Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА МЕТОДУ АНАЛІЗУ ЗОБРАЖЕНЬ В ЕЛЕКТРОННИХ КОЛЕКЦІЯХ ТЕКСТОВИХ ДОКУМЕНТІВ ДЛЯ ВИРІШЕННЯ ПРОБЛЕМИ ВИЯВЛЕННЯ ПЛАГІАТУ ЗОБРАЖЕНЬ
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-19-1

Іщенко О.І.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Яковлева О.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти перший (бакалаврський)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Іщенку Олексію Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка методу аналізу зображень в електронних колекціях текстових документів для вирішення проблеми виявлення плагіату зображень

затверджена наказом університету від 15 травня 2023 року № 474 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 22 травня 2023 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека комп'ютерного зору з відкритим кодом OpenCV, методи кластеризації *k*-means, elbow, DBSCAN, метод зменшення розмірності ознаки t-SNE, навчена згорткова нейронна мережа MobileNetV2 з бібліотеки Keras, зображення з датасету Coco, зображення схем, діаграм, видів екрану, мови програмування Java та Python, інтерактивний блокнот Jupyter Notebook для розробки на мові програмування Python, фреймворк Spring Framework для розробки на мові програмування Java.

4. Перелік питань, що потрібно опрацювати в роботі
1. Сучасний стан питання плагіату графічного контенту в текстових документах.
2. Методи, сучасні бібліотеки та програмні засоби для вирішення задач аналізу зображень.
3. Розробка методу аналізу зображень в колекції текстових документів.
4. Розробка застосунку та практичні дослідження.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми, постановка задачі, розробка методу аналізу графічного контенту в колекції тестових документів, визначення типу зображення, кластеризація датасету зображень, порівняння зображень на основі детекторного підходу, алгоритм аналізу графічного контенту, проектування застосунку, застосування запропонованого методу для аналізу графічного контенту реальної колекції документів.

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
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	10.04.2023	
2	Аналіз завдання, підбір літератури	11.04.23-17.04.23	
3	Аналіз літератури з досліджуваної проблеми	18.04.23-20.04.23	
4	Аналіз бібліотек, необхідних для розробки методу	21.04.23-24.04.23	
5	Розробка методу	25.04.23-10.05.23	
6	Програмна реалізація	11.05.23-20.05.23	
7	Оформлення пояснювальної записки	20.05.23-21.05.23	
8	Перевірка на плагіат	27.05.23	
9	Рецензування	28.05.23	
10	Підготовка презентації та доповіді	29.05.23-30.05.23	
11	Занесення роботи в електронний архів	31.05.23	
12	Попередній захист кваліфікаційної роботи	31.05.23	

Дата видачі завдання 10 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ _____доц. Яковлева О.В._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 60 с., 29 рис., 2 дод., 33 джерела.

ОБРОБКА ЗОБРАЖЕНЬ, КЛАСТЕРИЗАЦІЯ, ПОРІВНЯННЯ ОЗНАК НА ОСНОВІ ГІСТОГРАМ, ДЕТЕКТОР SIFT, МЕТОД NNDR, МЕТОД RANSAC.

Об’єктом роботи є питання виявлення плагіату зображень в електронних колекціях текстових документів.

Метою роботи є розробка методу аналізу зображень в електронних колекціях текстових документів для вирішення проблеми виявлення плагіату зображень.

Проведено роботу щодо графічного контенту колекції текстових документів. Використано методи обробки зображень, кластеризації, порівняння ознак на основі гістограм, пошуку відповідностей точок, отриманих з використанням детектору SIFT, за допомогою методів NNDR та RANSAC.

У результаті роботи здійснена програмна реалізація системи для проведення дослідження.

IMAGE PROCESSING, CLUSTERING, HISTOGRAM-BASED FEATURE COMPARISON, SIFT DETECTOR, NNDR METHOD, RANSAC METHOD.

The object of the work is the question about detecting plagiarism of images into text document electronic collections.

The aim of the work is to develop analysis methods for images into text document electronic collections for solution of detecting image plagiarism.

Work was carried out on the graphic content of the collection of text documents. Were used image processing, clustering, histogram-based feature comparison, finding points accordance, that has been received with using SIFT detector, with the help of NNDR and RANSAC methods.

As a result of implemented software implementation of the system for doing research.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 7
Вступ 8
1 Сучасний стан питання плагіату графічного контенту в текстових документах 9
1.1 Проблема плагіату та успіхи в вирішенні задачі виявлення плагіату 9
1.2 Існуючі сервіси для вирішення задачі пошуку схожих зображень 11
1.3 Методи вирішення задачі пошуку схожих зображень 15
1.3.1 Класичні підходи 15
1.3.2 Нейромережеві підходи 16
1.3.3 Датасети для навчання нейронних мереж 17
1.4 Сучасні бібліотеки та програмні засоби для вирішення задач аналізу зображень 19
1.5 Постановка задачі 20
2 Розробка методу аналізу зображень в колекції текстових документів 23
2.1 Визначення типу зображення 23
2.1.1 Формування датасету із схем та фотозображень 23
2.1.2 Перенавчання моделі MobileNetV2 26
2.2 Кластеризація датасету зображень 28
2.2.1 Отримання ознак на основі гістограм з урахуванням області розташування пікселів 28
2.2.2 Метод *K*-means 29
2.2.3 Використання підходу Elbow для визначення кількості кластерів для методу *K*-means 30
2.2.4 Метод DBSCAN 31
2.2.5 Використання методів зменшення розмірності для візуалізації результатів кластеризації 32
2.3 Використання детектору SIFT для отримання ознак ключових точок зображення 34

2.4 Алгоритм аналізу графічного контенту колекції текстових документів..... 38
2.5 Використання методу аналізу зображень в колекції текстових документів для вирішення задачі виявлення зображень, що є підозрілими на плагіат……....39
2.5.1 Розробка критерію щодо визначення зображення підозрілим на плагіат……39
2.5.2 Розробка алгоритму щодо виявлення зображень, підозрілих на плагіат……41
3 Розробка застосунку та практичні дослідження 43
3.1 Налаштування програмного середовища 43
3.2 Проєктування застосунку..... 44
3.3 Застосування запропонованого методу щодо аналізу кваліфікаційних робіт магістрів 45
3.3.1 Створення колекції текстових документів..... 45
3.3.2 Формування колекції зображень..... 47
3.3.3 Проведення аналізу графічного контенту..... 47
3.3.4 Висновок щодо аналізу графічного контенту..... 50
Висновки 52
Перелік джерел посилання..... 54
Додаток А Кластеризація на основі ознак за гістограмами..... 57
Додаток Б Тестові зображення 59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

СТЗ – система технічного зору

ІП – інваріантна пряма

HMM – Hidden Markov Model (прихована модель Маркова)

SIFT – Scale-Invariant Feature Transform (масштабно-інваріантна трансформація ознак)

DBSCAN – Density-base Spatial Clustering of Application with Noise (базована на густині просторова кластеризація для застосунків з шумами)

РО – розпізнавання образів

НМ – нейронна мережа

МН – машинне навчання

ОЗ – обробка зображень

ВСТУП

Плагіат – привласнення чужого твору, наукової роботи або винаходу. Із прискоренням розвитку технологій час отримання інформації зменшився з декількох днів до декількох секунд. Із полегшенням отримання інформації стало легше й прибігати до плагіату.

Прибігання до плагіату та невирішення цієї проблеми може привести до знецінення старань багатьох науковців, авторів творів, художників та зменшить швидкість розвитку рівня культури та науки.

На даний момент існують сервіси, які перевіряють текстові документи на наявність плагіату. Проблема цих сервісів у тому, що вони перевіряють тільки текстову частину, ігноруючи графічну. Тому проблему плагіату можна вважати частково вирішеною.

Вирішити проблему відсутності перевірки графічної частини на плагіат можливо, прибігнувши до методів та алгоритмів в області розпізнавання образів, обробки зображень та штучного інтелекту, за допомогою яких можливо порівнювати зображення.

Актуальність проблеми полягає в існуванні сервісів пошуку подібних зображень та відсутності спеціалізованих сервісів, які були б здатні аналізувати документ та перевіряти графічну частину на наявність плагіату.

Робота присвячена дослідженню графічного змісту електронної колекції текстових документів та виявленню в неї зображень, підозрілих на плагіат з використанням нейромережевого підходу (для класифікації зображень) та класичних піходів, які включають в себе попереднє порівняння зображень за ознаками на основі гістограм з ціллю скорочення списку схожих зображень та використання дескрипторного підходу із застосуванням методів пошуку відповідних точок, після яких можна зробити висновок, чи є зображення підозрілим на плагіат. Запропонований метод був досліджен на реальній колекції текстових документів.

1 СУЧАСНИЙ СТАН ПИТАННЯ ПЛАГІАТУ ГРАФІЧНОГО КОНТЕНТУ В ТЕКСТОВИХ ДОКУМЕНТАХ

1.1 Проблема плагіату та успіхи в вирішенні задачі виявлення плагіату

Створення спеціалізованих архівів для зберігання наукових робіт має велике значення у розвитку технологій не тільки в інформаційній, а й в усіх галузях. Це дало змогу зменшити час отримування потрібного матеріалу. Теперь користувач замість декількох годин, або навіть днів витратить лише декілька хвилин. Електронні архіви дають змогу не витрачати час на шлях від робочого місця до бібліотеки, пошук книжкових шаф, які відносяться до потрібної теми, пошук потрібної літератури в одній або декількох шафах, замість чого користувач на сторінці електронного архіву вводить у поле пошуку назву матеріалу, в результаті чого зможе обрати потрібний матеріал серед робіт, які надав сервіс, на обрану тему.

Хоча створення даних архівів і сприяє подальшому розвитку галузі, ці сервіси можуть нести й проблеми. Головна проблема, яку створили дані сервіси, це збільшення кількості плагіату. Плагіат – це привласнення чужого твору, винаходу, відкриття, ідеї або використання матеріалу з іншої роботи без посилання на автора.

Роботи з плагіатом так само як і інші роботи будуть впливати на галузі, але з різних боків. Нові роботи без плагіату будуть розвивати галузь, а роботи з використанням плагіату сприятимуть створенню великої кількості однотипних робіт зі схожим змістом без розвитку думки. Це приведе до зменшення швидкості розвитку галузей, до яких написані роботи, а згодом – розвиток може зупинитися.

Тому, для збереження темпу розвитку галузей, важливим буде вирішити дану проблему. Оскільки інформаційні технології в останній час розвиваються дуже швидко, вони дозволяють вирішувати будь-які проблеми душе швидко.

Проблема плагіату в наукових роботах не вийняток. Для вирішення цієї проблеми були створені сервіси для виявлення плагіату в наукових роботах.

Серед багатьох сервісів пошуку на плагіат можна виділити декілька популярних: Dupli Checker [1], Plagiarisma [2], Grammarly [3], Search Engine Report Plagiarism Checker [4], Paperrater [5], Edubirdie [6], Plagium [7]. Дані сервіси безкоштовні та мають функцію пошуку на плагіат як основну або додаткову. Для пошуку плагіату, ці сервіси за допомогою певних алгоритмів аналізу текстової частини документу та порівняння його із вмістом інших робіт з різних ресурсів проводить пошук абзаців, розділів або речень, які фігурують в інших текстах або у такому ж вигляді, або з подозрою на рерайтінг цих речень. Рерайтінг – перефразування текстового матеріалу або його фрагменту зі збереженням змісту. Тобто ці сервіси не просто шукають фрагменти тексту роботи, а й передбачають можливість перефразування тексту. Таким чином, після аналізу текстової частини документу, сервіс видає фрагменти тексту, які мають співпадіння із текстом з інших матеріалів, або мають підозру на рерайтінг.

Дані сервіси мають велике значення у написанні якісних та унікальних наукових робіт, але все ж вони не ідеальні. Як було зазначено, вони перевіряють текстову частину, але ігнорують графічну. Плагіат може полягати в привласненні не тільки текстів (віршів, ідей тощо), а й графічного матеріалу (схем, намальованих картин художником). Тому для подальшого вирішення проблеми плагіату важливим буде аналізувати не тільки текстову, а й графічну частини документу.

У вік технологій було створено велику кількість засобів для роботи з зображеннями. Ці засоби існують як і окремі застосунки, так і бібліотеки із функціями та класами, які можуть бути додані у проєкт та запрограмовані у новій програмі. Для вирішення проблеми плагіату зображень можна використати не тільки один метод, а комбінувати декілька засобів. Таким чином, можна використати нейронну мережу у поєднанні з функціоналом бібліотеки комп’ютерного зору.

