

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління  
(повна назва)

Кафедра електронних обчислювальних машин  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

Рівень вищої освіти другий (магістерський)

Модель прогнозування на основі глибоко навчання  
для класифікації об'єктів

(тема)

Виконав:

студент II курсу, групи СПМ-21-2  
Бевзюк Р.Г.  
(прізвище, ініціали)

Спеціальність 123 «Комп'ютерна інженерія»  
(код і повна назва спеціальності)

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

Освітня програма Системне програмування  
(повна назва освітньої програми)

Керівник: доц. Федорченко В.М.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерної інженерії та управління \_\_\_\_\_

Кафедра \_\_\_\_\_ електронних обчислювальних машин \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 «Комп'ютерна інженерія» \_\_\_\_\_  
(код і повна назва)

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

Освітня програма \_\_\_\_\_ Системне програмування \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту \_\_\_\_\_ Бевзюку Роману Григоровичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Модель прогнозування на основі глибоко навчання  
для класифікації об'єктів

затверджена наказом по університету від “ 03 ” квітня 2023 р. № 318 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 17 травня 2023 р.

3. Вхідні дані до роботи \_\_\_\_\_

Нейроні мережі \_\_\_\_\_

Машинне навчання \_\_\_\_\_

Розпізнавання об'єктів \_\_\_\_\_

Робота CPU, GPU \_\_\_\_\_

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

Аналіз предметної області \_\_\_\_\_

Вимоги до об'єктів \_\_\_\_\_

Реалізація системи класифікації за допомогою ЗНМ \_\_\_\_\_

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 13

---

---

---

---

---

---

---

---

---

---

6. Консультанти розділів роботи (заповнюється 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	Термін виконання етапів роботи	Примітка
1	Аналіз систем розпізнавання та їх класифікації	03.04.23-11.04.23	
2	Аналіз методів глибокого навчання	12.04.23-20.04.23	
3	Аналіз програмно-апаратного комплексу	21.04.23-26.04.23	
4	Дослідження недоліків в існуючих системах	27.04.23-30.04.23	
5	Розробка програмного забезпечення		
6	Проведення експериментів	1.05.23-05.05.23	
7	Оформлення матеріалів кваліфікаційної роботи	06.05.23-10.05.23	
8	Подання кваліфікаційної роботи керівникові та її попередній захист	10.05.23-11.05.23	
9	Подання кваліфікаційної роботи на рецензування	12.05.23-16.05.23	

Дата видачі завдання 03 квітня 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Федорченко В.М.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 50 с., 16 рис., 1 дод., 8 джерел.

НЕЙРОНА МЕРЕЖА, МАШИННЕ НАВЧАННЯ, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, CPU, GPU.

Метою кваліфікаційної роботи є побудова моделі прогнозування для класифікації об'єктів для системи яка використовується в складських підприємствах.

У ході виконання кваліфікаційної роботи розглянуто правила розміщення штрих-кодів на товарах. Обрано інструментарій для реалізації моделі класифікації товарів в яких може бути скритий штрих-код. Реалізовані вимоги до системи та проведено моделювання системи з отриманням результатів.

## ABSTRACT

Master's thesis: 50 pages, 16 figures, 1 appendices, 8 sources.

NEURAL NETWORK, MACHINE LEARNING, OBJECT  
RECOGNITION, CPU, GPU

The purpose of the qualification work is to build a forecasting model for the classification of objects for the system used in warehouse enterprises.

In the course of the qualification work, the rules for placing barcodes on goods were considered. A toolkit was selected for the implementation of the model of classification of goods in which a hidden barcode may be. The requirements for the system were implemented and the system was modeled with the results obtained.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	7
ВСТУП .....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	9
1.1 Основні питання при розпізнаванні об'єктів .....	9
1.2 Машинне навчання .....	11
1.3 Вибір оптимального алгоритму навчання під наглядом.....	12
1.4 Впровадження машинного навчання .....	14
1.4.1 Деревя рішень.....	16
1.4.2 Штучні нейронні мережі .....	18
1.4.3 Support Vector Machines.....	20
2 ВИМОГИ ДО ОБ'ЄКТІВ.....	23
2.1 Правила розміщення штрих-коду.....	23
2.2 Визначення параметрів.....	24
3 РЕАЛІЗАЦІЯ СИСТЕМИ КЛАСИФІКАЦІЇ ЗА ДОПОМОГОЮ ЗНМ .....	28
3.1 Згорткова нейронна мережа .....	28
3.2 Створення ЗНМ в Matlab.....	32
3.3 Навчання ЗНМ.....	35
ВИСНОВКИ.....	40
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	42
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	43

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ  
І ТЕРМІНІВ

ШНМ – штучні нейроні мережі

ЗНМ – згорткова нейронна мережа

CNN – Convolutional neural network

SVM – Support Vector Machines

## ВСТУП

Однією з основних причин швидкого зростання глобального ринку автоматизації складів є значне зростання електронної комерції та вимоги до виконання замовлень, які створюються в результаті (швидка доставка, відстеження тощо). Існує підвищена потреба в системах, здатних обробляти великі обсяги замовлень із кількох позицій.

Найбільша проблема в реалізації повністю автономних систем виконання замовлень полягає в ідентифікації та обробці сторонніх предметів. Сучасні методи сегментації зображення забезпечують достатню точність для обробки сторонніх об'єктів для операцій вибору та розміщення. Ці методи сегментації зображення дозволяють ідентифікувати та розрізнити геометричні форми, краї, поверхні та інші характеристики, достатні для надійної обробки сторонніх об'єктів. Однак цих ознак недостатньо для ідентифікації сторонніх предметів. Це обмеження технології зумовлює необхідність присутності фізичного працівника, який забезпечує ідентифікацію сторонніх предметів у процесі виконання замовлення. Для реалізації повної автоматизації система виконання замовлень повинна мати можливість автономно розпізнавати об'єкт, який обробляється, щоб об'єкт можна було транспортувати до відповідного місця призначення.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Основні питання при розпізнаванні об'єктів

Розробити систему, яка може розпізнавати сторонні об'єкти та обробляти їх, важко через безліч візуальних явищ, які відбуваються в реальному світі, наприклад оклюзію об'єктів або світлові ефекти. Відрізнити два об'єкти, які майже однакові за формою та кольором, але є абсолютно різними, є однією з найскладніших проблем, яку можна вирішити за допомогою сучасних технологій комп'ютерного зору. Приклад цієї ситуації наведено на рисунках 1.1 та 1.2, де два майже однакові на вигляд продукти все ще мають різний вміст.



Рисунок 1.1 – Паприка копчена (мелена)



Рисунок 1.2 – Паприка (мелена)

Щоб розробити надійну систему обробки замовлень, багато з цих складних або незвичайних випадків потрібно правильно визначити. Щоб досягти цього, властивості, унікальні для кожного елемента, повинні бути знайдені та використані для правильної класифікації кожного елемента. У деяких із найскладніших випадків, коли майже не видно візуальної різниці, залишається унікальна властивість, яку можна використовувати для правильної класифікації: штрих-код. Кожен товар, який обробляється на складах, містить штрих-код, який у певний момент сканується фізичним працівником, щоб визначити його призначення під час процесу повторного збирання замовлення.

Основна увага в системі розпізнавання об'єктів буде зосереджена на використанні штрих-коду як унікальної властивості для правильної ідентифікації об'єктів. Якщо штрих-код кожного об'єкта можна знайти та надійно розпізнати, кожен об'єкт можна ідентифікувати та правильно обробити. Незважаючи на те, що існують стандартні вказівки щодо розміщення штрих-кодів, певно, не всі товари відповідають цим правилам. Дуже цікаво дослідити, чи існує корисний зв'язок між характеристиками об'єкта та розташуванням штрих-коду. Якщо такий зв'язок існує, його можна використовувати для дуже ефективного підбору та сканування товарів на

складі, усуваючи потребу людей у ручній ідентифікації об'єктів.

## 1.2 Машинне навчання

Машинне навчання – це галузь, яка зосереджена на розробці алгоритмів, які роблять прогнози на основі даних. Завдання машинного навчання має на меті ідентифікувати (вивчити) функцію  $f: X \rightarrow Y$ , що відображає вхідну область  $X$  (даних) на вихідну область  $Y$  (можливих прогнозів) [1]. Функції  $f$  вибираються з різних класів функцій, залежно від типу алгоритму навчання, який використовується. Для завдання класифікації точність системи зазвичай вибирається як міра ефективності, де точність визначається як пропорція, для якої система правильно створює вихідні дані. Досвід, якому піддаються алгоритми машинного навчання, це набори даних. Ці набори даних містять набір прикладів, які використовуються для навчання та тестування цих алгоритмів.

