

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

ВИПУСКНА РОБОТА

на тему:

**«Інтелектуальна інформаційна система
розпізнавання зображень»**

Завідувач

випускаючої кафедри

Довбиш А.С.

Керівник роботи

Берест О.Б.

Студента групи ІН – 61

Шерстюк А.Г.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2020 р.

Завдання
до випускної роботи

Студента четвертого курсу, групи ІН-61 спеціальності “Інформатика”
денної форми навчання Шерстюка Андрія Геннадійовича.

**Тема: “Інтелектуальна інформаційна система розпізнавання
зображень”**

Затверджена наказом по СумДУ

№ _____ від _____ 2020 р.

Зміст пояснювальної записки: 1) аналітичний огляд методів створення інформаційних систем розпізнавання зображення; 2) постановка завдання й формування завдань дослідження; 3) огляд і опис засобів для розробки; 4) розробка інтелектуальної інформаційної системи; 5) тестування створеної системи.

Дата видачі завдання “ _____ ” _____ 2020 г.

Керівник випускної роботи _____ Берест О.Б.

Завдання прийняв до виконання _____ Шерстюк А.Г.

РЕФЕРАТ

Записка: 43 стор., 26 рис., 0 додатків, 18 джерел.

Об'єкт дослідження — Інтелектуальна інформаційна система розпізнавання зображень

Мета роботи — розробити інформаційну систему для розпізнавання певних класів зображень (літаки, автомобілі, птахи, коти, олені, собаки, жаби, коні, кораблі, вантажівки), система повинна наблизитись до максимального значення правильності передбачення відповідного класу зображень, підібрати набір даних для навчання та тестування, систему потрібно протестувати, використовуючи зображення всіх класів.

Результати — створено інформаційну систему для розпізнавання певних класів зображень, досягнуто максимального значення правильності передбачення відповідного класу зображень, ця величина дорівнює ~80%, обрано набір даних для навчання системи та її тестування, система протестована використовуючи зображення всіх обраних класів.

ЗНМ, ПЕРЕНАВЧАННЯ, РЕГУЛЯРИЗАЦІЯ, PADDING, STRIDE,
СИНАПС, ШАР, ЯДРО ЗГОРТКИ

Зміст

Завдання	2
РЕФЕРАТ	3
Зміст.....	4
Вступ	5
Актуальність	6
Огляд основної ідеї нейронних мереж.....	7
1 Огляд існуючих рішень	16
1.1. Персептрон.....	16
1.2. Згорткові нейронні мережі	18
2 Вибір методу рішення	25
2.1. Конфігурація бібліотек потрібних для виконання задачі	25
2.2. Конфігурація каталогу для збереження навченої моделі та зображень.....	26
2.3. Вибір зображень для навчання мережі та для її тестування.....	26
2.4. Програмна реалізація	26
2.5. Перевірка правильності навчання моделі.....	29
Висновок	41
Список використаних джерел.....	42

Вступ

Робота присвячена проблемі розпізнавання об'єктів на зображенні, а також створенню інформаційної системи для розпізнавання одиночних об'єктів на зображенні.

Проблематика розпізнавання об'єктів на зображенні зустрічається в багатьох сферах нашого життя, починаючи з ігрових, де потрібно визначити є на зображенні тварина чи ні, до більш серйозних, де роботу потрібно розпізнати об'єкт для подальшої взаємодії. Зараз основну практичну реалізацію складає розробка безпілотних автомобілів, адже щоб оцінити ситуацію на дорозі машині потрібно дуже швидко розпізнавати оточення. Також технологія розпізнавання об'єктів може бути використана спецслужбами для пошуку потрібної людини в соціальних мережах.

Актуальність

В наші часи існує безліч алгоритмів обробки зображень, кожен з них чимось відрізняється від інших, наприклад, швидкість виконання, меншою затратою ресурсів комп'ютера чи більшою ефективністю. Всі ці компоненти показують якість створеної нейронної мережі, але на який показник треба звернути найбільше уваги. Я вважаю, що найкращий показник успішності нейронної мережі – максимальна ефективність алгоритму, тому основною ціллю моєї системи є пошук рішення для збільшення цього показника.

Огляд основної ідеї нейронних мереж

Нейронна мережа - це послідовність нейронів, з'єднаних між собою синапсами. Завдяки цій структурі машина отримує здатність аналізувати та навіть запам'ятовувати різну інформацію. Нейронні мережі також здатні не тільки аналізувати вхідну інформацію, але і відтворювати її зі своєї пам'яті. Іншими словами, нейронна мережа - це машинна інтерпретація людського мозку, в якій є мільйони нейронів, що передають інформацію у вигляді електричних імпульсів.

Нейронні мережі використовуються для рішення складних проблем, які потребують аналітичних розрахунків, аналогічних тому, що робить людський мозок. Найпоширеніші приклади використання нейронних мереж:

- Класифікація – визначення приналежності вхідного об'єкту, представленого функціональним вектором, до одного із заздалегідь визначених класів. Наприклад, визначення хвороби за певними критеріями.
- Прогнозування – це здатність передбачати наступний крок або події. Наприклад, зростання або падіння акцій, збільшення цін та інші.
- Розпізнавання – є найширшим застосуванням нейронних мереж. Використовується при розпізнаванні фотографії або в камерах телефонів, коли вона визначає контури вашого обличчя та виділяє його, а також в багатьох інших напрямках.

Виділяють два основних класи нейронних мереж:

- Одношарова нейронна мережа (див. рис. 1) – мережа, в котрій сигнал передається від вхідного шару до вихідного шару без особливих перетворень, вихідний шар зробить процеси перетворення та виводу результату.

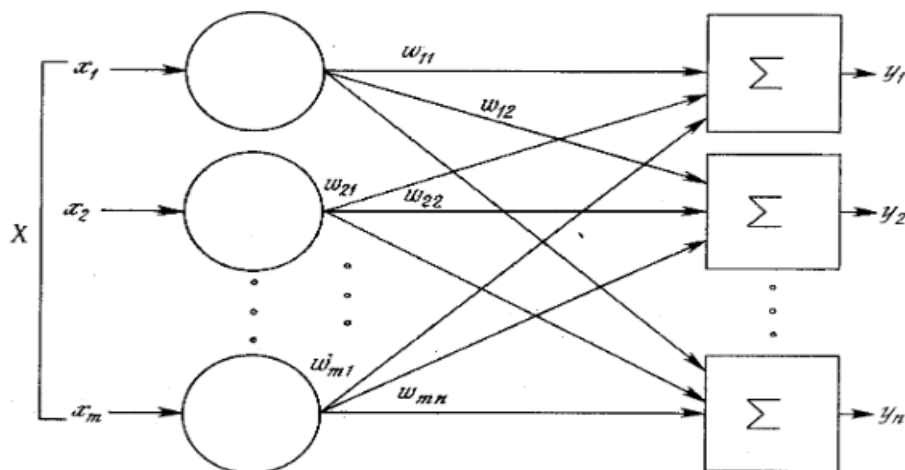


Рисунок 1 – Вигляд одношарової нейронної мережі

- Багатошарові нейронні мережі (див. рис. 2) – нейронні мережі, які складаються з вхідного, вихідного та одного або декількох прихованих шарів розташованих між ними. Багатошарові нейронні мережі мають більші можливості у використанні ніж одношарові, проте багатошарові мережі призведуть до збільшення обчислювальної потужності в порівнянні з одношаровими нейронними мережами лише у випадках, коли функція активації буде не лінійною. Кожний прихований шар виконує процес перетворення вхідних сигналів у деякі проміжні результати.

