

Міністерство освіти і науки України Харківський національний
університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Програмно-технічні комп'ютерні засоби розпізнавання контуру
біологічних об'єктів з використанням штучного інтелекту
(тема)

Виконав: студент 2 курсу, групи СКСм-18-2

Негода М.С.
(прізвище, ініціали)

Спеціальність
6.050102 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані
комп'ютерні системи
(повна назва освітньої програми)

Керівник проф. Кривуля Г.Ф.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Чумаченко С.В.
(підпис) (прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Автоматизації проектування обчислювальної техніки
Рівень вищої освіти другий (магістерський)
Спеціальність 123 – Комп'ютерна інженерія
Тип програми Освітньо-професійна
Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові Негоді Миколі Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Програмно-технічні комп'ютерні засоби розпізнавання контуру біологічних об'єктів з використанням штучного інтелекту

затверджена наказом по університету від 04 листопада 2019 р. № 1624 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20__ р.

3. Вихідні дані до роботи _____

Операційна система Windows 10,

Мова програмування Python,

Бібліотека NumPy, OS, Pillow, Matplotlib,

Архів зображень

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз предметної галузі та постановка задачі

Аналіз методів обробки графічних зображень

Розробка модулів розпізнавання зображень

Тестування системи розпізнавання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

15 слайд

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	02.09.2019-08.09.2019	
2	Аналіз предметної області	09.09.2019-15.09.2019	
3	Аналіз джерел з проблемної галузі	16.09.2019-29.09.2019	
4	Розробка алгоритму обробки графічних зображень	30.09.2019-13.10.2019	
5	Розробка модулів розпізнавання зображень	14.10.2019-20.10.2019	
6	Розробка нейронної мережі	21.10.2019-27.10.2019	
7	Тестування системи розпізнавання	28.10.2019-03.11.2019	
8	Оформлення пояснювальної записки	04.11.2019-01.12.2019	
9	Оформлення графічного матеріалу	02.12.2019-08.12.2019	

10	Перевірка виконаного проекту керівником	09.12.2019-15.12.2019	

Дата видачі завдання

Студент _____

(підпис)

Керівник роботи _____ проф. Кривуля Г.Ф.

(підпис)

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 60 сторінок, 28 рисунків, 13 джерел за переліком посилань.

ЗОБРАЖЕННЯ, РОЗПІЗНАВАННЯ, PУТНОН, АЛГОРИТМ, МЕТОД

Метою роботи є синтез інтелектуальних програмно-технічних комп'ютерних засобів для розпізнавання зображень біологічних об'єктів за їх контуром.

Об'єктом дослідження є процеси розпізнавання зображень біологічних об'єктів.

Предметом дослідження є моделі, методи та програмні засоби розпізнавання біологічних об'єктів в умовах поганої видимості.

Наукова новизна роботи полягає у підвищенні якості розпізнавання контуру людини в умовах поганої видимості, з використанням методів штучного інтелекту.

Практична цінність роботи полягає в можливості автоматизувати процеси розпізнавання біологічного об'єкту за його контуром, що дає змогу заощадити ресурси при охороні державного кордону, будинків та інших важливих об'єктів.

Результатом роботи є програмне забезпечення, де в якості вхідних даних виступає зображення, на якому містяться лише контури об'єкту, а в якості вихідних – відповідь, в якій вказується класифікація об'єкта.

ABSTRACT

The explanatory note contains 60 pages, 28 drawings, 13 references.

IMAGE, RECOGNITION, PYTHON, ALGORITHM, METHOD

The purpose of the work is to synthesize intelligent software and computer software for recognizing images of biological objects along their contour.

The object of the study are the processes of image recognition of biological objects.

The subject of the study are models, methods and software for recognizing biological objects in poor visibility.

The scientific novelty of the work is to improve the quality of recognition of the contour of the person in conditions of poor visibility, using the methods of artificial intelligence.

The practical value of the work is the ability to automate the processes of recognition of the biological object along its contour, which allows to save resources while protecting the state border, buildings and other important objects.

The result is software where the image is an input containing only the contours of the object, and the output is an answer indicating the classification of the object.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	Ошибка! Закладка не определена.
ВСТУП.....	8
1 РОЗПІЗНАВАННЯ ОБ'ЄКТІВ	9
1.1 Проблеми розпізнавання об'єктів	9
1.2 Загальний алгоритм класифікації об'єктів	11
1.3 Методи підготовки та обробки зображення.....	13
2 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ТА АЛГОРИТМІВ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ	17
2.1 Вилучення ознак об'єктів	17
2.1.1 Методи знаходження контуру	17
2.1.2 Метод Віюлі-Джонса.....	18
2.1.3 Метод гістограм спрямованих градієнтів.....	20
2.1.4 Локальні бінарні шаблони.....	21
2.2 Класифікація	22
2.2.1 Метод k-найближчих сусідів	22
2.2.2 Метод опорних векторів	24
2.2.3 Дерево ухвалення рішень	26
2.2.4 Методи штучного інтелекту	26
3 НЕЙРОННІ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТА ПО КОНТУРУ	27
3.1 Нейронні мережі як метод штучного інтелект	27
3.2 Модель нейрона	30
3.3 Архітектура нейронних мереж	31
3.4 Обробка вхідних та вихідних параметрів	34
3.5 Багатошаровий перцептрон.....	35
3.6 Математичний опис нейронної мережі	36

3.7 Розпізнавання об'єкту за допомогою нейронної мережі та границ	39
4 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ	39
4.1 Мова програмування	40
4.2 Модуль NumPy	41
4.3 Модуль OS	42
4.4 Модуль Pillow	43
4.5 Модуль Matplotlib	43
4.6 Схема алгоритму розпізнавання	44
4.7 Структура зберігання інформації	45
4.8 Опис функцій для роботи з зображеннями	46
4.9 Опис функцій для роботи з нейронною мережею	48
5 ТЕСТУВАННЯ	51
5.1 Тестування функції розпізнавання границь об'єкта	51
5.2 Розпізнавання простого зображення	53
5.3 Тестування системи розпізнавання об'єктів	55
ВИСНОВКИ	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	60
ДОДАТОК А Графічна частина проекту	61
ДОДАТОК Б Матеріали публікації	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

HOG – Histogram of Oriented Gradients (гістограма спрямованих градієнтів);

LBP – Local Binary Pattern (локальні бінарні шаблони);

SVM – Support Vector Machine (метод опорних векторів);

ЕОМ – електронна обчислювальна машина;

ПК – персональний комп'ютер;

ФНЧ – фільтр нижніх частот;

ФВЧ – фільтр верхніх частот.

ВСТУП

Потреба в автоматизованому вирішенні багатьох видів завдань стає все більш актуальною. На сьогоднішній день обчислювальні можливості комп'ютерів дають змогу реалізувати ті алгоритми, які були розроблені ще в 50-х роках минулого століття, чи навіть раніше, але не були впроваджені через їх велику обчислювальну складність. Поєднання алгоритмів та їх вдосконалення разом з сучасними комп'ютерами дає змогу заощадити велику кількість ресурсів.

В епоху технологічної революції розробки однієї галузі часто тягнуть за собою прориви в іншій. Подібно до промислової революції XIX століття, останні десятиліття можна назвати епохою комп'ютерної революції. Протягом багатьох років спостерігається стрімкий розвиток мікрочіпів, що призводило до постійного зростання обчислювальної потужності при все менших витратах. Це було підкріплено нещодавніми розробками паралельних обчислень обробки графіки. Усі ці апаратні досягнення також відкривають нові області застосування та можливості у прагненні змусити комп'ютер розпізнавати те що він “бачить” [1].

Однією з галузей, яка потребує застосування сучасних комп'ютерних систем є розпізнавання об'єктів. Теорія розпізнавання образу – розділ інформатики та суміжних дисциплін, що розвиває основи і методи класифікації та ідентифікації предметів, явищ, процесів, сигналів, ситуацій, об'єктів, які характеризуються кінцевим набором деяких властивостей і ознак [2]. Системи розпізнавання образу можуть бути використані для розпізнавання осіб, мови, зображень, літер, штрих-кодів, автомобільних номерів. Розпізнавання об'єктів – це першочергова мета в області комп'ютерного зору. Однак, хоча швидкі комп'ютери дуже допомагають у цьому відношенні, але насправді важливими є нові та кращі методи обробки та їх реалізація [1].

Велика кількість інформаційного потоку, який потрібно обробити людині, сприяє покращенню та вдосконаленню алгоритмів розпізнавання. Сьогодні вже не здається дивним, коли за допомогою камери на кордоні двох країн можна ідентифікувати особу за її обличчям, чи автомобіль, який може рухатися без водія. У всіх цих прикладах, а особливо в військовій справі чи системах безпеки точність розпізнавання повинна бути дуже високою. Нерідко ця точність повинна зберігатися і при умовах поганої видимості: при поганому освітленні чи погодніх умовах.

Виявлення об'єктів буде корисним для інтелектуальних середовищ, наприклад, системи спостереження можуть автоматично ідентифікувати зловмисників або людей похилого віку, які потребують допомоги, а розумні автомобілі можуть використовувати виявлення предметів або людей у поєднанні з іншими сигналами, щоб уникнути зіткнень. Подальші сфери застосування включають ігри, робототехніку, розваги, рекламу та виробництво – будь-які програми, де інтелектуальним системам потрібно спостерігати або взаємодіяти з людьми, предметами чи тваринами [3].

1 РОЗПІЗНАВАННЯ ОБ'ЄКТІВ

1.1 Проблеми розпізнавання об'єктів

Ключовою проблемою теорії розпізнавання є дослідження, розробка та реалізація методів розпізнавання за допомогою навчання алгоритмічних процедур по перетворенню та аналізу інформації, що призначені для вирішення таких завдань розпізнавання, для яких відповідні алгоритми невідомі, а само розпізнавання складається з комбінації декількох алгоритмів. Завдання, що вимагають використання таких методів, виникають у зв'язку з обробкою і перетворенням на ЕОМ структур, утворених з символів, тобто

структур, що представляють в програмах штучного інтелекту знання про проблемну область у цілому і знання, які стосуються конкретного завдання [4]

Люди та тварини виконують задачу розпізнавання об'єктів без особливих зусиль у повсякденному житті, щоб рухатися без зіткнень, знаходити їжу, уникати загроз тощо. Люди здатні інтерпретувати візуальну інформацію, розмежовувати подібні категорії об'єктів, наприклад, мотоцикл та велосипед, ідентифікувати об'єкти одного класу, незважаючи на дуже різні форми чи зовнішність. Психофізичні дослідження говорять про те, що брічна дитина може розпізнати 10-30 тисяч категорій об'єктів [5]. Однак подібні комп'ютерні методи та алгоритми для аналізу не настільки прості, незважаючи на їх безпрецедентний розвиток та злагожені зусилля дослідників протягом кількох десятиліть. Тим не менш, біологічні системи після ретельних спостережень та аналізу дають певні підказки для їх машинної реалізації. Гарним прикладом є штучні нейронні мережі, які за своєю різноманітністю нагадують біологічні системи нейронів і які за своєю програмною реалізацією часто використовуються комп'ютерами для розпізнавання об'єктів [1] .

