинним та глибоким навчанням, головне, що слід врахувати, - чи можна використовувати потужний комп'ютер і чи використовується велика кількість зображень для тренувань. Якщо відповідь на будь-яке з цих питань негативна, найкращим варіантом може стати підхід до машинного навчання. Технології глибокого навчання, як правило, працюють краще з більшою кількістю зображень, а графічний процесор (графічний процесор) допомагає скоротити час, необхідний для навчання моделі.

Насправді функціональність виявлення об'єкта складається з трьох підфункцій або етапів, реалізація яких не повинна бути пов'язана, оскільки вона може виконуватися різними підсистемами з різними інструментами та підходами. Головне, щоб підсистеми знали формат зв'язку між ними. Це дозволяє вибрати оптимальний інструмент для реалізації кожного з етапів без прив'язки до єдиного стосу. Ці три фази:

- detection (виявлення) - знаходження об'єкту на зображенні.
- Recognition (упізнавання) – визнання, отримання інформації про виявлений об'єкт.
- Identification (ідентифікація) – розпізнавання деталей об'єкту, вилучення інформації.

## 1.2 Визначення поняття «нейронної мережі» та його сутність

Нейронна мережа (НМ) являє собою ланцюг нейронів, іншими словами штучна нейронна мережа (ШНМ) мережа складається з штучних нейронів або вузлів (Рис. 1.1). Таким чином, нейронна мережа - це або біологічна нейронна мережа, що складається з реальних біологічних нейронів, або штучна нейронна мережа, яка створюється для вирішення завдань штучного інтелекту. ШНМ можуть використовуватися для прогнозного моделювання, адаптивного управління і додатків, де їх можна навчати за допомогою набору даних. Самонавчання на основі досвіду може відбуватися в мережах, які можуть робити висновки зі складного і, здавалося б, незв'язаного набору інформації [1].

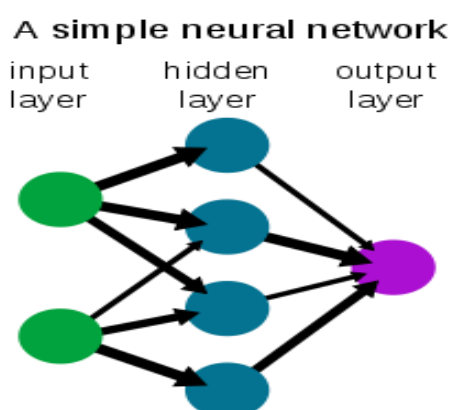


Рис. 1.1 Спрощений вид штучної нейронної мережі з прямим зв'язком

НМ складається з груп, хімічно чи функціонально пов'язаних один з одним нейронів. Один нейрон може пов'язуватися з багатьма іншими нейронами, і загальне число нейронів та з'єднань в мережі іноді доволі значне. Зв'язки, що носять назву синапсів, зазвичай утворюються від аксонів по дендритам, хоча можливі і інші зв'язки. Крім електричних сигналів, існують і інші форми передачі сигналів, що виникають в результаті дифузії нейромедіаторів. Штучний інтелект, когнітивне моделювання та НМ - це парадигми обробки інформації, натхненні тим, як біологічні нейронні системи обробляють дані. Штучний інтелект і когнітивне моделювання намагаються імітувати деякі властивості біологічних нейронних мереж. В сфері штучного інтелекту ШНМ успішно застосовуються для розпізнавання мови, аналізу зображень та адаптивного управління з метою створення програмних агентів (в комп'ютерних і відеоіграх) або автономних роботів [2].

Варто додати, що успішний розвиток дисципліни ШНМ за останні роки дозволив впровадити комплекс таких глобальних характеристик, як:

- навчання,
- узагальнення,
- абстрагування.

ШНМ імітують в першому наближенні ознаки біологічних нейронів. На вхід ШН поступає множинність сигналів, що є виходами інших нейронів. Кожен такий вхід множиться на значення відповідної вагу, що є аналогічною його синаптичній силі. Потім загальна кількість виходів підсумовується, визначається ступінь активації нейронів [3].

Отже, аналіз ШНМ перш за все пов'язаний з тим, що вони допомагають наблизитися до здатності оброблювати інформацію людським мозком, що являє собою надскладний, нелінійний, паралельний ПК системи оброблення даних. Мозок може організовувати власні компоненти, які називаються нейронами, таким чином, аби вони змогли виконати конкретні завдання на кшталт розпізнавання образів, оброблення сигналів органів чуття чи моторні функції в рази швидше, ніж це

дозволяють зробити найшвидкодійочі сучасні ПК. Наразі є багато прикладів застосування ШНМ для галузі прогнозування, класифікацій, оптимізаційних рішень і розпізнавання образів.

Моделі нейронних мереж поділяються на:

- програмні моделі,
- апаратні моделі,
- моделі із застосування розпізнавання образів,
- моделі прогнозування,
- моделі проектування асоціативної пам'яті.

### **1.3 Класифікація нейронних мереж**

Штучні нейронні мережі поділяються [3]:

1) По типу вхідної інформації:

- аналогові (використання інформації в формі дійсних чисел);
- двійкові (інформація представлена в двійковому коді).

2) По характеру навчання:

- з «вчителем» (в процесі навчання відомо набір вихідних значень);
- без «вчителя» (формує вихідні значення тільки на основі вхідних впливів), їх називають такими, що самоорганізуються.

3) По характеру налаштування синапсів:

- з фіксованими зв'язками (вагові коефіцієнти вибираються на початковому етапі виходячи з даної задачі);
- з динамічними зв'язками (в процесі навчання відбувається настройка вагових коефіцієнтів).

4) За методом навчання:

- навчання по алгоритму зворотного розповсюдження помилки;
- з конкурентним навчанням;
- з навчанням по правилу Хебба;
- з гібридним навчанням.

5) По характеру зв'язків:

– з прямими зв'язками (розповсюджується інформація тільки в одному напрямку від рівня до рівня: це різноманітні перцептронні мережі);

– із зворотнім поширенням інформації:

а) релаксаційні – циркуляція інформації відбувається доти, поки не перестають змінюватись вихідні значення НМ – це НМ Хопфілда, Хемінга;

б) багаторівневі мережі – в них відсутній процес релаксації. Це рекурентні – в них існує зворотній зв'язок між входом і виходом, вихідне значення визначається як залежність вхідних і вихідних значень на попередньому кроці; рециркуляційні – характеризуються як прямим, так і зворотнім перетворенням інформації. Навчання відбувається без «вчителя», тобто вони самоорганізуються в процесі роботи[4].

Отже, сьогодні існують сотні нейронних мереж для вирішення задач, специфічних для різних виробничих галузей. Розглянемо різні типи базових нейронних мереж в порядку зростання складності. Ці типи нейронних мереж поділяються на[5]:

1. Дрібні НМ (спільна фільтрація). Такі НМ складаються з груп перцептронів, що імітують нейронну структуру людського мозку. Дрібні НМ мають єдиний прихований шар перцептронів. Одним з поширених прикладів неглибоких НМ є спільна фільтрація. Прихований шар перцептронів буде навчений відображати подібності між сутностями для вироблення рекомендацій. Наприклад, система рекомендацій в Netflix, Amazon, YouTube Використовує версію спільної фільтрації, щоб рекомендувати свої продукти відповідно до інтересів користувачів.

2. Багатошаровий перцептрон (глибокі нейронні мережі). Нейронні мережі з більш ніж одним прихованим шаром називаються глибокими нейронними мережами. Усі наступні нейронні мережі є формою глибокої НМ, налаштованої / покращеною для вирішення проблем, специфічних для предметної галузі. В цілому вони

допомагають нам домогтися універсальності. З огляду на достатню кількість прихованих шарів нейрона, глибока НМ може наближатися, тобто вона може вирішити будь-яку складну реальну проблему. Кожна версія глибокої нейронної мережі розробляється за допомогою повністю пов'язаного шару максимального об'єднаного продукту множення матриць, який оптимізується алгоритмами зворотного поширення (Рис. 1.2).

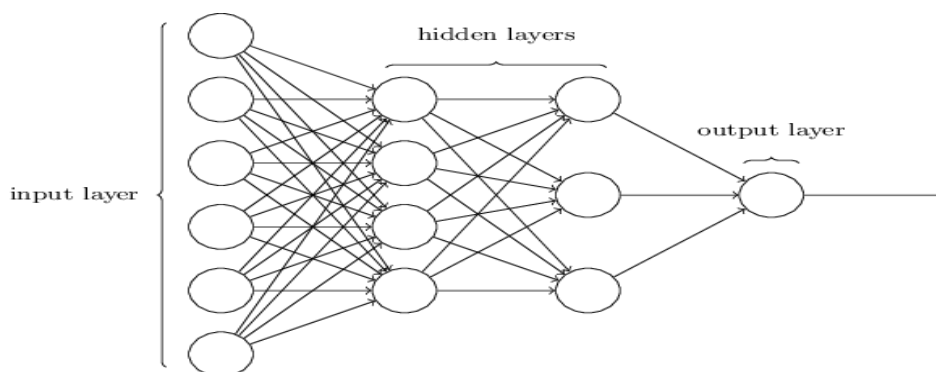


Рис. 1.2 Приклад багат шарового перцептронну

3. Згорткові нейронні мережі (ЗНМ) - це найбільш зріла форма глибоких нейронних мереж, що дозволяє отримувати найбільш точні, тобто кращі, ніж людей, результати комп'ютерного зору. ЗНМ складаються з шарів згорток, створених шляхом сканування кожного пікселя зображень в наборі даних. У міру того, як дані апроксимуються шар за шаром, ЗНМ починає розпізнавати шаблони і тим самим розпізнавати об'єкти на зображеннях. Ці об'єкти широко використовуються в різних додатках для ідентифікації, класифікації. Недавні практики, такі як трансферне навчання в ЗНМ, привели до значних поліпшень в неточності моделей. Google Translator і Google Lens - найсучасніші приклади ЗНМ. Застосування ЗНМ є експоненціальним, оскільки вони навіть використовуються для вирішення проблем, які в першу чергу не пов'язані з комп'ютерним зором (Рис. 1.3).

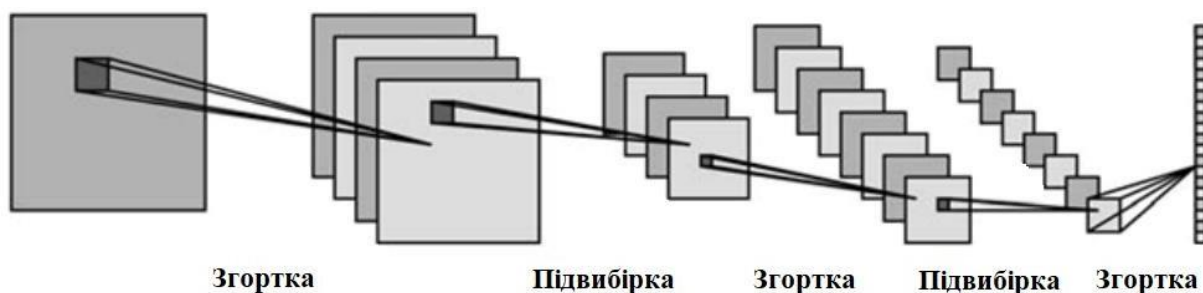


Рис. 1.3 Архітектура згорткової нейронної мережі

4. Рекурентна нейронна мережа (РНМ). РНМ - це новітня форма глибоких нейронних мереж для вирішення проблем в нейрон-лінгвістичному програмуванні. Простіше кажучи, РНМ передають вихідні дані кількох прихованих шарів назад на вхідний рівень для агрегування і перенесення наближення на наступну ітерацію (епоху) вхідного набору даних.

Це також допомагає моделі самостійно навчатись і в деякій мірі швидше коригувати прогнози. Такі моделі дуже корисні для розуміння семантики тексту в операціях нейрон-лінгвістичного програмування. Існують різні варіанти РНМ, такі як Long Short Term Memory, Gated Recurrent Unit. На наведеній нижче діаграмі активація з  $h_1$  і  $h_2$  подається з входами  $x_2$  і  $x_3$  відповідно (Рис. 1.4).

