

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Штучного інтелекту  
(повна назва)

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

рівень вищої освіти другий (магістерський)

Дослідження ефективності існуючих нейронних мереж для  
розпізнавання одягу e-commerce системи  
(тема)

Виконав:  
студент 2 курсу, групи СШМ-21-1  
Борисенко В.О.  
(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  
(повна назва спеціалізації)

Керівник проф. Смеляков К.С.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

В.О. Філатов  
(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)  
Кафедра Штучного інтелекту  
(повна назва)  
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 КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Борисенку Владиславу Олеговичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження ефективності нейронних мереж для розпізнавання одягу e-commerce системи

затверджена наказом університету від 31 березня 2023 р. № 306Ст

2. Термін подання студентом роботи до екзаменаційної комісії 23 травня 2023 р.

3. Вихідні дані до роботи Науково-технічні публікації, відкриті джерела в мережі Інтернет, відкриті набори даних

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

1) Аналіз предметної області та постановка задачі дослідження

2) Дослідження існуючих нейронних мереж для розпізнавання одягу

3) Методологія дослідження

4) Експериментальні результати та аналіз

5) Практична реалізація

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) \_\_\_\_\_

Рисунок 1 – Графік залежності втрат при навчанні від коефіцієнту швидкості навчання, Рисунок 2 – Новий графік залежності втрат при навчанні від коефіцієнту швидкості навчання, Рисунок 3 – Графік втрат, Рисунок 4 – Графік зміни точності моделі, Рисунок 5 – Ілюстрація до дилеми зміщення та дисперсії, Рисунок 6 – Дані класифікатора системи датасету для топ-3 та топ-5 результатів, Рисунок 7 – Приклад передбачень моделлю над зображеннями датасету, Рисунок 8 – Приклад передбачень моделлю над зображеннями датасету, Рисунок 9 – Графічне представлення середніх втрат від кількості ітерацій.

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	03.04.2023	виконано
2	Вивчення предметної області дослідження	03.04.2023– 08.04.2023	виконано
3	Дослідження літератури	09.04.2023– 14.04.2023	виконано
4	Вивчення проблеми, що потребує рішення	15.04.2023– 16.04.2023	виконано
5	Формування постановки задачі дослідження	17.04.2023– 19.04.2023	виконано
6	Аналіз існуючих підходів	20.04.2023– 21.05.2023	виконано
7	Концептуальне проектування методу рішення задачі	22.04.2023– 25.04.2023	виконано
8	Програмна реалізація та тестування	26.04.2023– 05.05.2023	виконано
9	Написання пояснювальної записки	06.05.2023– 11.05.2023	виконано
10	Попередній захист	17.05.2023	виконано
11	Захист перед ЕК	23.05.2023	виконано

Дата видачі завдання 3 квітня 2023 р.

Студент ВБ  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Смеляков К.С.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 58 с., 9 рис., 4 табл., 1 дод., 30 джерел.

### ГЛИБИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, РОЗПІЗНАВАННЯ ОБРАЗІВ, CNN, RESNET34, YOLO3

Об'єкт дослідження – процеси та методи використання нейронних мереж, для розпізнавання та класифікації одягу в системах електронної комерції.

Предмет дослідження – ефективність та точність нейронних мереж у задачах розпізнавання та класифікації одягу в контексті їх використання в системах електронної комерції.

Мета роботи – дослідження, порівняння та оцінка ефективності нейронних мереж для розпізнавання та класифікації одягу в e-commerce системах з метою виявлення найбільш оптимального рішення.

Дослідження зосереджується на вивченні та порівнянні ефективності нейронних мереж YOLO3 та ResNet34 для автоматичного розпізнавання та класифікації одягу в системах електронної комерції.

Робота оцінює потенціал цих моделей з точки зору точності, швидкості та використання ресурсів, при цьому наголошує на важливість автоматизованого розпізнавання одягу для покращення ефективності каталогізації товарів та персоналізації рекомендацій в e-commerce системах.

В роботі демонструється процес тренування, налаштування та оцінювання цих моделей, а також надається пропозиція по їх впровадженню в реальні e-commerce системи. В результаті, робота пропонує цінні вказівки для вибору найбільш ефективної моделі для конкретних систем електронної комерції.

## **ABSTRACT**

Explanatory note: 58 p., 9 fig., 4 tabl., 1 ann., 30 sources.

CNN, DEEP LEARNING, NEURAL NETWORKS, PATTERN RECOGNITION, RESNET34, YOLO3

The object of research is the processes and methods of using neural networks to recognize and classify clothes in e-commerce systems.

The subject of research is the efficiency and accuracy of neural networks in the tasks of recognizing and classifying clothes in the context of their use in e-commerce systems.

The purpose of the study is to investigate, compare and evaluate the effectiveness of neural networks for recognizing and classifying clothes in e-commerce systems in order to identify the most optimal solution.

The research focuses on studying and comparing the effectiveness of YOLO3 and ResNet34 neural networks for automatic recognition and classification of clothing in e-commerce systems.

The paper evaluates the potential of these models in terms of accuracy, speed, and resource utilization, while emphasizing the importance of automated clothing recognition for improving the efficiency of product cataloging and personalizing recommendations in e-commerce systems.

The paper demonstrates the process of training, tuning, and evaluating these models, and provides a proposal for their implementation in real e-commerce systems. As a result, the paper offers valuable guidance for choosing the most effective model for specific e-commerce systems.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	8
Вступ.....	9
1 Аналіз предметної області та постановка задачі дослідження.....	10
1.1 Аналіз предметної області.....	10
1.2 Аналіз існуючих методів розпізнавання одягу з використанням нейронних мереж .....	11
1.3 Постановка задачі.....	12
2 Дослідження існуючих нейронних мереж для розпізнавання одягу .....	14
2.1 Пояснення принципу роботи нейронних мереж.....	14
2.2 Огляд існуючих моделей нейронних мереж .....	16
2.3 Аналіз їх ефективності та точності .....	17
2.4 Оцінка швидкодії та продуктивності моделей.....	17
2.5 Ідентифікація переваг та недоліків кожної моделі.....	18
3 Методологія дослідження .....	23
3.1 Вибір датасету для навчання та тестування .....	23
3.2 Порівняння різних моделей нейронних мереж.....	24
3.3 Параметризація та тренування моделей .....	25
3.4 Метрики оцінки ефективності розпізнавання одягу .....	27
4 Експериментальні результати та аналіз.....	31
4.1 Вибір датасету та моделі .....	31
4.2 Проектування нейронних мереж.....	31
4.3 Оцінка роботи мережі.....	36
4.4 Аналіз отриманих результатів та вибір найефективнішої моделі .....	47
5 Практична реалізація .....	49
5.1 Інтеграція найефективнішої моделі у e-commerce систему.....	49
5.2 Навчання моделі ResNet34 .....	49
5.3 Інтеграцію у e-commerce систему .....	50
5.4 Оцінка результатів інтеграції моделі .....	51

5.5 Доповнення та розвиток системи .....	52
Висновки .....	53
Перелік джерел посилання .....	54
Додаток А Відомість кваліфікаційної роботи .....	58

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

AI – Artificial Intelligence – штучний інтелект;

CNN – Convolutional Neural Networks – згорткові нейронні мережі;

DL – Deep Learning – глибоке навчання;

E-commerce – електронна комерція;

FPS – Frames Per Second – кадрів в секунду;

ImageNet – велика база даних візуальних об'єктів, яка часто використовується для тренування моделей глибокого навчання;

ML – Machine Learning – машинне навчання;

NLP – Natural Language Processing – обробка природної мови;

ResNet – Residual Network – архітектура згорткової нейронної мережі;

RNN – Recurrent Neural Networks – рекурентні нейронні мережі;

SOTA – State-of-the-Art – найкращий поточний показник;

YOLO – You Only Look Once – метод розпізнавання об'єктів на зображеннях.

## ВСТУП

У сучасному цифровому світі електронна комерція (e-commerce) займає все більш значне місце, надаючи споживачам можливість зручно та ефективно придбати товари та послуги в онлайн-середовищі. Проте, однією з ключових проблем, з якими стикаються споживачі при здійсненні покупок в електронному форматі, є візуальна оцінка одягу на основі фотографій. Зрозуміння, як даний одяг виглядає насправді, може бути вирішальним чинником при прийнятті рішення про покупку.

Одним з можливих рішень для цієї проблеми є використання нейронних мереж для розпізнавання одягу на зображеннях в електронних комерційних системах. Нейронні мережі є потужними інструментами машинного навчання, які можуть виявляти та класифікувати об'єкти з високою точністю. Проте, важливо дослідити та порівняти ефективність існуючих нейронних мереж для розпізнавання одягу в e-commerce системах з метою вибору найкращих підходів та моделей.

Метою даної роботи є проведення дослідження ефективності існуючих нейронних мереж для розпізнавання одягу в e-commerce системах. Зокрема, робота має на меті визначення найоптимальніших архітектур нейронних мереж та методів тренування, які забезпечують високу точність та надійність розпізнавання одягу на зображеннях. Дослідження буде базуватися на аналізі різних існуючих моделей та їхніх результатів.

У даній роботі буде використано систематичний підхід до дослідження ефективності існуючих нейронних мереж для розпізнавання одягу в e-commerce системах. Результати дослідження сприятимуть розширенню знань у галузі машинного навчання та електронної комерції, а також забезпечать практичну користь для розробників та операторів e-commerce систем.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

## 1.1 Аналіз предметної області

Розпізнавання одягу в електронних комерційних системах є актуальною та важливою задачею, яка вимагає високої точності та швидкодії для забезпечення задоволення потреб споживачів. Для вирішення цієї задачі нейронні мережі виявилися потужним інструментом, оскільки вони можуть автоматично вчитися з великого обсягу даних та виявляти складні залежності в зображеннях.

В області розпізнавання образів застосування нейронних мереж вже привело до значних досягнень, зокрема в розпізнаванні об'єктів на зображеннях. Однак, розпізнавання одягу має свої виклики через велику варіативність форм, кольорів, текстур та контекстуального значення одягу. Ці особливості вимагають розробки спеціалізованих моделей та архітектур нейронних мереж, що враховують специфіку розпізнавання одягу.

Огляд літератури вказує на існування різних підходів до розпізнавання одягу в e-commerce системах з використанням нейронних мереж. Деякі з них базуються на конволюційних нейронних мережах (Convolutional Neural Networks, CNN), які добре себе зарекомендували у багатьох задачах розпізнавання образів. Інші підходи використовують комбінацію різних типів мереж, таких як глибокі згорткові нейронні мережі (Deep Convolutional Neural Networks, DCNN), рекурентні нейронні мережі (Recurrent Neural Networks, RNN) та генеративні приховані моделі (Generative Adversarial Networks, GAN), для досягнення більш точного розпізнавання та генерації зображень одягу.

Дослідження ефективності існуючих нейронних мереж для розпізнавання одягу в e-commerce системах полягає в порівнянні та аналізі різних моделей та методів, які були застосовані в цій галузі. Це включає

вивчення різних архітектур нейронних мереж, їхніх характеристик та можливостей, а також методів тренування та оптимізації моделей. Крім того, буде проведений аналіз наборів даних, що використовуються для навчання та тестування моделей, з метою розуміння їхньої репрезентативності та потенційних обмежень.

