

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

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

Розпізнавання об'єктів заданого типу на зображеннях із використанням
конволюційної нейронної мережі
(тема)

Виконав:
студент 2 курсу, групи СШМ-18-2
Косолапов К.С.
(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва спеціальності)

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

Освітня програма Системи штучного
інтелекту (СШ)
(повна назва спеціалізації)

Керівник проф. Рябова Н.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 – Комп'ютерні науки _____

(код і повна назва)

Тип програми _____ освітньо-наукова _____

(освітньо-професійна або освітньо -наукова)

Освітня програма _____ Системи штучного інтелекту (СШІ) _____

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Косолапову Костянтину Сергійовичу _____

(прізвище, ім'я, по батькові)

1. Тема роботи Розпізнавання об'єктів заданого типу на зображеннях із використанням конволюційної нейронної мережі _____

затверджена наказом по університету від 30 березня 2020 р. № 480 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 20 травня 2020 р.

3. Вихідні дані до роботи науково-технічні публікації та дані Інтернет-джерел щодо тематики атестаційної роботи; програма Matlab R2020a _____

4. Перелік питань, що потрібно опрацювати в роботі

аналіз сучасного стану та перспектив розвитку глибинного навчання, загальна характеристика основних видів глибинних нейронних мереж, постановка задач дослідження, аналіз основних видів задач розпізнавання об'єктів та методів їх вирішення, обґрунтування вибору архітектури конволюційної нейронної мережі, алгоритм навчання конволюційної нейронної мережі, аналіз отриманих результатів, практичне застосування отриманих результатів дослідження _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів,

комп'ютерних ілюстрацій (слайдів)

графічне зображення dropout _____

архітектура мережі Хопфілда _____

структура LTSM-мережі _____

архітектура Fast R-CNN _____

базова архітектура конволюційної нейронної мережі AlexNet _____

приклад узагальненого представлення R-CNN _____

тренувальний процес CNN _____

головне вікно програмної системи _____

результат класифікації об'єкта _____

результат пошуку об'єкта на зображенні _____

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

Найменування Розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Рябова Н.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	30.03.2020	Виконано
2	Аналіз предметної області та постановка задачі	31.03.2020-07.04.2020	Виконано
3	Дослідження методів розпізнавання зображень	08.04.2020-15.04.2020	Виконано
4	Розробка та тестування архітектур CNN	16.04.2020-26.04.2020	Виконано
5	Розробка базового прототипу	27.04.2020-04.05.2020	Виконано
6	Підготовка пояснювальної записки та презентації	05.05.2020-12.05.2020	Виконано
7	Надання пояснювальної записки на перевірку	13.05.2020-20.05.2020	Виконано
8	Захист атестаційної роботи	21.05.2020	Виконано

Дата видачі завдання 30.03.2020 р.

Студент _____
(підпис)

Керівник роботи _____ — проф. Рябова Н.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 75 с., 22 рис., 2 дод., 44 джерела.

ГЛИБИННЕ НАВЧАННЯ, ГЛИБИННІ НЕЙРОННІ МЕРЕЖІ, КОНВОЛЮЦІЙНА НЕЙРОННА МЕРЕЖА, МЕТОДИ КЛАСИФІКАЦІЇ, ЗОБРАЖЕННЯ

Об'єкт дослідження - глибинні нейронні мережі.

Предмет дослідження - розпізнавання об'єктів заданого типу на зображеннях із використанням конволюційної нейронної мережі.

Мета роботи – дослідження властивостей глибинних нейронних мереж, їх основних переваг та особливостей архітектури, предметних галузей та задач, які доцільно вирішувати за їх допомогою, на прикладі вирішення задачі розпізнавання об'єктів заданого типу на зображеннях із використанням конволюційної нейронної мережі, аналізу її основних можливостей та поведінки у ході вирішення поставленої задачі, розробка програмного додатку розпізнавання об'єктів заданого типу на зображеннях.

Методи дослідження - методи глибинного навчання, методи класифікації об'єктів певного типу, моделювання процесів класифікації об'єктів з використанням конволюційних нейронних мереж.

В результаті проведених досліджень вирішено задачу розпізнавання об'єктів заданого типу на зображеннях на підставі методів глибинного навчання. Отримані результати використовуються у розробці додатку для розпізнавання певного типу об'єктів. В якості програмної платформи використано пакет прикладних програм та мову програмування Matlab.

Пропонована розробка буде корисною для застосування в різних галузях, пов'язаних із використанням фруктів та овочів, наприклад, у фермерських господарствах, сільському господарстві в цілому, кулінарії, ресторанному бізнесі, у продуктових супермаркетах тощо.

РЕФЕРАТ

Записка поясните: 75 с., 22 рис., 2 прил., 44 источника.

ГЛУБОКОЕ ОБУЧЕНИЕ, ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ, КОНВОЛЮЦИОННАЯ НЕЙРОННАЯ СЕТЬ, МЕТОДЫ КЛАССИФИКАЦИИ, ИЗОБРАЖЕНИЯ

Объект исследования - глубокие нейронные сети.

Предмет исследования - распознавание объектов заданного типа на изображениях с использованием конволюционной нейронной сети.

Цель работы - исследование свойств глубоких нейронных сетей, их основных преимуществ и особенностей архитектуры, предметных областей и задач, которые целесообразно решать с их помощью, на примере решения задачи распознавания объектов заданного типа на изображениях с использованием конволюционной нейронной сети, анализа ее основных возможностей и поведения в ходе решения поставленной задачи, разработка программного приложения распознавания объектов заданного типа на изображениях.

Методы исследования - методы глубокого обучения, методы классификации объектов определенного типа, моделирование процессов классификации объектов с использованием конволюционных нейронных сетей.

В результате проведенных исследований решена задача распознавания объектов заданного типа на изображениях на основании методов глубокого обучения.

Предлагаемая разработка будет полезна для применения в различных отраслях, связанных с использованием фруктов и овощей, например, в фермерских хозяйствах, сельском хозяйстве в целом, кулинарии, ресторанном бизнесе, в продуктовых супермаркетах и тому подобное.

ABSTRACT

Explanatory note: 75 p., 22 fig., 2 ann., 44 sources.

DEEP LEARNING, DEEP NEURAL NETWORKS, CONVOLUTIONAL NEURAL NETWORK, CLASSIFICATION METHODS, IMAGES

The object of research is deep neural networks.

The subject of research is the recognition of objects of a given type in images using a convolutional neural network.

The purpose of the work is to study the properties of deep neural networks, their main advantages and features of architecture, subject areas and problems that should be solved with their help, on the example of solving the problem of recognizing objects of a given type in images using a convolutional neural network. and behavior in solving the problem, development a software application for recognizing objects of a given type in images.

Research methods are deep learning methods, methods of classification of certain types of objects, modeling of object classification processes using convolutional neural networks.

As a result of the conducted researches the problem of recognition of objects of the set type on images on the basis of methods of deep training is solved. The results are used in the development of an application for recognizing certain types of objects. The application platform and the Matlab programming language were used as a software platform.

The proposed development will be useful for use in various industries related to the use of fruits and vegetables, such as farms, agriculture in general, cooking, restaurant business, grocery supermarkets, etc.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	9
Вступ.....	10
1 Аналіз проблемної області та постановка задач дослідження	14
1.1 Аналіз сучасного стану та перспектив розвитку глибинного навчання	14
1.2 Загальна характеристика основних видів глибинних нейронних мереж	18
1.3 Конволюційні нейронні мережі.....	22
1.4 Постановка задач дослідження.....	28
2 Аналіз основних видів задач розпізнавання об'єктів та методів їх вирішення.....	30
2.1 Постановка задачі розпізнавання	30
2.2 Розпізнавання заданого об'єкта серед інших об'єктів.....	31
2.3 Розпізнавання усіх об'єктів, які присутні на зображенні.....	32
3 Розпізнавання об'єктів заданого типу на зображеннях з використанням конволюційної нейронної мережі.....	35
3.1 Обґрунтування вибору архітектури нейромережі.....	35
3.2 Алгоритм навчання конволюційної нейронної мережі	42
3.3 Аналіз отриманих результатів	47
4 Практичне застосування отриманих результатів досліджень	50
4.1 Постановка практичної задачі щодо застосування результатів досліджень	50

4.2 Обґрунтування та вибір програмних засобів для реалізації додатку	52
4.3 Реалізація програмної системи.....	54
Висновки	60
Перелік джерел посилання.....	61
Додаток А.....	66
Додаток Б	74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- СШ - системи штучного інтелекту;
- CNN - Convolutional Neural Network - конволюційна (згорткова) нейронна мережа;
- CI - Computational Intelligence - обчислювальний інтелект;
- CUDA - Compute Unified Device Architecture – обчислена уніфікована архітектура пристрою;
- DNN - Deep Neural Network - глибинна нейронна мережа;
- DPM - Deformable Part-based Model - деформована модель на основі частин;
- FAST - Features from accelerated segment test - ознаки з прискореного тестового сегменту;
- HOG - Histogram of Oriented Gradients - гістограма орієнтованих градієнтів;
- LTSM - Long Short-Term Memory - довга короткострокова пам'ять;
- MSE - Mean Squared Error - середньоквадратична помилка;
- ONNE - Open Neural Network Exchange – відкритий обмін нейронної мережі;
- ReLU - Rectified Linear Unit - випрямлена лінійна одиниця;
- R-CNN - Regions with CNN features - регіони з ознаками конволюційної нейронної мережі;
- RoI - Region of Interest - область зацікавленості;
- RNN - Recurrent Neural Networks - рекурентні нейронні мережі;
- SIFT - Scale-invariant feature transform - масштабно незалежне перетворення ознак;
- SVM - Support Vector Machine - опорно-векторна машина.

ВСТУП

Науково-практичні дослідження в галузі сучасного штучного інтелекту можна віднести до досліджень в галузі обчислювального інтелекту або когнітивного інтелекту [1]. Термін когнітивний інтелект передбачає напрямок науково-практичних досліджень, спрямованих на вирішення інтелектуальних задач, які потребують вивчення й пізнання предметної галузі та використання експертних знань щодо досягнення поставленої мети.

У цьому напрямку зусилля дослідників направлені на розробку й використання методологій, методів та алгоритмів щодо побудови як класичних експертних систем для заданих предметних областей та чітко окресленого кола задач, систем обробки природної мови у різних варіантах, так і веб-орієнтованих систем штучного інтелекту (СШІ), у тому числі розподілених, формування онтологій та онтологічних баз знань задля подання та обміну знаннями у Інтернет-просторі. Слід зазначити, що в останні роки системи, розробка яких здійснюється у цьому напрямку досліджень, все частіше отримують назву когнітивних систем. Тим самим робиться акцент на здібності систем до пізнання та поповнення своїх знань у процесі функціонування завдяки методам машинного навчання, використанню великих даних, баз знань та логічного виведення на знаннях.

Одним із підходів до побудови когнітивних систем є сервісно-орієнтований підхід на принципах відкритої модульної архітектури. Найвідомішою системою такого типу є когнітивна система IBM Watson. Слід зазначити, що у даній системі, як і у всьому напрямку досліджень когнітивного інтелекту сьогодні активно використовуються методи обчислювального інтелекту. Таким чином, ми можемо спостерігати об'єднання та взаємовплив обох напрямів розвитку сучасного штучного інтелекту.

Термін та дисципліна з назвою обчислювальний інтелект (Computational Intelligence, CI) були запропоновані професором Лотфі Заде (Lotfi A. Zadeh) на початку 1990-х років. Слід зауважити, що зміст та організація цього напрямку штучного інтелекту потерпають суттєвих змін протягом часу. Початкове означення обчислювального інтелекту у центр досліджень ставило логіку нечітких множин, нейронні мережі, генетичні алгоритми та ймовірнісне виведення наряду із вивченням їх взаємодії та взаємовпливу у вирішенні складних інтелектуальних задач, таких як розпізнання образів, підтримка прийнятті рішень у складних ситуаціях з урахуванням нечіткості, прогнозування тощо [2].

Сучасне означення та розвиток обчислювального інтелекту формуються під великим впливом біологічно натхненних моделей машинного інтелекту. У цьому напрямку розвиваються моделі гранулярного комп'ютингу (granular computing), нейрокомп'ютингу (neural computing), еволюційних обчислень (evolutionary computing) та їх взаємодія із штучним життям (artificial life), теорією хаосу та інше.

Гранулярний комп'ютинг включає такі підобласті досліджень як нечіткі множини, грубі множини та робота з ними, ймовірнісне виведення. Еволюційні обчислення розвиваються в рамках генетичного програмування, генетичних алгоритмів та ройового інтелекту. Нейрокомп'ютинг передбачає у першу чергу використання нейронних мереж різного типу як систем обробки інформації. Такі системи здатні автономно генерувати методи, правила та алгоритми обробки у вигляді адаптивної відповіді в умовах функціонування в конкретному інформаційному середовищі. У рамках нейрокомп'ютингу використовуються та розвиваються різні типи нейронних мереж та методи їх навчання: з учителем (supervised), без вчителя (unsupervised), з підкріпленням (reinforcement). Слід зазначити, що вибір конкретного типу нейронної мережі залежить від специфіки задачі, яку потрібно вирішити [3, 4].

На цей час в рамках обчислювального інтелекту найбільш актуальними є дослідження у галузі глибинного навчання та глибинних нейронних мереж.

Глибинне навчання - клас алгоритмів машинного навчання, які:

- використовують каскад багатьох шарів вузлів нелінійної обробки для виділення ознак та перетворення. Кожен наступний шар використовує вихід із попереднього шару як вхід;
- ґрунтуються на навчанні (спонтанному) декількох шарів ознак або представлень даних. Ознаки вищих рівнів виводяться з ознак нижчих рівнів для формування ієрархічного представлення;
- є частиною ширшої області машинного навчання з навчання представлень даних;
- навчаються кільком рівням представлень, що відповідають різним рівням абстракції; ці рівні формують ієрархію понять.

До глибинних відносять такі типи нейронних мереж: глибинні мережі переконань, рекурентні нейронні мережі (RNN), конволюційні (згорткові) нейронні мережі (CNN). Глибинне навчання застосовують у комп'ютерному зорі, автоматичному розпізнаванні мовлення, обробці природної мови, розпізнаванні звуків, біоінформатиці, економіці тощо.

За допомогою глибинних нейронних мереж вже успішно вирішується багато науково-практичних задач. Серед них задача класифікації - одна з популярних задач.

Класифікація зображень є важливою темою в системах штучного зору і викликає значний інтерес протягом останніх десятиліть. Це поле спрямоване на класифікацію вхідного зображення на основі візуального контенту. Однак, коли дано багато зображень, знайти ознаки з них занадто складно. Це одна з причин появи глибинної нейронної мережі.

CNN є сучасним підходом до розпізнавання зображень. Вони мають велику кількість шарів, в порівнянні з класичним багатошаровим персептроном. За рахунок спільних ваг, які використовуються відразу

декількома нейронами в кожному шарі, вдається знизити загальну кількість параметрів мережі і прискорити навчання. Також, на відміну від багат шарового персептрона, конволюційні мережі сприйнятливі до топології вхідного зображення.

Враховуючи вище згадане, темою магістерської атестаційної роботи обрано розпізнавання об'єктів заданого типу на зображеннях із використанням CNN.

1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз сучасного стану та перспектив розвитку глибинного навчання

Глибинне навчання - форма машинного навчання, яка передбачає витяг або моделювання ознак даних з використанням складних багатошарових фільтрів [5,6].

Існують різні методи підвищення точності навчання у різних моделях глибинного навчання. Вони класифікуються на методи: підвищення швидкості навчання, трансферне навчання, навчання з нуля та відсічення (dropout) [7].