1.2 Існуючі сервіси для вирішення задачі пошуку схожих зображень

На даний момент існує багато сервісів, які працюють із зображеннями. В їх функціонал входять пошук зображень за текстом, пошук схожих зображень, пошук зображень за фрагментом зображення, розпізнавання кольорової гами на зображенні, розпізнавання об’єктів на зображенні та багато інших. Для вирішення проблеми плагіату зображень слід розглянути сервіси, які дозволяють робити пошук схожих зображень. Серед найпопулярніших сервісів можна виділити Google Lens [8]. Для користування цим сервісом слід натиснути на іконку об’єктива (рис. 1.1).

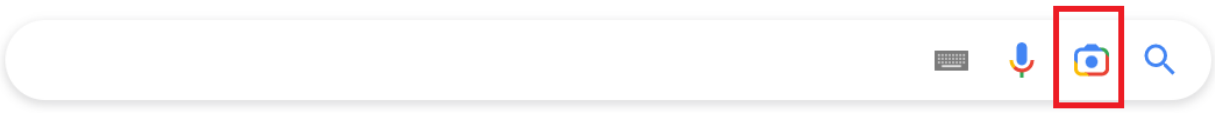


Рисунок 1.1 – Іконка гугл об’єктива у полі пошуку Google

Далі треба завантажити зображення та побачити схожі зображення (рис. 1.2).

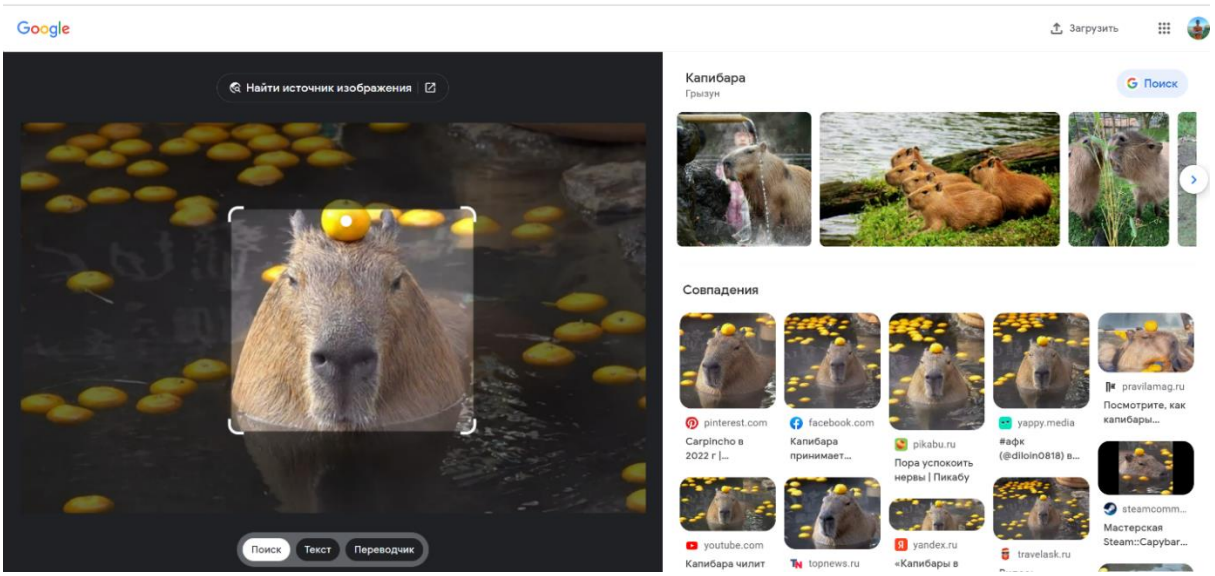


Рисунок 1.2 – Результат пошуку за зображенням у Google Lens

Слід відмітити, що накладання фільтрів на зображення не заважатимуть даному сервісу знайти схожі оригінальні зображення (рис. 1.3).

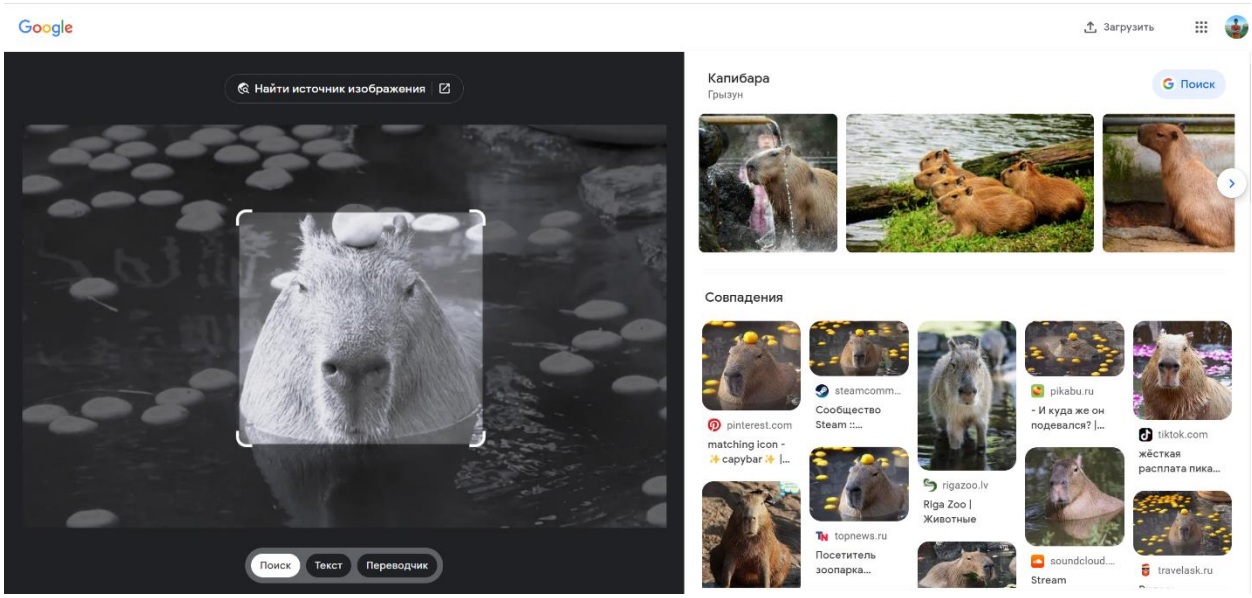


Рисунок 1.3 – Результат пошуку за зображенням з накладеним фільтром «чорно-білий» у Google Lens

Ще один сервіс пошуку схожих зображень надає бізнес-гігант Microsoft в якості пошукової системи Bing [9]. За схожістю функціоналу та принципом роботи її можна зазначити як альтернативу пошуковій системі Google. Як із Google Lens, для початку роботи із пошуком за зображеннями, слід натиснути на іконку об’єктиву (рис. 1.4).

Рисунок 1.4 – Іконка об’єктива у полі пошуку Microsoft Bing

Microsoft Bing має схожий принцип роботи із пошуковою системою Google, але на відміну від Google, даний сервіс надає декілька популярних зображень, які можна використати для першого запиту (рис. 1.5).

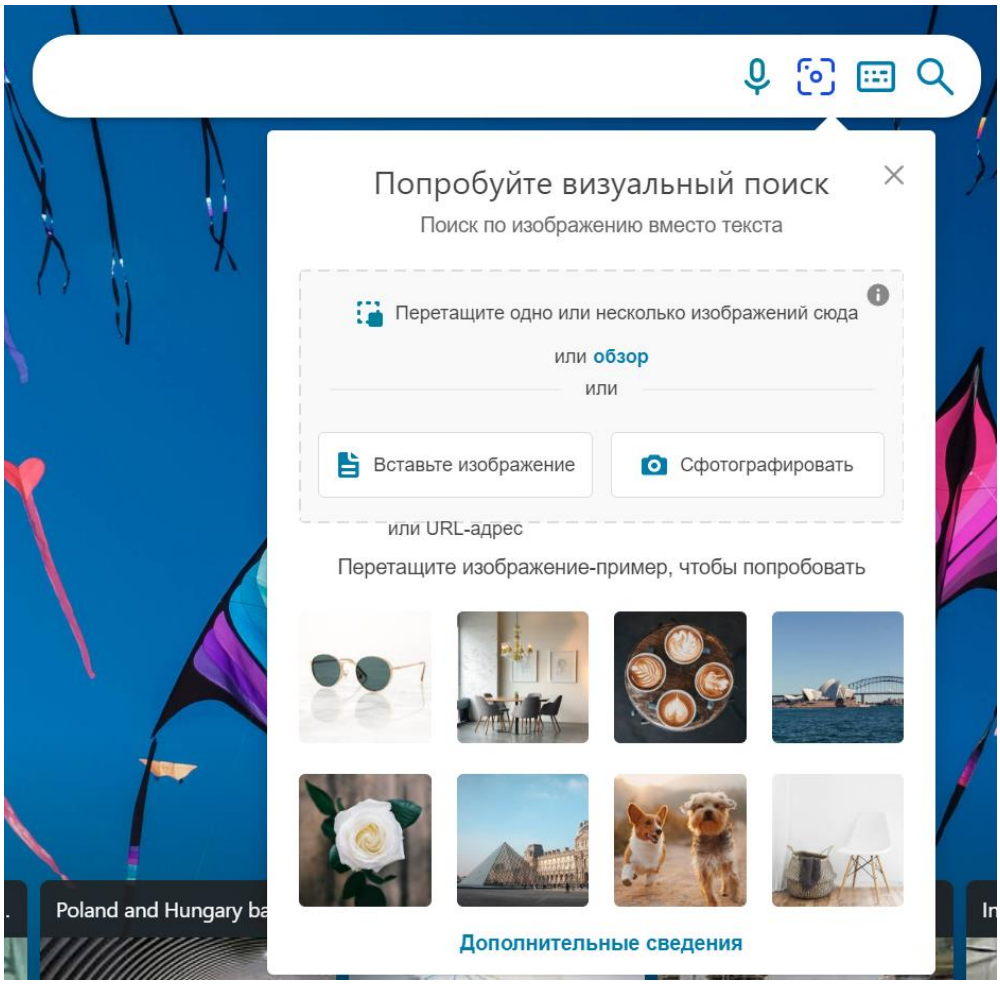


Рисунок 1.5 – Вікно вибору зображення для пошуку схожих зображень

Якщо обрати одне з запропонованих зображень, користувач може побачити результат пошуку (рис. 1.6).

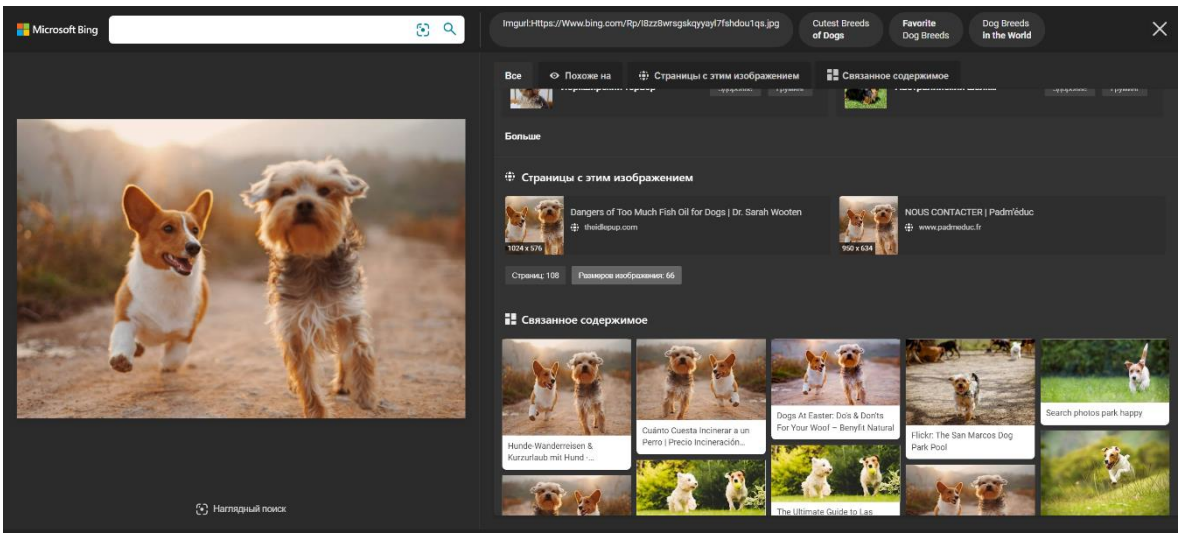


Рисунок 1.6 – Результат пошуку зображення у Microsoft Bing

Окрім вбудованих у пошукову систему функцій пошуку за зображеннями, існують також спеціалізовані сервіси для пошуку схожих зображень. Один з таких сервісів – TinEye [10]. Даний сервіс відрізняється від двох раніше вказаних тим, що він спеціалізується на пошуку тільки за зображеннями та створений компанією, яка не так відома, як Google або Microsoft. Результат роботи даного сервісу можна побачити нижче (рис. 1.7).