Алгоритми машинного навчання можна в основному класифікувати на три категорії за типом наборів даних, які використовуються як досвід. Ці категорії включають контрольоване навчання, неконтрольоване навчання та навчання з підкріпленням. Системи контрольованого навчання використовують мічені набори даних  $(x, y) \in X \times Y$ , де  $x$  представляє точку даних, а  $y$  відповідне істинне передбачення для  $x$ . Цей навчальний набір пар введення-виведення використовується для пошуку детермінованої функції, яка відображає будь-який вхід на вихід, прогнозуючи майбутні спостереження вводу-виводу, мінімізуючи помилки, наскільки це можливо. Системи неконтрольованого навчання використовують немарковані набори даних для навчання системи. Метою неконтрольованого навчання є отримання структури з немаркованих даних шляхом дослідження подібності між парами об'єктів і зазвичай пов'язане з оцінкою щільності або групуванням даних. Системи навчання з підкріпленням відчують не фіксований набір даних, а петлю зворотного зв'язку між системою та її

досвідом [2]. Розглядається динамічне середовище, в якому як дані спостерігаються трійки стан-дія-винагорода. Метою навчання з підкріпленням є відображення ситуацій у діях з метою максимізації винагорода. Існують інші навчальні системи, які є комбінацією двох категорій, наприклад напівкероване навчання, яке використовує як позначені, так і немарковані дані [3].

Існує велика різноманітність завдань, які можна вирішити за допомогою машинного навчання. Двома популярними завданнями машинного навчання є регресійний аналіз і класифікація. У регресійному аналізі зв'язок між змінними є наближеним для успішного прогнозування значення за певних вхідних даних. Ця задача розв'язується виведенням функції  $f: R^n \rightarrow R$ , що відповідає даним [4]. Регресійний аналіз можна використовувати, наприклад, для прогнозування майбутніх цін на акції у світі торгівлі. Під час класифікації машину просять визначити категорію  $n$ , до якої належить певний вхід. Задачу можна розв'язати, вивівши функцію  $f: R^n \rightarrow \{1, \dots, n\}$  [5]. Популярною проблемою класифікації є розпізнавання об'єктів для інтелектуальних систем. Класифікація може бути використана, наприклад, для класифікації об'єктів на складі, щоб визначити правильне призначення кожного об'єкта, за допомогою сучасного розпізнавання об'єктів із використанням алгоритмів глибокого навчання [5].

### 1.3 Вибір оптимального алгоритму навчання під наглядом

Щоб визначити, яка техніка машинного навчання підходить для конкретної програми, можна проаналізувати аспекти, необхідні для оптимального конвеєра керованого машинного навчання. В [6] описується конвеєр, який можна використовувати для створення успішного класифікатора, який добре узагальнює нові екземпляри даних. Цей алгоритм зображено на рисунку 1.3.

Перші два кроки цього конвеєра є найважливішими і значною мірою визначають продуктивність класифікатора. Ідентифікація необхідних даних складається з визначення та вибору найбільш релевантних характеристик. Виключаючи нерелевантні або надлишкові функції, розмірність даних можна зменшити. Забагато нерелевантної або надлишкової інформації може перешкодити алгоритму навчання знайти шаблони в даних або навіть призвести до помилкових результатів. Етап попередньої обробки використовується для вирішення цієї надлишковості інформації та часто використовується для роботи з шумом або відсутніми значеннями. Кінцевим результатом етапу попередньої обробки є введення навчального набору даних.

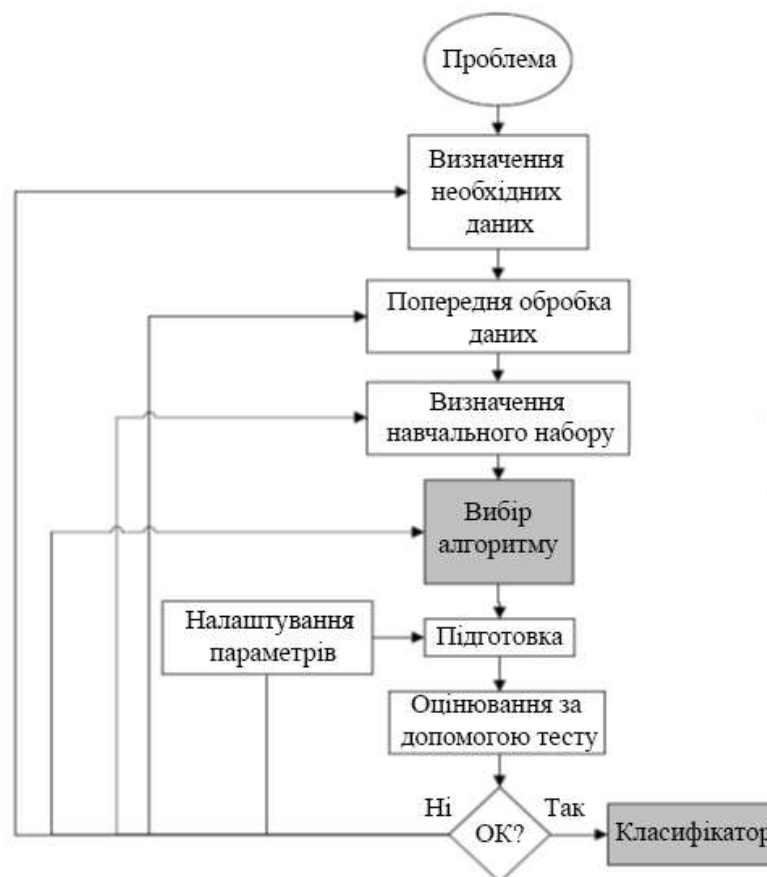


Рисунок 1.3 – Конвеєр керованого машинного навчання, який використовується для створення успішного класифікатора

Метод для опису наборів даних, які використовуються для навчання та тестування алгоритмів машинного навчання, полягає в матриці проектування [3]. Матриця проектування – це матриця, яка складається з усіх точок даних, де кожен стовпець може відповідати певній ознаці. Наприклад, якщо використовувати набір із 10 фотографій об'єкта з роздільною здатністю 1600x1200 із трьома функціями для кожної фотографії, набір даних можна представити за допомогою матриці дизайну  $X \in R^{10 \times 1600 \times 1200 \times 3}$ . Набори даних, які використовуються для навчання та тестування алгоритмів машинного навчання, варіюються від відносно простих до великих і складних наборів даних. Прикладом відносно простого набору даних є набір даних Iris [7]. Цей набір даних містить 150 зразків, кожен з яких містить чотири екземпляри даних і може бути описаний матрицею проектування  $X \in R^{150 \times 4}$ . З іншого боку, під час роботи з фотографіями набори даних можуть стати дуже великими через високу роздільну здатність зображення. Зображення з роздільною здатністю 1900x1080 пікселів, де кожен піксель представляє точку даних зі значеннями  $x$ ,  $y$  і  $z$ , може легко створити кількість 6156000 точок даних на зображення. Через велику кількість точок даних для кожного прикладу необхідна велика обчислювальна потужність для навчання та тестування алгоритму машинного навчання. Оскільки кількість отриманих даних зростає щодня, дедалі складніші дані можна використовувати для навчання алгоритмів машинного навчання. У сфері комп'ютерного зору набори даних можуть складатися з тисячі зображень, кожне з яких може містити десятки функцій.

#### 1.4 Впровадження машинного навчання

Останнє десятиліття ознаменувалося швидким і значним зростанням широкомасштабних програм машинного навчання, що пояснюється двома

основними причинами:

- 1 збільшення великих наборів даних у багатьох сучасних програмах;
- 2 зростання та еволюція структур програмування та апаратних архітектур, які можуть використовуватися багатьма алгоритмами навчання [8].

Великі набори даних часто збираються та зберігаються на великих розподілених платформах зберігання, що спонукає до розробки алгоритмів машинного навчання, які можуть використовувати ці ефективно збережені дані. Удосконалення підходів програмування та архітектура апаратного забезпечення дозволяють ефективно й одночасно обробляти ці великі обсяги збережених даних. Крім того, збільшення використання датчиків, які приймають рішення в режимі реального часу на основі великої суми багатовимірних і складних функцій, викликає зростаючий попит на використання апаратних архітектур, які дозволяють проводити ефективні паралельні обчислення в навчальних програмах. Два приклади технологій, які мають спорідненість з такими даними, це візуальне виявлення об'єктів для автономних систем і розпізнавання мови [9].

Мета дослідження алгоритмів машинного навчання – надати огляд найефективніших алгоритмів і дізнатися, наскільки добре вони можуть бути реалізовані у великомасштабних програмах машинного навчання. З метою проведення ретельного, але ефективного дослідження, були сформульовані наступні питання дослідження літератури.

1 Які алгоритми контрольованого машинного навчання є найбільш успішними?

Це досліджується шляхом аналізу опублікованих статей, які порівнюють продуктивність цих алгоритмів на різних наборах даних. Ефективність оцінюється за трьома основними критеріями:

- швидкість класифікації: час, необхідний для класифікації екземпляра запиту;
- точність: кількість зроблених правильних прогнозів, поділена на

загальну кількість прогнозів;

- стійкість до шуму: наскільки добре алгоритм машинного навчання може впоратися з шумом, таким як нерелевантні або зайві функції.

Найважливіше те, що досліджуються найбільші вузькі місця та проблеми, а також береться до уваги зрозумілість моделі.