Метод підвищення швидкості навчання полягає в процесі адаптації швидкості навчання для підвищення продуктивності та скорочення часу на навчання. Швидкість навчання слугує фактором, який визначає систему або встановлює умови для її роботи перед навчальним процесом. Він контролює, наскільки зміниться досвід моделі у відповідь на виявлену та оцінену помилку, щоразу, коли ваги моделі змінюються [8,9]. Занадто високі швидкості показників навчання можуть призвести до нестабільних тренувальних процесів або до вивчення неоптимального набору ваг. Занадто малі показники навчання можуть викликати затягнення процесу навчання.

Трансферне, тобто індуктивне навчання передбачає використання навчання з вчителям, при цьому використовуючи частину попередньо засвоєної моделі в новій мережі, на яку покладено іншу, але схожу проблему [10]. Використання заздалегідь підготовленої моделі значно скорочує час, необхідний для інженерної роботи та навчання.

Метод трансферного навчання базується на наступному алгоритмі, який складається з таких етапів:

- вибір вихідної моделі, в ідеалі - з великим набором даних для навчання;
- визначення шарів для повторного використання у власній мережі. Метою визначення шарів є створення структури, кращою, ніж початкова модель. Зазвичай більш глибокі шари повторно використовуються, оскільки вони мають більш загальне значення, тоді як верхні шари використовуються повторно, оскільки мають більш тонке налаштування до певної проблеми [11]. Найчастіше нова модель навчається на новому наборі даних. Перевага трансферного навчання полягає в тому, що модель має тенденцію до конвергенції набагато швидше і, таким чином, потрібно менше даних та обчислювального часу.

Іншим підвищення точності навчання у різних моделях глибинного навчання є навчання з нуля. Воно вимагає від розробника зібрати великий розмічений набір даних і налаштувати мережеву архітектуру, яка може вивчити ознаки та модель. Цей прийом особливо корисний для нових програм, а також для додатків з великою кількістю категорій виводу. Однак, це менш поширений підхід, оскільки він вимагає великої кількості даних, через що навчання стає занадто тривалим [11].

Також одним із методів підвищення точності навчання нейронної мережі є використання методу відсіву випадково обраних нейронів, який отримав назву dropout. Метод dropout широко використовується для рішення однієї з найпоширеніших проблем глибинних нейронних мереж - перенавчання (overfitting). Перенавчання полягає в наступному: модель добре пояснює тільки приклади з навчальної вибірки, адаптуючись до навчальних прикладів, замість того, щоб вчитися класифікувати приклади, які не брали участі в навчанні (втрачаючи здатність до узагальнення).

Метод dropout полягає у випадковому скиданні нейронів (разом із їх зв'язками) з нейронної мережі під час тренувань [12].

Виключені нейрони не вносять свій внесок в процес навчання ні на одному з етапів алгоритму зворотного поширення помилки

(backpropagation). Тому виключення хоча б одного з нейронів рівнозначно ситуації, коли необхідно здійснити навчання нової нейронної мережі. Графічне представлення здійснення методу dropout [13] показано на рисунку 1.1, 1.2.

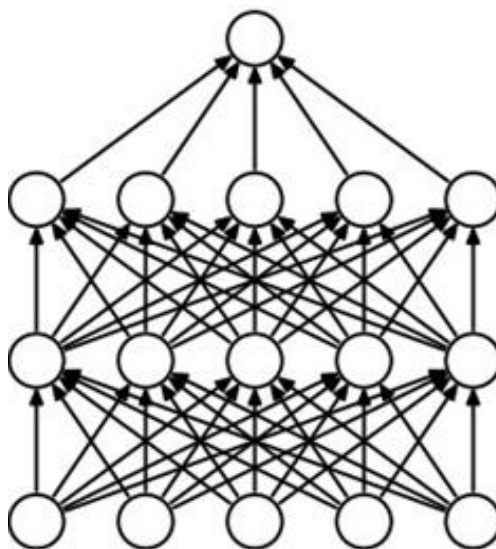


Рисунок 1.1 - Dropout. Вихідна нейронна мережа

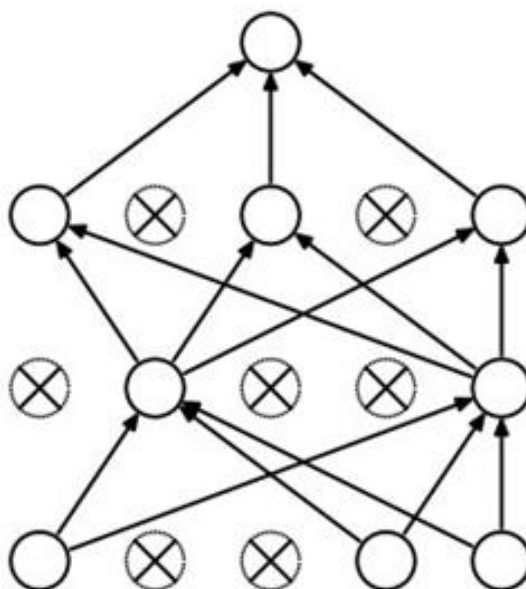


Рисунок 1.2 – Dropout. Результат відсіву випадково обраних нейронів

На рисунку 1.1 зображена вихідна нейронна мережа щодо демонстрації роботи техніки dropout, на рисунку 1.2 – та ж нейронна мережа

після того, як до неї застосували dropout з відсівом випадково обраних нейронів.

Глибинне навчання в даний час використовується в найбільш поширених інструментах розпізнавання зображень, програмному забезпеченні для обробки природної мови та розпізнаванні мовлення. На цей час методи глибинного навчання все активніше застосовуються у різних галузях, причому список таких галузей постійно зростає. Серед найпоширеніших галузей та прикладів використання методів глибинного навчання щодо вирішення прикладних задач можна перелічити такі як:

- генерація тексту. Програми з використанням штучного інтелекту навчаються граматиці та стилю фрагмента тексту, а потім використовують цю модель для автоматичного створення абсолютно нового тексту, що відповідає правильному написанню, граматиці та стилю оригінального тексту;

- промислова автоматизація. Глибинне навчання поліпшує безпеку працівників у таких середовищах, як заводи (металургійні, хімічні тощо), атомні реактори та склади, автоматично визначаючи, коли працівник або небезпечний предмет стають занадто близькими;

- медичне дослідження. Дослідники онкологічних захворювань почали впроваджувати глибинне навчання у свою практику як спосіб автоматичного виявлення ракових клітин;

- комп'ютерний зір. Глибинне навчання значно покращило комп'ютерний зір, забезпечивши комп'ютерам надзвичайну точність для виявлення об'єктів та класифікації зображень, відновлення та сегментації;

- екологічний моніторинг: національний, промислових об'єктів, водного середовища тощо. Глибинне навчання використовуючи велику кількість параметрів прогнозує майбутнє забруднення довкілля та методи за допомогою яких це можна зменшити;

- молекулярний дизайн біологічно активних сполук, розробка ефективних фармацевтичних препаратів. Глибинне навчання,

використовуючи дані онлайн бібліотек хімічних сполук, прогнозує найактивніші сполуки з певним рівнем біологічної дії (протизапальна, діуретична тощо), що дозволяє цілеспрямованим синтезом одержати ці сполуки та виявити їх фармакологічну активність;

- оптимізація технологічних процесів у різних галузях хімічних виробництв, металургії, нафтопереробних виробництв.

Наведені приклади демонструють широкі перспективи методів глибинного навчання щодо вирішення прикладних задач у різних галузях народного господарства. Безсумнівно, що у майбутньому глибинне навчання буде ще активніше розвиватися та використовуватися у різних сферах як наукової так і практичної діяльності.

1.2 Загальна характеристика основних видів глибинних нейронних мереж

Глибинне навчання ґрунтується на побудові та використанні різних типів глибинних нейронних мереж, які, в свою чергу, засновані на класичних архітектурах нейронних мереж, але нарощують кількість шарів, комбінують інші основні характеристики нейронних мереж задля отримання найкращих результатів при вирішенні ускладнених задач на великих об'ємах даних. Серед найпоширеніших видів глибинних нейронних мереж можна відмітити такі як CNN, RNN, нейронна мережа Хопфілда, нейронні мережі типу довга короткострокова пам'ять (LSTM) [13].

Слід відмітити, що кожен тип глибинних нейронних мереж постійно динамічно удосконалюється, набирає нових властивостей щодо покращення результатів своєї роботи. Так, наприклад, конволюційні нейронні мережі, які *de facto* визнані найкращим інструментом для вирішення задач розпізнавання та класифікації зображень різного типу, ідентифікації об'єктів, сьогодні мають вже декілька вдосконалених поширень своєї архітектури, та, здається, це ще далеко не межа можливостей глибинного навчання у цьому напрямку науково-практичних досліджень.

RNN є видом нейронних мереж, де зв'язки між елементами утворюють спрямовану послідовність. Завдяки цьому з'являється можливість обробляти серії подій у часі або послідовні просторові ланцюжки. RNN можуть використовувати свою внутрішню пам'ять для обробки послідовностей довільної довжини. RNN застосовуються в таких завданнях, де щось цілісне розбите на частини, наприклад: розпізнавання рукописного тексту або розпізнавання мови [14].

Нейронна мережа Хопфілда представляє собою повнозв'язну нейронну мережу із симетричною матрицею зв'язків. В процесі роботи динаміка таких мереж сходиться (конвергує) до одного з положень рівноваги [15]. Ці положення визначаються заздалегідь в процесі навчання, вони є локальними мінімумами функціонала, яке називається енергією мережі. Архітектура мережі Хопфілда [16] зображена на рисунку 1.3.

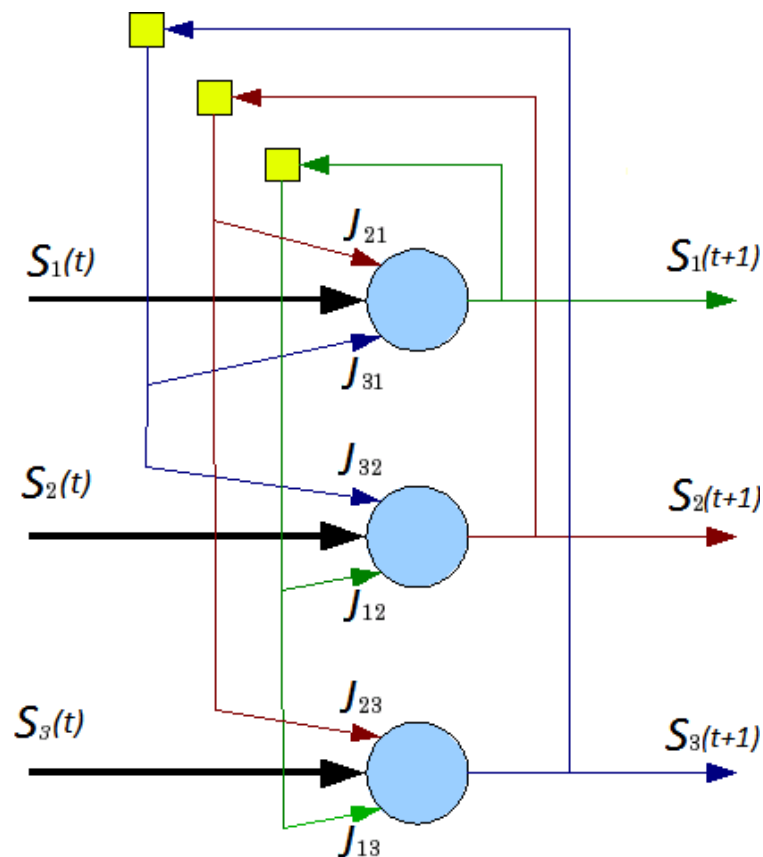


Рисунок 1.3 - Архітектура мережі Хопфілда

Алгоритм навчання мережі Хопфілда полягає в тому, що замість послідовного наближення до потрібного стану з обчисленням помилок, всі коефіцієнти матриці розраховуються за однією формулою за один цикл, після чого мережа відразу готова до роботи. Мережа Хопфілда може працювати в 2 режимах: синхронний, асинхронний. При синхронному режимі послідовно проглядаються нейрони, однак їх стани в пам'яті фіксуються окремо і не змінюються доки вони не будуть пройдені всі. Коли всі нейрони переглянуті, їх стани одночасно змінюються на нові. Таким чином, досягається моделювання паралельної роботи послідовним алгоритмом. При синхронному режимі можливо (хоч і далеко не завжди спостерігається) нескінченне чергування двох станів з різною енергією - так званий динамічний аттрактор. Тому синхронний режим практично для мережі Хопфілда не використовується. При синхронному режимі можливо (хоча і далеко не завжди спостерігається) нескінченне чергування двох станів з різною енергією - так званий динамічний аттрактор. Тому синхронний режим практично для мережі Хопфілда не використовується. В асинхронному режимі стани нейронів в наступний момент часу змінюються послідовно: обчислюється локальне поле для першого нейрона в момент t , визначається його реакція, і нейрон встановлюється в новий стан (яке відповідає його виходу в момент $t + 1$), потім обчислюється локальне поле для другого нейрона з урахуванням нового стану першого, при цьому змінюється стан другого нейрона, і так далі - стан кожного наступного нейрона обчислюється з урахуванням всіх змін станів розглянутих раніше нейронів. В асинхронному режимі неможливий динамічний аттрактор: незалежно від кількості образів, що запам'яталися, і початкового стану, мережа неодмінно прийде до стійкого стану (статичному аттрактору). Нейронна мережа Хопфілда застосовується в задачах обробки візуальних образів (фільтрації та асоціативної пам'яті), а також в комбінаторних задачах оптимізації, якщо вони можуть бути сформульовані як задачі

мінімізації енергії. Класичною проблемою такого типу є задача комівояжера.

Ще одним популярним видом глибинних нейронних мереж є так звані LSTM мережі. Це специфічний тип рекуррентних нейронних мереж, який дозволяє вирішувати певні типи задач набагато ефективніше стандартних методів завдяки тому, що LSTM-мережа здатна навчатися довгостроковим залежностям. Такий тип нейронної мережі було розроблено спеціально для усунення проблеми довгострокової залежності. Спеціалізація таких нейронних мереж полягає у запам'ятовуванні інформації впродовж тривалих періодів часу, тому їх практично не потрібно навчати. Нейронні мережі такого типу містять LSTM-модулі замість або в доповнення до інших мережевих модулів [17]. LSTM-модуль - це рекуррентний модуль мережі, здатний запам'ятовувати значення як на короткі, так і на довгі проміжки часу. Структуру LSTM-мережі наведено на рисунку 1.4 [18].

LSTM-модулі часто групуються в блоки, що містять різні LSTM-модулі. Подібний пристрій характерний для глибинних багат шарових нейронних мереж і сприяє виконанню паралельних обчислень із застосуванням відповідного обладнання. LSTM-блоки містять три або чотири «вентиля», які використовуються для контролю потоків інформації на входах і на виходах пам'яті даних блоків.

Вентилі LSTM реалізовані у вигляді логістичної функції для обчислення значення в діапазоні $[0, 1]$. Множення на це значення використовується для часткового допуску або заборони потоку інформації всередину і назовні пам'яті. Наприклад, «вхідний вентиль» контролює міру входження нового значення в пам'ять, а «вентиль забування» контролює міру збереження значення в пам'яті. «Вихідний вентиль» контролює величину того, в якій мірі значення, що знаходиться в пам'яті, використовується при розрахунку вихідної функції активації для блоку. LSTM застосовується для аналізу часових рядів, для розпізнавання мови, в

задачах розпізнавання рукописного введення, для розпізнавання людської активності.