Існує велика кількість проблем, від яких залежить успіх в процесі розпізнавання. Так, якість зображення залежить від освітленості, параметрів камери та місця розташування камери. Якщо брати до уваги технічні проблеми, які найчастіше можуть заважати в процесі розпізнавання об'єкта, можна виділити:

- освітлення лише з однієї сторони, або взагалі його відсутність може зашкодити в процесі знаходження границь об'єкту або виділення особливостей зображення;
- більшість алгоритмів є чутливими до повороту зображень, що впливає на здатність розпізнавання;

- часткове перекриття предметів або наявність на зображенні сторонніх об'єктів можуть спровокувати хибні відповіді алгоритму;
- зміна розміру об'єкта може викликати труднощі при розпізнаванні, оскільки алгоритми потребують певних тренувальних даних.

Виходячи з вищесказаного можна зробити висновок, що складність розпізнавання залежить від того, чи будуть зроблені знімки у подібних умовах (освітленість, фон, параметри камери та точка зору), як у моделях. Умови різко впливають на зображення того самого об'єкта. За різних умов продуктивність різних детекторів ознак буде значно відрізнятись. Щоб визначити, які особливості можна ефективно та надійно виявити, слід враховувати характер фону, інших об'єктів та освітленість.

Також до однієї з проблем системи розпізнавання можна віднести невелику кількість об'єктів у базі даних моделей. Якщо кількість вивчених об'єктів дуже мала, відповідна система може не помітити закономірність і хибно сформулювати гіпотезу про об'єкт.

На зображенні також може бути декілька об'єктів, які можуть бути розпізнані системою. Зі збільшенням кількості об'єктів на зображенні збільшується ймовірність неправильно сформулювати відповідь. Ця проблема призводить до відсутності очікуваних особливостей та генерування хибних результатів. Її вирішення потрібно розглядати на етапі перевірки гіпотез. Як правило, складність завдання розпізнавання зростає із кількістю об'єктів на зображенні.

1.2 Загальний алгоритм класифікації об'єктів

Система розпізнавання об'єктів повинна мати такі компоненти для виконання завдання:

- база даних моделей;
- детектор ознак;

- компонент побудови гіпотези;
- компонент верифікації гіпотези.

Структурна схема, що показує взаємодію та потік інформації між різними компонентами системи наведена на рисунку 1.1.

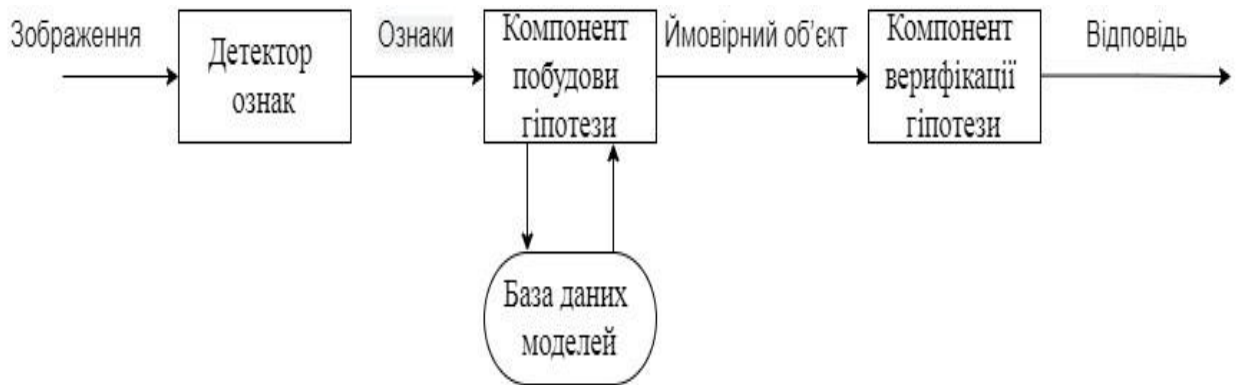


Рисунок 1.1 – Структурна схема системи розпізнавання об'єктів

База даних моделей містить усі моделі, які відомі системі. Інформація в базі даних моделей залежить від підходу, який використовується для розпізнавання. У багатьох випадках моделі об'єктів є абстрактними векторами ознак. Особливістю є деякий атрибут об'єкта, який вважається важливим при описі та розпізнаванні відносно інших об'єктів: розмір, колір, форма.

Детектор ознак застосовує оператор до зображення, який допомагає формувати гіпотезу. Функції, які використовує система, залежать від типу об'єктів, які слід розпізнати, та організації модельної бази даних.

Використовуючи виявлені особливості на зображенні, компонент побудови гіпотези призначає ймовірності об'єктам, присутнім на зображенні. База моделей організована за допомогою певного типу схеми індексації для полегшення вилучення ймовірних кандидатів з існуючих. Після чого верифікатор використовує об'єктні моделі для перевірки гіпотез та уточнення ймовірності об'єктів та вибирає об'єкт з найбільшою ймовірністю на основі всіх доказів.

Майже всі системи розпізнавання об'єктів використовують моделі та детектори ознак. Деякі системи не використовують компонент побудови гіпотези, а після компоненту формування гіпотези обирається об'єкт з найбільшою ймовірністю. Класифікація зображень є гарним прикладом такого підходу. З іншого боку, багато систем штучного інтелекту мало покладаються на формування гіпотез і виконують більше роботи на етапах перевірки. Залежно від складності проблеми один або кілька компонентів системи розпізнавання об'єктів можуть стати тривіальними [6].

1.3 Методи підготовки та обробки зображення

Проблема виявлення об'єктів є надто складною для аналітичного моделювання, тому тут доречно використовувати підхід, який заснований на навчанні, де різноманітний набір зображень використовується як приклад для моделі. Таким чином, виявлення об'єктів представляється як проблема класифікації потенційних кандидатів, які запропонувала основна модель. Машинне навчання використовується для вивчення правила прийняття рішень для цих гіпотез з репрезентативного навчального набору. Перш ніж використовувати зображення в якості навчального, потрібно підготувати його, виділити найважливіші характеристики. Вибір набору функцій для роботи з зображенням є вкрай важливим. Він повинен охоплювати найбільш істотні подібності між об'єктами одного класу та відмінності з об'єктами других класів, незважаючи на зміненій зовнішній вигляд об'єкта, позу, освітлення, фонову текстуру [3].

Для вирішення проблеми розпізнавання контурних зображень може бути застосоване автоматичне виявлення закономірностей в наборі даних із застосуванням комп'ютерних алгоритмів на основі методів штучного інтелекту. У більшості випадків вхідні змінні попередньо обробляються, щоб перетворити їх в новий простір змінних, де проблема розпізнавання образів

матиме більш просте рішення. Для цього доцільно використовувати методи фільтрації і підготовки зображень, методи логічної обробки результатів фільтрації, алгоритми прийняття рішень на основі логічної обробки зображень [7].

До одного із методів підготовки зображення можна віднести бінаризацію. Зображення складається із набору значень інтенсивності пікселів. Щоб інтерпретувати його, необхідно проаналізувати зміни цих значень. Найбільш часто для відтінків сірого використовується 256 рівнів. Однак, можна зустріти цифрові зображення до 32, 64, 128 або 512 і навіть до 4096 рівнів інтенсивності. Очевидно, що більш високі рівні інтенсивності дозволяють краще представити об'єкт. На початку розвитку машинного зору пам'ять і обчислювальні можливості були дуже обмеженими і дорогими. Ці обмеження змусили дослідників зосередити свої зусилля на бінарних системах. Бінарне зображення містить тільки два рівня сірого (рис 1.). Крім того, слід відзначити, що люди розуміють лінійні малюнки, силуети і інші зображення, сформовані з використанням тільки двох рівнів сірого. Вони також, як правило, дешевше і швидше, ніж системи відеоспостереження, які кольорових зображеннях.

Також до методів підготовки зображення можна віднести фільтр низьких частот (рисунок 1.2) та фільтр верхніх частот (рисунок 1.3). ФВЧ – фільтр, що пропускає високі частоти вхідного сигналу і послаблює сигнали з частотою меншою, ніж частота зрізу [8] та ФНЧ – фільтр, який пропускає низькі частоти, та послаблює частоти, розташовані вище частоти відсікання фільтру [9].



Рисунок 1.2 – Фільтр низьких частот



Рисунок 1.3 – Фільтр верхніх частот

Часто для покращення результатів обробки зображення використовується математична морфологія. Математична морфологія – це теорія, яка надає ряд корисних інструментів для аналізу зображень. Цей підхід базується на припущенні, що зображення складається з структур, якими може керуватися теорія множин. Для обробки використовується спеціальне бінарне зображення, яке називають примітивом або структурним елементом (рисунок 1.4). Зазвичай примітив набагато менший від вхідного зображення і має розмір 3x3 або 5x5 пікселів. До операцій математичної морфології можна віднести: ерозію, нарощування, розмикання і замикання.

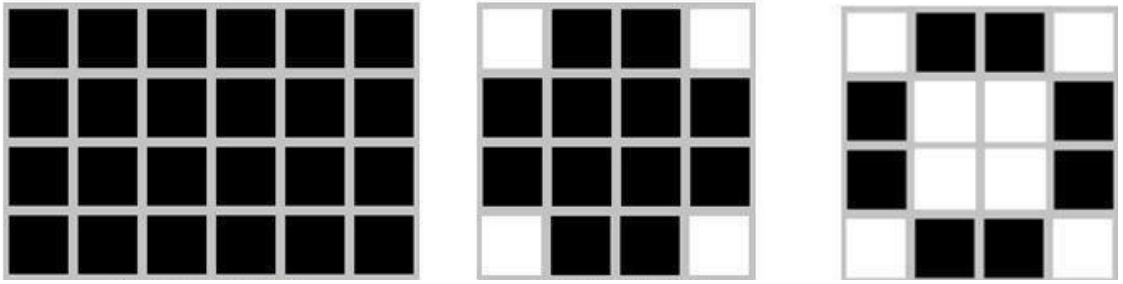


Рисунок 1.4 – Структурні елементи математичної морфології

2 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ТА АЛГОРИТМІВ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ

2.1 Вилучення ознак об'єктів

На ранніх етапах обробки зображення визначаються їх особливості, які мають значення для оцінки структури та властивостей об'єкту. До методів, якими можна виділити особливості об'єкта можна віднести: знаходження контуру, метод Віюлі-Джонса, метод гістограм спрямованих градієнтів та інші.

2.1.1 Методи знаходження контуру

Краї – одна з особливостей об'єкту. Як правило, вони виникають на межі між двома різними областями зображення, що є важливими ознаками для аналізу. Виявлення країв часто є першим кроком у відновленні інформації із зображення. Завдяки своїй важливості цей метод продовжує залишатися активною дослідницькою сферою.

Край на зображенні – це значна локальна зміна пікселів, зазвичай пов'язана з різким переходом їх інтенсивності. Розриви інтенсивності зображення можуть бути або ступінчастими, коли інтенсивність зображення різко змінюється від одного значення на одній стороні розриву до іншого значення на протилежному боці, або у вигляді лінії, де інтенсивність зображення різко змінює значення, але потім повертається до початкового значення на деякій невеликій відстані. Однак в реальних зображеннях такі розриви рідко можна зустріти у чистому вигляді. Через згладжування, що впроваджуються більшістю чутливих пристроїв, у реальних сигналах рідко виникають різкі розриви.

При використанні детектора границь можна отримати набір ліній, які поділяються на:

- правильно виділені краї, тобто ті, що існують на представленому об'єкті;
- помилково виділені краї;
- краї, які представлені на об'єкті, але не були виділені.

