Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

РОЗРОБКА ПРОТОТИПУ СИСТЕМИ ДЛЯ АНАЛІЗУ ЗОБРАЖЕНЬ НА ПЛАГІАТ З ВИКОРИСТАННЯМ АЛГОРИТМУ SSIM

Текстова частина до курсової роботи
за спеціальністю „Інженерія програмного забезпечення” 121

Керівник курсової роботи
с.в.___Вовк Н.Є._____
(прізвище та ініціали)
_____
(підпис)
“____” _____ 2020 р.

Виконав студент
____Попова А.В.___
(прізвище та ініціали)
“____” _____ 2020 р.

Київ 2020

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем,

доцент, к.ф. – м.н. О. П. Жежерун _____

(підпис)

„____”_____2019 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту__Поповій А.В.__ факультету _інформатики_ _____3-го____курсу

Розробити Комп’ютерну систему перевірки зображень у PDF-документі на плагіат, використовуючи алгоритм SSIM та результати Google Images

Зміст ТЧ до курсової роботи:

Індивідуальне завдання
Календарний план виконання роботи
Зміст
Анотація
Вступ
1 Огляд існуючих сервісів перевірки зображень на плагіат
2 Пошук подібних зображень за допомогою Google Cloud Vision API
3 Алгоритми вимірювання схожості зображень та переваги алгоритму SSIM
4 Розробка системи та алгоритм її роботи
Висновки
Список літератури
Додатки

Дата видачі „___” _____ 2020 р. Вовк. Н. Є. _____

(підпис)

Завдання отримав _____

(підпис)

Тема: РОЗРОБКА ПРОТОТИПУ СИСТЕМИ ДЛЯ АНАЛІЗУ ЗОБРАЖЕНЬ НА ПЛАГІАТ З ВИКОРИСТАННЯМ АЛГОРИТМУ SSIM

Календарний план виконання роботи

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	13.10.2019	
2.	Аналіз літератури та технічних джерел за темою роботи	12.11.2019	
3.	Дослідження інструментів для розробки	04.12.2019	
4.	Розробка алгоритму роботи системи	20.01.2020	
5.	Програмування цілісної системи з інтерфейсом	28.01.2020	
6.	Аналіз результатів з керівником та корегування роботи системи	02.02.2020	
7.	Написання текстової частини до курсової роботи	30.03.2020	
8.	Аналіз виконаної роботи з керівником, виправлення недоліків	10.04.2020	
9.	Створення слайдів для доповіді та написання доповіді	21.04.2020	
10.	Остаточне оформлення доповіді, слайдів та текстової частини до курсової роботи	02.05.2020	
11.	Захист курсової роботи	19.05.2020	

Попова А. В. _____

Вовк Н. Є. _____

“_____” _____

Зміст

Анотація

Розробка системи порівняння зображень на плагіат була виконана задля задовільнення потреби сучасного ринку комп’ютерних застосунків у програмі, що знаходить збіги обраного зображення із наявними картинками у відкритому доступі в мережі Інтернет. Система шукає збіги у базі зображень Google Images та вираховує коефіцієнти подібності вхідного та вихідних картинок за допомогою алгоритму SSIM. Результатом роботи є повноцінна програма, що приймає на вхід PDF-файл і видає відсотки подібності та посилання на подібні зображення для кожної картинки у файлі.

Ключові слова: плагіат зображень, перевірка картинок на плагіат, порівняння зображень, пошук ідентичних картинок.

Вступ

Проблема плагіату зображень полягає у декількох аспектах. По-перше, пересічному користувачу доступні терабайти зображень, що знаходяться у відкритому доступі; їх можна завантажити у локальне сховище та використати в особистих цілях. З іншого боку, навіть за умови перебування картинок у мережі Інтернет, право власності залишається за автором зображення. Отже, кожен користувач Інтернету повинен мати можливість відстежити використання поширених ним зображень. По-друге, існує протилежна потреба: перевірити, чи є зображення авторським. Найбільш доцільним прикладом в академічному контексті буде перевірка зображень на плагіат у дослідницькій роботі. Саме на потребі перевірки наукових та академічних документів на плагіат картинок ґрунтується подальша робота над розробкою відповідної системи.