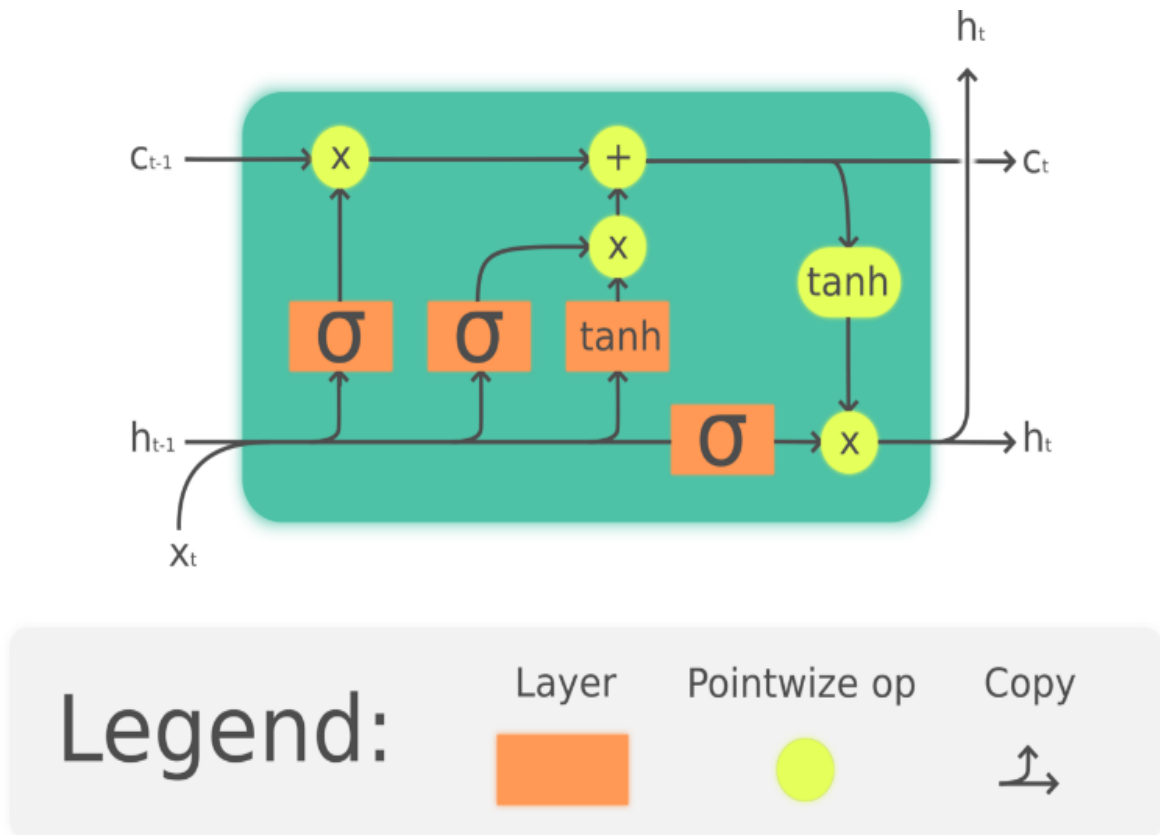


Рисунок 1.4 – Структура LSTM-мережі

1.3 Конволюційні нейронні мережі

CNN - клас глибоких штучних нейронних мереж, який успішно застосовується до аналізу візуальних зображень [19].

На рисунку 1.5 зображена архітектура CNN[20].

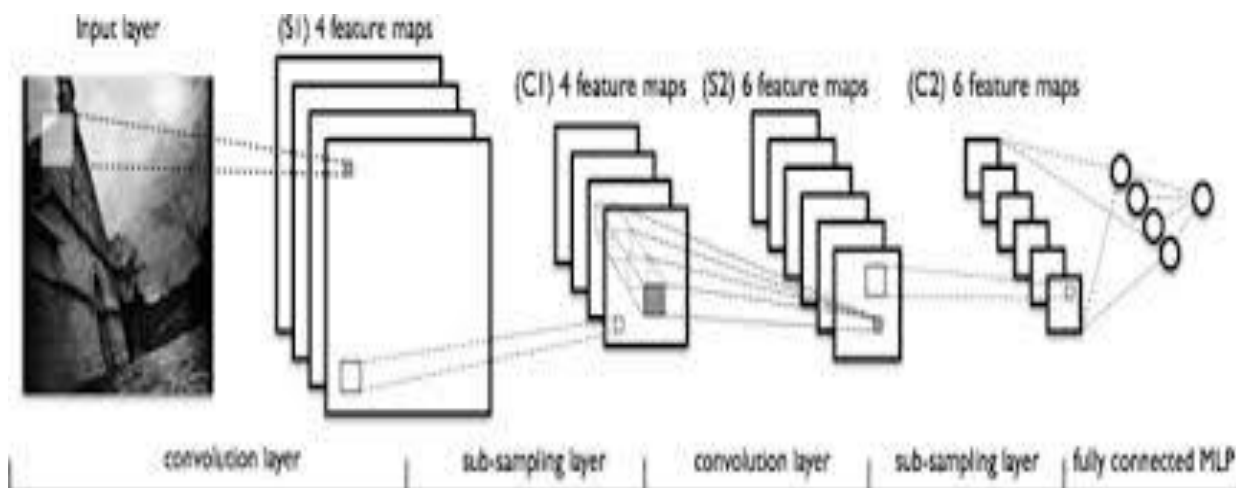


Рисунок 1.5 - Архітектура CNN

CNN складаються з шару згортки, шару активації, пулінгу (шару дискретизації) та повнозв'язної нейронної мережі [20].

Шар згортки - це основний блок CNN. Шар згортки включає в себе для кожного каналу свій фільтр, ядро згортки якого обробляє попередній шар за фрагментами (підсумовуючи результати матричного добутку для кожного фрагмента). Вагові коефіцієнти ядра згортки (невеликої матриці) невідомі і встановлюються в процесі навчання. В конволюційній нейронній мережі в операції згортки використовується лише обмежена матриця ваг невеликого розміру, яку «рухають» по всьому оброблюваному шару (на самому початку - безпосередньо по вхідному зображенню), формуючи після кожного зсуву сигнал активації для нейрона наступного шару з аналогічною позицією. Тобто для різних нейронів вихідного шару використовуються одна і та ж матриця ваг, яку також називають ядром згортки. Її інтерпретують як графічне кодування якої-небудь ознаки, наприклад, наявність похилої лінії під певним кутом. Тоді наступний шар, що вийшов в результаті операції згортки такою матрицею ваг, показує наявність даної ознаки в оброблюваному шарі і її координати, формуючи так звану карту ознак. Природно, в конволюційній нейронній мережі набір ваг не один, а ціла гама, що кодує елементи зображення (наприклад лінії і дуги під різними

кутами). При цьому такі ядра згортки не закладаються дослідником заздалегідь, а формуються самостійно шляхом навчання мережі класичним методом зворотного поширення помилки. Прохід кожним набором ваг формує свій власний примірник карти ознак, роблячи нейронну мережу багатоканальною (багато незалежних карт ознак на одному шарі). Також слід зазначити, що при переборі шару матрицею ваг її пересувають зазвичай не на повний крок (розмір цієї матриці), а на невелику відстань [20].

Скалярний результат кожної згортки потрапляє на функцію активації, яка представляє собою якусь нелінійну функцію. Шар активації зазвичай логічно пов'язують з шаром згортки (вважають, що функція активації вбудована в шар згортки). Для функції нелінійності використовують функцію активації ReLU, яка дозволила істотно прискорити процес навчання і одночасно спростити обчислення (за рахунок простоти самої функції) [20].

Операція субдискретизація виконує зменшення розмірності сформованих карт ознак. У даній архітектурі мережі вважається, що інформація про факт наявності шуканої ознаки важливіше точного знання його координат, тому з кількох сусідніх нейронів карти ознак вибирається максимальний і приймається за один нейрон ущільненої карти ознак меншої розмірності. За рахунок цієї операції, крім прискорення подальших обчислень, мережа стає більш інваріантною до масштабу вхідного зображення [20].

Шар пулінгу (інакше підвибірki, субдискретизація) являє собою нелінійне ущільнення карти ознак, при цьому група пікселів (зазвичай розміру 2×2) ущільнюється до одного пікселя, проходячи нелінійне перетворення. Перетворення зачіпають непересічні прямокутники або квадрати, кожен з яких скорочується в один піксель, при цьому вибирається піксель, що має максимальне значення. Операція пулінг дозволяє істотно зменшити просторовий обсяг зображення. Інтерпретація пулінгу полягає в наступному: якщо на попередній операції згортки вже були виявлені деякі

ознаки, то для подальшої обробки настільки детальне зображення вже не потрібно, і воно ущільнюється до менш детального. До того ж фільтрація вже непотрібних деталей допомагає не перенавчатися. Шар пулінгу, як правило, вставляється після шару згортки перед шаром наступної згортки.

Після кількох проходжень згортки зображення і ущільнення за допомогою пулінгу система перебудовується від конкретної сітки пікселів з високою роздільною здатністю до більш абстрактним карткам ознак, як правило на кожному наступному шарі збільшується число каналів і зменшується розмірність зображення в кожному каналі. Зрештою залишається великий набір каналів, що зберігають невелику кількість даних (навіть один параметр), які інтерпретуються як самі абстрактні поняття, виявлені з вихідного зображення [20].

Ці дані об'єднуються і передаються на звичайну повнозв'язну нейронну мережу, яка теж може складатися з декількох шарів. При цьому повнозв'язні шари вже втрачають просторову структуру пікселів і мають порівняно невелику розмірність (по відношенню до кількості пікселів вихідного зображення). З метою більш агресивного зменшення розміру одержуваних уявлень, все частіше знаходять поширення ідеї використання менших фільтрів або повна відмова від шарів пулінгу.

Найбільш простим і популярним способом навчання є метод навчання з учителем (на маркованих даних) - метод зворотного поширення помилки і його модифікації. Але існує також ряд технік навчання CNN без вчителя. Наприклад, фільтри операції згортки можна навчити окремо і автономно, подаючи на них вирізані випадковим чином шматочки вихідних зображень навчальної вибірки і застосовуючи для них будь-який відомий алгоритм навчання без вчителя (наприклад, автоасоціатор або навіть метод k-середніх) - така техніка відома під назвою patch-based training. Відповідно, наступний шар згортки мережі буде навчатися на шматочках від уже навченого першого шару мережі .

Для поліпшення роботи мережі, підвищення її стійкості і запобігання перенавчання застосовується також dropout - метод тренування підмережі з викиданням випадкових одиничних нейронів.

Відомими моделями конволюційних нейронних мереж є LeNet, AlexNet, ZfNet, VGG, GoogleLeNet.

LeNet була запропонована Лекуном у 1998 році [21]. Вона відома своєю історичною швидкодією, оскільки це була перша CNN, яка показала надзвичайну продуктивність в задачах розпізнавання ручних знаків. Вона має можливість класифікувати цифри, не впливаючи на невеликі спотворення, обертання, зміну положення та масштабу. LeNet – це мережа прямого поширення, що складається з п'яти чергувань шарів згортки і пулінгу, та двох повністю зв'язаних шарів.

AlexNet вважається першою глибокою архітектурою CNN, яка показала новаторські результати для класифікації та розпізнавання зображень. AlexNet була запропонована Крижевським, який підвищив навчальний потенціал CNN шляхом його поглиблення та застосування декількох стратегій оптимізації параметрів. На рисунку 1.6 зображена базова архітектура AlexNet [21].

У AlexNet глибина була розширена з 5 (LeNet) до 8 шарів, щоб зробити CNN застосованим для різних категорій зображень. Незважаючи на те, що загалом глибина покращує узагальнення для різних розширень зображень, але головним недоліком, пов'язаним із збільшенням глибини, є надмірне перенавчання. Для вирішення цього недоліку Крижевський використовує ідею Хінтона, згідно з якою їх алгоритм випадковим чином пропускає деякі трансформаційні одиниці під час навчання для застосування моделі при вивчення більш надійних ознак [21].

У 2013 році створена модель ZfNet [21]. ZfNet була розроблена для кількісної візуалізації продуктивності мережі. Ідея візуалізації мережевої діяльності полягала в тому, щоб контролювати продуктивність CNN,

інтерпретуючи активацію нейрона. DeconvNet працює таким же чином, як і CNN прямого поширення, але змінює порядок операцій згортки та пулінгу.

VGG була зроблена глибиною на 19 шарів [21]. VGG замінила фільтри 11×11 і 5×5 на набір шарів фільтрів 3×3 і експериментально продемонструвала, що одночасне розміщення фільтрів невеликого розміру (3×3) може викликати ефект фільтра великих розмірів (5×5 і 7×7). Використання фільтрів невеликого розміру забезпечує додаткову перевагу низької обчислювальної складності за рахунок зменшення кількості параметрів. Ці результати встановили нову тенденцію в дослідженні роботи з фільтрами меншого розміру в CNN. VGG опинився на 2-му місці у змаганні 2014-го ILSVRC.

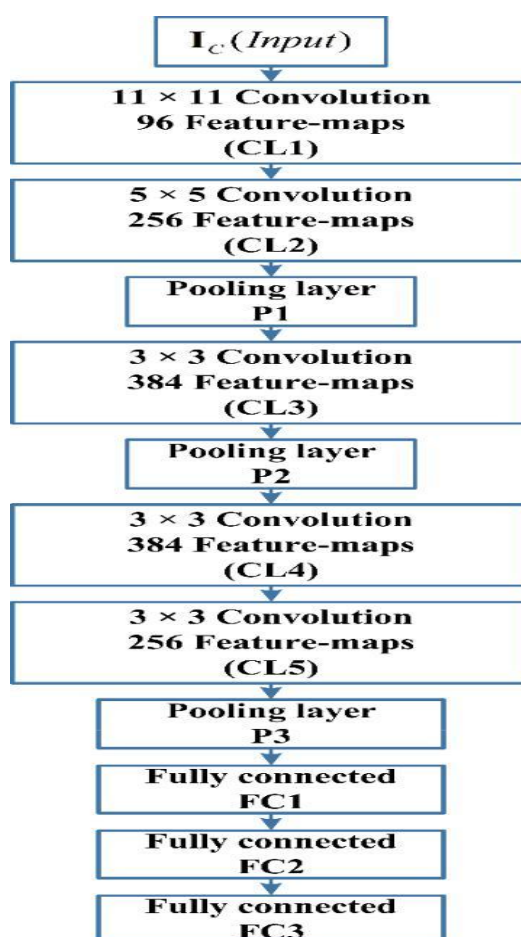


Рисунок 1.6 – Базова архітектура конволюційної нейронної мережі AlexNet

CNN GoogleLeNet, також відома як Inception-V1, розроблена у дослідницькому відділі Google AI та стала переможцем конкурсу 2014-ILSVRC. Головною метою архітектури GoogleNet було досягнення високої точності із зменшенням обчислювальної вартості [21]. Вона вела нову концепцію блоку створення в CNN, завдяки чому він включає в себе багатомасштабні згорткові перетворення, використовуючи ідею розділення, перетворення та злиття.

1.4 Постановка задач дослідження

Глибинне навчання та нейронні мережі сьогодні є одним з актуальних та популярних напрямів розвитку машинного навчання. Як і традиційні методи машинного навчання, глибинні нейронні мережі дозволяють навчити модель передбачати результат по набору вхідних даних. Але на відміну від традиційних нейронних мереж глибинні використовують більше одного прихованого шару та працюють з великими обсягами даних, так званими Big Data. Такий підхід дозволяє отримувати вражаючі результати при вирішенні задач розпізнавання образів різної природи, кола задач автоматичної обробки природної мови, прогнозування в економіці, соціології та інше.

Дана магістерська атестаційна робота присвячена дослідженню властивостей одного з найбільш популярних типів глибинних нейронних мереж, а саме, CNN. Таким чином, об'єктом дослідження даної роботи є глибинні нейронні мережі, а предметом дослідження - розпізнавання об'єктів заданого типу на зображеннях із використанням CNN.

Метою даної роботи є дослідження моделі CNN та динаміки її поведінки при вирішенні низки задач розпізнавання об'єктів заданого типу на зображеннях.