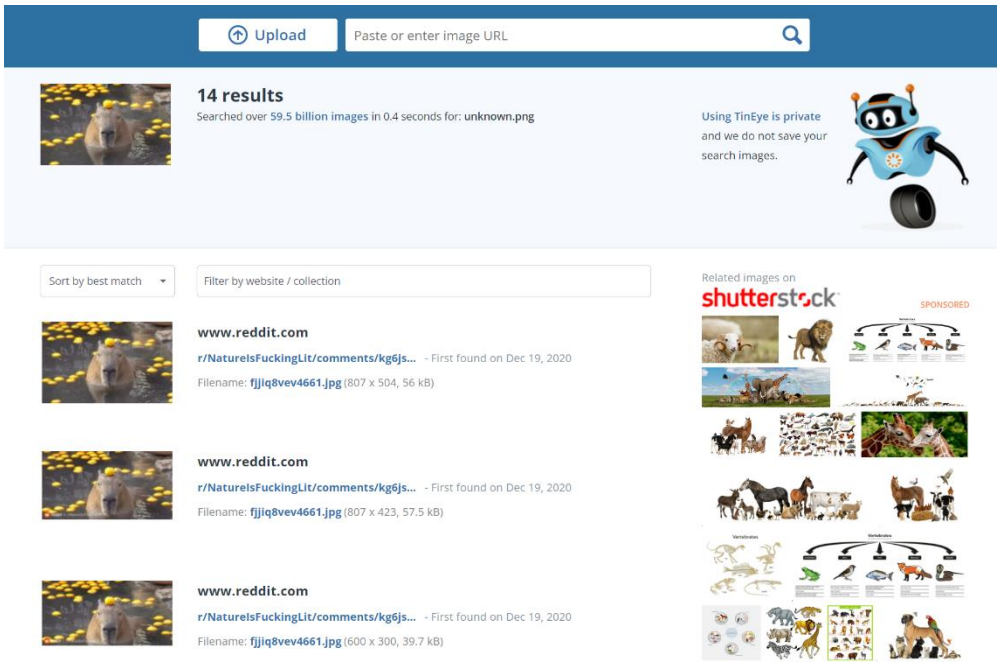


Рисунок 1.7 – Результат пошуку зображення у TinEye

Дані сервіси здатні досить точно знаходити схожі зображення, але вони не мають достатнього функціоналу для пошуку плагіату за зображеннями. На даний момент не існує спеціалізованого сервісу, який здатен перевірити графічну частину текстового документу та підвести підсумки щодо кількості зображень, які були віднесені до групи підозрілих на плагіат.

1.3 Методи вирішення задачі пошуку схожих зображень

1.3.1 Класичні підходи

Для людського ока будь-яке зображення – це композиція кольорів, які нагадують певні об’єкти, місця, природні явища тощо. Порівняти будь-які зображення людина може лише поглянувши та порівнявши ці зображення. І хоча цифрові зображення – це набір пікселів, для комп’ютера ці пікселі нічого не значать, тому такий простий спосіб, як візуально порівняти зображення, для комп’ютера недоступний.

Для реалізації методу порівняння зображення комп’ютером необхідно вивести ознаки та властивості, які можна порівняти. Оскільки можливі варіанти, коли зображення може бути нахилене, масштабоване або із накладеним фільтром, але в результаті ці зображення повинні бути схожими, недоцільним буде порівнювати колір кожного пікселю або розміри зображень. Для реалізації порівняння зображення можна прибігнути до функціоналу комп’ютерного зору.

Комп’ютерний зір – це дисципліна, яка охоплює створення технологій та систем, які отримають інформацію у вигляді мультимедія. Одним з підрозділів комп’ютерного зору, який працює із зображеннями як із об’єктами, є розпізнавання образів. Даний підрозділ охоплює класифікацію та ідентифікацію предметів, явищ, процесів та об’єктів, які охарактеризовані певною кількістю властивостей та ознак.

Один із способів реалізувати функцію порівняння зображень – за допомогою дескрипторів. Дескриптор – це об’єкт, який містить інформацію про візуальну складову зображення. Спосіб порівняння зображень за допомогою дескрипторів передбачає створення декількох важливих точок на зображеннях із деякими характеристиками, після чого характеристики цих точок порівнюються між заданими зображеннями. Цей спосіб передбачає те, що зміщення, нахил, масштаб та яскравість є інваріантними характеристиками зображень, тобто вони незалежні від усіх властивостей. Порівняння зображень

за дескрипторами – непоганий спосіб, який не потребує ідеальних умов для порівняння зображень та виконується без виконання багатьох непотрібних розрахунків.

1.3.2 Нейромережеві підходи

Для отримання більш продуктивного методу, окрім використання класичних піходів, буде реалізований ще один – з використанням штучних нейронних мереж. Штучні нейронні мережі – це обчислювальні системи, які відображають роботу мозку живих істот. Дані системи здатні навчатися за прикладами, не потребуючи програмування мережі спеціально під задачу. Нейронна мережа складається з нейронів та зв’язків між ними. Дані нейрони приймають на вхід дані (найчастіше – дійсні числа), виконують нелінійну функцію та віддають результат у наступний шар. Існує три типи шарів нейронної мережі – шар входу, прихований шар та шар виходу. У будь-якій нейронній мережі може бути тільки один шар входу та один шар виходу. Прихованих шарів може бути безліч. Окрім змінної кількості шарів, у кожному шарі нейронної мережі може бути змінна кількість нейронів. Кількість шарів впливає не тільки на точність результату, а й на час розрахунку, тому при роботі із нейронними мережами важливо найти баланс у точності та часі розрахунків.

При використанні нейронних мереж важливим буде обрати тип мережі, яку треба навчати. Мережі різних типів використовуються при розв’язку задач з різних областей та потребують різні способи навчання нейронної мережі. Для розв’язку задач розробки методу аналізу зображень підійде згорткова нейронна мережа. Згорткова нейронна мережа – це нейронна мережа, яка відноситься до класу глибинних нейронних мереж. Нейронні мережі даного типу використовуються для рішення задач в області розпізнавання образів.

Згорткова нейронна мережа просторово інваріантна. За основу роботи даної мережі взята схема з’єднання нейронів зорової кори.

Для навчання нейронної мережі буде застосовано спосіб навчання з учителем. Навчання з учителем – це спосіб навчання нейронної мережі, при якому нейронна мережа примусово навчається за рахунок навчальних прикладів, які складають масив об’єктів «стимул-реакція». Ціль навчання нейронної мережі – визначити реакцію для прикладів, які не входять у масив навчальних прикладів мають тільки стимул. У задачі аналізу зображень для навчання на вхід нейронній мережі будуть приходити приклади, які складаються з пари «зображення-клас». Ціль навчання нейронної мережі – знайти залежність, яка допоможе відрізняти графічні дані зображень від схем та задавати їм відповідний клас.

Таким чином, використання підходу класифікації зображень за допомогою згорткової нейронної мережі дозволить зробити метод аналізу зображень більш оптимізованим. За рахунок наявності у графічних даних класів, кількість прикладів, які можуть співпадати з вхідним зображенням, для порівняння зменшиться, що приведе до збільшення швидкості аналізу при незмінній точності аналізу.

1.3.3 Датасети для навчання нейронних мереж

Навчання нейронної мережі з вчителем полягає у наданні мережі початкових даних, у яких вказан клас та значення. Підвищити точність нейронної мережі можна надавши велику кількість коректних вхідних даних з правильно заданими класами та великою кількістю прикладів для усіх випадків.

Для навчання нейронної мережі для аналізу зображень існує велика кількість датасетів. Вони містять багато зображень з прив’язаними до них класами. Ці датасети відрізняються наборами класів, які містяться у датасеті,

кількістю зображень, котрі здатні надати датасети, та цільовими задачами, які здатна вирішувати нейронна мережа, навчена за тим чи іншим датасетом.

Одним з найпопулярніших датасетів є ImageNet. ImageNet – це база даних зображень, який містить понад 14 мільйонів екземплярів. Даний датасет призначений для тестування методів розпізнавання образів та комп'ютерного зору. Датасет містить зображення, які описані класами та об'єктами, які знаходяться або не знаходяться на зображення. Даний датасет доповнювався за краудсорсинговою системою, тобто зображення та анотації до них додавалися великою кількістю людей безкоштовно. Даний датасет підійде для тренування нейронних мереж для вирішування задач класифікації.

Ще один датасет, який здатний навчити нейронну мережу для класифікації зображень, називається Coco. Даний датасет має менше зображень у базі, ніж попередній, але він активно доповнюється та підтримується такими бізнес гігантами, як Facebook та Google. У базі даних Coco міститься понад 330 тисяч зображень, 80 категорій об'єктів та 5 описів до кожного зображення. Окрім задач класифікації, даний датасет підійде для навчання нейронних мереж для вирішення задач сегментації зображень.

Існують датасети, які фокусуються на зображеннях з людьми та їх обличчям. Моделі, навченні за цими датасетами, здатні розрізняти обличчя або проводити підрахунок кількості людей у натовпі. Для навчання нейронної мережі для детекції обличчя можна використовувати датасет Wider Face. Даний датасет має приблизно 32 тисячі зображень із 393 тисячами осіб, на яких обличчя зображено під різним масштабом, ракурсом, освітленням та кутом повороту обличчя, які розділені у 61 класи. Для навчання нейронної мережі у підрахунку людей у натовпах існують такі датасети, як Shanghai Tech та UCF-CC-50. Shanghai Tech у своїй базі має приблизно тисячу зображень з зазначеними 330 тисячами особами, а UCF-CC-50 спеціалізується на навчаннях мереж для підрахунку людей у дуже великих натовпах, оскільки цей датасет має тільки 50 зображень, але кількість осіб на зображенні може бути до 4,5 тисячі.

Для отримання більшої точності у вирішенні задачі необхідно зібрати датасет, екземпляри якого будуть близькими до реальних даних.

1.4 Сучасні бібліотеки та програмні засоби для вирішення задач аналізу зображень

Розробка методу аналізу зображень із документів не може бути обмеженою використанням тільки базових функцій мови програмування. Тому для розробки було обрано декілька бібліотек, які будуть використані для розробки певної частини методу.

Основна бібліотека, яка буде використовуватися у розробці методу – Open CV [11]. Open CV – це бібліотека з відкритим вихідним кодом. Вона містить методи комп'ютерного зору, обробки зображень та чисельних алгоритмів. Після оновлення до версії 2.2 бібліотека була розділена на декілька невеликих бібліотек, які містять функції для своєї області. Так, дана бібліотека була розділена на бібліотеки для обробки зображень, UI виводу зображень та відео, моделей машинного навчання, розпізнавання та опису дескрипторів, аналізу руху та відслідковування об'єктів, калібровки камери, швидкого пошуку ближніх сусідів та відокремлені бібліотеки, які містять застарілий код та деякі оптимізовані функції за рахунок використання певних технологій. Розділ великої бібліотеки на декілька маленьких та використання однієї з багатьох бібліотек дозволить зробити кінцевий застосунок менш громіздким, який буде займати менше місця на диску. У даному випадку, для розробки методу аналізу зображень, які містяться у документах потрібним буде використання бібліотеки, яка містить функціонал для роботи з дескрипторами.

Метод аналізу зображень передбачає використання не тільки дескрипторного підходу, а його комбінування із методом гістограм та класифікацією графічних даних на клас зображення або схеми. Останній метод

потребує використання штучного інтелекту. Для цього можна використати бібліотеку TensorFlow [12]. TensorFlow – це бібліотека для машинного навчання. Дана бібліотека виконує задачі створення, навчання та тренування нейронної мережі з ціллю класифікації образів. Використання даної бібліотекою у поєднанні із тестовими даними дозволить створити потужну систему, яка може розпізнати графічні дані як зображення або схему.

Для роботи із штучним інтелектом підійде бібліотека Keras [13]. Keras – це бібліотека для мови програмування Python, яка була створена для взаємодії з нейронними мережами. Дана бібліотека підтримує нейромережу бібліотеку TensorFlow, якою керує на верхньому рівні. Також, дана бібліотека містить готову модель MobileNetV2 [14], яка може мати як стандартну структуру, так і приєднувати до себе іншу структуру та поведінку для верхніх шарів нейронної мережі. MobileNetV2 – це згорткова нейронна мережа із глибиною 53 шари. Дана нейронна мережа має вбудовану функцію навчання за допомогою датасету ImageNet – великої бази даних зображень, яка складається з 14 мільйонів екземплярів та використовується для навчання мереж для вирішення задач класифікації. Навченна нейронна мережа може класифікувати зображення за тисячами класів.