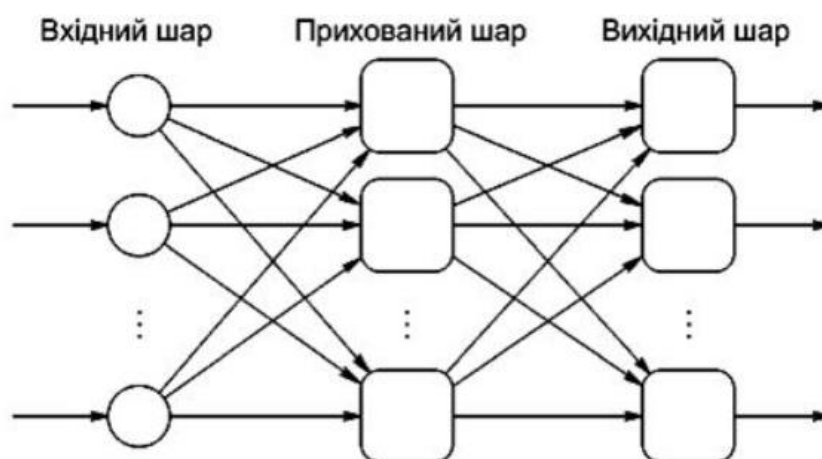


Рисунок 2 – Вигляд багатошарової нейронної мережі

В теорії нейронних мереж важливим фактором є спосіб передачі сигналу. Виділяють два основних класи нейронних мереж за способом розповсюдження сигналу: мережі прямого розповсюдження (див. рис. 2) та мережі зі зворотнім зв'язком (див. рис. 3).

- Мережі прямого розповсюдження – це штучні нейронні мережі, в яких сигнал поширюється від вхідного шару до вихідного без зворотних зв'язків. Мережі, які відносяться до цього класу зазвичай використовуються в задачах прогнозування, кластеризації та розпізнавання.
- Мережі зі зворотнім зв'язком – це штучні нейронні мережі, в яких вихід нейрона може знов використовуватися як вхідний сигнал.

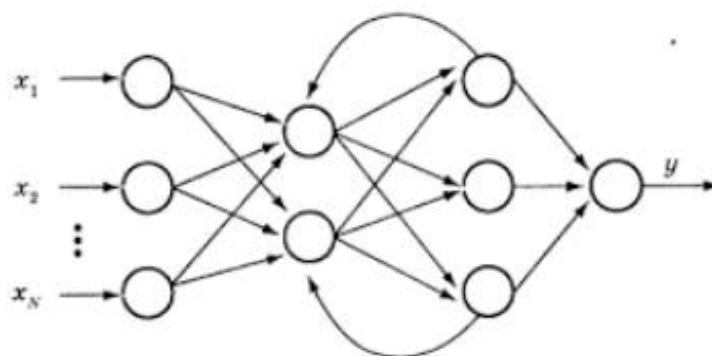


Рисунок 3 – Вигляд нейронної мережі зі зворотнім зв'язком

Основні компоненти нейронних мереж [13]:

1. Нейрон;
2. Синапс;
3. Вага;
4. Шари;
5. Функція активації.

Нейрон [18] - це обчислювальна одиниця, що після отримання інформації, виконує прості обчислювальні дії та передає її у вигляді вихідного сигналу далі. Нейрони поділяються на три основні типи (див. рис. 4): вхідний (синій), прихований (червоний) та вихідний (зелений). У

випадку, коли нейронна мережа складається з великої кількості нейронів, вводить термін шар. Виділяють три основних види шарів: вхідний шар, який приймає інформацію, один або декілька прихованих шарів, які її обробляють, і вихідний шар, який відображає результат. У кожного нейрона є 2 основні параметри: вхідні дані та вихідні дані. У випадку вхідного нейрона: вхідне значення дорівнює вихідному значенню. В іншому загальна інформація про всі нейрони з попереднього шару надходить у вхідне поле, після чого нормалізується за допомогою функції активації і виводиться у вихідне поле.

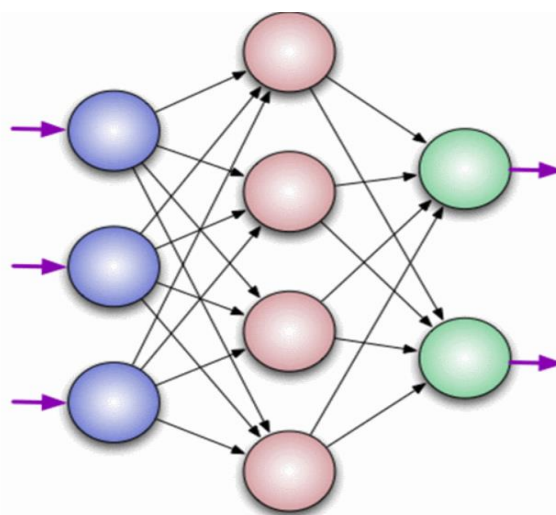


Рисунок 4 – Схематичне зображення нейронів

Синапсом називається зв'язок між двома нейронами. Синапси мають 1 параметр - вага. Завдяки йому вхідна інформація змінюється, при передачі від одного нейрона до наступного. Припустимо, є 3 нейрони, які передають інформацію наступному. Тоді маємо 3 ваги, які відповідають кожному з цих нейронів. Той нейрон, який матиме більше ваги, ця інформація буде домінуючою в наступному нейроні (наприклад, змішування кольорів). Насправді набір ваг нейронної мережі або матриця ваг є своєрідним мозком всієї системи. Завдяки цим масштабам вхідна інформація обробляється і перетворюється на результат.

Вага - це міцність зв'язку між нейронами. Наприклад, якщо вага з'єднання вузлів 1 і 3 більша, ніж між вузлами 2 і 3, це означає, що нейрон 1 має більший вплив на нейрон 3. Нульова вага означає, що зміни введення не вплинуть на вихід. Від'ємна вага вказує на те, що збільшення введення зменшить вихід. Вага визначає вплив введення на вихід.

Шар – це один або декілька нейронів, які використовують один набір вхідних сигналів.

Функція активації - це спосіб нормалізувати вхідні дані. Тобто, функція активації дозволяє перетворити будь-яке вхідне значення у значення в певному діапазоні. Функцій активації дуже багато. Основні функції активації: лінійна (див. рис. 5), сигмоїдна (див. рис. 6) та гіперболічний тангенс (див. рис. 7). Основною відмінністю функцій активації - є діапазон значень.