З урахуванням сформульованої мети в даній атестаційній роботі необхідно вирішити низку таких задач:

- провести аналіз науково-технічних публікацій та даних Інтернет-джерел в області глибинного навчання та глибинних нейронних мереж;
- проаналізувати основні типи задач розпізнавання об'єктів на зображеннях та методи їх вирішення;
- обґрунтовано здійснити вибір об'єктів розпізнавання для проведення досліджень;
- вибір та формування наборів даних, навчальної множини об'єктів;
- обґрунтування та вибір архітектури конволюційної нейронної мережі;
- вибір програмних засобів для перевірки працездатності побудованої конволюційної мережі;
- розробка програмного прототипу системи розпізнавання об'єктів заданого типу;
- дослідження можливостей розширення функціоналу прототипу системи.

2 АНАЛІЗ ОСНОВНИХ ВИДІВ ЗАДАЧ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ТА МЕТОДІВ ЇХ ВИРІШЕННЯ

2.1 Постановка задачі розпізнавання

Постановка задачі розпізнавання об'єктів полягає у визначенні місця розташування об'єктів у заданому зображенні (локалізація об'єкта) та категорії, до якого належить кожен об'єкт (класифікація об'єкта). Таким чином, пайплайн традиційних моделей виявлення об'єктів в основному можна розділити на три етапи: вибір інформаційного регіону, вилучення ознак та класифікація.

Вибір інформаційного регіону полягає у використанні природного ковзаючого вікна, оскільки різні об'єкти можуть з'являтися в будь-яких положеннях зображення та мати різні співвідношення сторін чи розмірів. Хоча ця вичерпна стратегія дозволяє з'ясувати всі можливі положення об'єктів, її недоліки також очевидні. Завдяки великій кількості вікон-кандидатів, це обчислювально дорого і виробляє занадто багато зайвих вікон. Однак якщо застосовано лише фіксовану кількість шаблонів ковзаючих вікон, можуть бути створені незадовільні регіони.

Вилучення ознак полягає в тому, що для розпізнавання різних об'єктів потрібно витягти візуальні особливості, які можуть забезпечити семантичне та надійне зображення. Репрезентативними є функції: масштабонезалежне перетворення ознак (SIFT, Scale-invariant feature transform), гістограма орієнтованих градієнтів (HOG, Histogram of oriented gradient) та Haar-подібні [22,23,24].

Крім того, потрібен класифікатор, щоб відрізнити цільовий об'єкт від усіх інших категорій та зробити уявлення більш ієрархічними, семантичними та інформативними для візуального розпізнавання. Зазвичай, опорна векторна машина (SVM), AdaBoost та модель на основі деформованої частини (DPM) є хорошим вибором.

Серед цих класифікаторів DPM - це гнучка модель, що поєднує деталі об'єкта з вартістю деформації для обробки важких деформацій. У DPM за допомогою графічної моделі ретельно розроблені низькорівневі характеристики та кінематично натхненні декомпозиції деталей. А дискримінаційне вивчення графічних моделей дозволяє побудувати високоточні моделі на основі деталей для різних класів об'єктів.

2.2 Розпізнавання заданого об'єкта серед інших об'єктів

Розпізнавання заданого об'єкта спрямовано на виявлення раніше не бачених екземплярів об'єкта. Задача по розпізнаванню об'єктів вирішується з використанням детекторів та дескриптору. До детекторів відносять ознаки з прискореного тестового сегменту (FAST), кутовий детектор Харріса, а до дескриптору - SIFT.

Витягнення ознак включає обчислення дескриптора, що зазвичай робиться в регіонах, зосереджених навколо виявлених ознак. Дескриптори покладаються на обробку зображень для перетворення локального піксельного сусідства в компактне векторне зображення. Це нове представлення дозволяє порівнювати між районами незалежно від зміни масштабу чи орієнтації.

FAST представляє собою кутовий метод виявлення, який може бути використаний для вилучення точок функцій і пізніше використовується для відстеження та картографування об'єктів у багатьох завданнях комп'ютерного зору. Найбільшою перевагою кутового детектора FAST є його обчислювальна ефективність. Обмеження FAST полягає в тому, що він не виявляє кути на зображеннях, створених комп'ютером, які ідеально вирівнюються до осей x та y [25].

Кутовий детектор Харріса - кутовий оператор виявлення, який використовується в алгоритмах для вилучення кутів та виведення особливостей зображення. Кутовий детектор є більш точним у розрізненні країв і кутів. Кути є важливими особливостями зображення, і їх, як правило,

називають точками інтересу, які інваріантні для перекладу, обертання та освітлення. Ідея кутового розпізнавання полягає в тому, щоб розглянути невелике вікно навколо кожного пікселя p на зображенні. Треба визначити всі такі піксельні вікна, які є унікальними. Унікальність можна виміряти, переміщуючи кожне вікно на невелику кількість у заданому напрямку та вимірюючи величину змін, що відбуваються у значеннях пікселів [26].

SIFT є алгоритмом виявлення локальних ознак для виявлення та опису локальних особливостей у зображеннях [27]. Цей алгоритм складається з чотирьох етапів:

- вибір пікового масштабу в просторі: потенційне місце знаходження особливостей;
- локалізація ключових точок: точне розташування ключових точок функції;
- призначення орієнтації: присвоєння орієнтації ключовим точкам;
- дескриптор ключових точок: опис ключових точок як векторного розміру.

2.3 Розпізнавання усіх об'єктів, які присутні на зображенні

Загальне виявлення об'єктів спрямоване на пошук та класифікацію існуючих об'єктів на будь-якому одному зображенні та маркування їх прямокутними обмежувальними полями, щоб показати конфіденційність існування. Для розпізнавання усіх об'єктів на зображенні використовують такі нейронні мережі, як регіони з ознаками конволюційної нейронної мережі (R-CNN), Fast R-CNN [28].

R-CNN є мережею, яка додає до властивостей класичної CNN виділення об'єктів (регіонів) задля можливостей мережі здійснювати не тільки класифікацію об'єктів, але й розпізнавання. Приклад R-CNN наведено на рисунку 2.1 [29].

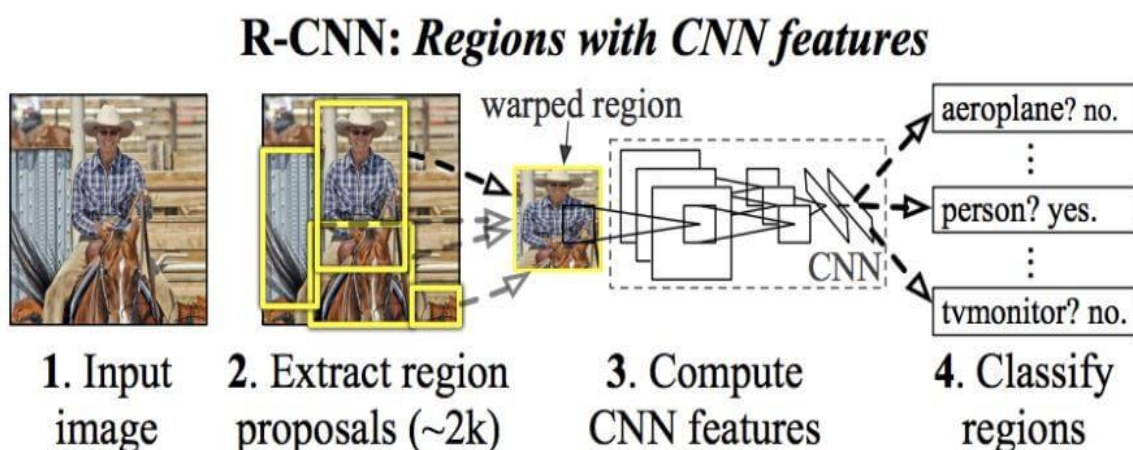


Рисунок 2.1 – Приклад узагальненого представлення R-CNN

Система розпізнавання R-CNN складається з трьох модулів. Перший генерує незалежні від категорії, пропозиції регіону. Ці пропозиції визначають набір розпізнаваних кандидатів, присутніх зображенні. Другий модуль являє собою глибинну CNN, яка витягує вектор ознак з кожної області. Третій модуль - це набір класифікатора, специфічного для класу, лінійних опорній векторній машині (SVM). R-CNN створює ці обмежувальні поля, використовуючи процес, який називається селективний пошук. Селективний пошук розглядає зображення через вікна різних розмірів, і для кожного розміру намагається групувати суміжні пікселі за текстурою, кольором або інтенсивністю для ідентифікації об'єктів. На останньому шарі CNN R-CNN додає підтримку SVM, яка класифікує, чи це об'єкт, і якщо так, то який об'єкт [29].

Fast R-CNN – покращений алгоритм Fast-CNN, який більш точний та швидкий. На рисунку 2.2 зображена архітектура Fast R-CNN [29].

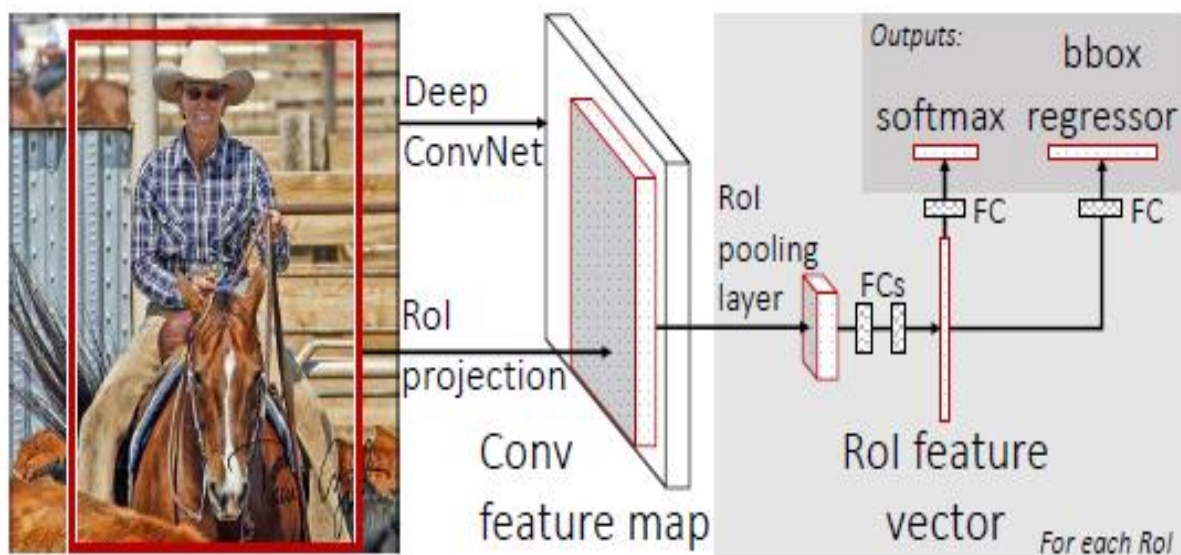


Рисунок 2.2 – Архітектура Fast R-CNN

Fast R-CNN приймає як вхід ціле зображення та набір об'єктних пропозицій. Мережа спочатку обробляє все зображення декількома згортковими та max pooling об'єднаннями шарів для створення згорткової карти ознак. Потім для кожної об'єктної пропозиції область зацікавленості (RoI) pooling шару витягує функціональний вектор фіксованої довжини з карти ознак. Кожен вектор ознак подається у послідовність повністю з'єднаних шарів (fully connected), які, нарешті, розгалужуються на два вихідних шари однорідних передач: той, який виробляє оцінки ймовірності softmax для K класів об'єктів плюс загальний "фонний" клас та інший шар, який видає чотири реальні - числа для кожного з класів об'єктів K . Кожен набір з 4 значень кодує вдосконалені положення обмежувального прямокутника для одного з K класів. Шар об'єднання RoI використовує максимальне об'єднання для перетворення розміщення в будь-якій дійсній області, що цікавить, невелику карту зображень з фіксованою просторовою мірою $H \times W$ (наприклад, 7×7), де H і W - гіперпараметри шару, які не залежать від конкретного RoI [29].

3 РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗАДАНОГО ТИПУ НА ЗОБРАЖЕННЯХ З ВИКОРИСТАННЯМ КОНВОЛЮЦІЙНОЇ НЕЙРОННОЇ МЕРЕЖІ

3.1 Обґрунтування вибору архітектури нейромережі

Найпростіший варіант CNN складається з шару згортки, ReLU шару, макспулінгу, повнозв'язного шару, softmax та classification шару.

На рисунку 3.1 зображена архітектура найпростішої CNN для розпізнавання об'єктів заданого типу.

Вона містить шар введення зображення (*input image layer*), що вводить у мережу двовимірні зображення та застосовує нормалізацію даних; 2-D конволюційний шар (*convolution2Dlayer*), що застосовує ковзаючі згорткові фільтри до входу. Шар згортає вхід, переміщуючи фільтри вздовж входу вертикально і горизонтально, обчислюючи крапковий добуток ваг і введення, а потім додаючи термін зміщення. Звитий шар складається з нейронів, які поєднуються з субрегіонами вхідних зображень або виходами попереднього шару. Цей шар вивчає функції, локалізовані в цих регіонах під час сканування через зображення.

Набір ваг, який застосовується до області на зображенні, називається фільтром. Фільтр рухається уздовж вхідного зображення вертикально і горизонтально, повторюючи однакові обчислення для кожної області. Іншими словами, фільтр згортає вхід. Кількість ваг у фільтрі дорівнює обчислюється за формулою 3.1:

$$NumWeights = w * h * c, \quad (3.1)$$

де *NumWeights* – кількість ваг у фільтрі;

h – висота фільтра;

w - ширина фільтра;

c - кількість каналів на вході.

Наприклад, якщо вхід являє собою кольорове зображення, кількість кольорових каналів дорівнює 3. Кількість фільтрів визначає кількість каналів на виході згорткового шару. Коли фільтр рухається по входу, він використовує той же набір ваг і однакові зміщення для згортання, утворюючи карту особливостей. Кожна карта зображень є результатом згортки з використанням різного набору ваг і різного зміщення. Отже, кількість карт функцій дорівнює кількості фільтрів. Загальна кількість параметрів у згортковому шар обчислюється за формулою 3.2:

$$TotalParameters = (h * w * c + 1) * \text{Number of Filters}, \quad (3.2)$$

де $TotalParameters$ – загальна кількість параметрів у згортковому шарі;

h – висота фільтру;

w – ширина фільтру;

c – глибина фільтру;

1 – зміщення;

Number of Filters – кількість фільтрів.

Вихідна висота згорткового шару обчислюється за формулою 3.3:

$$OutputHeight = (Input\ Size - ((Filter\ Size - 1) * Dilation\ Factor + 1) + 2 * Padding) / Stride + 1, \quad (3.3)$$

де $OutputHeight$ – вихідна висота конволюційного шару;

$Input\ Size$ – розмір вхідних даних;

$Filter\ Size$ – розмір фільтру;

$Dilation\ Factor$ – коефіцієнт розширеної згортки;

$Padding$ – падінг, який застосовується до межі зображення;

$Stride$ – розмір кроку для переміщення фільтра.

Вихідна ширина згорткового шару обчислюється за формулою 3.4:

$$OutputWeight = (Input\ Size - ((Filter\ Size - 1) * Dilation\ Factor + 1) + 2 * Padding) / Stride + 1, \quad (3.4)$$

де *OutputWeight* – вихідна ширина конволюційного шару;
Input Size – розмір вхідних даних;
Filter Size – розмір фільтру;
Dilation Factor – коефіцієнт розширеної згортки;
Padding – падінг, який застосовується до межі зображення;
Stride – розмір кроку для переміщення фільтра.

Ці значення повинні бути цілим числом, щоб зображення було повністю вкрите. Якщо поєднання цих параметрів не призводить до повного покриття зображення, програмне забезпечення за замовчуванням ігнорує решту частини зображення вздовж правого та нижнього країв згортки .