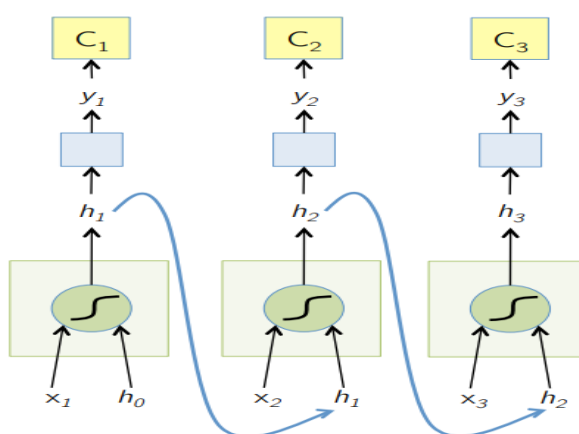


Рис. 1.4 Діаграма робочого процесу РНМ

5. Мережі довготривалої короткострокової пам'яті, що розроблені спеціально для вирішення проблеми зникаючих градієнтів в РНМ. Зникаючі градієнти

трапляються з великими нейронними мережами, де градієнти функцій втрат мають тенденцію наближатися до нуля, змушуючи НМ припинятися для навчання. Такі мережі вирішують цю проблему, запобігаючи функції активації в своїх повторюваних компонентах і не змінюючи збережені значення (Рис. 1.5).

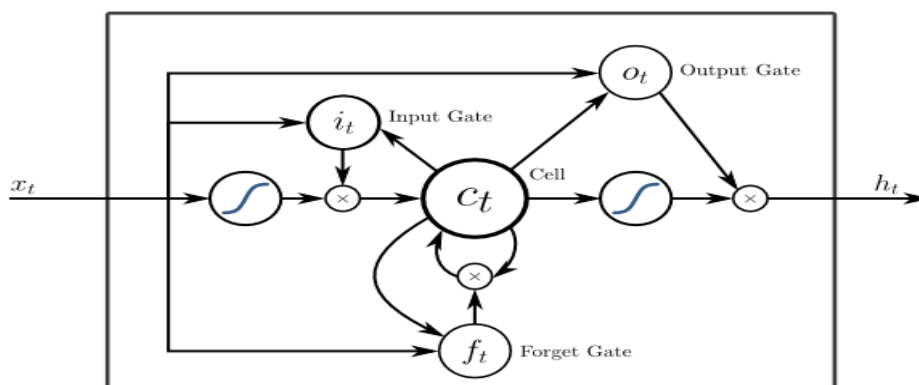


Рис. 1.5 Приклад мережі довготривалої короткострокової пам'яті

6. Нейронні мережі на основі уваги. Моделі уваги повільно беруть на себе навіть нові РНМ на практиці. Моделі уваги будуються шляхом зосередження на частини підмножини інформації, яку їм дають, тим самим усуваючи переважна кількість фонові інформації, яка не потрібна для поточного завдання. Моделі уваги побудовані на поєднанні принципів застосування уваги і підгонки шляхом поширення такого принципу. Ієрархічно складена модель множинної уваги називається трансформатором. Ці перетворювачі більш ефективні для паралельного запуску стеків. Розподіл уваги стає дуже потужним при використанні з ЗНМ / РНМ і може створювати текстовий опис зображення, як показано нижче (Рис. 1.6).



Рис. 1.6 Приклад текстового опису зображення за допомогою ШНМ.

7. Генеративна змагальна мережа. Хоча моделі глибокого навчання забезпечують найсучасніші результати, їх можуть обдурити набагато розумніші люди, додавши шум до реальних даних. Генеративна змагальна мережа - це остання розробка в галузі глибокого навчання для вирішення таких сценаріїв. Такі мережі використовують навчання без учителя, коли глибокі НМ навчаються з використанням даних, створених за допомогою моделі П, разом з фактичним набором даних, щоб підвищити точність і ефективність моделі. Ці змагальні дані в основному використовуються, щоб обдурити дискримінаційну модель, щоб побудувати оптимальну модель. Результуюча модель має тенденцію бути кращим наближенням, ніж може подолати такий шум (Рис. 1.7).

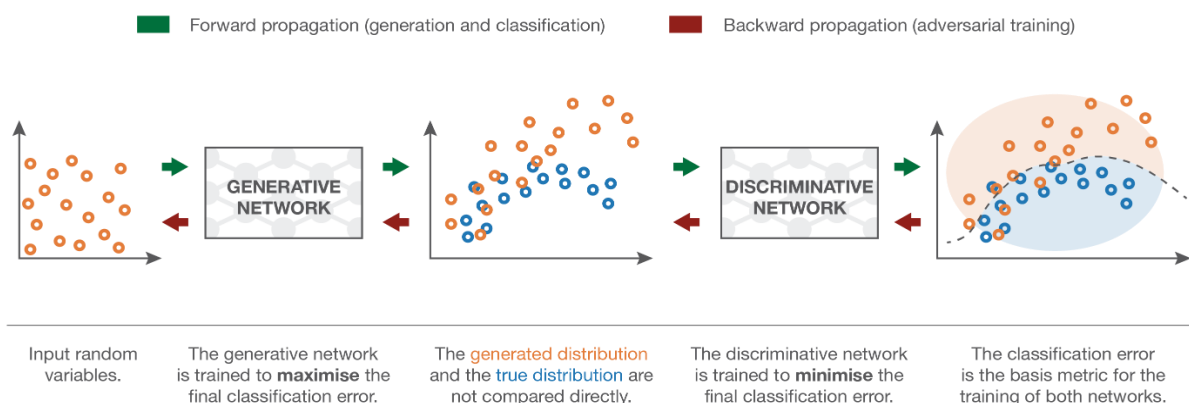


Рис. 1.7 Приклад генеративної змагальної мережі

Отже, глибокі нейронні мережі розширюють межі ПК. Вони не обмежуються тільки класифікацією ЗНМ та РНМ або прогнозами та навіть здатні генерувати дані. Ці дані можуть варіюватися від гарної форми мистецтва до спірних глибинних підробок, але кожен день вони перевершують людей по завданню. Отже, важливим завданням є усвідомлення впливу ШНМ та створення ефективної моделі ШНМ для конкретних задач.

## 1.4 Алгоритми навчання нейронних мереж

НМ навчаються на оброблених прикладах, кожен з яких вже включає відомі вхідні дані і результат, формуючи між ними асоціації, з врахуванням ймовірностей,

які знаходяться в структурах даних самої мережі. З даного прикладу видно, що навчання нейромережі відбувається шляхом знаходження різниці між обробленим висновком, часто пророкований, і цільовим висновком помилки. НМ коригує свої нові асоціації відповідно до правил навчання, які прийняті і з використанням значення помилки [6].

Послідовні налаштування НМ приведе до того, що буде видаватись результат, який ставатиме все більш точним та близьким до правильного. Після того, як пройде достатня кількості налаштувань, навчання може бути закінчене за певними критеріями.

Як правило, системи такого типу вчать виконувати завдання, аналізуючи приклади, без запрограмування конкретних правил. При розпізнаванні зображень, наприклад, вони можуть ідентифікувати фотографії, що містять кішок, шляхом аналізу допоміжних зображень, які були власноруч позначені людиною як «кішка» або «без кішок», і використовувати результати для опізнювання кішок на інших зображеннях. НМ проводять свої операції, не знаючи заздалегідь про кішок, про їх розпізнавальні ознаки, наприклад, що у них вуса, шерсть, хвости, лапки і котячі очі. Замість цього НМ автоматично ідентифікує сгенеровані характеристики із обсягу даних, які вона обробила.

Нейрони розподіляються на кілька рівнів, частіше всього при глибокому навчанні. Нейрони кожного шару з'єднуються тільки з нейронами, що знаходяться на наступних шарах. Шар нейронів, який отримує дані на опрацювання, є вхідним шаром. Шар нейронів, який дає кінцевий результат, – це вихідний шар. Між ними може бути нуль прихованих шарів, а може бути і безліч. Також можна використовувати одношарові і не шарові мережі [7].

З'єднання між двома рівнями може відбуватися за допомогою кількох схем. Рівні можуть бути пов'язані цілком, при цьому кожний нейрон першого шару зв'язується з кожним нейроном наступного шару. Нейрони можуть об'єднуватися в групи, декілька нейронів в одному шарі з'єднується з одним нейроном в наступному, таким чином зменшуючи загальну кількість нейронів в новому шарі. Такі зв'язки

утворюють орієнтований ациклічний граф і називаються мережами прямого поширення.

Також, варто зазначити, що мережі, в яких нейрони зв'язані з одним або декількома нейронами з попередніх шарів, мають назву - повторювані мережі.

Навчання – це пристосування НМ для оптимального вирішення завдання з урахуванням більшості вибіркового спостережень. Для підвищення точності результату, навчання нейромережі включає в себе налаштування ваг і порогових значень мережі. Досягається це за рахунок максимального зменшення спостережуваних помилок. Навчання закінчується тоді, коли збільшення додаткових спостережень не знижує частоту похибки. Навіть після всіх налаштувань і навчальних процесів частота помилок зазвичай не досягає абсолютного 0. Якщо після процесу навчання НМ допускає більшу за допустиму кількість помилок, потребується перепроєктування мережі.

Нижче наведено традиційну схему алгоритму навчання ШНМ (Рис. 1.8).



Рис. 1.8 Схема навчання НМ

На практиці це досягається шляхом оцінювання функції витрат, яка періодично вимірюється під час навчання. Поки обсяг витрат виробництва продовжує зменшуватись, навчання триває. Вихідними даними є числа, тому, коли коефіцієнт помилки малий, різниця між результатом – “напевно кішка” і правильним варіантом - “кішка”, невелика. Навчання НМ з кожним кроком відчутно зменшує загальну різницю між спостереженнями. Більшість існуючих моделей навчання можна розглядати як прямий додаток теорії оптимізації і статистичної оцінки [8].

Швидкість навчання визначає розмір коригувальних кроків, які модель робить для коригування помилок в кожному спостереженні. Висока швидкість навчання скорочує час навчання, але з меншою кінцевою точністю, тоді як більш низька швидкість навчання займає більше часу, але з потенціалом для більшої точності.

Оптимізація, така як Quicksprogr, в першу чергу націлена на прискорення мінімізації помилок, в той час як інші поліпшення в основному спрямовані на підвищення надійності.

Доступні два режими навчання: стохастичний і пакетний. У стохастичному навчанні кожен вхід створює коригування ваги. При пакетному навчанні ваги коректуються на основі пакету вхідних даних, накопичуючи помилки по пакету. Стохастичне навчання вносить «шум» в процес, використовуючи локальний градієнт, розрахований з однієї точки даних; це знижує ймовірність застрягання мережі в локальних мінімумах.

Однак пакетне навчання зазвичай дає більш швидкий і стабільний спуск до локального мінімуму, оскільки кожне оновлення виконується в напрямку середньої помилки пакета. Поширеним компромісом є використання «міні-партій», невеликих партій з вибірками в кожній партії, обраними стохастично з усього набору даних [9].

Отже, варто констатувати, що всі дані, яку мережа має в якості задачі, утримується в певній сукупності прикладів. Через це якість навчання НМ безпосередньо залежить від числа прикладів у вибірці навчання, а також від того, наскільки повно дані приклади характеризують конкретне завдання. Так, наприклад, немає сенсу застосовувати НМ для прогнозів фінансової кризи, якщо в навчальній вибірці таких криз не існує. На думку дослідників, для повноцінного тренування потрібно мати біля сотні прикладів. Таким чином, навчання мережі – це досить непростий та наукомісткий процес. Алгоритми навчання здебільшого мають різні властивості та налаштування, для управління якими необхідно розуміти їхній вплив.

### **1.5 Постановка задачі дослідження**

Метою роботи є поліпшення якості розпізнавання номерних знаків автомобілів на зображенні за допомогою нейронної мережі .

Для досягнення мети необхідно розв'язати завдання:

1. Розглянути теоретичні засади навчання нейронних мереж.
2. Дослідити класифікацію нейронних мереж.
3. Визначити алгоритми навчання нейронних мереж.