Поєднання бібліотек для комбінування методу аналізу зображення дозволить отримати оптимізований та точний метод. Оптимізація комбінованого методу буде отримуватися за рахунок відкидання даних, які були віднесені до іншого класу та попереднього використання більш простого підходу з використанням гістограм, що дозволить уникнути зайвих порівнянь важливих точок за дескрипторним підходом.

1.5 Постановка задачі

На основі проведеного вище аналізу можна зробити висновок, що питання плагіату графічного контенту текстових документів є невирішеним на сьогоднішній день актуальним завданням.

Об’єктом роботи є питання виявлення плагіату зображень в електронних колекціях текстових документів.

Метою роботи є розробка методу аналізу зображень в електронних колекціях текстових документів для вирішення проблеми виявлення плагіату зображень.

Для досягнення цієї мети необхідно вирішити такі теоретичні та практичні завдання:

– оглянути сучасний стан питання пошуку плагіату, а також існуючі сервіси пошуку плагіату;

– розглянути існуючи методи комп’ютерного зору для пошуку та порівняння зображень;

– вивчити бібліотеки та програмні засоби для аналізу зображень;

– розробити метод для аналізу графічного контенту електронних колекцій текстових документів;

– вивчити основні принципи згорткових нейронних мереж, їх навчання та перенавчання;

– перенавчити вже навчену нейрону нейронну мережу для класифікації зображень (фотозображення або схема);

– створити датасет для перенавчання нейронної моделі;

– вивчити питання отримання ознак зображення, які дають загальний опис зображення;

– вивчити методи кластеризації колекції зображень на основі загальних ознак та обрати найбільш підходящий для кластеризації на основі гістограм зображень;

– провести практичні дослідження графічного контенту колекції наукових звітів та кваліфікаційних робіт бакалаврів та магістрів;

– вивчити питання отримання ознак зображення, які дають опис ключових точок зображення;

– розробити критерій щодо прийняття рішення про підозру на плагіат;

– розробити алгоритм виявлення зображення, підозрілого на плагіат, для нового документу, що завантажується до колекції;

– спроєктувати застосунок для аналізу зображень колекції текстових документів та перевірки графічного контенту нового документу на наявність зображень, що є підозрілі на плагіат, та розробити прототип застосунку;

– зробити висновки щодо запропонованого методу аналізу зображень в електронних колекцій текстових документів.

2 РОЗРОБКА МЕТОДУ АНАЛІЗУ ЗОБРАЖЕНЬ В КОЛЕКЦІЇ ТЕКСТОВИХ ДОКУМЕНТІВ

2.1 Визначення типу зображення

Для визначення типу зображення необхідно створити модель та навчити за відповідним датасетом. Віднесення зображення до певного типу дозволить робити менше переборів при порівнянні ознак зображень, оскільки порівнювати зображення різних типів не має сенсу.

2.1.1 Формування датасету із схем та фотозображень

Для отримання потужної моделі з використанням навчання з вчителем, яка буде здатна видавати правильні висновки майже в усіх випадках, необхідно використовувати великий датасет, в якому для кожного випадку буде велика кількість екземплярів.

Для формування екземплярів для навчання моделі використовувався датасет Coco. Дана вибірка має 1100 екземплярів: 550 для фотозображень та 550 для схем. Для підвищення якості навчання усі зображення мають розмір 224 x 224 пікселів. На фотозображеннях можуть бути як люди (рис. 2.1), так і тварини (рис. 2.2) або різні предмети (рис. 2.3).

Окрім фотозображень, датасет сформований також і з схем. Схеми – це зображення, які демонструють інформацію у різних виглядах. Декілька способів, якими схеми можуть демонструвати інформацію (рис. 2.4): блок-схемами; графіками функцій; гістограмами або діаграмами; код програми; налаштування середовища або деяких параметрів; короткий опис деякої технології.

а) б)

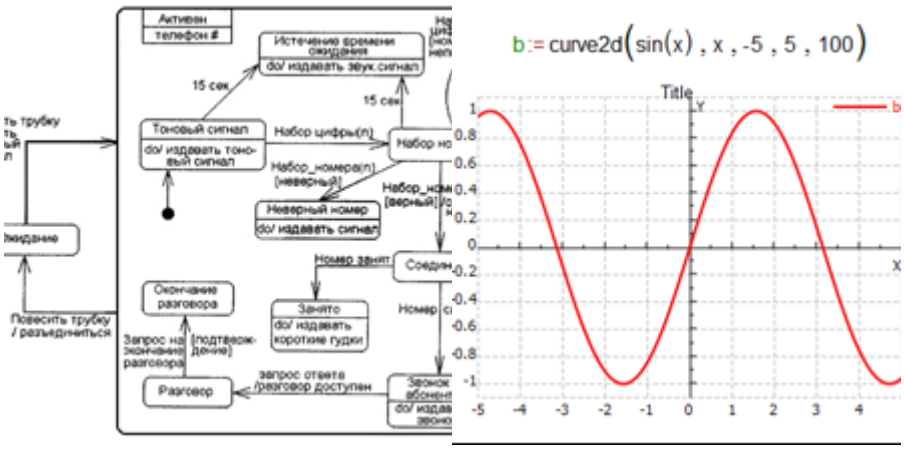
Рисунок 2.1 – Приклад фотозображень із людинами з датасету Coco:
а) приклад 1; б) приклад 2

а) б) в)

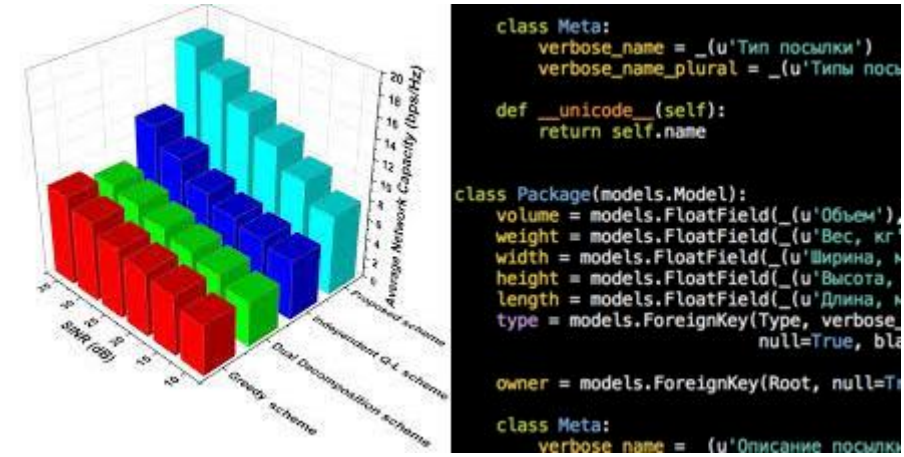
Рисунок 2.2 – Приклад зображень із тваринами з датасету Coco:
а) приклад 1; б) приклад 2; в) приклад 3

а) б) в)

Рисунок 2.3 – Приклад зображень із предметами:
а) приклад 1; б) приклад 2; в) приклад 3



а) б)



в) г)

ґ) д)

Рисунок 2.4 – Приклади зображень схем:

а) блок-схема; б) графік функції; в) гістограма; г) код програми;

ґ) налаштування; д) короткий опис методології

Таким чином, підбір правильного датасету буде сприяти навчанню більш потужної та якісної моделі, яка буде вірно класифікувати майже усі зображення. Для більш якісної роботи моделі важливим буде включати в датасет не зовсім вдалі зображення, але які б повинні бути прийняті моделлю.

2.1.2 Перенавчання моделі MobileNetV2

Перенавчання моделі – це процес, при якому перша частина моделі (вхідний та частину внутрішніх шарів) може бути навчена за одним правилом, а верхня частина – за іншим. MobileNetV2 [14] пропонує готову структуру нейронної мережі, яка може бути швидко створена та навчена для виконання задач комп'ютерного зору. Оскільки верхній шар даної мережі розподіляє зображення на велику кількість класів, її поведінку потрібно змінити. Для цього створюється модель MobileNetV2 без верхнього шару. Потім створюється об'єкт, який містить структуру верхніх шарів, після чого вони об'єднуються в одну модель. Вхідні шари та вигляд моделі MobileNetV2 до об'єднання продемонстровано на рисунку 2.5.

Після приєднання до моделі MobileNetV2 верхніх шарів, структуру верхніх шарів моделі показано на рисунку 2.6.

Як можна побачити на рисунку, до моделі MobileNetV2 було приєднано декілька шарів. Приєднані шари містять 2 внутрішніх шара та один вихідний. Внутрішні шари складаються з 128 нейронів, але їх кількість при розрахунках може зменшитися до 64, що зменшує вплив окремого нейрону на результати моделювання та робить модель більш стійкою та гнучкою для нових вхідних даних. Вихідний шар має задачу підводити підсумки та повертати результат приналежності зображення до одного з двох класів.

Таким чином, для виконання завдання класифікації зображень на два класи була створена модель MobileNetV2 із доданими верхніми шарими із специфічною для поточного завдання поведінкою. Таким чином, точність

побудованої та навченої моделі за валідаційними даними складає 1 (100%), за тренувальними – 0,99 (99%), а за тестовими – 0,97 (97%). Точність навченої моделі на тестових даних показано на рисунку 2.7.

Model: "mobilenetv2_1.00_224"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	[]
Conv1 (Conv2D)	(None, 112, 112, 32)	864	['input_2[0][0]']
bn_Conv1 (BatchNormalization)	(None, 112, 112, 32)	128	['Conv1[0][0]']
Conv1_relu (ReLU)	(None, 112, 112, 32)	0	['bn_Conv1[0][0]']

а)

block_16_depthwise_relu (ReLU)	(None, 7, 7, 960)	0	['block_16_depthwise_BN[0][0]']
block_16_project (Conv2D)	(None, 7, 7, 320)	307200	['block_16_depthwise_relu[0][0]']
block_16_project_BN (BatchNorm alization)	(None, 7, 7, 320)	1280	['block_16_project[0][0]']
Conv_1 (Conv2D)	(None, 7, 7, 1280)	409600	['block_16_project_BN[0][0]']
Conv_1_bn (BatchNormalization)	(None, 7, 7, 1280)	5120	['Conv_1[0][0]']
out_relu (ReLU)	(None, 7, 7, 1280)	0	['Conv_1_bn[0][0]']

б)

Рисунок 2.5 – Модель MobileNetV2:

а) вхідні шари; б) останні шари

Conv_1 (Conv2D)	(None, 7, 7, 1280)	409600	['block_16_project_BN[0][0]']
Conv_1_bn (BatchNormalization)	(None, 7, 7, 1280)	5120	['Conv_1[0][0]']
out_relu (ReLU)	(None, 7, 7, 1280)	0	['Conv_1_bn[0][0]']
flatten (Flatten)	(None, 62720)	0	['out_relu[0][0]']
dense (Dense)	(None, 128)	8028288	['flatten[0][0]']
dropout (Dropout)	(None, 128)	0	['dense[0][0]']
dense_1 (Dense)	(None, 2)	258	['dropout[0][0]']