ReLU шар виконує порогову операцію для кожного елемента входу, де будь-яке значення менше нуля встановлюється на нуль. За шарами згортки та batch нормалізації зазвичай використовується нелінійна функція активації, така як ReLU, визначена шаром ReLU. Шар ReLU не змінює розмір свого входу [30].

Шар максимального об'єднання (`maxPooling2dLayer`) виконує вибірку вниз шляхом поділу вводу на прямокутні області об'єднання та обчислення максимуму кожної області. Шари максимального об'єднання слідує за згортковими шарами для відбору проб вниз, отже, зменшуючи кількість зв'язків із наступними шарами. Вони не виконують ніякого навчання самостійно, але зменшують кількість параметрів, які слід вивчити в наступних шарах. Вони також допомагають зменшити надмірний наліт. Шар максимального об'єднання повертає максимальні значення

прямокутних областей. Шари об'єднання сканують через вхід по горизонталі та вертикалі в розмірах кроків. Якщо розмір пулу менший або рівний кроку, то регіони об'єднання не перетинаються [31].

Повністю з'єднаний шар (`fullyConnectedLayer`) множить вхід на вагову матрицю і потім додає вектор зміщення. Всі нейрони у повністю зв'язаному шарі з'єднуються з усіма нейронами попереднього шару. Цей шар поєднує в собі всі функції (локальну інформацію), засвоєні попередніми шарами по всьому зображенню, щоб визначити більші шаблони. Для проблем з класифікацією останній повністю пов'язаний шар поєднує в собі функції класифікації зображень [32].

В класифікаційному (`classification`) шарі, призначається кожному входу один з взаємно-виключних класів K , використовуючи функцію перехресної ентропії для схеми кодування $1 - K$ [33]. Розрахунок функції перехресної ентропії проводиться за формулою 3.5:

$$loss = - \sum_{i=1}^N \sum_{j=1}^K t_{ij} \ln y_{ij}, \quad (3.5)$$

де N - кількість вибірок;

K - кількість класів;

t_{ij} - показник, що i -й зразок належить до j -го класу;

y_{ij} - вихід вибірки i для класу j ;

$loss$ - поперечна втрата ентропії.

Конволюційна мережа складається з одного шару згортки. Перший шар – шар згортки, що приймає 16 5×5 фільтрів. На цьому шарі застосовується `max pooling` з фільтром розміром 2×2 з кроком 2, який вказує, що об'єднані регіони не перетинаються. Після конволюційного шару застосовується ReLU шар. ReLU шар виконує порогову операцію для кожного елемента входу, де будь-яке значення менше нуля встановлюється в нуль. Після ReLU шару знаходиться повнозв'язний шар. Цей шар поєднує

в собі всі функції (локальну інформацію), засвоєні попередніми шарами по всьому зображенню для визначення більших шаблонів. Наступним після повнозв'язного шару є softmax шар. Останнім є classification шар. Classification шар обчислює поперечні втрати ентропії для задач класифікації класів із взаємовиключними класами.

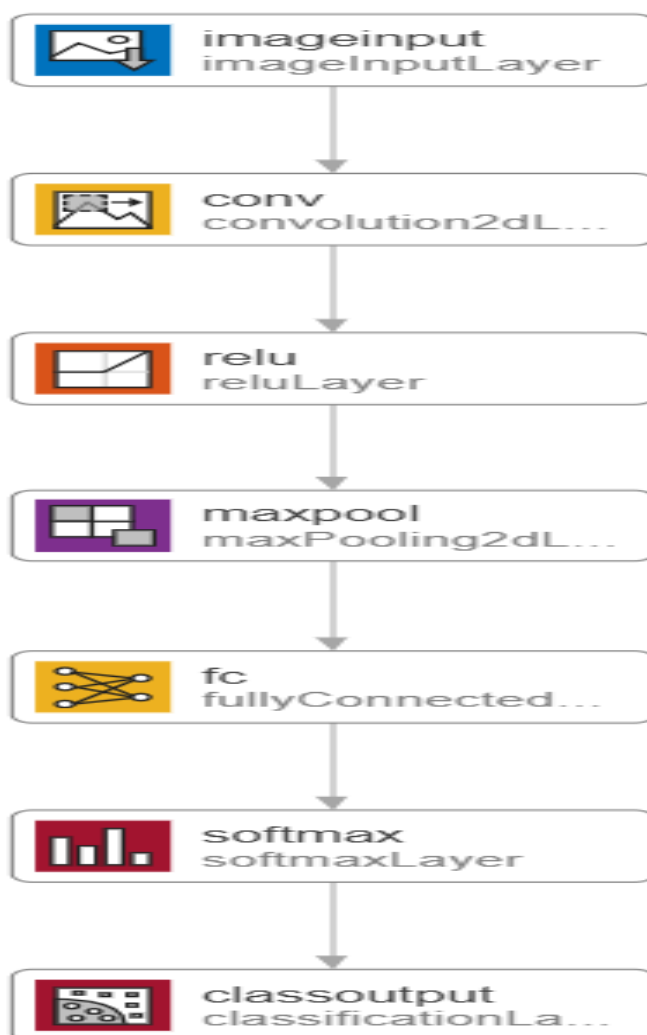


Рисунок 3.1 - Архітектура найпростішої CNN

Використання найпростішої CNN не дозволяє одержати надійні результати, тому що не враховуються всі особливості зображення. Тому на практиці використовують більш складні CNN. В даній атестаційній роботі пропонується CNN, яка має більш складну архітектуру у порівнянні із

класичною. На рисунку 3.2. наведено архітектуру запропонованої CNN з більш складною архітектурою.

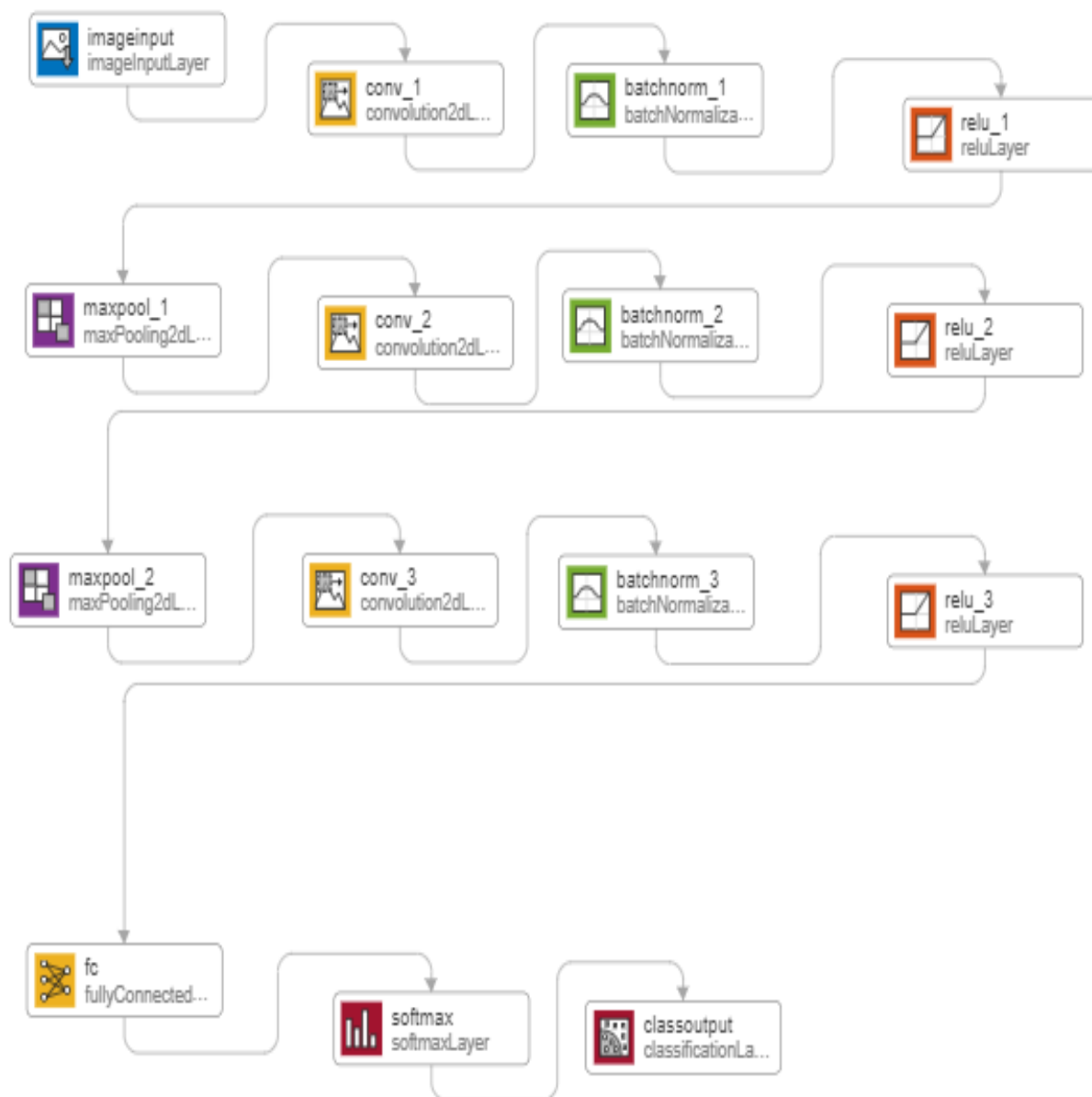


Рисунок 3.2 – Архітектура більш складної нейронної мережі щодо розпізнавання об'єктів заданого типу на зображеннях

Пропонована конволюційна мережа складається з трьох шарів згортки. Перший шар – шар згортки, що приймає шістнадцять фільтрів розміром 5x5. На цьому шарі застосовується max pooling з фільтром розміром 2x2 з кроком 2, який вказує, що об'єднані регіони не

перетинаються (перший max pooling). Другий шар згортки приймає тридцять два фільтри розміром 5x5, які виводять 32 активаційні карти. На другому шарі застосовується такий самий вид max pooling, як на першому шарі, розміром 2x2 та кроком 2. Третій шар згортки приймає 64 фільтрів розміром 5x5. Після нього maxpooling з фільтром розміром 2x2 з кроком 2. Після кожного конволюційного шару застосовується batch normalization шар.

Batch normalization шар нормалізує кожен вхідний канал через міні-пакет. Batch normalization шар знаходиться між згортковим шаром та нелінійним ReLU шаром для прискорення навчання конволюційних нейронних мереж. ReLU шар виконує порогову операцію для кожного елемента входу, де будь-яке значення менше нуля встановлюється в нуль. Після останнього ReLU шару знаходиться повнозв'язний шар. Цей шар поєднує в собі всі функції (локальну інформацію), засвоєні попередніми шарами по всьому зображенню для визначення більших шаблонів. Наступним після повнозв'язного шару є softmax шар. Останнім є classification шар [34].

Classification шар обчислює поперечні втрати ентропії для задач класифікації із взаємовиключними класами [35].

Для апробації архітектури CNN в Matlab потрібно цю мережу навчити.

Параметрами тренувального процесу є: validation accuracy, epoch, iteration, iterations per epoch, maximum iterations, frequency, learning rate schedule, learning rate, training [36].

Параметр validation accuracy означає точність класифікації валідаційних даних [36].

Параметр epoch означає гіперпараметр, який визначає кількість разів, коли алгоритм навчання буде працювати через весь навчальний набір даних. Одна епоха означає, що кожен приклад навчального набору даних

має можливість оновити внутрішні параметри моделі. Епоха складається з однієї або декількох серій [36].

`Iteration` - це кількість серій або кроків через секціоновані пакети навчальних даних, необхідних для завершення однієї епохи [36].

`Learning rate` є параметром налаштування в алгоритмі оптимізації, який визначає розмір кроку на кожній ітерації, рухаючись в напрямку мінімуму функції втрат. Оскільки `learning rate` впливає на те, наскільки новопридбана інформація перекриває стару інформацію, вона представляє собою швидкість, з якою вчиться модель машинного навчання. Існує компроміс між швидкістю конвергенції та перерегулюванням, коли встановлюється швидкість навчання. Занадто високий `learning rate` робить перехід навчання через мінімуми, але занадто низький `learning rate` забиратиме занадто багато часу, щоб сходиться або застрягати в небажаному місцевому мінімумі [36].

`Learning rate schedule` змінює `learning rate` під час навчання і найчастіше змінюється між епохами. В основному це робиться з двома параметрами: занепадом і імпульсом. Існує багато різних графіків `learning rate`, але найпоширенішими є часові, поетапні та експоненціальні [36].

`Training` представляє собою згладжену точність тренувань, отриману шляхом застосування алгоритму згладжування до точності тренувань. Він менш шумний, ніж незгладжена точність, полегшує помітити тенденції [36].

3.2 Алгоритм навчання конволюційної нейронної мережі

Навчання алгоритмом зворотного поширення помилки передбачає два проходи по всім шарам мережі: прямого і зворотного. При прямому проході вхідний вектор подається на вхідний шар нейронної мережі, після чого поширюється по мережі від шару до шару. В результаті генерується набір

вихідних сигналів, який і є фактичною реакцією мережі на даний вхідний образ. Під час прямого проходу всі синаптичні ваги мережі фіксовані. Під час зворотного проходу всі синаптичні ваги налаштовуються відповідно до правила корекції помилок, а саме: фактичний вихід мережі віднімається з бажаного, в результаті чого формується сигнал помилки. Цей сигнал згодом поширюється по мережі в напрямку, протилежному напрямку синаптичних зв'язків [37].

Спочатку конволюційна нейронна мережа не підготовлена (не налаштована). Процес навчання означає послідовне пред'явлення зображення до входу в нейронну мережу з навчальної вибірки, отримана відповідь порівнюється з потрібним виходом

Таким чином, підготовка нейронної мережі зводиться до мінімізації функції помилки шляхом коригування ваги коефіцієнтів синапальних зв'язків між нейронами. Функція помилки - різниця між отриманим відповіддю і бажаною. Потім ваги вихідного шару нейронів коригуються відповідно до помилки. Для нейронів вихідного шару відомі свої фактичні і бажані значення виходів. Тому встановлення ваг зв'язків для таких нейронів є відносно простим. Однак для нейронів у попередніх шарів, налаштування не таке очевидне. Довгий час не було відомого алгоритму для розповсюдження помилки в прихованих шарах.

Для навчання CNN був використаний `backpropagation`.

Для вихідного шару корективна вага інтуїтивно зрозуміла, але для прихованих шарів аналогічний алгоритм довго був не відомим. Ваги прихованого нейрона повинні змінюватись прямо пропорційно помилці тих нейронів, з якими даний нейрон пов'язаний. Тому зворотнє поширення помилок через мережу дозволяє коректно налаштувати ваги зв'язків між всіма шарами. В цьому випадку величина функції помилки зменшується і мережа навчається [38].

Значення помилки визначається через середньоквадратичну помилку (MSE), яка наведена у формулі 3.6:

$$E_p = \frac{1}{2} \sum_j (t_{pj} - y_{pj})^2, \quad (3.6)$$

де E_p – величина помилки для зображення p ;

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

y_{pj} – активований вихід нейрона j для зображення p ;

Неактивований стан кожного нейрона j для зображення p записується у вигляді зваженої суми за формулою 3.7:

$$S_{pj} = \sum_i w_{ij} y_{pi} \quad (3.7)$$

де S_{pj} – зважена сума добутків виходів пов'язаних нейронів попереднього шару та вагів зв'язку або неактивований стан нейрона j для зображення p ;

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

y_{pi} – активований стан нейрона i попереднього шару для зображення p .

Вихід кожного нейрона j є значенням активаційної функції f_j , яка переводить нейрон в активований стан, яка обчислюється за формулою 3.8.

$$y_{pj} = f_j(S_{pj}), \quad (3.8)$$

де y_{pj} – активований стан нейрона j попереднього шару для зображення p ;

f_j – функція активації;

S_{pj} – неактивований стан нейрона j для зображення p .