Результат пошуку вже існуючих програм для виявлення плагіату зображень, що містяться у деякому документі, продемонстрував відсутність на ринку програмного забезпечення комплексного застосунку для такої потреби. Отже, мета роботи — визначити необхідні інструменти для роботи з зображеннями у документі, пошуку подібних зображень та вирахування відсотку їх подібності; використати ці інструменти для розробки комп’ютерної системи порівняння зображень у PDF-документі на плагіат.

Робота складається з чотирьох розділів.

У першому розділі проаналізовано існуючі сервіси перевірки зображень на плагіат. Наведено переваги та недоліки кожного сервісу.

Другий розділ присвячено дослідженню технологій Google Cloud Vision API. Проведено огляд умов користування платформою, розглянуто алгоритм пошуку подібних зображень, обрано функції, необхідні для розробки власної системи.

У третьому розділі коротко розглянуто існуючі алгоритми визначення подібності зображень, проаналізовано алгоритм SSIM та визначено його переваги над рештою алгоритмів.

Четвертий розділ присвячено покроковому опису логіки розробки системи. Продемонстровано використання обраних інструментів. Виконано аналіз результатів роботи системи.

Створено програмний продукт, що аналізує зображення у PDF-документі на плагіат.

Постановка задачі

1. Знайти та порівняти існуючі сервіси перевірки зображень на плагіат.

2. Ретельно дослідити технології та умови користування Google Cloud Vision API, обрати необхідні для подальшої розробки функції.

3. Зробити критичний огляд найбільш відомих алгоритмів визначення подібності зображень, обгрунтувати вибір алгоритму для подальшої розробки.

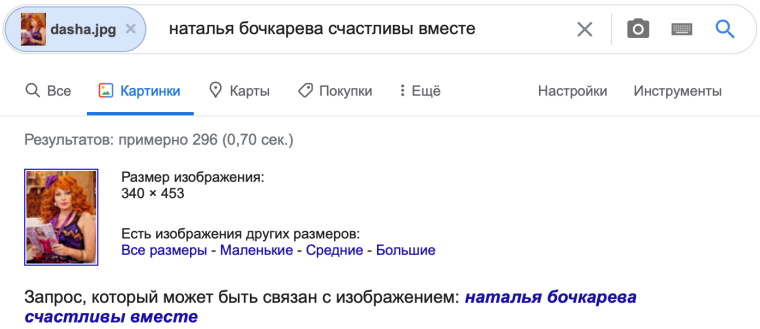
4. За допомогою обраних інструментів розробити систему аналізу зображень на плагіат, проаналізувати результати роботи системи

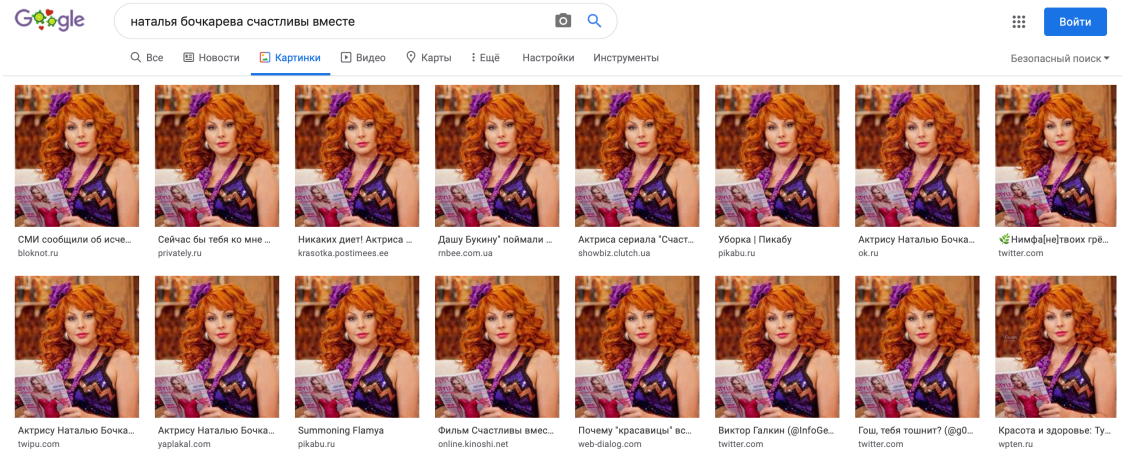
Розділ 1: Огляд існуючих сервісів перевірки зображень на плагіат

