



## Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Яловенку Андрію Сергійовичу  
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження застосування текстонів у комп'ютерному зорізатверджена наказом по університету від « 22 » жовтня \_\_\_\_\_ 2021 року № 1574 Ст2. Термін подання студентом роботи до екзаменаційної комісії 6 грудня \_\_\_\_\_ 2021 р.3. Вихідні дані до роботи математичні моделі комп'ютерного зору, перелік використуваних програмних засобів: теоретичні відомості про методи сегментації текстурних зображень.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Огляд методів розпізнавання об'єктів, комп'ютерного зору та машинного навчання2. Дослідження існуючих методів опису та визначення текстур та текстонів3. Моделі застосування текстонів зображення4. Практична реалізація

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Архітектура згорткової нейронної мережі для визначення текстур зображення, Метод градієнтного спуску, Приклади екранних форм

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

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

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	22.10.2021	
2	Аналіз завдання, підбір літератури	23.10.21-30.10.21	
3	Аналіз літератури з досліджуваної проблеми	31.10.21-05.11.21	
4	Моделі застосування текстонів зображення	06.11.21-10.11.21	
5	Програмна реалізація	11.11.21-22.11.21	
6	Оформлення пояснювальної записки	23.11.21-30.11.21	
7	Перевірка на плагіат	02.12.2021	
8	Рецензування	03.12.2021	
9	Підготовка презентації та доповіді	05.12.2021	
10	Занесення роботи в електронний архів	07.12.2021	
11	Попередній захист кваліфікаційної роботи	07.12.2021	

Дата видачі завдання 22 жовтня 2021 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Машталір В.П.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Кваліфікаційна робота містить 66 сторінки, 23 рисунки, 1 таблиця, 42 джерела.

КОМП'ЮТЕРНИЙ ЗІР, ТЕКСТОН, ТЕКСТУРА, РОЗПІЗНАВАННЯ 3D-ОБ'ЄКТІВ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ФІЛЬТРАЦІЯ, НЕЙРОННА МЕРЕЖА.

Об'єктом дослідження є текстон як найменший елемент зображення з властивостями яскравості та кольору, що характеризують цифрове зображення об'єкту на сцені.

Мета дослідження – методи комп'ютерного зору та методи машинного навчання для розробки програмної системи, що виконує пошук та класифікацію текстонів на графічних зображеннях.

Побудована та навчена згортова нейронна мережа забезпечує задовільну точність розпізнавання об'єктів на основі представлених текстонів зображення та текстур.

Результати цієї роботи можуть бути використані в комерційних та особистих цілях (управління транспортними засобами, пошук товарів по фотографії в інтернет-магазині, вирішення пошукових задач).

COMPUTER VISION, TEXTON, TEXTURE, RECOGNITION OF 3D OBJECTS, CLASSIFICATION OF IMAGES, FILTRATION, NEURAL NETWORK.

The object of study is text as the smallest element of the image with the properties of brightness and color that characterize the digital image of the object on the stage.

The purpose of the work is to study the methods of computer vision and machine learning methods for the development of a software system that searches for and classifies texts on graphic images.

The constructed and trained convolutional neural network provides satisfactory accuracy of object recognition based on the presented image texts and textures.

The results of this work can be used for commercial and personal purposes (driving vehicles, searching for goods on the photo in the online store, solving search problems).

## ЗМІСТ

Вступ.....	7
1. Огляд методів розпізнавання об'єктів, комп'ютерного зору та машинного навчання .....	9
1.1 Введення в комп'ютерний зір.....	9
1.1.1 Принцип роботи комп'ютерного зору.....	10
1.1.2 Перетворення рівня яскравості.....	14
1.1.3 Детектор кутів Харріса.....	15
1.1.4 Фільтрації контурів.....	15
1.2 Згорткові нейронні мережі.....	19
1.3 Розпізнавання об'єктів за допомогою нейронної мережі.....	21
1.3.1 Згортковий шар нейронної мережі.....	22
1.3.2 Підвибірковий шар.....	24
1.3.3 Повнозв'язковий шар.....	24
1.4 Огляд програмних інструментів для створення веб-сервісу.....	25
1.5 Постановка задачі дослідження.....	28
2. Дослідження існуючих методів опису та визначення текстур та текстонів.....	30
2.1 Аналіз задачі класифікації та пошуку схожих зображень.....	30
2.2 Аналіз проблем опису текстур та текстонів.....	32
2.3 Дослідження існуючих методів для аналізу текстур та текстонів.....	33
3. Моделі застосування текстонів зображення.....	41
3.1 Побудова згорткової нейронної мережі для опису текстонів зображення.....	41
3.2 Навчання згорткової нейронної мережі з використанням алгоритму зворотного розповсюдження помилки.....	47
4. Практична реалізація.....	51
4.1 Побудова та моделювання згорткової нейронної мережі для визначення текстонів зображення.....	51

4.2 Результат роботи програми .....	53
Висновки .....	55
Перелік джерел посилання .....	58
Додаток А .....	62

## ВСТУП

Сьогодні одними з найважливіших галузей досліджень та розробок сучасної прикладної математики та кібернетики є – комп'ютерний зір, розпізнавання образів, машинне навчання, а також прогнозування та аналіз даних. Прискорення темпів розвитку технологій інформаційного суспільства, розвиток концепцій «розумний дім» і «розумне місто», розвиток інтернету та системи штучного інтелекту визначають особливе місце для цих областей сучасного світу. Багато прикладних завдань програмування використовують методи збору даних, кластеризації і класифікації, статистичного виводу. У повсякденне життя активно впроваджуються технології розпізнавання образів. Раніше завдання розпізнавання, що вважалися складними, сьогодні вирішуються за допомогою звичайних мобільних пристроїв і смартфонів.

Поширені методами та підходами до вирішення завдань детектування, розпізнавання та класифікації є:

- порівняння із зразком – класифікація за найближчим середнім, по відстані до найближчого сусіда. Також до групи порівняння із зразком можна віднести структурні методи розпізнавання.
  - зіставлення із шаблоном – метод розпізнавання, в якому використовується невелике зображення або шаблон для пошуку областей у збільшеному зображенні.
  - нейронні мережі – клас методів глибокого навчання, який використовується для автоматичного вивчення властивостей об'єкта та його подальшої ідентифікації.
- Відмінною рисою цих методів від інших є здатність навчатися.

Мета роботи – дослідження методів комп'ютерного зору та методів машинного навчання для розробки програмної системи, що виконує пошук та класифікації текстонів на графічних зображеннях (фото-, відео).

Для досягнення поставленої мети у роботі поставлені та вирішені наступні завдання:

- провести огляд методів розпізнавання об'єктів, комп'ютерного зору та машинного навчання;
- дослідити методи та існуючі алгоритми для детектування та класифікації об'єктів, проаналізувати існуючі методи опису та визначення текстур та текстонів зображення;
- представити архітектуру згорткової нейронної мережі для визначення текстонів зображення, виконати навчання нейронної мережі;
- розробити програмну реалізацію системи визначення текстонів зображення.

# 1. ОГЛЯД МЕТОДІВ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, КОМП'ЮТЕРНОГО ЗОРУ ТА МАШИННОГО НАВЧАННЯ

## 1.1 Введення в комп'ютерний зір

Комп'ютерний зір – це технологія, яка в залежності від поставленого завдання може знаходити, відстежувати, класифікувати та ідентифікувати об'єкти, витягуючи та аналізуючи отриману інформацію з зображень або відео. Цей напрямок виник у рамках штучного інтелекту. Основним завданням є розпізнавання образів, тому для повної та правильної інтерпретації того, що зображено, потрібно мати необхідну інформацію як масив пікселів, витягнутих із зображення. Кожне зображення складається з набору пікселів. Для комп'ютера немає більш тонкої деталізації, крім пікселя.

Під комп'ютерним зором також розуміється автоматичне вилучення інформації з зображень. У ролі інформації може виступати 3D-моделі, положення камери, виявлення та розпізнавання об'єктів, групування зображень та пошук зображень за змістом. Застосування комп'ютерного зору набуває досить велику значущість у різних сферах нашого життя. За допомогою цієї технології реалізовано розумну систему відеоспостереження, яка обробляє вхідні відео та, у відповідність навченого алгоритму, приймає необхідні рішення.

До вирішення завдань детектування та класифікації застосовуються різні підходи: статистичні, спеціально розроблені теорії ключових точок, застосування алгоритмами класифікації зображень за змістом, а також затребуваний підхід – машинне навчання.

Найбільш поширені завдання комп'ютерного зору [1]:

– розпізнавання – одне з базових і першорядних завдань у обробці зображень, комп'ютерному та машинному зору. Воно допомагає класифікувати та ідентифікувати об'єкти, що характеризуються певним набором властивостей та ознак;

- відновлення зображень – це видалення шуму з використанням різних методів, наприклад, розмиття за допомогою фільтрів на основі машинного навчання (шум датчика, розмитість об'єкта, що рухається і т. ін.);

- аналіз руху – завдання використовує комп'ютерний зір для оцінки швидкості руху об'єктів у відео. Також застосовується для оцінки рухів, в яких послідовність відеоданих обробляється для знаходження швидкості кожної точки зображення або 3D сцени;

- відновлення чи реконструкція сцени – допомагає відтворити тривимірну модель зображення чи сцени, яка вводиться за допомогою зображень чи відео. Найчастіше моделлю служить набір точок тривимірного простору;

- обробка та аналіз зображення – завдання зосереджено на перетворенні одного 2D-зображення в інше. Реалізується за допомогою піксельних операцій, таких як підвищення контрастності або повертання зображення;

- високорівнева обробка – подає невеликий набір даних. Вона використовує різні методи для вилучення інформації з сигналів в цілому, наприклад, набір точок або ділянка зображення, в якому імовірно знаходиться певний об'єкт, або частина даних, що цікавить.

### 1.1.1 Принцип роботи комп'ютерного зору

Комп'ютери інтерпретують зображення як послідовність пікселів, кожен з яких має власний набір значень кольору. Пікселі є необробленими будівельними блоками зображення. Кожне зображення складається з набору пікселів. Зазвичай піксель вважається кольором або яскравістю світла, який з'являється на зображенні [2]. Якщо ми розглядаємо зображення як сітку, кожен квадрат містить один піксель.

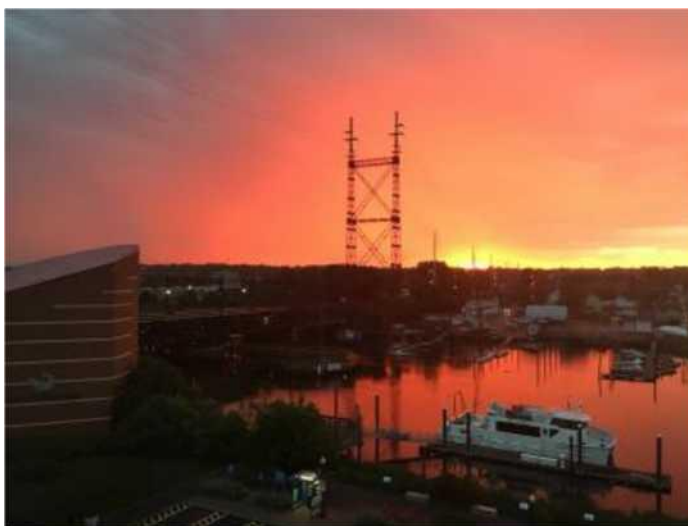


Рисунок 1.1 – Приклад зображення шириною 1000 пікселів та висотою 750 пікселів

Зображення рисунку 1.1 має роздільну здатність 1000 x 750 пікселів, де 1000 це ширина, а 750 - висота. Ми можемо уявити зображення у вигляді матриці. У цьому випадку наша матриця має 1000 стовпців (ширина) та 750 рядків (висота) і містить  $1000 \times 750 = 750\,000$  пікселів, які представлені двома способами: а) відтінки сірого (один канал); б) колір.