Ідея градієнтного спуску полягає в мінімізації функції витрат. Функція витрат представляє собою різницю між прогнозами, зробленими алгоритмом, і вірним значенням. Цю різницю можна обчислити багатьма способами, але, найпопулярнішим методом є використання MSE за формулою 3.6. На рисунку 3.3 зображено градієнтний спуск [39].

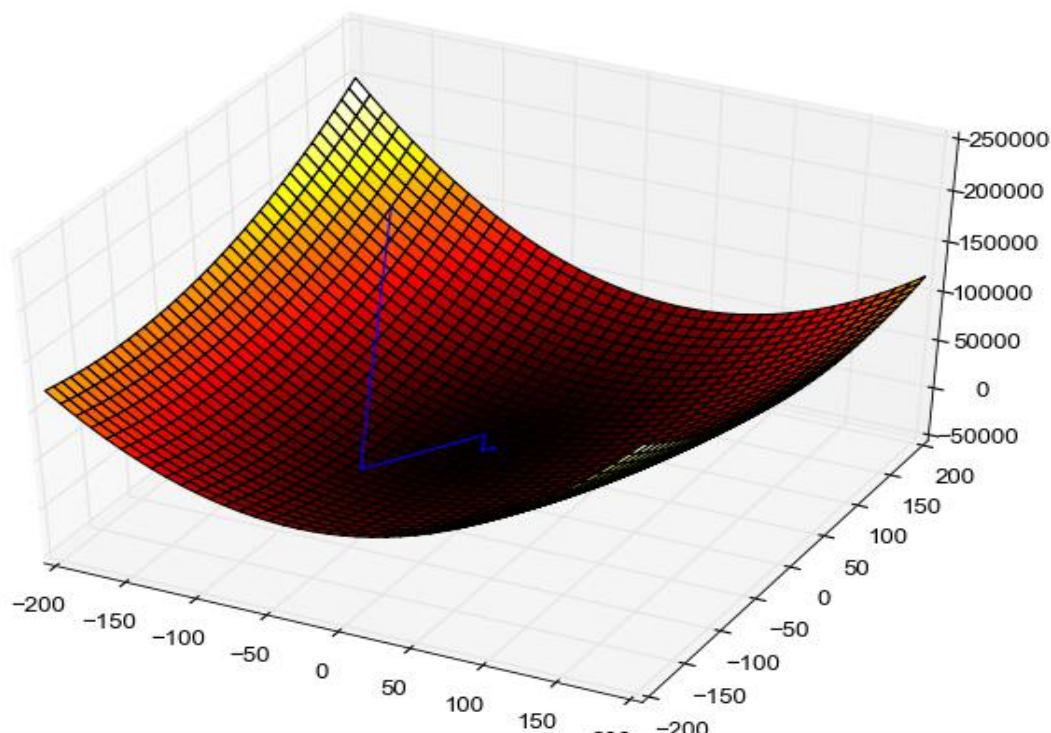


Рисунок 3.3 – Градієнтний спуск

Спочатку визначаємо випадкові значення x та z , які потім будуть використані у моделі. У наведеній на рисунку 3.3 моделі це приблизно -50 , 50 і втрата функції в цій точці становить приблизно 200000 . Тепер градієнтний спуск дивиться в кожний бік і намагається знайти напрямок, де спуск є найкрутішим (математично це обчислення градієнта). З нової точки він знову обчислює градієнт (найкрутіший спуск) та змінює параметри, використовуючи ці значення. Градієнтний спуск продовжить доти, поки він не досягне мінімуму. На рисунку 3.3 синя лінія позначає хід градієнтного спуску, тобто алгоритм дозволив визначити правильні параметри. Але на практиці одного проходження градієнтного спуску для отримання вірних результатів недостатньо [39].

Параметр швидкість навчання (розмір кроку), говорить про те, скільки градієнтів витрачається на кожен крок. Після кожного кроку градієнт шукає новий напрямок (де найкрутіший спуск). На рисунку 3.4 наведено графік залежності втрат від швидкості навчання у логарифмічній шкалі [39].

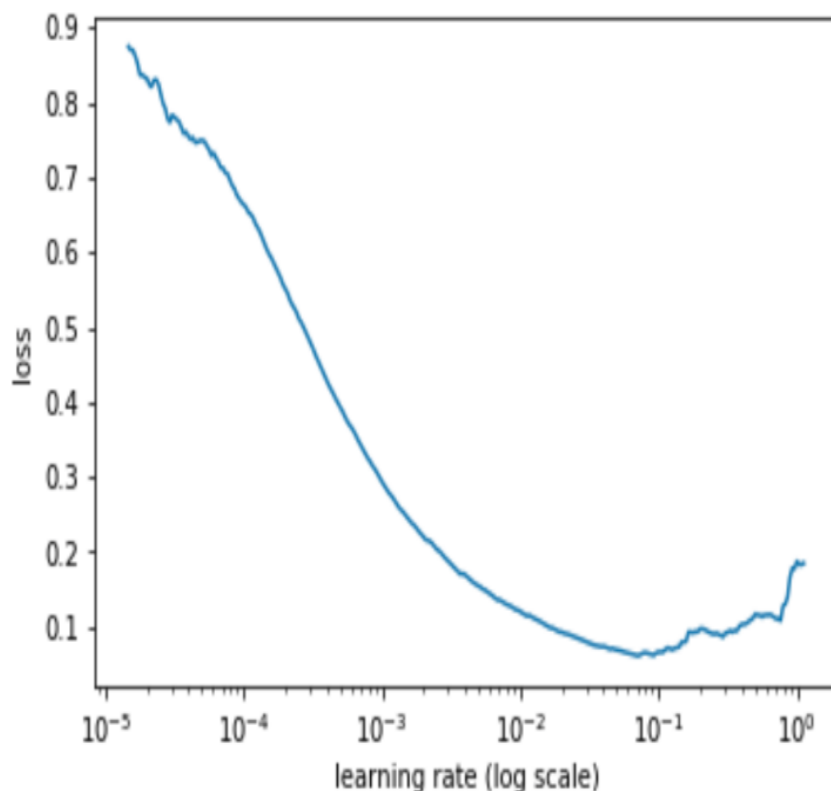


Рисунок 3.4 – Графік залежності втрат від швидкості навчання у логарифмічній шкалі

Але при цьому виникає затруднення у визначенні розміру кроку, тому що якщо обрано занадто великий його розмір, градієнтний спуск може перейти за найнижчу точку або якщо рівень навчання занадто низький, він рухається повільно і знадобиться занадто багато часу, щоб знайти найнижчу точку. Однак ця проблема була вирішена в [40].

Спочатку обирають невелику швидкість навчання (наприклад, 0,000001), а потім збільшують її, помноживши на 2 за кожну ітерацію, поки не досягнуть деякої великої кількості (наприклад, 5). Ідея полягає в тому, що градієнтний спуск спочатку рухається повільно і з часом стає занадто великим. Потім відкладають швидкість навчання на координатну вісь x і середню квадратичну помилку на вісь y та будують графік.

3.3 Аналіз отриманих результатів

Задля вирішення поставлених в атестаційній роботі задач спочатку була використана найпростіша конволюційна мережа, архітектура якої показана на рисунку 3.1.

Результати навчання цієї мережі наведені на рисунку 3.5.

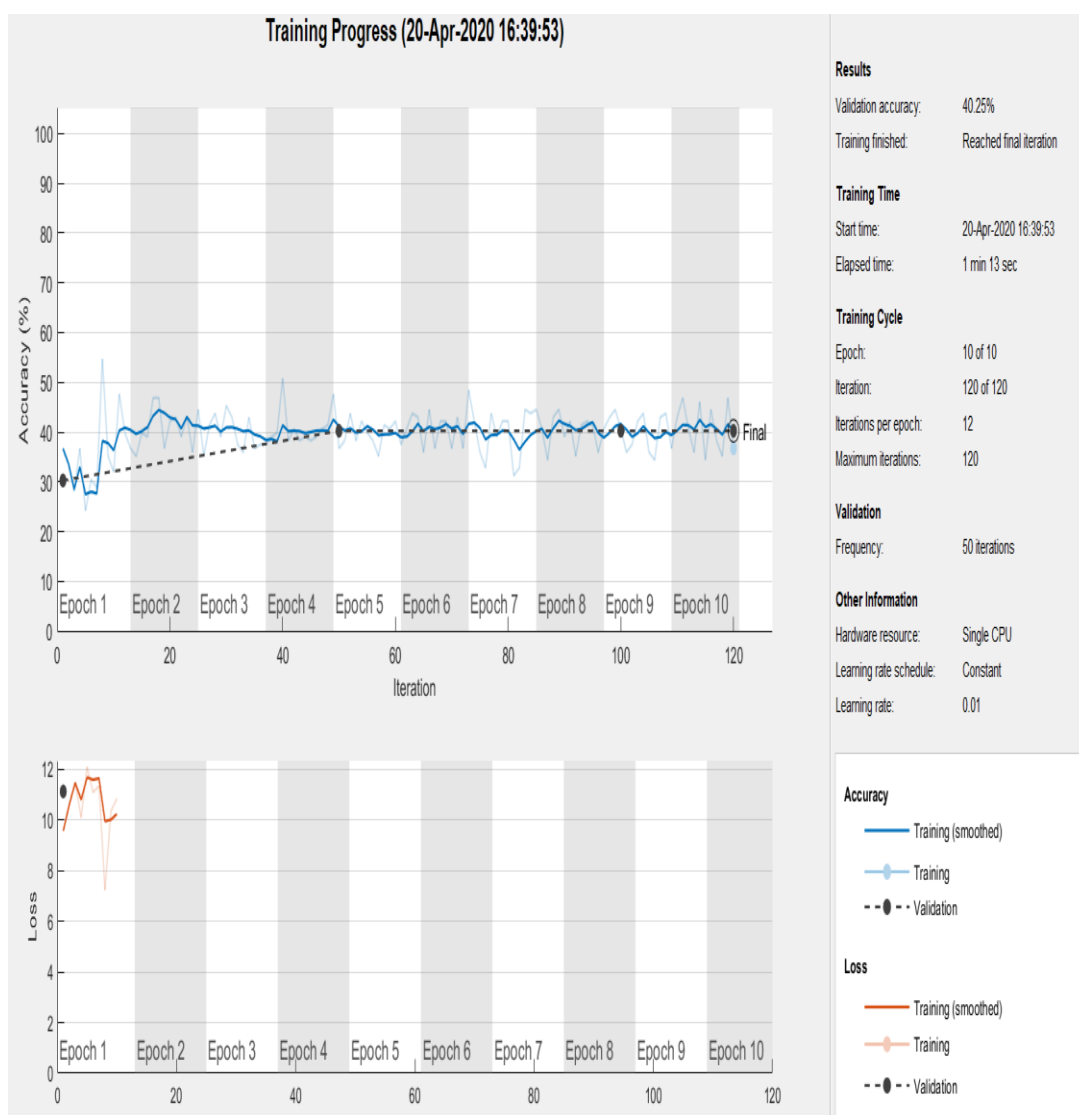


Рисунок 3.5 – Тренування CNN

Отримані наступні значення параметрів:

- Validation accuracy – 40.25%;

- Epoch – 10 of 10;
- Iteration – 120 of 120;
- Frequency – 50 iterations;
- Iterations per epoch -12;
- Maximum iterations – 120;
- Frequency – 50 iterations;
- Learning rate schedule – constant;
- Learning rate – 0.01.

Звертає увагу, що точність валідації занадто мала, тому для одержання більш надійних результатів була обрана більш складна конволюційна нейронна мережа.

Результати навчання більш складної мережі наведені на рисунку 3.6.

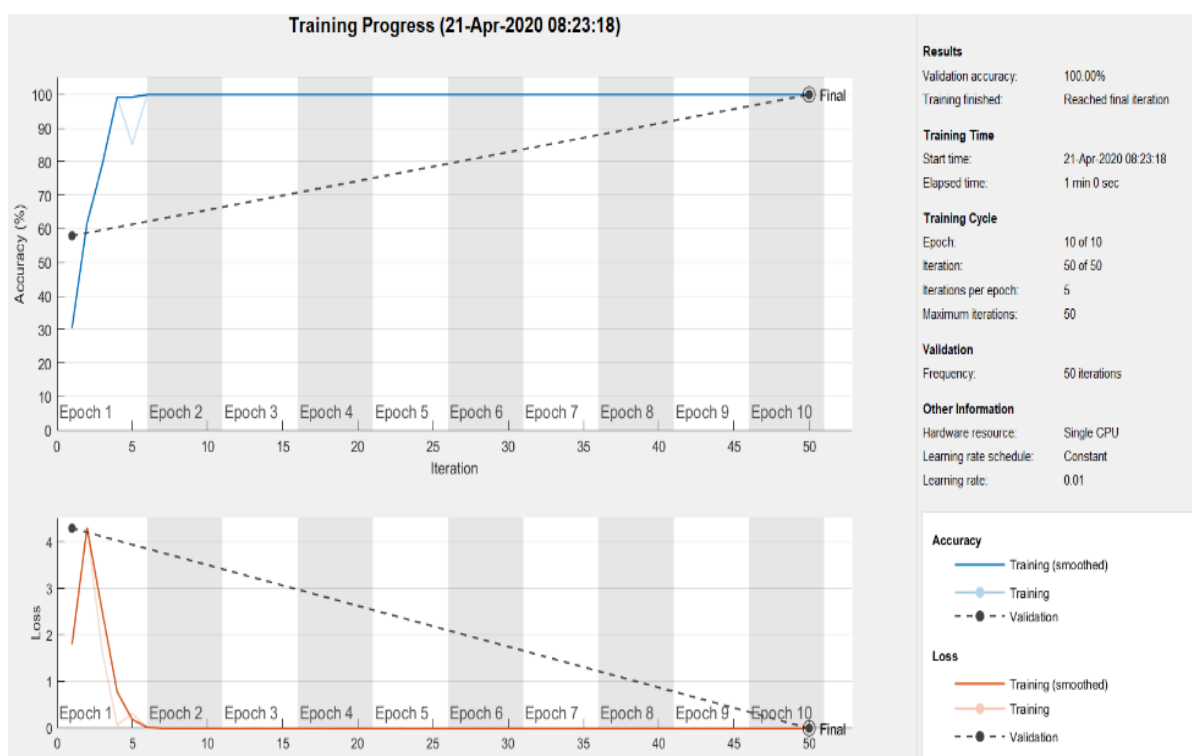


Рисунок 3.6 - Тренувальний процес CNN

Отримані наступні значення параметрів:

- Validation accuracy – 100.00%,

- Epoch – 10 of 10;
- Iteration – 50 of 50;
- Frequency – 50 iterations;
- Iterations per epoch – 5;
- Maximum iterations – 10;
- Frequency – 50 iterations;
- Learning rate schedule – constant;
- Learning rate – 0.01.

Її використання підвищило точність валідації більш ніж у два рази. Тому саме цю конволюційну мережу і було обрано для розв'язку задачі, що полягає у розпізнаванні об'єктів заданого типу.

4 ПРАКТИЧНЕ ЗАСТОСУВАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

4.1 Постановка практичної задачі щодо застосування результатів досліджень

Практична задача, яка вирішується у даній атестаційній роботі, полягає у розпізнаванні об'єктів заданого типу на зображенні за допомогою CNN. Основна задача розбивається на дві незалежні підзадачі, які вирішуються за допомогою однієї і тієї ж CNN:

- розпізнавання певного типу об'єкту із заданої множини класів об'єктів різних типів;
- розпізнавання об'єкту певного класу із заданої множини класів об'єктів одного типу.

Для реалізації першої задачі обрано предметну область «Фрукти», тому що ці об'єкти в великій кількості використовуються та споживаються населенням України. Пропонована задача розпізнавання фруктів на зображеннях є актуальною та цікавою для логістики, магазинів, як спеціалізованих та і універсамів, сільгоспвиробників тощо.