- Лінійна функція – дана функція майже не використовується, виключення є випадки, коли потрібно виконати тестування нейронної мережі або передати значення без перетворень.

$$f(x) = x$$

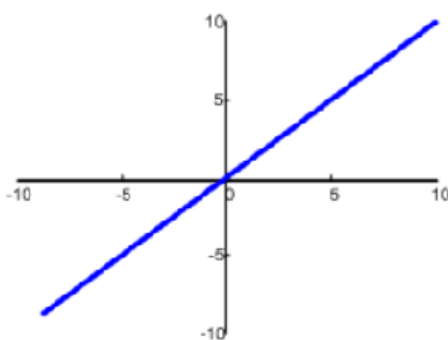


Рисунок 5 – Вигляд лінійної функції

- Сигмоїд – одна з найрозповсюдженіших функцій активації, діапазон значень цієї функції $[0, 1]$. Зазвичай вона використовується для

нейронних мереж де спостерігаються тільки додатні значення. Якщо потрібно використовувати від'ємні значення, то функцією активації потрібно обрати іншу функцію.

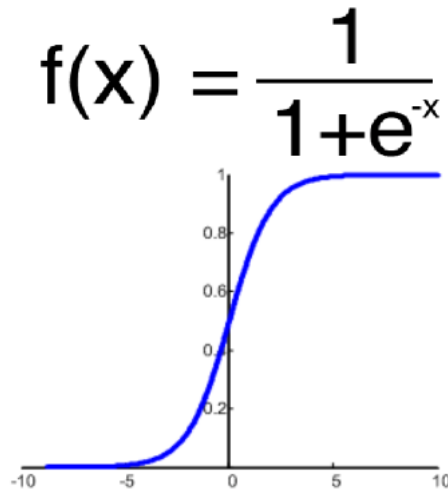


Рисунок 6 – Вигляд сигмоїдної функції активації

- Гіперболічний тангенс – функція активації, що використовується для роботи з додатними та від'ємними значеннями. Діапазон функції – $[-1, 1]$. Використання даної функції для роботи тільки з додатними значеннями не є гарною практикою, адже це може значно зменшити ефективність.

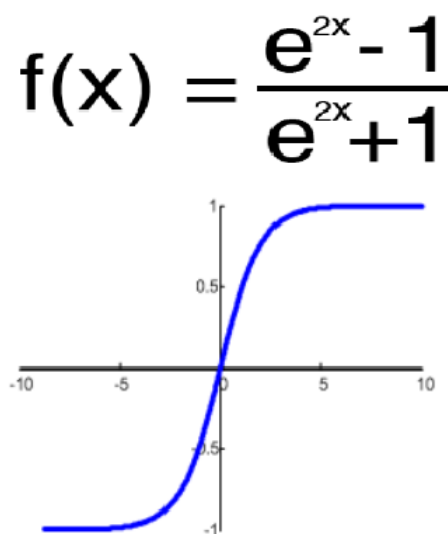


Рисунок 7 – Вигляд функції гіперболічного тангенсу

Похибка – це величина, що вимірюється у відсотках і означає відмінність між очікуваним та отриманим результатами. Похибка знаходиться на кожній епосі, з кожною новою епохою похибка повинна зменшуватися. Отримати числове значення похибки можливо використовуючи відповідні функції, основні способи: MSE (Mean Squared Error) (див. рис. 8), Root MSE (див. рис. 9) та Arctan (див. рис. 10). Кожен метод підраховує похибку по своєму. У Arctan похибка майже завжди буде більшою за інші методи, так як він працює за принципом – чим більша різниця, тим більша похибка. У Root MSE буде найменша похибка, тому найчастіше обирають MSE, адже цей метод зберігає баланс у знаходженні похибки, але найкращим методом для пошуку похибки є той спосіб, що дає найкращі результати.

$$\frac{(i_1 - a_1)^2 + (i_2 - a_2)^2 + \dots + (i_n - a_n)^2}{n}$$

Рисунок 8 – Формула знаходження похибки методом MSE

$$\sqrt{\frac{(i_1 - a_1)^2 + (i_2 - a_2)^2 + \dots + (i_n - a_n)^2}{n}}$$

Рисунок 9 – Формула знаходження похибки методом Root MSE

$$\frac{\arctan^2(i_1 - a_1) + \dots + \arctan^2(i_n - a_n)}{n}$$

Рисунок 10 – Формула знаходження похибки методом Arctan

Алгоритм знаходження похибки однаковий у всіх методах. Для кожного сету знаходиться похибка, виконуючи операцію віднімання від ідеального варіанту, отриманий. Після чого, обчислюємо квадратний \arctan або підносимо до квадрату, далі додаємо отримані значення та ділимо їх на кількість сетів.

Тренувальний сет – це певна послідовність даних, які використовуються нейронною мережею.

Епоха – це набір ітерацій, також інколи вона включає перевірку якості навчання на підготовлених даних.

Ітерація – це певний лічильник, яких збільшується при використанні одного тренувального сету. Кількість ітерацій дорівнює кількості тренувальних сетів використаних нейронною мережею.

В нейронних мережах велике значення має поняття – навчання нейронних мереж. Навчання – є одним з найважливіших переваг нейронних мереж перед звичайними алгоритмами, адже в процесі навчання, нейронна мережа буде спроможна виявляти складні залежності між вхідними та вихідними даними, які не завжди зможе знайти висококваліфікований спеціаліст. Після операції навчання нейронна мережа зможе знаходити відповідь на основі даних, що входять до тренувальних сетів. Процес навчання полягає в знаходженні оптимальної ваги для синапсів. Існує чотири підходи для навчання нейронних мереж: навчання з учителем, навчання без учителя, навчання з частковим використанням вчителя та навчання з підкріпленням.

- Навчання з учителем. Характерною ознакою навчання з учителем є надання правильних відповідей для відповідних тренувальних сетів. Вага синапсів змінюється в залежності від правильності результуючої відповіді знайденої нейронною мережею. Таким чином тренувальний сет навчить нейронну мережу, після чого при наступній операції вона буде використовувати вже знайдені параметри з тренувальних сетів

для знаходження правильної відповіді. Цей метод переважно використовується, коли є певний великий набір даних для навчання нейронної мережі.

- Навчання без вчителя. Алгоритм навчання без вчителя використовується в тих випадках, коли відомі тільки вхідні сигнали. На їх основі нейронна мережа вчиться пошуку найкращого значення. Означення “найкращого значення” визначається алгоритмом навчання. Зазвичай алгоритм використовує значення так, щоб нейронна мережа знаходила однакові результати для близьких значень. У навчанні без вчителя складно поррахувати точність алгоритму, тому що в вхідних даних немає правильних відповідей або позначок.
- Навчання з частковим використанням вчителя. Характерною ознакою цього методу є те, що використовується два типи тренувальних сетів, розмічені (ті, що мають правильну відповідь) та нерозмічені (ті, що не мають правильних відповідей). Цей метод особливо гарно підходить для використання, коли важко визначити важливі ознаки чи процес розмітки всіх об’єктів є трудомістким завданням.
- Навчання з підкріпленням. Характерною ознакою цього методу є те, що для навчання система використовує взаємозв’язок з іншим середовищем. Відповіддю середовища на отримані рішення є сигнали підкріплення, які використовуються для навчання нейронної мережі.