У зображеннях з градаціями сірого кольору кожен піксель є скалярним значенням від 0 до 255, де нуль відповідає «чорному» кольору, а 255 - «білому». Значення між 0 і 255 мають різні відтінки сірого, де значення ближче до 0 темніше, а значення ближче до 255 світліше. Градієнтне зображення в градаціях сірого (рисунок 2) демонструє темніші пікселі з лівого боку, а світліші з правого. З рисунка 2, можна зрозуміти те, як значення в градаціях сірого перетворюються на двовимірний масив цілих чисел [9]:

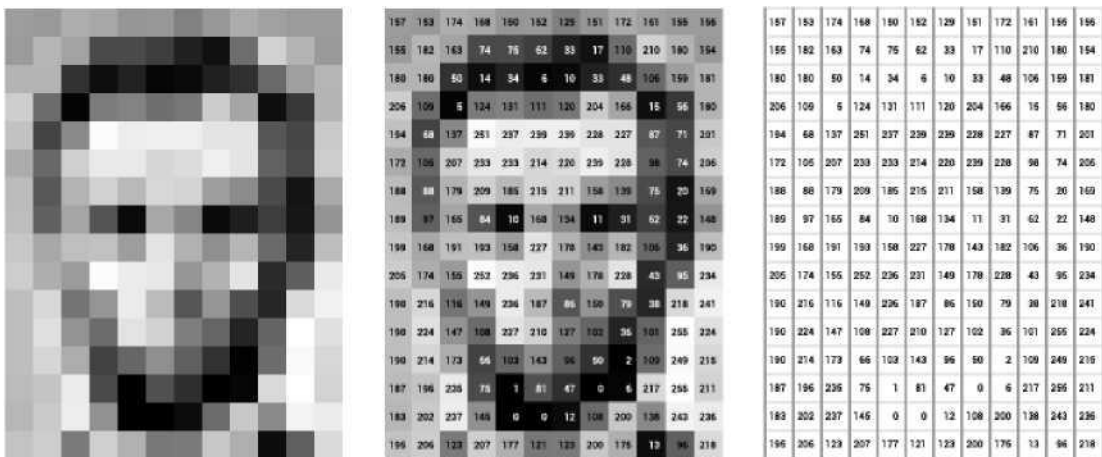


Рисунок 1.2 – Зображення та їх матричні уявлення

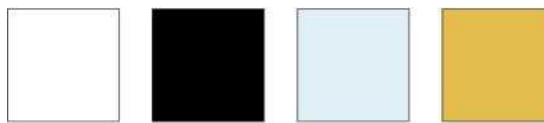
Ряди чисел праворуч (крайній правий рисунок) – комп'ютерне представлення введеного зображення. У прикладі зображення має 12 стовпців та 16 рядків, що означає 192 вхідних значень для цього зображення.

Ще потрібно описати як комп'ютер перетворює собі кольорові зображення як матриць. Пікселі у кольорному просторі RGB більше не є скалярними значеннями, як було у зображеннях у градаціях сірого, на одному каналі. Натомість пікселі представлені списком із трьох значень: одне значення для компонента червоного (Red), друге для зеленого (Green) та третє для синього (Blue)[9]. Щоб визначити колір у колірній моделі RGB, все, що нам потрібно зробити, це підрахувати кількість червоного, зеленого та синього кольорів, що містяться в одному пікселі. Кожен канал Red, Green і Blue може мати певні значення в діапазоні  $[0, 255]$ , всього 256 відтінків, де 0 означає відсутність уявлення, а 255 - це повне уявлення. Враховуючи, що значення пікселя має бути лише в діапазоні  $[0, 255]$ , ми зазвичай використовуємо 8-бітові цілі числа без знаку для представлення яскравості.

Розглянемо приклад створення різних кольорів зображення на рисунку 1.3.

### Creating colors with RGB pixels

White Black Pale-Blue Gold



**RGBRGBRGBRGB**

**255, 255, 255 0, 0, 0 255, 239, 248 228, 189, 79, 255 0**

Рисунок 1.3 – Створення кольорів у пікселі RGB

З рисунка 1.3 можна зрозуміти, що кожен піксель складається з трьох різних кольорів, при зміні їх відтінків можна отримати абсолютно різні кольори.

Для вирішення завдань детектування, розпізнавання та класифікації об'єктів на графічному зображенні необхідно попередньо обробляти зображення (наприклад, виконати операцію середнє віднімання або масштабування). Оскільки типи даних, використовувані бібліотеками (наприклад, OpenCV), завантажують зображення з диска, їх необхідно перетворювати, як безпосередньо застосовувати алгоритми навчання до зображень. Враховуючи наші три значення Red, Green та Blue, ми можемо об'єднати їх у кортеж RGB (червоний, зелений, синій). Цей кортеж представляє цей колір у колірному просторі RGB.

Ми можемо перетворити зображення RGB як складається з трьох незалежних матриць, шириною  $W$  і висотою  $H$ , по одній для кожного з компонентів RGB, як показано на рисунку 1.3. Ми можемо об'єднати ці три матриці для отримання багатовимірного масиву з формою  $W \times H \times D$ , де  $D$  – глибина або кількість каналів. Для колірному простору RGB  $D=3$ .



Рисунок 1.4 – Вихідне зображення та його RGB канали

Для комп'ютерів завдання інтерпретація вмісту зображення менш тривіальне, ніж просто завдання відображення зображення. Все, що бачить наш комп'ютер-це велика матриця чисел. Щоб зрозуміти зміст зображення, ми повинні застосувати класифікацію зображень, яка є завданням використання алгоритмів комп'ютерного зору та машинного навчання[9]. У цій роботі були використані різні способи, крім глибокого навчання, що покращують комп'ютерний зір. Тим не менш, вони добре працюють для більш простих завдань, але оскільки дані стають величезними, а завдання стає складним, вони не замінюють згорткові нейронні мережі.

### 1.1.2 Перетворення рівня яскравості

Перетворюючи за допомогою NumPy [1] зображення в масив, ми можемо застосувати до нього різні математичні операції. Найпростіший приклад - перетворення рівня яскравості напівтонового зображення. Візьмемо довільну функцію  $f$ , що відображає інтервал  $0 \dots 255$  (або, якщо завгодно,  $0 \dots 1$ ) в себе, тобто область значень збігається з областю визначення.

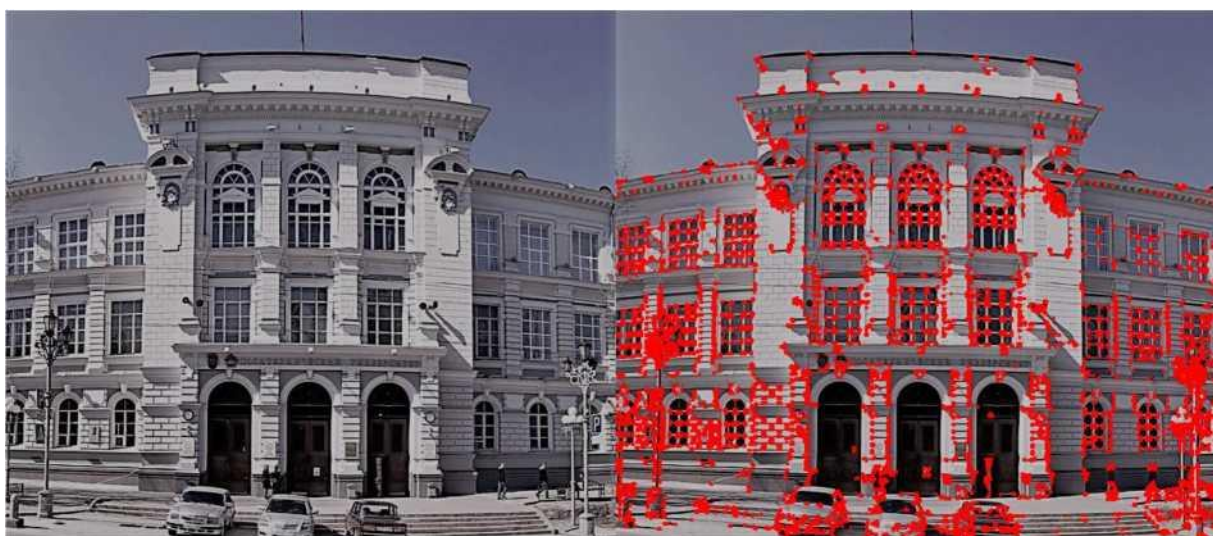
Іншим прикладом перетворення яскравості є вирівнювання гістограми. Ця

операція змінює гістограму яскравості, щоб результуюча гістограма містила всі можливі значення яскравості і при цьому приблизно в однаковій кількості. Вона часто застосовується для нормування яскравості перед подальшою обробкою, а також підвищення контрастності. У цьому випадку для перетворення використовується функція розподілу (cumulative distribution function, CDF) значень пікселів у зображенні.

### 1.1.3 Детектор кутів Харріса

Алгоритм виявлення кутів Харріса (детектором кутів Харріса-Стівенса) - один із найпростіших детекторів кутів об'єктів [2]. Ідея полягає в тому, щоб знайти особливі точки, в околицях яких є межі в кількох напрямках, і є кутові точки.

Розглянемо спосіб з прикладу зображення будинку:



а)

б)

Рисунок 1.5 - а) вихідне зображення; б) зображення з виявленими кутами за алгоритмом Харріса

### 1.1.4 Фільтрації контурів

Контури дуже корисні, коли ми хочемо перейти від роботи із зображенням до

роботи з об'єктами на цьому зображенні. Коли об'єкт досить складний, але добре виділяється, часто єдиним способом роботи з ним є виділення його контурів.

Оператор Кенні є популярним алгоритмом виявлення меж і найчастіше використовується виділення контуру об'єктів [18].

Алгоритм Кенні для виявлення меж об'єктів складається з п'яти етапів:

1. Пригнічення шуму;
2. Розрахунок градієнта;
3. Пригнічення країв зображення;
4. Подвійний поріг;
5. Відстеження країв за гістерезою.

Ще одна важлива річ, яку варто згадати, що алгоритм ґрунтується на зображеннях у градаціях сірого. Отже, попередньою умовою є перетворення зображення на відтінки сірого кольору перед виконанням вищезгаданих кроків.

Коротко дамо визначення етапів алгоритму Кенні.

Придушення шуму – є один із способів позбутися шумів на зображенні. Для того щоб згладити шум застосовують розмиття Гауссом. Для цього застосовується метод згортки зображень з ядром гауса (наприклад,  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  і т. д. пікселів). Розмір ядра залежить від очікуваного ефекту розмиття. В основному найменше ядро - це менш помітна пляма. При обробці зображенні ядро (згортка матриці) є невеликою матрицею. Воно використовується для розмиття, підвищення різкості, тиснення, виявлення країв та багато іншого.

Розрахунок градієнта – визначає інтенсивність та напрямок краю шляхом обчислення градієнта зображення з використанням операторів виявлення краю. ГраДієнт - це векторна величина, що показує напрямок якнайшвидшого зростання двовимірної функції яскравості зображення.

Краї відповідають зміні інтенсивності пікселів. Щоб виявити її, найпростіше застосувати фільтри, які виділяють цю зміну інтенсивності в обох напрямках: горизонтальному (x) та вертикальному (y).

Зменшення країв зображення – в ідеалі кінцеве зображення повинне мати тонкі краї. Таким чином, ми повинні виконати немаксимальне придушення, щоб виявити

краї. Для цього можна застосувати наступний алгоритм. Обійти через усі крапки на матриці інтенсивності градієнта та знайти пікселі з максимальним значенням у напрямках ребер (рисунок 1.6).

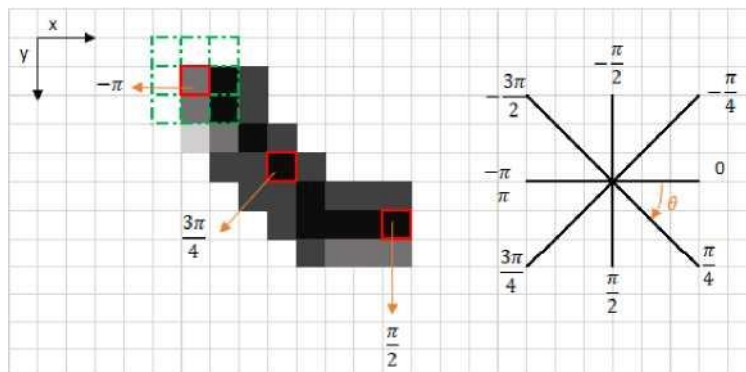


Рисунок 1.6 – частина краю об'єкта у вигляді пікселів

Червоний прямокутник у верхньому лівому куті, представляє піксель інтенсивності оброблюваної градієнтної матриці інтенсивності. Відповідний напрямок країв позначено помаранчевим

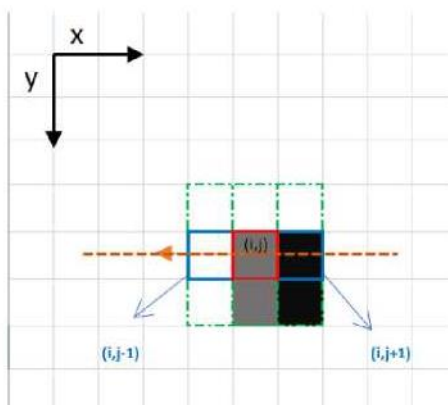


Рисунок 1.7 – Зображення з неповним зменшенням країв

Напрямок краю – помаранчева пунктирна лінія (горизонтальна лінія зліва направо, рисунок 1.7). Мета алгоритму – перевірити, чи є пікселі в одному і тому ж напрямку більш менш інтенсивними, ніж оброблювані [19].

У наведеному вище прикладі піксель  $(i, j)$  обробка, і пікселі в тому ж напрямку виділяються синім кольором  $(i, j-1)$  і  $(i, j+1)$ . Якщо один з цих двох пікселів є більш

інтенсивним, ніж оброблюваний, то зберігається тільки більш інтенсивний піксель ( $i, j-1$ ) здається більш інтенсивним, тому що він білий (значення 255). Отже, значення інтенсивності поточного пікселя ( $i, j$ ) встановлюється в 0. інтенсивні значення, то значення поточного пікселя зберігається [19].

Подвійний поріг. Мета застосування даної операції спрямована на виявлення трьох видів пікселів і за допомогою них виявляти контури:

- сильні пікселі – пікселі з високою інтенсивністю (яскравістю);
- слабкі пікселі – це пікселі, які мають достатньо значення інтенсивності. Ці пікселі не можна вважати сильними, але їх значення інтенсивності не є маленькими, щоб їх вважати, як пікселі, що не стосуються країв;
- інші пікселі вважаються такими, що не мають відношення до країв.