4. Проаналізувати особливості застосування нейромережного розпізнавання об'єкта на зображенні.
5. Розробити алгоритм ШНМ для розпізнавання номерних знаків автомобілів в системі відеоспостереження.
6. Реалізувати програмне забезпечення для розпізнавання номерних знаків автомобілів на зображенні.

### **Висновки до розділу 1**

Підсумовуючи перший розділ, можна зробити наступні висновки:

1. Проведено аналіз основних принципів області розпізнавання об'єктів.
2. Визначено теоретичні засади навчання нейронних мереж, окреслено поняття «нейронна мережа», з'ясовано його сутність.
3. Окреслено класифікацію НМ.
4. Досліджено алгоритми навчання нейронних мереж.
5. Визначені завдання дослідження

## РОЗДІЛ 2

# ОСОБЛИВОСТІ ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖНОГО РОЗПІЗНАВАННЯ ОБ'ЄКТА НА ЗОБРАЖЕННІ

### 2.1 Специфіка застосування нейронних мереж

Нейронні мережі не є універсальними для вирішення всіх обчислювальних проблем. Комп'ютери та обчислювальні способи в більшості випадків є ідеальними для реалізації. Комп'ютерні технології перевершують людину в питаннях математичних обчислень та символічних обчислень. Проте людина здатна без особливих зусиль вирішувати доволі складні завдання сприйняття зовнішньої інформації з такою швидкістю та точністю, з якою жоден ПК у на її фоні буде виглядати тугодумом.

Володіючи людської здатністю вирішувати проблеми і застосовувати цю навичку до величезних наборів даних, НМ мають такі потужними атрибутами:

- адаптивне навчання: як і люди, НМ моделюють нелінійні і складні відносини і спираються на попередні знання. Наприклад, програмне забезпечення використовує адаптивне навчання для навчання математики та мовним мистецтвам.

- самоорганізація: здатність кластеризувати і класифікувати величезні обсяги даних робить нейронні мережі унікальними для організації складних візуальних проблем, що виникають при аналізі медичних зображень.

- робота в реальному часі: НМ можуть (іноді) надавати відповіді в реальному часі, як у випадку з безпілотними автомобілями і навігацією дронів.

- прогноз: здатність НМ прогнозувати на основі моделей має широкий спектр додатків, в тому числі для погоди і дорожнього руху.

- відмовостійкість: коли значна частина мережі втрачена або відсутній, нейронні мережі можуть заповнити прогалини. Ця здатність особливо корисна при освоєнні космосу, де завжди можлива відмова електронних пристроїв.

Розглянемо детальніше галузі використання ШНМ для розв'язання емпіричних завдань.

Нейронні мережі дуже цінні, тому що вони можуть виконувати завдання з аналізу даних, зберігаючи при цьому всі свої інші атрибути. Ось важливі завдання, які виконують нейронні мережі [3]:

- Класифікація: мережеві мережі організують шаблони або набори даних в заздалегідь визначені класи.

- Прогноз: вони виробляють очікуваний результат на основі заданих вхідних даних.

- Кластеризація: вони ідентифікують унікальну особливість даних і класифікують її, не знаючи попередніх даних.

- Зв'язування: ви можете навчити НМ «запам'ятовувати» шаблони. Коли ви покажете незнайому версію патерну, мережа пов'яже її з найбільш порівнянної версією в своїй пам'яті і повертається до останньої. Нейронні мережі мають фундаментальне значення для глибокого навчання, надійного набору технік НМ, які дозволяють вирішувати абстрактні проблеми, такі як біоінформатика, розробка ліків, фільтрація соціальних мереж і переклад на природну мову.

- Глибоке навчання - це те місце, де ми вирішуємо найскладніші завдання в науці і техніці, включаючи передову робототехніку. У міру того як НМ стають розумнішими і швидше, ми щодня вдосконалюємося.

Ось список інших додатків для проектування нейронних мереж, які в даний час використовуються в різних галузях:

- Аерокосмічна промисловість: детектори і моделювання несправностей компонентів літака, системи управління літаком, високопродуктивне автопілотування і моделювання траєкторії польоту.

- Автомобільна промисловість: вдосконалені системи наведення, розробка силових агрегатів, віртуальних датчиків і аналізаторів гарантійного обслуговування. Електроніка: аналіз відмов мікросхеми, схеми мікросхем, машинне зір, нелінійне моделювання, прогнозування кодової послідовності, управління процесом і синтез голосу.

– Виробництво: аналіз конструкції хімічної продукції, динамічне моделювання систем хімічних процесів, управління процесами, діагностика процесів і машин, проектування і аналіз продукції, прогнозування якості паперу, торги по проектам, планування і управління, аналіз якості комп'ютерних мікросхем, системи візуального контролю якості та аналіз якості зварювання.

– Механіка: моніторинг стану, моделювання систем і управління.

– Робототехніка: роботи-навантажувачі, контролери маніпуляторів, системи управління траєкторією і системи технічного зору.

– Телекомунікації: управління мережею банкоматів, автоматизовані інформаційні послуги, системи обробки платежів клієнтів, стиснення даних, еквалайзери, усунення несправностей, розпізнавання рукописного введення, проектування мережі, управління, маршрутизація і контроль, моніторинг мережі, переклад розмовної мови в реальному часі і розпізнавання образів (особи, об'єкти, відбитки пальців, семантичний аналіз, перевірка орфографії, обробка сигналів і розпізнавання мови).

Крім того, існують додаткові поточні приклади бізнес-додатків ШНМ. Серед них [10]:

– Банківська справа: виснаження кредитних карт, оцінка кредитних і кредитних заявок, оцінка шахрайства і ризиків, а також прострочення по кредитах.

– Бізнес-аналітика: моделювання поведінки клієнтів, сегментація клієнтів, схильність до шахрайства, дослідження ринку, структура ринку, структура ринку і моделі відсіву, дефолту, покупок і продовжень.

– Захист: боротьба з тероризмом, розпізнавання осіб, виділення функцій, придушення шуму, розпізнавання об'єктів, датчики, гідролокатор, радар і обробка сигналів зображення, ідентифікація сигналу / зображення, відстеження цілі і управління зброєю.

– Освіта: програмне забезпечення для адаптивного навчання, динамічне прогнозування, аналіз і прогнозування системи освіти, моделювання успішності учнів і профілювання особистості.

- Фінанси: рейтинги корпоративних облігацій, корпоративний фінансовий аналіз, аналіз використання кредитної лінії, прогнозування курсів валют, консультування по кредитах, перевірка іпотечних кредитів, оцінка нерухомості і торгівля портфелями.

- Медицина: аналіз ракових клітин, аналіз ЕКГ і ЕЕГ, консультації по тестуванню в відділенні невідкладної допомоги, скорочення витрат і підвищення якості лікарняних систем, оптимізація процесу трансплантації і дизайн протеза.

- Цінні папери: автоматичний рейтинг облігацій, аналіз ринку і консультаційні системи з торгівлі акціями.

- Транспорт: системи маршрутизації, системи діагностики гальм вантажівок і планування транспортних засобів.

Перспективами розробки нейромережевих технологій можуть бути наступні сфери [11]:

- Інтеграція нечіткої логіки: нечітка логіка розпізнає більше, ніж просто справжні і несправжні значення - вона бере до уваги поняття, які є відносними, наприклад, щось, іноді і зазвичай. Нечітка логіка і нейронні мережі інтегровані для використання в самих різних областях, таких як відбір кандидатів на вакансії, автомобільна інженерія, управління будівельними кранами і моніторинг глаукоми. Нечітка логіка стане важливою функцією в майбутніх додатках нейронних мереж.

- Імпульсні нейронні мережі: нещодавно дані нейробіологічних експериментів прояснили, що біологічні нейронні мережі ссавців з'єднуються і взаємодіють за допомогою пульсації і використовують синхронізацію імпульсів для передачі інформації і виконання обчислень. Це визнання прискорило значні дослідження, включаючи теоретичний аналіз, розробку моделей, Нейробіологічні моделювання та розгортання обладнання, спрямованих на те, щоб зробити обчислення ще більш схожими на те, як функціонує наш мозок.

- Спеціалізоване обладнання: в даний час спостерігається бурхливий ріст розробки обладнання, яке прискорить і в кінцевому підсумку знизить вартість нейронних мереж, машинного навчання та глибокого навчання. Відомі компанії і

стартапи прагнуть розробити поліпшені мікросхеми та блоки графічної обробки, але справжня новина - це швидкий розвиток блоків обробки нейронних мереж та іншого обладнання для II, разом іменуються нейросінаптичною архітектурою. Нейросінаптичні чіпи мають фундаментальне значення для розвитку III, тому що вони більше схожі на біологічний мозок, ніж на ядро традиційного комп'ютера. Зі свого технологією Brain Power компанія IBM є лідером в розробці нейросінаптичних чіпів. На відміну від стандартних мікросхем, які працюють безперервно, мікросхеми Brain Power управляються подіями і працюють в міру необхідності. Технологія об'єднує пам'ять, обчислення і зв'язок.

– Поліпшення існуючих технологій: завдяки новому програмному забезпеченню та обладнанню, а також поточним технологій нейронних мереж і збільшеною обчислювальною потужністю нейросінаптичних архітектур нейронні мережі тільки почали показувати, на що вони здатні. Безліч бізнес-додатків більш швидких, дешевих і більш людських методів вирішення проблем і поліпшених методів навчання дуже прибуткові.

– Робототехніка: було безліч прогнозів щодо роботів, які зможуть відчувати себе, як ми, бачити як ми, і робити прогнози щодо світу навколо них.

Перш ніж ці системи зможуть по-справжньому мислити рухомим, непорушним чином, має відбутися так багато речей. Один з найважливіших факторів - це здатність встановлювати і діяти відповідно до самовизначення цінностями в режимі реального часу, що ми, люди, робимо тисячі разів в день. Без цього ці системи будуть давати збій кожен раз, коли умови виходять за межі заздалегідь заданій області.

Отже, нові можливості обчислень потребують нових умінь розробника поза межами класичних обчислень. Спочатку обчислення були виключно апаратними та програмісти зробили їх функціональними. Згодом, були фахівці з ПЗ: системні інженери, фахівці з баз даних та проектувальники. Зараз вже існують нейронні архітектори. Новий кадровий склад спеціалістів повинен мати відповідні кваліфікаційні вміння, яка має бути вищою за їх попередників. Наприклад, вони мають володіти знаннями відносно статистики для вибору та оцінки навчальної та

тестової множини. Зазвичай, при проектуванні ефективних ШНМ, важливим для сучасних програмістів є логічне мислення та досвід.

## **2.2 Огляд програмних засобів, що використовують нейронні мережі**

На сьогоднішній день існує понад двохсот нейропакетів, які випускаються багатьма компаніями та окремими науковцями та допомагають конструювати, навчати і застосовувати ШНМ для вирішення практичних задач. Трудомісткість проектування НМ скорочується при використанні готового нейромережевого ПЗ. Нижче перерахуємо найбільш поширені програми для ШНМ [12]:

1. Neural Designer – це настільний додаток для інтелектуального аналізу даних, в якому використовуються нейронні мережі, які є парадигмою машинного навчання. Neural Designer володіє більшістю передових методів підготовки даних, машинного навчання та розгортання моделей. Його візуальний графічний користувацький інтерфейс забезпечує вичерпні та наочні результати без необхідності написання коду або збірки блоків. Програмне забезпечення реалізує багатоядерну обробку для аналізу великих обсягів даних за менший час.

2. Neuroph – це проект з відкритим вихідним кодом, розміщений на SourceForge під ліцензією Apache. Це бібліотека для створення нейронних мереж і використання машинного навчання. Neuroph - це полегшена структура нейронної мережі Java для розробки загальних архітектур нейронних мереж. Neuroph містить красивий графічний редактор нейронної мережі для швидкого створення компонентів нейронної мережі Java. Програмне забезпечення спрощує розробку нейронної мережі, надаючи бібліотеку нейронної мережі Java і інструмент з графічним інтерфейсом користувача, який підтримує створення, навчання і збереження нейронних мереж.