Помилкові краї називають помилковими позитивами, а відсутні – називають помилковими негативами.

До найвідоміших методів виділення границь відносяться методи Робертса, Превітта і Собеля, які ґрунтуються на розривності – базовій властивості сигналу яскравості. Для використання цих операторів потрібно виконати операцію згортки зображення з відповідними ядрами.

Оператор Робертса забезпечує просте наближення до величини градієнта. Якщо використовувати маски згортки (рисунок 2.1), формулу можна записати як:

$$G(x, y) \approx \sqrt{G^2_x + G^2_y}, \quad (2.1)$$

де, $G(x, y)$ – точка зображення.

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Рисунок 2.1 – Маски оператора Робертса

2.1.2 Метод Віюли-Джонса

Один з методів ідентифікації об'єктів є метод Віюли-Джонса, який був розроблений ще в 2001 році. Призначений для ідентифікації обличчя, проте він

цілком здатний розпізнавати різні класи зображень. Метод використовує механізм ковзаючого вікна, після чого застосовується класифікатор до кожного положення. В алгоритмі також використовується механізм інтегрального представлення зображення, завдяки якому можна порахувати суму інтенсивності пікселів області будь-якого розміру.

Основу алгоритму Віоли-Джонса складають примітиви Хаара (рисунок 2.2).

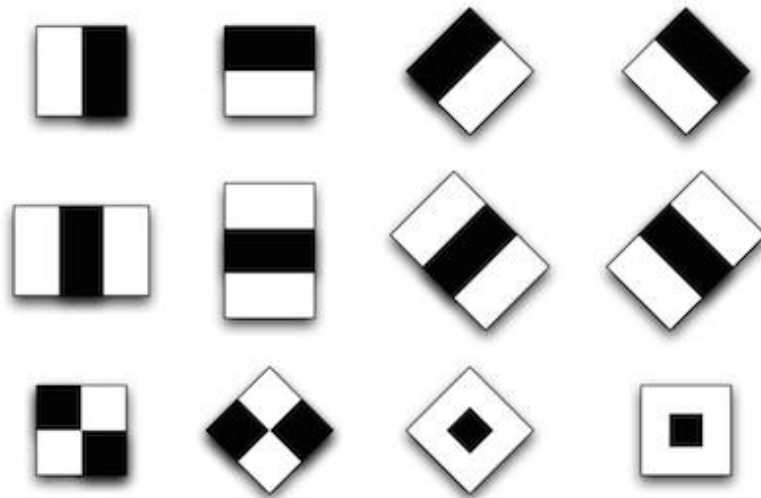


Рисунок 2.2 – Примітиви Хаара

Розрахунок признаку здійснюється за формулою:

$$F \square X \square Y, \quad (2.2)$$

де X – сума значень яскравості точок, які закриваються світлою частиною ознаки;

Y – сума значень яскравості точок, які закриваються темною частиною ознаки.

Метод використовує алгоритм навчання AdaBoost, так як висока швидкість обрахунку ознаки не компенсує значну кількість різних можливих

ознак. Алгоритм працює дуже швидко, але має велику кількість помилкових спрацювань. Кількість хибних спрацювань можна значно зменшити при навчанні системи на великій кількості даних.

2.1.3 Метод гістограм спрямованих градієнтів

Метод побудови ознак на основі гістограм орієнтованих градієнтів був вперше представлений в роботі [10] і є одним з найуспішніших останніх функцій для візуального розпізнавання. HOG заснований на припущенні, що вміст зображення може бути ефективно представлений локальними розподілами напрямків краю або градієнтами інтенсивності, навіть не записуючи їх точних місць [3].

Спочатку даний метод побудови ознак був призначений для завдань виявлення фігур людей на зображеннях (рисунок 2.3), проте подальші дослідження показали ефективність методу для інших завдань класифікації.

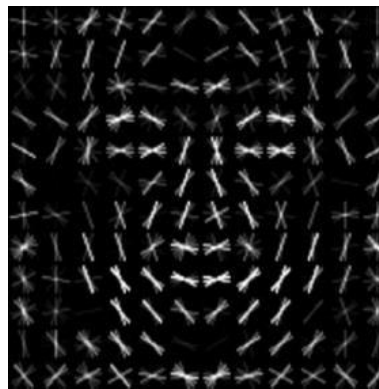


Рисунок 2.3 – HOG-структура обличчя

Основна ідея полягає в тому, що зовнішній вигляд та форму локального об'єкта часто можна досить добре характеризувати розподілом локальних градієнтів інтенсивності чи напрямків краю, навіть без точного знання відповідних позицій градієнта чи краю. Для обробки зображення методом HOG, потрібно для кожного пікселю зображення виявити напрямок, в якому сусідній піксель має більш темний колір.

Для збору різних рівнів інформації можуть використовуватися різні роздільні здатності комірок, наприклад велика комірка з великою роздільною здатністю може бути використана для зйомки загальної форми об'єкта, тоді як менша та більш чітка роздільна здатність фіксують деталі об'єкта.

НОГ виявився особливо ефективним для збору інформації про контур об'єкта, з сильною стійкістю до змін освітленості і деякою стійкістю до невеликих просторових змін.

2.1.4 Локальні бінарні шаблони

Локальні бінарні шаблони (англ. Local Binary Patterns, LBP) – простий оператор, який використовується для класифікації текстур в комп'ютерному зорі. Цей оператор виявився успішними для класифікації текстур та розпізнавання облич через їх стійкість до монотонних змін сірого.

ЛБШ, як і НОГ та більшість сучасних методів, розділяє вікно зображення на сітку комірок. Для обчислення дескрипторів LBP використовує сусідство кожного пікселя. Якщо значення інтенсивності пікселя більше, ніж значення центрального, то цей піксель приймає значення одиниці, якщо менше – нулю (рисунок 2.4).

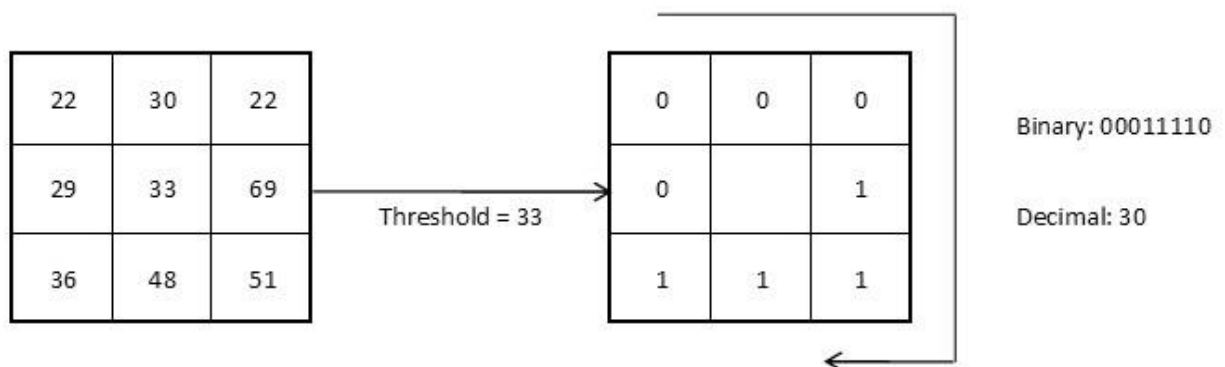


Рисунок 2.4 – LBP

В залежності від отриманого коду можна стверджувати про наявність ліній, кутів, плям на зображенні. Наприклад, отриманий бінарний код у вигляді: 11111001 буде з великою ймовірністю означати кінець лінії, а 00111100 – грань об'єкту [3].

2.2 Класифікація

Найпоширеніша задача пошуку даних – це класифікація. Для класифікації вхідні дані заздалегідь розподілені на класи або категорії. Модель передачі даних вивчає великий набір таких записів. Приклади завдань класифікації можуть бути знайдені майже в усіх сферах діяльності:

- банківська справа: визначення того, чи є транзакція підозрілою або шахрайською;
- закон: розподілення справ в суді, які уже виникали раніше;
- безпека: розпізнавання об'єкту, який повинен або навпаки не повинен знаходитися в певному місці;
- медицина: діагностика наявності певного захворювання;
- та інші.

2.2.1 Метод k-найближчих сусідів

Метод k-найближчих сусідів найчастіше використовується для класифікації, хоча він також може бути також використаний для оцінки та прогнозування. Він заснований на екземплярі, в якому зберігається набір даних про навчання, так що класифікацію нового незнайомого запису можна знайти просто порівнявши його з найбільш подібними записами в навчальному наборі.

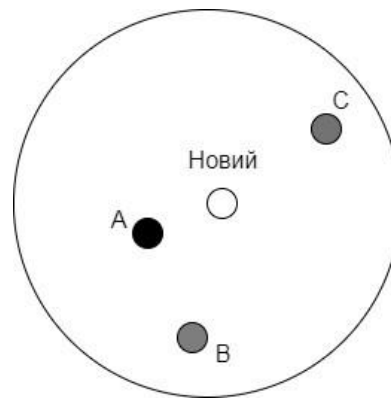


Рисунок 2.5 – Метод k-найближчих сусідів

На рисунку 2.5 можна помітити, що новий приклад найближче розташований до класу “А”. Однак, якщо потрібно класифікувати схожість прикладів по різним параметрам потрібна функція, яка могла б підрахувати їх схожість. Для цього аналітики даних визначають показники відстані для вимірювання подібності, тобто складають метрику. Функція відстані – це реальне значення d , яке відповідає наступним вимогам для будь-яких координат x , y і z :

– відстань d не може бути від’ємною, тобто $d(x, y) \geq 0$. Відстань d

може бути рівна нулю тільки в тому випадку, якщо $x = y$;

– повинен виконуватися закон комутативності, тобто $d(x, y) = d(y, x)$;

– при додаванні третьої точки відстань не може бути меншою від відстані між двома точками, що можна описати наступним виразом:

$$d(x, z) \leq d(x, y) + d(y, z).$$

Найпоширенішою функцією відстані є евклідова відстань, яка виглядає наступним чином:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}. \quad (2.3)$$

Вимірюючи відстань, деякі ознаки, які мають великі значення, можуть перевершити вплив інших атрибутів, які вимірюються меншими масштабами. Щоб цього уникнути, аналітик даних повинен переконатися в нормалізації значень атрибутів. Для нормалізації може бути використана формула:

$$X^* = \frac{(X - \min(X))}{(\max(X) - \min(X))}. \quad (2.4)$$

Евклідова відстань не вказує клас об'єкту, натомість може визначити наскільки один об'єкт відрізняється від іншого [11].

2.2.2 Метод опорних векторів

Моделі SVM спочатку застосовувались для випадків класифікації лінійно розділених класів об'єктів (рисунок 2.6). Для цих двовимірних об'єктів, що належать до класів 1 та 2, легко знайти лінію, яка їх ідеально відокремлює.