Відстеження країв – результати порогового значення гістерезис, складається з перетворення слабких пікселів у сильні, якщо хоча б один з піксель навколо краю, що обробляється, є сильним [19].

Подивимося отримані нами результати після застосування алгоритму виявлення країв різних об'єктів (рисунок 1.8-1.9).

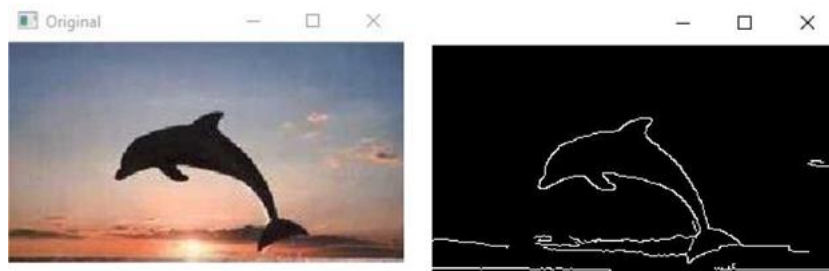


Рисунок 1.8 - а) вихідне кольорове зображення; б) результат виявлення країв

В результаті видно, що даний метод може знайти всі краї об'єкта (дельфін), видаляючи при цьому багато країв об'єктів на задньому фоні (хмари). Далі розглянемо складніший об'єкт і застосуємо той самий алгоритм (рисунок 1.9).



Рисунок 1.9 - а) вихідне кольорове зображення; б) результат виявлення країв об'єкта

З прикладів видно, що алгоритм виявлення країв Кенні, дає задовільний результат, як простих, так складніших об'єктів.

## 1.2 Згорткові нейронні мережі

Штучні нейронні мережі – це клас моделей машинного навчання, в основі яких лежать центральної нервової системи ссавців.

Нейронна мережа складається з кількох взаємопов'язаних різних шарів, таких як вхідний шар, щонайменше один прихований шар та вихідний шар (рисунок 1.10). Їх найкраще використовувати при виявленні об'єктів для розпізнавання образів, країв (вертикальні/горизонтальні), форми, кольору та текстури. Приховані шари є шарами згортки. У даному типі нейронної мережі згорткові шари діє як фільтр, який спочатку отримує вхідні дані, перетворює їх, використовуючи певний алгоритм або функцію, і відправляє його на наступний шар. Основні параметри нейронів є вхідний (синій колір) та вихідний шар (зелений колір).

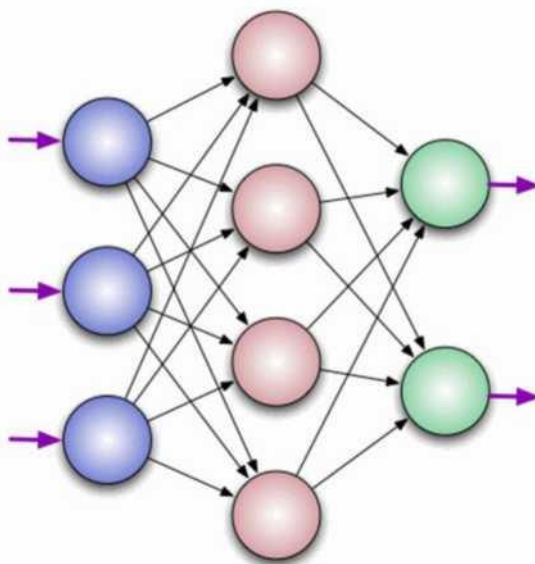


Рисунок 1.10 – Модель нейронної мережі

З великою кількістю згорткових шарів, кожен раз, коли новий вхід відправляється на наступний згортковий шар, він змінюється по-різному. Наприклад, у згортковому шарі фільтр може ідентифікувати форму/колір у певній області, останній згортковий шар, може класифікувати об'єкт.

У загальному випадку згорткова нейронна мережа складається з великої кількості шарів. На останніх етапах зазвичай використовується один чи кілька повнозв'язкових.

Згорткові нейронні мережі забезпечують часткову стійкість до змін масштабу, зсувів, поворотів, зміни ракурсу та інших спотворень. Згорткові нейронні мережі поєднують три архітектурні ідеї, для забезпечення інваріантності до зміни масштабу, повороту зсуву та просторових спотворень:

- локальні рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів);
- загальні синоптичні коефіцієнти (забезпечують детектування деяких характеристик у будь-якому місці зображення [19] і зменшують загальну кількість вагових коефіцієнтів);
- ієрархічна організація із просторовими підвиборками.

### 1.3 Розпізнавання об'єктів за допомогою нейронної мережі

Для того, щоб навчити нейронну мережу виявляти об'єкти на будь-якому зображенні, з приблизно однаковою формою і кольорами, слід застосовувати різні фільтри (рисунок 1.11).

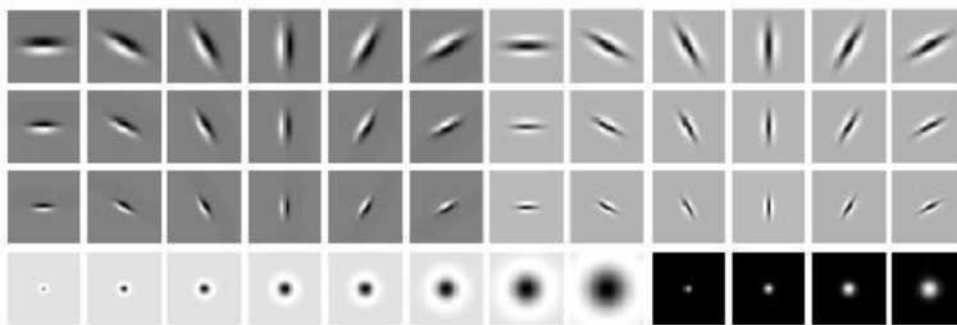
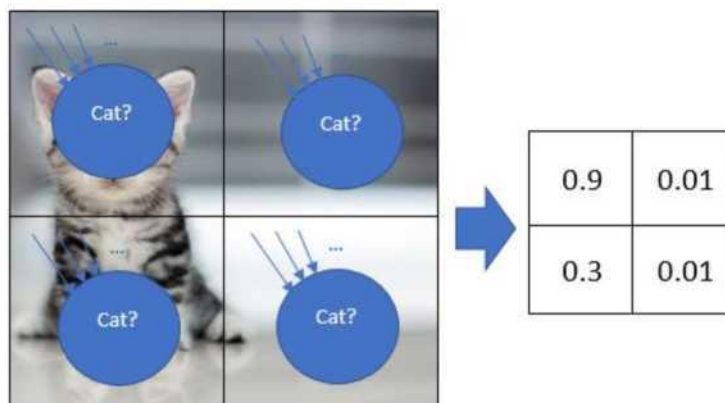


Рисунок 1.11 - Фільтри для різних фрагментів зображення

За допомогою різних фільтрів можна виділяти різні фрагменти зображення, які потім виходить виявити та дослідити у вигляді окремих властивостей та передавати іншими шарами нейрона (рисунок 1.11).

Для того, щоб мережам не доводилося окремо розпізнавати об'єкти в різних частинах зображення, ми розділяємо ваги, що відповідають за розпізнавання, між різними фрагментами вихідного зображення. Розглянемо зображення, на якому необхідно не просто виділити об'єкт, а встановити кількісну точність рішення чотирьох фрагментів (рисунок 1.12).



## Рисунок 1.12 - Розпізнавання об'єкта та характеристики точності рішення

У таблиці вказані кількісні характеристики рішення, що показують ступінь точності розпізнавання об'єкта у цьому фрагменті зображенні. Знаходимо максимальне значення  $\max\{0.9, 0.3, 0.01, 0.01\} = 0.9$ ,

яке і буде необхідною характеристикою (рисунок 1.13).

0.9	0.01
0.3	0.01



## Рисунок 1.13 – Отримання результату для виявленого об'єкта

Основна ідея алгоритму полягає в наступному:

- використовуємо поділ ваг (weight sharing) для створення «вікна, що фільтрує», що пробігає по зображенню.
- фільтр, що застосовується до зображення, допомагає виділити фрагменти, важливі для розпізнавання.
- у той час як у традиційному машинному зорі фільтри конструювали вручну, нейромережі дозволяють нам сконструювати оптимальні фільтри за допомогою навчання.
- фільтрування зображення можна поєднати з обчисленнями нейронної мережі.

### 1.3.1 Згортковий шар нейронної мережі

Згортковий шар є набір карт (карти ознак, features maps). Кожна карта має ядро, що сканує (скануюче ядро - являє собою фільтр, який ковзає по всьому зображенню і знаходить задані ознаки в будь-якому його місці). Кількість карток визначається вимогами до завдання, якщо взяти велику кількість карток, то підвищиться якість розпізнавання, але збільшиться обчислювальна складність.

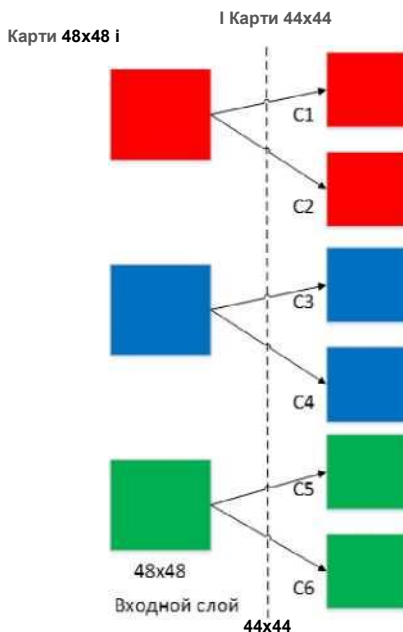


Рисунок 1.14 – Організація зв'язків між картами згорткового та вхідного (попереднього) шару

Згортка виконується наступним способом: вікном розміру ядра  $g$  проходимо із заданим кроком (зазвичай 1) все зображення  $f$ , кожному кроці поелементно множимо вміст вікна на ядро  $g$ , результат підсумовується і записується в матрицю результату, (рисунок 1.15).

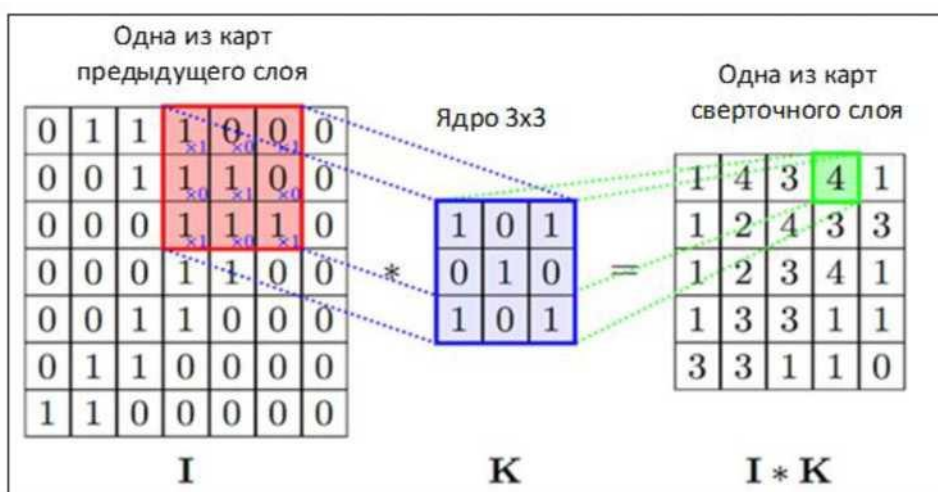


Рисунок 1.15 – Операція згортки та отримання значень карти ознак

### 1.3.2 Підвибірковий шар

Підвибірковий шар також, як і згортковий, має карти, але їх кількість збігається з попереднім (згортковим) шаром. Ціль використання даного шару - зменшити розмірності карт попереднього шару. Якщо попередньої операції згортки вже було виявлено деякі ознаки, то подальшої обробки настільки докладне зображення не потрібно, і воно ущільнюється до менш детального. До того ж, фільтрація вже непотрібних деталей допомагає не перевчитися.

У процесі сканування ядром підвибіркового шару (фільтром) карти попереднього шару, скануюче ядро не перетинається на відміну від згорткового шару. Зазвичай кожна карта має ядро розміром  $2 \times 2$ , що дозволяє зменшити попередні карти згорткового шару в 2 рази. Вся карта ознак поділяється на комірки  $2 \times 2$  елементи, у тому числі вибираються максимальні за значенням. Зазвичай, у підвибірковому шарі застосовується функція активації (ReLU, Rectifiedlinearunit). Операція підвиборки виконується (Max-Pool – вибір максимального) відповідно до рисунку 1.16.