3. Darknet – це фреймворк нейронної мережі з відкритим вихідним кодом, написаний на C і CUDA і підтримує обчислення CPU і GPU. Це ЗНМ, що складається з дев'ятнадцяти шарів. Попередньо навчена мережа може класифікувати зображення по 1000 категоріям об'єктів, таким як клавіатура, миша, олівець і багато тварин. У

результаті мережа навчилася відображати великі можливості для широкого діапазону зображень.

4. Keras – це бібліотека глибокого навчання для Theano і TensorFlow. Бібліотека нейронних мереж високого рівня написана на Python і здатна працювати поверх обох додатків. Keras – це API, розроблений для людей, а не для машин. Програмне забезпечення слід передовим методам зниження когнітивної навантаження. Він пропонує послідовні і прості API-інтерфейси і зводить до мінімуму кількість дій користувача, необхідних для типових випадків використання. Keras надає чіткі і корисні повідомлення про помилки, а також велику документацію і керівництва для розробників. Бібліотека глибокого навчання Keras дозволяє легко і швидко створювати прототипи завдяки повній модульності, мінімалізму і розширюваності. Він підтримує згорткові нейронні мережі та рекурентні мережі, а також їх комбінації.

5. NeuroSolutions – це середовище розробки програмного забезпечення для нейронних мереж, розроблена NeuroDimension. Він поєднує в собі модульний інтерфейс мережевого дизайну на основі значків з реалізацією розширених процедур навчання, таких як пов'язані градієнти, Levenberg Marquardt і зворотне поширення в часі. Сімейство продуктів NeuroSolutions – це передове програмне забезпечення нейронних мереж для інтелектуального аналізу даних для створення високоточних і прогнозних моделей з використанням передових методів обробки, інтелектуального автоматичного пошуку топології нейронної мережі за допомогою передових розподілених обчислень.

Це дизайнерський інтерфейс з передовим штучним інтелектом і алгоритмами навчання з використанням інтуїтивно зрозумілих майстрів або простого у використанні інтерфейсу Excel. Програмне забезпечення надає три окремих майстра для автоматичної побудови моделей нейронних мереж:

- Менеджер даних,
- Neural Builder,
- Neural Expert.

Отже, перевагою вищеперерахованого програмного забезпечення є простота розробки та зрозумілість процесу виведення, відсутність проблемних рішень відносно редагування. В якості недоліку можна відзначити високу вартість програмного забезпечення.

### **2.3 Загальна модель згорткової нейронної мережі**

Модель глибокого навчання складається з купи обробних шарів, які можуть вивчати різні особливості даних за допомогою різних рівнів абстракції. Кількість рівнів дозволяють мережі вивчати різні особливості. Поглиблене навчання стало підходом для досягнення багатообіцяючих результатів у різних сферах, таких як розпізнавання зображень, розпізнавання мови, класифікація тем, аналіз настроїв літератури, переклад мови, розуміння природної мови, обробка сигналів, розпізнавання обличчя, прогнозування біоактивності малих молекул та ін. Існують різні архітектури глибокого навчання, такі як мережі глибоких передбачень, періодичні нейронні мережі, нейронні мережі згортки тощо.

Нейронна мережа згортки (ЗНМ), яку часто називають ConvNet, має глибоку зворотню архітектуру та має дивовижну здатність узагальнювати краще, ніж у мережах із повнозв'язаними рівнями. [13] описує ЗНМ як концепцію ієрархічних детекторів які натхнені біологічними особливостями. Він може засвоїти високо абстрактні ознаки та може ефективно ідентифікувати об'єкти [14]. Суттєві причини, чому ЗНМ розглядається вище інших класичних моделей, полягають у наступному. По-перше, ключовий інтерес для застосування ЗНМ полягає в ідеї використання концепції розподілу ваги, завдяки чому кількість параметрів, які потребують навчання, істотно зменшується, що призводить до поліпшення узагальнення. Через менші параметри, ЗНМ може тренуватися плавно і не зазнає переобладнання. По-друге, етап класифікації поєднаний із етапом вилучення ознак, в обох використовується процес навчання. По-третє, набагато важче реалізувати великі мережі, використовуючи загальні моделі ШНН, ніж впроваджувати в ЗНМ. На додаток до цього, він обчислює швидкість навчання для кожного окремого параметра.

**Загальна модель.** Типова модель ШНН має один вхідний і вихідний рівень разом з безліччю прихованих шарів[15]. Конкретний нейрон приймає вхідний вектор  $X$  і виробляє вихід  $Y$ , виконуючи на ньому якусь функцію  $F$ , представлену загальним рівнянням (1), заданим, показаним нижче.

$$F(X, W) = Y \quad (1)$$

де  $W$  позначає ваговий вектор, який представляє силу взаємозв'язку між нейронами двох сусідніх шарів. Отриманий вектор ваги тепер можна використовувати для класифікації зображень. Існує значна кількість літератури, що стосується піксельної класифікації зображень. Однак контекстна інформація, подібна до форми зображення, дає кращі результати або перевершує [16]. ЗНМ - це модель, яка привертає увагу завдяки можливості класифікації на основі контекстної інформації. Загальна модель ЗНМ описана нижче(Рис. 2.1). Загальна модель ЗНМ складається з чотирьох компонентів, а саме (а) рівня згортки, (b) шару об'єднання, (c) функції активації та (d) повністю пов'язаного шару. Функціональність кожного компонента проілюстрована(Рис. 2.1).

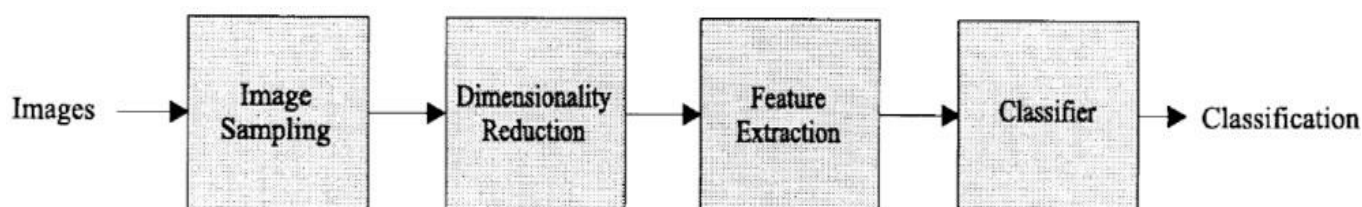


Рис. 2.1 Елементарні складові ЗНМ

**Згортковий шар(Convolution Layer).** Класифіковане зображення подається на вхідний рівень, а вихідним є передбачувана мітка класу, обчислена з використанням витягнутих із зображення характеристик. Окремий нейрон у наступному шарі з'єднаний з деякими нейронами попереднього шару, це місцеве співвідношення називається рецептивним полем. Місцеві особливості із вхідного зображення витягуються за допомогою рецептивного поля. Сприйнятливим полем нейрона, пов'язане з певною областю в попередньому шарі, утворює ваговий вектор, який залишається рівним у всіх точках площини, де площина відноситься до нейронів наступного шару. Оскільки нейрони в площині мають однакові ваги, таким чином

можна виявити подібні ознаки, що трапляються в різних місцях у вхідних даних[16]. Це показано на(Рис. 2.2).

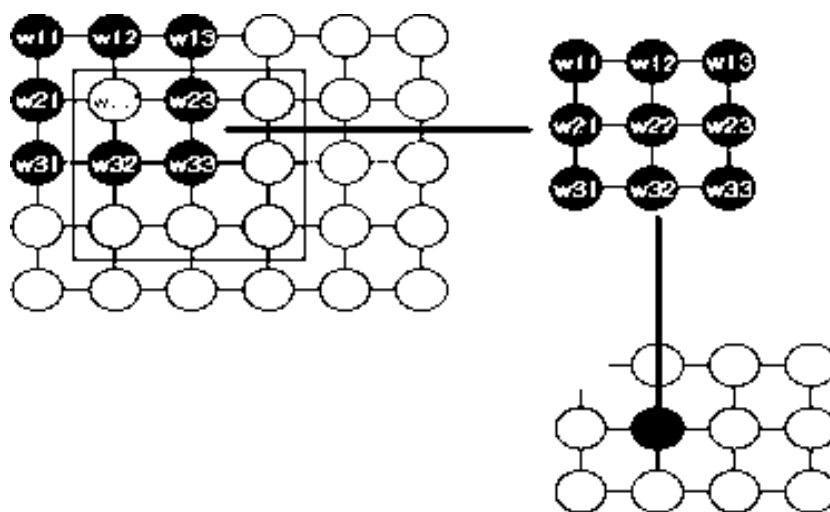


Рис. 2.2 Сприйнятливий поле конкретного нейрона в наступному шарі

Вектор ваги, також відомий як фільтр або ядро, ковзає по вхідному вектору, щоб сформувати карту функцій. Цей метод ковзання фільтра як по горизонталі, так і по вертикалі називається операцією згортки. Ця операція витягує  $N$  кількості об'єктів із вхідного зображення в один шар, що представляє різні особливості, що веде до  $N$  фільтрів та  $N$  карт функцій. Через явище локального рецептивного поля кількість параметрів, що піддаються навчання, значно зменшується. Вихід  $a_{ij}$  у наступному шарі для розташування  $(i, j)$  обчислюється після застосування операції згортки за формулою, наведеною в [16], як показано нижче:

$$a_{ij} = \sigma((W * X)_{ij} + b) \quad (2)$$

де  $X$  - вхід, що надається шару,  $W$  - фільтр або ядро, яке ковзає по входу,  $b$  - зміщення,  $*$  що представляє операцію згортки, а  $\sigma$  - нелінійність, введена в мережу.

**Об'єднання шару (Pooling Layer).** Точне розташування ознаки стає менш значущим після того, як її було виявлено. Отже, за згортковим шаром слідує шар об'єднання або субвибірки. Головною перевагою використання методики об'єднання є те, що вона значно зменшує кількість параметрів, що піддаються навчання, і вводить незмінність трансляції. Для виконання операції об'єднання вибирається

вікно, і вхідні елементи, що лежать у цьому вікні, пропускаються через функцію об'єднання, як показано на (Рис. 2.3).

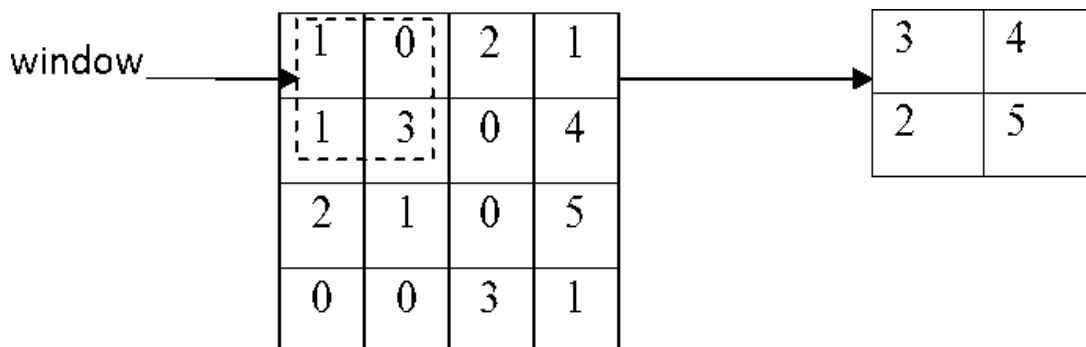


Рис. 2.3 Операція об'єднання, виконана шляхом вибору вікна 2 x 2

Функція об'єднання генерує ще один вихідний вектор. Існує декілька методів об'єднання, таких як середнє об'єднання та максимальне об'єднання, з яких максимальне об'єднання є найбільш часто використовуваним методом, який дуже суттєво зменшує розмір карти[15]. Під час обчислення помилка не поширюється назад до виграшної одиниці, оскільки вона не бере участі в прямому потоці.