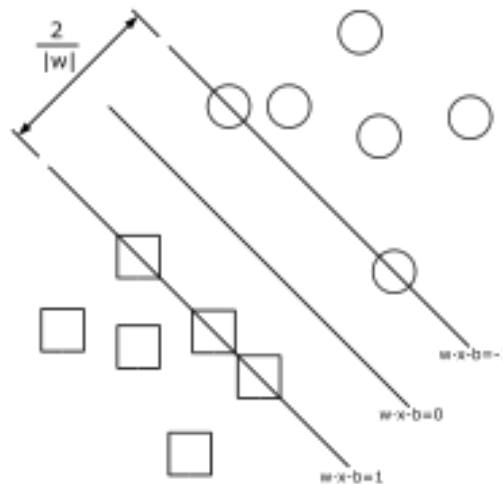


Рисунок 2.6 – Класифікація лінійно розділених класів об'єктів

Для будь-якого конкретного набору об'єктів двох класів, SVM знаходить унікальну площину, що має максимальний запас ($2/|w|$) (рисунок 2.6). Лінія $wx-b=1$ визначає межу з об'єктами класу 1, тоді як лінія $wx-b=-1$ визначає межу з об'єктами класу 2. Три об'єкти з класу 1 проходять через лінію L1, а два об'єкти з класу 2 перетинають лінію L2. Ці об'єкти, через які проходять лінії, називаються опорними векторами. Особливою характеристикою SVM є те, що розв'язання задачі класифікації представлено опорними векторами, які визначають максимально віддалену лінію від другого класу.

SVM також використовується для розділення класів, які не можна розділити лінійним класифікатором (рисунок 2.7). У таких випадках координати об'єктів відображаються у просторі за допомогою нелінійних функцій. Простір функцій – це простір, в якому два класи можна розділити лінійним класифікатором.

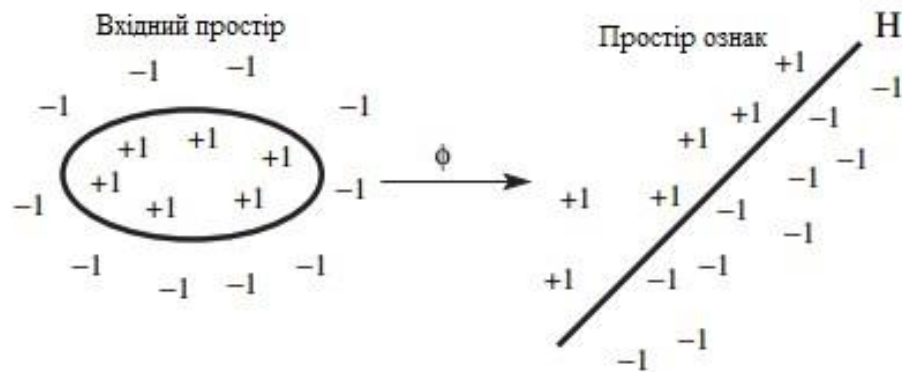


Рисунок 2.7 – Приклад класів, які не можна розділити лінійним класифікатором

2.2.3 Дерево ухвалення рішень

Дерево рішень – засіб прийняття рішень, який використовується в машинному навчанні, аналізі даних і статистиці. Дерево рішень можна представити у вигляді бінарної класифікації, а саме $Y = \{0, 1\}$, хоча вони можуть застосовуватися і для інших завдань прогнозування (рисунок 2.8.). Для кожного вузла на шляху від кореня до листа вибирається відповідь, яка відповідає опису вхідного об'єкта. Якщо дерево рішень використовується для завдань класифікації, воно називається деревом класифікації, якщо для регресійних завдань – деревом регресії.

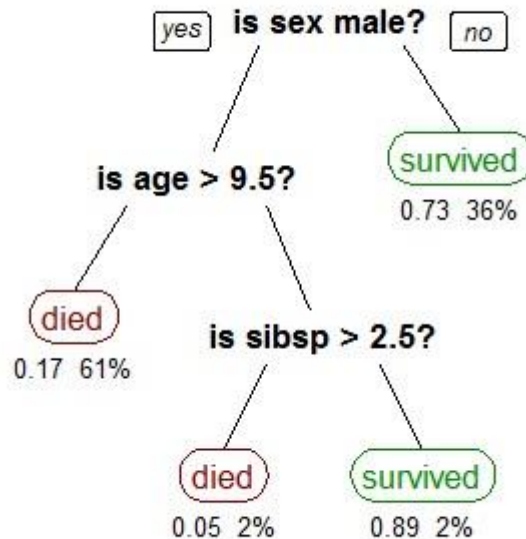


Рисунок 2.8 – Дерево рішень

2.2.4 Методи штучного інтелекту

Для завдань розпізнавання об'єктів можуть бути запропоновані нейронні мережі. Нейронні мережі реалізують класифікаційний підхід. Їх привабливість полягає в здатності розділити простір зображень, використовуючи нелінійні межі для класів. Ці межі отримують за допомогою тренування мережі. Якщо навчальний набір ретельно відібраний для

представлення всіх об'єктів, які зустрічаються пізніше на етапі розпізнавання, тоді мережа може вивчити межі класифікації у своєму просторі функцій. Під час фази розпізнавання мережа працює як і будь-який інший класифікатор. Найпривабливішою особливістю нейронних мереж є їх здатність використовувати нелінійні межі класифікації та здібності до навчання.

3 НЕЙРОННІ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТА ПО КОНТУРУ

3.1 Нейронні мережі як метод штучного інтелекту

Робота над штучними нейронними мережами була розпочата з визнанням того, що людський мозок обчислюється абсолютно іншими способами, ніж звичайний цифровий комп'ютер. Мозок – це дуже складний, нелінійний та паралельний комп'ютер. Він має можливість організовувати свої структурні складові, відомі нам як нейрони, для того, щоб виконувати певні обчислення, наприклад, розпізнавати візерунки, сприймати та керувати рухами, набагато швидше, ніж це може зробити цифровий комп'ютер. Людський зір – це функція візуальної системи забезпечувати уявлення про навколишнє середовище і, що важливіше, надавати інформацію, необхідну для взаємодії з навколишнім середовищем. Мозок, зазвичай, виконує завдання розпізнавання знайомого обличчя приблизно за 100–200 мс, тоді як завдання значно меншої складності займають значно більше часу на потужному комп'ютері [12].

Від народження мозок вже має значну структуру і здатність будувати свої власні правила поведінки через “досвід”. Досвід накопичується з плином часу. Розвиток здатності мозку людини розпізнавати об'єкти відбувається протягом перших двох років від народження. Проте, не можна стверджувати, що цей процес зупиняється після двох років.

У своїй найбільш загальній формі нейромережа – це машина, яка створена для моделювання способу, яким мозок виконує певну задачу чи функцію. Зазвичай реалізується за допомогою електронних компонентів або моделюється в програмному забезпеченні на цифровому комп'ютері. Для досягнення ефективної роботи нейронні мережі використовують масивний взаємозв'язок простих обчислювальних одиниць, які називаються "нейрони".

Процедура, що використовується для виконання навчального процесу, називається алгоритмом навчання, основною функцією якого є зміна синаптичних ваг мережі для досягнення бажаної мети проектування.

До переваг нейронної мережі можна віднести, по-перше, паралельно розподілену структуру і, по-друге, її здатності вчитися а, отже, узагальнювати. Узагальнення стосується вироблення правильних результатів нейронної мережі для входів, які не зустрічались під час навчання. Ці дві можливості обробки інформації дозволяють нейронним мережам знайти приблизні рішення складних проблем.

На практиці, однак, нейронні мережі не можуть забезпечити гарантовано правильне рішення у складній системі. Зокрема, проблема розкладається на низку відносно простих завдань, а нейронним мережам присвоюється підмножина завдань, які відповідають її можливостям.

Треба визнати, що нейромережа не є комп'ютерною архітектурою людського мозку, однак має наступні корисні властивості.

1. Нелінійність. Штучний нейрон може бути лінійним або нелінійним. Нейронна мережа, що складається з взаємозв'язку нелінійних нейронів, сама по собі є нелінійною. Ця властивість є надзвичайно важлива, особливо якщо основний фізичний механізм, що відповідає за генерування вхідного сигналу (наприклад, мовного сигналу), за своєю суттю нелінійний.

2. Навчанням з викладачем. Навчанням з викладачем – популярна парадигма, яка передбачає зміну синаптичних ваг мережі, застосовуючи набір

розмічених навчальних прикладів. Кожен приклад складається з унікального вхідного сигналу та бажаної відповіді. Алгоритм вибирає приклад випадковим чином із набору даних, а синаптичні ваги мережі модифікуються таким чином, щоб мінімізувати різницю між бажаною відповіддю і фактичною реакцією мережі. Навчання повторюється на багатьох прикладах у наборі, до тих пір, доки мережа не досягне стійкого стану, тобто коли не буде подальших значних змін коефіцієнтів. Представлені раніше приклади можна повторно застосовувати під час тренінгу, але в іншому порядку. Таким чином, мережа вчиться на прикладах, будуючи їх узагальнення. Типовим завданням є задача класифікації шаблонів, де метою є призначення вхідному сигналу, що представляє фізичний об'єкт чи подію, одну з декількох заздалегідь визначених категорій.

3. Адаптивність. Нейронні мережі мають вбудовану можливість адаптувати свої синаптичні ваги до змін навколишнього середовища. Нейронну мережу, навчену працювати в конкретних умовах, можна легко перекваліфікувати для вирішення іншої задачі, яка представлена з незначними змінами. Більше того, коли система працює в нестационарному середовищі, тобто, якщо статистика змінюється з часом, нейронна мережа може бути спроектована для зміни своїх коефіцієнтів у режимі реального часу. Архітектура нейронної мережі для класифікації шаблонів у поєднанні з адаптаційними можливостями мережі робить її корисним інструментом у адаптивній класифікації шаблонів. Слід зазначити, що адаптивність не завжди призводить до стійкості. Наприклад, адаптивна система з короткочасними константами може швидко змінюватися і схильна реагувати на помилкові порушення, викликаючи різке погіршення продуктивності системи.

4. Впевненість у відповіді. У контексті класифікації модель нейронної мережі може бути розроблена для надання інформації не тільки про те, який конкретний клас вибрати, а й про впевненість у прийнятому рішенні. Ця інформація може бути використана для відхилення неоднозначних

відповідей, якщо вони виникають, і тим самим поліпшити класифікаційну ефективність мережі.

5. Контекстна інформація. Кожен нейрон в мережі потенційно впливає на глобальну активність усіх інших нейронів у мережі.

6. Відмовостійкість. Нейронна мережа, реалізована в апаратному вигляді, має потенційну властивість бути стійкою до відмов. Наприклад, якщо нейрон пошкоджений, можна очікувати незначне погіршення результату, а не катастрофічний збій.

7. Паралельність. Паралельний характер нейронної мережі робить її потенційно швидкою для обчислення певних завдань. Ця ж функція робить нейронну мережу добре придатною для впровадження в складну систему.

8. Нейробіологічна аналогія. Конструкція нейронної мережі була розроблена за аналогією з мозком. Нейробіологи розглядають штучні нейронні мережі як інструмент дослідження для інтерпретації нейробіологічних явищ. З іншого боку, інженери шукають у нейробіології нових ідей, щоб використати їх у своїх цілях.

3.2 Модель нейрона

Нейрон – це набір операцій над даними, які в сукупності є основоположними для роботи нейронної мережі. Структурна схема моделі нейрона зображена на рисунку 3.1. Вона є основою для проектування нейронних мереж.