В результаті проведення дослідження очікується визначити найефективніші моделі та методи для розпізнавання одягу в e-commerce системах, зокрема ті, які забезпечують найвищу точність та швидкодію. Такий аналіз може сприяти покращенню процесу візуального оцінювання товарів в електронних магазинах та підвищити рівень задоволення споживачів. Крім того, результати дослідження можуть послужити основою для подальшого вдосконалення нейронних мереж та розробки нових алгоритмів розпізнавання одягу в e-commerce системах.

## 1.2 Аналіз існуючих методів розпізнавання одягу з використанням нейронних мереж

Розпізнавання одягу в e-commerce системах вимагає використання потужних і ефективних алгоритмів обробки зображень. Нейронні мережі є одним із найбільш успішних підходів до розв'язання цієї задачі. Наступні методи розпізнавання одягу на основі нейронних мереж були розглянуті у літературі та дослідженнях:

- застосування згорткових нейронних мереж (CNN): Згорткові нейронні мережі є популярним вибором для розпізнавання об'єктів, включаючи одяг. Вони використовуються для виявлення та виділення важливих ознак одягу з вхідних зображень;

- використання передбачувальних моделей (generative models): Цей підхід використовує генеративні моделі, такі як автокодувальні згорткові мережі або генеративно-протилежні мережі (GAN), для створення нових зображень одягу. Ці моделі можуть бути використані для створення

синтетичних даних для тренування моделей розпізнавання одягу;

- використання пре-тренованих моделей: Існують пре-треновані моделі нейронних мереж, такі як VGG, ResNet, Inception, які були навчені на великих наборах даних, включаючи зображення одягу. Ці моделі можуть бути використані для передбачення класу або характеристик одягу;

- використання рекурентних нейронних мереж (RNN): Рекурентні нейронні мережі використовуються для аналізу послідовностей даних, таких як описи або коментарі до зображень одягу. Вони можуть допомогти врахувати контекст та залежності між словами або символами в текстовому описі одягу.

Ці методи представляють різні підходи до розпізнавання одягу з використанням нейронних мереж. У цій дипломній роботі ми проведемо дослідження, порівняємо їх ефективність та визначимо найбільш підходящі моделі для розпізнавання одягу в e-commerce системах.

### 1.3 Постановка задачі

Метою даної роботи в дослідження та оцінка ефективності існуючих нейронних мереж для розпізнавання одягу в e-commerce системах. Для досягнення цієї мети, необхідно вирішити наступні підзадачі:

- вибір набору даних: Першочергово, потрібно вибрати відповідний набір даних, який відобразить різноманіття одягу, який може зустрічатись в e-commerce системах. Це може включати різні типи одягу, такі як футболки, штани, сукні, взуття, аксесуари та інші, а також різні категорії та стилі. Набір даних повинен бути достатньо репрезентативним для забезпечення надійної оцінки ефективності моделей;

- огляд методів та архітектур нейронних мереж: Потрібно проаналізувати існуючі методи та архітектури нейронних мереж, які застосовуються для розпізнавання одягу. Це включає згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), передбачувальні моделі

та пре-треновані моделі. Розглядатимуться особливості цих методів та архітектур в контексті розпізнавання одягу, зокрема їх здатність до виявлення структури, відносин та контексту між різними частинами одягу;

– розробка та тренування моделей: На основі обраного набору даних та огляду існуючих методів, будуть розроблені та навчені різні моделі нейронних мереж для розпізнавання одягу. Це включатиме настройку гіперпараметрів, вибір оптимальної архітектури мережі та використання відповідних методів тренування, таких як зворотне поширення помилки (backpropagation);

– оцінка ефективності моделей: Розроблені моделі будуть оцінені за допомогою визначених метрик, таких як точність, чутливість, специфічність, F1-показник та інші. Також будуть враховані швидкодія та ресурсомісткість моделей. Порівняння результатів дозволить визначити, які моделі найкраще впораються з завданням розпізнавання одягу в e-commerce системах;

– висновки та рекомендації: На основі проведених досліджень та результатів оцінки ефективності моделей будуть зроблені висновки щодо їх придатності для розпізнавання одягу в e-commerce системах. Також будуть надані рекомендації щодо подальшого вдосконалення та використання нейронних мереж у даній області, включаючи можливі шляхи оптимізації моделей та використання новітніх технологій.

## 2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ОДЯГУ

### 2.1 Пояснення принципу роботи нейронних мереж

Щоб продовжити обговорення використовуваних технологій та процесу створення нейронної мережі, важливо розуміти, як вона працює і який результат вона може дати після тренування. Нейронна мережа – це мультішарова система, яка включає кілька шарів, кожен з яких містить вузли для виконання різних математичних операцій. Мережа працює з матрицями і використовує матрицю числових значень (модель «коефіцієнтів»), яка використовується в математичних розрахунках кілька разів для поступового вдосконалення моделі.

Коли мережа отримує зображення з датасету, вона інтерпретує його як матрицю значень у діапазоні  $[0;255]$ , що в даному випадку розподілені не тільки за пікселями, але й за трьома каналами RGB. Потім мережа використовує матрицю зображення та власну модель для виконання математичних операцій на кожному шарі.

Залежно від архітектури мережі, на різних шарах використовуються так звані активаційні функції, які обчислюють значення цих функцій (forward propagation) за допомогою двох матриць, а потім коригують ваги в моделі (backward propagation). В залежності від активаційної функції, мережа нормалізує матрицю зображення до діапазону  $[0;1]$  або  $[-1;1]$  перед введенням в модель [12][8].

Ці операції повторюються для кожного нового зображення, що в кінцевому підсумку призводить до створення моделі, яка містить оптимальні ваги для виведення ймовірності того, що дане зображення належить до певного класу одягу. За суттю, мережа «налаштовує» ваги у моделі для досягнення найвищого показника точності при тренуванні на конкретному датасеті для конкретної задачі. Для класифікації обирається

клас одягу, що отримав найвище число ймовірності внаслідок роботи алгоритму з моделлю та зображенням, що тестують користувачі.

Важливою частиною роботи нейронної мережі, яка суттєво змінює швидкість навчання та покращує результати, є використання функцій активації. Їх основна задача – математично перетворити вхідні дані до одного шару мережі у такі, що можна буде далі подати до наступного. Схема розрахунку:

$$output_k = activation\left(\sum W_{k,i} * input_i\right) \quad (2.1)$$

Кожен вузол  $k$  певного шару отримує число, що є результатом використання функції активації до суми добутків вузлів  $i$  минулого шару на відповідні числа у матриці вагів.

Використовується ціла низка функцій, в різних комбінаціях, в залежності від моделі та задачі, що ставиться перед розробниками. Серед них є 6 найпопулярніших:

Лінійна  $f(x) = \alpha x$  – при backward propagation буде завжди використовуватися одне і те саме значення похідної (в данному випадку константа).

Двійкова ступінчата  $f(x) = 1, x \geq 0$  or  $f(x) = 0, x < 0$  – не містить компоненту  $X$  через що функція містить нульовий градієнт.

Сигмоїд  $\frac{1}{e^x}$  – дуже популярна функція серед нелінійних, має значення  $[0;1]$ , тому і використовується зазвичай в останньому шарі мережі, щоб видати число, яке може виражати ймовірність передбачення моделі.

Тангенс  $\tanh(x)$  – симетрична відносно нуля, має значення  $[-1;1]$ , через що краще «врівноважує» числа на виході з шару мережі.

1.  $ReLU f(x) = x, x \geq 0, or f(x) = 0, x < 0$  – одна з найпопулярніших нелінійних функцій, адже вона активує не усі нейрони

одночасно, лише певну їх кількість.

2. *Leaky ReLu*  $f(x) = x, x \geq 0, or f(x) = 0,01x, x < 0$  – через зміну у ситуаціях з від'ємними значеннями, така функція вирішує проблему нульового градієнту, що іноді виникає при використанні ReLu.

## 2.2 Огляд існуючих моделей нейронних мереж

У даному розділі буде проведений огляд різних моделей нейронних мереж, які використовуються для розпізнавання одягу в e-commerce системах. Представлені моделі мають різні архітектури та методи, які виявилися ефективними у розв'язанні даної задачі.

Один з найпоширеніших типів моделей, використаних для розпізнавання одягу, є згорткові нейронні мережі (Convolutional Neural Networks, CNN). Вони використовують згорткові шари та пулінг для виявлення різних ознак у зображеннях, що дозволяє точно розпізнавати форму, текстуру та інші характеристики одягу [2], [10].

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) використовуються для обробки послідовних даних, таких як текст або послідовність зображень. У контексті розпізнавання одягу, RNN можуть моделювати зв'язки між різними частинами одягу та враховувати контекстуальну інформацію, наприклад, розпізнавати послідовності слів на футболках або етикетках [20].

Глибинне навчання з підсиленням (Deep Reinforcement Learning, DRL) використовується для навчання моделей, які можуть взаємодіяти з навколишнім середовищем та приймати рішення. В контексті e-commerce систем, DRL може застосовуватися для покращення рекомендаційних систем та персоналізації вибору одягу для користувачів.

Також існують пре-треновані моделі, які були навчені на великих наборах даних, наприклад, ImageNet. Ці моделі можуть використовуватися для передбачення ознак одягу в e-commerce системах. Пре-треновані моделі,

такі як ResNet або Inception, мають велику потужність для розпізнавання зображень [1], [9].

### 2.3 Аналіз їх ефективності та точності

Після вивчення різних моделей нейронних мереж, варто провести аналіз їх ефективності та точності у контексті розпізнавання одягу в e-commerce системах. Це допоможе визначити, які моделі можуть забезпечити найкращі результати для даної задачі.

Ефективність моделей може бути виміряна за допомогою різних метрик, таких як швидкодія, ресурсоємність та масштабованість. Швидкодія визначає час, необхідний для обробки одного зображення моделлю. Ресурсоємність вказує на обсяг ресурсів (наприклад, обчислювальна потужність або пам'ять), необхідних для роботи моделі. Масштабованість оцінює здатність моделі працювати з великими обсягами даних [26].

Точність моделей також є критичним фактором. Вона може бути виміряна за допомогою метрик, таких як точність (accuracy), точність (precision), відновлення (recall) та F-мера (F-measure). Ці метрики вказують на здатність моделі правильно класифікувати одяг на зображеннях. Важливо порівняти результати моделей та вибрати ті, які досягають найвищих показників точності [7].

Після аналізу ефективності та точності різних моделей нейронних мереж, можна зробити висновки про те, які моделі є найбільш підходящими для розпізнавання одягу в e-commerce системах. Це надасть підґрунтя для подальшого дослідження та розробки оптимального рішення для даної задачі.

### 2.4 Оцінка швидкодії та продуктивності моделей

У даному розділі буде проведена оцінка швидкодії та продуктивності

різних моделей нейронних мереж для розпізнавання одягу в e-commerce системах. Це дозволить зрозуміти, які моделі мають найкращу ефективність у відношенні обробки зображень одягу.

Оцінка швидкодії включає вимірювання часу, необхідного для обробки одного зображення моделлю. Цей час може бути вплинутий розміром зображення, складністю моделі та обчислювальними ресурсами, доступними для обробки. Для вимірювання швидкодії можуть використовуватись різні інструменти та метрики, такі як час обробки на одну одиницю даних або кадрів на секунду (FPS).

Продуктивність моделей оцінюється на основі ресурсоемності, тобто обсягу обчислювальних та пам'ятових ресурсів, необхідних для ефективної роботи моделі. Це може включати кількість параметрів моделі, потужність обчислювальних пристроїв, обсяг пам'яті та швидкість передачі даних. Оцінка продуктивності допоможе вибрати моделі, які забезпечують оптимальне співвідношення між точністю та використанням ресурсів.