**Повністю підключений шар(Fully Connected Layer).** Повністю підключений шар схожий на повністю підключену мережу в звичайних моделях. Вихід першої фази (включає згортку та об'єднання повторно) подається у повністю з'єднаний шар, а точковий добуток вагового вектора та вхідного вектора обчислюється для отримання кінцевого результату[17]. Градієнтний спуск, також відомий як пакетний режим навчання або автономний алгоритм, зменшує функцію витрат, оцінюючи вартість по всьому набору навчальних даних, і оновлює параметри лише через одну епоху, де епоха відповідає обходу всього набору даних. Це дає загальні мінімуми, але якщо розмір набору навчальних даних великий, час, необхідний для навчання мережі, істотно збільшується. Цей підхід до зменшення функції витрат був замінений стохастичним градієнтним спуском.

**Функція активації(Activation Function).** Існує величезна кількість літератури, яка використовує функцію активації сигмоїдної форми в звичайних алгоритмах машинного навчання. Для того, щоб запровадити нелінійність, використання випрямленого лінійного блоку (ReLU) виявилось кращим, ніж перший, через два

основних фактори. По-перше, розрахунок часткової похідної ReLU є простим[17]. По-друге, незважаючи на те, що час навчання вважається одним із факторів, насичувальні нелінійності, такі як сигмоїдна, представлена  $f(x) = (1 + e^{-x})^{-1}$ , повільніші, ніж ненасичувальні нелінійності, такі як ReLU, представлена[18].

$$f(x) = \max\{0, x\}$$

По-третє, ReLU не дозволяє градієнтам зникати. Але ефективність ReLU погіршується, коли через мережу протікає великий градієнт, і оновлення ваги призводить до того, що нейрон не активується, що призводить до проблеми з помиранням ReLU, що є значною проблемою, яка часто виникає. Цю проблему можна вирішити за допомогою Leaky ReLU, якщо  $x > 0$ , функція активується як  $f(x) = x$ , а якщо  $x < 0$ , функція активується як  $\alpha x$ , де  $\alpha$  - мала константа.

**Алгоритми навчання**, які часто називають алгоритмами оптимізації, приносять користь мережі шляхом мінімізації цільової функції (часто називається функцією втрат  $E(x)$ ), залежно від різних параметрів, що підлягають вивченню, таких як вага, упередження тощо. Алгоритми оптимізації переважно можна розділити на дві категорії, тобто оптимізацію першого порядку. алгоритми та алгоритми оптимізації другого порядку. Оптимізація першого порядку включає обчислення градієнта, представленого матрицею Якобіана, широко застосовуваною методикою є градієнтний спуск. Існує ряд варіантів градієнтного спуску, таких як міні-пакетний градієнтний спуск та стохастичний градієнтний спуск. З метою покращення результатів були зроблені вдосконалення у таких варіантах, як введення імпульсу, Адаград, АдаДельта .. Тоді як оптимізація другого порядку включає похідну другого порядку, представлену матрицею Гесія. Однією з таких методик є Адамова оптимізація. Градієнтний спуск та оптимізація Адама були коротко пояснені в наступному розділі

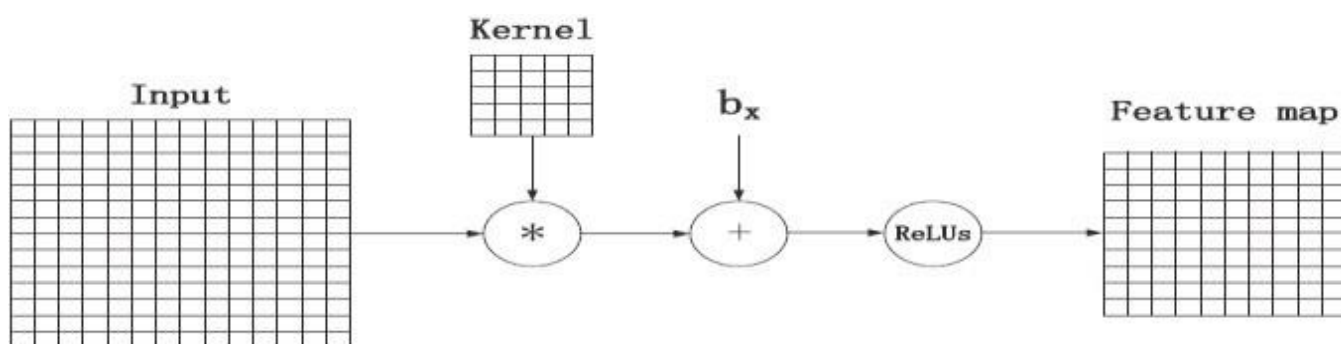
**Градієнтний спуск(Gradient Descent).** Під час навчання фільтрів зворотне розповсюдження помилок - це механізм, який використовується для модифікації попередньо ініціалізованих параметрів мережі для досягнення оптимізованих параметрів мережі, які можуть давати результати, близькі до цільових результатів. У

ЗНМ така мережа може бути досягнута за допомогою зворотного розповсюдження помилок. Оскільки загальна мережа є мережею прямого передавання, процес починається з обчислення виходів на кожному шарі по одному і обчислення компонента помилки, внесеного на останньому шарі. Тепер, щоб отримати оптимізовану мережу, обчислені градієнти зворотне розповсюдження. Виконуйте ті самі дії, поки не спостерігається ефективність. Спочатку вхідний вектор подається в мережу. Тепер виконайте операцію згортки на вхідному векторі, використовуючи рівняння 3, аналогічне та детальне рівняння зображено у рівнянні 4.

$$C_q^l = \left( \sum_{p=1}^n S_p^{l-1} * k_{p,q}^l + b_q^l \right) \quad (3)$$

$$C_q^l = \left( \sum_{p=1}^n \sum_{u=-x}^x \sum_{v=-x}^x S_p^{l-1}(i-u, j-v) \cdot k_{p,q}^l(u, v) + b_q^l \right) \quad (4)$$

де  $n$  означає кількість карт об'єктів в останньому шарі,  $p$  і  $q$  позначає індекси карт об'єктів поточного шару та попереднього шару відповідно,  $\phi$  позначає функцію активації, застосовану, наприклад, ReLU та сигмоїд,  $l$  позначає шар,  $*$  позначає операцію згортання,  $b$  та  $x$  представляють зміщення та розмір фільтра відповідно. Спочатку  $S_p^0$  являє собою вхідне зображення, на якому повинна бути виконана перша згортка, а  $S_p^1$  являє собою вхід, на якому повинна бути виконана друга згортка, яке можна отримати після застосування об'єднання на  $S_p^0$ .



## Conclusion

Рис. 2.4 Операція згортки на вхідному зображенні 14 x 14 шляхом ковзання ядра 5 x 5, що дає карту функцій 10 x 10

Основною перевагою глибокого навчання в порівнянні зі звичайною технікою машинного навчання є те, що він може самостійно виявляти відповідні особливості у великих розмірах даних порівняно з неглибокими мережами. Існує достатня кількість літератури про різні техніки глибокого навчання, такі як періодичні нейронні мережі, мережі глибоких переконань та ЗНМ. Це дослідження кинуло світло на базове розуміння ЗНМ, яке є глибоким навчальним підходом для вирішення багатьох складних проблем. У цьому дослідженні описана загальна модель, різні архітектури та два важливі алгоритми навчання ЗНМ. ЗНМ став відомим методом, що використовується для класифікації на основі контекстної інформації. Він має величезну здатність засвоювати контекстуальні особливості і тим самим долає проблеми, пов'язані з класифікацією пікселів. Це значно зменшує кількість параметрів. ЗНМ широко використовується для класифікації в області дистанційного зондування [16], завдання розпізнавання фронту океану, даних з високою роздільною здатністю, розпізнавання дорожніх знаків, звукової сцени, сегментації зображень мозку MR. Це дослідження забезпечить широке розуміння дослідникам, які хочуть займатися у цій галузі. Це діятиме як засіб для учнів, дослідників та тих, хто зацікавлений у цій галузі.

## **Висновки до розділу 2**

Підсумовуючи другий розділ, можемо зробити такі висновки:

1. Розглянуто галузі застосування нейронних мереж для розв'язання практичних задач, а також деталізовано специфіку застосування нейронних мереж під час:

- розпізнавання образів і класифікації;
- кластеризації;
- прогнозування і апроксимації;
- рішення завдань при невідомих закономірностях;
- прийняття рішень і керування;
- стиску даних і асоціативної пам'яті;

– оптимізації;

2. Проведено системний огляд програмних засобів, що використовують нейронні мережі.

3. Здійснено опис загальної моделі ЗНМ та адаптацію методів сегментації до умов задачі.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ НЕЙРОМЕРЕЖНОГО МОДУЛЮ РОЗПІЗНАВАННЯ СИМВОЛІВ (НА ПРИКЛАДІ НОМЕРНИХ ЗНАКІВ АВТОМОБІЛІВ )

#### 3.1 Обґрунтування вибору середовища програмної реалізації

В даній роботі для написання програми було обрано мову програмування Python, середовище розробки PyCharm на основі IntelliJ IDEA, модель для сегментації об'єкта YOLOv5, детектор тексту CRAFT, та візуальних інтерфейс tkinter.

##### 3.1.1 Мова програмування Python

Python – високорівнева мова програмування загального призначення з динамічною строгою типізацією і автоматичним управлінням пам'яттю, орієнтований на підвищення продуктивності розробника, читання коду і його якості, а також на забезпечення переносимості написаних на ньому програм. Мова є повністю об'єктно–орієнтованим – все є об'єктами. Синтаксис ядра мови мінімалістичний, за рахунок чого на практиці рідко виникає необхідність звертатися до документації. Сама ж мова відома як інтерпретуєма і використовується в тому числі для написання скриптів.

Недоліками мови є часто більш низька швидкість роботи і більш високе споживання пам'яті написаних на ньому програм в порівнянні з аналогічним кодом, написаним на компільованих мовах, таких як C або C ++. Але вчені не є обов'язково програмістами і можуть не знати складного синтаксису C, C ++, Java. А Python – це мова програмування з низьким порогом входу. Ніхто не пише з нуля, не витрачає час – просто використовують вже написані іншими бібліотеки. В результаті навколо Python склалася спільнота по розробці і розвитку НМ.

Також НМ – це, як правило, програми не дуже великі за обсягом, але які доводиться часто змінювати, щоб підібрати оптимальну архітектуру, доопрацювання

даних та інше. Тому проблем з legacy кодом майже немає. Якщо є вимоги до швидкості розробки, то Python оптимальніший, ніж C++ або Java.

### 3.1.2 Середовище розробки PyCharm на основі IntelliJ IDEA

PyCharm – це крос-платформна середовище розробки, яка сумісна з Windows, macOS, Linux. Є інтегрованою середовищем розробки для мови програмування Python. Надає засоби для аналізу коду, графічний відладчик, інструмент для запуску юніт-тестів і підтримує веб-розробку на Django. PyCharm розроблена компанією JetBrains на основі IntelliJ IDEA.

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

Серед з широким набором інтегрованих інструментів для рефакторінгу, 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.

IntelliJ IDEA надає інтегрований інструментарій для розробки графічного інтерфейсу користувача. Серед інших можливостей, середа добре сумісна з багатьма популярними вільними інструментами розробників, такими як CVS, Subversion, Apache Ant, Maven і JUnit.

### 3.1.3 Модель YOLOv5

YOLO (You Only Look Once) - це сімейство моделей, які стали популярні завдяки своїй малій вазі та якості прогнозів. Такі характеристики дозволяють використовувати YOLO для задач розпізнавання об'єктів в реальному часі і на портативних пристроях.

Реліз YOLOv5 включається в себе 5 моделей різних розмірів: YOLOv5s (найменша), YOLOv5m, YOLOv5l, YOLOv5x (найбільша).

YOLOv5 - це перша модель з сімейства YOLO, яку писали на PyTorch. Попередні моделі були написані на Darknet, фреймворку творця архітектури. Darknet програє PyTorch в контексті продуктивності і можливостям конфігурації моделей. Імплементация моделі на PyTorch дозволяє користуватися перевагами фреймворку: підтримка і Деплой моделі.

У Colab ноутбуці з Tesla P100 YOLOv5 модель видає передбачення на інференснє зі швидкістю 0.007 секунд за зображення. Це рівноцінно 140 кадрів в секунду. Для порівняння YOLOv4 працює зі швидкістю 50 кадрів в секунду.