1 Огляд існуючих рішень

Існує багато способів вирішення даної задачі. Серед яких є одношаровий перцептрон, багатшаровий перцептрон, згорткові нейронні мережі та інші.

1.1. Перцептрон

Перцептрон – це математична або комп'ютерна модель сприйняття інформації мозком. [2, 4] Перцептрон складається з трьох елементів, а саме: сигналів, асоціативних елементів, реагуючих елементів. (див. Рис. 11)

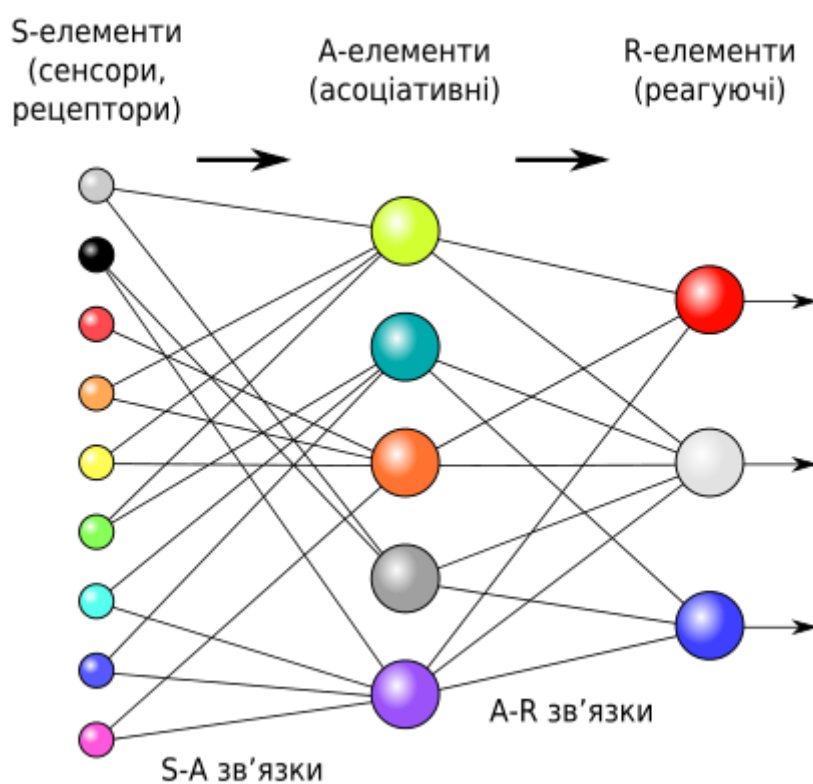


Рисунок 11 – Логічна схема перцептрону

- S-елементом (сенсорним) – це початковий елемент, який від певної дії створює сигнал. Зазвичай вхідний сигнал порівнюють з певною величиною, якщо значення сигналу більше за цю величину, то вихідним сигналом буде значення +1, в іншому випадку — 0.

- А-елементом (асоціативним) називається логічний елемент, значення якого на пряму залежить від його вхідних сигналів, якщо сума значень вхідних сигналів більша за певну величину, то вихідний сигнал цього елемента буде дорівнювати +1, в іншому випадку значення дорівнюватиме 0.
- R-елементом (реагуючим) - це елемент, який на пряму залежить від вхідних сигналів, якщо значення суми вхідних сигналів додатна, то вихідний сигнал дорівнюватиме +1. У випадку, коли сума – від’ємна, вихідний сигнал дорівнюватиме -1, у іншому випадку вихідний сигнал вважається таким, що дорівнює нулю або є невизначеним.

Існують такі види перцептронів:

1. Перцептрон з одним прихованим шаром – це звичайний перцептрон, який складається з трьох шарів (сенсорного, асоціативного та реагуючого).
2. Одношаровий перцептрон – це перцептрон, у якому вхідні елементи на пряму з’єднані з вихідними за допомогою синапсів, у якого кожен S-елемент однозначно відповідає одному A-елементові, S-A зв’язок має вагу +1 і всі A-елементи мають поріг, який дорівнює 1.
3. Багатошаровий перцептрон – це перцептрон у якого присутні додаткові асоціативні шари, методом навчання мережі – є метод зворотного поширення помилки, таким чином проводиться навчання всіх шарів перцептронів.

Перцептрон є гарним способом розпізнавання об’єкта на зображенні, але є декілька недоліків у виборі даного метода:

1. при збільшенні розмірів зображення, кількість ваг буде стрімко збільшуватись;

2. зображення подається у вигляді одновимірного масиву, таким чином ми аналізуємо сусідні пікселі тільки по горизонталі, а при роботі з зображеннями важливо враховувати пікселі і по горизонталі, і по вертикалі.

Для вирішення недоліків цього метода зазвичай використовуються Згорткові нейронні мережі.

1.2. Згорткові нейронні мережі

Згорткові нейронні мережі (ЗНМ) – це один із класів глибинних штучних нейронних мереж прямого поширення, який зазвичай використовується для аналізу зображень. [1, 3]

ЗНМ складається з шарів входу, виходу та декількох прихованих шарів. Зазвичай вони складаються з агрегувальних шарів, згорткових шарів, шарів нормалізації та повноз'єднаних шарів.

1. Згорткові шари

Згортковий шар – це шар отриманий у результаті процесу згортки сигналів виходів попереднього шару, вага ядра згортки є параметром, що навчається, змінюється з кожною вдалою епохою. Згортка – це процес змішування інформації для знаходження потрібних характерних ознак. (див. Рис. 1.2.1) Основними параметрами згорткового шару є доповнення зображення (padding) та зсув (stride). Доповнення зображення – це процес доповнення нульовими пікселями попереднього шару таким чином, щоб зберегти розмір шару після процесу згортки. Зсув – це параметр, що відповідає інтервалу використання ядра згортки, зазвичай він дорівнює 1, таким чином будуть задіяні всі пікселі попереднього шару.

2. Агрегувальні шари

Роботою цього шару є об'єднання сигналів виходів нейронів певного шару до нейрону наступного шару. Наприклад, максимізаційне агрегування знаходить максимальне значення з кожного об'єднання нейронів попереднього шару, вихідним результатом буде максимальне знайдене значення. Процес агрегування доволі простий, спочатку вхідне зображення поділяють на певну кількість прямокутників, які не перекривають один одного, після чого знаходиться максимальне значення в цьому прямокутнику, у випадку використання максимізаційного агрегування. Цей шар дозволяє зменшити просторовий розмір зображення та виділити певний, важливий критерій.

3. Повноз'єднані шари

Повноз'єднаний шар – це кінцевий шар, який використовуються для визначення типу класу аналізуючи кожен нейрон попереднього шару. На вхід даному шар подаються вихідні сигнали з попереднього шару, після чого він визначає властивості, що характерні для певного класу. Зазвичай результатом аналізу цього шару є список ймовірностей відношення до одного із класів.