Рисунок 2.6 – Верхні шари створеної моделі

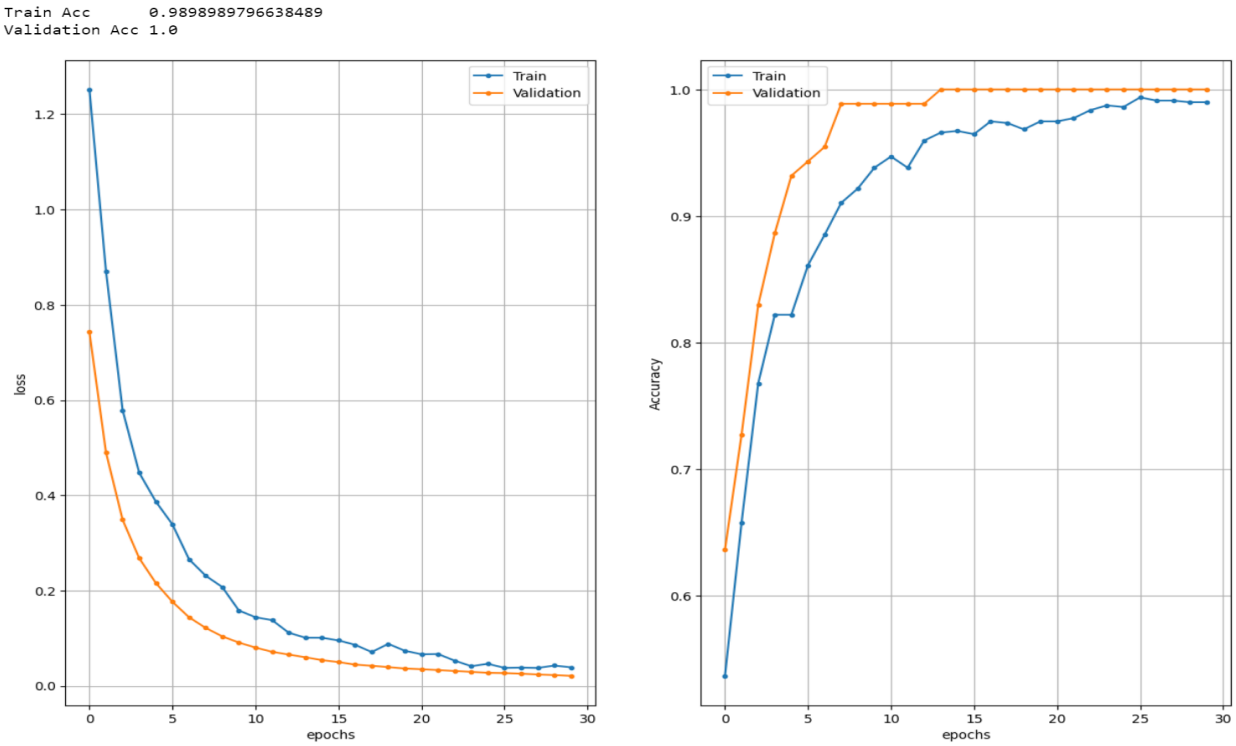


Рисунок 2.7 – Результат навчання моделі

2.2 Кластеризація датасету зображень

2.2.1 Отримання ознак на основі гістограм з урахуванням області розташування пікселів

Для оптимізації роботи порівняння зображень можна робити попереднє порівняння зображень за ознаками на основі гістограм. Гістограма – це вид діаграми, яка представлена у вигляді стовпців.

У задачі отримання ознак, гістограма буде демонструвати кількість пікселів, які належать до кожної групи яскравостей, де горизонтальна ось гістограми представляє групи яскравостей, а вертикальна – кількість пікселів, які належать до тієї чи іншої групи.

Проблемою використання ознак на основі гістограм може бути наступне: фотозображення з великою гамою кольорів будуть отримувати зовсім різні ознаки на основі гістограм відносно інших зображень, але схематичні зображення будуть мати майже схожі ознаки, оскільки більшість

схем представлені чорним текстом на білому фоні. Тому для вирішення цієї проблеми можна зібрати ознаки чотирьох частин зображення та об’єднати їх в один вектор, який буде використовуватися у подальшому аналізу.

2.2.2 Метод *K*-means

Алгоритм *k*-середніх (*k*-means) – один з алгоритмів машинного навчання, який виконує задачу кластеризації (рис. А.1, А.2). Також цей алгоритм використовують для створення фільтрів при розпізнаванні зображень. Даний алгоритм є ітераційним та має велику популярність через свою простоту. Основна задача алгоритма – зменшення сумарного квадратичного відхилення точок кластерів від центроїдів даних кластерів. Даний алгоритм може використовувати різні відстані, але найчастіше за всього використовується Євклидова відстань:

$$\rho(x, y) = \sqrt{\sum_{p=1}^{n}(x_p - y_p)^2}, \tag{2.1}$$

де $x, y \in R^n$.

Робота алгоритму починається з зазначення кількості кластерів k та центроїдів μ_i кожного кластеру.

Алгоритм *k*-means працює за наступною схемою:

Крок 1. Зазначається кількість кластерів k та центроїдів μ_i кожного кластеру.

Крок 2. Кожна точка відноситься до кластеру, до центроїду якого вона має найменшу відстань.

Крок 3. Розраховуються нові центроїди за наступною формулою:

$$\mu_j = \frac{1}{S_j} \sum_{x^{(j)} \in S_i} x^{(j)}. \quad (2.2)$$

Крок 4. Робота алгоритму продовжується до тих пір, поки не виконується вираз $\mu_i^{крок\ t} = \mu_i^{крок\ t+1}$ (тобто алгоритм зупиняє свою роботу, якщо центроїдами залишилися ті самі точки), інакше – перехід до Кроку 2.

Таким чином, в основі алгоритму полягає розрахунок центроїдів кожного кластеру та перенос точок до інших кластерів у кожній ітерації, за рахунок чого досягається розподіл кожної точки до потрібного кластеру. Недоліком цього алгоритму можна вважати те, що результат та точність роботи алгоритму залежить від вхідних даних. Даний недолік вирішується використанням удосконаленого алгоритму *k*-means++, який передбачає вибір більш оптимальних початкових значень для роботи алгоритму, або використання алгоритму, який виконується для розрахунку потрібної кількості кластерів.

2.2.3 Використання підходу Elbow для визначення кількості кластерів для методу *K*-means

Більшість алгоритмів кластеризації не передбачають розрахунку кількості кластерів, а отримують це значення як вхідне. Неправильно підібрана кількість кластерів може призвести до отримання неоптимальної моделі, тому важливим буде використати методи для розрахунку кількості кластерів, при яких модель буде найоптимальніша.

Одним з таких алгоритмів є Elbow. Даний алгоритм передбачає розрахунок суми квадратів відстаней від точок до центроїдів кластерів при різних кількостях кластерів. Далі за кількістю кластерів та сумою квадратів відстаней від центроїдів до точок будується графік. Цей графік має вигляд гіперболи (рис. 2.8).

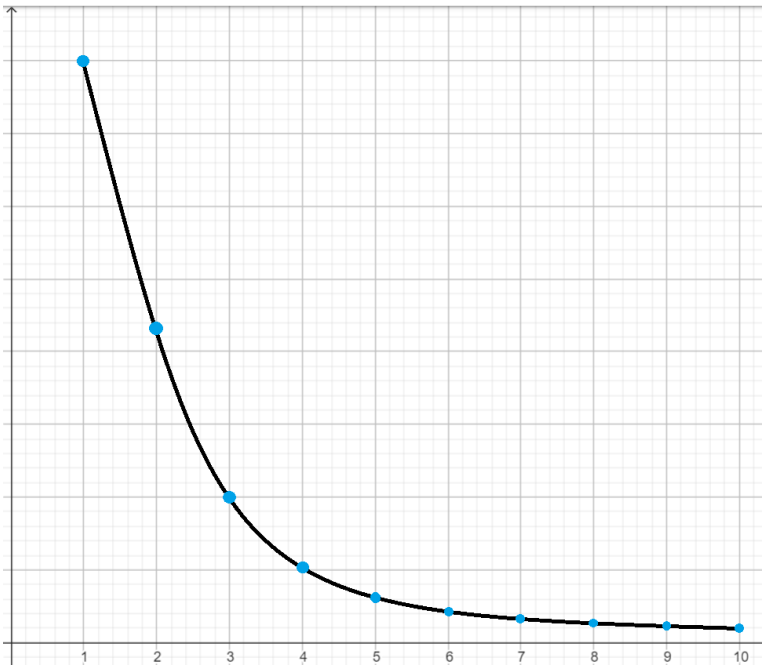


Рисунок 2.8 – Графік залежності суми квадратів відстаней точок від центру свого кластеру від кількості кластерів

За цим графіком виявляється точка, після якої зменшення суми відстаней відбувається за лінійною залежністю. За рисунком вище, це може бути точка із аргументом функції 4, яка буде називатися «ліктем», оскільки має найостріший вигин лінії, тому у даній моделі найбільш оптимальною кількістю кластерів буде 4.

Недоліком даного методу є те, що даний метод суб'єктивний, тому ненадійний. На графіку моделі може не бути острого вигину, тому вибір «ліктя» буде ускладнений.

2.2.4 Метод DBSCAN

Існує інший метод кластеризації даних, який сильно відрізняється від алгоритму *k*-means – це DBSCAN (рис. А.3). DBSCAN – це алгоритм кластеризації даних, який заснований на щільності, тобто він групує точки, які знаходяться на близькій відстані одне від одного. Даний алгоритм приймає на вхід 2 параметри, серед яких не треба вказувати кількість кластерів, на відміну

від *k*-means, алгоритм сам розрахує кількість кластерів, на які розподілить точки.

Алгоритм поведінки методу DBSCAN виглядає наступним чином:

Крок 1. Зафіксувати на вході 2 параметри: ϵ – радіус області (найчастіше за все – Евклідове) та minPoints – мінімальна кількість точок, які повинні створювати щільну область.

Крок 2. Обирається точка та перевіряється кількість точок, які знаходяться у зоні з радіусом ϵ від неї. Якщо ця кількість більша за minPoints, то утворюється кластер, інакше – точка відмічається як шум. Якщо ця точка вже належить до кластеру, то усі точки в області ϵ також належать до цього кластеру.

Крок 3. Проводити минулі Кроки, доки не будуть перевірені усі точки.

Перевагою даного алгоритму є те, що він не потребує у вхідних даних вказувати кількість кластерів, у які необхідно розподілити точки, що не призведе до погіршення моделі за рахунок фіксованої кількості кластерів. Ще одною перевагою алгоритму є те, що він може розподілити точки у кластери довільних форм, можуть бути випадки, коли один кластер оточений іншим. Окрім цього, даний алгоритм потребує малу кількість вхідних даних та не залежить від порядку точок.

Окрім переваг цей алгоритм має і недоліки. Результат алгоритму залежить від порядку перевірки точок, тобто одна точка може належати до декількох кластерів при різних послідовностях перевірки точок.

2.2.5 Використання методів зменшення розмірності для візуалізації результатів кластеризації

Кожне зображення складається з великої кількості пікселів, кожен з яких має колір. Якщо розкласти зображення на набір гістограм, воно буде представлено набором найчастіше з 8-16 чисел, тобто такої кількості класів гістограми, яка ефективно описує зображення. Така розмірність підійде для

кластеризації зображень, але за такою кількістю значень неможливо візуалізувати результат кластеризації. Для візуалізації результату кластеризації, точки якої складаються з великої розмірності даних, використовують алгоритми, які зменшують розмірність до такої, яку можна представити на графіку.

Одним з таких алгоритмів є t-SNE. Стохастичне вкладення сусідів із *t*-розподілом (*t*-distributed Stochastic Neighbour Embedding) – це алгоритм машинного навчання для візуалізації даних. Даний алгоритм використовує техніку нелінійного зменшення розмірності, який заносить дані великої розмірності у низькорозмірний простір (2 або 3 виміри). Алгоритм розподіляє точки таким чином, щоб схожі точки знаходилися поруч, а несхожі – з великою імовірністю знаходилися поодаль одне від одного. Робота алгоритму проводиться у наступному порядку:

Крок 1. Задаємо початковий набір даних: набір багатовимірних точок $X = \{x_1, x_1, \ldots, x_n\}$; відхилення σ для кожної точки x_i ; кількість ітерацій T; швидкість навчання μ та коефіцієнт інерції $\alpha(t)$.

Крок 2. Розрахувати умовну схожість $p_{j|i}$ між точками за формулою:

$$p_{j|i} = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)}. \quad (2.3)$$

Крок 3. Розрахувати спільну схожість p_{ij}, яка розраховується наступним чином:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}. \quad (2.4)$$

Крок 4. Розташувати точки $Y = \{y_1, y_2, .., y_n\}$ за нормальним розподілом. Точки Y– це аналоги точок *X,* які мають менше вимірів та можуть бути розташовані на площині або просторі.

Крок 5. Почати першу (наступну) ітерацію покращення результату розташування точок *Y.* Розрахувати спільну схожість q_{ij} для точок Y аналогічно до спільної східності p_{ij} для точок *X.*

Крок 6. Розрахувати градієнт наступним чином:

$$\frac{\partial Cost}{\partial y_i} = 4\sum_j (p_{ij} - q_{ij})\,(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}. \qquad (2.5)$$

Крок 7. Перерахувати значення для точок Y для поточної ітерації t, використовуючи значення градієнту, значень точок з попередніх двох ітерацій та інших параметрів:

$$Y(t) = Y(t-1) + \mu\frac{\partial Cost}{\partial y_i} + \alpha(t)[Y(t-1) - Y(t-2)]. \qquad (2.6)$$

Крок 8. Якщо відбулася остання ітерація – робота алгоритму закінчується, інакше – починається наступна ітерація з переходом на Крок 5.