За результатами тестування на датасета VCCD, модель видає передбачення з 0.895 mAP після навчання за 100 епох. При цьому ваги моделі важать близько 27 мегабайт. У YOLOv4 файл з вагами важить 244 мегабайта. YOLOv5 менше, ніж YOLOv4, приблизно на 90%.

Ця модель є доволі новою і не повністю дослідженою, але перші результати є дуже обнадійливими. Вона набагато швидша і легша за своїх конкурентів. Вона оптимально підходить для знаходження об'єкту номерного знака на зображені. Звичайно, з її допомогою можна шукати будь-які об'єкти, в яких у вас виникла потреба. Але в нашій реалізації YOLOv5 виконує процедуру локалізації.

### **3.1.4 Детектор тексту CRAFT**

Методи виявлення тексту сцени на основі нейронних мереж з'явилися нещодавно і показали багатообіцяючі результати. Попередні методи, навчені жорсткими обмежувальними полями на рівні слова, мають обмеження у поданні текстової області у довільній формі. CRAFT пропонує новий метод виявлення тексту сцени для ефективного знаходження текстової області шляхом вивчення кожного символу та спорідненості між символами. Щоб подолати відсутність окремих анотацій на рівні символів, запропонована структура використовує як дані анотації на рівні символів для синтетичних зображень, так і передбачувані істини на рівні символів для реальних зображень, отриманих за допомогою вивченої проміжної моделі. Широкі експерименти над шістьма тестами, включаючи набори даних

TotalText та CTW-1500, що містять сильно вигнуті тексти в природних зображеннях, демонструють, що розпізнавання тексту на рівні символів значно перевершує найсучасніші детектори. Відповідно до результатів, запропонований метод гарантує високу гнучкість при виявленні складних зображень тексту сцени, таких як довільно орієнтовані, вигнуті чи деформовані тексти.

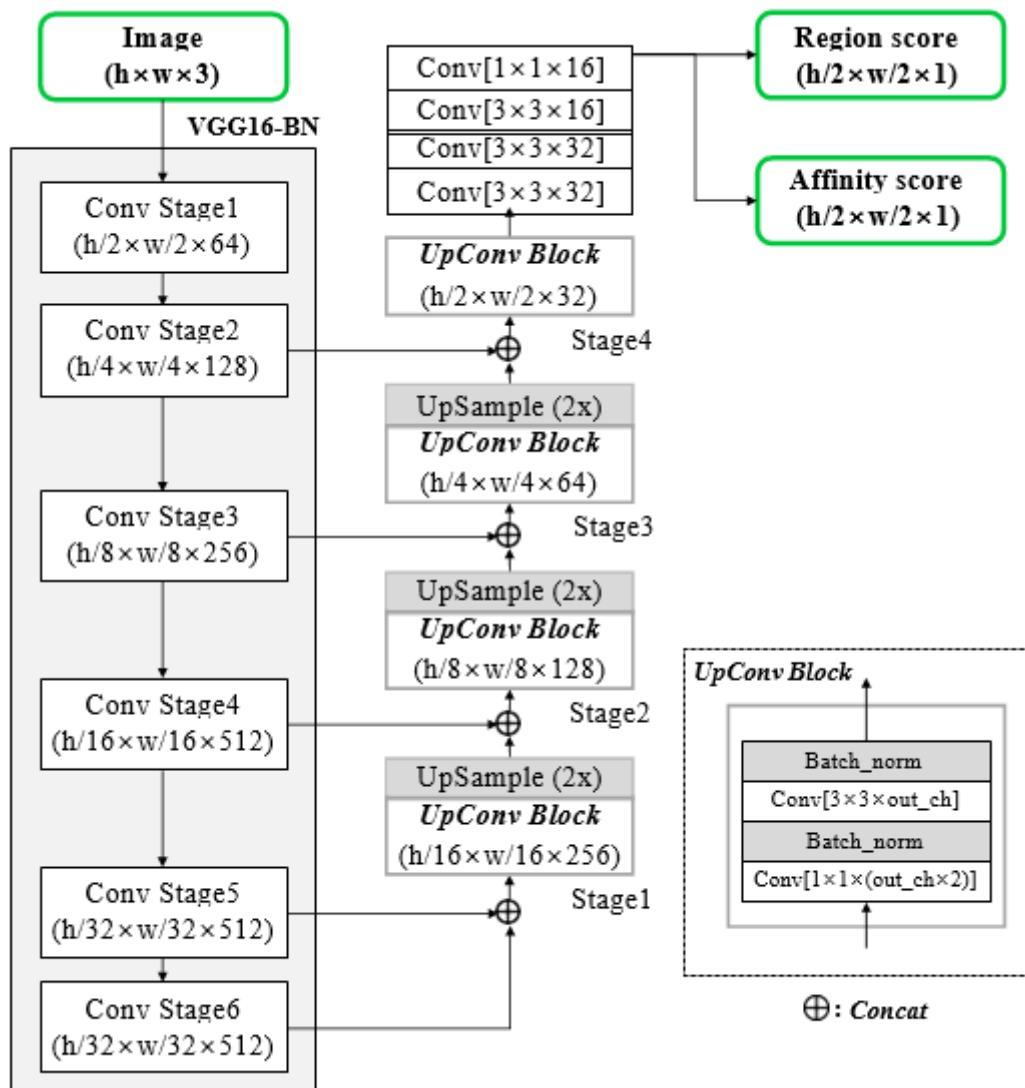


Рис. 3.1 Схематична ілюстрація архітектури мережі

Повністю згорнута архітектура мережі, заснована на VGG-16 з пакетною нормалізацією, прийнята в якості основи. Модель має з'єднання пропуску в частині декодування, яка схожа на U-net, оскільки вона об'єднує низькорівневі функції. Кінцевий результат має два канали як оціночні карти: оцінку регіону та оцінку спорідненості. Архітектура мережі схематично проілюстрована (Рис. 3.1)

Запропоновано новий детектор тексту під назвою CRAFT, який може виявляти окремі символи, навіть коли анотації на рівні символів не даються. Запропонований метод забезпечує оцінку області символів та оцінку спорідненості символів, які разом повністю охоплюють різні форми тексту способом знизу вгору. Оскільки реальні набори даних, що містять анотації на рівні символів, є рідкісними, запропоновано метод навчання, який слабо контролюється, який генерує псевдо-наземні факти з проміжної моделі. CRAFT демонструє найсучасніші вистави на більшості загальнодоступних наборів даних та демонструє здатність до узагальнення, показуючи ці вистави без доопрацювання.

### **3.2 Програмна реалізація**

Моя програмна реалізація базується на використанні вже побудованій моделі YOLOv5 та детектор тексту CRAFT. Виконання та керування програмою відбувається з файлу GUI.py. Всю роботу програми можна розділити на 5 головних процедур та вивести алгоритм її роботи (Рис. 3.2).

1. Локалізація
2. Нормалізація
3. Сегментація
4. Розпізнавання
5. Синтаксичний аналіз

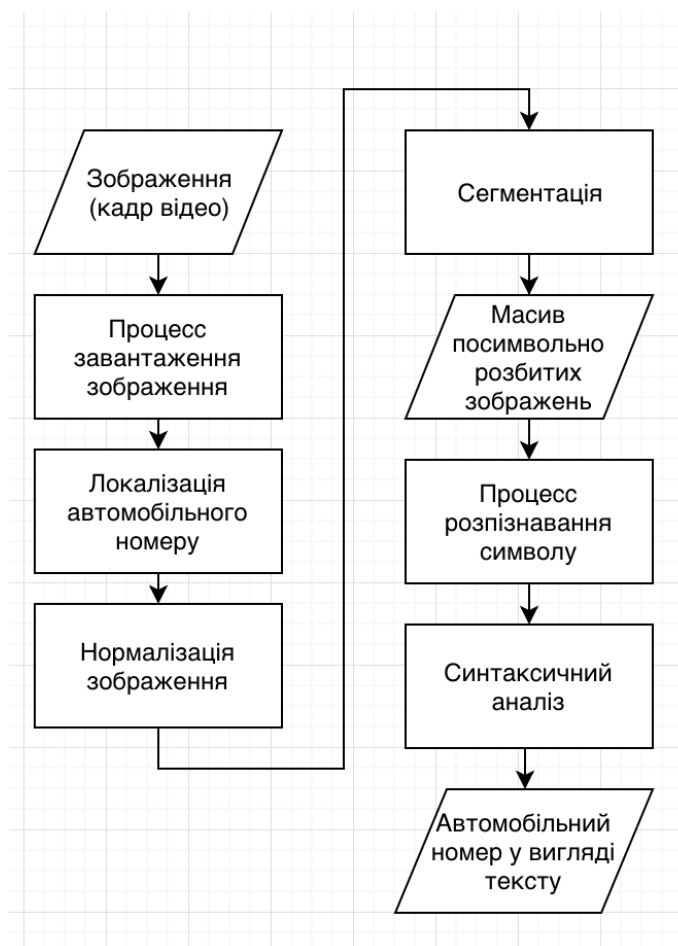


Рис. 3.2 Алгоритм роботи програми розпізнавання номерних знаків

При завантаженні зображення першим чином відпрацьовує процедура локалізації. За її виконання відповідна YOLOv5. Суть процедури полягає в знаходженні на зображенні самого автомобільного номеру та виділенні об'єкта номерного знаку (Рис 3.3).



Рис. 3.3 Локалізація номерного знаку

Наступним кроком є нормалізація знаку. Є ряд менш «глобальних» інструментів, за допомогою яких нам потрібно буде нормалізувати область з номерним знаком. Привести його в такий вид, при якому розпізнавання тексту буде можливим. Для таких перетворень використовують *OpenCV* (Рис 3.4).

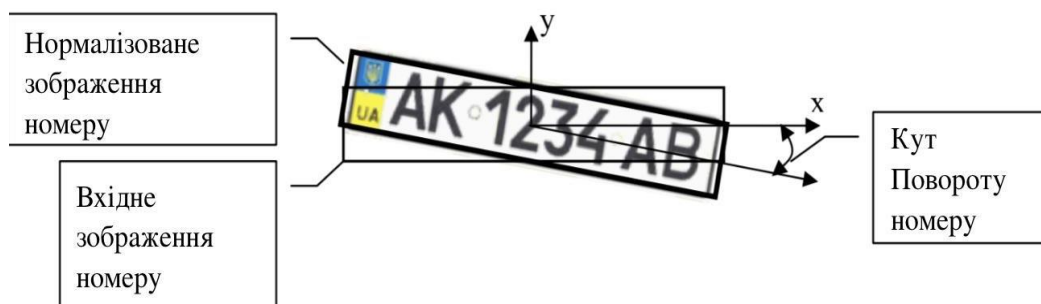


Рис. 3.4 Нормалізація номерного знаку

Після нормалізації слідує процес сегментування. Суть його полягає в розділенні всього номеру на окремі цифри, для подальшого їх розпізнання (Рис. 3.5).



Рис. 3.5 Приклад роботи процедур нормалізації та сегментації

Процес сегментації, розпізнавання та синтаксичний аналіз символів виконує детектор тексту *CRAFT*. Сегментовані цифри проходять процес розпізнавання і синтаксичного аналізу мовною моделлю. Після отримання вже розпізнаного номеру у вигляді символного поля відбувається процес пошуку регіону у локальній базі даних (Рис. 3.6).