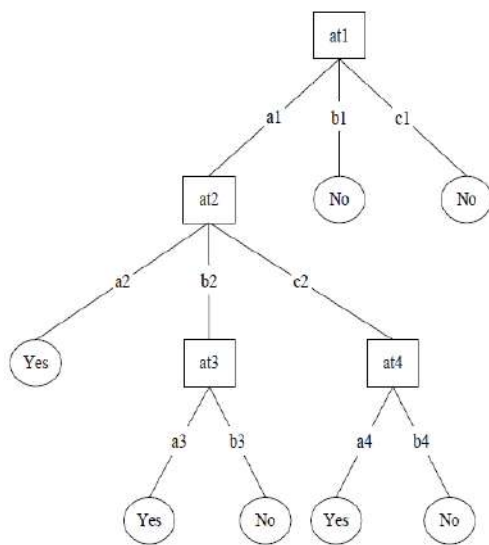
2 Які можливості масштабування для цих успішних алгоритмів машинного навчання?

Теоретичний аналіз складності виконується на найуспішніших алгоритмах, щоб визначити їх потенціал масштабованості. Крім того, аналізується практична життєздатність цих масштабованих алгоритмів.

#### 1.4.1 Древа рішень

Алгоритми дерев рішень складаються з дерев, які класифікують дані за значеннями ознак. Кожен вузол у дереві рішень зображує ознаку, яка має бути класифікована, а гілки зображують значення, які розглядаються таким вузлом. Огляд існуючої роботи з побудови дерева рішень зроблено в [10], який описує використання дерев рішень у багатьох дисциплінах, таких як статистика, розпізнавання образів і машинне навчання. Одним із найпопулярніших алгоритмів дерева рішень в літературі є алгоритм C4.5 [17], покращена версія попереднього алгоритму дерева рішень ID3 [11]. C4.5 використовує стратегію «розділяй і володарюй», щоб виростити дерево, вибираючи одну функцію з принаймні двома результатами, які найбільш ефективно розподіляють набір зразків. Коли всі екземпляри в певному наборі належать до одного класу або розмір набору занадто малий, листку дерева рішень призначається мітка, що дорівнює найпопулярнішій мітці класу [12]. Дерево рішень показано на рисунку 1.4. Найвищий вузол прийняття рішень називається кореневим вузлом. Функція, яка може сортувати дані найефективніше, вибирається як кореневий вузол. Ця стратегія дублюється для кожного підрозділу навчальних даних, доки всі дані не будуть розділені

на певні пакети класів. Визначити, яка функція найефективніше сортує навчальні дані, можна здійснити за допомогою багатьох методів, таких як «Gini Index» [12], «Information Gain» або «алгоритм ReliefF». Незважаючи на те, що кожна з цих методик відрізняється від іншої, огляд алгоритмів дерева рішень, проведений в [13], показав, що не існує єдиного найкращого методу для визначення найкращої функції сортування. Однак для конкретного набору даних порівняння окремих методів може дати корисну інформацію для вибору функції сортування.



at1	at2	at3	at4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

Рисунок 1.4 –Дерево рішень для навчального набору з класами «Так» і «Ні».

Примірник [at1=a1, at2=b2, at3=a3, at4=a4] сортує до вузлів at1, at2 і at3, класифікуючи екземпляр як позитивне представлене значенням «Так»

Щоб отримати дерева потрібного розміру, в [14] запропоновано техніку «обрізання», яка використовується для зменшення складності дерев рішень шляхом видалення розділів дерева, які надають мало інформації для класифікації екземплярів. Щоб запобігти переобладнанню навчальних даних, можна використати два загальні підходи:

- зупинка алгоритму навчання до досягнення ідеальної відповідності;
- скорочення дерева рішень.

Основна перевага алгоритмів дерева рішень полягає в тому, що їх легко зрозуміти. Причину, чому конкретне дерево рішень класифікує дані до певного класу, можна легко інтерпретувати навіть людям, які не є фахівцями в цій галузі. В дослідженні було проведено порівняльне дослідження дерев рішень з іншими алгоритмами машинного навчання та дійшли висновку, що алгоритм C4.5 має дуже гарне поєднання швидкості класифікації та частоти помилок [15]. В [16] провів дослідження, де він перевіряв продуктивність алгоритму C4.5 під час виконання. На основі цього дослідження він створив покращену версію C4.5, EC4.5, яка обчислювала ідентичні дерева рішень, як і в C4.5, із збільшенням продуктивності до 500% [17].

#### 1.4.2 Штучні нейронні мережі

Нейронні мережі отримали широке визнання як ефективний алгоритм машинного навчання, перевершуючи багато алгоритмів, таких як Support Vector Machines, у різних відповідних програмах, таких як розпізнавання образів [18]. Нейронна мережа — це архітектура, яка складається з одиниць, які називаються нейронами. Ці архітектури зазвичай складаються з трьох різних рівнів: вхідний рівень, який містить вхідний вектор ознак; вихідний рівень, який складається з відповіді нейронної мережі; і шар між ними, який містить нейрони, які підключаються до входу та виходу. Приклад нейронної мережі показано на рисунку 1.5. Ця штучна нейронна мережа, яка називається прямою нейронною мережею, дозволяє сигналам лише проходити від входу до виходу.

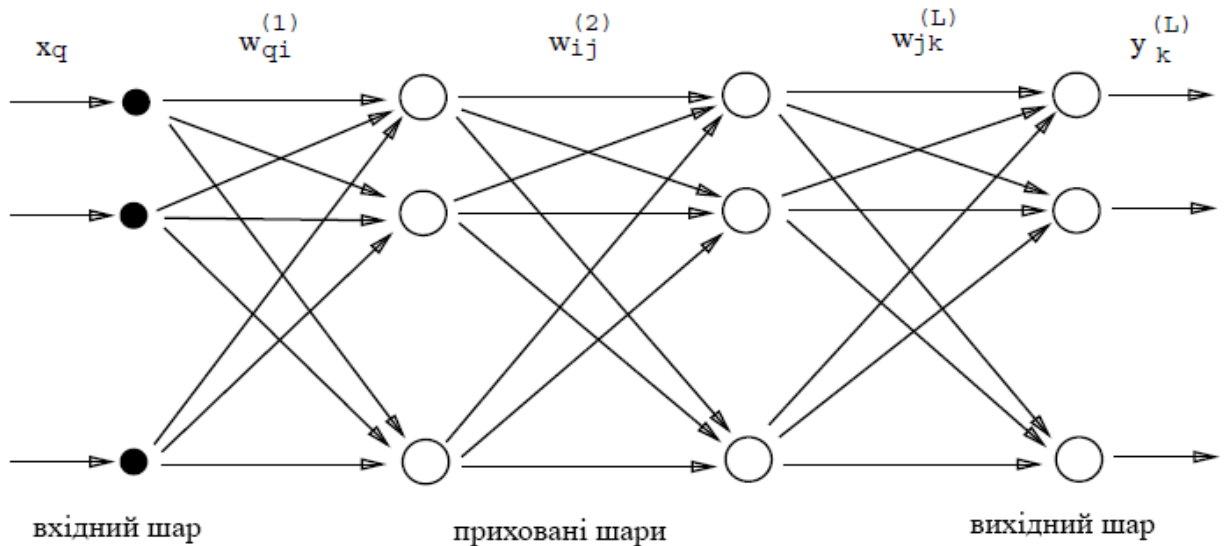


Рисунок 1.5 – Штучна нейронна мережа прямого зв'язку

Штучні нейронні мережі складаються з трьох основних характеристик: мережевої архітектури; функції введення та активації; і вага входних з'єднань. Архітектура та функції мережі вибираються на початковому етапі та залишаються незмінними під час навчання. Продуктивність нейронної мережі залежить від значення вагових коефіцієнтів. Ваги налаштовуються під час тренувань, щоб досягти певного результату. ШНМ можна навчити за допомогою безлічі різних навчальних програм [19]. Дуже відомим методом навчання є алгоритм зворотного поширення [0]. Інші методи включають алгоритм усунення ваги, який автоматично визначає топологію мережі, і генетичні алгоритми, які намагаються вивести архітектуру мережі та навчити її ваги [21][22].

Для обговорення масштабованості нейронних мереж розглядається широко використовується алгоритм зворотного поширення. Алгоритм зворотного розповсюдження, як описано в [23], включає наступні шість кроків.

1 Примірник надається як вхідний файл для мережі.

2 Вихідні дані системи співвідносяться з бажаним результатом зразка, а

помилка збільшується для кожного нейрона.

3 Обчисліть локальну помилку для кожного нейрона.

4 Зміна ваги для мінімізації локальної помилки.

5 За помилку покарання приписується нейронам попереднього рівня, де більше значення надається нейронам із сильнішою вагою.

6 Виконується повторення, коли покарання кожного нейрона використовується як його помилка.

Основною проблемою реалізації ШНМ є отримання правильного розміру прихованого шару. Якщо кількість нейронів не визначено належним чином, отримана система погано узагальнює невидимі екземпляри. З іншого боку, коли використовується забагато вузлів, може статися переобладнання, і бажаний оптимум може взагалі не бути знайденим. Виведення правильної кількості нейронів обговорюється в [24]. Основною перевагою використання штучних нейронних мереж є їх здатність обробляти дані з великими розмірними характеристиками, наприклад зображеннями. Недоліками штучних нейронних мереж є високі обчислювальні витрати, які споживають велику кількість обчислювальної потужності та використання фізичної пам'яті, а також важка зрозумілість для середніх користувачів машинного навчання [9].