Для оцінки швидкодії та продуктивності моделей будуть проведені експерименти, використовуючи відповідні набори даних та обчислювальні ресурси. Результати будуть представлені у вигляді часу обробки зображень, кількості кадрів на секунду та ресурсоемності моделей. Це дозволить зробити порівняльний аналіз та визначити найшвидші та найбільш продуктивні моделі для розпізнавання одягу в e-commerce системах.

## 2.5 Ідентифікація переваг та недоліків кожної моделі

В даному розділі будуть проаналізовані переваги та недоліки кожної моделі нейронної мережі, що використовується для розпізнавання одягу в e-commerce системах. Це дозволить зрозуміти, які аспекти кожної моделі можуть бути корисними та в яких аспектах вони можуть мати обмеження.

Одна з переваг моделей може бути пов'язана з їхньою точністю. Деякі моделі можуть досягати високих показників точності у розпізнаванні одягу,

що робить їх ефективними для використання в e-commerce системах. Додатково, деякі моделі можуть мати можливість враховувати контекст та взаємодію об'єктів у зображенні, що сприяє поліпшенню точності та реалістичності розпізнавання.

Недоліки моделей також варто враховувати. Деякі моделі можуть бути вимогливими до обчислювальних ресурсів, що обмежує їхню швидкість та продуктивність на певних платформах. Крім того, деякі моделі можуть мати складну архітектуру та вимагати багато параметрів, що ускладнює їх тренування та налаштування.

Під час ідентифікації переваг та недоліків кожної моделі, будуть враховуватися такі фактори, як точність, швидкість, продуктивність, обчислювальні вимоги та складність архітектури. Це дозволить зробити обґрунтований вибір моделей для подальшого дослідження та розробки ефективної системи розпізнавання одягу в e-commerce.

В рамках дослідження ефективності існуючих нейронних мереж для розпізнавання одягу в e-commerce системах, були досліджені три типи моделей: згорткові нейронні мережі (Convolutional Neural Networks, CNN), рекурентні нейронні мережі (Recurrent Neural Networks, RNN) та комбіновані моделі, що поєднують у собі згорткові та рекурентні шари.

Згорткові нейронні мережі:

- архітектура: згорткові нейронні мережі складаються зі згорткових шарів, пулінгових шарів та повнозв'язаних шарів. Згорткові шари використовують фільтри для виявлення локальних особливостей зображень, пулінгові шари зменшують розмір зображення, а повнозв'язані шари виконують класифікацію;

- особливості: згорткові нейронні мережі показують високу ефективність у розпізнаванні зображень. Вони можуть автоматично вивчати характеристики одягу, такі як форма, кольори та текстури, завдяки своїй здатності до ієрархічного виявлення ознак;

- результати: згорткові нейронні мережі показали високу точність у

розпізнаванні одягу в e-commerce системах. їх можна успішно використовувати для класифікації типу одягу, виявлення певних властивостей (наприклад, взуття, сукні, футболки) та пошуку схожих товарів [24].

Рекурентні нейронні мережі:

– архітектура: рекурентні нейронні мережі мають рекурентні шари, які дозволяють моделі зберігати інформацію про попередні стани в процесі обробки послідовних даних. Такі моделі, наприклад, Long Short-Term Memory (LSTM) і Gated Recurrent Unit (GRU), використовуються для аналізу послідовних зображень одягу, наприклад, зображень з відео або моделей, що рухаються;

– особливості: рекурентні нейронні мережі мають здатність до аналізу послідовностей, що дозволяє їм враховувати контекстуальну інформацію. Вони можуть ефективно моделювати залежності між різними частинами одягу, а також враховувати зміни в часі;

– результати: рекурентні нейронні мережі показали хороші результати в розпізнаванні послідовних зображень одягу. Вони здатні до виявлення змін у позиції одягу, руху та зміни в стилі.

Комбіновані моделі:

– архітектура: комбіновані моделі поєднують у собі згорткові та рекурентні шари для кращої обробки зображень та аналізу послідовностей. Вони можуть використовуватися для змішаного розпізнавання одягу, де аналізуються як статичні зображення, так і динамічні відео;

– особливості: комбіновані моделі поєднують переваги згорткових та рекурентних нейронних мереж і можуть бути ефективними у розпізнаванні одягу з урахуванням контексту та взаємодії між різними частинами одягу;

– результати: комбіновані моделі показали добрі результати в розпізнаванні одягу в e-commerce системах. Вони можуть бути ефективними у виявленні стилю одягу, визначенні комбінацій речей або прогнозуванні модних тенденцій [5].

Виявлені переваги та недоліки кожної моделі в контексті розпізнавання одягу в e-commerce системах наведені нижче.

Згорткові нейронні мережі:

а) переваги:

- висока точність у розпізнаванні зображень одягу;
- здатність виявляти характеристики одягу, такі як форма, кольори та текстури;

б) недоліки:

- можуть бути обмежені у врахуванні контексту та послідовності даних.

Рекурентні нейронні мережі:

а) переваги:

- здатність до аналізу послідовних зображень одягу;
- можливість враховувати зміни в часі та залежності між різними частинами одягу;

б) недоліки:

- потребують більшої кількості обчислювальних ресурсів.

Комбіновані моделі:

а) переваги:

- здатність поєднувати переваги згорткових та рекурентних нейронних мереж;
- ефективність у розпізнаванні одягу з урахуванням контексту та взаємодії між частинами одягу;

б) недоліки:

- потребують більшої кількості обчислювальних ресурсів порівняно з окремими моделями.

Враховуючи переваги та недоліки кожної моделі, вибір певної моделі для розпізнавання одягу в e-commerce системах залежатиме від конкретних потреб проекту, доступних ресурсів та контексту застосування.

Крім того, важливо враховувати взаємодію з існуючими системами e-

commerce. Деякі моделі можуть бути складними для інтеграції з існуючими рішеннями або потребувати специфічних форматів даних для використання. Тому, перед вибором моделі для дослідження, необхідно зважити на сумісність з наявною інфраструктурою та забезпечити легку інтеграцію моделі в e-commerce системи.

## 3 МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ

### 3.1 Вибір датасету для навчання та тестування

Вибір підходящого датасету для навчання та тестування є критичним кроком у дослідженні ефективності існуючих нейронних мереж для розпізнавання одягу в e-commerce системах. Правильно обрані дані визначають можливості моделі та її здатність до узагальнення на нові зображення одягу, а також вплинуть на точність та надійність отриманих результатів.

При виборі датасету необхідно враховувати декілька факторів. По-перше, має бути доступ до достатньої кількості зображень, що представляють різні категорії одягу, стилі, фасони, кольори та інші варіації. Чим більша розмаїтість даних, тим краще модель зможе навчитися розпізнавати різноманітні образи одягу. Також важливо, щоб датасет був достатньо великим, щоб забезпечити достатню репрезентативність та статистичну значимість результатів [30].

По-друге, потрібно врахувати якість та достовірність даних. Датасет повинен бути анотованим, тобто містити позначки або мітки, які вказують на класи або категорії одягу на кожному зображенні. Якісна анотація є важливою для належного навчання моделі та оцінки її ефективності. Крім того, датасет повинен бути без помилок та відповідати актуальним стандартам, щоб уникнути спотворення результатів.

Третій аспект, який варто враховувати, – це баланс між обсягом датасету та обчислювальними обмеженнями. Більші обсяги даних можуть привести до кращих результатів моделі, але це також вимагатиме більше обчислювальних ресурсів та часу для тренування. Тому необхідно забезпечити баланс між вимогами до датасету та можливостями обчислювального середовища.

При виборі датасету можна розглянути кілька популярних варіантів, таких як FashionMNIST, DeepFashion, Open Images Dataset, CelebA та інші. Кожен з них має свої особливості та переваги, які можна оцінити з точки зору потреб дослідження. Наприклад, FashionMNIST є добрим варіантом для початкового етапу дослідження, оскільки він містить широкий спектр категорій одягу та невеликий розмір. З іншого боку, DeepFashion містить значно більше зображень та категорій, що дозволяє глибше досліджувати розпізнавання моделей та стилів одягу [19].

У підсумку, вибір датасету для навчання та тестування є критичним аспектом дослідження ефективності нейронних мереж для розпізнавання одягу в e-commerce системах. Правильно обраний датасет забезпечить якісне навчання моделі, аналіз результатів та висновки щодо ефективності моделей. Враховуючи вищезазначені фактори, можна здійснити обґрунтований вибір датасету, що сприятиме успішному проведенню дослідження та досягненню поставлених цілей [15].

### 3.2 Порівняння різних моделей нейронних мереж

У даному підрозділі проводиться порівняння різних моделей нейронних мереж для розпізнавання одягу в e-commerce системах. Порівняння базується на ряді критеріїв, які включають архітектуру мережі, параметри навчання, швидкість збіжності, точність класифікації, та ефективність в реальному часі.

Архітектура мережі. Перш за все, розглядаються різні архітектури нейронних мереж, такі як згорткові нейронні мережі (Convolutional Neural Networks – CNN), рекурентні нейронні мережі (Recurrent Neural Networks – RNN), та їх комбінації. Оцінюється адекватність використовуваних архітектур для задачі розпізнавання одягу, їх здатність до виділення важливих ознак та здатність до розуміння контексту зображення.

Параметри навчання. Досліджуються різні параметри навчання, такі як швидкість навчання (learning rate), розмір пакета (batch size), функція втрати (loss function) та оптимізатори. Оцінюється, як ці параметри впливають на процес навчання моделі, швидкість збіжності та якість отриманих результатів.

Швидкість збіжності. Важливим критерієм є швидкість збіжності моделей нейронних мереж. Аналізується кількість епох навчання, необхідних для досягнення оптимальної точності, а також час, необхідний для тренування моделі на заданому обсязі даних.

Точність класифікації. Проводиться оцінка точності класифікації моделей на тестовому наборі даних. Порівнюються значення метрик, таких як точність (accuracy), точність за класами (precision), відновлення (recall), та F-мера (F-measure). Враховується як загальна точність моделі, так і її здатність до розпізнавання конкретних класів одягу.

Ефективність в реальному часі. Оцінюється швидкодія моделей під час роботи в реальному часі. Аналізуються час, необхідний для класифікації одного зображення та можливість роботи моделі на різних пристроях з обмеженими обчислювальними ресурсами.

Інші критерії. Розглядаються інші важливі критерії, такі як стійкість до зміни освітлення, масштабування та зміни позиції одягу на зображенні.

Після проведення порівняння різних моделей нейронних мереж за даними критеріями, буде здійснено вибір найбільш ефективної моделі, яка буде використана для подальшого дослідження ефективності нейронних мереж для розпізнавання одягу в e-commerce системах.

### 3.3 Параметризація та тренування моделей

Параметризація включає визначення архітектури моделі, вибір параметрів навчання та налаштування гіперпараметрів.

Перш за все, вибір архітектури моделі є важливим кроком. Розглядаються різні типи архітектур, такі як згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та їх комбінації. Враховуються розмір шарів, кількість шарів, типи шарів та їх послідовність. Підбирається архітектура, яка найкраще відповідає поставленій задачі розпізнавання одягу.