2.3 Використання детектору SIFT для отримання ознак ключових точок зображення

Одним з способів порівняння зображень є порівняння ознак ключових точок зображень. Для отримання ознак ключових точок зображення необхідно використовувати дескриптори. Одним з таких дескрипторів є Scale-Invariant Feature Transform (SIFT). SIFT – це алгоритм, який використовують для виявлення ознак ключових точок зображень. Робота алгоритма починається з виявлення цих точок. Для цього зображення згортається за фільтрами Гауса у

різних масштабах, після чого розраховується різниця послідовно розмитих за Гаусом зображень. Ключовими точками будуть максимум/мінімум різниці гаусианів. Різниця гаусианів розраховується за наступною формулою:

$$D(x, y, \delta) = L(x, y, k_i\delta) - L\big(x, y, k_j\delta\big), \tag{2.7}$$

де $L(x, y, k\delta)$ – згортка зображення $I(x, y)$ з розмиттям за Гаусом $\mathrm{G}(x, y, k\delta)$ у масштабі $k\delta$.

Згортка зображення розраховується наступним чином:

$$L(x, y, k\delta) = I(x, y) * \mathrm{G}(x, y, k\delta). \tag{2.8}$$

Отримані точки є ключовими точками зображення, але деякі з них можуть бути нестікими та їх потрібно викинути. Тому наступним кроком буде відкидання точок з низьким контрастом. Необхідно провести розрахунок інтерполяції за квадратичним розкладанням Тейлора функції різниці Гаусианів масштабного простіру із кандидатом у ключові точки за наступною формулою:

$$D(x) = D + \frac{\partial D^T}{\partial x}x + \frac{1}{2}x^T\frac{\partial^2 D}{\partial x^2}x, \tag{2.9}$$

де D та її похідна розраховуються у точці-кандидаті, а $\mathrm{x} = (x, y, \delta)^T$ – зміщення цієї точки.

Місце розташування екстремуму $\hat{x}$ розраховується взяттям першої похідної за x. Якщо зміщення $\hat{x}$ більше ніж 0,5, це означає що екстремум знаходиться ближче до іншого кандидату у ключові точки та для неї треба провести інтерполяцію. В іншому випадку це зміщення додається до кандидату у ключові точки для отримання оцінки місця екстремума.

Далі слід відкинути точки з низьким контрастом, яке розраховується за розкладанням Тейлора другого порядку $D(x)$ зі зміщенням $\hat{x}$. Якщо це

значення менше за 0,03, то ця точка відкидається, інакше – вона зберігається із місцем знаходження у просторі $y + \hat{x}$, де y є початковим місцем знаходження ключової точки.

Функція різниці гаусианів може мати потужні значення вздовж ребер, навіть якзо кандидат не має стійкості від невеликого шуму. Тому для підвищення стабільності слід відкидати точки, які мають великий внесок від ребер, але мають погано визначене місцезнаходження. Знаходження цих точок відповідає знаходження власних значень значень матриці Гессе другого порядку H:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}. \quad (2.10)$$

Для деякого порогового відношення власних значень r_{th}, якщо відношення власних значень R для кандидата у ключові точки більше, ніж $\frac{(r_{th}+1)^2}{r_{th}}$, то ця точка відкидається, оскільки вона має невдале місце знаходження. Як правило, використовують значення порогового відношення $r_{th} = 10$.

Після отримання набору ознак ключових точок необхідно розрахувати орієнтацію для отримання інваріантності за обертанням. Для розрахунку необхідно взяти розмите за Гаусом зображення $L(x, y, \delta)$ в ключових точках з масштабом δ. Таким чином, усі розрахунки будуть відбуватися у манірі, яка буде інваріантною за масштабом. Далі для зображення розаховується значення орієнтації $\theta(x, y)$ та градієнту $m(x, y)$:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}, \quad (2.11)$$

$$\theta(x,y) = atan2\big(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y)\big). \quad (2.12)$$

Розрахунок значень градієнта та орієнтації проводяться для кожного пікселя навколо ключової точки у розмитому за Гаусом зображенні L. Далі формується гістограма направлень, яка складається з 36 областей, у якій кожна область покриває 10 градусів. Кожна точка, яка додається до області гістограми, зважена за величиною градієнта та по гаусово-зваженому круговому вікну з δ, яке у півтора рази більше за масштаб ключової точки. У цих гістограмах можуть бути наявні домінуючі напрями – це напрями, які відповідають піковим значенням гістограми. Існують високі та локальні піки, які існують у межах 80% від високих піків. Вони призначаються ключовій точці. Якщо призначаються декілька направлень, то створюється додаткова ключова точка, яка має таке ж саме місце розташування та масштаб, як і оригінальна точка для кожного додаткового напрямку.

Далі відбувається розрахунок векторів дескрипторів для кожної ключової точки таким чином, щоб даний дескриптор був часткого інваріантним до освітлення, точки огляду тощо. Для цього в спочатку створюється набір гістограм на 4×4 сусідніх пікселях з 8 областями у кожній. Ці гістограми розраховуються із значень величини та орієнтації елементів в області 16×16 навколо ключової точки. Далі ці величини зважуються функцією Гауса з δ, яка рівна половині ширини вікна дескриптора. Після цього дескриптор стає вектором усіх значень гістограм та має 128 елементів, оскільки він має 16 (4×4) гістограм із 8 областями у кожній.

Для покращення результатів порівняння вектор можна нормалізувати. Спочатку його нормалізують до одиничної довжини для досягнення інваріантності змінам в освітленні. Для зменшення ефекту нелінійного освітлення застосовують значення порогу 0,2 та вектор знову нормалізують. Дане значення було обрано емпірично, що означає, що спеціально розраховане значення порогу може покращити результати порівнянь.

На теперішній час, детектор SIFT є одним з найпотужніших засобів для виконання задач порівняння зображень. Даний детектор здатний видавати

результат з великою точністю (понад 90%) на рівні найпотужніших та добре навчених нейронних мереж.

2.4 Алгоритм аналізу графічного контенту колекції текстових документів

Аналіз графічного контенту колекції текстових документів відбувається наступним чином:

Крок 1. На вхід подається колекція текстових документів.

Крок 2. Відбувається завантаження першого (поточного) документу у базу даних.

Крок 3. Відкривається документ та зчитуються зображення, які знаходяться у документі у базу даних.

Крок 4. Аналізуються зображення на наявність плагіату, використовуючи поєднання різних алгоритмів фільтрації, класифікації, кластеризації та порівняння.

Крок 5. Якщо залишилися документи, графічний зміст яких не був проаналізований – переходимо до Кроку 2, інакше – Крок 6.

Крок 6. Проводяться підсумки та демонструється статистика, яка містить у собі наступні пункти:

– загальна кількість проаналізованих документів;

– загальна кількість проаналізованих зображень;

– середня кількість зображень у документах;

– мінімальна та максимальна кількості зображень у документі;

– кількість (%) зображень, які є плагіатом;

– кількість зображень, які мають клас схем та фотозображень;

– кількість зображень, які не пройшли фільтрацію та їх середня кількість по всім документам.

Аналіз графічного змісту документу проводиться, порівнюючи зображення із зображеннями тільки з інших документів. Зображення може не пройти попередню фільтрацію, якщо воно буде мати розмір не більше ніж 50 пікселів або якщо воно буде складатися тільки із білого або чорного кольорів (не валідне зображення).

2.5 Використання методу аналізу зображень в колекції текстових документів для вирішення задачі виявлення зображень, що є підозрілими на плагіат

2.5.1 Розробка критерію щодо визначення зображення підозрілим на плагіат

Для аналізу зображення на плагіат необхідно визначити критерії, за виконання яких це зображення буде вважатися підозрілим. Для деякого зображення P_0 можна зробити висновок, що воно є плагіатом, якщо виконуються усі умови. Якщо поточна умова не виконується – виконувати подальший аналіз не має сенсу.

Умова 1. Дана умова базується на порівнянні ознак зображень на основі гістограм. Для цього необхідно розрахувати дані ознаки для зображення P_0, яке проходить аналіз на плагіат та отримати ознаки з бази даних для колекції зображень $P = \{P_1, P_2, \ldots, P_n\}$. Ознаки на основі гістограм складаються з 10 груп відтінків, на основі яких буде розраховуватися міра схожості з використанням нормованої Манхетенської відстані:

$$q(H_0, H_i) \leq \sigma_1, \tag{2.13}$$

де $H_0 = (h_{0_1}, \ldots, h_{0_N})$, $H_i = (h_{i_1}, \ldots, h_{i_N})$ – гістограми зображення, над яким проводиться аналіз, та зображень з бази даних;

N – кількість груп гістограм (використовується значення 10);

σ_1 – порогове значення у порівнянні гістограм, використовується значення 0,05;

$q()$ – міра схожості заданих гістограм, яка описується наступною формулою:

$$q(H_0, H_i) = \frac{\sum_{j=1}^{N} \left| h_{0_j} - h_{i_j} \right|}{N * S}, \quad (2.14)$$

де S – розмір зображення (для порівняння ознак використовується розмір зображення 600×400).

Порогове значення $\sigma_1 = 0{,}05$ характеризує те, що кольорові гами заданих зображень є близькими та необхідно проводити подальший аналіз та перевіряти виконання наступних умов. В результаті, після перевірки умови 1, сформується підмножина зображень $P' \subset P$, для яких виконується умова 1 при аналізі з зображенням P_0. Якщо множина пуста – дане зображення не є підозрілим на плагіат, інакше – необхідно перейти до перевірки наступної умови.

Умова 2. Далі використовується метод NNDR, який шукає відповідності між зображеннями та задовольняє умові:

$$NM > \sigma_2, \quad (2.15)$$

де NM (number of matches) – кількість знайдених відповідностей між зображеннями за допомогою методу NNDR;

σ_2 – порогове значення у пошуку відповідностей за допомогою методу NNDR, використовується значення 40.

Усі зображення, які задовольняють цій умові, сформують множину $P'' \subset P'$. Якщо ця множина пуста – зображення не є плагіатом.

Умова 3. На даному етапі проводиться порівняння кількості пар ознак, які були отримані за допомогою методів NNDR та RANSAC. Залишаються ті

відповідності, які були визнані істиними за методом RANSAC. Зображення буде вважатися підозрілим при виконанні умови:

$$\frac{NI}{NM} > \sigma_3, \tag{2.16}$$

де NI (number of inliers) – кількість вірних відповідностей за методом RANSAC;

σ_3 – порогове значення частки знайдених відповідностей за допомогою методу NNDR від кількості істиних відповідностей, отриманих за допомогою методу RANSAC, використовується значення 75%.

Таким чином, зображення буде вважатися підозрілим на плагіат, якщо ознака на основі гістограм близька до одного чи декількох зображень з бази даних, між даним зображенням буде знайдено 40 відповідностей до іншого зображення та понад 75% відповідностей будуть визнані істиними. В результаті буде сформований набір зображень $M''' \subset M''$, які схожі до зображення, яке проходило аналіз.

2.5.2 Розробка алгоритму щодо виявлення зображень, підозрілих на плагіат

Процес аналізу графічного вмісту документу на наявність підозрілих на плагіат зображень включає в себе комплекс дій по виконанню фільтрації та методів над зображеннями. Процес виконується за наступним алгоритмом:

Крок 1. Завантажими перше (наступне) зображення.

Крок 2. Перевірити зображення на валідність. Висота та ширина зображення повинні бути більше за 50 пікселів, не бути абсолютно білим або чорним.

Крок 3. Провести класифікацію зображення за допомогою моделі та зберегти клас.

Крок 4. Сформувати ознаки на основі гістограм.

Крок 5. Віднести зображення до підкласу на основі отриманої ознаки.

Крок 6. Провести порівняння зображення за ознаками на основі гістограм. Ознаки зображень, які не задовольняють критерію (2.13) вважаються не підозрілими на плагіат та можна перейти до Кроку 9.

Крок 7. Знайти відповідні точки за допомогою методу NNDR та перевірити виконання критерію (2.15). Якщо даний критерій не виконується – перейти до Кроку 9.

Крок 8. За допомогою методу RANSAC відкинути відповідні точки, які не є істиними, після чого перевірити виконання критерію (2.16). Якщо даний критерій виконується – дане зображення можна вважати підозрілим на плагіат.

Крок 9. Якщо зображення не є останнім, то перейти до Кроку 1, інакше – Крок 10.

Крок 10. Підвести статистику за документом.

3 РОЗРОБКА ЗАСТОСУНКУ ТА ПРАКТИЧНІ ДОСЛІДЖЕННЯ

3.1 Налаштування програмного середовища

Для розробки методу аналізу графічного контенту колекції текстових документів використовувалася низка технологій та застосунків.

Перший застосунок, який використовувався при створенні основної логіки методу – IntelliJ Idea [15]. Це інтегроване середовище для розробки, яке підтримує такі мови програмування, як Java, Groovy, Javascript, PHP тощо. Дане середовище було застосовано для створення методу на мові програмування Java 11 версії з застосуванням фреймворку Spring Framework 2.1.6.RELEASE [16].