### 1.4.3 Support Vector Machines

Машини опорних векторів (SVM) — це метод дискримінаційної класифікації, який походить від принципу мінімізації структурного ризику з теорії обчислювального навчання. Метою SVM є пошук найоптимальнішої функції класифікації, яка розрізняє одиниці класів у навчальних даних. З лінійно роздільним набором даних найоптимальнішу функцію класифікації можна визначити шляхом побудови гіперплощини, яка максимізує запас між двома наборами даних і таким чином створює найбільшу можливу відстань між наборами даних [25]. Візуалізація цієї стратегії проілюстрована на

рисунку 1.6.

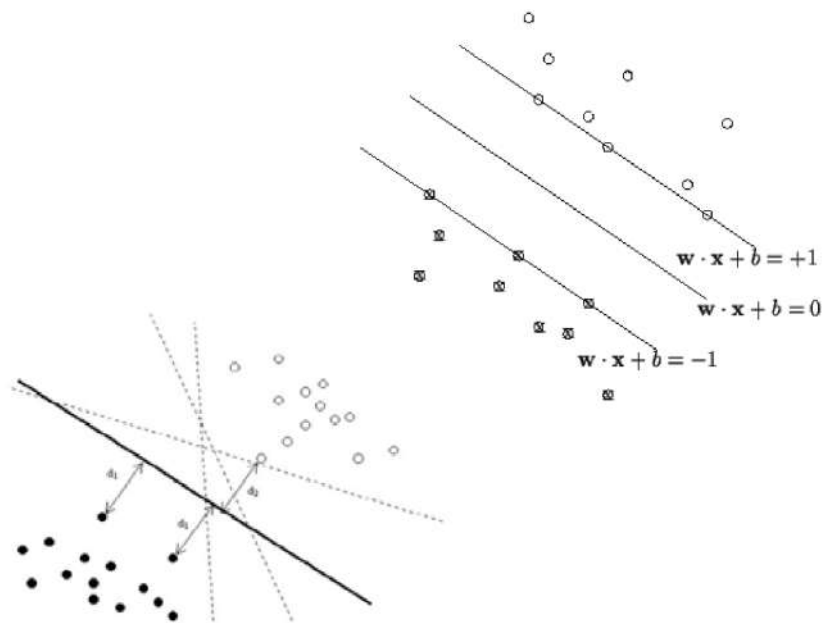


Рисунок 1.6 – Створення оптимальної розділової гіперплощини з використанням опорних векторів

Ідея, яка лежить в основі SVM, полягає в тому, що шляхом знаходження максимального запасу  $\gamma$ , отже, найоптимальнішої гіперплощини, досягається найкраща здатність узагальнення. Це призводить до найкращої продуктивності класифікації як для навчальних даних, так і для майбутніх даних.

Використання SVM надає багато переваг. Алгоритм базується на встановленій теорії, вимагає лише десятків навчальних зразків і не реагує на кількість вимірів [6]. Проте методика навчання має відносно складні алгоритми навчання та категоризації, а також високе використання пам'яті та часу під час фаз навчання та класифікації [9].

Навчання допоміжних векторних машин виконується шляхом вирішення  $N$ -вимірної задачі квадратичного програмування (QP), де  $N$  представляє кількість екземплярів навчання. Реалізація стандартних методів QP для вирішення цієї проблеми передбачає великі матричні операції та

трудомісткі чисельні обчислення. Цей метод дуже неефективний і непрактичний для великомасштабних додатків, і це добре відомий недолік методу SVM. Існують більш ефективні методи, які можуть відносно швидко вирішити проблему QR, такі як послідовна мінімальна оптимізація (SMO) [26]. Цей метод фрагментує проблему QR на підпроблеми QR і вирішує проблему квадратичного програмування SVM без використання числових кроків оптимізації QR або додаткового зберігання матриць. Новий підхід SVM знаходить оцінку найменшої навколишньої сфери групи елементів [10]. Методи SVM є двійковими, і коли потрібно вирішити багатокласову задачу, проблема поділяється на групу різних класифікаційних завдань.

## 2 ВИМОГИ ДО ОБ'ЄКТІВ

### 2.1 Правила розміщення штрих-коду

Існує багато програм, які можуть знаходити та зчитувати штрих-коди зі складних сцен за допомогою камери та використовувати їх для ідентифікації об'єктів. Різниця в освітленні, позах і спотворенні перспективи може ускладнити ідентифікацію штрих-коду, але існують алгоритми, які пропонують адаптовані рішення для цих складних сцен. Найцікавішим є випадок, коли штрих-код приховано в результаті пози продукту, і його неможливо ідентифікувати в результаті цього без зміни положення об'єкта. Складський контейнер складається з предметів, на яких видно штрих-код, а також інших предметів зі штрих-кодами, прихованих за іншими поверхнями.

Відомо, що існує певна схема розміщення штрих-коду на предметах різного розміру та форми. Більшість товарів, які обробляються на складах, відповідають інструкціям щодо розміщення символів штрих-кодів. Ці настанови містять загальні принципи, які застосовуються, обов'язкові правила та рекомендації щодо розміщення символу штрих-коду на певних типах упаковки.

Ці вказівки можуть виявитися дуже зручними для прогнозування правильного розташування штрих-коду, коли штрих-код приховано. Однак у багатьох випадках розміщення штрих-коду на товарах може порушувати правила. Хоча не всі предмети відповідають цим правилам розміщення штрих-кодів, дуже цікаво дослідити, чи існує корисний зв'язок між такими характеристиками об'єкта, як розмір або форма, і розташуванням штрих-коду. Якщо цей зв'язок існує, його можна використовувати для дуже ефективного вибору та сканування товарів на складі, усуваючи потребу в ідентифікації об'єкта за допомогою людини. Машинне навчання буде впроваджено для отримання цих моделей розміщення штрих-кодів на

об'єктах різних форм і розмірів. Аналізуючи форму, текстуру та набору інших можливих функцій пропонується нова структура розпізнавання об'єктів, яка повинна надійно визначати розташування штрих-коду об'єктів.

В ході досліджень було визначено структуру повної системи розпізнавання об'єктів. Було вирішено, що систему буде створено з використанням згорткової нейронної мережі (ЗНМ). Тоді для цього методу локалізації штрих-коду потрібно побудувати схему розміщення штрих-коду на продуктах за допомогою згорткової нейронної мережі, щоб визначити, на якій поверхні товару лежить штрих-код.

Щоб автоматизувати фазу повторного генерування системи виконання замовлень, нещодавно розроблена структура розпізнавання об'єктів має задовольнити двом важливим вимогам.

1 Система повинна досягати 40% точності передбачення локалізації штрих-коду: оскільки системи виконання замовлень обробляють величезну кількість продуктів, нові системи бачення повинні прагнути до принаймні 40% точності передбачення. У подальшому можна покращити, додавши більше даних для навчання, або запровадивши більш оптимальні методи навчання.

2 Система повинна бути масштабованою до дуже великих наборів продуктів: спроектована система бачення повинна мати можливість обробляти поточні набори продуктів, які складаються в порядку 10<sup>7</sup> різних продуктів.

## 2.2 Визначення параметрів

Щоб побудувати структуру локалізації штрих-коду, спочатку визначаються вхід, вихід і визначення класу.

Складські підприємства розробляють рішення автоматизації для різних ринків: посилок, складів і аеропортів, що означає, що об'єкти, з якими ці рішення обробляють, зазвичай мають велику різноманітність за формою,

кольором, вагою та іншими фізичними властивостями. Важливо розуміти, що в цьому проєкті акцент робиться на складському сегменті. Крім того, продукти вибираються зі стандартизованих категорій, оскільки це дозволяє уникнути неоднозначності щодо визначення та тлумачення класу. Сфера дії цього дослідження обмежена елементами категорії "Коробки", для яких будується доказ концепції. Після створення робочої структури до неї можна додати інші категорії продуктів.

Створений набір даних складається із зображень елемента категорії коробки, взятих із вигляду зверху та збоку. Приклади цих зображень з окремими предметами в різних позах можна знайти на рисунках 2.1 та 2.2



Рисунок 2.1 – Коробка пластівців із видимим штрих-кодом у верхній частині



Рисунок 2.2 – Коробка печива з прихованим штрих-кодом.

І для коробки пластівців, і для коробки з печивом система локалізації штрих-коду повинна мати можливість визначати розташування штрих-коду.

Зображення будуть імпортовані у застосунок як дані RGB з роздільною здатністю 400x300. Таким чином, вхідні дані складаються з матриці дизайну  $X \in \mathbb{R}^{N \times 400 \times 300 \times 3}$ , де «N» представляє кількість зразків у наборі даних.