Після вибору архітектури моделі, параметри навчання визначаються для досягнення оптимальних результатів. Це включає в себе вибір функції втрат (наприклад, категоріальна кросс-ентропія), оптимізатора (наприклад, стохастичний градієнтний спуск) та його гіперпараметри (наприклад, швидкість навчання). Параметри навчання дозволяють моделі пристосуватися до навчальних даних та здійснювати поступове оновлення ваг моделі для покращення її прогностичних можливостей [14][23].

Для досягнення оптимальних результатів тренування моделі, важливо відповідно налаштувати гіперпараметри. Це можуть бути параметри, такі як розмір пакета (batch size), кількість епох навчання, регуляризація (наприклад, Dropout) та розмір прихованого шару. Варіюючи ці гіперпараметри, можна досягти кращої загальної точності та швидкості збіжності моделі.

Після визначення параметрів моделі та гіперпараметрів, проводиться процес тренування моделі на вибраному датасеті. Це включає передачу навчальних зображень через мережу, обчислення втрати, виконання зворотного поширення (backpropagation) та оновлення ваг моделі з використанням оптимізатора. Тренування проводиться на підготовленому навчальному датасеті, щоб модель набула здатності розпізнавати різні типи одягу.

Після завершення процесу тренування, оцінюється досягнута точність моделі на навчальному датасеті. Використовуються метрики, такі як точність (accuracy), показник F1 (F1 score) та матриця помилок (confusion matrix), для оцінки прогностичних можливостей моделі. Це дозволяє

оцінити якість навчання моделі та з'ясувати, чи потрібно внести якісь зміни до параметрів чи архітектури моделі [11].

### 3.4 Метрики оцінки ефективності розпізнавання одягу

У цьому пункті розглядаються різні метрики, які використовуються для оцінки ефективності розпізнавання одягу моделями нейронних мереж в e-commerce системах. Ці метрики дозволяють кількісно оцінити точність та надійність моделей у виконанні поставленої задачі.

Однією з ключових метрик є точність (accuracy), яка визначає відношення правильно розпізнаних зображень одягу до загальної кількості зображень у тестовому наборі. Чим вище точність моделі, тим краще вона впоралася з розпізнаванням одягу. Однак, точність сама по собі не є достатньою для повної оцінки ефективності моделі.

Для більш детального аналізу прогностичних можливостей моделей використовуються додаткові метрики, такі як показник F1 (F1 score) та матриця помилок (confusion matrix). Показник F1 є гармонічним середнім між точністю і повнотою і враховує як правильно класифіковані позитивні приклади, так і уникнені помилкові класифікації. Матриця помилок дозволяє визначити кількість правильно та неправильно класифікованих зображень для кожного класу одягу.

Показник F1 (F1 score) є однією з ключових метрик оцінки ефективності моделей машинного навчання і використовується для вимірювання точності та повноти класифікації. Він представляє собою гармонічне середнє між точністю (precision) та чутливістю (recall) і використовується для визначення балансу між цими двома метриками.

Точність (precision) вимірює, яка частка позитивних прогнозів моделі є правильними. Вона визначається як співвідношення кількості правильно класифікованих позитивних зразків до загальної кількості зразків, які

модель визначила як позитивні. Висока точність означає, що модель робить мало помилок, класифікуючи негативні зразки як позитивні.

Чутливість (*recall*), також відома як частка правильно виявлених позитивних зразків, вимірює, яка частка всіх позитивних зразків була правильно виявлена моделлю. Вона визначається як співвідношення кількості правильно виявлених позитивних зразків до загальної кількості наявних позитивних зразків. Висока чутливість свідчить про здатність моделі виявляти більшу кількість позитивних зразків.

Показник F1 об'єднує точність та чутливість у єдиний числовий показник, який відображає здатність моделі до балансованої класифікації. Він обчислюється за формулою:

$$F1 = \frac{2 \times (\textit{precision} \times \textit{recall})}{\textit{precision} + \textit{recall}} \quad (3.1)$$

Цей показник приймає значення від 0 до 1, де 1 означає ідеальну ефективність моделі, а 0 – повну некоректність. Вище значення показника F1 вказує на кращу ефективність моделі з точки зору точності та повноти класифікації.

Показник F1 особливо корисний у випадках, коли у вас є незбалансовані класи даних, де кількість зразків одного класу може бути значно більшою або меншою за кількість зразків іншого класу. Використання показника F1 дозволяє забезпечити адекватну оцінку ефективності моделі, враховуючи обидві метрики – точність та чутливість [3].

Точність (*precision*) вимірює, яка частка позитивних прогнозів моделі є правильними. Вона визначається як співвідношення кількості правильно класифікованих позитивних зразків до загальної кількості зразків, які модель визначила як позитивні. Висока точність означає, що модель робить мало помилок, класифікуючи негативні зразки як позитивні.

Матриця помилок (confusion matrix) є інструментом для визначення ефективності моделі класифікації, що дозволяє зробити детальний аналіз результатів класифікації для кожного класу. Вона використовується для визначення кількості правильно та неправильно класифікованих зразків кожного класу і дає можливість виміряти рівень помилок, що виникають під час класифікації [4].

Матриця помилок представляється у вигляді квадратної таблиці, де кожен ряд відповідає дійсному класу, а кожен стовпчик – передбаченому класу моделі. Таким чином, матриця помилок має розмірність  $N \times N$ , де  $N$  – кількість класів. Кожен елемент матриці показує кількість зразків, які належать дійсному класу  $i$  і були класифіковані моделлю як клас  $j$ .

У матриці помилок можна виділити чотири основні категорії:

1. True Positive (TP) – правильно класифіковані позитивні зразки. Це випадки, коли модель визначила зразок як позитивний і він дійсно є позитивним;

2. True Negative (TN) – правильно класифіковані негативні зразки. Це випадки, коли модель визначила зразок як негативний і він дійсно є негативним;

3. False Positive (FP) – неправильно класифіковані позитивні зразки. Це випадки, коли модель помилково визначила зразок як позитивний, хоча він є негативним;

4. False Negative (FN) – неправильно класифіковані негативні зразки. Це випадки, коли модель помилково визначила зразок як негативний, хоча він є позитивним.

Матриця помилок дозволяє розрахувати різні метрики ефективності класифікації, включаючи точність (precision), чутливість (recall), специфічність (specificity) та F1-показник (F1 score). Ці метрики використовуються для оцінки рівня помилок моделі і забезпечують збалансовану оцінку її ефективності з урахуванням обох типів помилок.

Детальний аналіз матриці помилок може допомогти виявити особливості роботи моделі, такі як схильність до певних типів помилок або домінування певних класів. Це може бути важливим для подальшого покращення моделі шляхом налаштування параметрів або збільшення кількості тренувальних даних для певних класів [29].

Крім вищезгаданих метрик, можливо розглянути також середньоквадратичну помилку (mean squared error), яка вимірює середню квадратичну різницю між прогнозованими значеннями і дійсними значеннями. Ця метрика особливо корисна у випадку, коли необхідно оцінити рівень точності прогнозування числових параметрів одягу, наприклад, цін, розмірів або ваги.

Загалом, використання відповідних метрик оцінки ефективності розпізнавання одягу дозволяє отримати більш повне уявлення про продуктивність та точність нейронних мереж у контексті e-commerce систем. Враховуючи ці метрики при виборі та оптимізації моделей, можна досягти більш високої якості розпізнавання одягу, що є ключовим фактором для успішної реалізації e-commerce рішень.

## 4 ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ ТА АНАЛІЗ

### 4.1 Вибір датасету та моделі

Під час роботи з машинним навчанням однією з ключових складових успіху є вибір якісного набору даних для тренування нейронної мережі. Отже, основною ціллю цієї роботи було знайти відповідний набір даних, який би містив достатній обсяг інформації та був правильно розподілений між класами. Не завжди корисно вибирати найбільший набір даних для тренування, або будь-який набір даних може бути непридатний через неправильний розподіл даних між класами, які він містить. Деякі набори даних, незважаючи на їхню величину і здавалося б, ефективність для тренування нейронних мереж, можуть викликати перенавчання або передискретизацію. Перший датасет, який часто використовується при розв'язанні задач класифікації, – це MNIST, призначений для навчання простої моделі класифікації зображень цифр. Існує також кілька відомих наборів даних для класифікації одягу, наприклад, Fashion MNIST, який містить 70 тисяч зображень низької якості в чорно-білому кольорі. Однак для цієї роботи необхідний більш складний класифікатор, тому було вирішено використовувати інший датасет. DeepFashion - це набір даних, що містить 800 тисяч кольорових зображень невисокої якості з реальних веб-сайтів магазинів одягу. Він дозволяє розробити класифікатор, який може працювати з 50 класами та категоріями жіночого одягу, що є достатньо високим для класифікатора. Сам набір даних демонструє показує здатність до тренування мереж високого рівня ефективності та точності, що для певних задач перевищує 90%.

### 4.2 Проектування нейронних мереж

Для розробки та тренування нейронної мережі на великих наборах даних, як-то датасет DeepFashion, важливо використовувати платформу, що забезпечує швидке читання файлів та вміщує велику кількість даних. Google Colaboratory – це одна з таких платформ, що надає можливість ефективно працювати з графічним представленням файлів, швидко до них доступатися, а також зберігати до 150Гб даних. Після завантаження датасету в Google Colaboratory, наступним кроком є розробка архітектури для доступу до тренувальних даних.

В датасеті, крім самого зображення, є CSV-файли, які містять інформацію про шлях до файлу і деякі деталі про нього. Наприклад, файл `train.csv` включає таблицю із назвами зображень та відповідними категоріями одягу, що є необхідною для тренування мережі, оскільки вона використовує ці позначені дані для навчання.

Процес тренування моделі включає використання бібліотек `Fastai` та `PyTorch`. `Fastai` використовується для роботи з даними, включаючи читання CSV-файлів, а також для використання методів навчання мережі, таких як `fastai.vision.learner.cnn_learner`, `fastai.vision.learner.fine_tune` та `fastai.vision.learner.fit_one_cycle`. Ці методи дозволяють розробляти нейронну мережу без занурення у деталі її архітектури. Використовуючи ці методи, можна навчити мережу виконувати необхідні завдання.

Важливим аспектом навчання є вибір типу мережі. У даному випадку було вибрано залишкову мережу – `ResNet34` з бібліотеки `torchvision.models`, яка складається з 34-х шарів і була попередньо навчена на датасеті `ImageNet` – даних, що використовуються для створення моделі, що розпізнає образи на зображеннях, без конкретної класифікації.

Під час створення нейронної мережі важливо враховувати декілька ключових аспектів для її оптимізації. Ці елементи відомі як гіперпараметри нейронної мережі, які впливають на ефективність мережі. Вони забезпечують можливість більш продуктивного навчання мережі, зменшуючи кількість необхідних ітерацій та збільшуючи прогрес за кожен

ітерацію. Одним із таких параметрів є коефіцієнт швидкості навчання. Він дозволяє налаштувати швидкість навчання мережі, передаючи його як параметр при тренуванні. Проте, перед цим потрібно визначити оптимальне значення цього коефіцієнта. За допомогою методу `learner.lr_find()` ми можемо побудувати графік (рисунок 4.1), що відображає залежність втрат в процесі навчання від величини коефіцієнта швидкості навчання [9], [16].