1. Пошукові системи Інтернету

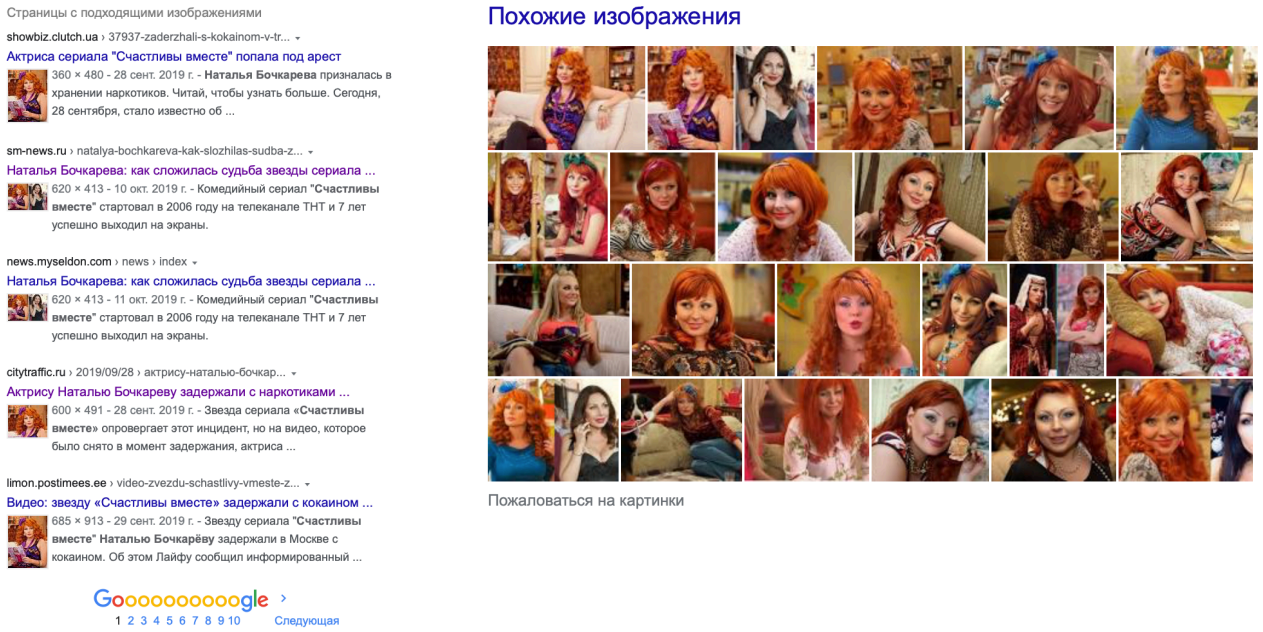
Пошукові системи Інтернету Google та Bing пропонують користувачу пошук за зображенням чи посиланням на зображення. Використовуючи запропонований інструмент, можна знайти подібні або ідентичні зображення у базах зображень відповідних пошукових систем.

Розглянемо користувацький досвід пошуку подібних зображень у Google Images:





Рисунки 1.1.1, 1.1.2 — пошук у Google Images ідентичних фото. Сервіс пропонує текстовий запит; якщо натиснути на завантажене фото — буде запропоновано перелік ідентичних картинок у мережі, з різною роздільною здатністю.



Рисунки 1.1.3, 1.1.4 — Google Images пропонує веб-сторінки, що містять ідентичні зображення; також користувачу представлені візуально схожі зображення.