Для вирішення проблем попереднього метода використовуються наступні принципи:

- Локальне сприйняття
- Роздільність ваг
- Зменшена розмірність

Локальне сприйняття

Якщо в повністю зв'язаній мережі кожен нейрон підключається до кожного пікселя вхідного зображення, то в ЗНМ нейрон отримує на вході тільки обмежену кількість пікселів, як правило це ділянки зображень 3x3, 5x5 пікселів, інколи зустрічається і більше. Наступний нейрон працює з наступною ділянкою, яку може частково перетинати. (див. Рис. 12)

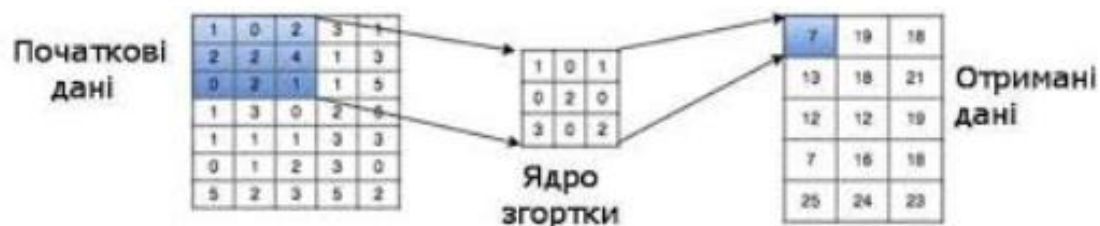


Рисунок 12 – Локальне сприйняття

Операція, що виконується нейроном в цьому типі мережі називається згорткою. Операція згортки виконується за допомогою ядра згортки, розміри якого збігаються з розмірами ділянки зображення. Значення в згортковому шарі отримується за допомогою по елементного множення, ми беремо значення інтенсивності пікселя і множимо його на відповідний елемент ядра згортки, після чого всі отримані значення додаються. Після отримання значення до нейрону може виконуватись обрано функція активації.

Роздільність ваг

Всі нейрони мережі використовують однакові ядра згортки для подальшої обробки різних ділянок зображення, таким чином ми шукаємо нейрон, який є важливим в різних ділянках зображення. За допомогою використання однакових ядер згортки, значно зменшується кількість ваг, які потрібно отримати в процесі навчання.

Зменшення розмірності

Основною метою являється можливість розпізнавання об'єктів, в не залежності від масштабу. На практиці важливіше визначити, що певна ознака знаходиться на зображенні, а не точне її розташування. Для зменшення розмірності в ЗНМ використовуються так звані шари підвибірки. Один нейрон шару підвибірки зв'язаний з певною кількістю нейронів попереднього шару, наприклад 2x2 (див. Рис. 13). Значення в новому шарі визначається на базі значень попереднього шару, зазвичай використовується правило Максимального значення.

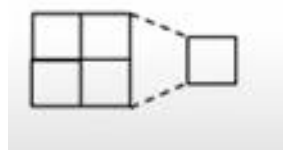


Рисунок 13 – Шар підвибірки

Згорткові нейронні мережі складаються з почергових шарів згортки і підвибірки, з'єднаних між собою. (див. Рис. 14)



Рисунок 14 – Вигляд згорткової нейронної мережі

Важливим фактом, що сильно впливає на правильність знаходження вагових коефіцієнтів є регуляризація. Регуляризація – це процес при якому додаються додаткові данні для рішення поставлених завдань, або для уникнення процесу перенавчання. Перенавчання – це процес при якому нейронна мережа адаптується до особливостей тренувального сету, набір, що використовується для навчання, а не до спільних особливостей і при роботі з новими даними, які не входять до тренувального сету, нейронна мережа працює погано. При процесі перенавчання зменшується узагальнююча здатність, що означає зменшення ефективності роботи на нових даних, тому зазвичай використовують два набори даних: тренувальний набір та тестовий набір. Тренувальний набір – це набір даних, який зазвичай використовується для навчання нейронної мережі.

Тестовий набір – це набір даних, який зазвичай використовується для перевірки якості навчання мережі.

Основні типи регуляризації:

1. Емпіричні

1.1. Виключення – це процес, при якому на кожному етапі навчання виділяється нова, випадково обрана мережа для навчання. Нейрони, що використовуються для навчання, обираються випадково.

1.2. Виключення з'єднань – це процес, при якому є певний коефіцієнт, який називається коефіцієнтом дропауту, що показує можливість відключення певного з'єднання. Тобто, кожен наступний вузол має на вході випадкову кількість активних вузлів попереднього шару.

1.3. Штучні дані – це процес додавання або створення нових даних для тренувального набору, наприклад, при розпізнаванні зображень можливо створити нові дані використовуючи вже існуючі, пропорційно обрізаючи певний набір зображень.

2. Явні

2.1. Рання зупинка – це процес спостереження за результатами навчання і тестування на перевірконому наборі даних, при якому є можливість виходу з навчання нейронної мережі при зменшенні її узагальнюючих здібностей.

2.2. Кількість параметрів – це процес обмеження певних параметрів нейронної мережі. До таких параметрів зазвичай входять: кількість вузлів у кожному прихованому шарі, глибина мережі та розмір фільтру.

Основні гіперпараметри нейронної мережі, які впливають на швидкість та якість навчання:

- Кількість шарів та нейронів в мережі;
- Кількість епох навчання;
- Кількість зображень для навчання;

- Коефіцієнт швидкості навчання мережі;
- Архітектура нейронної мережі (шарові та повністю зв'язані нейронні мережі) [12].

Шарові нейронні мережі – це нейронні мережі, нейрони яких поділяються на окремі шари, таким чином інформація оброблюється пошарово.

Коефіцієнт швидкості навчання мережі [11] – це параметр градієнтних алгоритмів навчання нейронних мереж, який дозволяє регулювати зміну ваг на кожній ітерації.

Архітектура нейронної мережі – це

Для реалізації поставленої задачі існує багато мов програмування з великим функціоналом та базою бібліотек. Бібліотека – це набір підпрограм та об'єктів, які використовуються, зазвичай, для зручної та швидкої розробки програм. На мою думку, найкращими представниками мов програмування для найбільш легкої та точної розробки нейронних мереж є Java (Deeplearning4j) за легкість у вивченні та велику кількість створених іншими користувачами бібліотек та Python (Keras) за доволі легку структуру та велику популярність, що означає можливість покращення алгоритмів з часом.

- Deeplearning4j (DL4J) [8] – бібліотека, що використовується при створенні нейронних мереж глибокого навчання. Глибоке навчання [7] – це один з методів машинного навчання, який дозволяє вирішувати задачу передбачення результату використовуючи набір вхідних даних. У DL4J є можливість створення згорткових нейронних мереж, що є важливим фактором, наприклад, при обробці зображень. Всі налаштування відбуваються за допомогою програмного коду. В бібліотеці використовуються згорткові, повнозв'язні та пулінгові шари, за допомогою цих особливостей у

DL4J є можливість реалізації майже будь-яких нейронних мереж глибокого навчання.

- Keras [9] – це бібліотека, що орієнтована на нейронні мережі, використовує мову програмування Python. Keras – гарний варіант для швидкого та ефективного розроблювання нейронних мереж, адже вона зручна в використанні, модульна та розширювальна. Бібліотека складається з багатьох широко вживаних блоків для створення нейронних мереж, таких як шари, функції активації, оптимізувальники та безліч інших інструментів для полегшення роботи із зображеннями та тестом. Також Keras містить підтримку згорткових нейронних мереж, що ще значно спрощує створення мереж глибокого навчання.