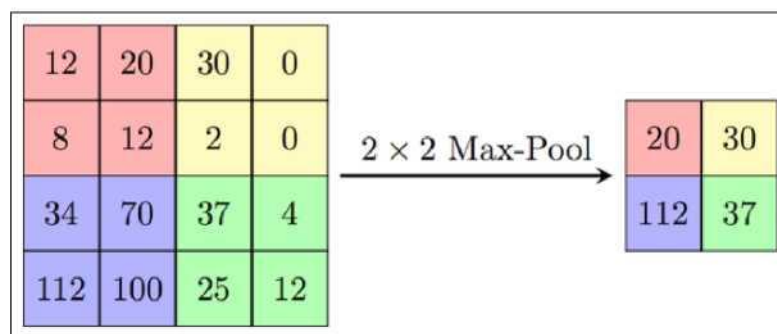


Рисунок 1.16 –Формування нової карти підвибіркового шару на основі попередньої карти згорткового шару. Операція підвиборки (MaxPooling)

### 1.3.3 Повнозв'язковий шар

Останній із типів шарів – це шар звичайного багатошарового персептрона. Мета застосування даного шару - це звернення до виходу попереднього шару та визначення

властивостей, які пов'язані з певним класом нейронів

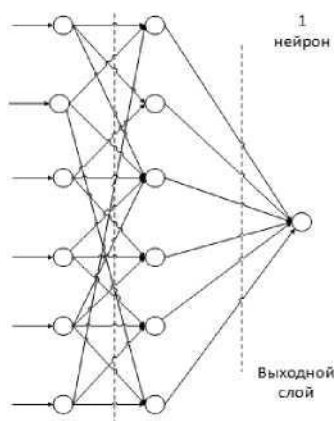


Рисунок 1.17 – Приклад повнозв'язкових шарів

Нейрони кожної карти попереднього підвиборчого шару пов'язані з одним нейроном прихованого шару. Таким чином число нейронів прихованого шару дорівнює числу карт підвибіркового шару, але зв'язки можуть бути не обов'язково такими, наприклад, тільки частина нейронів якоїсь із карт підвибіркового шару пов'язана з першим нейроном прихованого шару, а частина, що залишилася з другим, або всі нейрони першої карти пов'язані з нейронами 1 та 2 прихованого шару.

#### 1.4 Огляд програмних інструментів для створення веб-сервісу

Веб-сервіс – це стандартизоване середовище для встановлення зв'язку між клієнтськими та серверними програмами у всесвітньому павутинні (World Wide Web). Веб-сервіс є програмним модулем, призначеним для виконання певного набору завдань, такі як взаємодія між сайтами, обмін даними між різними програмами, які написані навіть різними мовами та розподілені на різних вузлах мережі, взаємодія з користувачами [3].

Для того, щоб веб-служба була повністю функціональна, потрібна наявність певних компонентів. Ці компоненти повинні бути присутні незалежно від того, яка мова розробки використовується для програмування [4]. Основним компонентом веб-

сервісу є дані, що передаються між клієнтом та сервером. Дані можуть бути розмічені з використанням мови розмітки (XML), яка є аналогом HTML, зрозумілий для проміжної мови.

У цій роботі як програмні інструменти для розробки веб-сервісу ми використовували такі мови програмування.

Мова гіпертекстової розмітки (HyperText Markup Language, HTML) – проста кросплатформова мова розмітки, яка використовується для створення веб-сайтів, гіпертекстових документів та документів на веб-сторінці. HTML створює систему для ідентифікації та опису різних компонентів документа (заголовки, абзаци, таблиці, списки), а також може представляти гіпертекстові новини, пошту, документацію та гіпермедіа (меню опцій, результати запиту до бази даних), прості структуровані документи з вбудованою графікою, гіпертекстові подання існуючих масивів інформації [5].

Каскадні таблиці стилів (CSS) – визначають зовнішній вигляд веб-сторінки, представленої для користувачів. За допомогою CSS можна легко та швидко керувати шрифтами, інтервалами між рядками, кольорами, фоновими зображеннями, макетами та іншими атрибутами оформлення елементів веб-сайту [6]. Каскадні таблиці стилів - це інструмент, що дозволяє автоматизувати процес розробки, а також управляти поданням документів не тільки в браузері, але під час друку. Крім того, в таблицях стилів існують правила для визначення не візуального подання документів, наприклад, як буде звучати текст документа при його програмному зчитуванні.

JavaScript інтерпретована, кросплатформена об'єктно-орієнтована мова програмування, яка підтримується більшістю веб-браузерів, наприклад, Chrome, Firefox, Internet Explorer, Opera, Safari, Edge та більшість мобільних браузерів та смартфонів [6, 7]. Мова JavaScript в основному використовується як мова сценаріїв для керування елементами вебсторінки, надаючи їм інтерактивність та варіативність поведінки:

- створення інтерфейсних віджетів, що розкриваються меню;
- перевірка коректності значень, введених у елементи форми;
- створення обробників подій, які відповідають за користувацькі та інші

події;

- забезпечує динаміку та зміну стилів оформлення елементів графічних елементів інтерфейсу;
- виконання вимог від браузера (наприклад, запам'ятати інформацію про користувача).

Django – веб-фреймворк, є набором модульних інструментів, які допомагають розробнику абстрагуватися від багатьох формальностей і рутинних операцій, властивих веб-розробці. Наприклад, більшість веб-сайтів потребують однакової базової функціональності: можливість підключатися до бази даних, задавати маршрути URL-адрес, відображати вміст на сторінці, правильно керувати безпекою тощо.

Фреймворк Django реалізований мовою Python і включає підтримку вирішення спільних завдань веб-розробника:

- автентифікація користувача;
- шаблони, маршрути та уявлення;
- інтерфейс адміністратора;
- забезпечення надійної безпеки;
- підтримка кількох баз даних.

Як основний програмний інструмент, обрана мова програмування Python. Ця мова програмування є кросплатформною і дозволяє працювати швидше та ефективніше інтегрувати в різних операційних системах. Більше того, Python широко застосовується у сфері машинного навчання, аналізу великих даних, комп'ютерного зору тощо.

В даний час існує кілька сучасних фреймворків для роботи із зображеннями, що підтримують глибоке навчання:

– TensorFlow – відкрита програмна бібліотека для машинного навчання, розроблена Google для вирішення завдань побудови та тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття [24].

- OpenCV – бібліотека з відкритим вихідним кодом, використовується для всіх

видів обробки та аналізу зображень і відео, яка включає кілька сотень алгоритмів комп'ютерного зору: інтерпретації зображень, калібрування камери по еталону, усунення оптичних спотворень, визначення подібності, аналіз переміщення об'єкта, визначення форми об'єкта і стеження за об'єктом, 3D- реконструкція, сегментація об'єкта, розпізнавання жестів тощо.

– Keras – це мінімалістична відкрита нейромережева бібліотека, написана мовою Python для глибокого навчання, яка може працювати поверх TensorFlow. Дана бібліотека націлена для оперативної роботи з мережами глибокого навчання, при цьому спроектована так, щоб бути компактною, модульною та розширюваною [1].

– NumPy – це бібліотека, з відкритим кодом для мови програмування Python, з підтримкою багатовимірних масивів (включаючи матриць) і підтримкою високорівневих математичних функцій, призначених для роботи з багатовимірними масивами [1].

## 1.5 Постановка задачі дослідження

На підставі проведеного аналізу сформульовано мету роботи: дослідження алгоритмів та методів штучних нейронних мереж для аналізу текстонів зображення та відеоінформації для розпізнавання 3D-об'єктів, дослідження методів комп'ютерного зору для розробки веб-сервісу, що виконує пошук та класифікацію текстонів на графічних зображеннях.

Для досягнення поставленої мети будуть вирішені такі задачі:

– провести огляд методів розпізнавання об'єктів, комп'ютерного зору та машинного навчання;

– дослідити методи та існуючі алгоритми для детектування та класифікації об'єктів, проаналізувати існуючі методи опису та визначення текстур та текстонів зображення, обґрунтувати використання згорткових нейронних мереж;

– представити архітектуру згорткової нейронної мережі для визначення текстонів зображення, виконати навчання нейронної мережі за допомогою алгоритму зворотнього розповсюдження помилки;

– розробити програмну реалізацію системи визначення текстонів зображення.

## 2. ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕТОДІВ ОПИСУ ТА ВИЗНАЧЕННЯ ТЕКСТУР ТА ТЕКСТОНІВ

### 2.1 Аналіз задачі класифікації та пошуку схожих зображень

Прикладом завдання класифікації є бінарна класифікація, коли є питання про зображення, яке припускає бінарну відповідь: так чи ні, «так», 0 – «ні».

Багатокласова класифікація – сімейство завдань, в яких необхідно відповісти на питання про зображення, який допускає одну з декількох відповідей, зазвичай безліч відповідей обмежена. Відповідь видається у вигляді ймовірності правильної відповіді, тобто абсолютної (еталонної) відповіді  $u \in [1, S]$  – дати мітку класу, якому відповідає це зображення, але оскільки алгоритм який завжди може це зробити, відповідь видається як ймовірності.

За допомогою завдань класифікації зображення можна визначати атрибути зображення. Атрибути – «типові» характеристики об'єкта.

До завдання класифікації відноситься визначення людини по особі, верифікація.

Найосновніше завдання розпізнавання людини – порівняння двох зображень. Схожим завданням є пошук зображень за змістом (content-based image retrieval), яка полягає в наступному: є колекція зображень, необхідно знайти в ній схоже зображення на зображення-запит, зображення порівнюються за візуальною подібністю. Таким чином, пошук зображення за змістом – це завдання класифікації з відкритим та наперед невідомим набором класів.

Загалом завдання класифікації зображень можна декомпонувати на такі етапи:

- отримання зображення;
- визначення ознак зображення;
- аналіз ознак;
- отримання відповіді.

Для того, щоб вирішувати задачу за допомогою машинного навчання,

необхідно:

- тестова (навчати класифікатор) та навчальна вибірка (оцінювати роботу класифікатора);
- критерій якості роботи
- метод побудови векторної ілюстрації.

Далі застосовується обраний метод машинного навчання побудови класифікатора.

Після складання великої еталонної колекції необхідно визначити критерій якості. Для цього використовуються такі метрики, як

- відсоток правильно класифікованих зображень (для якого відсотка тестової вибірки класифікатор спрацював правильно);
- рангова метрія Rank X;
- якщо класифікація багатокласова, ранжуються усі виходи за якістю;
- якщо справжня відповідь потрапляє у перші X виходів, тоді результат вважається вірним.

Далі слід визначити ознаки. При роботі із зображенням, які б ознаки не конструювалися, вони спиратимуться на ознаки, які можна обчислити за пікселями зображення:

- колір: простору кольорів RGB;
- градієнти у кожному пікселі;
- наявність та орієнтація краю в кожному пікселі.

Першими методами розпізнавання зображення, які використовували мішок слів, були методи аналізу текстури.

Схема методу «мішок слів»:

1. Побудова словника:
  - вилучення фрагментів з навчальної вибірки;
  - кластеризація та побудова «візуального» словника.
2. Побудова дескрипторів:
  - вилучення фрагментів із зображення;
  - квантування фрагментів за словником;

- побудова гістограми частот візуальних слів;

Текстура – властивість поверхні, переважна орієнтація елементів, що становлять матеріал.

Для аналізу текстур пропонується такий спосіб:

- вибирається згортка (фільтр), чутливий до країв певної орієнтації;
- результат фільтрації згладжується, посиливши сильні відгуки та придушивши слабкі відгуки;
- в результаті будуть «підсвічені» області, що містять текстуру із краями заданої орієнтації.

Щоб класифікувати безліч різних текстур, логічно згладжувати зображення не одним фільтром, а набором фільтрів, який буде чутливий до різних орієнтацій і не тільки до країв. Такий набір згорток називається банком фільтрів. Кожен піксель зображення після обробки банком фільтрів дає вектор ознак, який ефективно визначає локальну текстуру околиці пікселя. Тепер ці вектори можна кластеризувати і побудувати словник. Після кластеризації виходить набір класів.

Текстон – характерний базовий фрагмент текстури зображення. Можна проаналізувати зображення, зіставивши пікселу найближчий фрагмент зі словника за набором символів. Візуалізувати це можна за допомогою карти кольорів, зіставивши кожному текстону свій колір. Тепер як ознака області зображення можна брати частоту зустрічі текстонів в області і класифікувати, яка частина зображення відповідає текстурі.

## 2.2 Аналіз проблем опису текстур та текстонів

Текстурна сегментація на сьогоднішній день є одним з напрямків, що найбільш динамічно розвиваються в області розпізнавання образів і обробки зображень. Незважаючи на досить тривалий період розвитку, досі немає чіткого формального визначення текстури [25–28] і, як наслідок, досить чіткої класифікації типів текстур.

Основні підходи цього напрямку ще тільки формуються і не мають глибоко

опрацьованої та усталеної теоретичної бази. Активно досліджуються завдання пошуку текстурних ознак, визначення структуруючого елемента текстури (текстону, текселю), побудови опису текстури для компактного її представлення та можливості подальшої класифікації.

У більшості випадків для текстурної сегментації використовують маски, що відповідають текстуруючому елементу, статистичні ознаки, застосовуються також нейронні мережі, приховані марківські моделі, фрактальний аналіз, формальна граматики [26, 29-31].

Текстура – зв'язна область елементів цифрового зображення з різними яскравостями, колірними тонами та іншими величинами, що характеризують цифрове зображення, що візуально передає характер поверхні об'єкта вихідної сцени.

У всіх методів опису текстур загальним є те, що текстура є властивістю сусідства.