Розглянемо користувацький досвід пошуку подібних зображень у Bing:

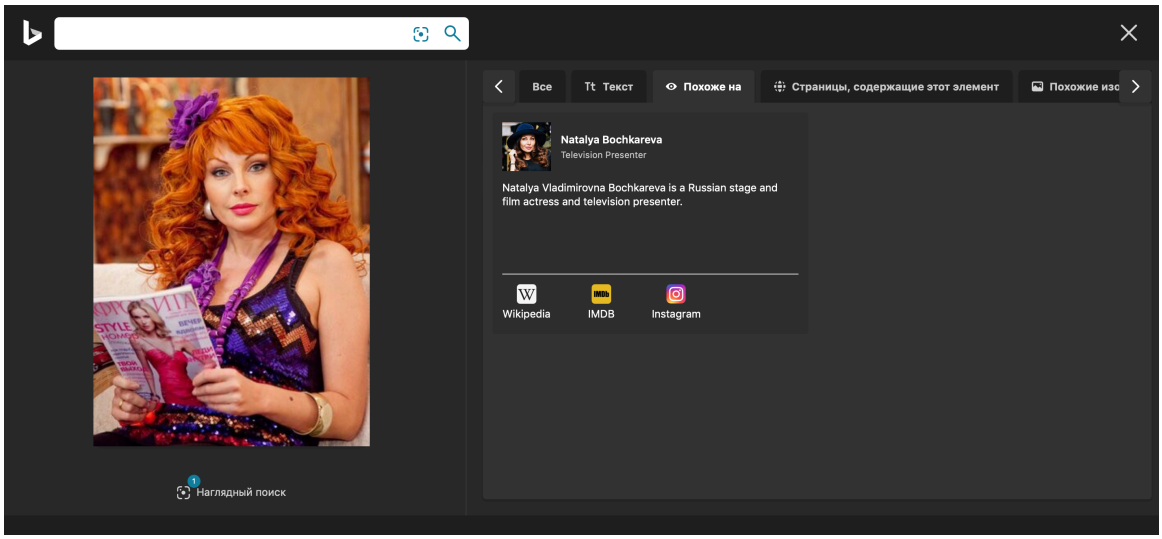
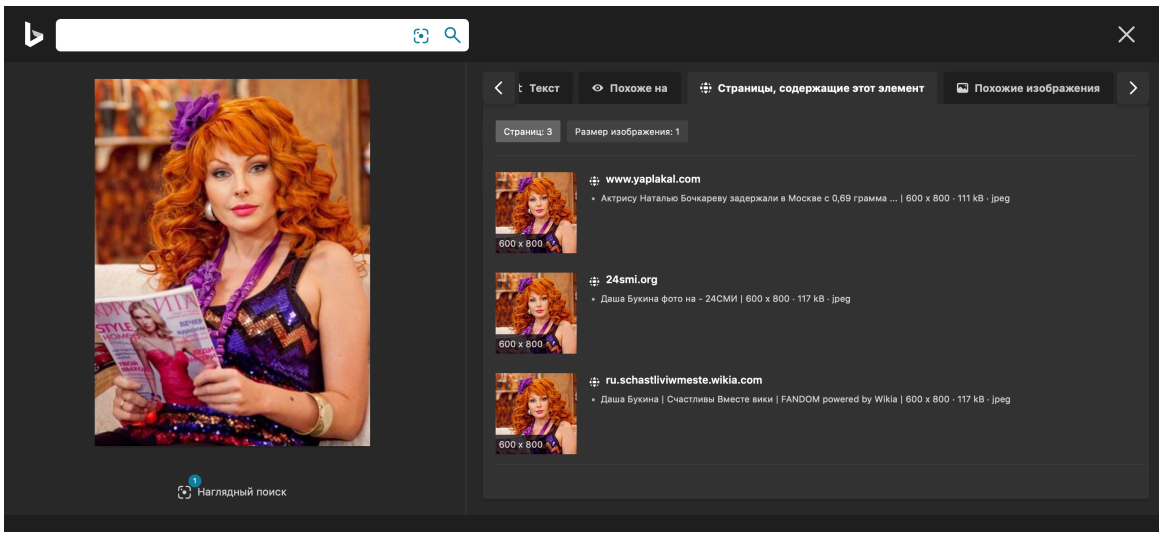
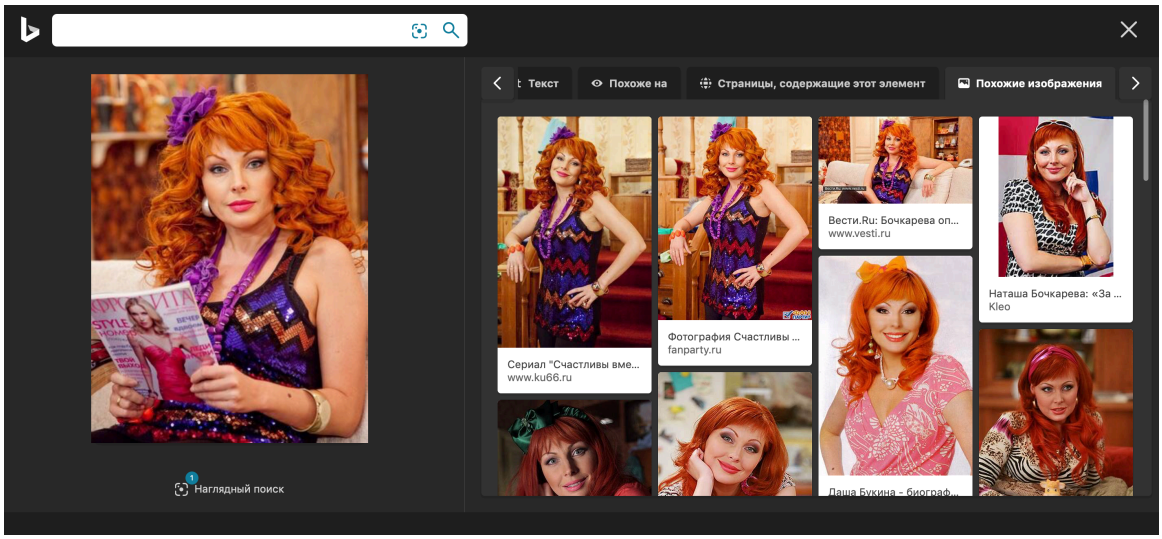