2 Вибір методу рішення

Для рішення поставленої задачі обрано Згорткову нейронну мережу (див. рис. 15). Мову програмування – Python та бібліотеку для реалізації - Keras. Методом навчання обрано – навчання з учителем методом зворотного поширення помилки. Метод зворотного поширення помилки [10] – це метод для обчислення градієнту, який використовується при знаходженні нових вагових коефіцієнтів. Характерною ознакою цього методу є поширення сигналу помилки починаючи з виходу мережі до її входу, у напрямку зворотному до напрямку прямого розповсюдження сигналу. Градієнт – це вектор, який вказує на напрямок найбільшого зростання деякої величини.

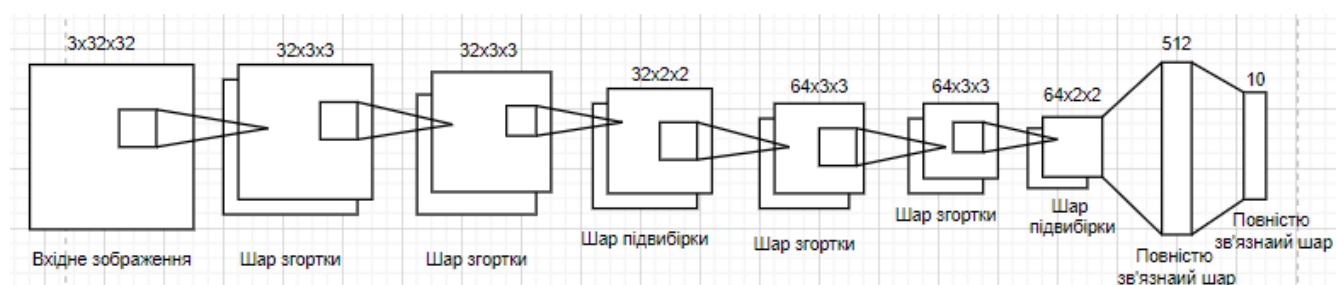


Рисунок 15 – Схематичне зображення згорткової нейронної мережі

2.1. Конфігурація бібліотек потрібних для виконання задачі

Для налаштування в використанні допоміжного функціоналу були підключені наступні бібліотеки:

```
import keras
import cv2
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D
from IPython.display import Image
from keras.preprocessing import image
```

```
from tensorflow.keras.models import load_model
```

2.2. Конфігурація каталогу для збереження навченої моделі та зображень

За допомогою допоміжного функціоналу було обрано основний каталог для збереження основних робочих даних:

```
import os
from google.colab import drive
drive.mount('/content/drive')
os.chdir('/content/drive/My Drive/test')
```

2.3. Вибір зображень для навчання мережі та для її тестування

Для навчання та тестування Згорткової нейронної мережі обрано набір CIFAR-10. Даний набір складається з 10 класів: літаків, автомобілів, птахів, котів, оленів, собак, жаб, коней, кораблів та вантажівок. Кількість зображень для навчання складає 50000, 5000 зображень для кожного класу. Кількість зображень для тестування – 10000, 1000 зображень для кожного класу. Розмір зображень 32x32, кількість кольорів 3 (RGB).

2.4. Програмна реалізація

Основні методи і параметри бібліотеки, що були використані:

- Sequential – послідовна модель, в якій шари розташовані послідовно;
- Conv2D – згортковий шар, що працює з двомірними даними, перший параметр вказує на кількість фільтрів, потім визначається розмір ядра згортки, далі є можливість вказати не обов'язкові параметри, такі як padding та input_shape – розміри вхідних даних. Зазвичай виділяють 3 види padding: valid, same, full. Перший відповідає за зменшення розмірності матриці, адже згортка виконується тільки там, де вхідні дані та фільтр повністю перекриваються. Другий відповідає за

збереження розмірів матриці, за допомогою додавання додаткових нульових пікселів. Третій допомагає збільшити вихідну матрицю, зазвичай не використовується;

- Activation – визначає функцію активації;
- MaxPooling2D – шар підвибірки, метод MaxPooling, тобто вихідним значенням буде максимальне значення із заданих розмірів ядра підвибірки, для цього передається параметр pool_size;
- Dropout – шар регуляризації, який показує з якою ймовірністю не будуть задіяні нейрони, наприклад, Dropout(0.25) відповідає відключенню нейрона з ймовірністю 25%, це дозволяє запобігти перенавчанню мережі;
- Flatten - спосіб перетворення двовимірних даних в масив;
- Dense – пов'язаний шар, потрібний для класифікації зображення;
- Softmax – функція активації, що відповідає ймовірності відповідності деякого класу, сума ймовірностей дорівнює 1;
- Shuffle – параметр, який відповідає за перемішання всіх даних на початку епохи.

Далі представлена програмна реалізація обраного методу:

```
# кількість зображень при навчанні за одиницю проходження
batch_size = 50
# кількість класів
num_classes = 10
# кількість епох для навчання
epochs = 70

# каталог у який буде збережено результат навчання
directory_for_save = os.path.join(os.getcwd(), 'saved_models')
model_name = 'model.h5'

# дані для навчання та тестування
# cifar10.load_data() - спеціально функція для зчитування потрібних з
ображень
(train_data, train_ra_data), (test_data, test_ra_data) = cifar10.load_data()
```

```

print('x_train shape:', train_data.shape)
print(train_data.shape[0], 'кількість зображення для навчання')
print(test_data.shape[0], 'кількість зображень для тестування')

# Перетворити вектори класів у категорії
Train_ra_data = keras.utils.to_categorical(train_ra_data, num_classes)
Test_ra_data = keras.utils.to_categorical(test_ra_data, num_classes)

model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same',
                input_shape=(32, 32, 3)))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.35))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.35))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes))
model.add(Activation('softmax'))

model.compile(loss='categorical_crossentropy',
              optimizer='SGD',
              metrics=['accuracy'])

# Нормалізація даних інтенсивності пікселів
train_data = train_data.astype('float32')
test_data = test_data.astype('float32')
train_data /= 255
test_data /= 255

model.fit(train_data, Train_ra_data,

```

```
batch_size=batch_size,
epochs=epochs,
validation_split=0.15,
shuffle=True)
```

```
# Збереження навченої моделі в заданий каталог
model.save(os.path.join(directory_for_save, model_name))
```

```
# Оцінка тренованої моделі
scores = model.evaluate(test_data, Test_ra_data, verbose=1)
print('Точність передбачення на тестовій вирібці:', scores[1])
# Точність передбачення на тестовій вирібці: 0.7947999835014343
```