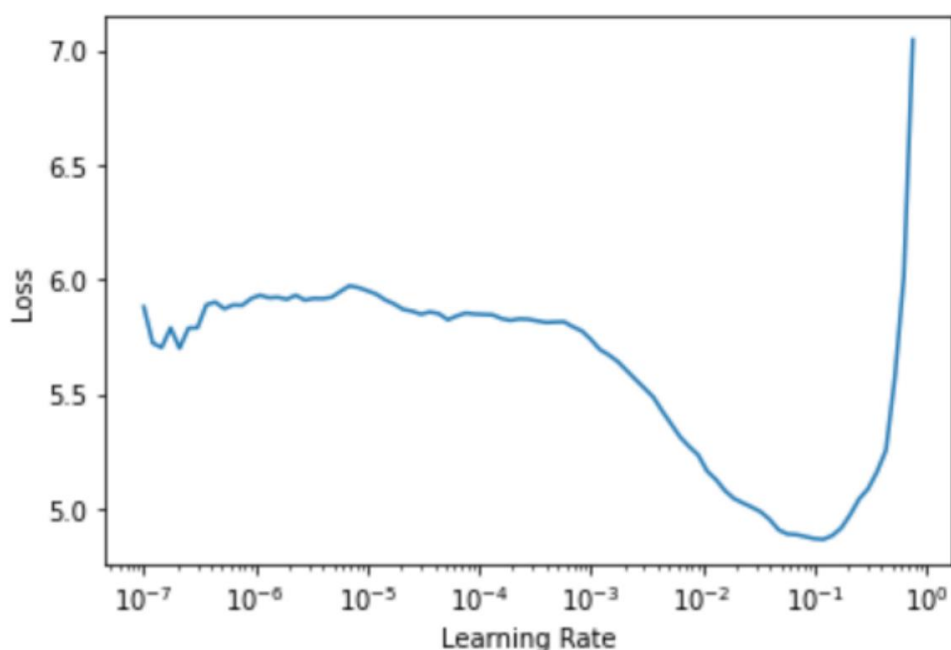


Рисунок 4.1 – Графік залежності втрат при навчанні від коефіцієнту швидкості навчання

З теоретичної точки зору, оптимальним коефіцієнтом для навчання мережі на початкових стадіях є той, при якому градієнт графіка найбільш крутий. Аналіз графіка дозволив припустити, що це значення лежить в діапазоні між  $10^{-3}$  та  $10^{-2}$ . Додатково, цей коефіцієнт можна визначити за допомогою методу `learner.lr_steep()`. Результат виконання цього методу був  $9.12e-03$ , що еквівалентно числу 0.009 і відповідає очікуваному діапазону, будучи ближче до  $10^{-2}$ . Отже, на початкових стадіях було використано метод `learn.fine_tune(2, base_lr=9.12e-03)`, де перший параметр вказує кількість

ітерацій навчання, а другий - оптимальний коефіцієнт швидкості навчання. Метод `fine_tune` застосовується, коли потрібно адаптувати модель до нового контексту, тренуючи модель в мережі кілька разів для поступового покращення результатів. Як вже було зазначено, базова модель ResNet, яку було використано для тренування нейронної мережі, була натренована на базовому наборі даних, тому використання методу `fine_tune` з самого початку є адаптацією вже натренованої моделі до конкретного завдання [14].

Але, використання одного й того самого коефіцієнту швидкості навчання мережі не є правильною тактикою для більшої кількості ітерацій, адже це може призвести до ефекту перенавчання, тому з цим коефіцієнтом буде проведено лише дві ітерації. Це так само і підтверджує видозмінений графік залежності втрат від коефіцієнту швидкості навчання, який побудовано вже після проходження перших ітерацій (рисунок 4.2).

SuggestedLRs(lr\_min=1.4454397387453355e-06, lr\_steep=9.999999747378752e-06)

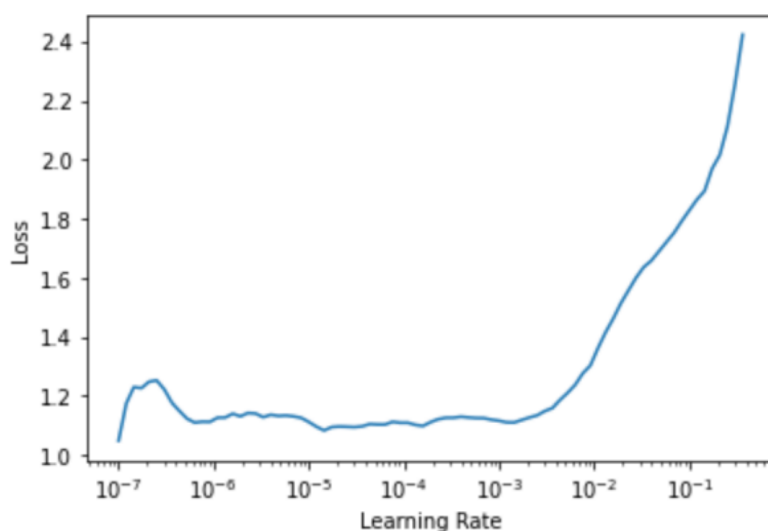


Рисунок 4.2 – Новий графік залежності втрат при навчанні від коефіцієнту швидкості навчання

Система автоматично рекомендує коефіцієнти, які було б оптимально застосувати для подальшого навчання. Ці рекомендації базуються на даних

залежностей, де, подібно до попереднього випадку, ідеальними є значення коефіцієнтів, при яких втрати зменшуються або майже залишаються без змін. Для додаткового тестування використовується метод `learn.fit_one_cycle(5, lr_max=slice(1e-7, 5e-3))`, який дозволяє не лише встановлювати коефіцієнт швидкості навчання, але й задавати діапазон, в якому він буде коливатися з початку до кінця ітерацій. Це необхідно для забезпечення того, щоб коефіцієнт зростав по мірі продовження навчання, оскільки навчання з дуже малим коефіцієнтом, при якому втрати знижуються (в даному випадку в межах від  $10^{-7}$  до  $10^{-6}$ ) може бути неефективним, якщо проводити його протягом всіх ітерацій. Ось чому метод `fit_one_cycle()` надає можливість вказати діапазон коефіцієнтів швидкості навчання, що дозволяє збільшити його ефективність і різноманітність. В даній ситуації, як на графіку, так і згідно з рекомендаціями системи, вибрано діапазон від  $10^{-7}$  (що дорівнює  $1e-7$ ) до  $0.005$ , точки, де графік починає зростати, між  $10^{-3}$  і  $10^{-2}$ , що дорівнює  $5e-3$ . Перший параметр методу – це кількість ітерацій. Вибір саме п'яти ітерацій обумовлено експериментально, це детально описано в наступному розділі, але в загальному – експеримент показав, що більша кількість ітерацій не є корисною для навчання [12][17].

Для порівняння результатів у цій роботі використовувалася також модель YOLO. Розроблена Джозефом Редмоном у 2016 році, YOLO є однією з найшвидших мереж для виявлення об'єктів у реальному часі. До YOLO для виявлення об'єктів модифікували мережі класифікаторів. Але тепер, з YOLO, виявлення об'єктів формулюється як «задача регресії до просторово розділених обмежувальних рамок і пов'язаних з ними ймовірностей класів». YOLO використовує єдину нейронну мережу, яка прогнозує обмежувальні рамки та ймовірності класів на основі повного зображення безпосередньо за одну оцінку. Саме тому вона отримала назву «You Only Look Once». Основна задача, яку виконує YOLO для розпізнавання, дуже проста. Вхідні зображення масштабуються до розміру  $448 \times 448$  і передаються нейронній

мережі. Наша система обмежує отримані результати виявлення певним рівнем достовірності, встановленим у моделі. Це означає, що на виході відображаються лише ті виявлення, достовірність яких вища за фіксоване значення. Пізніше були розроблені новіші версії YOLO, які є кращими і швидшими за початкову версію [27].

У цій версії реалізована остання версія YOLO, версія 3. YOLO v3 використовує Darknet 53 як основу. На додаток до існуючих 53 шарів в даркнеті, які попередньо підготовлені за допомогою ImageNet, на них накладаються ще 53 шари, що робить YOLO v3 106-шаровою мережею.

У Yolo v3 розпізнавання в основному відбувається на 3 шарах. Перше виявлення відбувається на 82-му шарі, де крок буде 32, а карта об'єктів матиме розміри 13x13. Таким чином, вихід на цьому шарі має вигляд 13x13x255. На цьому шарі виявляються більші об'єкти. Наступне виявлення здійснюється на 94-му шарі, де крок буде 16, а карта об'єктів матиме розмір 26x26. На цьому етапі виконується висхідна дискретизація, що є основною зміною у версії 3 порівняно з першими двома моделями. Після висхідної дискретизації карта особливостей на 79-му шарі об'єднується за глибиною з картою особливостей з 61-го шару. Таким чином, вихідна форма на 94-му шарі має розмір 26x26x225. На цьому шарі виявлено об'єкти середнього розміру. Знову ж таки, карти об'єктів піднімаються вгору, і цього разу карти об'єктів з 36-го шару конкатенуються перед останнім шаром. Остаточне детектування відбувається в останньому шарі, тобто 106-му, де крок дорівнює 8, а розмір отриманої карти об'єктів буде 52x52. Таким чином, вихідна форма на цьому шарі має вигляд 52x52x225. На цьому шарі можна легко виявити невеликі об'єкти [2].

### 4.3 Оцінка роботи мережі

Початкові ітерації тренування показали серйозні покращення

результату як у точності моделі, так і у зменшенні втрат. Результат перших двох ітерацій показано у таблиці 4.1, 4.2

Таблиця 4.1 – Перша ітерація тренування нейронної мережі

epoch	Train_loss	Valid_loss	accuracy	time
0	1.551926	1.412060	0.589022	33:11

Таблиця 4.2 – Друга ітерація тренування нейронної мережі

epoch	Train_loss	Valid_loss	accuracy	time
0	1.338351	1.221115	0.650659	37:45
1	1.161446	1.0729.70	0.691656	37:43

Важливо підмітити, що не тільки підвищилась точність розпізнавання на 10%, а ще і зменшились втрати, майже у півтора рази. Але, подальші ітерації мережі не показали суттєвих змін, тому для уникнення ризику перенавчання, що може почати зменшити точність, навіть якщо втрати будуть знижуватися, після 5 ітерацій навчання було припинено. Результат показаний у таблиці 4.3.

Саме п'ятої ітерації виявилось достатньо для навчання моделі для високого рівня точності у виборі топ-1 ймовірності для результату. Пошук ідеальної кількості ітерацій – процес експериментальний. Ці приклади показують, що на етапах подальшого навчання за такої динаміки виникають проблеми перенавчання моделі, адже починають рости втрати навіть за невеликого збільшення відсотка точності, що не компенсує факт того, що втрати при навчанні – показник того, що мережа тренується неповністю.

Таблиця 4.3 – Результат тренування після 5-ї ітерації

epoch	train_loss	valid_loss	accuracy	time
-------	------------	------------	----------	------

0	1.102886	1.072060	0.691174	37:42
1	1.104428	1.063833	0.694908	37:41
2	1.105640	1.055722	0.696234	37:39
3	1.067472	1.048353	0.698121	37:40
4	1.064548	1.043491	0.700249	37:56

З таблиці 4.4 видно, що на шостій ітерації точність моделі знижується, а втрати зростають. Втрати також збільшуються на етапі другої ітерації, але потім знову падають. Графік втрат показує, що в деяких місцях втрати перевищують показники попередніх ітерацій, що свідчить про те, що процес навчання не приносить очікуваних результатів (рисунок 4.3).

Використовуючи бібліотеку `matplotlib`, ми можемо побудувати графік, який демонструє, як змінюється точність моделі в процесі навчання (рисунок 4.4), і бачимо, що починаючи з шостої ітерації точність знижується.