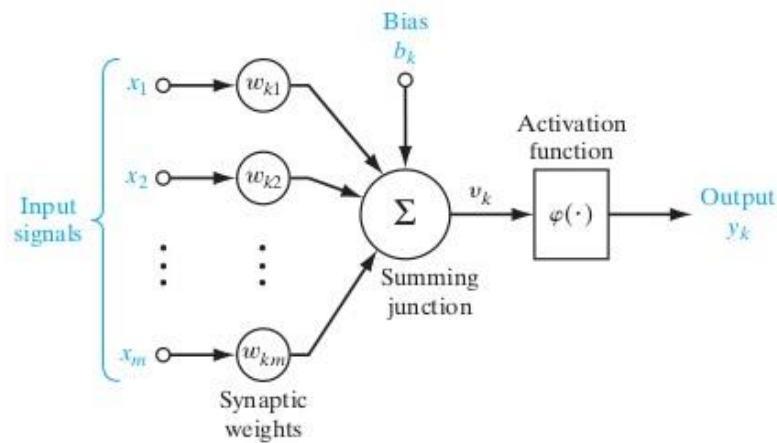


Рисунок 3.1 – Модель нейрона

На зображенні можна виділити такі основні елементи нейронної моделі:

- сукупність вхідних сигналів, кожен з яких характеризується власною вагою. На відміну від ваги синапсу в мозку, синаптична вага штучного нейрона може лежати в діапазоні, що включає як негативні, так і позитивні значення;
- суматор для підсумовування вхідних сигналів;
- функція активації, задача якої обмежити допустимий діапазон амплітуди вихідного сигналу до деякого кінцевого значення. В якості функції активації зазвичай виступають порогова функція або сигмоїдна функція;
- коефіцієнт зміщення bias.

3.3 Архітектура нейронних мереж

Існує багато видів нейромереж. Серед них можна виділити три базові:

- одношарові мережі;
- багатшарові мережі; –
- рекурентні мережі.

Одношарові нейронні мережі складаються із вхідного шару та вихідного (рисунок 3.2). В назві не рахується вхідний шар, оскільки там не проводиться обчислень.

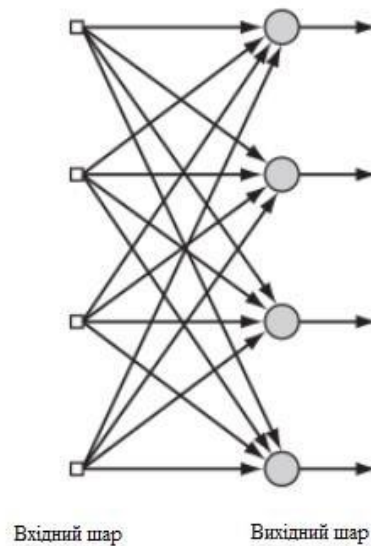


Рисунок 3.2 – Одношарова нейронна мережа

Другий клас нейронної мережі – багатошарові мережі, які відрізняються наявністю одного або декількох прихованих шарів, обчислювальні вузли яких називаються прихованими нейронами (рисунок 3.3). Термін "прихований" означає те, що ця частина нейронної мережі недоступна ні з входу, ні з виходу системи. Функція прихованих нейронів полягає в тому, щоб втручатися в роботу вхідних нейронів для покращення результатів виходу. Додавши один або кілька прихованих шарів, мережа може отримати статистику вищого порядку. Вихідні сигнали другого шару використовуються як входи до третього шару і так далі для решти мережі. Вихідні сигнали нейронів у кінцевому шарі мають описувати загальну реакцію мережі на схему активації, тобто на нейрони першого шару.

Найчастіше можна зустріти модель, при якій кожен нейрон зв'язаний з кожним нейроном наступного шару. Таку нейромережу називають "fully

connected”, тобто повноз'єднана. Якщо деякі синаптичні зв'язки відсутні, то мережу називають частково з'єднаною (partially connected).

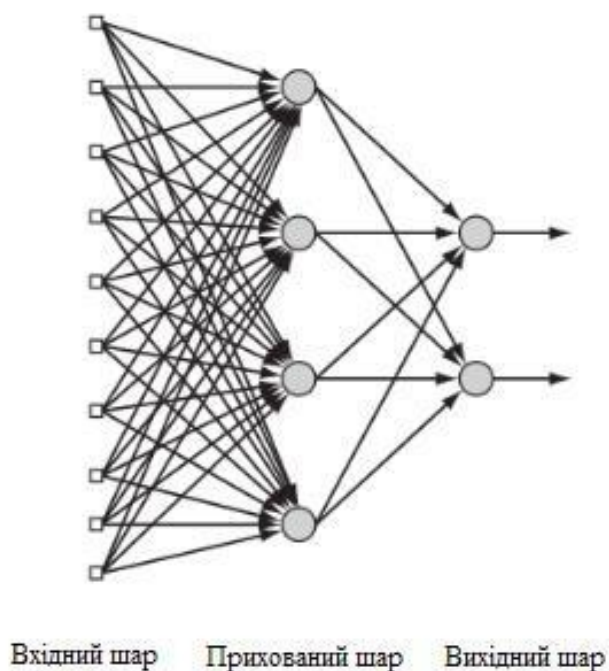


Рисунок 3.3 – Багатошарова нейронна мережа

Рекурентна нейронна мережа відрізняється тим, що вона має щонайменше один цикл зворотного зв'язку. Наприклад, мережа може складатися з одного шару нейронів, при цьому кожен нейрон подає свій вихідний сигнал назад на входи всіх інших нейронів, як показано на рисунку 3.4.

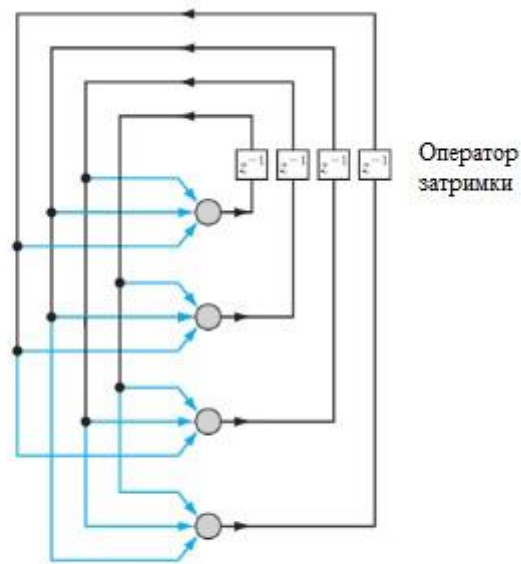


Рисунок 3.4 – Рекурентна нейронна мережа

3.4 Обробка вхідних та вихідних параметрів

Основним завданням нейронної мережі є вивчення моделі середовища, в яку вона вбудована, тобто тих даних, які їй були представлені.

Дані можуть бути як розмічені, так і не розмічені. У розмічених прикладах кожен вхідний сигнал поєднується з бажаною відповіддю. Набір таких пар називається набором даних тренувань, або просто зразком.

В залежності від завдання для нейронної мережі підбирається відповідна архітектура. Наприклад, якщо це задача розпізнавання цифр на основі пікселів зображення, то вхідним шаром може бути кількість нейронів, які відповідають кількості пікселів вхідного зображення, а в якості вихідного шару може бути представлено 10 нейронів, тобто по одному на кожну цифру.

Якість розпізнавання навченої мережі тестується з раніше небаченими даними. Зокрема, вхідне зображення подається в мережу, але цього разу не повідомляється ідентичність цифри, яку представляє саме це зображення.

Потім якість мережі оцінюється шляхом порівнянням відповіді нейромережі з фактичною відповіддю [13] .

3.5 Багатошаровий перцептрон

Багатошаровий перцептрон можна виділити наступними особливостями:

- модель кожного нейрона в мережі включає нелінійну функцію активації, яка є диференційованою;
- мережа містить один або більше прихованих шарів;
- мережа має високий ступінь зв'язності, що визначається синаптичними вагами мережі.

Слід зазначити, що всі ці характеристики значно ускладнюють процес теоретичного аналізу, уявлення про процес навчання та кількість обчислень для навчання мережі. Якщо розбити весь алгоритм на фази, то можна виділити дві фази.

На першій фазі синаптичні ваги мережі фіксуються і вхідний сигнал поширюється по мережі пошарово, поки не досягне результату. Таким чином, на цій фазі зміни обмежуються потенціалами активації та виходами нейронів у мережі.

На другій фазі подається сигнал про помилку шляхом порівняння виходу мережі з потрібною відповіддю. Отриманий сигнал про помилку поширюється по мережі пошарово, але цього разу розповсюдження виконується у зворотному напрямку. На цій фазі проводяться послідовні корективи синаптичних вагів мережі. Розрахунок коригувань для вихідного шару є простим, але для прихованих шарів це потребує деяких обчислень.

Кожен прихований або вихідний нейрон багатошарового перцептрона призначений для виконання обчислення сигналу функції, що з'являється на

виході кожного нейрона та обчислення оцінки градієнтного вектора, необхідних для проходження назад через мережу.

Приховані нейрони виконують роль детекторів. Вони відіграють вирішальну роль у роботі багатошарового перцептрона. По мірі того, як процес навчання проходить через багатошаровий перцептрон, приховані нейрони починають поступово «вивчати» певні закономірності, що характеризують тренувальні дані. Це стає можливим завдяки нелінійному перетворенню вхідних даних у новий простір, який називається простором функцій. У цьому новому просторі класи, які потрібно класифікувати, можуть бути легше відокремлені один від одного. Саме формування цього простору за допомогою контрольованого навчання відрізняє багатошаровий перцептрон від перцептрона Розенблатта.

3.6 Математичний опис нейронної мережі

Нехай вхідне значення нейрона буде позначено як x , вихід – y , синаптичні ваги як w , а b – bias, нейрони для коректування зміщення.

Вихідне значення нейрону, перед проходженням функції активації можна описати наступною формулою:

$$u_i = \sum_{j=1}^n w_{ji} x_j + b_i, \quad (3.1) \text{ до}$$

де x_i – вхідне значення, а w_{ji} – ваги нейрону.

В якості функції активації було обрано сигмоїду:

$$S(x) = 1/(1 + e^{-x}). \quad (3.2)$$

Сигмоїда – неперервно диференційована монотонна нелінійна Сподібна функція (рисунок 3.5), яка часто застосовується для “згладжування” значень деякої величин. Сигмоїда застосовується в нейронних мережах для того, щоб ввести деяку нелінійність в роботу мережі, але при цьому не дуже сильно змінити результат її роботи.

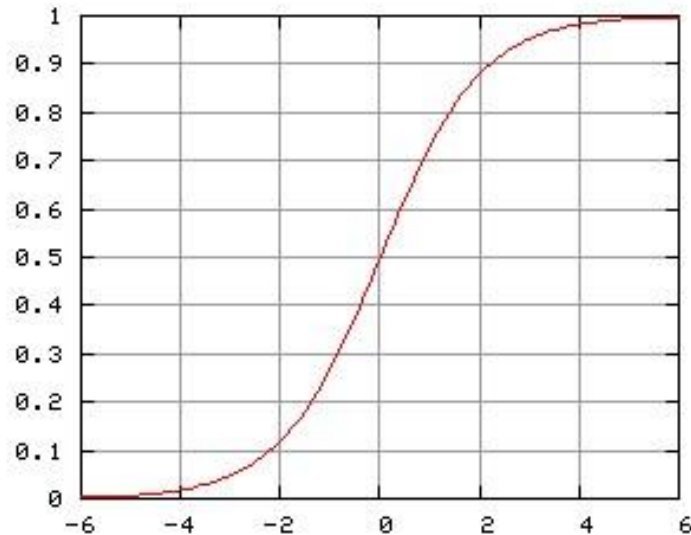


Рисунок 3.5 – Сигмоїда

Після суми всіх вхідних коефіцієнтів значення нейрону поступає на вхід функції активації, що відповідає наступній формулі:

$$y_i = f(u_i), \quad (3.3)$$

де u_i – вихідне значення i -го нейрону, перед проходженням функції активації.