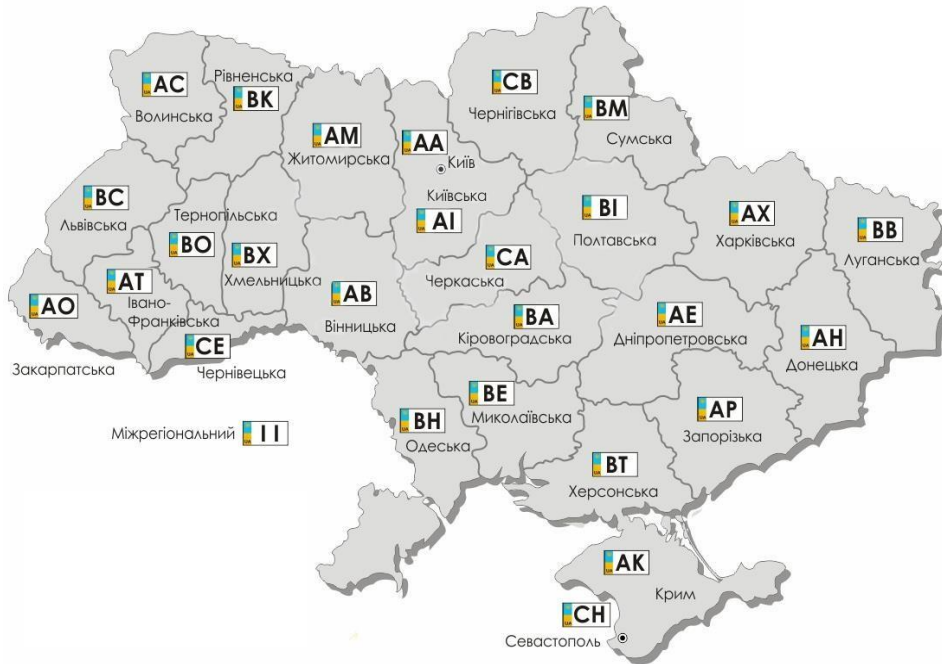


Рис. 3.6 Серії автомобільних номерів згідно регіону

Результати виводяться на візуальний інтерфейс користувача(Рис. 3.7).

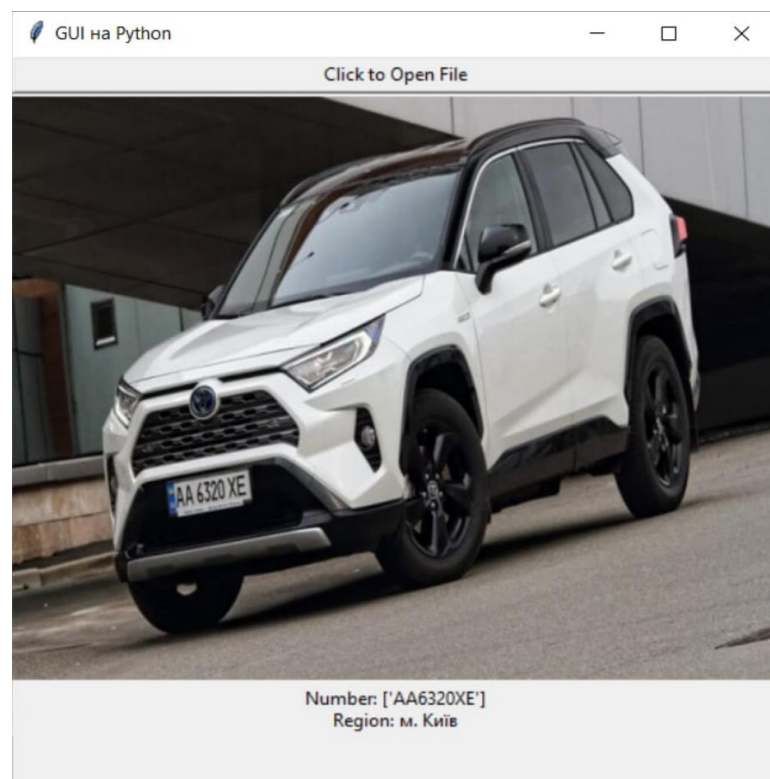


Рис. 3.7 Кінцевий вивід результатів

### **Висновки до розділу 3**

Підсумовуючи третій розділ, можемо зробити такі висновки:

1. Охарактеризовано методику синтезу ШНМ для розпізнавання символів та побудовано модель ШНМ для поставленого завдання розпізнавання символів. У результаті впровадження інтелектуального прототипу, який функціонує згідно із пропонованим алгоритмом, можуть розпізнаватися символи, які належать до різних алфавітів, цифрові символи та знаки в залежності від об'єму пам'яті ІС, яка включає в себе певні еталони із зашумленістю до 40 %.

2. Проаналізовано принцип використання алгоритму комбінації двох мереж з однаковою структурою. Об'єднуючись у мережі, ШНМ формують систему обробки інформації, яка забезпечує ефективне пристосування моделі до постійних трансформацій з боку оточуючого середовища. Під час роботи НМ відбувається видозміна вхідного вектора сигналів у вихідний. Конкретний вид трансформації задається архітектурою НМ, властивостями нейронних компонентів, інструментами управління та синхронізації інформаційних даних між нейронами.

3. Розроблено та впроваджено програму розпізнавання номерних знаків автомобілів.

## РОЗДІЛ 4

### ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

#### 4.1 Інструкція користувача

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

Для запуску програми на комп'ютері треба мати встановлену версію Python не менше 3, 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 файли програми. Через менеджер пакетів pip3 потрібно буде встановити кілька модулів на python3, вони будуть перераховані в окремому файлі requirements.txt. Якщо качати окремо, то ось перелік модулів.

- cython
- setuptools
- numpy>=1.16.\*
- imgaug>=0.4.\*
- tensorflow>=2.3.\*
- opencv\_python
- scikit\_image
- asyncio
- gitpython
- pycocotools
- tqdm
- matplotlib
- torch>=1.8
- torchvision>=0.9
- PyYAML>=5.4
- seaborn

Після встановлення всіх необхідних компонентів, відкриваємо командну строку и вказуємо шлях до місця розташування папки з файлами програми. Наприклад, ось повна команда – cd C:\Users\lalis\Desktop\DiplomAuto1. Опинившись у папці вводимо команду – python GUI.py. Програма починає свою роботу і з'являється початковий екран(Рис. 4.1).

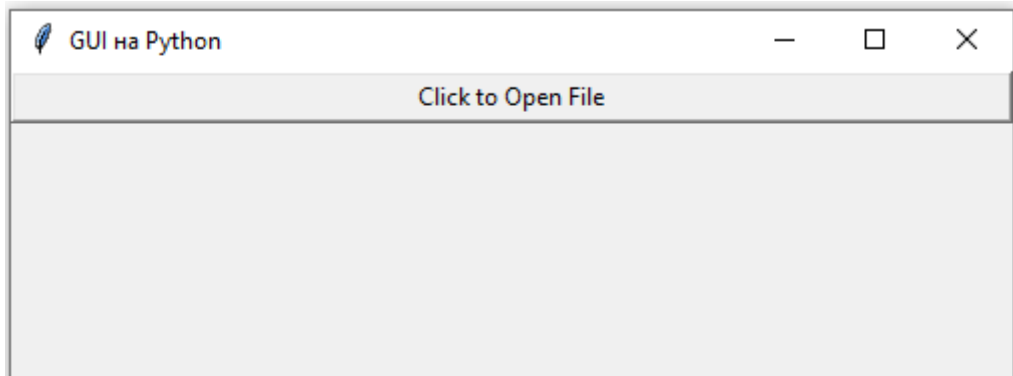


Рис. 4.1 Початковий екран

При натисканні на кнопку відкривається папка в яку треба завантажити зображення або вибрати з числа тих які вже є (Рис. 4.2).

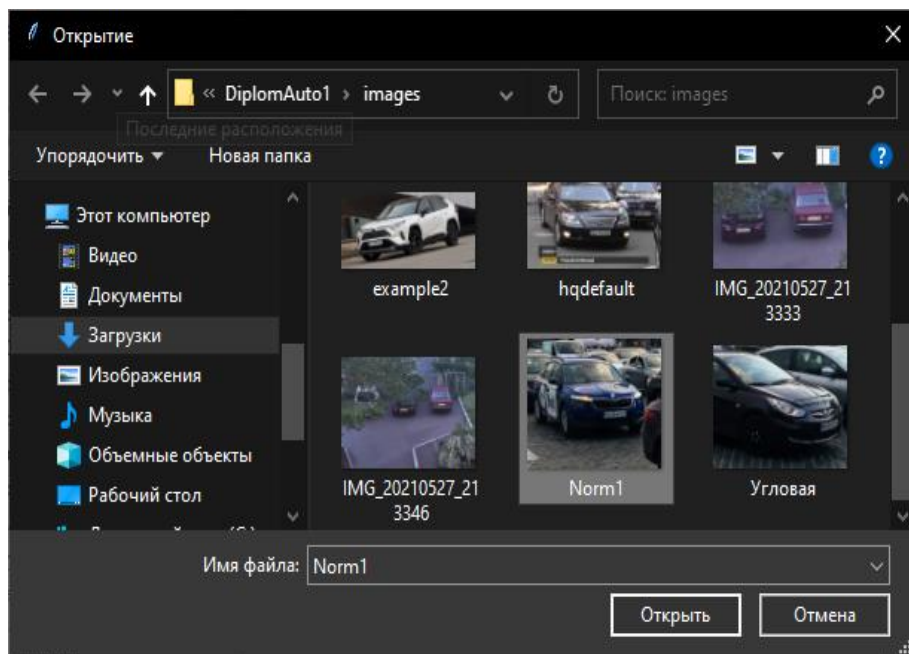


Рис. 4.2 Вибір зображення

Відкриваємо і отримуємо результат. Мінімалістичний дизайн програми, нічого зайвого тільки виконання головної задачі.

## 4.2 Тестування розробленої моделі

Тестування проводиться на ПК з операційною системою Windows 10, обсягом оперативної пам'яті 16,00 ГБ та відеопам'яттю на 4 ГБ, з усіма актуальними модулями.

Метою тестування додатку було виявлення сильних і слабких сторін даної версії. При тестуванні використовувались різні варіації розміру та чіткості автомобільного номеру на зображеннях.

Головні фактори, що впливають на якість розпізнавання – це зашумленість зображення, кількість світла, кут нахилу номера та чистота номерного знаку від забруднень (рис. 4.3).



Рис. 4.3 Тест на неякісному зображенні

Якщо номер автомобіля не відповідає сьогоdnішньому формату або такий формат номера не записаний в локальній базі даних, замість регіону виводиться повідомлення «Регіон не знайдений» (рис. 3.4).

Ще однією з проблем додатку є неточність розпізнання тексту, бо в деяких випадках цифри плутаються з буквами («О» та 0, «І» та 1, «В» та 8) (Рис. 4.4). Але ця проблема легко вирішується при збільшенні кількості тестових зображень та подальшому навчанні мережі.

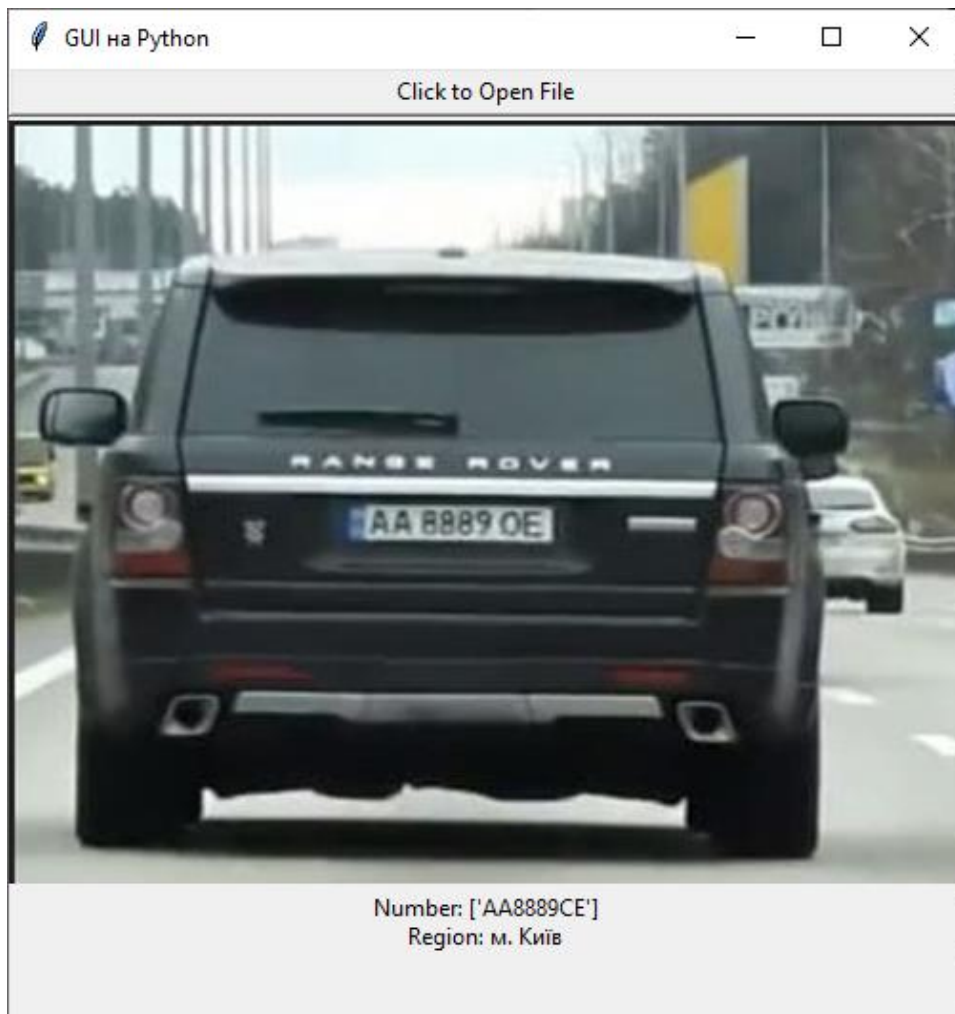


Рис. 4.4 Розпізнання «О» як «С»

Загалом, при якісній фотографії НМ видає результат з 95% розпізнавання номеру, що є гарним показником. Більшість проблем вирішуються за допомогою подальшого навчання НМ.