Для роботи зі штучним інтелектом використовувалася мова програмування Python із застосування Jupyter Notebook [17]. Дане середовище має перевагу в тому, що код програми можна розділити на декілька незалежних блоків з кодом, які можна запускати у будь-якому порядку. Окрім цього, існують текстові блоки, якими можна описати один чи декілька блоків з кодом.

У розробці використовувалася база даних Postgres [18]. Дана база була піднята в якості контейнеру Docker [19]. Docker – це програмне забезпечення для автоматизації розгортання та управління застосунками. Окрім розгортання бази даних, дане програмне забезпечення може також піднімати контейнер із сервером FTP (SFTP), пустий контейнер або будь-який інший, який може бути створений розробником.

Для роботи з клієнтською частиною, яки виводить статистику, використовувалося середовище для розробки WebStorm [20]. WebStorm – це середовище для розробки від розробників середовища IntelliJ Idea, яке використовують у веброзробці.

3.2 Проєктування застосунку

Для проведення аналізу був автоматизований метод аналізу графічного контенту електронної колекції текстових документів на базі вебзастосунку [21, 22]. Даний застосунок складається з клієнтської та серверної частини, бази даних та моделі [23-29], яка класифікує зображення. Структура застосунка продемонстрована на рисунку 3.1.

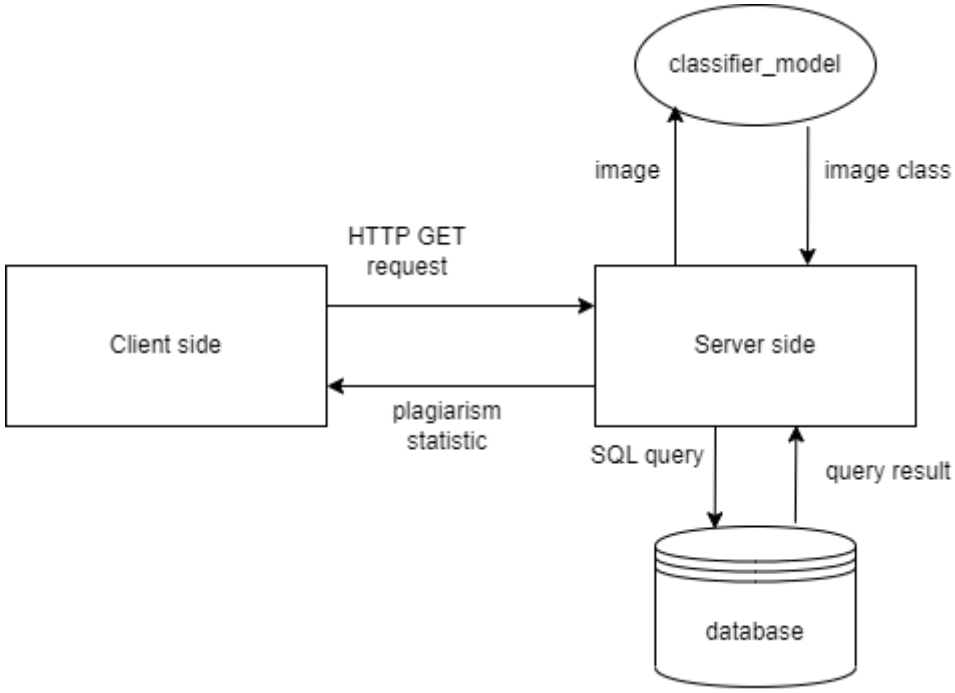


Рисунок 3.1 – Структура застосунку

Аналіз починається з команди з клієнтської частини шляхом надсилання HTTP запиту на сервер. Далі сервер обробляє запит та виконує його, звертаючись до моделі та надсилаючи запити до бази даних та виконуючи прописаний код на своїй частині. Після обробки запиту серверна части повертає відповідь (response), яку клієнтська частина обробляє та виводить на екран у вигляді статистики. Статистика на клієнтській частині буде показуватися так, як показано на рисунку 3.2.

Аналіз зображень в колекції текстових документів на наявність плагіату зображень		
Загальна кількість документів:	0	(100%)
Загальна кількість зображень:	0	(100%)
Середня кількість зображень у роботі:	0.00	(100%)
Максимальна/мінімальна кількість зображень у роботі:	0/0	
Кількість зображень, які не пройшли попередню фільтрацію (за розміром та кольором):	0	(0%)
Середня кількість зображень, які не пройшли попередню фільтрацію (за розміром та кольором):	0.00	(0%)
Кількість схем:	0	(0%)
Кількість фотозображень:	0	(0%)
Кількість зображень, підозрілих на плагіат:	**0**	**(0%)**

Рисунок 3.2 – Форма клієнтського застосунку для демонстрації статистики

3.3 Застосування запропонованого методу щодо аналізу кваліфікаційних робіт магістрів

3.3.1 Створення колекції текстових документів

Для проведення аналізу графічного контенту колекції текстових документів необхідно завантажити дані документи на локальний диск. Колекція текстових документів містила у собі кваліфікаційні роботи магістрів кафедри Інформатики Харківського Національного Університету Радіоелектроніки, які розташовані на відкритому архіві університету (рис. 3.3).

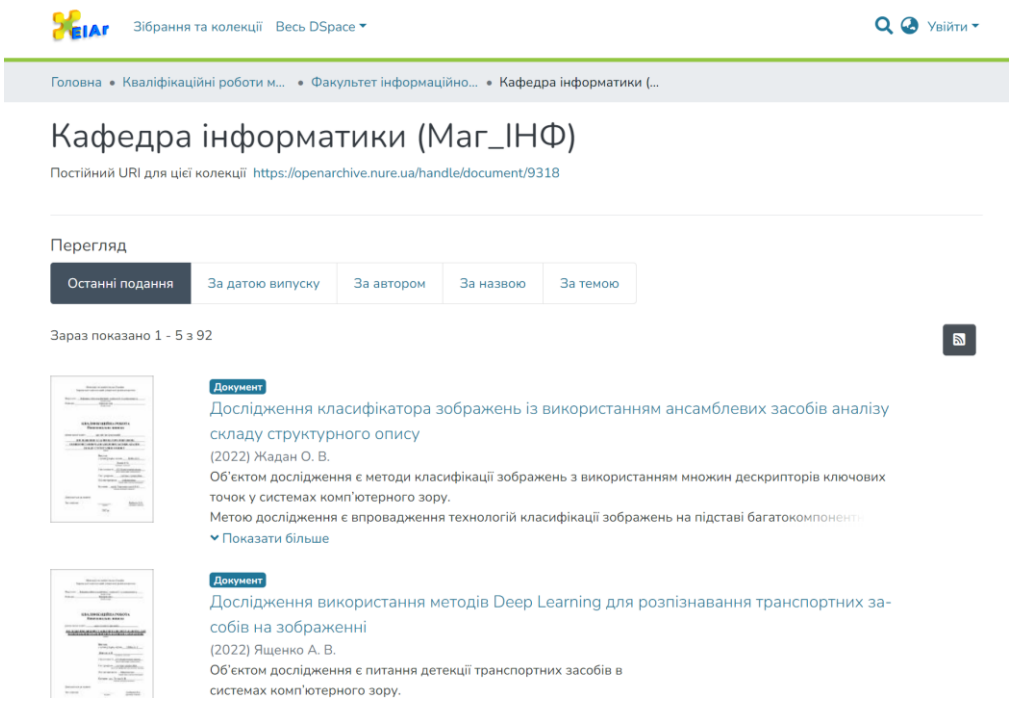


Рисунок 3.3 – Сторінка списку робіт магістрів кафедри інформатики у відкритому архіві ХНУРЕ

Для завантаження роботи слід нажати на назву текстового файлу під зображенням титульної сторінки документу у вкладці документу. Сторінка документу у відкритому архіві зображена на рисунку 3.4.

Рисунок 3.4 – Сторінка документу у відкритому архіві ХНУРЕ

Таким чином, була сформована колекція для проведення аналізу графічного зображення, де кожен документ був названий у форматі «порядковий номер від 1_унікальний ідентифікатор файлу».

3.3.2 Формування колекції зображень

Формування колекції зображень проводилося за наступним алгоритмом:

Крок 1. Відкрити перший (наступний документ).

Крок 2. Дістати перший (наступний) об’єкт.

Крок 3. Якщо даний об’єкт є зображення, то його необхідно зберегти на диску, інакше – пропустити об’єкт.

Крок 4. Якщо залишился об’єкти в документі – перейти до Кроку 2, інакше – Крок 5.

Крок 5. Якщо залишился документи, з яких не було завантажено зображень – перейти до Кроку 1, інакше – закінчити роботу алгоритма.

3.3.3 Проведення аналізу графічного контенту

Аналіз алгоритму починається з класифікації зображення, в результаті якої зображення буде віднесене до класу фотозображень або схем. Використання даного підходу дасть змогу зменшити кількість переборів при порівнянні ознак зображень, що оптимізує метод аналізу графічного контенту електронної колекції текстових документів. Робота класифікації продемонстрована на рисунку 3.5 та рисунках Б.1-Б.4.

Далі порівнюються ознаки на основі гістограм з ознаками, збереженими у базі даних, у рамках свого класу. Для отримання інваріантності за кольором на зображення накладається сірий фільтр та створюються гістограми. Для підвищення ефективності можна розділити зображення на 4 частини та

створити гістограми для усіх його частин. Таким чином, для заданого зображення (рис. 3.6 (а)) так буде виглядати гістограма за 4 частинами зображення (рис. 3.6 (б)) та за цілим зображенням (рис. 3.6 (в)).

Далі на зображенні шукаються важливі точки за детектором SIFT [30, 31], за якими шукаються співпадіння на зображеннях. Детектор SIFT має велику точність пошуку, яка досягає рівня найпотужніших нейронних мереж для порівняння зображень, тому час виконання може бути досить великим, тому перед порівнянням зображень за даним детектором було проведено класифікацію та порівняння за ознаками на основі гістограм, які за короткий час здатні відсіяти більшу частину зображень, які точно не будуть схожі.

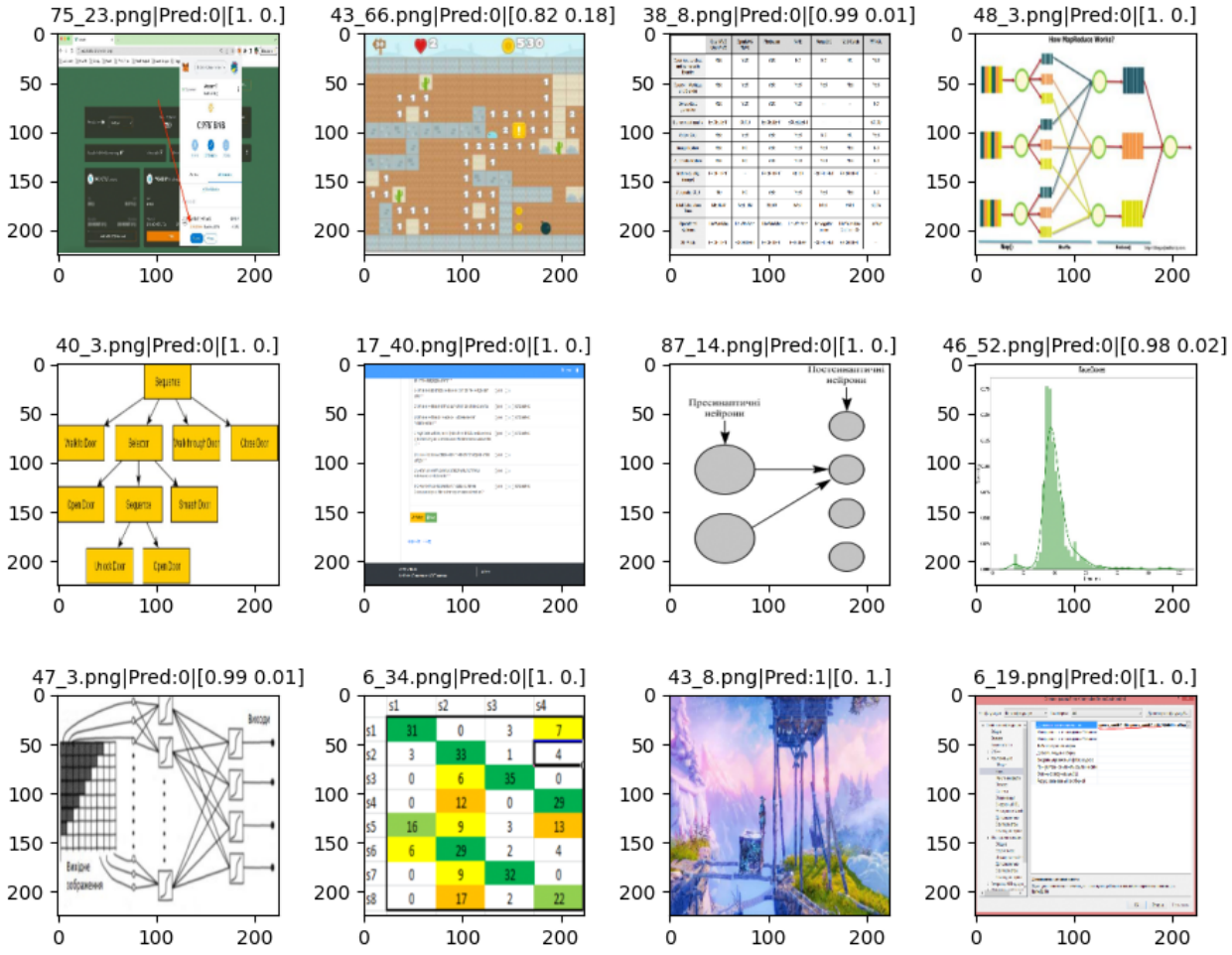
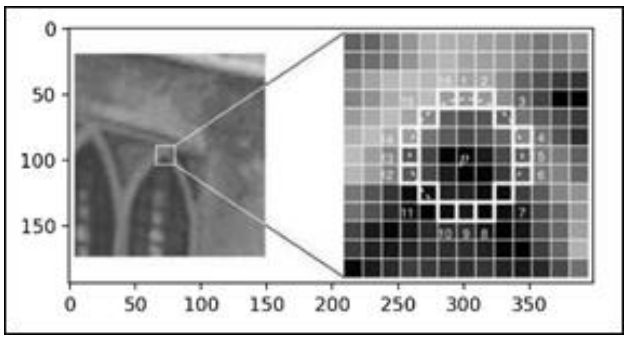