Рисунок 1.1.5 — пошук у Bing ідентичних фото. Сервіс пропонує визначення змісту фотографії.





Рисунки 1.1.6, 1.1.7 — запропоновано веб-сторінки, що містять ідентичні зображення; також користувачу представлені візуально схожі зображення.

Безперечно, база зображень Google Images значно більша за базу будь-якої іншої пошукової системи, адже за даними NetMarketShare [1], наразі Google займає 71.75% ринку усіх пошукових систем. Одразу після Google іде Bing із відсотком 12.34%. Ці дані підтверджуються наочним прикладом того, що пошук по фотографії акторки у Google видав більше 100 веб-сторінок, що містять ідентичні зображення, а пошук того самого фото у Bing показав лише 3 веб-сторінки.

Переваги пошукових систем Інтернету:

- Найбільша база проіндексованих зображень
- Можливість пошуку видозмінених картинок

• За допомогою технологій машинного навчання є можливість пошуку візуально схожих картинок

• Користувачу демонструються технічні характеристики віднайдених зображень

Недоліки перевірки зображень на плагіат за допомогою пошукових систем Інтернету:

• Відсутність можливості перевірки документу із зображеннями, які необхідно перевірити на плагіат

• Відсутність кількісних показників схожості зображень

2. TinEye

TinEye — система, що знаходить зображення за допомогою технології reverse image search [2]. TinEye не використовує назву або будь-які інші дані шуканого зображення. Коли виконується пошук ідентичних зображень, TinEye створює унікальний цифровий підпис (або «відбиток пальця») для шуканої картинки за допомогою технологій розпізнавання зображень, а потім порівнює цей відбиток з усіма зображеннями в базі проіндексованих картинок, і таким чином знаходить відповідності [2].