Функція втрат – функція, яка оцінює якість роботи всієї моделі. Для прикладу, в якості такої функції може бути:

Тепер ми можемо позначити $\frac{\partial E}{\partial x_i}$ як δ_i , де $\frac{\partial E}{\partial y_i}$ та вивести

формулу для зміни вагів матриці:

$$\frac{\partial E}{\partial w_{ji}} = \delta_j \cdot y_i - \delta_i \cdot w_{ji} \quad (3.7)$$

3.7 Розпізнавання об'єкту за допомогою нейронної мережі та границь

Перед використанням нейронної мережі зображення потрібно підготувати. Попередня обробка зображення для поставленого завдання полягає в тому, що виділяються лише границі досліджуваного об'єкта. Використання границь об'єкта на зображенні дає можливість перейти в більш компактний дискретний простір змінних. При відсутності кольору і шуму на зображенні, контур залишається єдиним джерелом для подальшої обробки даних: зберігання; навчання; розпізнавання.

Після виділення контуру, об'єкт можна представити у вигляді набору пікселів з числовим значенням від 0 до 1 для кожного пікселя в залежності від його інтенсивності. В якості методу для класифікації отриманих даних застосовується багатошарова нейронна мережа, на вхід якої послідовно подаються значення пікселів.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ

4.1 Мова програмування

Для реалізації алгоритму розпізнавання об'єктів потрібно обрати мову програмування. Основною вимогою є високий рівень абстракції, що дозволить зосередити увагу саме на перевірці працездатності алгоритму, добре реалізовані бібліотеки для роботи з масивами та зображеннями, якісні методи для візуалізації даних та роботи з машинним навчанням.

Враховуючи вищезазначені вимоги було обрано мову програмування Python. Python – широко використовувана мова програмування більш високого рівня ніж Pascal, C++ чи C, що досягається, в основному, за рахунок вбудованих високорівневих структур даних (списки, словники). Python дозволяє швидко працювати та ефективніше інтегрувати системи. Досить простий у використанні, має легкий та елегантний синтаксис. Спочатку він був розроблений Гвідо ван Россумом у 1991 році для акцентування на читабельності коду, що дозволяє програмістам виражати поняття в меншій кількості рядків коду.

Python має бібліотеки для завантаження даних, візуалізації, статистики, обробки природних мов, обробки зображень тощо. Цей величезний набір інструментів забезпечує великий масив функцій загального та спеціального призначення. Однією з головних переваг використання Python є можливість безпосередньо взаємодіяти з кодом, використовуючи термінал або інші інструменти, такі як Jupyter Notebook. Машинне навчання та аналіз даних є фундаментально ітераційними процесами, в яких дані керують аналізом. Для цих процесів важливо мати інструменти, які дозволяють швидку ітерацію та просту взаємодію. Як мова програмування загального призначення, Python також дозволяє створювати складні графічні інтерфейси користувачів (GUI) та веб-сервіси, а також інтегруватись у існуючі системи.

Єдиним недоліком є порівняно невисока швидкість виконання Python програм. Однак це окупається перевагами мови при написанні програм не дуже критичних до швидкості виконання.

Для роботи були використані такі модулі як NumPy, Pillow, OS, Matplotlib.

4.2 Модуль NumPy

NumPy є одним із основних пакетів наукових обчислень в Python. Він містить функціонал для багатовимірних масивів, математичних функцій високого рівня, таких як операції лінійної алгебри та перетворення Фур'є, а також генератори псевдовипадкових чисел. У більшості реалізованих методів масив NumPy – це основна структура даних. Основна функціональність NumPy – клас ndarray, багатовимірний (n-мірний) масив. Усі елементи масиву повинні бути одного типу.

Лістинг 4.1 – Приклад застосування масиву NumPy

```
>>> import numpy as np
>>> a = np.arange(15).reshape(3, 5)
>>> a
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])
>>> a.shape
(3, 5)
>>> a.ndim
2
>>> a.dtype.name
'int64'
>>> a.itemsize
```

```
8
>>> a.size
15
>>> type(a)
<type 'numpy.ndarray'>
>>> b = np.array([6, 7, 8])
>>> b
array([6,
       7, 8])
>>> type(b)
<type 'numpy.ndarray'>
```

4.3 Модуль OS

Модуль OS у python пропонує функції взаємодії з операційною системою. OS входить до стандартних утилітних модулів Python. Цей модуль пропонує портативний спосіб використання функціональних можливостей, залежних від операційної системи. Модулі “os” і “os.path” містять багато функцій взаємодії з файловою системою.

Лістинг 4.2 – Приклад застосування модулю OS

```
>>> import os
>>> path = './test.txt'
>>> fd = os.open(path, os.O_RDWR | os.O_CREAT)
>>> str = "Test"
>>> os.write(fd, str.encode())
>>> os.lseek(fd, 0, 0)
>>> str = os.read(fd, os.path.getsize(fd))
>>> print(str.decode())
>>> os.close(fd)
```

4.4 Модуль Pillow

Pillow це форк PIL (Python Image Library), яка з'явилася завдяки підтримці Алекса Кларка та інших учасників. Заснована на коді PIL, а потім перетворена в поліпшену, сучасну версію. Надає підтримку при відкритті, управлінні і збереженні багатьох форматів зображення.

Лістинг 4.3 – Приклад застосування модулю PIL, який створює ескізи всіх зображень JPEG у поточному каталозі

```
>>> from PIL import Image
>>> import glob, os
>>> size = 128, 128

>>> for infile in glob.glob("*.jpg"):
...     file, ext = os.path.splitext(infile)
...     im = Image.open(infile)
...     im.thumbnail(size)
...     im.save(file + ".thumbnail", "JPEG")
```

4.5 Модуль Matplotlib

Matplotlib є первинною науковою бібліотекою побудови графіків в Python, яка надає функції для створення візуалізації якості публікацій, таких як лінійні діаграми, гістограми, розсіювання графіків тощо (рисунок 4.1).

Лістинг 4.4 – Приклад застосування модулю Matplotlib

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x = np.linspace(0, 2, 100)
```

```

>>> plt.plot(x, x, label='linear')
>>> plt.plot(x, x**2, label='quadratic')
>>> plt.plot(x, x**3, label='cubic')
>>> plt.xlabel('x label')
>>> plt.ylabel('y label')
>>> plt.title("Simple Plot")
>>> plt.legend()
>>> plt.show()

```

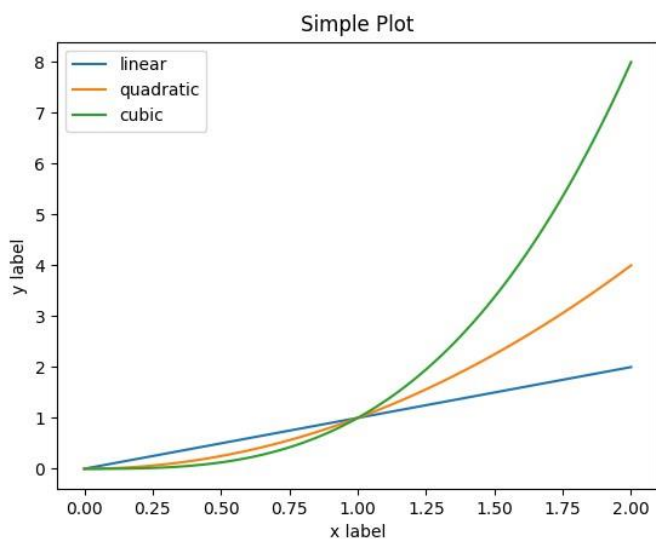


Рисунок 4.1 – Побудова графіків за допомогою matplotlib

4.6 Схема алгоритму розпізнавання

Схема алгоритму розпізнавання зображена на рисунку 4.2.

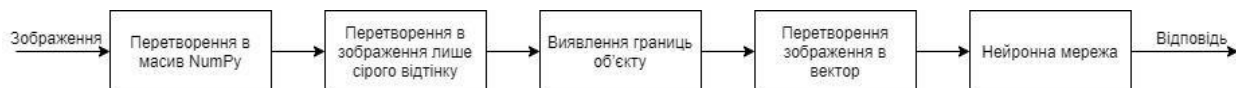


Рисунок 4.2 – Схема алгоритму розпізнавання

4.7 Структура зберігання інформації

Для виконання основної частини алгоритму результат розрахунку роботи нейронної мережі зберігається в словнику Python, який має назву `neural_network_layers`.

Лістинг 4.5 – Структура зберігання інформації

```

if not
os.path.isfile(dump_file):
neural_network_layers = {
    "fully_connected_layer_1": {
        "neurons": np.zeros(4070),
        "weights": (np.random.rand(4070, 30000) - 0.5),
        "bias": (np.random.rand(30000) - 0.5),
    "act_func": "sigmoid"
    },
    "fully_connected_layer_2": {
        "neurons": np.zeros(10),
        "weights": (np.random.rand(30000, number_of_classes) -
0.5),
        "bias": (np.random.rand(number_of_classes) -
0.5),
        "act_func": "softmax"
    },
    "output_layer": {
        "neurons": np.zeros(number_of_classes),
    },
} else:
    neural_network_layers = np.load(dump_file,
allow_pickle=True).item()

```

У першому рядку коду відбувається перевірка на існування файлу з назвою, яка міститься в змінній `dump_file`. Якщо такий файл існує, то за допомогою функції `load` бібліотеки NumPy завантажуються дані про модель нейронної мережі. До структури словника відносяться назви шарів нейронної

мережі, наприклад, "fully_connected_layer_1", які містять відомості про такі складові як:

- neurons;
- weights;
- bias;
- act_func.

Після проходження нейронною мережею одного зображення, за допомогою функції "save" бібліотеки NumPy, відкоректовані ваги зберігаються в файл, що забезпечує продовження навчання після повторного запуску програми.

4.8 Опис функцій для роботи з зображеннями

Зображення поділяються на дві частини: для тренування та для перевірки. За допомогою функції load_images_list, в залежності від режиму програми, до змінної images_list потрапляють зображення з папок "train" або "test".

Лістинг 4.6 – Функція завантаження зображень

```
def load_images_list(dir_name):
    images = []
    for root, dirs, files in
os.walk(dir_name):
        for name in files:
            images.append(os.path.join(root, name))

    return random.sample(images, len(images))
```

В якості параметрів до функції load_images_list потрібно передати назву директорії, де знаходяться зображення. Результатом роботи функції є список шляхів до зображення у випадковому порядку. Ім'я зображення має

починатися з номеру класу, для того, щоб в подальшому його можна було правильно ідентифікувати.

Для перетворення зображення у відтінок сірого кольору використовується метод `convert('L')` з бібліотеки `PIL`.

Для виділення границь використовується функція `detect_edges`, основа якої складається з операції згортки. В якості ядра матриці використовується оператор Робертса. Для виділення лише контурів до функції передається параметр “`threshold`”, який відповідає за порог виділення пікселів.

Лістинг 4.7 – Функція знаходження границь