Таким чином, необхідно здійснити розпізнавання певного класу фруктів із сформованої множини різних видів фруктів. В роботі розглядаються такі класи фруктів: «яблука», «апельсин», «банан», «виноград».

Обрання цих об'єктів пов'язано з великими об'ємами споживань населенням України.

Припускаємо, що пропонована в даній роботі архітектура CNN повинна допускати поширення множини видів фруктів з метою їх подальшого розпізнавання на зображеннях без втрати точності розпізнавання.

Реалізація другої задачі має відмінності від попередньої в тому сенсі, що усі об'єкти, які необхідно розпізнавати, належать до одного класу. В якості предметної області в даній роботі пропонується предметна область «Яблука різних сортів».

Вибір задачі даного класу обумовлений тим, що яблука займають перше місце споживання серед фруктів населенням України через вирощування їх в усіх областях країни, великою кількістю сортів, тривалістю зберігання, широким використанням процесів консервування, сушки, відносно низькими цінами та доступністю їх всім верствам населення.

Для розпізнавання об'єкту заданого типу використовується тренувальний набір даних [41], що включає 4 класи: Apple, Banana, Grape, Orange. Клас Apple містить 654 зображення формату JPG розміром 100 x 100 пікселів, Banana містить 490 зображень формату JPG розміром 100 x 100 пікселів, Grape 492 зображення формату JPG розміром 100 x 100 пікселів, Orange 479 зображення формату JPG розміром 100 x 100 пікселів. Тестовий набір даних складається з 4 класів: Apple, Banana, Grape, Orange. Клас Apple містить 401 зображення формату JPG розміром 100 x 100 пікселів, Banana містить 166 зображень формату JPG розміром 100 x 100 пікселів, Grape містить 164 зображення формату JPG розміром 100 x 100 пікселів, Orange містить 160 зображень формату JPG розміром 100 x 100 пікселів.

Для розпізнавання сортів яблук використовується тренувальний набір даних [41], що включає 4 класи: Apple Braeburn, Apple Golden, Apple Granny Smith, Apple Red Delicious. Клас Apple Braeburn містить 164 зображення формату JPG розміром 100 x 100 пікселів, Apple Golden містить 164 зображень формату JPG розміром 100 x 100 пікселів, Apple Granny Smith 164 зображення формату JPG розміром 100 x 100 пікселів, Apple Red Delicious 166 зображень формату JPG розміром 100 x 100 пікселів. Тестовий набір даних складається з 4 класів: Apple Braeburn, Apple Golden, Apple Granny Smith, Apple Red Delicious. Клас Apple Braeburn містить 164 зображення

формату JPG розміром 100 x 100 пікселів, Apple Golden містить 164 зображень формату JPG розміром 100 x 100 пікселів, Apple Granny Smith містить 164 зображення формату JPG розміром 100 x 100 пікселів, Apple Red Delicious містить 166 зображень формату JPG розміром 100 x 100 пікселів.

4.2 Обґрунтування та вибір програмних засобів для реалізації додатку

Для реалізації додатку використовуватиметься Matlab.

Платформа програмування Matlab має численні переваги перед іншими методами чи мовами. Функціональність Matlab можна значно розширити, додавши панелі інструментів. Це набори конкретних функцій, які забезпечували більш спеціалізовану функціональність. Ці функції роблять мову програмування дуже ефективною для впровадження глибинного навчання .

Основними особливостями Matlab є:

- сумісність: підтримує сумісність з іншими фреймворками глибинного навчання з відкритим кодом, такими як відкритий обмін нейронної мережі (ONNX). Користувачі можуть вибрати Matlab для розміщення можливостей та попередньо встановлених цілей та додатків, які не доступні іншими мовами програмування;
- область попередньої обробки: надає можливість для активного попереднього опрацювання наборів даних із доменними програмами для аудіо, відео та зображень. Користувачі можуть візуалізувати, перевіряти та виправляти проблеми перед навчанням програми Deep Network Designer для створення складних мережевих архітектур або модифікації навчальних мереж для навчання передачі;
- мультипрограмне розгортання: може використовувати моделі глибинного навчання будь-де, включаючи обчислену уніфіковану архітектуру пристрою (CUDA), код C, корпоративні системи або хмару. Це дає велику продуктивність, коли користувач може виробляти код, який

підтримує оптимізовані бібліотеки, такі як Intel (MKL-DNN), NVIDIA (TensorRT, cuDNN) і ARM (ARM Compute Library);

- панель інструментів глибинного вивчення: Deep Learning Toolbox реалізує фреймворк для створення та виконання глибинних нейронних мереж за допомогою алгоритмів, навчених моделей та додатків. Користувач може застосовувати CNN та LSTM мережі для забезпечення класифікації та регресії даних зображень, часових рядів та текстових даних. Додатки та графіки допомагають користувачам візуалізувати активації, редагувати мережеві архітектури та контролювати хід підготовки. Для скромних навчальних наборів користувач може керувати трансферним навчанням за допомогою навчених глибинних мережевих моделей та моделей, що впливають від TensorFlow-Keras та Caffe. Для подальшого навчання великим наборам даних користувачі можуть призначити обчислення та дані за межами багатоядерних процесорів та графічних процесорів на робочому столі або масштабувати до кластерів та хмар, включаючи примірники GPU Amazon EC2 P2, P3 та G3;

- MatConvNet. MatConvNet є процесом виконання CNN для Matlab. MatConvNet дозволяє швидко прототипувати різні архітектури CNN, і в той же час він підтримує ефективні обчислення на процесорі та графічному процесорі, забезпечуючи здійснення складних моделей на великих наборах даних.

- Cuda-Convnet. Ця бібліотека є захищеною C++/CUDA реалізацією CNN [42].

Перевагами цієї мови:

- дуже велика (і зростаюча) база даних із вбудованими алгоритмами для обробки зображень та програм комп'ютерного зору;

- дозволяє протестувати алгоритми негайно без перекомпіляції. Можна вести щось у командному рядку та одразу побачити результати, що значно полегшує розробку алгоритму;

- можливість читання в широкому спектрі як загальних, так і доменних форматів зображень;
- можливість викликати зовнішні бібліотеки;
- можливість обробки як зображень, так і відео;
- можливість автоматичного генерування коду С, використовуючи Matlab Coder, для великої (і зростаючої) підмножини обробки зображень та математичних функцій, які потім можна використовувати в інших середовищах, таких як вбудовані системи або як компонент іншого програмного забезпечення [43].

4.3 Реалізація програмної системи

Головне вікно програмної системи наведено на рисунку 4.1.

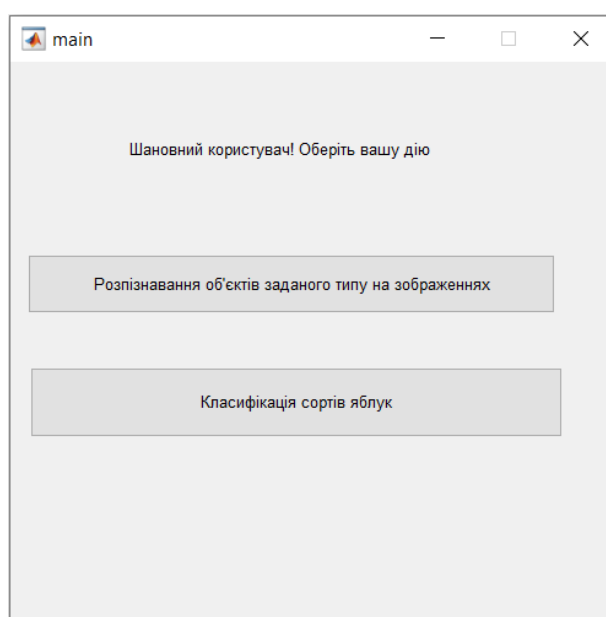


Рисунок 4.1 – Головне вікно програмної системи

Користувач обирає, що він хоче зробити: розпізнати об'єкт заданого типу на зображеннях або вирішити більш конкретну задачу на множині об'єктів одного обраного типу, наприклад, визначити сорт яблук. Після вибору першої дії відкривається вікно, де користувач завантажує зображення. Результат класифікації об'єктів показано на рисунку 4.2.

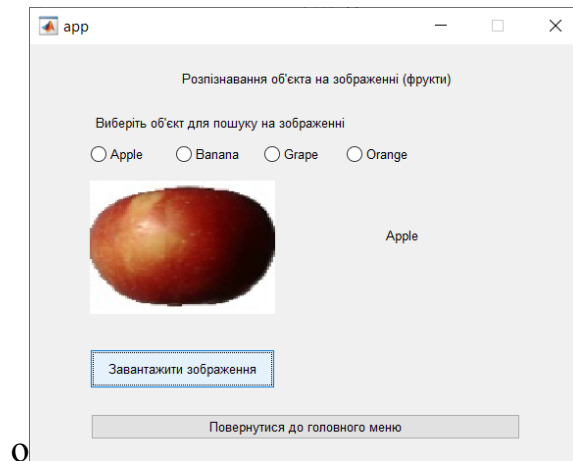


Рисунок 4.2 – Результат класифікації об'єкту

Для перевірки, чи містить зображення потрібний об'єкт, користувач вибирає об'єкт для пошуку з переліку. Якщо зображення містить потрібний об'єкт, то виводиться модальне вікно, що об'єкт знайдений.

На рисунку 4.3 наведено результат пошуку об'єкту на зображенні.

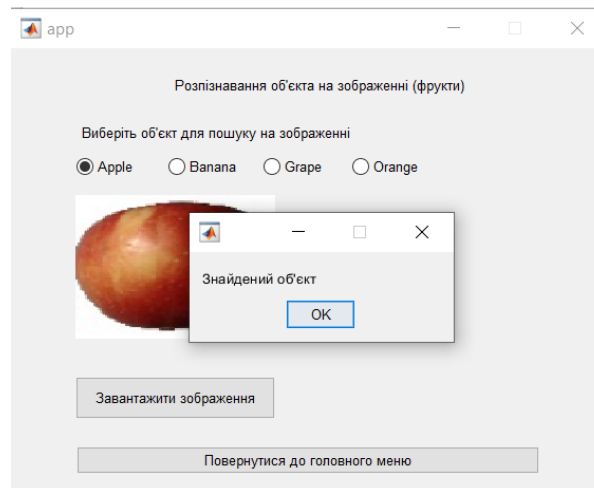


Рисунок 4.3 – Результат пошуку об'єкту на зображенні

При відсутності шуканого об'єкту на зображенні виводиться модальне вікно, що об'єкт не знайдений. На рисунках 4.4-4.6 наведено результат пошуку об'єкта Banana, Grape, Orange на зображенні.

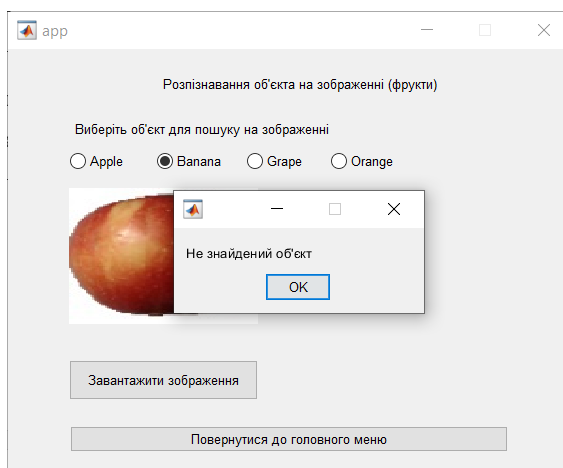


Рисунок 4.4 - Результат пошуку об'єкта Банана на зображенні

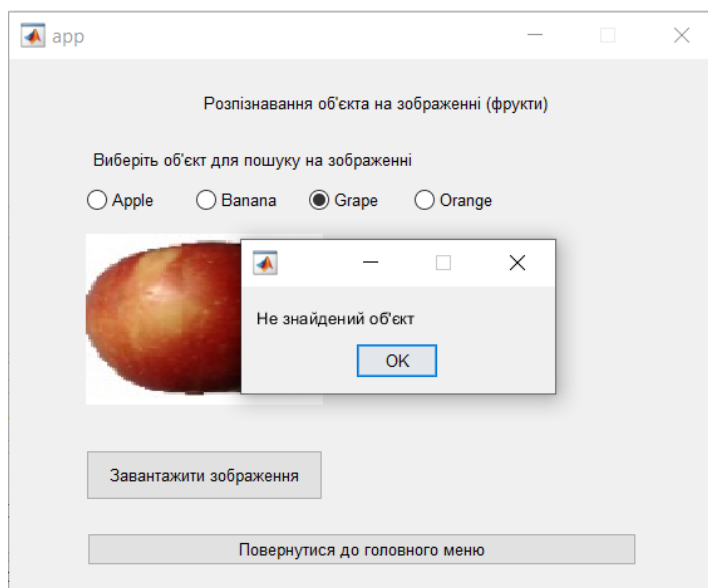


Рисунок 4.5 - Результат пошуку об'єкта Grape на зображенні

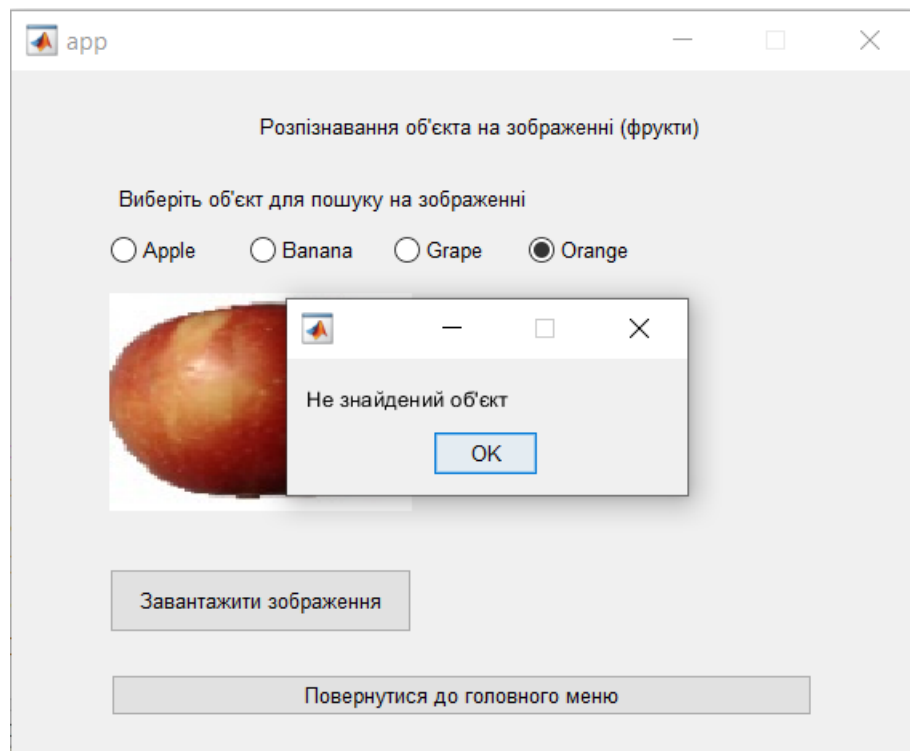


Рисунок 4.6 - Результат пошуку об'єкта Orange на зображенні

При виборі в головному вікні другої дії з'являється наступний вікно, де наведені зображення яблук різних сортів та вільне місце для завантаження зображення яблука невідомого сорту.

На рисунку 4.7 наведено вікно програмної системи при вирішенні задачі розпізнавання на множині об'єктів одного типу та віднесення конкретного об'єкта до одного з відомих класів. В якості предметної області використовується класифікація яблук за їх сортами та віднесення конкретного зображення до певного сорту яблук.