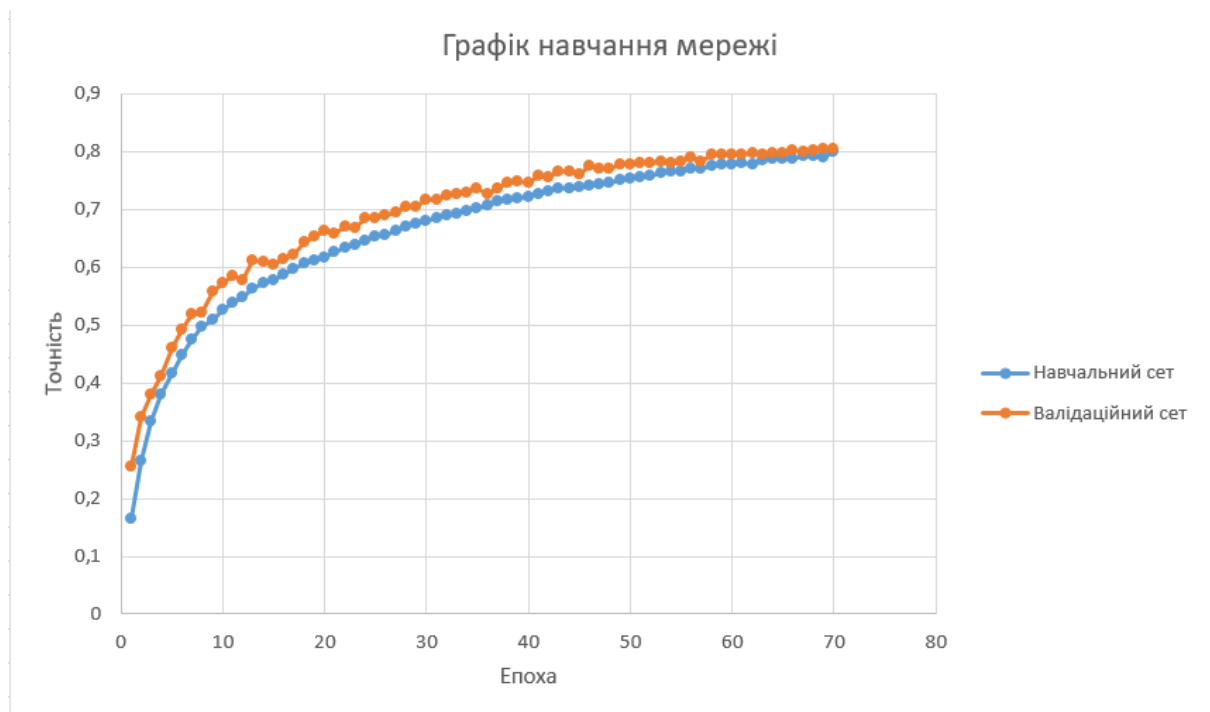


Рисунок 16 – Графік навчання нейронної мережі

2.5. Перевірка правильності навчання моделі

Для тестування обрано по одному зображенню кожного класу. Приклад тестування:

```
classes = ['Літак', 'Автомобіль', 'Птах', 'Кіт', 'Олень', 'Собака', 'Жаба', 'Кінь', 'Корабель', 'Вантажівка']
```

```
# завантаження зображення зі зміною розміру
img = image.load_img(os.path.join('7.jpg'), target_size=(32, 32))

# завантаження навченої нейронної мережі
model = load_model(os.path.join('saved_models/model.h5'))

test = image.img_to_array(img)
test = test.astype('float32')
test /= 255

test = test.reshape(1, 32, 32, 3)
prediction = model.predict(test)

max_index = prediction.argmax()

# Вивід інформації про зображення
print ('Номер класу - ', max_index)
print ('Назва класу - ', classes[max_index])
print ('Класифікація зображення(%%)')
for i in range (10):
    print (classes[i], '\t', prediction[0][i] * 100)
```

Зображення над яким проводиться тестування (див. рис. 17)



Рисунок 17 – Пташка

Вивід результатів перевірки

Номер класу - 2

Назва класу - Птах

Класифікація зображення(%)

Літак	32.358309626579285
Автомобіль	0.014640072186011821
Птах	67.59305000305176
Кіт	0.0021359182937885635
Олень	2.5954834725894216e-06
Собака	0.006910759111633524
Жаба	0.02124554885085672
Кінь	0.0005533674084290396
Корабель	0.0007195073521870654
Вантажівка	0.0024323773686774075

Далі представлено тестування інших класів зображень:

- Перевірка зображення класу «Автомобіль» (див. рис. 18)



Рисунок 18 – Автомобіль

Номер класу - 1

Назва класу - Автомобіль

Класифікація зображення(%)

Літак	0.11056540533900261
Автомобіль	53.06816101074219
Птах	0.00024734790713409893
Кіт	4.783101417160651e-05
Олень	0.000217419074033387
Собака	3.470086653578619e-05
Жаба	0.02494936343282461
Кінь	9.505059210823674e-05
Корабель	0.025794937391765416
Вантажівка	46.76989018917084

- Перевірка зображення класу «Олень» (див. рис. 19)



Рисунок 19 – Олень

Номер класу - 4

Назва класу - Олень

Класифікація зображення(%)

Літак	5.915205553174019
Автомобіль	0.2452577231451869
Птах	0.9237437509000301
Кіт	0.41532330214977264
Олень	63.115233182907104
Собака	0.19647935405373573
Жаба	4.633767902851105
Кінь	8.852211385965347
Корабель	3.9515718817710876
Вантажівка	11.75120398402214

- Перевірка зображення класу «Кінь» (див. рис. 20)



Рисунок 20 – Кінь

Номер класу - 7

Назва класу - Кінь

Класифікація зображення(%)

Літак	2.184496819972992
Автомобіль	1.115687657147646
Птах	5.580891668796539
Кіт	5.325676873326302
Олень	23.915232717990875
Собака	19.15712058544159
Жаба	10.637897998094559
Кінь	27.342355251312256
Корабель	0.7807529531419277
Вантажівка	3.9598889648914337

- Перевірка зображення класу «Корабель» (див. рис. 21)



Рисунок 21 – Корабель

Номер класу - 8

Назва класу - Корабель

Класифікація зображення(%)

Літак	18.146105110645294
Автомобіль	0.07483469089493155
Птах	0.7190419360995293
Кіт	0.0037543271901085973
Олень	0.11527976021170616
Собака	0.0005126946234668139
Жаба	0.0019701523342519067
Кінь	0.3138904692605138
Корабель	79.69565987586975
Вантажівка	0.928951520472765

- Перевірка зображення класу «Кіт» (див. рис. 22)



Рисунок 22 – Кіт

Номер класу - 3

Назва класу - Кіт

Класифікація зображення(%)

Літак 5.275500938296318

Автомобіль 3.149699419736862

Птах 6.915020942687988

Кіт 23.07952046394348

Олень 14.760759472846985

Собака 12.684302031993866

Жаба 13.0398228764534

Кінь 7.592355459928513

Корабель 5.89064247906208

Вантажівка 7.612379640340805

- Перевірка зображення класу «Собака» (див. рис. 23)