Текстон зазвичай визначають як найменшу область зображення текстурного типу, що передає характер та основні особливості текстури.

Методи, які використовують у явному чи неявному вигляді характеристики текстонів для розпізнавання текстур, можуть призводити до суттєвих спотворень поблизу межі текстурної області. Насамперед це пов'язано з тим, що на граничних областях можуть бути лише частини текстону, які не передають повністю характеру та основних особливостей текстури.

Таким чином, є необхідність пошуку методів, що використовують описи, що зберігають властивості текстури навіть при неповному текстоні.

### 2.3 Дослідження існуючих методів для аналізу текстур та текстон

В області комп'ютерного зору вилучення візуальних особливостей складається з математичних перетворень, розрахованих на пікселях цифрового зображення. Візуальні характеристики зазвичай дозволяють краще враховувати певні візуальні властивості зображення, які використовуються для подальшої обробки в рамках

додатків, таких як виявлення об'єктів або пошук зображень за вмістом. Для аналізу текстур та текстонів використовуються такі методи:

- розташування функцій;
- визначення загальних показників низького рівня;
- обчислення моментів кольору;
- побудова колірної автокорелограми;
- опис форм;
- визначення місцевих (локальних) показників;
- агрегування.

Розташування функцій. Як правило, розрізняють глобальні характеристики, які розраховуються по всьому зображенню, та локальні характеристики, які розраховуються навколо точок інтересу. Також можна говорити про напівлокальні характеристики, коли вони витягуються в обмежених областях зображення в результаті сегментації зображення по областях або просто відповідно до довільної сітки [32].

Відмінність між глобальними та локальними характеристиками представляє таксономічний інтерес. Локальні особливості відрізняються тим, що вони виразні, стійкі до оклюзій (оскільки їх багато на зображенні або області) і не вимагають сегментації [33]. Локальний дескриптор, який обчислюється для кожного пікселя зображення або області, отриманої шляхом сегментації, потім накопичується в гістограмі, тому є глобальним описом зображення або області.

Загальні характеристики низького рівня:

- колір;
- гістограма та похідні.

Аналіз кольору зображень історично одна з очевидних та перших характеристик, що використовуються для дослідження зображень на основі контенту, і досі у багатьох випадках дає найкращі результати на певних підставах. У 1991 році Суейн та Баллард запропонували використовувати кольорову гістограму для такого роду завдань [33]. Метод є загальним розподілом кольорів на зображенні. Його розрахунок складається з кількісної оцінки обраного колірному простору (наприклад,

RGB) з подальшим обчисленням гістограми перетворених таким чином пікселів. Наприклад, при розгляді класичного зображення RGB та кількісної оцінки кожної колірної площини по 4 осередках, отримана гістограма матиме розмір  $4 \times 4 \times 4 = 64$ . Якщо зображення спочатку закодовано на 24 біти (значення кожної площини знаходиться в інтервалі [0-255]), колірна гістограма (рисунок 2.1) на 64 осередках може бути представлена "кубом"  $4 \times 4 \times 4$ .

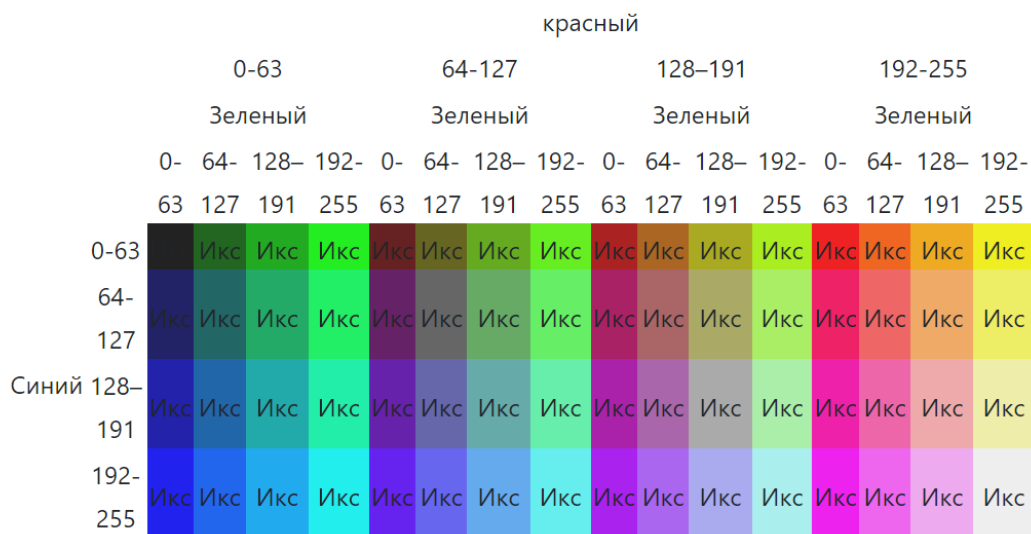


Рисунок 2.1 - Колірна гістограма

Кожен колір представляє один із 64 квантованих інтервалів, а  $x$  - кількість пікселів зображення, що відповідає цьому інтервалу. Насправді цей дескриптор зазвичай представлений одномірним 64-мерним вектором.

Багато дескрипторів можуть бути отримані з подання гістограми, деякі з яких були включені в експериментальну модель стандарту MPEG-7. До них, зокрема, належать:

- домінуючий колір – повідомляє про домінуючий колір зображення, розрахований як колір комірки з найвищим значенням;
- масштабований колір – це перетворення Хаара колірної гістограми, що призводить до дуже компактного коду;
- колірна структура – це «локальна» колірна гістограма. Його конструкція ідентична принципу, представленому раніше (гістограма глобального кольору), крім

того, що вікно зменшеного розміру ( $8 \times 8$  пікселів) проходить через зображення при підрахунку кольорів у кожному пікселі. Таким чином він відображає деяку інформацію про просторову структуру розподілу кольору.

– колірна схема – обчислюється дискретне косинусове перетворення подання гістограми, і лише частина коефіцієнтів може використовуватися для підпису. Це дає можливість отримати відносно грубий, але дуже компактний дескриптор або тонший дескриптор, який займає більше місця. Такий дескриптор цікавий для додатків, які потребують певної масштабованості.

Моменти кольору. У 1995 р. було запропоновано використати моменти, обчислені в заданому колірному просторі для характеристики зображень [34]. Як правило, використовуються чотири моменти (середнє значення, дисперсія, асиметрія, ексцес), але вони обчислюються в кожній із трьох площин обраного колірному простору. Одне з можливих удосконалень – розділити зображення по сітці та обчислити ці моменти у кожному осередку.

Колірна автокорелограма. Показує ймовірність того, що піксель на певній відстані від пікселя, що розглядається, пофарбований в конкретний колір.

Форма. Опис форм вимагає попереднього визначення регіонів. Це може бути результатом сегментації зображень або виявлення контурів. Потім можна охарактеризувати регіони за допомогою різних показників, таких як їхня основна орієнтація (наприклад, вісь), їхня симетрія.

Зокрема, використовується метод опису, запропонований Мохтаріаном, який називається простором шкали кривизни.[35]. Він полягає в описі області як функції зміни кривизни її контуру.

Цей опис інваріантний по відношенню до обертання та переміщення. Що стосується масштабування, необхідно стандартизувати хоча б по довжині контуру. Що ще важливіше, дескриптор зазвичай обчислюється в різних масштабах, кожен з яких відповідає пакунку по ядра Гауса.

Текстури. У певному масштабі натовп можна як текстуру. Визначення текстури є неочевидним і залежить від масштабу [36–37].

Існує три підходи до визначення поняття текстури:

- перший підхід розглядає текстури як результат випадкового процесу та намагається описати їх з урахуванням статистичних властивостей значень та відносного положення пікселів[38];
- другий підхід передбачає існування фундаментальних примітивів, що дозволяють описувати текстури, такі як текстони, визначені Джулесом[38]. Отже, ця текстура описується як складна комбінація цих примітивів, виражена, наприклад, за допомогою графіків;
- третій, пізніший підхід заснований на візуальному сприйнятті і має тенденцію враховувати процес формування текстури з людської точки зору (спектральний аналіз, дискретне косинусове перетворення)[39].

Розглянемо ці підходи детальніше.

Стохастичний підхід. Текстуру можна описати статистично, враховуючи те, як рівні сірого зображення організуються один до одного.

Іншим широко використовуваним статистичним підходом є модель MSAR (моделі одночасної авторегресії з кількома дозволами), яка, крім розпізнавання текстур, використовується для розпізнавання природних сцен.

Структурний підхід передбачає наявність фундаментальних примітивів для опису текстур. Одним із перших прикладів таких примітивів є текстони, запропоновані Джулесом [12]. Ця текстура описується як складна комбінація цих примітивів, виражена, наприклад, за допомогою графіків. Ці графіки можна використовувати як дескриптори текстури.

Спектральний підхід. Вираз періодичності та інших закономірностей у зображенні чи сигналі природно відбувається у рамках спектрального аналізу. Таким чином, дискретне перетворення Фур'є зображення може бути основою створення дескрипторів. Однак найчастіше використовується набір фільтрів Габора, розташованих у кількох масштабах та орієнтаціях. Це дозволяє зокрема ідентифікувати «основні риси сприйняття». Дескриптор, що ґрунтується на такому спектральному описі, може пояснити домінуючу просторову структуру природної сцени.

Альтернативою перетворенню Фур'є є обчислення дискретного косинусового

перетворення (DCT). На практиці дескриптори, що ґрунтуються на коефіцієнтах DCT, дозволяють розрізняти внутрішні та зовнішні зображення, зображення міських ландшафтів порівняно з природними ландшафтами та, у поєднанні з іншими дескрипторами, одночасно кілька категорій природних сцен.

Місцеві властивості. Характеристика зображення може бути розрахована на обмеженій кількості пікселів. Для цього спочатку необхідно виявити цікаві області зображення, а потім обчислити характеристичний вектор у кожній з цих областей. Такими областями є, наприклад, краї або точки зображення, що виступають (області з високою контрастністю). Це можуть бути точки, взяті випадково чи регулярно на зображенні (так звана щільна вибірка).

Вектор ознак іноді містить дані виявлення, такі як орієнтація краю або величина градієнта в області, що цікавить. Зазвичай характеристичний вектор пікселі обчислюється в околиці цього пікселя, тобто із зображення з центром в цьому пікселі. Його можна розрахувати в різних масштабах, щоб позбавитися коефіцієнта масштабування. Серед локальних характеристик, що розраховуються в даний час, є шаблони, що раніше використовувалися глобально, такі як гістограми кольору або вектори, що враховують орієнтацію градієнтів рівнів сірого.

Три приклади околиць, що використовуються для визначення текстури та обчислення локального двійкового шаблону (LBP) представлені на рисунку.

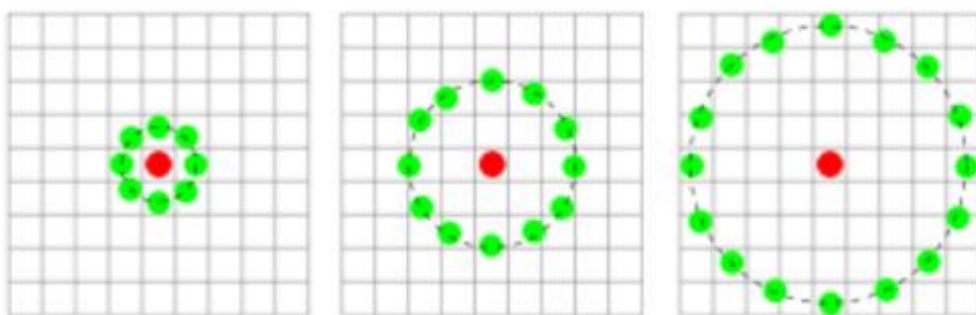


Рисунок 2.2—Приклади околиць, що використовуються для визначення текстури

Деякі методи, такі як масштабно-інваріантна трансформація ознак (Scale-invariant Feature Transform, SIFT) або прискорені стійкі ознаки (Speeded-up Robust

Features, SURF) включають як виявлення області, що цікавить, так і обчислення характеристичного вектора в кожній з цих областей. Для визначення характеристичного вектора SIFT є гістограмою орієнтації градієнта, а SURF складається з обчислення наближення вейвлету Хаара. Аналогічно, локальні бітові шаблони є одночасне обчислення локальних рівнів сірого зображення, а гістограма спрямованих градієнтів (Histogram of Oriented Gradients, HOG) – це орієнтовані градієнтні гістограми, дуже схожі на SIFT, розраховані з використанням щільної вибірки.

Агрегування. Опис зображення за допомогою локальних характеристик зазвичай має змінний вимір, що залежить від кількості вилучених точок інтересу (фактично, ця кількість множиться на розмірність локального дескриптора). Таким чином, таке уявлення не підходить для опису традиційно використовуваних алгоритмів навчання (метод опорних векторів – SVM, прискорення). Щоб повернутись до представлення у векторному просторі фіксованої розмірності, агрегування використовує методи агрегування дескрипторів, такі як мішки слів (Bag of visterms). Таким чином, результатом такого накопичення є загальна характеристика зображення чи його частини (області).

Характеристики зображень, витягнутих описаними вище методами, використовуються у багатьох сферах комп'ютерного зору.