Оскільки згорткові нейронні мережі є типом контрольованого навчання, для етапу навчання `rfcnjseure` потрібні дані з мітками. Для ефективного маркування всіх даних вибрано наступне визначення класу:

$$c_n = \{1, 2, 3, 4, 5, 6\}$$

У цьому визначенні класу кожне з чисел представляє одну з шести граней об'єкта розміром з коробку. Це визначення класу завжди застосовується з точки зору камери, рисунок 2.3. Для всіх випадків, грань яка знаходиться найближче до камери, визначається як верхнє грань ( $c=1$ ), звідти визначаються інші обличчя. Грань, протилежна верхній грані, є нижньою ( $c=3$ ). Гранями, що прилягають до верхньої та нижньої граней, є передня ( $c=2$ ), задня ( $c=4$ ), права ( $c=5$ ) і ліва грані ( $c=6$ ). Враховуючи перспективу камери, визначення тлумачиться та ілюструється таким чином:

$$c_n = \{1, 2, 3, 4, 5, 6\} == \{\text{верх, спереду, знизу, ззаду, справа, зліва}\}$$

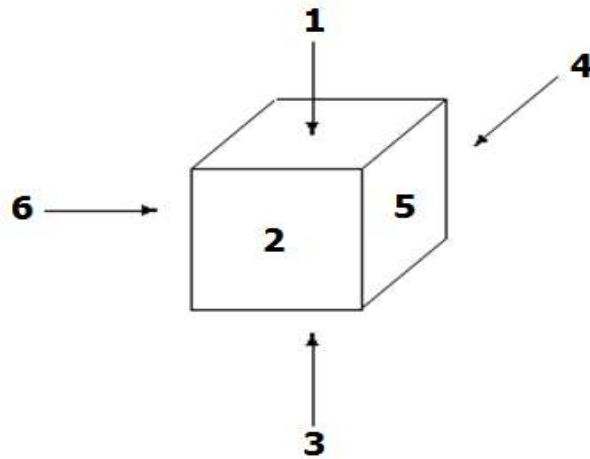


Рисунок 2.3 – Пропонована структура розпізнавання об’єктів із визначенням класу

Вихідні дані застосунку складаються з категоричного вектора розподілу ймовірностей, який визначає ймовірність наявності штрих-коду на певній грані. Можливий результат можна визначити за допомогою:

$$c_n = \{1, 2, 3, 4, 5, 6\} == \{0.8, 0.1, 0.05, 0.02, 0.02, 0.01\}$$

Цей результат означає, що з імовірністю 80% штрих-код буде розміщено на верхній стороні, 10% – на лицьовій стороні тощо. Повний каркас зображено на рисунку 2.4.



Рисунок 2.4 – Визначення класу для об’єкта у формі коробки

## 3 РЕАЛІЗАЦІЯ СИСТЕМИ КЛАСИФІКАЦІЇ ЗА ДОПОМОГОЮ ЗНМ

### 3.1 Згорткова нейронна мережа

Згорткові нейронні мережі наразі є одним із найвідоміших алгоритмів глибокого навчання із зображеннями. У той час як для традиційного машинного навчання відповідні функції потрібно витягувати вручну, глибоке навчання використовує необроблені зображення як вхідні дані для вивчення певних функцій. ЗНМ складаються з вхідного та вихідного рівня та кількох прихованих рівнів між входом і виходом. Прикладами проміжних шарів є згорткові шари, шари максимального об'єднання та повністю зв'язані шари [6].

Архітектури ЗНМ відрізняються кількістю та типом шарів, реалізованих для конкретного застосування. Для безперервних відповідей мережа повинна включати рівень регресії в кінці мережі, тоді як для категоричних відповідей система повинна включати функцію класифікації та рівень. Нейрони в кожному шарі ЗНМ розташовані в тривимірному порядку та перетворюють тривимірний вихід із тривимірного вхідного. Для нашої конкретної програми вхідний рівень містить зображення як 3D-вхідні дані зі значеннями висоти, ширини та RGB як розмірів. Далі на згортковому шарі нейрони прикріплюються до областей зображення та перетворюються на тривимірний результат

Конфігурації ЗНМ складаються з безлічі прихованих шарів. У кожному шарі обсяги активації змінюються за допомогою диференційованих функцій. Існують чотири основні типи шарів, які використовуються для побудови конфігурацій ЗНМ.

1 Згортковий рівень – згорткові фільтри використовуються для отримання карти активації з вхідних даних.

2 Rectified Linear Unit Layer (ReLU) – фільтрує від'ємні значення, щоб

отримати лише додатні значення для набагато швидшого часу навчання.

3 Рівень об'єднання – виконує нелінійну низьку дискретизацію та скорочує кількість параметрів для простішого виведення.

4 Повністю зв'язаний рівень – обчислює оцінки ймовірності класу шляхом виведення вектора розмірів  $C$ , де  $C$  є кількістю класів. Усі нейрони підключені до цього шару.

Для задач розпізнавання в умовах використання великих об'ємів даних необхідно, щоб модель мала високу здатність до навчання та великий відсоток правильних припущень щодо ознак зображення. У порівнянні з традиційними нейронними мережами прямого поширення зі схожою кількістю шарів, згорткові нейронні мережі мають вищу здатність до навчання, оскільки містять набагато менше параметрів та зв'язків. Для традиційних повнозв'язних нейронних мереж для їх правильного навчання в задачах розпізнавання образів необхідна набагато більша кількість даних, оскільки кожне вхідне зображення може мати розмірність у щонайменше кілька сотень тисяч, що вимагатиме відповідну кількість прикладів з різними значеннями для кожного виміру.

Ще однією відмінністю згорткових нейронних мереж є те, що вони зазвичай не вимагають попередньої обробки даних, оскільки використовують дані зображень напряму, що дозволяє спрощувати структуру мережі, зважаючи на упорядкованість початкових даних. Водночас, незважаючи на численні переваги, їх застосування все ще вимагає значної витрати ресурсів, особливо для розв'язання задач із зображеннями з високою роздільною здатністю, тому виникає необхідність розробки згорткових нейронних мереж із високим рівнем оптимізації операцій.

Окрім застосування для задач розпізнавання та задач класифікації зображень, згорткові нейронні мережі завдяки модифікаціям широко застосовуються до задач обробки природної мови (ОПМ, англ. Natural language processing, NLP), зокрема, вони показали високу ефективність в задачах семантичного парсингу, а також моделюванні та класифікації речень.

Структура згорткової нейронної мережі відповідає загальноприйнятій структурі нейронної мережі. Мережа складається з вхідного шару, певної кількості прихованих шарів та вихідного шару. Приховані шари зазвичай складаються зі згорткових шарів, шарів агрегування (субдискретизації), нормалізуючих та повнозв'язних шарів. Ці шари пов'язані між собою шарами з визначеними активаційними функціями. Головним елементом згорткової нейронної мережі є згорткові шари, де до даних з попереднього шару застосовується операція згортки.

Згорткові нейронні мережі здобули широку популярність в задачах розпізнавання зображень завдяки тому, що вони уникають або зменшують головні проблеми, пов'язані з обробкою зображень в традиційних повнозв'язних мережах. Велика кількість зв'язків в таких мережах та велика кількість параметрів у кожному зображенні потребує великої кількості вхідних прикладів, а у випадку їх нестачі швидко виникає перенавчання мережі, тобто, модель стає занадто складною та навчається на нехарактеристичних другорядних ознаках, знайдених у вхідних даних. Ще більшою проблемою є те, такі моделі не є стійкими до збурень або будь-яких змін у зображеннях, наприклад, до розташування об'єкта в іншій частині зображення або зміні кута погляду на нього. Крім того, такі мережі не здатні до врахування топології вхідних даних: зображення є структурованими даними, де пікселі, що розташовані близько один від одного, мають високу кореляцію, що робить необхідним знаходження локальних ознак та взаємозв'язків для ефективного визначення образів.

В згорткових нейронних мережах ці проблеми вирішуються завдяки використанню кількох ідей, найголовнішою з яких є локальна обробка значень на прихованих рівнях. Кожен з рівнів поділений на частини, кожна з яких сприймає лише дані попереднього рівня, розташовані в певній невеликій області. Такі області називаються локальними рецептивними полями. Використовуючи локальні рецептивні поля, мережа може визначати найпростіші елементи та ознаки зображення, як повороти або грані між

ділянками зображення. Іншою ідеєю в основі згорткових нейронних мереж є спільні ваги, використання яких зменшує чутливість мережі до змін положення об'єктів на зображенні, незначних поворотів та його спотворення: так, детектор певної простої ознаки зображення, що використовується на певній його ділянці, може визначати аналогічні ознаки і на іншій його ділянці. Завдяки цьому детектори для локальних рецептивних полів на різних ділянках будуть мати однакові ваги, і, як наслідок, в різних частинах зображення для схожих ознак виконуються однакові перетворення.

Крім того, важливим елементом є операція субдискретизації, яка дозволяє зменшувати загальну розмірність даних, при цьому зменшуючи їх чутливість до значних збурень. Зазвичай субдискретизація в архітектурі мережі чергується із операціями згортки, таким чином поступово зменшуючи розмірність даних та збільшуючи кількість карт ознак.

Розглянуто згорткові нейронні мережі, один з підвидів нейронних мереж, який показав себе ефективним в задачах класифікації та розпізнавання зображень, але також може використовуватися і для інших цілей. Основними їх елементами є згорткові рівні з операцією згортки, які розпізнають ознаки на зображення, агрегувальні рівні з операціями, які узагальнюють зображення та забезпечують покращену стійкість моделі, повнозв'язні рівні для обробки результатів, отриманих на згорткових та агрегувальних, рівні втрат, що забезпечують оновлення параметрів моделі при навчанні та покращення її роботи, та функції активації, які забезпечують нелінійність системи та роблять можливими багатошарову побудову мереж, а також розпізнавання складних характеристик.

Згорткові нейронні мережі навчаються завдяки градієнтним алгоритмам, що змінюють параметри моделей в напрямку зменшення помилки. Для задач класифікації, які розглядаються в цій роботі, результатом роботи згорткової нейронної мережі є імовірності належності вхідного зображення до кожного з вибраних класів.