```
def detect_edges(img, threshold=180):
    vertical_conv = convolution(np.asarray(img),
np.array([[1, 0], [0, -1]]))
    for i in range(len(vertical_conv)):
        for j in range(len(vertical_conv[0])):
            if vertical_conv[i][j] > threshold:
                vertical_conv[i][j] = 0

    horizontal_conv = convolution(np.asarray(img),
np.array([[0, 1], [-1, 0]]))
    for i in range(len(horizontal_conv)):
        for j in range(len(horizontal_conv[0])):
            if horizontal_conv[i][j] > threshold:
                horizontal_conv[i][j] = 0

    y = (vertical_conv+horizontal_conv)
    y = y.astype(np.uint8)
    return
y
```

Функція “Convolution” дозволяє виконати згортку зображення X з матрицею W . Вихідне зображення буде мати висоту та ширину меншу на висоту/ширину матриці $+ 1$, адже згортка виконується з відступами від країв зображення. При проходженні алгоритму до змінної x_piece потрапляє ділянка зображення висота та ширина якої дорівнює матриці W . Ділянка x_piece поелементно перемножується з матрицею W та сумується, а результат записується до вихідного зображення.

Лістинг 4.8 – Функція згортки

```
def convolution(X, W):
    x_height = X.shape[0]
    x_width = X.shape[1]
    y_height = x_height - W.shape[0] + 1
    y_width = x_width - W.shape[1] + 1

    Y = np.zeros((y_height, y_width))
    for h in range(y_height):
    for w in range(y_width):
        x_piece = X[h:h + W.shape[0], w:w + W.shape[1]]
        Y[h, w] = int(np.sum(x_piece * W))

    Y = Y.astype(np.uint8)
    return
Y
```

4.9 Опис функцій для роботи з нейронною мережею

На першому етапі навчання нейронної мережі відбувається пряме проходження зображення через всі нейрони першого шару. Для цього зображення переводиться з двовимірного подання в вектор за допомогою функції “concatenate”, яка є стандартною в бібліотеці NumPy. Далі вектор

зображення перемножується зі значеннями нейронів, та проходить через функцію активації “forward_activation_func”.

Лістинг 4.9 – Функція прямого проходження

```
def forward_fully_connected(y_minus_1, w, b,
act_func):
    x = np.dot(y_minus_1, w) + b    y =
forward_activation_func(x, act_func)
return y
```

Першим параметром функції “forward_activation_func” є вхідний вектор значень нейронів, другим – назва функції активації. В залежності від другого параметру вихідні значення нейронів будуть відрізнятися.

Лістинг 4.10 – Функція активації

```
def forward_activation_func(vector,
func_name):
    output_vector = None
    if func_name ==
'sigmoid':
        output_vector = 1 / (1 + np.exp(-vector))
if func_name == 'softmax':
    output_vector = np.exp(vector) / np.exp(vector).sum()

return output_vector
```

Після проходження всіх слоїв нейронної мережі для підрахунку похибки використовується функція forward_func_error. Результатом функції є оцінка якості роботи всієї моделі.

Лістинг 4.11 – Функція для підрахунку похибки

```
def forward_func_error(y_truth, y_predicted):
    return - y_truth * np.log(y_predicted)
```

Для зміни коефіцієнтів в тренувальному режимі використовується метод зворотного поширення помилки. Для цього спочатку розраховується часткова похідна функції втрат.

Лістинг 4.12 – Функція для знаходження часткової похідної функції втрат

```
def backward_func_error(y_truth, y_predicted):
    return - (y_truth / y_predicted)
```

Після знаходження часткової похідної функції втрат, кожен із слоїв нейронної мережі подається на функцію `backward_fully_connected`. Параметрами функції є вектор похибок, який був отриманий на попередньому кроці, вектор нейронів попереднього шару, вектор нейронів поточного шару, ваги, коефіцієнт зміщення та назва функції активації.

Лістинг 4.13 – Функція для зворотного поширення помилки

```
def backward_fully_connected(error_vector, y, y_minus_1, w,
b, act_func):
    if act_func == 'softmax':
        x = np.dot(error_vector, backward_activation_fn(y,
act_func))
    else:
        x = error_vector * backward_activation_fn(y,
act_func)
    d = np.dot(x, w.T)
    w = w - 0.01 * np.dot(y_minus_1.T, x)
    b = b - 0.01 * x
```

```
return d, w, b
```

Результатом функції `backward_fully_connected` є нове значення похибки, нові коефіцієнти ваги та зміщення. В якості допоміжної функції `backward_fully_connected` використовує `backward_activation_fn`, яка розраховує похідну зазначеної функції.

Лістинг 4.14 – Функція для розрахунку похідної функції активації

```
def backward_activation_fn(vector, func_name):
    output_vector = None    if
func_name == 'sigmoid':
        output_vector = vector * (1 - vector)
elif func_name == 'softmax':
        input_vector = np.copy(vector)
        output_vector = np.zeros((vector.shape[1],
vector.shape[1]))        for i in range(output_vector.shape[1]):
for j in range(output_vector.shape[1]):            if i == j:
                output_vector[i][i] = input_vector[0][i]
* (1 - input_vector[0][i])
        else:
                output_vector[i][j] = -
input_vector[0][i] * input_vector[0][j]
    return output_vector
```

5 ТЕСТУВАННЯ

5.1 Тестування функції розпізнавання границь об'єкта

Для розпізнавання границь об'єкта було використано математичну операцію згортки зображення з матрицею. Для першого тесту було обрано

матрицю, яка використовується в операторі Робертса (рисунок 5.1). Для другого тесту було обрано матрицю, яка використовується в операторі Собеля (рисунок 5.2), а для третього тесту матрицю з оператора Прюїтта (рисунок 5.3).

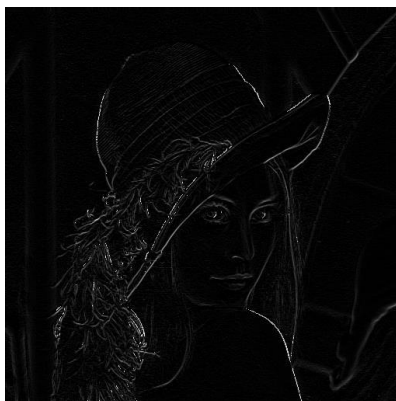


Рисунок 5.1 – Результат виділення границь оператором Робертса

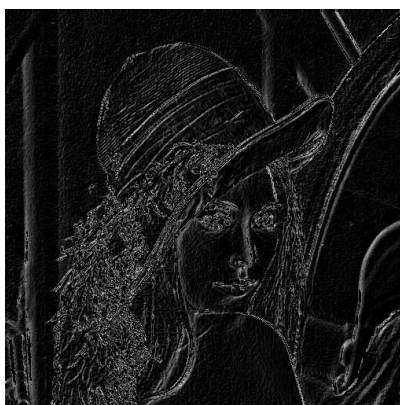


Рисунок 5.2 – Результат виділення границь оператором Собеля

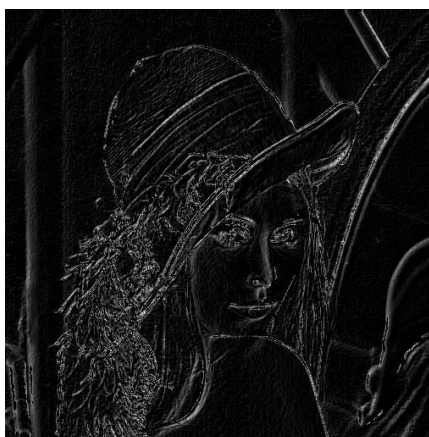


Рисунок 5.3 – Результат виділення границь оператором Прюїтта

5.2 Розпізнавання простого зображення

Для тестування роботи нейронної мережі потрібно перевірити правильність навчання її коефіцієнтів на прикладі простого зображення. В якості зображення буде виступати матриця 2 x 2 (рисунок 5.4).

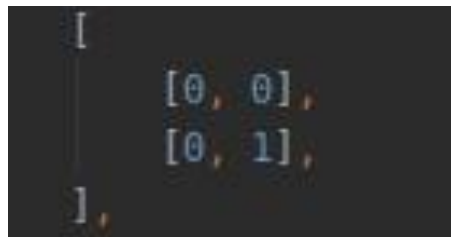


Рисунок 5.4 – Матриця 2 x 2

Згенеровані випадковим чином ваги (weights) першого шару представлені у лістингу 5.1. Перші 7 коефіцієнтів відповідають вагам першого нейрону першого шару.

Лістинг 5.1 – Коефіцієнти першого шару представлені