Пов'язані з анотованими зображеннями, вони можуть бути вхідними даними для контрольованого алгоритму навчання для класифікації. Це, зокрема, може використовуватися для виявлення їхнього вмісту і, зокрема, об'єктів, присутніх на зображенні. У тому ж ключі, але в більш конкретних областях вони можуть використовуватися для виявлення осіб або людей.

Ці дескриптори, які можуть бути підписом зображення, також використовуються при пошуку зображень за змістом.

Це базові технології, що використовуються в багатьох областях, де задіяний комп'ютерний зір: робототехніка, відеоспостереження, машинний зір, оптичне розпізнавання символів, виділення текстур і текстонів зображення.

Проте, всі розглянуті методи у сфері комп'ютерного зору мають недоліки, тому

найактуальнішим нині є використання машинного навчання та нейронних мереж на дослідження зображень.

У 2012 році міжнародна оцінна кампанія ImageNet Large Scale Visual Recognition Challenge була виграна з використанням підходу згорткової мережі (CNN), який був вивчений завдяки нещодавнім досягненням у галузі глибокого навчання. Поліпшення показників особливо помітне порівняно з попередніми роками. Незабаром після цього показано, що рівень згорткової мережі, попередньо вивчений в ImageNet, може використовуватися як вектор представлення зображення відповідно до схеми передачі навчання і, таким чином, досягати найвищої продуктивності. З цього періоду у переважній більшості робіт у цій галузі використовуються вектори ознак, вивчені на природних зображеннях за допомогою згорткової мережі, а «позначені вручну» особливості стають дуже незначними, за винятком областей дуже специфічного застосування.

Використання згорткової нейронної мережі буде розглянуто для дослідження застосування текстур і текстонів зображення.

### 3 МОДЕЛІ ЗАСТОСУВАННЯ ТЕКСТОНІВ ЗОБРАЖЕННЯ

#### 3.1 Побудова згорткової нейронної мережі для опису текстонів зображення

Робота із зображеннями – важлива сфера застосування технологій Machine Learning. Глобально всі зображення з усіх камер світу становлять бібліотеку неструктурованих даних. Задіявши нейромережі, машинне навчання та штучний інтелект, ці дані структурують та використовують для виконання різних завдань: побутових, соціальних, професійних та державних, зокрема, забезпечення безпеки.

Основою всіх архітектур для роботи із зображеннями є аналіз, першою фазою якого буде розпізнавання зображення (об'єкта). Потім штучний інтелект за допомогою машинного навчання розпізнає об'єкти та класифікує їх.

Найактуальніше завдання нейромереж – розпізнавання візуальних образів. Згорткова нейронна мережа має спеціальну архітектуру, яка дозволяє їй максимально ефективно розпізнавати образи. Сама ідея згорткової нейронної мережі ґрунтується на чергуванні згорткових (convolution) та субдискретизуючих шарів (pooling), а структура є односпрямованою. Згорткова нейронна мережа отримала свою назву від операції згортки, яка припускає, що кожен фрагмент зображення буде помножений на ядро згортки поелементно, при цьому отриманий результат повинен підсумовуватися та записуватись у схожу позицію вихідного зображення. Така архітектура забезпечує інваріантність розпізнавання щодо зсуву об'єкта, поступово укрупнюючи «вікно», на яке «дивиться» згортка, виявляючи дедалі більші структури та патерни в зображенні.

Згорткова нейронна мережа складається з різних видів шарів: згорткові (convolutional) шари, субдискретизуючі (subsampling, підвибірка) шари та шари «звичайної» нейронної мережі – перцептрона. Перші два типи шарів (convolutional, subsampling), чергуючись між собою, утворюють вхідний вектор ознак для багат шарового перцептрона.

Вхідними даними є зображення (тип: зображення, розмір: 48x48, формат: RGB, в градаціях сірого). Вихідними даними є кількість класів (два). Розв'язуване завдання – визначення текстонів зображення.

Вхідні дані являють собою кольорові зображення типу JPEG, розміром 48x48 пікселів. Якщо розмір буде занадто великий, то обчислювальна складність підвищиться, відповідно обмеження на швидкість відповіді будуть порушені, визначення розміру цього завдання вирішується методом підбору. Якщо вибрати розмір занадто маленький, мережа не зможе виявити ключові ознаки текстонів. Кожне зображення розбивається на три канали: червоний, синій, зелений. Таким чином виходить три зображення розміром 48x48 пікселів.

Вхідний шар враховує двовимірну топологію зображень і складається з декількох карт (матриць), карта може бути одна, у тому випадку, якщо зображення представлене у відтінках сірого, інакше їх три, де кожна карта відповідає зображенню з конкретним каналом (червоним, синім та зеленим) .

Згортковий шар є набором карт (інша назва - карти ознак, це звичайні матриці), у кожній карті є синаптичне ядро (скануюче ядро або фільтр).

Кількість карт визначається вимогами до завдання, якщо взяти велику кількість карт, то підвищиться якість розпізнавання, але збільшиться обчислювальна складність. Виходячи з аналізу наукових статей, в більшості випадків пропонується брати співвідношення один до двох, тобто кожна карта попереднього шару (наприклад, у першого згорткового шару, попереднім є вхідний) пов'язана з двома картами згорткового шару.

Підвибірковий шар також, як і згортковий має карти, але їх кількість збігається з попереднім (згортковим) шаром, їх шість. Мета шару – зменшення розмірності карт попереднього шару. Якщо у попередній операції згортки вже було виявлено деякі ознаки, то подальшої обробки настільки докладно зображення не потрібне, і воно ущільнюється до менш детального.

Останні шари – шари звичайного багат шарового перцептрона. Два повнозв'язних та один вихідний. Цей шар відповідає за класифікацію, з математичної точки зору моделює складну нелінійну функцію, оптимізуючи яку покращується

якість розпізнавання. Число нейронів у шарі шість за кількістю карт ознак підвиборного шару.

Повнозв'язний шар – останній з типів шарів – це шар звичайного багатозарового перцептрона. Мета шару – класифікація, моделює складну нелінійну функцію, оптимізуючи яку, покращується якість розпізнавання.

Нейрони кожної карти попереднього шару пов'язані з одним нейроном прихованого шару. Таким чином, число нейронів прихованого шару дорівнює числу карт підвибіркового шару, але зв'язки можуть бути не обов'язково такими, наприклад, тільки частина нейронів будь-якої з карт підвибіркового шару пов'язана з першим нейроном прихованого шару, а частина, що залишилася з другим, або всі нейрони першої карти пов'язані з нейронами 1 та 2 прихованого шару.

Вихідний шар пов'язаний із усіма нейронами попереднього шару. Кількість нейронів відповідає кількості класів, що розпізнаються, тобто два - текстон (повторювана ділянка зображення) і не текстон.

Архітектура згорткової нейронної мережі для визначення текстонів зображення представлена на рисунку 3.1. та у таблиці 3.1

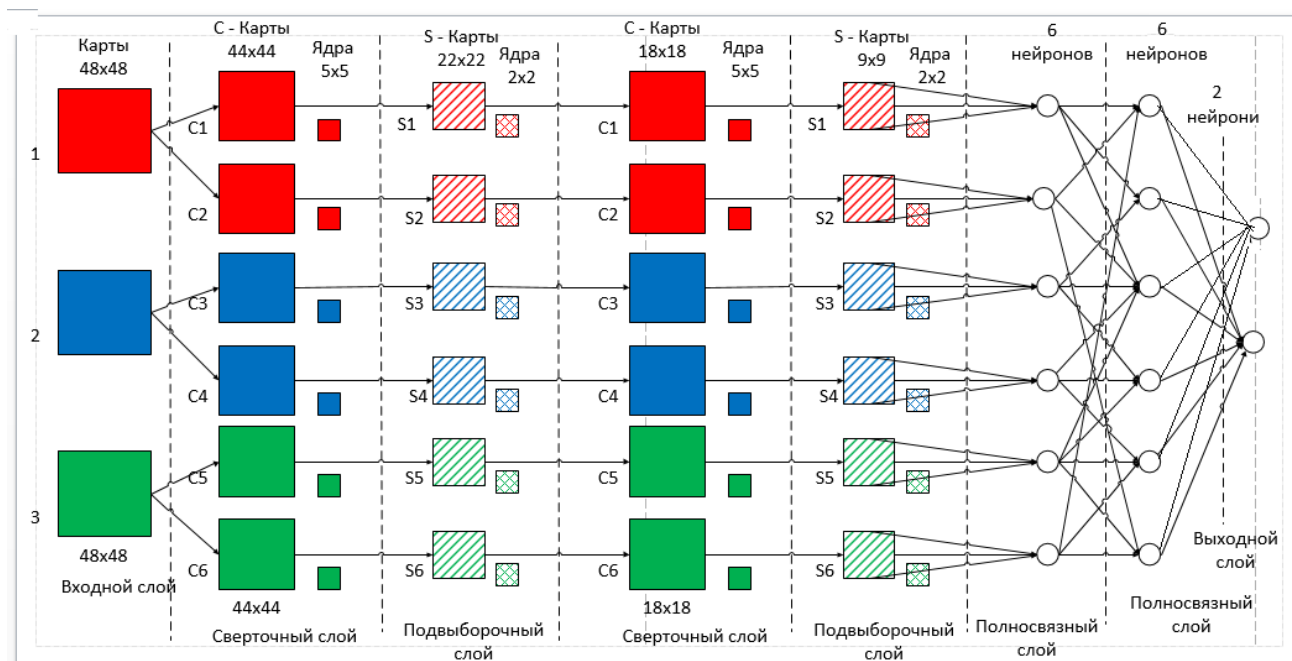


Рисунок 3.1– Архітектура згорткової нейронної мережі для визначення текстонів зображення

Таблиця 3.1. – Архітектура згорткової нейронної мережі для визначення текстур зображення

Номер шару	Тип шару	Кількість карт	Розмір шару (входу)	Розмір ядра згортки
1	Вхідний	3	48*48	
2	Згортковий (convolution)	6	44*44	5*5
3	Поєднуючий (pooling)	6	22*22	2*2
4	Згортковий (convolution)	6	18*18	5*5
5	Поєднуючий (pooling)	6	9*9	2*2
6	Повнозв'язаний (filter concatenate)	-	6*1	-
8	Кінцевий повністю пов'язаний шар (fully bonded output layer)	-	6*1	-
9	Вихідний шар	-	2*1	-

Для того, щоб розпізнати зображення, нейронна мережа має бути навчена на даних. Нейронні мережі вимогливі до розміру та якості набору даних, на якому вона навчатиметься.

У загальному сенсі під навчанням згорткової нейронної мережі розуміється послідовне пред'явлення образу на вхід нейронної мережі з набору бібліотеки даних, наприклад, TensorFlow, потім отримана відповідь порівнюється з бажаним виходом. Наприклад, «1» – образ представляє текстон (повторювана ділянка зображення), «-1» – образ представляє не текстон, отримана різниця між очікуваною відповіддю та отриманим є результатом функції помилки (дельта помилки). Потім цю дельту помилки необхідно поширити на пов'язані нейрони мережі.

На практиці до певної межі, чим більше прихованих шарів у нейронній мережі, тим точніше буде розпізнане зображення. Це реалізується так: картинка розбивається на маленькі ділянки, до кількох пікселів, кожен із яких є вхідним нейроном. За допомогою синапсів сигнали передаються від одного шару до іншого. Під час цього процесу сотні тисяч нейронів із мільйонами параметрів порівнюють отримані сигнали

з уже обробленими даними.

Наприклад, при розпізнаванні зображення, можна розділити його на маленькі фрагменти та порівнювати ці шари з мільйонами вже наявних зображень текстонів, значення ознак яких мережа вивчила. У якийсь момент збільшення числа шарів призводить до запам'ятовування вибірки, а не навчання.

Для розпізнавання зображень нейронна мережа працює за певними етапами.

Як розпізнані образи розглянемо текстуру зображення. Під час навчання мережі їй пропонуються різні зразки текстонів з позначкою того, до якого саме типу їх можна віднести. Як зразок застосовується вектор значень ознак, а сукупність ознак повинна дозволити однозначно визначити клас зразків.

Важливо під час навчання навчити мережу визначати достатню кількість і значення ознак, щоб видавати хорошу точність на нових текстурах, але й не перенавчитися, тобто, надмірно не «підлаштуватися» під навчальну вибірку з зображень. Після завершення правильного навчання нейронна мережа повинна вміти визначати текстури тих класів, з якими вона не мала справу в процесі навчання.

Важливо враховувати, що вихідні дані для нейромережі повинні бути однозначними і несуперечливими, щоб не виникали ситуації, коли НС видаватиме високі ймовірності приналежності одного об'єкта до кількох класів.

Створення згорткової нейронної мережі для розпізнавання текстур та текстонів зображень включає:

- збір та підготовку даних;
- вибір топології;
- підбір параметрів;
- підбір параметрів навчання;
- навчання;
- перевірку якості навчання;
- коригування;
- вербалізацію.

Розпізнавання зображень за допомогою нейронних мереж можливе лише за допомогою спеціального навчання, що є процесом, спрямованим на налаштування

параметрів нейронної мережі. Є кілька способів навчити згорткову нейромережу:

- машинне навчання з учителем;
- навчання без учителя;
- навчання із частковим залученням вчителя;
- навчання із підкріпленням.