### 3.2 Створення ЗНМ в Matlab

Структура, описана в цій роботі, розроблена в Matlab. Сам застосунок призначений для роботи на настільному комп'ютері. Використовуються два набори інструментів Matlab: Parallel Computing і Neural Network Toolboxes. Ці набори інструментів забезпечують алгоритми згорткового навчання, попередньо запрограмовані рівні та згорткові мережі, а також можливості паралельного обчислення для значного скорочення часу обчислення.

Програмний продукт MATLAB полегшує процедуру конструювання, навчання та використання ШНМ. Користувачу не потрібно досконало розуміти механізм створення нейромережі, варто лише керуватися запитами програми. Проте не можна починати роботу, не будучи обізнаним з функціонуванням нейромережі [4]. Перед початком роботи в середовищі будь-якої програми потрібно:

- визначити загальний тип виконуваного завдання і відповідно до нього обрати структуру або архітектуру нейронної мережі; при цьому потрібно зупинитися на найефективнішій і перевірити очікуваний результат.

- підготувати набір навчальних і тестових даних; перевірити, щоб приклади навчальної множини не повторювалися в тестовій, оскільки мережа просто запам'ятає результат і відтворить його.

- вибрати кількість вхідних та вихідних нейронів, прихованих шарів; для цього потрібно чітко визначити кількість вхідних параметрів – це і буде кількість вхідних нейронів, а вихідні параметри відповідно визначать кількість вихідних нейронів. Кількість прихованих шарів нейронної мережі можна скоригувати і в процесі роботи саме тут потрібно провести ряд експериментів, щоб визначити найоптимальніший варіант [27].

Програмний пакет MATLAB від компанії розробника MathWorks – один з найпотужніших інструментів оброблення даних на ринку програмних продуктів. Штучні нейронні мережі в середовищі MATLAB можна проектувати як за допомогою спеціального вбудованого пакета NNTool, так і

безпосередньо в командному вікні. Попри всі переваги програми, варто відзначити й недоліки – велику ціну програмного пакета та складність процесу функціонування.

MATLAB — чудовий інструмент для дослідження, аналізу та візуалізації даних. Він поєднує багато високоякісних інструментів і функцій, важливих для наукових досліджень. Він часто використовується дослідниками для проектування простих і складних систем і їх моделювання для аналізу ефективності моделі. Крім того, MATLAB має спеціальну функцію, яка називається набором інструментів глибокого навчання (DL), завдяки якій розробка та навчання глибоких нейронних мереж настільки прості, що дослідники з більшою ймовірністю використовують MATLAB замість python. Іншою причиною може бути те, що виконання операцій DL в MATLAB робить всю систему компактною. Раніше навчання нейронної мережі та симуляція системи проводилися у двох різних сегментах.

При визначенні конфігурації мережевого рівня важливо зазначити, що не існує точної формули для оптимальної конфігурації рівня. Замість цього найкращий підхід – це метод проб і помилок, коли кілька конфігурацій шарів досліджуються та порівнюються, щоб побачити, наскільки добре вони працюють. Попередньо навчені мережі, можна використовувати як початкову конфігурацію рівня. Однак у цьому випадку, переслідуються власні реалізації конфігурації шару.

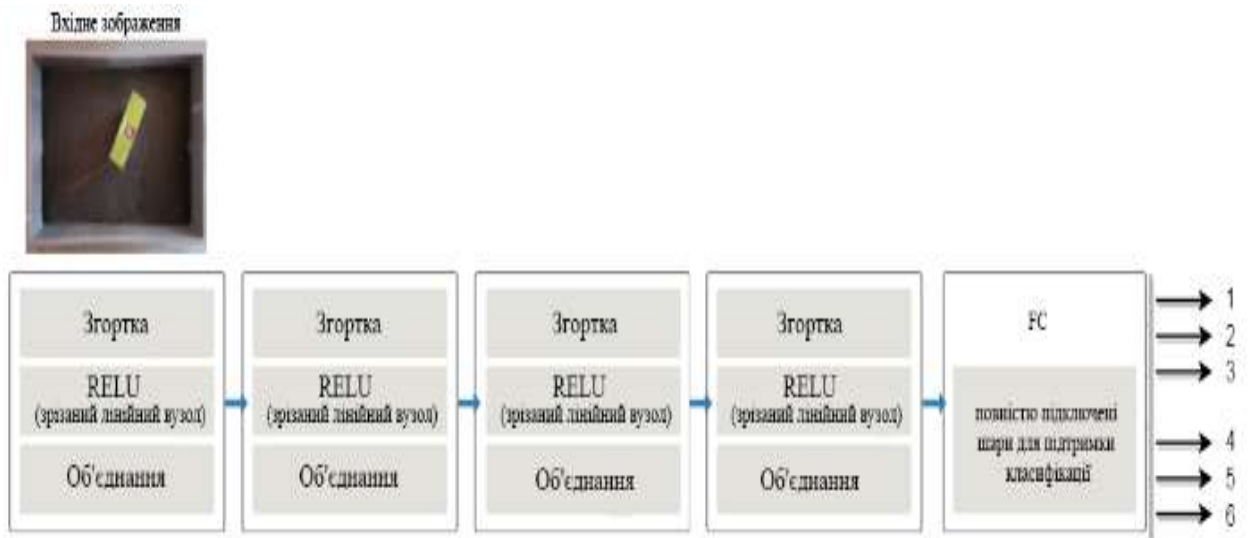


Рисунок 3.1 – Навчання глибокої згорткової мережі, зображеної

Навчання глибокої згорткової мережі, зображеної на рисунку 3.1, може зайняти різну кількість часу, залежно від розміру набору даних і доступної потужності обробки. Для навчання ЗНМ доступні три варіанти обчислень, і вибір найбільш оптимального є вирішальним залежно від кількості доступного часу для вирішення конкретного завдання:

- 1 обчислення на основі CPU;
- 2 обчислення на основі GPU;
- 3 хмарні обчислення GPU.

Варіанти обчислень на основі CPU є найбільш простими та доступними. Однак оскільки центральний процесор обчислює завдання в послідовній конфігурації, навчання мережі займає найдовше часу за допомогою цього методу. Використання графічного процесора значно скорочує час навчання, і це можна зробити за допомогою Matlab без додаткового програмування. Однак для паралельних обчислень необхідний графічний процесор NVidia на основі CUDA, щонайменше 3.0. Існує також можливість використовувати кілька графічних процесорів, що ще більше зменшує час обробки. Нарешті, хмарні обчислення GPU розглядають використання хмарних ресурсів для обчислювальної потужності. Написаний код MATLAB можна покращити для цілей хмарних обчислень. У рамках цієї

роботи обчислення, як на CPU, так і на GPU розглядаються для дослідження впливу використання GPU на зменшення часу обчислення.

### Лістинг 3.1 – Навчання ЗНМ

```

1 %% Loading and Exploring Image Data
2
3 Datasetpath = fullfile ( 'C:\Users\EliteBook\Desktop\Labeled Data800x600' );
4 productData = imageDatastore ( Datasetpath , . . .
5 'IncludeSubfolders' , true , 'LabelSource' , 'foldernames' );
6
7 %% Image and Label Count
8
9 labelCount = countEachLabel ( productData )
10
11 %% Defining Image Size
12
13 img = readimage ( productData , 1 );
14 size ( img )
15
16 %% Specify Training and Validation Sets
17
18 trainNumFiles = 56; % 65 = 86/14% Train/Val Split , 56 = 75%/25%
19 [ trainDigitData , valDigitData ] = splitEachLabel ( productData ,
trainNumFiles , 'randomize' ) ;
20
21 % %% Define Network Architecture
22 % % Define the convolutional neural network architecture.
23 layers = [
24 imageInputLayer ( [ 1 0 8 144 3 ] )
25
26 convolution2dLayer ( 7 , 3 2 , 'Padding' , 1 )
27 batchNormalizationLayer
28 reluLayer
29
30 maxPooling2dLayer ( 2 , 'Stride' , 2 )
31 %%%
32 convolution2dLayer ( 5 , 1 6 , 'Padding' , 1 )
33 batchNormalizationLayer
34 reluLayer
35
36 maxPooling2dLayer ( 2 , 'Stride' , 2 )

```

### 3.3 Навчання ЗНМ

Під час імпорту позначеного набору даних 75%/25% розподіляються між даними навчання та даними перевірки. Це означає, що 75% даних використовується для навчання мережі, а 25% даних використовується для перевірки мережі. Після досягнення достатньої точності перевірки можна

використати додатковий набір тестів, щоб побачити, наскільки добре працює мережа.

Перша конфігурація виконується на основі обчислювальної моделі ЦП. Використовується процесор Intel Core i7 720QM з чотирма фізичними та чотирма віртуальними ядрами. Хід навчання та результат показані на рисунку 3.2.