```

[[-0.20926026 -0.37948144 -0.09157494 -0.44792551 -
0.41699791 0.41018616 0.44778935]
[-0.25719187 -0.470063 -0.29953789 -0.13204956 0.45457206
0.28653696 -0.44787033]
[ 0.41396652 -0.46422919 -0.10103826 -0.2857184 -0.17144357
-0.09553563 -0.37131953]
[-0.10004213 0.28101881 -0.37982412 -0.19711796 0.1960904 -
0.35296267 0.01854169]]

```

Вихідні значення нейронів першого шару та подальші коефіцієнти зображені на рисунку 5.5.

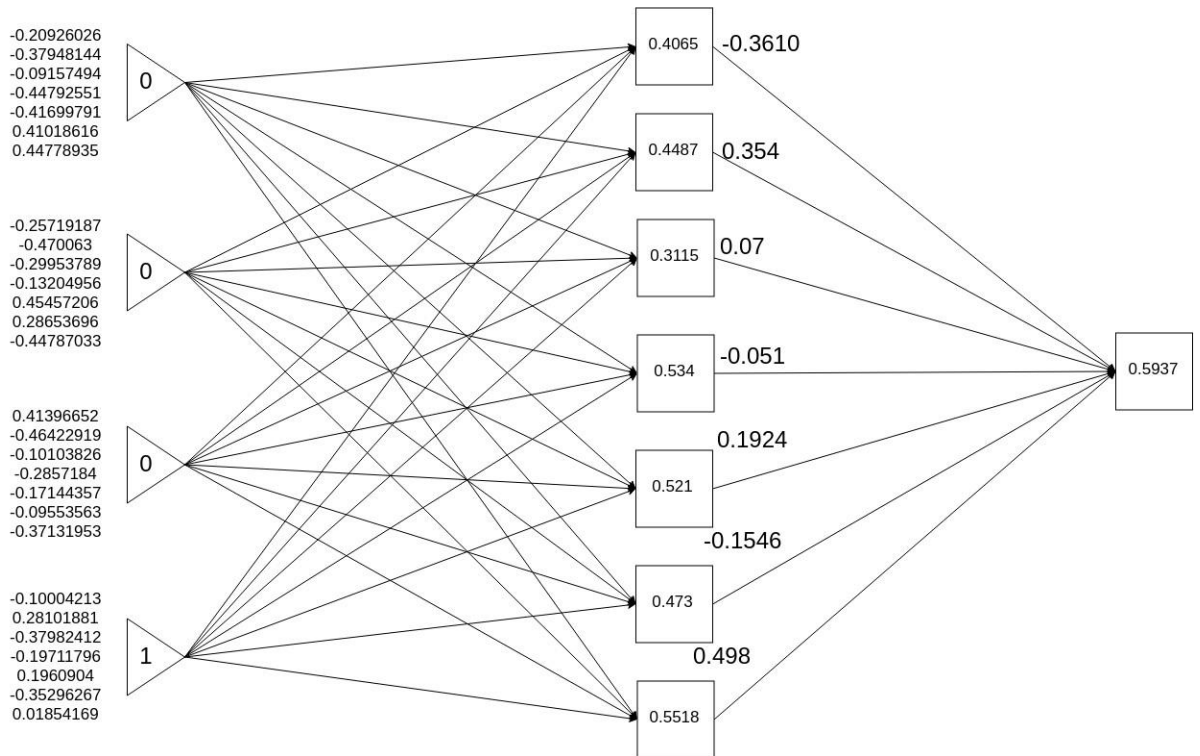


Рисунок 5.5 – Структура нейронної мережі з випадковими коефіцієнтами

Після проходження першого кроку, результат вихідного нейрона дорівнює 0,5937. Так як бажаний результат повинен бути наближений до 1, відбувається коригування вагових коефіцієнтів згідно формул, які описані в розділі 3.

Таблиця 5.1 – Результат проходження 20480 епох нейронною мережею

№ Епохи	Значення вихідного нейрону
10	0.5992
20	0.6052
40	0.6167

80	0.6381
160	0.67491306
320	0.72970401
640	0.79520518
1280	0.85607234
2560	0.90256895
5120	0.93463561
10240	0.95599684
20480	0.97017888

Як показано в таблиці 5.1, після проходження 20480 епох нейронна мережа правильно відредагувала коефіцієнти згідно визначених умов.

5.3 Тестування системи розпізнавання об'єктів

Для тестування було підбрано 212 різних зображень, які відносяться до 10 різних класів. Приклади набору даних зображені на рисунку 5.6.



...



Рисунок 5.6 – Набір зображень для тесту

Після використання методу знаходження границь, можна помітити, як кількість інформації на зображенні зменшилася, але завдяки контуру об'єкти можна добре відрізнити (рисунок 5.7).



Рисунок 5.7 – Виділення границь об'єктів

Після проходження першої епохи кількість правильно ідентифікованих зображень складає 35% для 10 класів. Проходження трьох епох покращує результат на 20%, а навчання нейронної мережі на десяти епохах дає змогу правильно класифікувати 58% зображень. Якщо набір даних класифікувати тільки на живі і неживі об'єкти, то відсоток правильних відповідей складає 90%.

З проходженням кожної епохи кількість правильно класифікованих об'єктів зростає, що можна побачити на рисунку 5.8 та таблиці 5.2.

Слід також зазначити, що при відсутності етапу виділення границь об'єкта, кількість правильних відповідей нейронної мережі значно знижується, що можна побачити порівнявши рисунок 5.8 та рисунок 5.9.

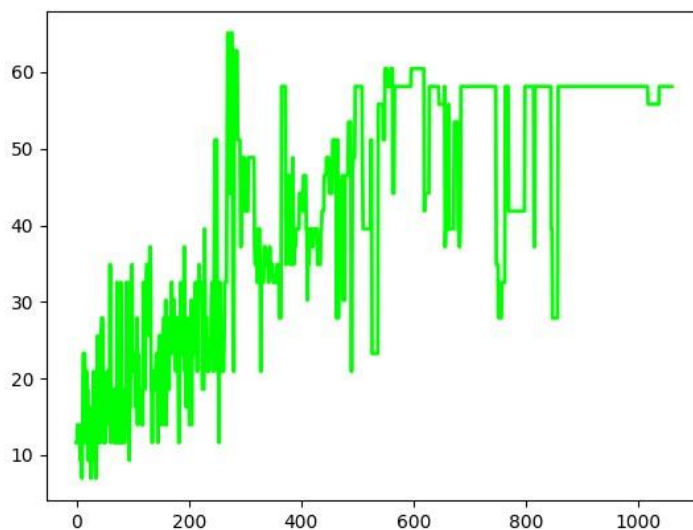


Рисунок 5.8 – Результат роботи алгоритму з виділенням границь об'єктів

Таблиця 5.2 – Результат проходження 1000 зображень нейронною мережею з виділенням границь об'єктів

К-ть пройдених зображень	Відсоток правильних відповідей
100	25
200	30
300	40
400	45
500	47
600	58
700	58
800	58

900	58
1000	58

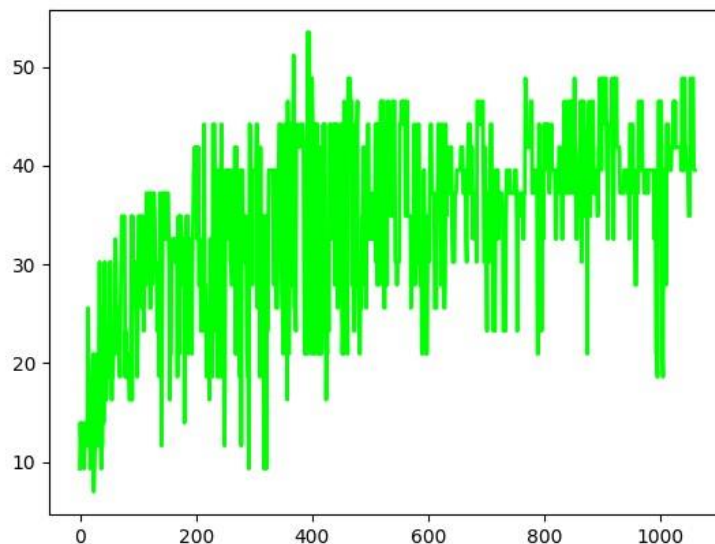


Рисунок 5.9 – Результат роботи алгоритму без виділення границь об'єктів

Таблиця 5.2 – Результат проходження 1000 зображень нейронною мережею без виділення границь об'єктів

К-ть пройдених зображень	Відсоток правильних відповідей
100	19
200	27
300	28
400	32
500	35
600	32

700	31
800	32
900	37
1000	37

ВИСНОВКИ

У даній роботі було розглянуто алгоритми, які можуть бути використані для розпізнавання об'єктів по їх контуру та існуючі методи для роботи з зображеннями.

В ході виконання роботи здійснено аналіз існуючих моделей та алгоритмів розпізнавання об'єктів. Було розглянуто методи знаходження контуру, метод Віюли-Джонса, метод гістограм спрямованих градієнтів, локальні бінарні шаблони. Для вирішення задачі класифікації були розглянуті методи k-найближчих сусідів, метод опорних векторів та дерево ухвалення рішень. Для реалізації нового алгоритму за основу була обрана багатошарова нейронна мережа з функцією активації softmax. Для виділення контуру об'єкта були застосовані операції згортки з матрицями оператора Робертса.

На основі аналізу було визначено, що розгляд об'єкта тільки по його контуру дозволяє покращити результат нейронної мережі та значно прискорити процес навчання.

Результати тестування показали, що цей метод можна використовувати в умовах з обмеженою видимістю для розпізнавання живих об'єктів при охороні державного кордону, будинків та інших важливих об'єктів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bogusław Cyganek, Object detection and recognition in digital images : theory and practice [Текст] / Bogusław Cyganek – New York : John Wiley & Sons, 2013. – 552 с
2. Теория распознавания образов [Электронный ресурс] / wikipedia.org. – Режим доступа : www/ URL: https://ru.wikipedia.org/wiki/Теория_распознавания_образов – 15.05.2018 г. –
Заголовок с экрана.
3. Sabit Ul Hussain, Machine Learning Methods for Visual Object Detection [Текст] / Sabit Ul Hussain – Saint-Martin-d'Hères: Université de Grenoble, 2011. – 127 с
4. Christopher, M. Bishop Pattern Recognition and Machine Learning [Текст] / Singapore: Springer, 2006. – 738 с
5. Biederman, I. Recognition-by-components: A theory of human image understanding. Psychological Review [Текст] / Biederman, I – Washington: American Psychological Association, 1987. – 33 с

6. Ramesh Jain, MACHINE VISION [Текст] / Ramesh Jain, Rangachar Kasturi, Brian G. Schunck – New York: McGraw-Hill, 1995. – 512 с

7. РАСПОЗНАВАНИЕ ОБЪЕКТОВ ПО ИХ КОНТУРУ С

ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА /

Негода Н.С., Кривуля Г.Ф., // 9th International Scientific and Practical Internet Conference – 2019. – №9. 599-602 с

8. Фільтр високих частот [Електронний ресурс] / wikipedia.org.
– Режим доступу : [www/ URL:
https://uk.wikipedia.org/wiki/Фільтр_високих_частот](http://www.wikipedia.org/wiki/Фільтр_високих_частот) – 18.05.2018 г. –
Заголовок з екрану.

9. Фільтр низьких частот [Електронний ресурс] / wikipedia.org.
– Режим доступу : [www/ URL:
https://uk.wikipedia.org/wiki/Фільтр_низьких_частот](http://www.wikipedia.org/wiki/Фільтр_низьких_частот) –
18.05.2018 г. – Заголовок з екрану.

10. Dalal N., Histograms of oriented gradients for human detection [Текст] / Dalal N. – Proceedings of the IEEE conference on computer vision and pattern recognition , 2005. – 8 с

11. Daniel T. Larose, Discovering knowledge in data [Текст] / New York : John Wiley & Sons, 2005. – 217 с

12. Хайкин, С. Нейронные сети полный курс [Текст] / Саймон Хайкин.
– М. : Вильямс, 2006. – 1104с

13. Simmons R., The Roles of Knowledge and Representation in Problem Solving [Текст] / R. Simmons, R. Davis – Berlin : Springer Science, 1993. – 27-43 с

Выводы и перспективы дальнейшего развития. Таким образом, анализ объекта только по его контуру позволяет значительно ускорить процесс обучения нейронной сети, а также использовать этот метод в условиях с ограниченной видимостью. Развитие и использование технологий распознавания контурных изображений требует дальнейших исследований, так как существующие алгоритмы недостаточно эффективны.

Список литературы:

1. Microsoft's Kinect Is Now Guarding the Korean Border [Электронный ресурс] / kotaku.com. Режим доступа: [www/ URL: https://kotaku.com/microsofts-kinect-is-now-guarding-the-korean-border1514792443](http://www.kotaku.com/microsofts-kinect-is-now-guarding-the-korean-border1514792443) – 13.10.2018 г. – Заголовок с экрана.
2. Теория распознавания образов [Электронный ресурс] / wikipedia.org. Режим доступа: [www/ URL: https://ru.wikipedia.org/wiki/Теория_распознавания_образов](https://ru.wikipedia.org/wiki/Теория_распознавания_образов) – 10.09.2019 г. – Заголовок с экрана.
3. Теория распознавания образов [Электронный ресурс] / wikipedia.org. Режим доступа: [www/ URL: https://ru.wikipedia.org/wiki/Теория_распознавания_образов](https://ru.wikipedia.org/wiki/Теория_распознавания_образов) – 10.09.2019 г. – Заголовок с экрана.
4. Выделение границ [Электронный ресурс] / wikipedia.org. – Режим доступа: [www/ URL: https://ru.wikipedia.org/wiki/Выделение_границ](https://ru.wikipedia.org/wiki/Выделение_границ) –

10.09.2019 г. – Заголовок с экрана.

5. Перекрёстный оператор Робертса [Электронный ресурс] / wikipedia.org. – Режим доступа: www/ URL: https://ru.wikipedia.org/wiki/Перекрёстный_оператор_Робертса –
10.09.2019 г. – Заголовок с экрана.
6. Хайкин, С. Нейронные сети полный курс [Текст] / Саймон Хайкин. – М. : Вильямс, 2006. – 1104с