Слід зазначити, що в системі передбачено як поширення бази сортів яблук за рахунок поповнення навчальної множини об'єктів (яблук), так і підключення навчальних множин и зображень інших фруктів та овочів з метою їх класифікації по сортах.

Система такого типу безумовно буде корисною для застосування в різних галузях, пов'язаних із використанням фруктів та овочів, наприклад,

у фермерських господарствах, сільському господарстві в цілому, кулінарії, ресторанному бізнесі, у продуктивних супермаркетах тощо.

Ще одним перспективним напрямом використання отриманих результатів може бути реалізація пропонованої системи для мобільних пристроїв та смартфонів задля онлайн розпізнавання сорту конкретного яблука або іншого певного фрукта чи овоча.

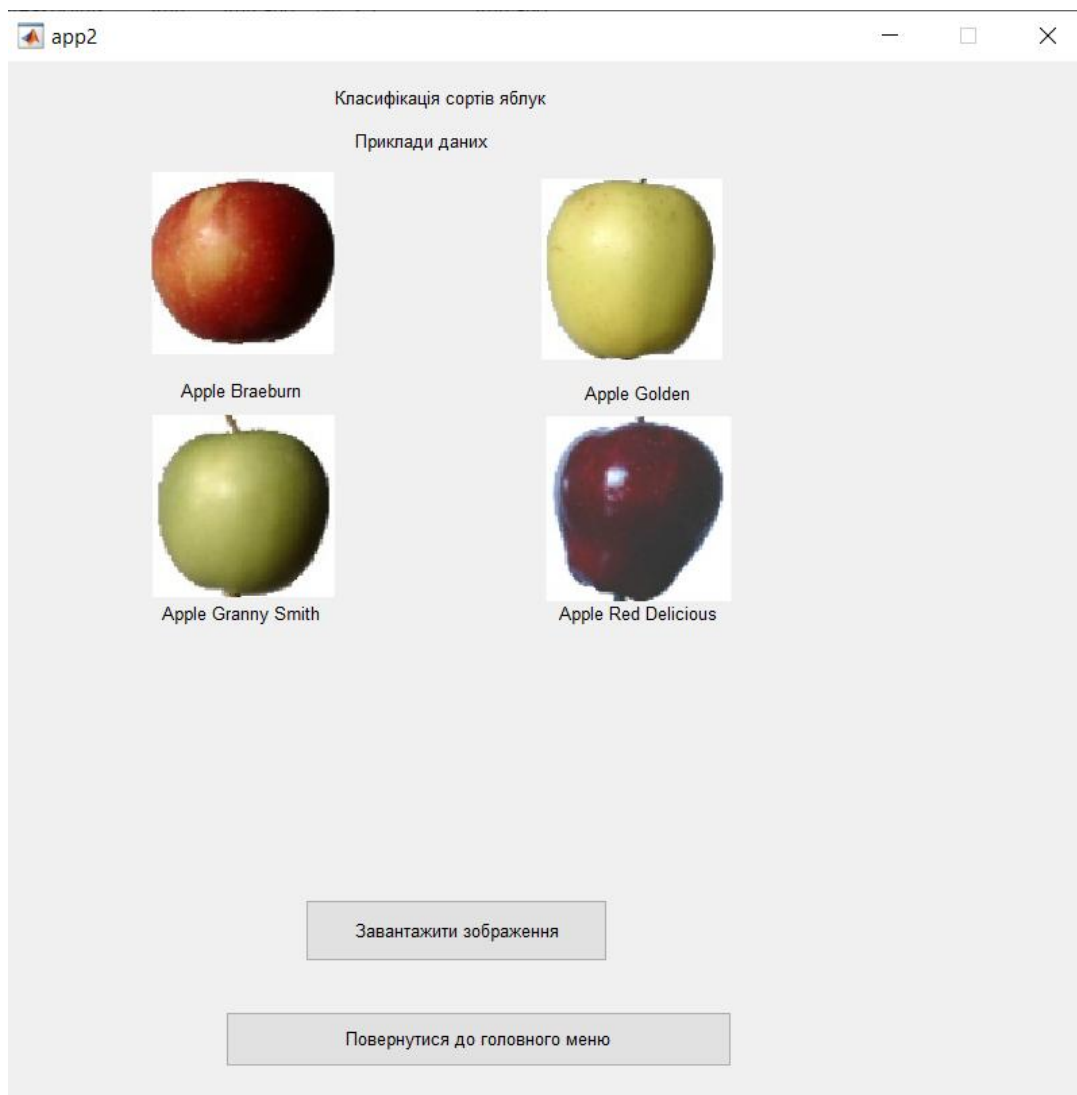


Рисунок 4.7 – Вікно системи для класифікації сортів яблук

При завантаженні зображення результат класифікації з'являється праворуч від екрану. На рисунку 4.8 наведено результат класифікації сорту яблук.

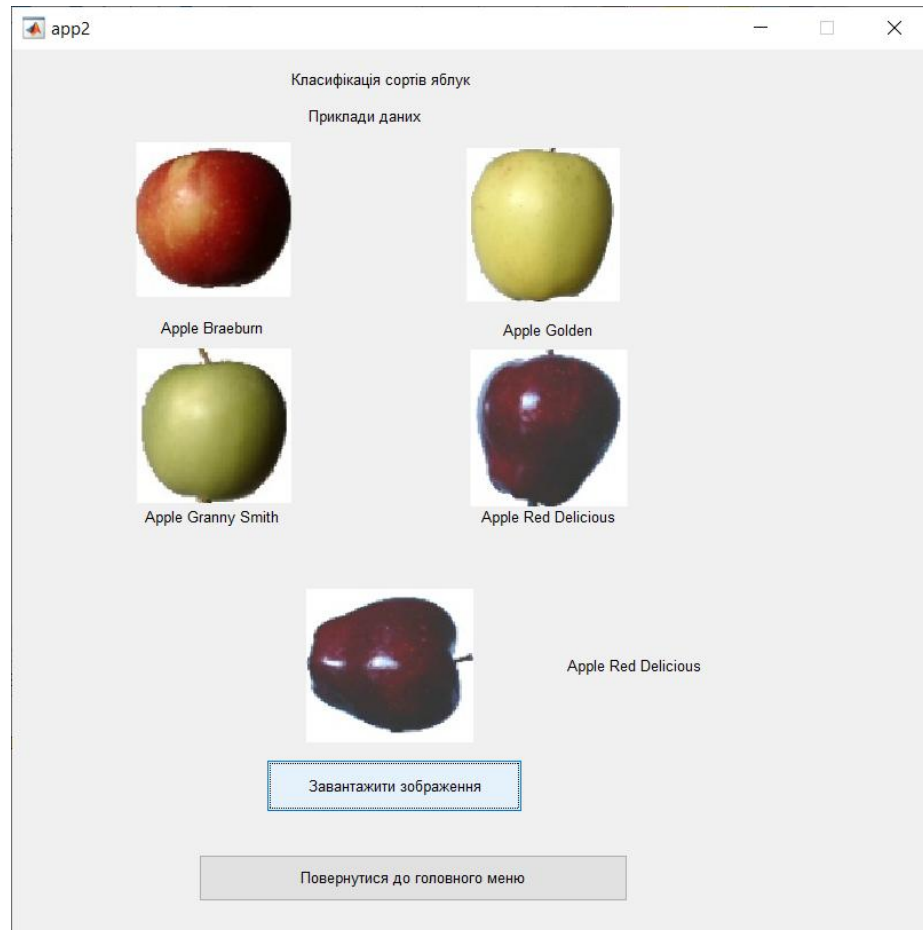


Рисунок 4.8 – Результат класифікації сорту яблук

ВИСНОВКИ

Дана магістерська атестаційна робота присвячена дослідженню властивостей глибоких нейронних мереж, їх основних переваг та особливостей архітектури, предметних галузей та задач, які доцільно вирішувати за їх допомогою. В роботі проаналізовано основні можливості та поведінка конволюційної нейронної мережі при вирішенні задач розпізнавання об'єктів заданого типу на зображеннях. В результаті виконання атестаційної роботи вирішено певну низу задач та отримано такі результати:

- проведено аналіз науково-технічних публікацій та даних Інтернет-джерел в області глибинного навчання та глибинних нейронних мереж;
- проаналізовано основні типи задач розпізнавання об'єктів на зображеннях та методи їх вирішення;
- обґрунтовано вибір об'єктів розпізнавання для проведення досліджень;
- здійснено вибір та формування наборів даних, навчальної множини об'єктів;
- проведено обґрунтування та вибір архітектури конволюційної нейронної мережі щодо вирішення поставлених задач;
- здійснено вибір програмних засобів для перевірки працездатності побудованої моделі конволюційної нейронної мережі;
- запропоновано узагальнену структуру програмного прототипу системи розпізнавання об'єктів заданого типу;
- досліджено можливості розширення функціоналу прототипу системи розпізнавання об'єктів заданого типу.

Результати роботи також відображені у тезах доповіді, опублікованої у Матеріалах 24-го Міжнародного молодіжного форуму «Радіоелектроніка і молодь у XXI столітті [45].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Heaton J. Artificial Intelligence for Humans, Volume 3: Neural Networks and Deep Learning. Heaton Research Inc., 2015. 268 p.
2. Sumathi S., Surekha P. Computational Intelligence Paradigms. Theory and Applications using Matlab. CRC Press, Taylor & Francis Group, LLC, 2010. 834 p.
3. Aggarwal Charu C. Neural Networks and Deep Learning. A Textbook. Springer, 2018. 512 p.
4. Бодянский Е.В., Рябова Н.В., Золотухин О.В. Многослойная адаптивная нечеткая вероятностная нейронная сеть в задачах классификации текстовых документов. *Радиоэлектроника, информатика, управление*. 2015. №1(32). С. 39-45
5. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение: пер. с англ. 2-е изд., испр. М.: ДМК Пресс, 2018. 652 с.
6. Шапиро Л., Стокман Дж. Компьютерное зрение пер. с англ. 3-е изд. М.: БИНОМ. Лаборатория знаний, 2015. 763 с.
7. Таненбаум Э. Грокаем глубокое обучение. СПб.: Питер, 2019. 352 с.
8. Analysis of Dropout. P. Galeone's blog. URL: <https://pgaleone.eu/deep-learning/regularization/2017/01/10/anaysis-of-dropout/>
(дата звернення 10.04.2020)
9. Метод дропаут. Кодкам. URL: <https://artemis.codecamp.ru/courses/deep-learning-convolutional-networks/regularization-techniques-deep-learning/dropout-regilarisation-method>
(дата звернення 10.04.2020)
10. Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, Qing He A Comprehensive Survey on Transfer Learning Arxiv 2019, [27] с. (Препринт. arXiv:1911.02685)

11. Srivastava N, Hinton G., Krizhevsky A, Sutskever I., Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 2014. Vol. 15, No. 56, P. 1929–1958.
12. Dropout in (Deep) Machine learning. *Medium*. URL: <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5> (дата звернення 10.04.2020)
13. Khan A. et al. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*. 2020. P.1–6
14. Rezk Nesma M., Purnaprajna M., Nordström T., Ul-Abdin Z. Recurrent Neural Networks: An Embedded Computing Perspective. *IEEE Access*. 2020. Vol.20. P.57967–57996.
15. Humayun Sulehria H.K., Zhang Y. Hopfield Neural Networks. A Survey *AIKED'07: Proceedings of the 6th Conference on 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*. 2007. Vol.6. P.125–130.
16. Нейронная сеть Хопфилда на пальцах. *Хабр*. URL: <https://habr.com/ru/post/301406/> (дата звернення 10.04.2020)
17. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*. 2017. Vol. 28, No. 10. P. 2222-2232.
18. Long short-term memory. *Wikipedia*. URL: https://en.wikipedia.org/wiki/Long_short-term_memory (дата звернення 10.04.2020)
19. Szegedy C. et al. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*. 2015. Vol. 57, No. 3. P. 1–9.
20. Zeiler M.D., Fergus R. Visualizing and Understanding Convolutional Networks. *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science. ECCV 2014. Lecture Notes in Computer Science*. 2014. Vol.8689 P. 818–833.

21. Khan, A., Sohail, A., Zahoora, U. et al. A survey of the recent architectures of deep convolutional neural networks. *ArtifIntell Rev.* 2020. Vol.53
22. Lowe D.G. Distinctive image features from scale-invariant keypoints *Int. J. of Comput. Vision.* 2004 , Vol. 60, No. 2. P. 91–110.
23. Dalal N, Triggs B. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* 2005. Vol.1. P. 886–893.
24. Lienhart R., Maydt J. An extended set of Haar-like features for rapid object detection. *Proceedings. International Conference on Image Processing.* 2002
25. Introduction to FAST (Features from Accelerated Segment Test). *Medium.* URL: <https://medium.com/data-breach/introduction-to-fast-features-from-accelerated-segment-test-4ed33dde6d65> (дата зверення 10.04.2020)
26. Introduction to Harris Corner Detector. *Medium.* URL: <https://medium.com/data-breach/introduction-to-fast-features-from-accelerated-segment-test-4ed33dde6d65> (дата зверення 10.04.2020)
27. Introduction to SIFT(Scale Invariant Feature Transform). *Medium.* URL: <https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40> (дата зверення 10.04.2020)
28. Girshick R., Donahue J., Darrell T., Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition.* 2014 P.580 – 588.
29. Zhong-Qiu Z., Zheng, Peng Z., Shou-Tao X., Xindong W. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems.* 2019. P. 1-21
30. Rectified Linear Unit (ReLU) layer. *Matlab.* URL: <https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.relu.html> (дата звернення 10.04.2020)

31. Max pooling layer. *Matlab.* URL: <https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.maxpooling2dlayer.html> (дата звернення 10.04.2020)
32. Fully connected layer. *Matlab.* URL: <https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.fullyconnectedlayer.html> (дата звернення 10.04.2020)
33. 2-D convolutional layer. *Matlab.* URL: <https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.convolution2dlayer.html> (дата звернення 10.04.2020)
34. Batch normalization layer. *Matlab.* URL: <https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.batchnormalizationlayer.html> (дата звернення 10.04.2020)
35. Classification output layer. *Matlab.* URL: <https://www.mathworks.com/help/deeplearning/ref/classificationlayer.html> (дата звернення 10.04.2020)
36. Options for training deep learning neural network. *Matlab.* URL: <https://www.mathworks.com/help/deeplearning/ref/trainingoptions.html> (дата звернення 10.04.2020)
37. Алгоритм обратного распространения ошибки. *Aiportal.* URL: <http://www.aiportal.ru/articles/neural-networks/back-propagation.html> (дата звернення 10.04.2020)
38. Голиков И.Сверточная нейронная сеть, часть 2: обучение алгоритмом обратного распространения ошибки. *Хабр.* URL: <https://habr.com/ru/post/348028/> (дата звернення 10.04.2020)
39. Gradient descent for dummies. *Medium.* URL: <https://medium.com/@lankinen/gradient-descent-for-dummies-8adb34123a54> (дата звернення 10.04.2020)
40. Smith L.N. Cyclical Learning Rates for Training Neural Networks. *IEEE Winter Conference on Applications of Computer Vision (WACV) 2017 P. 464-472*

41. Fruits 360 dataset. *Kaggle*.
URL:<https://www.kaggle.com/moltean/fruits> (дата звернення 10.04.2020)
42. Matlab features. *Educba*. URL: <https://www.educba.com/matlab-features/> (дата звернення 10.04.2020)
43. Matlab advantages. *Educba*.
URL:<https://www.educba.com/advantages-of-matlab/> (дата звернення 10.04.2020)
44. Косолапов К.С. Порівняльний аналіз нейромережевих методів в задачах класифікації об'єктів: тез. докл. 24-го Міжнародного молодіжного форуму «Радіоелектроніка і молодь у XXI столітті». - Харків: ХНУРЕ, 2020. Том 6, 2 с.