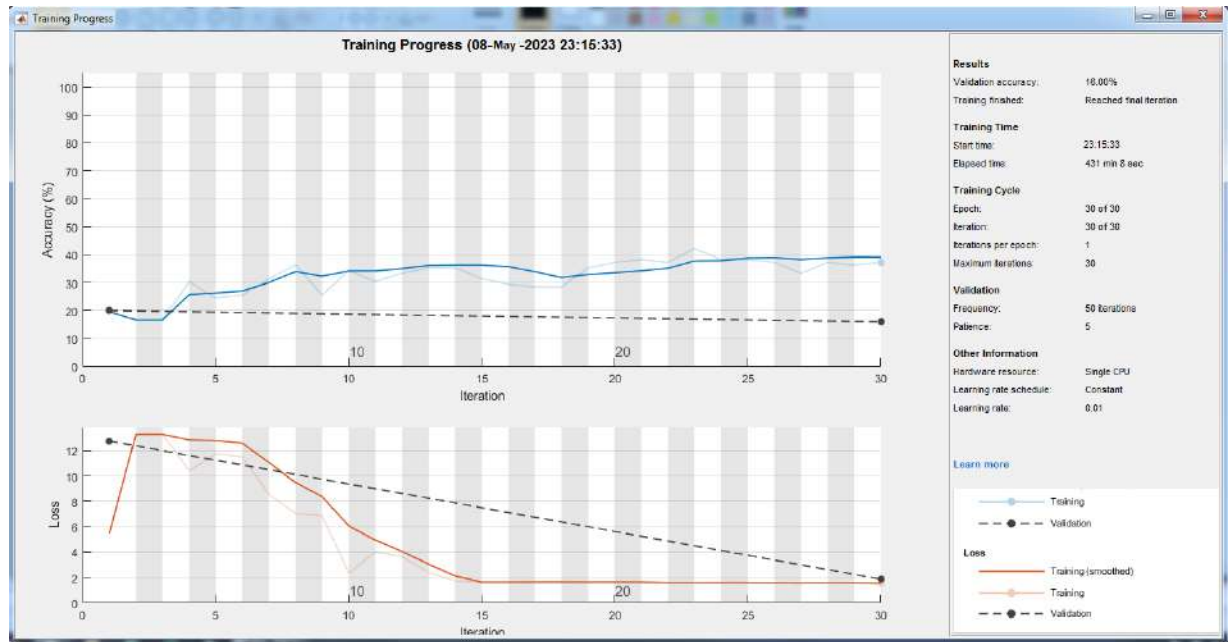


Рисунок 3.2 ЗНМ навчальний запуск 1, з використанням центрального процесора. Точність перевірки 16% досягається за час навчання приблизно 431 хвилину

Незважаючи на те, що використовуваний CPU має чотири фізичні ядра, за часом, що минув, можна відразу помітити, що навчання з CPU все одно займає надзвичайно багато часу, і його неможливо точно налаштувати в рамках цього проекту. Навіть якщо центральний процесор складається з кількох ядер і є найкращим у лінійці, велика тривалість обробки залишається проблемою. Дивлячись на результати тесту, точність перевірки в 16% є дуже недостатньою, і схоже, що ЗНМ не дізнався жодних цінних функцій, які можна було б використати для правильної класифікації.

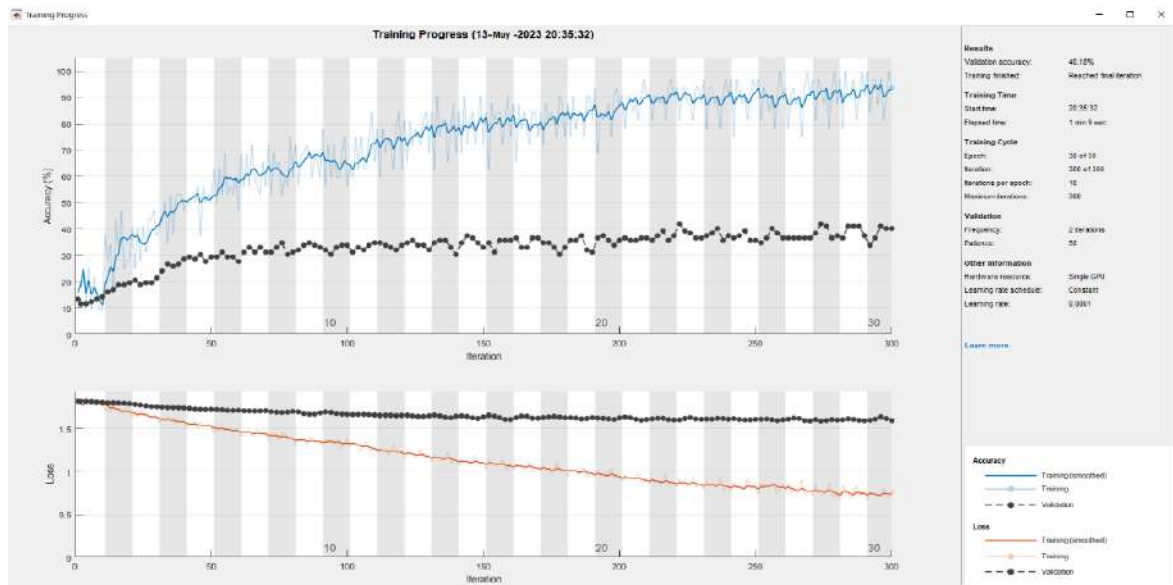


Рисунок 3.3 – Тренування ЗНМ 2 з використанням GPU. Параметри системи вже дещо налаштовані. За рахунок зменшення розміру фільтра та кількості фільтрів точність перевірки 40,18% досягається з часом навчання 69 секунд

Змінюючи кількість функцій, які обчислюють згорткові шари, і розмір фільтра, який використовує згортковий шар, система додатково налаштована для підвищення точності перевірки. Однак з другого запуску на основі GPU беруться обчислення. Для наступних прогонів використовується графічний процесор NVidia GTX1060 і процесор Intel i7-6700HQ. Результати другого запуску показані на рисунку 3.3.

Що відразу можна відзначити, так це те, що час обчислень зменшився з 431 хвилини до приблизно 1 хвилини. Це означає, що час обчислення зменшився приблизно в 400 разів. Це принаймні дозволяє ефективно налаштувати ЗНМ. Дивлячись на результати перевірки, досягнуто точності 40,16%. У той час як точність навчального набору збільшується приблизно до 100%, точність перевірки продовжує коливатися близько 40%. Порівнюючи втрату під час навчання з втратою підтвердження, можна побачити, що хоча навчальний набір справді збігається до нуля, набір підтвердження залишається приблизно 1,75. Подальше тонке налаштування виконується для ЗНМ, щоб дослідити, чи може зміна швидкості навчання, розміру мінібатча

чи епохи, або конфігурацій і налаштувань шару ще більше покращити ЗНМ. Це робиться кілька разів поспіль, показуючи деякі з цікавих результатів на рисунках 3.4-3.6

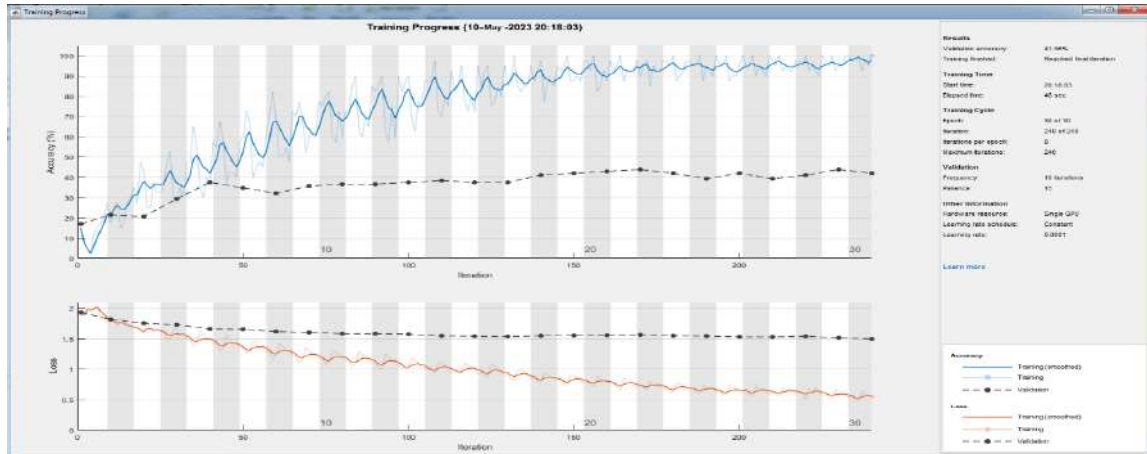


Рисунок 3.4 – Навчальний запуск ЗНМ (з GPU) зі зміненими темпами навчання та розмірами фільтрів шарів. Точність перевірки 41,96% досягається за час навчання 48 секунд