Таблиця 4.4 – Приклад тренування на 6 ітераціях

epoch	train_loss	valid_loss	accuracy	time
0	1.156718	1.115304	0.682220	26:32
1	1.204249	1.118955	0.684027	26:38
2	1.141088	1.095323	0.6855933	26:34
3	1.087024	1.0911421	0.685633	26:31
4	1.073382	1.073441	0.692620	26:27
5	1.082264	1.070463	0.692258	26:27

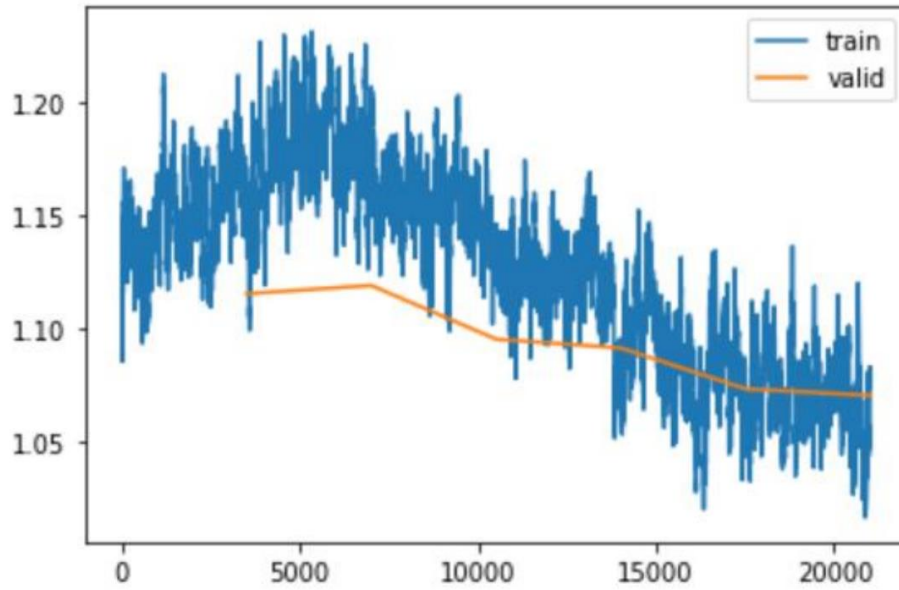


Рисунок 4.3 – Графік втрат

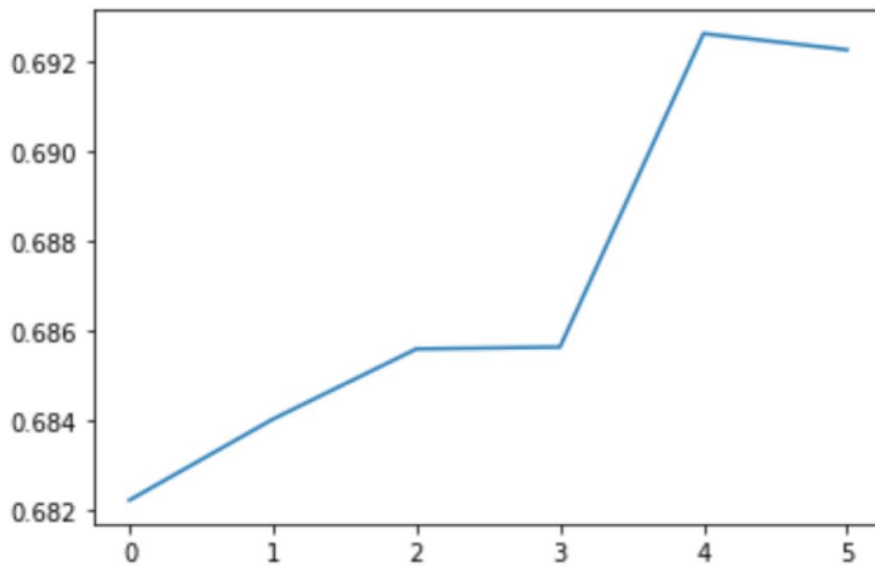


Рисунок 4.4 – Графік зміни точності моделі

Такий підхід до навчання може призвести до проблеми зміщення та дисперсії. Наприклад, якщо ми тренуємо мережу багато разів на одному й тому ж наборі даних, очікуючи покращення результатів, можна стикнутися з ситуацією, коли мережа перестає коректно реагувати на нові входні дані,

які повинна розпізнавати після навчання. Враховуючи те, що зазвичай в датасетах використовуються досить схожі зображення, це може призвести до того, що мережа стає менш чутливою до нових вхідних даних, оскільки вона занадто спеціалізується на конкретних даних з датасету, які можуть не відповідати тому, що буде подано користувачем у майбутньому. Модель з такою проблемою називають моделлю з високою дисперсією (рисунок 4.5).

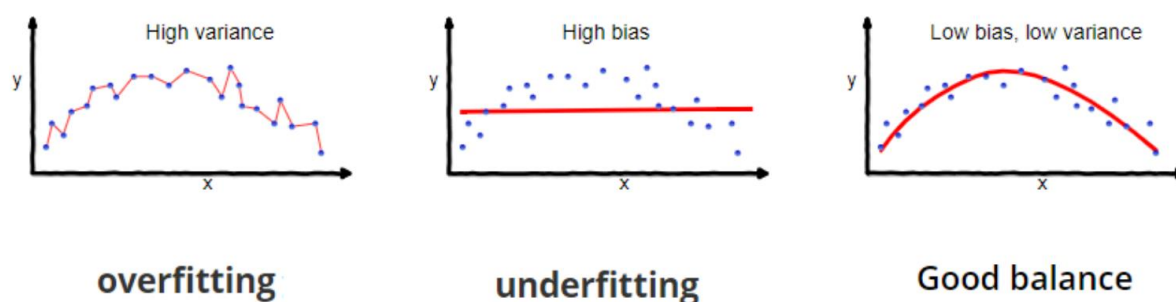


Рисунок 4.5 – Ілюстрація до дилеми зміщення та дисперсії

З першого погляду, може здатися, що недостатній обсяг ітерацій, неправильно обрані коефіцієнти навчання або інші гіперпараметри можуть призвести до недоучування і великого зміщення. В такому випадку, модель ще не набула потрібної чутливості для розпізнавання та класифікації майбутніх вхідних даних. Тому критично важливим є належний підхід до побудови архітектури мережі, обирання оптимальних засобів для її навчання, відповідні датасети, а також експерименти з різними параметрами для досягнення бажаного результату. Крім того, порівнюючи отриману точність у 70% при визначенні найкращого результату з точністю 82.6% при визначенні трьох кращих результатів, можна зробити висновок, що модель навчилася на достатньо високому рівні (рисунок 4.6).

	Category	
	top-3	top-5
WTBI [3]	43.73	66.26
DARN [10]	59.48	79.58
FashionNet+100	47.38	70.57
FashionNet+500	57.44	77.39
FashionNet+Joints [34]	72.30	81.52
FashionNet+Poselets [34]	75.34	84.87
FashionNet (Ours)	<b>82.58</b>	<b>90.17</b>

Рисунок 4.6 – Дані класифікатора системи датасету для топ-3 та топ-5 результатів

Для ілюстрації, як розраховується точність моделі, розглянемо модель з двома класами: «Позитивний» і «Негативний». Для розрахунку модель створює так звану матрицю помилок, яка включає в себе кілька комбінацій. Ці комбінації поділяються на 4 типи: Істинно Позитивний (True Positive, TP), Істинно Негативний (True Negative, TN), Хибно Позитивний (False Positive, FP), Хибно Негативний (False Negative, FN) [15].

В даному випадку:

- True Positive – передбачення збіглося з класом, обидва Positive;
- True Negative – передбачення збіглося з класом, обидва Negative;
- False Positive – передбачено Positive, а насправді Negative;
- False Negative – передбачено Negative, а насправді Positive.

Тоді точність моделі буде вираховуватися за формулою:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

Матриця подібного типу використовувалась для оцінки результатів навчання моделі, де чисельник – це значення, які знаходяться на головній діагоналі матриці, а знаменник – це всі значення матриці. Дійсно, точність моделі можна представити як відношення коректних прогнозів до загальної кількості прогнозів.

Для визначення втрат під час навчання і тестування нейронної мережі використовується відмінність між реальним класом зображення і передбаченим моделлю класом. Коефіцієнт втрат обчислюється за допомогою наступної формули:

$$\text{Cost function} = \frac{1}{m} \sum L(y', y) \quad (4.2)$$

де  $m$  – кількість зображень, що використано при ітерації;

$L$  – функція втрат для кожного конкретного зображення;

$y'$  – результат передбачення;

$y$  – реальний клас.

Для кожного зображення розраховується значення втрат за формулою 4.3:

$$\text{Loss function} = -y * \log y' - (1 - y) * \log (1 - y') \quad (4.3)$$

Отже, мережа обчислює втрати, які модель зазнала під час навчання, а також ті, які виникли на завершення ітерації навчання. При успішному навчанні, втрати `train_loss` і `validation_loss` мають якомога більше наближатися одне до одного. Якщо між ними значна різниця, це може вказувати на проблеми недонавчання або перенавчання. За допомогою методу `learner.show_results()` можна оцінити ефективність навченої моделі.

Цей метод з бібліотеки Fastai виводить кілька прогнозів для зображень з набору даних і порівнює їх із відповідними реальними класами. Метод використовує зображення з набору даних, демонструючи різноманітність класів і як модель працює для прогнозування різних результатів, при цьому він виділяє ті прогнози, які відрізняються від реальних класів (рисунок 4.7).