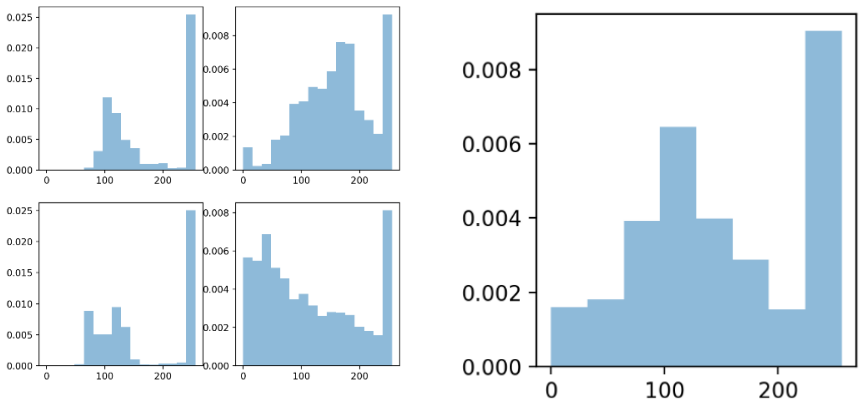
Рисунок 3.5 – Демонстрація роботи класифікатора

В кінці, в результаті аналізу, проводиться підрахунок зображень за наступними критеріями:

– загальна кількість документів;
– загальна кількість зображень;
– кількість зображень, які не пройшли фільтрацію;
– середня кількість зображень у роботі;
– максимальна/мінімальна кількість зображень у роботі;
– середня кількість зображень у роботі, які не пройшли фільтрацію;
– кількість фотозображень;
– кількість схематичних зображень;
– кількість зображень, підозрілих на плагіат.



(а)



(б) (в)

Рисунок 3.6 – Формування ознак на основі гістограм для цілого зображення: (а) задане зображення; (б) створена гістограма за 4 частинами зображення; (в) створена гістограма за цілим зображенням

Результат виконання методу аналізу графічного контенту електронної колекції текстових документів на наявність зображень, підозрілих на плагіат, показано на рисунку 3.7.

Аналіз зображень в колекції текстових документів на наявність плагіату зображень		
Загальна кількість документів:	90	(100%)
Загальна кількість зображень:	3533	(100%)
Середня кількість зображень у роботі:	39.26	(100%)
Максимальна/мінімальна кількість зображень у роботі:	310/5	
Кількість зображень, які не пройшли попередню фільтрацію (за розміром та кольором):	593	(16.78%)
Середня кількість зображень, які не пройшли попередню фільтрацію (за розміром та кольором):	6.59	(16.78%)
Кількість схем:	2299	(65.07%)
Кількість фотозображень:	641	(18.14%)
Кількість зображень, підозрілих на плагіат:	**355**	**(10.05%)**

Рисунок 3.7 – Результат аналізу графічного контенту електронної колекції текстових документів на наявність зображень, підозрілих на плагіат

3.3.4 Висновок щодо аналізу графічного контенту

За результатами аналізу графічного контенту електронної колекції текстових документів на наявність зображень, підозрілих на плагіат можна зробити декілька висновків.

Більшість зображень, яка використовується у роботах, є фотосхемами – трохи більше 65,07%. Це зумовлено тим, що кваліфікаційні роботи магістрів та бакалаврів Кафедри Інформатики носять науковий характер. Дані роботи містять багато формул, графіків, схем, діаграм тощо у вигляді рисунків.

Значення максимальної та мінімальної кількості зображень, які містяться у роботах, сильно відрізняються (у 62 рази – 310 проти 5 зображень). Значення середньої кількості зображень у роботах майже в 8 разів більше, ніж значення мінімальної кількості.

За статистикою, також наявні зображення, які не пройшли фільтрацію. Дані зображення могли бути представлені маленькими літерами, якими створюють формули, та які не можуть бути перевірені на плагіат через свій розмір. Їх кількість складає 593 (16,78%).

Головний пункт статистики «Кількість зображень, підозрілих на плагіат» має значення 355 зображень (10,05% відсотків від загальної кількості зображень, враховуючи ті, які не пройшли фільтрацію). Це означає, що кожне 10 зображення було взяте з іншої роботи.

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований метод аналізу зображень в електронних колекціях текстових документів для вирішення проблеми виявлення плагіату зображень, для чого була досліджена актуальність проблемі плагіату графічної проблеми та необхідність її вирішення, навчена модель нейронної мережі для класифікації зображень, налаштовані бібліотеки обробки зображень та розпізнавання образів для їх подальшого використання, вивчені та використані на практиці методи кластеризації, вивчено питання отримання ознак зображення на основі гістограм, проведено практичне дослідження графічного контенту колекції наукових звітів та кваліфікаційних робіт бакалаврів та магістрів та створено додаток, який автоматизує процес дослідження та підраховує статистику. В якості матеріалу для навчання нейронної мережі був обраний датасет Coco.

Дослідження показали, що процес класифікації недостатньо ефективний, оскільки існують зображення, в яких наявні як графіки, та і фото, тобто це зображення однаково може бути віднесеним до будь-якого з класів. Для вирішення цієї проблеми можна перенавчити модель на класифікацію зображень на 3 класи, де до 3 класу будуть відноситися такі зображення. Провівши дослідження таких методів кластеризації зображень, як *k*-means у поєднанні із методом Elbow та DBSCAN можна зробити висновок, що методи *k*-means із Elbow є більш ефективні у даній задачі, оскільки метод DBSCAN не здатний кластеризувати дані ефективно. Даний метод розподіляє дані або на 1 великий кластер та декілька маленьких, або створює понад 20 маленьких кластерів. При дослідженні нового зображення даний метод буде виконуватися спочатку та може видати зовсім інший результат, що призведе до неспрогнозованих наслідків. Хоча *k*-means має більшу ефективність від DBSCAN, він недостатньо добре проводить класифікацію. Даний метод створює такі кластери, у яких наявні точки, які знаходяться в іншому кластері,

тобто межи кластерів перетинаються. Це змушує проводити пошук зображень з сусідніх кластерів.

Базу даних, яка буде сформована в результаті аналізу графічного контенту електронної колекції текстових документів можна використовувати у майбутніх дослідженнях. Однією з задач, яка може бути при наступному дослідженні – розширення бази даних зображень, які використовуються в кваліфікаційних роботах.

Результати роботи апробовано у вигляді 2 тез доповідей під час 27-го Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ В XXI СТОЛІТТІ» [32] та 14-ої Міжнародної науково-практичної конференції «FREE AND OPEN SOURCE SOFTWARE» [33].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Duplichecker – plagiarism checker service. URL: https://www.duplichecker.com/ (дата звернення 10.04.2023).

2. Plagiarism Checker by Plagiarisma. URL: https://plagiarisma.net/ (дата звернення 10.04.2023).

3. Plagiarism checker by Grammarly. URL: https://www.grammarly.com/plagiarism-checker (дата звернення 10.04.2023).

4. Search Engine. Plagiarism Checker. URL: https://searchenginereports.net/plagiarism-checker (дата звернення 10.04.2023).

5. Paperrater. URL: https://www.paperrater.com/ (дата звернення 10.04.2023).

6. Edubirdie.Plagiarism checker that will help improve your originality. URL: https://edubirdie.com/plagiarism-checker (дата звернення 10.04.2023).

7. Plagium. URL: https://www.plagium.com/en/plagiarismchecker (дата звернення 10.04.2023).

8. Google lens. URL: https://lens.google/ (дата звернення 12.04.2023).

9. Microsoft Bing. URL: https://www.bing.com/ (дата звернення 10.04.2023).

10. TinEye. URL: https://tineye.com/ (дата звернення 12.04.2023).

11. OpenCV. URL: https://opencv.org/ (дата звернення 15.04.2023).

12. TensorFlow. URL: https://www.tensorflow.org/ (дата звернення 15.04.2023).

13. Keras. URL: https://keras.io/ (дата звернення 15.04.2023).

14. MobileNetV2. URL: https://keras.io/api/applications/mobilenet/#mobilenet-function (дата звернення 15.04.2023).

15. JetBrains IntelliJ Idea. URL: https://www.jetbrains.com/idea/ (дата звернення 20.04.2023).

16. Spring Framework 2.1.6.RELEASE. URL: https://spring.io/blog/2019/06/19/spring-boot-2-1-6-released (дата звернення 20.04.2023).

17. Jupyter Notebook. URL: https://jupyter.org/ (дата звернення 23.04.2023).

18. Postgres. URL: https://www.postgresql.org/ (дата звернення 25.04.2023).

19. Docker. URL: https://www.docker.com/ (дата звернення 25.04.2023).

20. JetBrains WebStorm. URL: https://www.jetbrains.com/webstorm/ (дата звернення 26.04.2023).

21. Cherednichenko, O., Kanishcheva, O., Yakovleva, O., & Arkatov, D. (2020). Collection and Processing of a Medical Corpus in Ukrainian. corpus, 2(4), 7-14.

22. Cherednichenko, O., Vovk, M., Yanholenko, O., & Yakovleva, O. (2020). Towards the Technology of Employers' Requirements Collection Development. In Integrated Computer Technologies in Mechanical Engineering (pp. 228-239). Springer, Cham.

23. V. Gorokhovatskyi, I. Tvoroshenko, Image Classification Based on the Kohonen Network and the Data Space Modification, in: CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020), 2020, pp. 1013–1026. doi:10.32782/cmis/2608-76.

24. А.Р. Ковтуненко, О.В. Яковлева, В.А. Любченко, & О.В. Янголенко (2020) Дослідження сумісного використання математичної морфології та згорткових нейронних мереж для вирішення задачі розпізнавання цінників. Вісник Національного технічного університету ХПІ (3). 24-31.

25. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Tools for fast metric data search in structural methods for image classification, *IEEE Access*, 10, pp. 124738-124746.

26. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the

structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

27. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.

28. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

29. Yakovleva, O., Kovtunenko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System (COLINS-2023). In CEUR Workshop Proceedings (Vol. 3403). pp. 69-86.

30. Yakovleva, O., & Nikolaieva, K. (2020). Research Of Descriptor Based Image Normalization And Comparative Analysis Of SURF, SIFT, BRISK, ORB, KAZE, AKAZE Descriptors. Advanced Information Systems, 4(4), 89-101. doi:10.20998/2522-9052.2020.4.13.

31. SytossResearch. (n.d.). SytossResearch/descriptorbasednormalization. GitHub. URL: https://github.com/SytossResearch/DescriptorBasedNormalization (дата звернення 03.05.2023).

32. Іщенко, О. (2023). Розробка методу аналізу зображень в електронній колекції текстових документів для вирішення задачі пошуку плагіатних зображень.

33. Іщенко, О. (2023). Огляд технології Spring Framework для розробки на мові програмування Java.