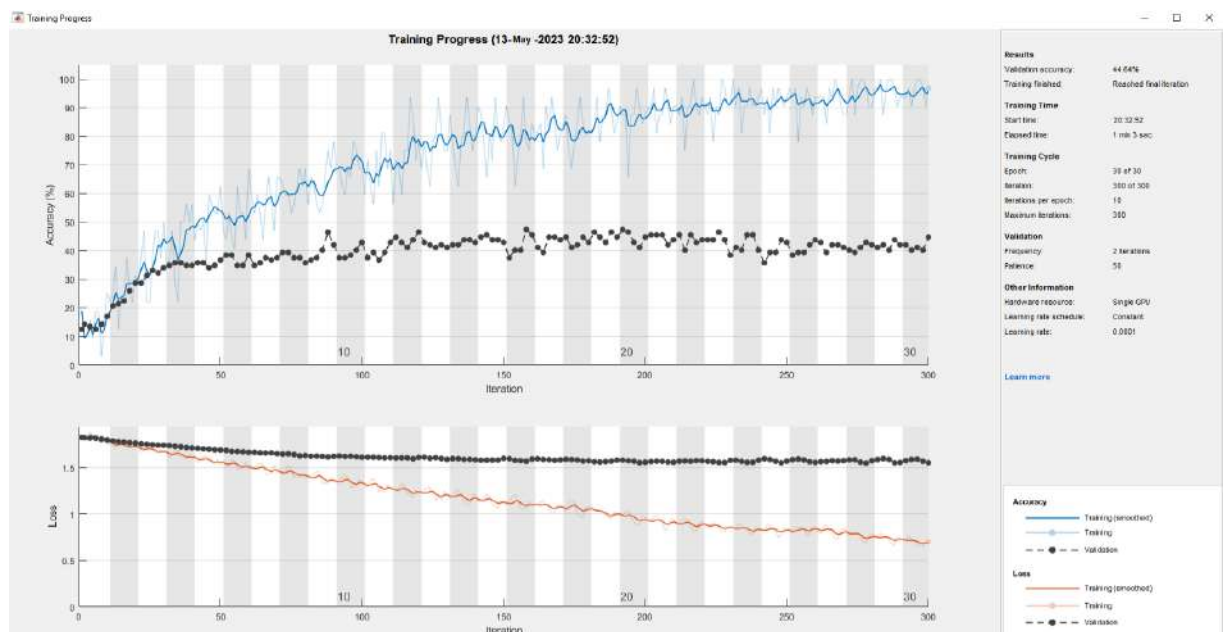


Рисунок 3.5 – Навчальний запуск ЗНМ (з GPU) зі зміненими темпами навчання та розмірами фільтрів шарів. Точність перевірки 44,64% досягається за час навчання 63 секунди

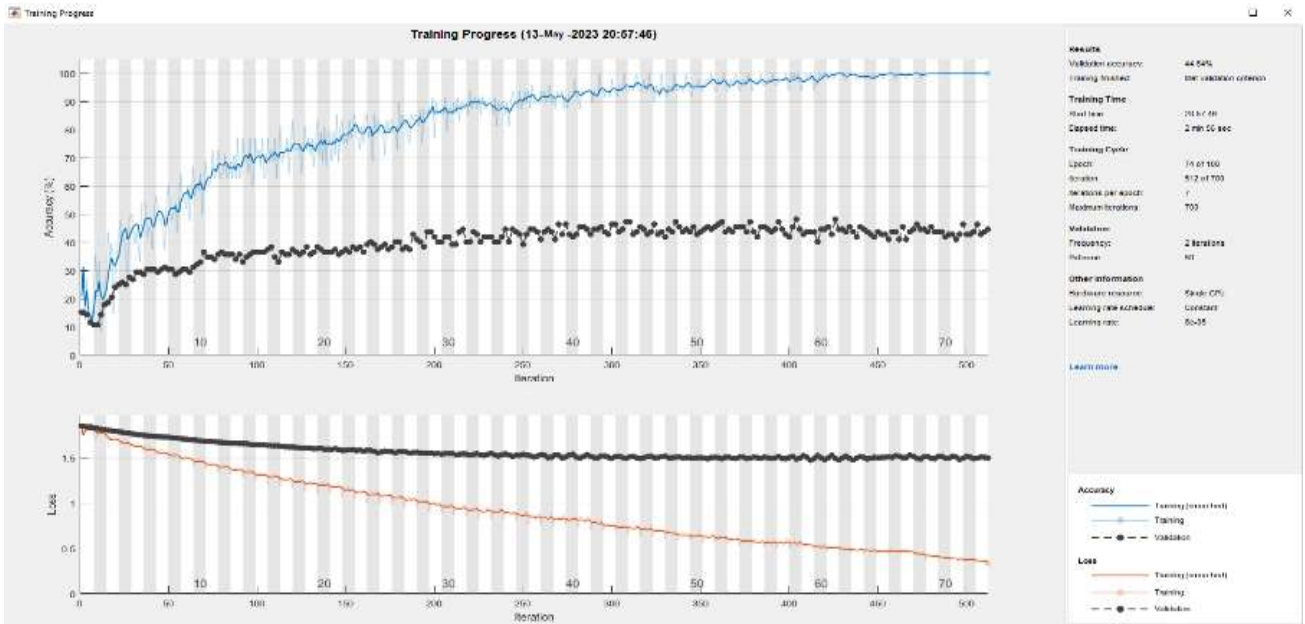


Рисунок 3.6 – Навчальний запуск ЗНМ (з GPU) зі зміненими темпами навчання та розмірами фільтрів шарів. Точність перевірки 44,64% досягається за час навчання 2 хвилини 56 секунд

Після проведення кількох десятків тестів було зроблено висновок, що точність перевірки набору даних становить приблизно 45% після точного налаштування системи. Точність навчального набору зростає до 100%, але набір перевірки продовжує коліватися на рівні 45%. З цих результатів можна зробити висновок, що з поточним розміром набору даних близько 1000 зображень ЗНМ не може достатньо розрізняти екземпляри даних. Ці результати є ідеальною ілюстрацією випадку, коли мережа не узагальнює достатньо добре, щоб не побачити екземпляри даних. ЗНМ переповнив дані, як видно зі 100% точності на навчальному наборі, а налаштування швидкості навчання та розмірів пакетів, здається, не мають жодного додаткового ефекту після точного налаштування, щоб досягти точності близько 45%. Висновок, який можна зробити, такий: для навчання ЗНМ потрібен більший набір даних, щоб зменшити переобладнання та досягти вищої точності.

## ВИСНОВКИ

Розглядаючи запропоновану структуру розпізнавання об'єктів у світлі визначених вимог і обмежень, можна зробити висновок, що досягнуто достатньої точності 44,64%, що перевищує вимогу 40%. Логічний висновок, який можна зробити з результатів навчання, полягає в тому, що набір даних має бути різко збільшений, щоб запобігти переобладнанню даних. Крім того, набори даних усіх інших категорій продуктів повинні бути додані до ЗНМ, щоб структура була реалізована на складі. Хоча генерація та збір цих даних може зайняти багато часу, з результатів, досягнутих на сьогоднішній день, можна побачити, що навіть з невеликими наборами даних показані багатообіцяючі результати.

Під час навчання ЗНМ необхідна реалізація GPU. Навчання ЗНМ лише за допомогою процесора може зайняти місяці для повного набору даних категорії продуктів. Логічним є висновок, що тонке налаштування системи під час цього процесу неможливе в цьому часовому режимі. Масштабованість, здається, не є проблемою при реалізації обчислень GPU, і, крім того, можна реалізувати хмарні обчислення, щоб ще більше скоротити час навчання під час навчання дуже великих наборів даних.

Коли досягається достатня точність класифікації ЗНМ, можна застосувати розумні стратегії комплектування, коли робот-комплектатор намагається класифікувати продукти, показуючи грані продукту вздовж вектора ймовірності класифікації. Цей метод забезпечує негайний вибір, коли продукт класифіковано правильно, і вторинний вибір у разі неправильної класифікації, коли робот намагається сканувати обличчя відповідно до вектора ймовірності класифікації, доки не буде знайдено штрих-код. Це гарантує мінімальні або повну відсутність витрат на помилки, оскільки продукт завжди сканується. У цьому випадку точність класифікації ЗНМ корелює з тим, наскільки швидко сканер штрих-кодів класифікує продукти з

першої спроби, де підвищення точності означає швидший вибір.

По результатам досліджень було написано публікація “Система розпізнавання об'єктів для автоматизації складів з використанням згорткової нейронної мережі ” та подано до фахового видання України категорії Б – Системи управління навігації та зв'язку

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kuchuk, H., Kovalenko, A., Ibrahim, B.F. and Ruban, I. (2019), "Adaptive compression method for video information", *Int. Journal of Advanced Trends in Computer Science and Engineering*, 8(1), pp. 66-69, DOI: <http://dx.doi.org/10.30534/ijatcse/2019/1181.22019>
2. Bevzyuk Roman Система розпізнавання об'єктів для автоматизації складів з використанням згорткової нейронної мережі / Roman Bevzyuk, Oleksii Lyashenko // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2023. – Т. 2 (72). – С. 68-74. – doi:<https://doi.org/10.26906/SUNZ.2023.2.068>.
3. R. Bekkerman, M. Bilenko, J. Langford, *Scaling Up Machine Learning*, Cambridge University Press, January 2012
4. Mitchell, T. (1997). *Machine Learning*. McGraw Hill.
5. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge:MIT Press
6. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2017), "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, Vol. 60, No. 6
7. Alexander Toshev, Christian Szegedy (2014), "DeepPose: Human Pose Estimation via Deep Neural Networks", DOI: <https://doi.org/10.1109/CVPR.2014.214>
8. I. Mykhailichenko, H. Ivashchenko, O. Barkovska and O. Liashenko, "Application of Deep Neural Network for Real-Time Voice Command Recognition," 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 2022, pp. 1-4, doi: 10.1109/KhPIWeek57572.2022.9916473.