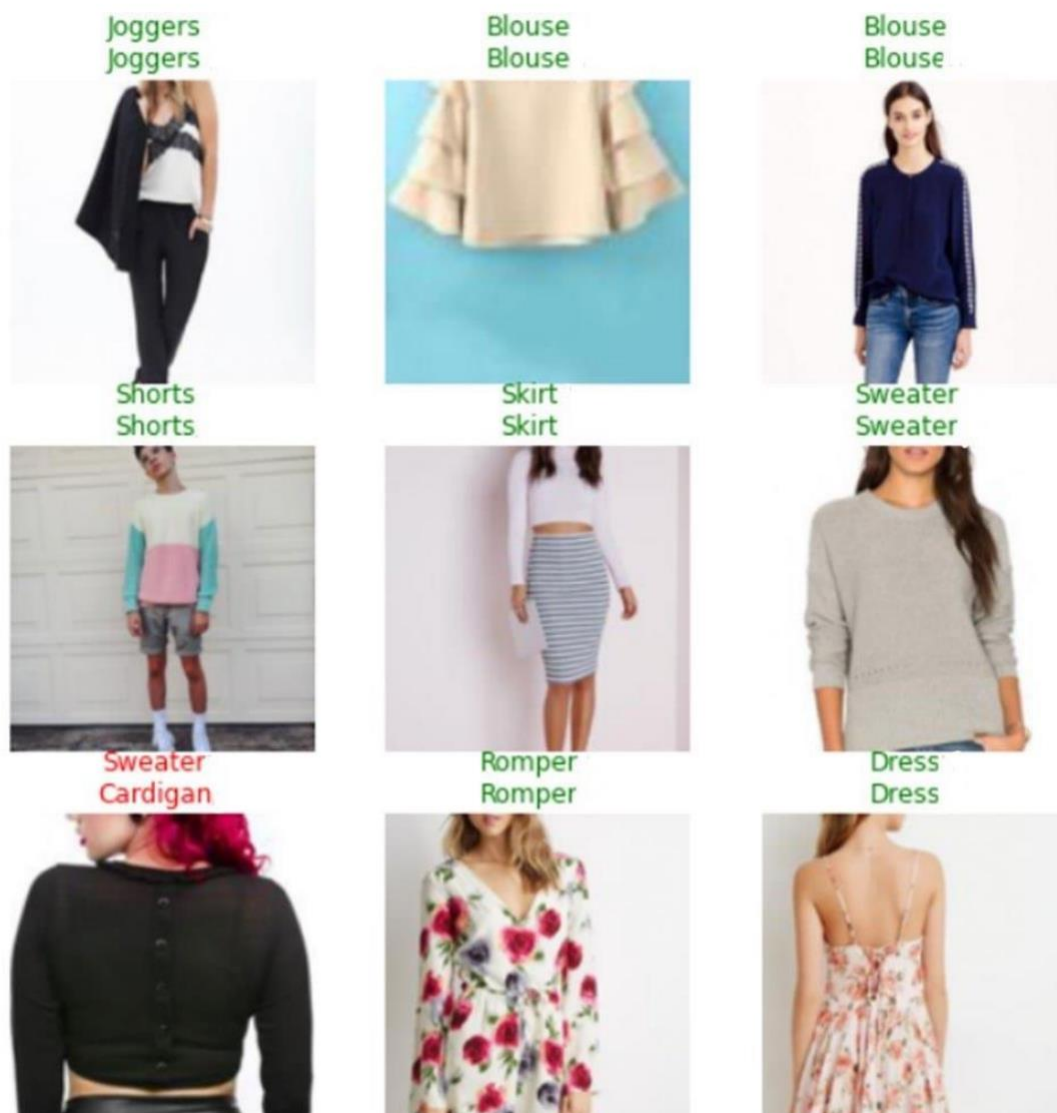


Рисунок 4.7 – Приклад передбачень моделлю над зображеннями датасету

Згідно з результатами, модель дійсно була навчена прогнозувати та розпізнавати з високою точністю. Під час тестування моделі також вдалося

переконалися, що вона дійсно працює ефективно. Однак є одне питання щодо результатів роботи навченої моделі – це питання про зміщення та дисперсію, яке є досить поширеним у теорії машинного навчання. У цьому дослідженні набір даних було навчено виключно на зображеннях, отриманих з веб-сайтів інтернет-магазинів одягу, які значно відрізняються від зображень з реального життя. Таким чином, система, безумовно, демонструє достатньо високу точність класифікації при тестуванні таких зображень, але є можливість, що при використанні звичайного зображення від людини - такі показники гарантувати не можна.

Для YOLOv3 ми використовуємо попередньо навчені вагові коефіцієнти моделей, навчені на наборі даних COCO, щоб створити нашу модель. Після підготовки набору даних, під час навчання мережі нам потрібно змінити конфігураційний файл відповідно до нашої моделі. Спочатку набір даних COCO містить 80 класів. У нашому випадку ми маємо 7 класів. Отже, фільтри, необхідні для 7 класів, обчислюються за формулою  $filters = (classes + 5) * 3$ , що дає нам загалом 36 фільтрів. Отже, на трьох шарах виявлення йоло значення класів та номери фільтрів дорівнюють 7 та 36 відповідно [20]. Вхідні зображення під час навчання задаються пакетом 64 з підрозділами по 16. Роздільна здатність вхідних даних для мережі YOLO має бути кратною 32, тому в нашому випадку ми використовуємо роздільну здатність 416x416.

Швидкість навчання встановлено на 0.001 з інтервалом 0.9 та спадом 0.0005. Оскільки ми маємо 7 класів, модель має бути навчена до 14000 ітерацій. Найкращою практикою є навчання мережі до точки, після якої середні втрати більше не зменшуються або залишаються постійними. Під час навчання ваги зберігаються для кожних 1000 ітерацій. Зазвичай останні значення мають кращі результати. Після навчання мережі на навчальній вибірці нам потрібно використати ці значення для валідації моделі. Для цього ми обчислюємо середню точність (mAP). mAP визначається шляхом порівняння прогнозованих результатів з вихідними базовими значеннями

Після навчання мережі моделі тестуються на зображеннях, знайдених у мережі. Виявлення одягу відбувається за допомогою детектора YOLO v3. Для аналізу продуктивності системи ми обираємо значення, які дають найкращі результати. Кожне зображення тестується індивідуально за допомогою мережі детекторів за допомогою мобільного телефону. Для знайдених 20 зображень ми отримали середній показник точності 89% [14].

mAP також тестується для контрольного набору. Контрольний набір містить 20% від загальної кількості зображень у наборі даних, тобто 364 зображення. Для цього контрольного набору ми отримали значення mAP 82% (рисунок 4.8).

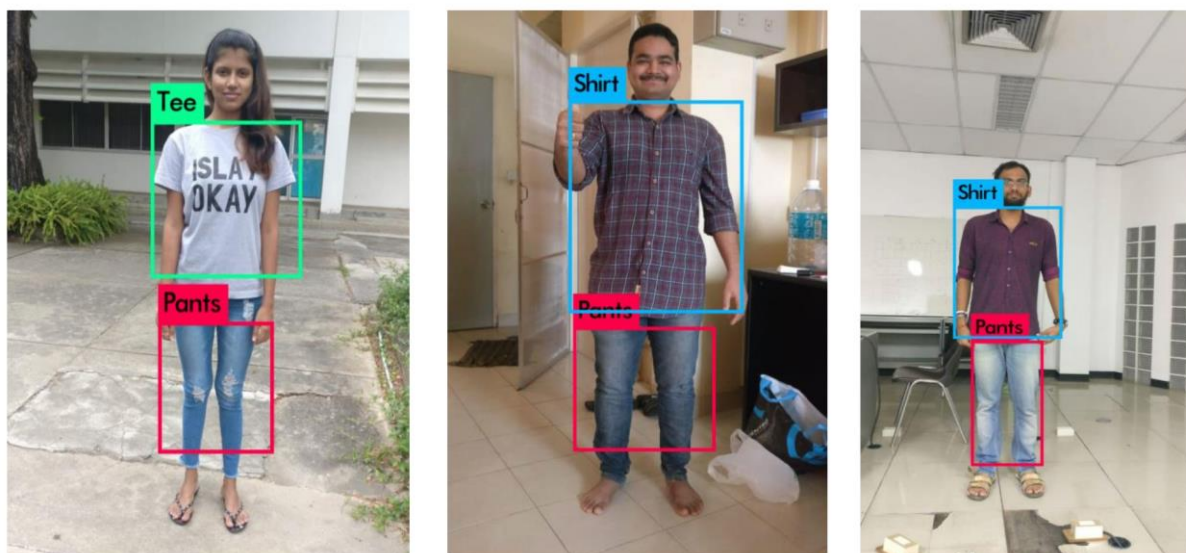


Рисунок 4.8 – Приклад передбачень моделлю над зображеннями датасету

Показник F1 з матриці плутанини можна розрахувати за формулою 4.4.

$$F1 = 2 * (Precision * Recall) / (Precision + Recall) \quad (4.4)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4.5)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4.6)$$

Показник F1 з таблиці 4.4 становить 95%.

Таблиця 4.4 – Матриця помилок для результатів тесту

	<b>Dress</b>	<b>Tee</b>	<b>Shorts</b>	<b>Pants</b>	<b>Shirt</b>	<b>Spectacles</b>	<b>Cap</b>	<b>Missing</b>
<b>Dress</b>	5	-	-	-	-	-	-	-
<b>Tee</b>	-	8	-	-	1	-	-	1
<b>Shorts</b>	-	-	4	-	-	-	-	-
<b>Pants</b>	-	-	-	9	-	-	-	-
<b>Shirt</b>	-	1	-	-	4	-	-	-
<b>Spectacles</b>	-	-	-	-	-	7	-	3
<b>Cap</b>	-	-	-	-	-	-	3	1

Таблиця 4.5 – Значення точності розпізнавання одягу на тестових зображеннях.

<b>Number of Iterations</b>	<b>Precision (%)</b>
1000	82.41
2000	90.03
3000	87.01
4000	87.88
5000	84.23
6000	88.74
7000	95.84
8000	83.84
9000	86.26
10000	87.12
11000	88.31
12000	89.33
13000	89.33
14000	89.33

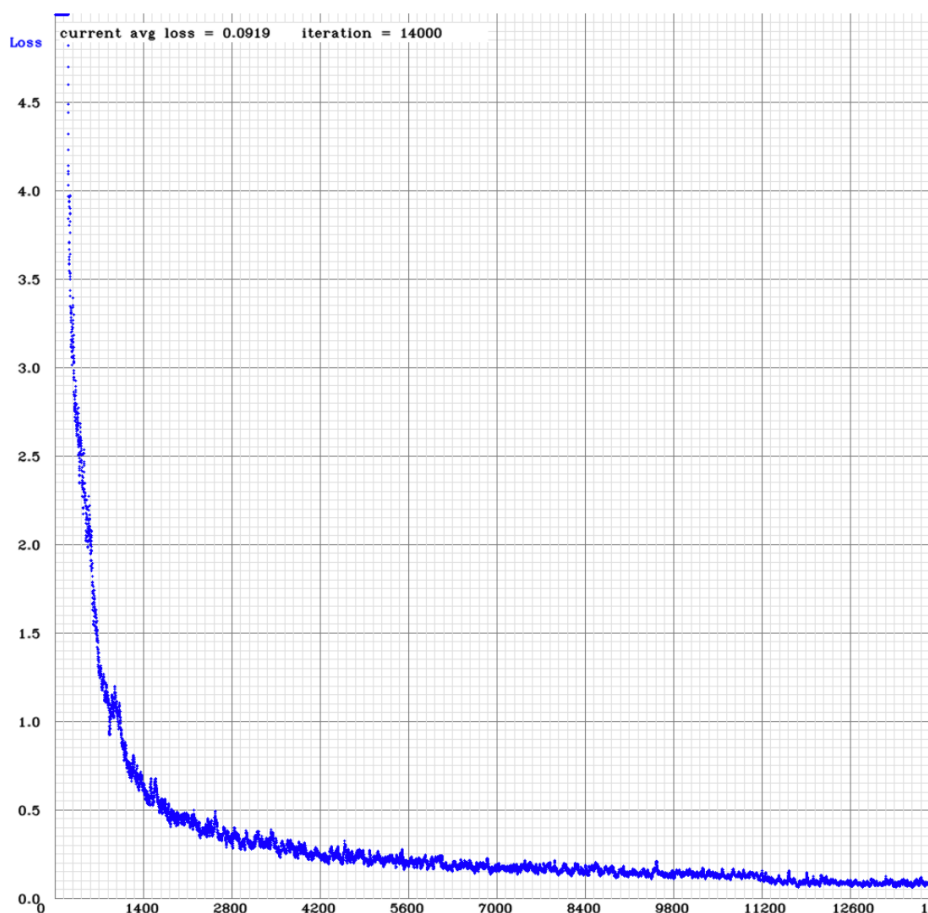


Рисунок 4.9 – Графічне представлення середніх втрат від кількості ітерацій.