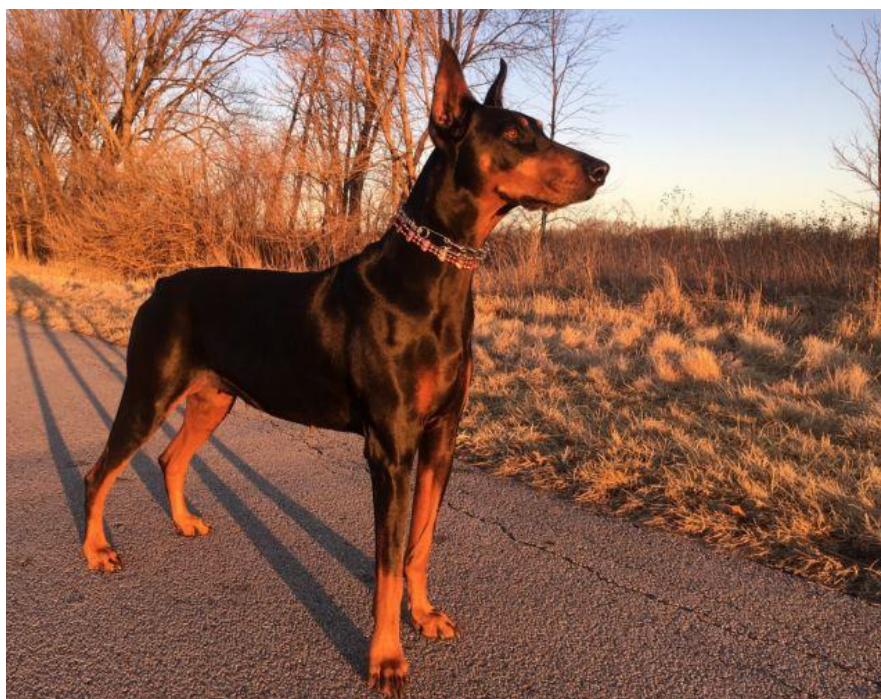


Рисунок 23 – Собака

Номер класу - 5

Назва класу - Собака

Класифікація зображення(%)

Літак	1.3887377455830574
Автомобіль	0.4543542396277189
Птах	1.3406538404524326
Кіт	4.112888872623444
Олень	6.031158939003944
Собака	42.91301155090332
Жаба	22.4804819598793983
Кінь	4.618049040436745
Корабель	1.0462163016200066
Вантажівка	15.614451467990875

- Перевірка зображення класу «Жаба» (див. рис. 24)



Рисунок 24 – Жаба

Номер класу - 6

Назва класу - Жаба

Класифікація зображення(%)

Літак	0.2825458301231265
Автомобіль	0.503220921382308
Птах	0.148959516081959
Кіт	0.04175304784439504
Олень	0.13595876516774297
Собака	0.07963296375237405
Жаба	96.18765115737915
Кінь	0.006320164538919926
Корабель	0.20656439010053873
Вантажівка	2.407388389110565

- Перевірка зображення класу «Літак» (див. рис. 25)



Рисунок 25 – Літак

Номер класу - 0

Назва класу - Літак

Класифікація зображення(%)

Літак	99.40244555473328
Автомобіль	1.9395098149743717e-05
Птах	0.005618139402940869
Кіт	6.262939677981194e-05
Олень	0.0030761737434659153
Собака	2.6532214292274148e-05
Жаба	7.316907613130752e-06
Кінь	3.629589002684952e-05
Корабель	0.5884717218577862
Вантажівка	0.00023719142063782783

- Перевірка зображення класу «Вантажівка» (див. рис. 26)



Рисунок 26 – Вантажівка

Номер класу - 9

Назва класу - Вантажівка

Класифікація зображення(%)

Літак	0.020580091222655028
Автомобіль	0.35914420150220394
Птах	5.0008402752155234e-06
Кіт	8.907669979407729e-06
Олень	1.1096415164502105e-05
Собака	7.276678815060222e-06
Жаба	7.980318628142413e-07
Кінь	0.00011478658734631608
Корабель	0.0006399460744432872
Вантажівка	99.6194839477539

Висновок

Виконано завдання з переддипломної практики на тему «Створення інтелектуальної інформаційної системи розпізнавання зображення».

Розглянуто існуючі способи створення нейронних мереж. На основі аналізу було обрано та реалізовано систему розпізнавання об'єктів на зображенні, а саме згорткову нейронну мережу.

Обрано основні зображення для навчання системи та її тестування. Підібрано параметри для збільшення величини ймовірності передбачення правильного класу зображень. Протестовано створену систему, використовуючи зображення обраних класів.

Створено інформаційну систему для розпізнавання певних класів зображень, досягнуто максимального значення правильності передбачення відповідного класу зображень, ця величина дорівнює ~80%, обрано набір даних для навчання системи та її тестування, система протестована використовуючи зображення всіх обраних класів.

Список використаних джерел

1. Основні ідеї нейронних мереж [Електронний ресурс] – Режим доступу до ресурсу: <http://masters.donntu.org/2006/kita/kornev/library/16.htm>
2. Опис алгоритму персептрона [Електронний ресурс] – Режим доступу до ресурсу: <https://neurohive.io/ru/osnovy-data-science/perseptron-rozenblatta-mashina-kotoraja-smogla-obuchatsja/>
3. Опис алгоритму згорткових нейронних мереж [Електронний ресурс] – Режим доступу до ресурсу: <https://cyberleninka.ru/article/n/obzor-metodov-obucheniya-glubokih-neyronnyh-setey/viewer>
4. Інформаційна сторінка основних методів Keras [Електронний ресурс] – Режим доступу до ресурсу: https://keras.io/guides/functional_api/
5. Основні положення нейронної мережі персептрон [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD>
6. Tariq Rashid Make your own neural network /T. Rashid, 2016 – 222 с.
7. Глубокое обучение Погружение в мир нейронных сетей / Николенко С., Архангельская Е., Кадурын А. – 481с.
8. Інформаційна сторінка набору CIFAR-10 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.cs.toronto.edu/~kriz/cifar.html>
9. Глибоке навчання [Електронний ресурс] – Режим доступу до ресурсу: <https://neurohive.io/ru/osnovy-data-science/glubokoe-obuchenie-deep-learning-kratkij-tutorial/>
10. Способи створення нейронної мережі мовою Java [Електронний ресурс] – Режим доступу до ресурсу: <https://moluch.ru/archive/152/43043/>
11. Основна ідея бібліотеки Keras [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Keras>

12. Основні положення методу зворотного поширення помилки [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Backpropagation>
13. Основна сторінка Keras [Електронний ресурс] – Режим доступу до ресурсу: <https://keras.io/>
14. Інформаційна сторінка коефіцієнту швидкості навчання [Електронний ресурс] – Режим доступу до ресурсу: <https://wiki.loginom.ru/articles/learning-rate.html>
15. Інформаційна сторінка архітектур нейронних мереж [Електронний ресурс] – Режим доступу до ресурсу: <https://www.intuit.ru/studies/courses/6/6/lecture/178?page=3>
16. Інформаційна сторінка структури нейронних мереж [Електронний ресурс] – Режим доступу до ресурсу: <https://proglib.io/p/about-neural-networks>
17. Стохастичний градієнтний спуск [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Stochastic_gradient_descent
18. Штучні нейронні мережі [Електронний ресурс] – Режим доступу до ресурсу: <https://www.slideshare.net/LesiaSobolevska/ai-8-75922490>