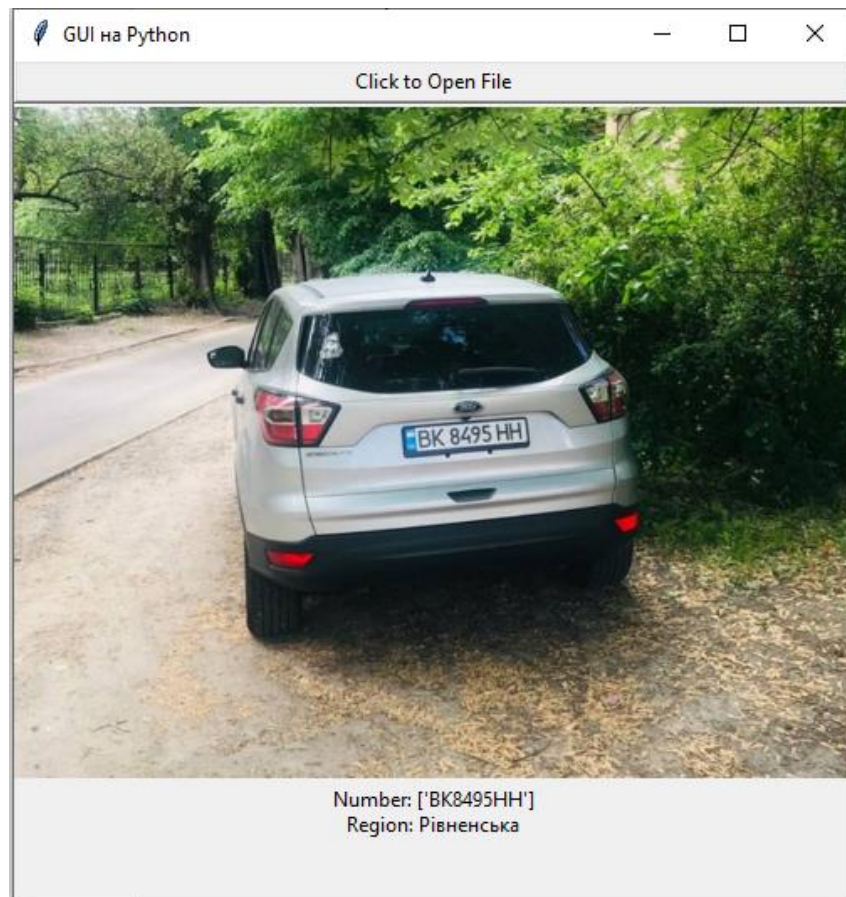


Рис. 4.5 Розпізнання номерного знаку на якісній фотографії

#### **Висновки до розділу 4**

1. Розроблена інструкція для користувача
2. Показано результат роботи додатку
3. Виявлено головні недоліки і недоробки програми для подальшого їх виправлення

## ВИСНОВКИ

В атестаційній роботі були проведені дослідження щодо актуальності, доцільності та практичності систем розпізнання номерних знаків на зображеннях. Був проведений аналіз методів розпізнавання зображень і було виявлено що нейромержеві 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. 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.

Завдання розпізнавання автомобільних номерів має безліч рішень. На кожному етапі алгоритму існує велика кількість шляхів, за якими може піти розробник. У даній роботі порівнювалися деякі з них. Зі збільшенням обчислювальних потужностей область машинного зору розвивається швидше і швидше. Деякі виробники почали звертати уваги на потреби дослідників і виробляти дійсно хороші продукти, що допомагають вченим. Крім того, тепер не треба прикладати великих зусиль для створення нейронних мереж. Практично на кожній мові програмування є доступні бібліотеки, що дозволяють налаштувати і навчити нейромерже без занурення в глибину її реалізації – все вже зроблено сторонніми розробниками. Необхідно тільки прочитати документацію і підготувати необхідну навчальну множину.

У ході атестаційної роботи був розроблений додаток і реалізований метод розпізнавання номерних знаків на зображеннях. Як середовище розробки було обрано PyCharm від IntelliJ IDEA сумісно з моделлю YOLOv5, детектор тексту CRAFT, а сам програмний код був написаний на мові Python. 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.

Також були виявлені загальні проблеми для всіх алгоритмів: брудні номери та погана якість зображення. Проблему бруду можна вирішити лише частково. Якщо якийсь із символів забруднений цілком, то розпізнавання стане неможливий.

У якості подальшого розвитку теми можна зазначити використання більшої кількості тестових зображень для покращення результатів розпізнання, 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.

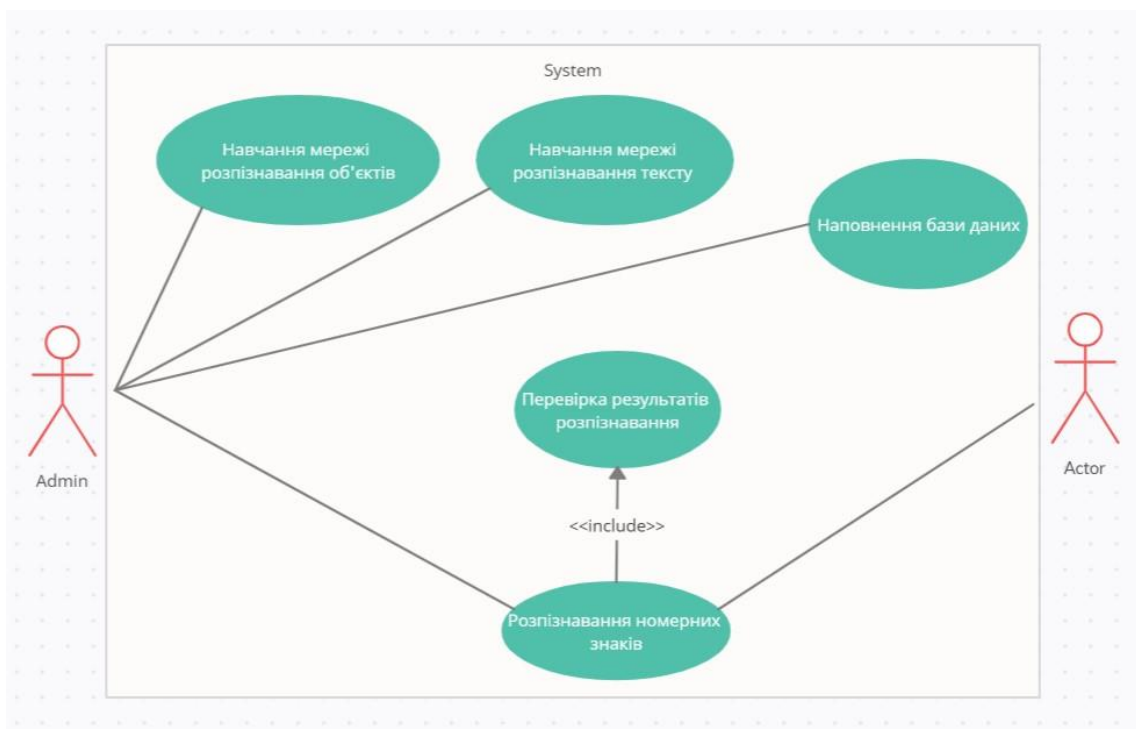
## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Головач Ю. Складні мережі / Ю. Головач, О. Олемской, К. фон Фербер та ін. // Журнал фізичних досліджень. – 2006. – 10, № 4. – С. 247–289.
2. Терехов В. А., Єфімов Д. В., Тюкин И. Ю. Нейромережні системи керування. – 1-е. – Высшая школа, 2002. – С. 156–184.
3. Комашинский В.И. Смирнов Д.А. Внедрение в нейро–информационные технологии. / В.И. Комашинский, Д.А. Смирнов – СПб, 1999. – С. 33–48.
4. Саймон Хайкин. Нейронні мережі: повний курс = NeuralNetworks: AComprehensiveFoundation. – 2-е. – М.: «Вильямс», 2006. – С. 1099–1104.
5. Советов Б.Я. Информационные технологии / Б.Я. Советов, В.В Цехановский – М.: Высшая школа, 2005 – 263 с.
6. Моделі нейронних мереж. – Режим доступу: <https://studme.com.ua/1246122010028/neural/models.htm> .
7. Поліщук Д. О. Комплексне детерміноване оцінювання складних ієрархічно–мережєвих систем: І. Опис методики / Д. О. Поліщук, О. Д. Поліщук, М. С. Яджак // Системні дослідження та інформаційні технології. – 2015. – № 1.– С. 21–31.
8. Воеводин В. Параллельные вычисления / Воеводин В .В., Воеводин В. В . – СПб .: БХВ – Петер – бург , 2002. – 608 с.
9. Советов Б.Я. Информационные технологии / Б.Я. Советов, В.В Цехановский – М.: Высшая школа, 2005 – 263 с.
10. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М. Горячая линия–Телеком 2004. – 112 с.
11. Шапиро Л., Стокман Д. Компьютерное зрение //М.: Бином. Лаборатория знаний. – 2006. – Т. 752. – С. 66–69.
12. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика. – 1-е. – М.: Горяча лінія – Телекому, 2001. – С. 382.

13. Fieres, J., Schemmel, J., and Meier, K. (2006) "Training convolutional networks of threshold neurons suited for low-power hardware implementation." In *Neural Networks, 2006. IJCNN'06. International Joint Conference on. IEEE.* (pp. 21-28).
14. Zhang, Z. (2016) "Derivation of Backpropagation in Convolutional Neural Network(CNN)".
15. Lee, K. B., Cheon, S., & Kim, C. O. (2017) "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes." *IEEE Transactions on Semiconductor Manufacturing*: 135-142. Lima, E., Sun, X.
16. Maggiori, E., Tarabalka, Y., Charpiat, G., and Alliez, P. (2017) "Convolutional neural networks for large-scale remote-sensing image classification." *IEEE Transactions on Geoscience and Remote Sensing*: 645-657.
17. Dong, J., Wang, H., Yang, Y., and Liu, L. (2017) "Learning and Transferring Convolutional Neural Network Knowledge to Ocean Front Recognition." *IEEE Geoscience and Remote Sensing Letters*: 354-358.
18. Liu, H., Li, B., Lv, X., and Huang, Y. (2017) "Image Retrieval Using Fused Deep Convolutional Features." *Procedia Computer Science*: 749-754

## ДОДАТКИ

### Додаток А. UML Use Case Diagram



### Додаток Б. UML Component diagram