При навчанні нейронної мережі для розпізнавання зразків з учителем є вибірка з істинними відповідями на питання, що зображено мітками класів. Нейронні мережі подаються на вхід ці зображення, після чого обчислюється помилка, що порівнює вихідні значення з точками класів. Залежно від ступеня та характеру невідповідності передбачення нейронної мережі, її ваги коригуються, відповіді мережі підлаштовуються під справжні відповіді, доки помилка стане мінімальною. Для побудови нейронної мережі для аналізу текстур та текстонів використовується навчання з учителем.

При навчанні без вчителя у навчальній вибірці немає міток класів, і перед нейронною мережею стоїть завдання знайти заздалегідь не відомі відповіді. Нейронна мережа намагається самостійно знайти закономірності у даних, витягуючи корисні ознаки та аналізуючи їх. Наприклад, кластеризація — найпоширеніша задача навчання без вчителя. Алгоритм підбирає схожі дані, знаходячи загальні ознаки і групує їх разом.

У навчанні без вчителя складно обчислити точність алгоритму, оскільки відсутні «правильні відповіді» чи мітки, але розмічені дані буває складно чи занадто дорого отримати. У таких випадках, надаючи моделі свободу дій для пошуку залежностей, можна отримати певний результат.

Під час навчання з частковим залученням вчителя навчальна вибірка містить як розмічені, і нерозмічені дані. Цей метод корисний, коли розмітити всі об'єкти — трудомістке завдання. Тим не менш, нейронна мережа може отримати інформацію з невеликої частки розмічених даних і покращити точність передбачень у порівнянні з моделлю, що навчається виключно на нерозмічених даних.

Навчання з підкріпленням (reinforcement learning) діє за принципом отримання зворотного зв'язку – нагороди за певні дії.

Навчання запропонованої згорткової нейронної мережі зводиться до мінімізації функції помилки шляхом коригування вагових коефіцієнтів синаптичних зв'язків між нейронами. Під функцією помилки розуміється різниця між отриманою відповіддю та бажаною. Наприклад, на вхід був поданий образ текстону, припустимо, вихід нейромережі був 0,73, а бажаний результат 1 (т.к. образ текстону), отримаємо, що помилка мережі є різницею, тобто 0,27. Потім ваги вихідного шару нейронів коригуються відповідно до помилки. Для нейронів вихідного шару відомі їх фактичні та бажані значення виходів. Тому налаштування ваг зв'язків для таких нейронів є простим. Однак для нейронів попередніх шарів налаштування не настільки очевидне. Тривалий час був невідомий алгоритм поширення помилки по прихованих шарах.

### 3.2 Навчання згорткової нейронної мережі з використанням алгоритму зворотного розповсюдження помилки

Для навчання описаної нейронної мережі використовується алгоритм зворотного поширення помилки (backpropagation). Цей метод навчання багат шарової нейронної мережі називається узагальненим дельта-правилом. Метод було запропоновано у 1986 р. Румельхартом, Макклеландом та Вільямсом. Цей алгоритм є першим і основним практично застосовним для навчання багат шарових нейронних мереж.

Для вихідного шару коригування ваг інтуїтивно зрозуміле, але для прихованих шарів довгий час не було відомо алгоритму. Ваги прихованого нейрона повинні змінюватися прямо пропорційно до помилки тих нейронів, з якими даний нейрон пов'язаний. Тому зворотне поширення цих помилок через мережу дозволяє коректно налаштовувати ваги зв'язків між усіма шарами. Величина функції помилки зменшується і мережа навчається.

Розглянемо алгоритм зворотнього розповсюдження помилки для навчання згорткової нейронної мережі.

У загальному випадку задача навчання нейронної мережі зводиться до

знаходження функціональної залежності  $Y = f(X)$ , де  $X$  - вхідний, а  $Y$  - вихідний вектори.

У загальному випадку така задача, при обмеженому наборі вхідних даних, має безліч рішень. Для обмеження простору пошуку при навчанні ставиться завдання мінімізації цільової функції помилки, яка знаходиться за методом найменших квадратів

$$E_p = 1/2 \sum_{j=1}^p (t_{pj} - y_{pj})^2,$$

де  $E_p$  – величина функції помилки для образу  $p$ ;

$t_{pj}$  – бажаний вихід нейрона  $j$  для образу  $p$ ;

$y_{pj}$  – значення активованого виходу нейрона  $j$  для образу  $p$ .

Неактивований стан кожного нейрона  $j$  для образу  $p$  можна записати у вигляді зваженої суми

$$s_{pj} = \sum_i w_{ij} * y_{pi},$$

де  $s_{pj}$  – зважена сума виходів зв'язаних нейронів попереднього шару на вагу зв'язку або неактивований стан нейрона  $j$  для образу  $p$ ;

$w_{ij}$  – вага зв'язку між  $i$  та  $j$  нейронами;

$y_{pi}$  – значення активованого стану нейрона  $i$  попереднього шару для образу  $p$ .

Вихід кожного нейрону  $j$  є значенням активаційної функції  $f_j$ , яка переводить нейрон у активований стан. Як функцію активації може використовуватися будь-яка безперервно диференційована монотонна функція. Активований стан нейрона обчислюється виразом

$$y_{pj} = f_j(s_{pj}),$$

де  $f_j$  – функція активації.

Як метод мінімізації помилки використовується метод градієнтного спуску, суть цього методу зводиться до пошуку мінімуму (або максимуму) функції за рахунок

руху вздовж вектору градієнта. Для пошуку мінімуму рух має здійснюватися у напрямку антиградієнта. Метод градієнтного спуску зображений на рисунку 3.2.

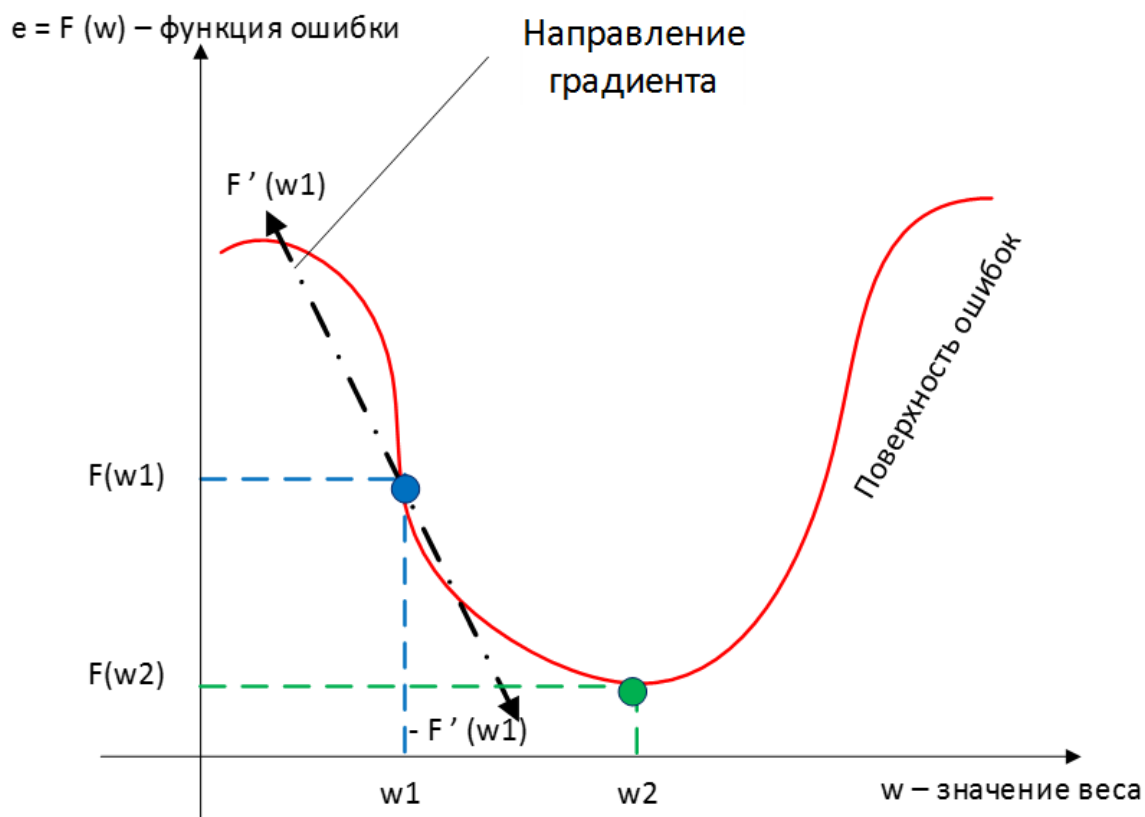


Рисунок 3.2 – Метод градієнтного спуску

Градієнт функції втрати є вектором приватних похідних, що обчислюється за виразом

$$\Delta E(W) = \left[ \frac{dE}{dw_1}, \dots, \frac{dE}{dw_n} \right],$$

де  $\Delta E(W)$  – градієнт функції втрати від матриці ваг;

$\frac{dE}{dw}$  – приватна похідна функції помилки по вазі нейрона;

$n$  – загальна кількість вагів мережі.

Похідну функції помилки за конкретним образом можна записати за допомогою виразу

$$\frac{dE}{dw_{ij}} = \frac{dE}{dy_j} * \frac{dy_j}{dS_j} * \frac{dS_j}{dw_{ij}},$$

де  $\frac{dE}{dw_{ij}}$  – значення похідної функції помилки за вагою  $w_{ij}$ , між  $i$  та  $j$  нейронами;

$\frac{dE}{dy_j}$  – помилка нейрону  $j$ ,  $\delta = \frac{dE}{dy_j}$ ;

$\frac{dy_j}{dS_j}$  – значення похідної функції активації по її аргументу для нейрона  $j$ ;

$\frac{dS_j}{dw_{ij}}$  – вихід  $i$  нейрона попереднього шару (по відношенню до нейрона  $j$ ).

Для вихідного шару помилка визначена у явному вигляді, тобто різниця між бажаним та отриманим виходом. Для того, щоб розрахувати помилку прихованих шарів використовується алгоритм зворотного поширення помилки. Суть його у послідовному обчисленні помилок прихованих шарів з допомогою значень помилки вихідного шару, тобто. Значення помилки розповсюджуються по мережі у зворотному напрямку від виходу до входу.

Помилка  $\delta$  для прихованого шару розраховується використовуючи вираз

$$\delta_i = \frac{dy_j}{dS_j} * \sum_j \delta_j * w_{ij},$$

де  $\delta_i$  – помилка нейрону  $i$  прихованого шару;

$\delta_j$  – помилка нейрону  $j$  наступного шару;

$w_{ij}$  – вага зв'язку між  $i$  (поточним) та  $j$  нейронами вихідного або прихованого шару.

## 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ

4.1 Побудова та моделювання згорткової нейронної мережі для визначення текстонів зображення

Для програмної реалізації використовується Python 10.0 та бібліотеки Keras для створення та навчання згорткової нейронної мережі, представленої в роботі.

Перевагами використання Python у порівнянні з C++ або Java є простота, лаконічність, швидка розробка та специфіка використання для формалізації технологій глибокого машинного навчання, що дозволяють з мінімальними витратами тимчасових та людських ресурсів розробляти складні алгоритми, нейронні мережі різних архітектур та алгоритмів навчання. Мова має потужний механізм інтероперабельності з C++, що відкриває доступ до швидких обчислень. Крім того, нейронні мережі, як правило, описуються програмами не дуже великими за обсягом, але вимагають внесення змін для практичного підбору оптимальної архітектури, передобробки даних.

Keras – це високорівневий API нейронних мереж, написаний на Python і здатний працювати поверх відкритих програмних бібліотек для машинного навчання для вирішення завдань побудови та навчання нейронних сетей з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття (TensorFlow, CNTK, Theano). Keras розроблено з упором на можливість швидкого експериментування. Здатність йти від ідеї до результату з найменшою можливою затримкою є ключем до проведення досліджень [40-42].

Бібліотека глибокого навчання Keras дозволяє:

- легко і швидко створювати прототипи (завдяки зручності, модульності та розширюваності);
- підтримує згорткові мережі, що повторюються, а також ансамблі мереж;
- легко працює на процесорі та графічному процесорі.

Метою роботи є правильна класифікація елементів зображення на текстон або не текстон.

У процесі роботи з датасетом необхідно виконати такі дії:

- завантажити зображення та мітки класу з диска;
- розділити дані на навчальну та тестову вибірки;
- побудувати згорткову нейронну мережу згідно з запропонованою в п.3.1

архітектурою;

- навчити нейромережу Keras за допомогою алгоритму зворотного розповсюдження помилки (п.3.2);

- оцінити модель на тестовій вибірці;

- використовувати навчену модель нових даних.

Програмний код побудови згорткової нейронної мережі знаходиться у додатку А.

Першим етапом є підключення необхідних модулів та виконується попередня обробка даних.

Далі виконується побудова згорткової нейронної мережі:

- побудова карт згорткового шару, визначення розміру ядра згортки;
- побудова карток субдескриптуючого (об'єднуючого) шару, визначення

розміру ядра згортки;

- побудова повнозв'язного шару;

- побудова вихідного шару.