#### 4.4 Аналіз отриманих результатів та вибір найефективнішої моделі

Таким чином, мета роботи, «Автоматичне розпізнавання товарів», була успішно реалізована за допомогою двох моделей YOLOv3 та ResNet34. Навчивши YOLO v3 на майже 1800 зображеннях, ми отримали точність близько 89%. Хоча більшість зображень містить студійні фонові зображення, модель продемонструвала винятково хороші результати на реальних зображеннях. Модель ResNet18 досягла 70% точності на валідаційному наборі. Якщо розглянути ці дві моделі, варто розглянути кілька аспектів. YOLOv3, або «You Only Look Once», є третім поколінням моделі, що спеціалізується на виявленні об'єктів. Вона була створена для

швидкості та точності, і включає багатокласову класифікацію та виявлення об'єктів різних розмірів. Однак, через свою складність, YOLOv3 може бути занадто надмірною для завдання розпізнавання одягу, яке може не вимагати такої розширеної функціональності. Незважаючи на її швидкість, впровадження YOLOv3 може бути більш трудомістким і вимагати більше обчислювальних ресурсів.

З іншого боку, ResNet34 є глибокою конволюційною нейронною мережею, яка використовує залишкові блоки для навчання на великій глибині без проблем з вимкненням градієнтів. Ця мережа, яка складається з 34 шарів, може бути більш відповідною для розпізнавання одягу, оскільки вона може навчатися на відносно простіших зображеннях. Незважаючи на те, що ResNet34 може бути повільніше за YOLOv3, вона менш вимоглива до ресурсів і легше для впровадження [15].

Використовуючи ResNet34, можна очікувати добру точність для завдань розпізнавання зображень. Хоча YOLOv3 також може виявляти об'єкти з високою точністю, вона може включати багато непотрібних особливостей для розпізнавання одягу.

Отже, ResNet34, ймовірно, буде більш відповідним вибором для розпізнавання одягу, оскільки вона простіша, менш вимоглива до ресурсів і все ще надає високу точність. На відміну від YOLOv3, що, незважаючи на свою високу швидкість і точність, може бути занадто складною для цього конкретного завдання.

## 5 ПРАКТИЧНА РЕАЛІЗАЦІЯ

### 5.1 Інтеграція найефективнішої моделі у e-commerce систему

Щоб ефективно реалізувати проект, нам потрібно зосередитися на зборі та підготовці якісного набору даних для задачі розпізнавання одягу. Наша мета - створити різноманітний набір даних, який відображає широкий спектр предметів одягу, що можуть зустрітися в реальних умовах.

Ми маємо зібрати наш датасет з різних відкритих джерел, забезпечуючи представленість різних типів одягу: від суконь і штанів до футболок, курток та взуття. Важливо врахувати різноманітність стилів, кольорів, форм і текстур, щоб модель була готова до роботи з різними сценаріями.

Якість зображень є ключовим елементом підготовки даних. Всі зображення повинні бути перевірені на наявність артефактів або шуму, що можуть вплинути на результати навчання. Вони також мають бути приведені до єдиного розміру і формату.

Після цього, нам потрібно поділити зібраний набір даних на тренувальний, валідаційний та тестовий піднабори. Тренувальний набір використовується для навчання моделі, валідаційний - для тюнінгу гіперпараметрів та моніторингу процесу навчання, а тестовий - для оцінки фінальної моделі.

Підготовка даних вимагає значних зусиль та ресурсів, але вона є ключовою для ефективного навчання моделі ResNet34, що здатна розпізнавати та класифікувати одяг в реальній e-commerce системі.

### 5.2 Навчання моделі ResNet34

Після підготовки даних, наступним кроком було б навчання моделі ResNet34. Модель ResNet34 є відомою за свою ефективність у задачах

комп'ютерного зору, і нам потрібно було використати це для нашої задачі розпізнавання одягу.

Ми маємо використовувати тренувальний набір даних для навчання моделі. Це процес, який може зайняти деякий час, залежно від складності моделі, розміру набору даних і потужності обчислювального обладнання. Нам варто постійно моніторити процес навчання, використовуючи валідаційний набір даних, щоб вчасно виявити проблеми, такі як перенавчання.

Важливо зазначити, що нам може знадобитися провести тюнінг гіперпараметрів моделі, щоб досягти найкращих результатів. Це може включати налаштування таких параметрів, як швидкість навчання, кількість епох навчання, розмір партії та ін.

Нарешті, після завершення навчання, ми повинні оцінити модель, використовуючи тестовий набір даних. Це допоможе нам визначити, наскільки добре модель справляється з новими, невідомими їй раніше даними, і чи готова вона до інтеграції в реальну e-commerce систему.

### 5.3 Інтеграцію у e-commerce систему

Після успішного навчання і валідації моделі ResNet34, наступним важливим кроком має бути її інтеграція в реальну e-commerce систему. Цей етап передбачає декілька ключових підзадач:

- розробка API для моделі. Щоб інтегрувати модель ResNet34 з нашою e-commerce системою, нам потрібно було б створити міцний і надійний API. Цей API має бути спроектований таким чином, щоб він міг приймати зображення від користувачів, передавати їх моделі для аналізу, і повертати висновки моделі в систему для подальшого використання;

- інтеграція моделі з базою даних продуктів. Щоб модель могла бути використана для рекомендацій товарів, нам потрібно було б інтегрувати її з нашою базою даних продуктів. По суті, це означає забезпечення того, що

кожна рекомендація, що видається моделлю, може бути відслідкована до конкретного продукту у нашій базі даних;

– використання моделі для особистих рекомендацій. З використанням даних про предпочтєння користувачів і історії покупок, модель має бути спроможна генерувати персоналізовані рекомендації продуктів, які, як очікується, відповідають смакам і потребам користувачів;

– тестування та моніторинг моделі після інтеграції. Після інтеграції, нам потрібно було б ретельно протестувати модель в реальних умовах роботи, переконавшись, що вона працює згідно очікувань і покращує досвід користувача. Додатково, ми маємо використовувати моніторинг для відстеження роботи моделі в часі і корегування її, якщо це потрібно.

#### 5.4 Оцінка результатів інтеграції моделі

Після впровадження моделі ResNet34 в нашу e-commerce систему, однією з наших основних відповідальностей стає її неперервна оцінка та оптимізація. Оцінка продуктової ефективності, включаючи такі показники, як зміни в продажах, конверсії, середньому чеку і повторних покупках, стане ключовим фактором визначення успіху інтеграції.

Більше того, важливість моніторингу роботи моделі не можна недооцінювати.

Застосування спеціалізованих метрик, таких як точність класифікації, повнота та F-міра, допоможе нам відстежувати, наскільки ефективно модель виконує свої прямі обов'язки по ідентифікації та класифікації одягу.

Не менш важливим є зворотній зв'язок від користувачів. Через організацію опитувань та збір відгуків, ми зможемо отримати цінні відомості про те, як користувачі взаємодіють з новою системою рекомендацій, а це, в свою чергу, дозволить нам постійно покращувати систему.

Кінцевим кроком є адаптація моделі на основі отриманих результатів. Цей процес може включати в себе налаштування гіперпараметрів, доопрацювання даних для навчання, або навіть зміну підходу до навчання моделі.

Все це спрямовано на те, щоб забезпечити, що інтеграція моделі ResNet34 не тільки технічно успішна, але й максимально корисна для нашої e-commerce системи.

### 5.5 Доповнення та розвиток системи

Після інтеграції моделі та першого циклу її оцінки та адаптації, ми маємо продовжувати розширювати та удосконалювати систему. Ми маємо розробити стратегію для постійного збагачення даних, що використовуються для навчання моделі. Ми можемо це зробити, зокрема, через активне збирання відгуків від користувачів, систематичне внесення змін у асортимент продуктів та увагу до модних трендів.

Більш того, важливо постійно вдосконалювати алгоритми, на основі яких працює модель. Ми можемо досліджувати нові методи навчання, розробляти більш складні архітектури мережі або використовувати нові типи даних для навчання моделі.

Останнім, але не менш важливим, є забезпечення сумісності нашої системи з майбутніми технологічними тенденціями. Це означає, що ми маємо розробити систему так, щоб вона могла легко адаптуватися до нових форматів даних, нових технологій взаємодії з користувачами або будь-яких інших нововведень, які можуть з'явитися в майбутньому[13].

Цей пункт підкреслює важливість довгострокового планування та постійного розвитку при впровадженні моделі ResNet34 в нашу e-commerce систему.

## ВИСНОВКИ

В ході виконання кваліфікаційної роботи було успішно виконано дослідження щодо ефективності застосування нейронних мереж YOLO3 та ResNet34 у контексті розпізнавання та класифікації одягу для систем електронної комерції. Обидві моделі були детально проаналізовані та протестовані, що дозволило виявити їх сильні та слабкі сторони, а також визначити головні чинники, що впливають на їх роботу. Результати дослідження підтвердили високу придатність цих моделей для вирішення поставленої задачі.

Нейронні мережі YOLO3 та ResNet34 демонстрували високий потенціал у сфері розпізнавання та класифікації одягу в e-commerce системах. В той же час, кожна з моделей мала свої особливі переваги: ResNet34 забезпечував більш високу точність результатів, тоді як YOLO3 виділявся кращою швидкістю роботи. Ці відмінності роблять обидві моделі цінними для різних сценаріїв використання в залежності від специфіки конкретної e-commerce системи.

В процесі дослідження було проведено детальний аналіз архітектури, принципів роботи, переваг та недоліків різних моделей. Такий підхід дозволив глибше зрозуміти особливості та потенціал кожної з моделей, а також визначити ключові аспекти, які можуть впливати на ефективність їх використання в задачах розпізнавання та класифікації одягу в системах електронної комерції. Здобуті в ході аналізу знання можуть слугувати солідною базою для подальших досліджень і вдосконалень в даній області.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Muthee A. The Basics of Genetic Algorithms in Machine Learning. URL: <https://www.section.io/engineering-education/the-basics-of-genetic-algorithms-in-ml/> (Last accessed: 05.04.2023)
2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
3. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT press.
5. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
7. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
8. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
9. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
10. Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

11. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
12. Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.
13. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2921-2929).
14. Lu, J., Hu, J., & Tan, Y. P. (2019). Clothes recognition: from internet images to real-world images. *Machine Vision and Applications*, 30(3), 423-437.
15. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
16. Machine Learning Models Efficiency Analysis for Image Classification Problem / K. Smelyakov et al. 6th International Conference on Computational Linguistics and Intelligent Systems : Conference Proceedings, Gliwice, 12 May 2023.
17. Burtsev, M., Seliverstov, A., & Airapetyan, R. (2019). Deep Learning in E-commerce. In *International Conference on Digital Transformation and Global Society* (pp. 16-27). Springer, Cham.
18. Luo, C., Liu, Z., Tang, J., & Huang, T. S. (2018). Clothes Changing and Retrieval on Internet Photos for E-commerce. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval* (pp. 263-271).
19. Comparative Efficiency Analysis of Gradational Correction Models of Highly Lighted Image / K. Smelyakov et al. 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8–11 October 2019. 2019. URL: <https://doi.org/10.1109/picst47496.2019.9061356> (date of access: 22.05.2023).
20. Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
22. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.
23. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).
24. Cheng, Y., Zhao, L., Wang, J., & Fu, Y. (2019). Recent Advances in Convolutional Neural Network Acceleration. *Neurocomputing*, 324, 11-24.
25. Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345-1359.
26. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).
27. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
28. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587), 484-489.
29. Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2), 157-166.
30. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. (2017, October). Speed/accuracy trade-offs for modern

convolutional object detectors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7310-7311).

31. Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Klingner, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