Також для кожного шару визначається активаційна функція (сигмоїда, ReLu (Rectified Linear Unit) – функції активації, що найчастіше використовуються при глибокому навчанні.

На наступному етапі виконується налаштування алгоритму зворотного розповсюдження помилки для побудованої нейронної мережі.

Наступним кроком виконується тестування нейронної мережі на навчальному наборі даних. За результатами тестової вибірки можна оцінити одержану модель.

Результат роботи програми представлений нижче у програмному коді, що представлений на рисунку 4.1.

```

Using TensorFlow backend.
[INFO] loading images...
[INFO] training network...
Train on 2250 samples, validate on 750 samples
Epoch 1/75
2250/2250 [=====] - 1s - loss: 1.1033 - acc: 0.3636 -
val_loss: 1.0811 - val_acc: 0.3707
Epoch 2/75
2250/2250 [=====] - 0s - loss: 1.0882 - acc: 0.3862 -
val_loss: 1.1292 - val_acc: 0.3227
Epoch 3/75
2250/2250 [=====] - 0s - loss: 1.0713 - acc: 0.4067 -
val_loss: 1.0525 - val_acc: 0.3907
...
Epoch 73/75
2250/2250 [=====] - 0s - loss: 0.7780 - acc: 0.6067 -
val_loss: 0.8438 - val_acc: 0.5813
Epoch 74/75
2250/2250 [=====] - 0s - loss: 0.7805 - acc: 0.5978 -
val_loss: 0.8463 - val_acc: 0.5893
Epoch 75/75
2250/2250 [=====] - 0s - loss: 0.7765 - acc: 0.6262 -
val_loss: 0.8144 - val_acc: 0.6733

0.6733
[INFO] evaluating network...
precision  recall   f1-score   support

texton      0.78      0.72      0.80      236
not texton      0.56      0.67      0.56      236
avg / total      0.67      0.69      0.68      472

[INFO] serializing network and label binarizer...

```

#### Рисунок 4.1 – Лістинг коду розробленої програм

Оскільки мережа невелика (як і набір даних) цей процес у середньому займає близько двох секунд. Нейромережа дала результати з точністю 68%. Можна стверджувати, що мережа фактично вивчила шаблони, які можна використовувати для розрізнення двох класів.

#### 4.2 Результат роботи програми

В роботі на основі проведених досліджень розглянута практична задача пошуку

людей після надзвичайної події. На рисунку, яке представлено, око людини може без зусиль визначити дві людини, але завдяки текстонам на даному фото можна знайти ще дві особи, які потребують невідкладної допомоги, в чому полягає практична цінність даної задачі.



Рисунок 4.1 –Зображення зі знайденими текстонами

На рисунку 4.2 наведений приклад навчання згорткової нейронної мережі для пошуку текстонів зображення.



Рисунок 4.2 –Приклад навчання згорткової нейронної мережі для пошуку текстонів зображення

## ВИСНОВКИ

Кваліфікаційна робота виконано згідно з методичними вказівками до виконання та захисту кваліфікаційної роботи другого (магістерського) рівня вищої освіти [43].

Мета роботи – дослідження методів комп'ютерного зору та методів машинного навчання для розробки програмної системи, що виконує пошук та класифікації текстонів на графічних зображеннях (фото-, відео).

Для досягнення поставленої мети у роботі поставлені та вирішені наступні завдання:

- проведено огляд методів розпізнавання об'єктів, комп'ютерного зору та машинного навчання;
- досліджено методи та існуючі алгоритми для детектування та класифікації об'єктів, проаналізовано існуючі методи опису та визначення текстур та текстонів зображення, обгрунтовано використання згорткових нейронних мереж;
- представлено архітектуру згорткової нейронної мережі для визначення текстонів зображення, виконано навчання нейронної мережі за допомогою алгоритму зворотнього розповсюдження помилки;
- розроблено програмну реалізацію системи визначення текстонів зображення.

Згорткові мережі – один з найуспішніших існуючих на сьогоднішній день алгоритмів розпізнавання зображень.

Моделям, що реалізують відповідну архітектуру, належать перші місця у змаганнях алгоритмів розпізнавання, таких як ImageNet: згорткові мережі розпізнають рукописні цифри вибірки MNIST з 0,23% помилок, людські особи з 2,4% помилок (використовувалась вибірка розміром у 5600 фотографій 10 людей); результат змагання ImageNet для переможця 2014 – згорткової мережі GoogLeNet компанії Google становить 0,06656% помилкових розпізнавань, що являє собою найкращий результат на поточний момент і порівняно з помилкою, що допускається

людиною на вибірці ImageNet.

Серед недоліків виділяють труднощі при обробці маленьких об'єктів, і нездатність справлятися з спотвореннями, такими як фільтр, що розмиває, або сильний шум (такі спотворення присутні в навколишньому світі, наприклад, при погляді через товсте скло).

При цьому згорткові мережі порівняно легко справляються з проблемами високоточного розпізнавання, які викликають складнощі у людей, наприклад, розпізнавання окремих моделей машин або порід собак, та інші завдання, які потребують виділення вузькоспецифічних ознак.

Однією з основних особливостей згорткових мереж є той факт, що така модель не має інформації про те, як саме локалізовано зображений шуканий об'єкт – його конкретному місцезнаходженню та орієнтації в просторі.

При цьому у вирішенні прикладних завдань управління та обробки інформації, знання параметрів локалізації є необхідною умовою – залежно від розташування або пози об'єкта система обробки інформації може класифікувати зображення по-різному відповідно до покладених на неї завдань.

З іншого боку, без знання параметрів локалізації, згорткова мережа вразлива до узагальненому класу проблем цілісності, тобто. модель, навчена деяким локальним ознаками зображеного об'єкта, позитивно класифікуватиме штучні зображення, де відповідні елементи розташовані в хаотичному порядку – помилка, яку легко уникає людина.

Операція згортки забезпечує інваріантність по відношенню до трансляції локальних ознак (зсуву по осях  $x$  та  $y$ ).

Архітектура згорткових мереж не передбачає стійкості до інших афінних перетворень, таких як обертання, дзеркальне відображення та масштабування.

Для вирішення цієї проблеми зазвичай використовуються евристичні методи (вирівнювання зображення по лінії горизонту, використання просторових пірамід та різних відбитих копій оригіналу).

Перетворення, які не є афінними по відношенню до плоского зображення, такі як обертання об'єкта в тривимірній площині, аналогічним чином не можуть бути

оброблені за допомогою згортки. Для об'єктів, представлених у різних орієнтаціях, моделі доводиться вчитися різним наборам ознак, що не перетинаються.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Джуллі А., Пал С. (2018) «Бібліотека Keras – інструмент глибокого навчання». / Пров. з англ. Слінкін А. А. – М.: ДМК Прес, – 294 с.
2. Содем Я. Еге (2016) «Програмування комп'ютерного зору мовою Python». / Переклад з англ. Слінкін А. А.,- 332 с.
3. Веб-сервіс та її архітектура. URL: <https://www.guru99.com/web-service-architecture.html> (дата звернення 07.11.2021)
4. Компоненти веб-сервісів. URL: [https://www.tutorialspoint.com/webservices/what\\_are\\_web\\_services.html](https://www.tutorialspoint.com/webservices/what_are_web_services.html) (дата звернення 07.11.2021)
5. Mashtalir S. & Mashtalir V. (2016). Sequential temporal video segmentation via spatial image partitions. International Journal “Information Theories and Applications”, pp. 50-59.
6. Дженніфер Р. Н. (2018). HTML5, CSS3 та JavaScript. Вичерпне керівництво. Перев. з англ. М. А. Райтман, 288 с.
7. Mashtalir S., Mashtalir V., Stolbovyi M. (2017). Video shot boundary detection via sequential clustering.
8. Mashtalir, S. V., Stolbovyi, M. I., & Yakovlev, S. V. (2019). Clustering Video Sequences by the Method of Harmonic k-Means. Cybernetics and Systems Analysis, 55(2), 200-206.
9. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). Advances in Spatio-Temporal Segmentation of Visual Data (Vol. 876). Springer Nature.
10. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2017). Video shot boundary detection via sequential clustering. International Journal “Information Theories and Applications, 24(1), pp. 50-59.
11. Zhang, J., Zhao, Z., Xue, Y., Chen, Z., Ma, X., & Zhou, Q. (2017). Time series analysis. Handbook of Medical Statistics, 269.

12. Bodyanskiy, Y. V., Tyshchenko, O. K., & Mashtalir, S. V. (2019, June). Fuzzy Clustering High-Dimensional Data Using Information Weighting. In International Conference on Artificial Intelligence and Soft Computing pp. 385-395. Springer, Cham.
13. Cao Z., Tomas Simon, Shih-En Wei Yaser, Sheikh The Robotics Institute, Carnegie Mellon, University: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields URL: <https://arxiv.org/pdf/1611.08050.pdf> (дата звернення 16.11.2021)
14. Колірна модель RGB. URL: [http://delovoygorod.com/e\\_rgb.html](http://delovoygorod.com/e_rgb.html) (дата звернення 17.11.2021)
15. Джуллі А., Пал С. (2018) «Бібліотека Keras – інструмент глибокого навчання». / Пров. з англ. Слінкін А. А. - М: ДМК Прес.
16. Голиков І. Згортьова нейронна мережа URL : <https://habr.com/ua/post/348000/> (дата звернення 17.11.2021)
17. Мальцев А. Кілька слів про розпізнавання об'єктів URL: <https://habr.com/ua/post/208090/> (дата звернення 17.11.2021)
18. OpenCV: Detecting Edges, Lines, і Shapes URL: <https://hub.packtpub.com/opencv-detecting-edges-lines-shapes/> (дата звернення 13.11.2021)
19. Ніколенко С., Кадурін А., Архангельська Є. (2018). «Глибинне навчання». - СПб.: Пітер. - 480 с.
20. Нульовий параметр, автоматичне визначення країв Canny з Python та OpenCV URL: <https://www.pyimagesearch.com/2015/04/06/zero-parameter-automatic-canny-edge-detection-with-python-and-opencv/> (дата звернення 12.11.2021)
21. Mashtalir V., Ruban I., Levashenko V. (2020) Advances in Spatio-Temporal Segmentation of Visual Data. Springer
22. Офіційний ресурс фреймворку СакТе URL: <http://caffe.berkeleyvision.org/> (дата звернення 13.11.2021)
23. Офіційний ресурс фреймворку TensorFlow URL: <https://www.tensorflow.org/> (дата звернення 14.11.2021)
24. Офіційний ресурс додатку SpyderIDE URL: <https://www.spyder-ide.org/> (дата звернення 14.11.2021)

25. Jain AK, Karu K. // Image Analysis and Processing: 8th International Conferences. Lecture notes in computer science. 1995. Vol. 974. P. 3-10.
26. Noriega L., Westland S. (2001). Pattern Recognition and Information Processing: Minsk, 2001. p. 121-125.
27. Zhou F., Feng J., Shi Q. (2001). Pattern Recognition and Information Processing: Proc. of 6th Intern. Conf. Minsk, p. 41-45.
28. Абламейко С.В., Лагуновський Д.М. (2000). Обробка зображень: технології, методи, застосування. Мінськ.
29. Козирев С.А. (1993). Розпізнавання образів та аналіз зображень: Наук.-техн. конф. країн СНД. Мінськ, с. 202-204.
30. Мотль В.В., Мучник І.Б., Іванова Т.О., Блінов А.Б. (1996). Огляд прикладної та промислової математики. Т. 3, вип. 1. С. 104-108.
31. Xu, J., Han, J., Xiong, K., & Nie, F. (2016, July). Robust and Sparse Fuzzy K-Means Clustering. In IJCAI pp. 2224-2230.
32. Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). Introduction to time series analysis and forecasting. John Wiley & Sons.
33. Jain AK, Karu K. // Image Analysis and Processing: 8th International Conferences. Lecture notes in computer science. 1995. Vol. 974. P. 3-10.
34. Женбинг, Х., Бодянский, Е. В., Тыщенко, А. К., & Ткачев, В. Н. (2017). Fuzzy Clustering Data Arrays with Omitted Observations.
35. Старовойтов В.В. Локальні геометричні методи цифрової обробки та аналізу зображень. Мінськ, 1997.
36. Гонсалес Р., Вудс Р. (2006). Цифрова обробка зображень.
37. Форсайт Д., Понс Ж. (2004) Комп'ютерний зір. Сучасний підхід.
38. Khachumov, M. V. (2012). Distances, metrics and cluster analysis. Scientific and Technical Information Processing, 39(6), 310-316..
39. Julesz V. (2009) Текстони, елементи сприйняття текстури та їх взаємодії. сс: 91-97.
40. Формати зображень. URL: <http://blogmonet.ru/chem-otlichayutsya-formaty-izobrazhenij-bmp- png-jpg-gif.html> (дата звернення 14.11.2021)

41. Набір даних CIFAR-10 URL: <https://www.cs.toronto.edu/~kriz/cifar.html>  
(дата звернення 14.11.2021)
42. Deep Learning for Computer Vision - Introduction to Convolution Neural Networks // [Електронний ресурс].- <https://www.analyticsvidhya.com/blog/2016/04/deep-learning-computer-vision-introduction-convolution-neural-networks/>  
(дата звернення 17.11.2021).