Розглянемо користувацький досвід пошуку подібних зображень у TinEye:

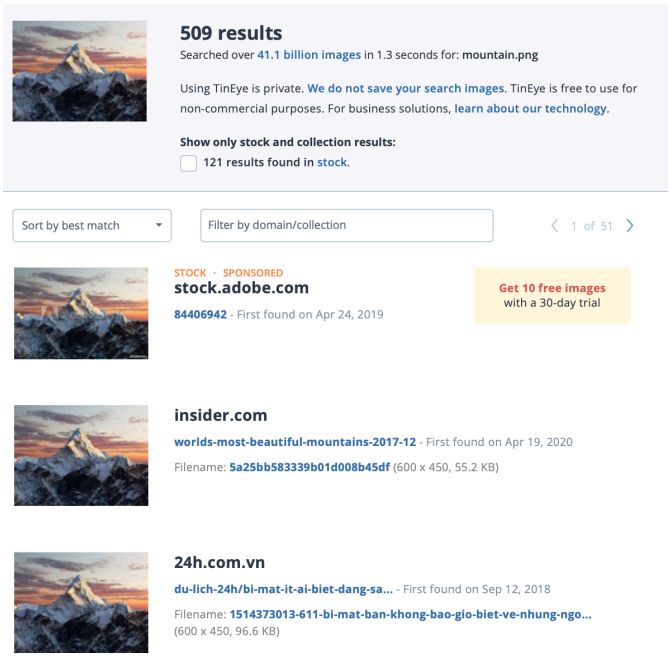


Рисунок 1.2.1 — пошук у TinEye ідентичних фото. Сервіс пропонує адреси знайдених зображень; веб-сторінки, на яких розміщені знайдені зображення; ідентичні зображення з різною роздільною здатністю; ідентичні зображення, що були обрізані або відредаговані.

Як можна побачити на рисунку 1.2.1, база проіндексованих зображень сервісу TinEye складає 41.1 млрд картинок, що є значним показником для стороннього, не пов'язаного з пошуковими системами Інтернету сервісу. Однак, варто зазначити, що сервіс TinEye не призначений для пошуку візуально подібних картинок, тобто, не виконується розпізнавання та пошук по змісту шуканого зображення [2].

Переваги сервісу TinEye:

• Відносно велика база проіндексованих зображень

• За допомогою технології «відбитка пальця» є можливість пошуку сильно видозмінених картинок

• Користувачу демонструються технічні характеристики віднайдених зображень

Недоліки сервісу TinEye:

• Відсутність пошуку візуально схожих картинок

• Відсутність можливості перевірки документу із зображеннями, які необхідно перевірити на плагіат

• Відсутність кількісних показників схожості зображень

3. Pixsy

Pixsy дійсно можна назвати сервісом, що перевіряє зображення саме на плагіат, а не просто на подібність. Система поєднує в собі технологію reverse image search та дозволяє користувачу, що знайшов збіг картинки в мережі, подати заявку на видалення незаконних копій зображення та отримати грошову компенсацію [3].

Розглянемо користувацький досвід пошуку подібних зображень у Pixsy:

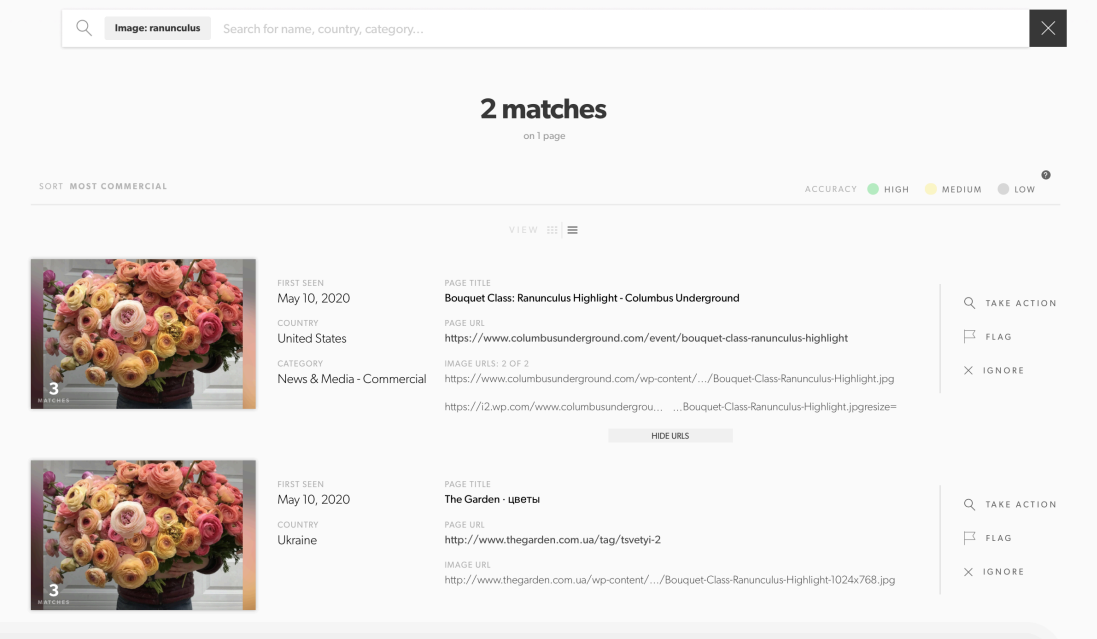


Рисунок 1.3.1 — пошук у Pixsy ідентичних фото. Сервіс пропонує адреси знайдених зображень; веб-сторінки, на яких розміщені знайдені зображення; країну та категорію веб-сторінки; можливість подати заявку про незаконне використання фотографії.

Якщо порівняти кількість результуючих зображень при пошуку за допомогою Google Images, буде очевидно, що база проіндексованих зображень Pixsy менша, ніж бази зображень пошукових систем Інтернету. Крім цього, інтерфейс сервісу більше підлаштований під вирішення проблеми плагіату зображення, аніж під дослідження відмінностей знімків: не вказано роздільну здатність, розмір та графічний формат ідентичних зображень.

Переваги сервісу Pixsy:

- Можливість дозволити системі зберігати картинку та відстежувати нові появи плагіату
- Можливість завантажити на перевірку зображення не тільки з локального сховища, а й з інших ресурсів, таких як Google Drive, Flickr, Instagram тощо
- Можливість вирішити проблему плагіату власних зображень засобами закону

Недоліки сервісу Pixsy:

- Відносно невелика база проіндексованих зображень

• Користувачу не демонструються технічні характеристики віднайдених зображень

• Відсутність можливості перевірки документу із зображеннями, які необхідно перевірити на плагіат

• Відсутність кількісних показників схожості зображень

4. Платні сервіси

Такі сервіси перевірки зображень на плагіат як RevIMG, Berify тощо передбачають плату за користування, отже, їх дослідити не вдалося.

Розділ 2: Пошук подібних зображень за допомогою Google Cloud Vision API

Для пошуку збігів вхідного зображення з іншими картинками у мережі, необхідно мати доступ до відповідної бази зображень та послуговуватися інструментами для роботи з цією базою. Отже, метою цього етапу роботи був пошук дружніх у користуванні програмних засобів, що призначені для операцій над об’ємною базою веб-зображень. У розділі 1 було визначено базу зображень Google Images як найбільшу в мережі, тому логічним рішенням був вибір програмних засобів від Google. Google Cloud Vision API, що входить до переліку хмарних технологій Google Cloud Platform, є безперечним лідером у сфері машинного навчання, технологій роботи з зображеннями, роботи з великими об’ємами інформації тощо.

1. Умови користування платформою

Вартість користування Vision API розраховується відповідно до кількості опрацьованих зображень. Для кожної функції на платформі, робота з 1000 зображень на місяць — безкоштовна, після безкоштовного ліміту оплата нараховується за кожну наступну тисячу опрацьованих зображень [4]. Однак, варто зазначити, що вартість розраховується для кожної функції окремо. Наприклад, якщо користувач платформи застосує функцію розпізнавання обличчя до 1000 зображень, а також застосує функцію розпізнавання тексту на картинках до 1000 зображень, вартість буде безкоштовною, адже для кожної функції є окремий ліміт у 1000 картинок.

В цілому, ціна користування різними функціями ідентична, хоча кілька функцій є дорожчими. Реєстрація на платформі вимагала введення даних банківської картки, хоча для вдалої розробки та аналізу результатів роботи системи вистачило безкоштовного ліміту. Для подальшого користування створеною програмою варто дослідити розцінки платформи:

Feature	Price per 1000 units		
	First 1000 units/month	Units 1001 - 5,000,000 / month	Units 5,000,001 - 20,000,000 / month
Label Detection	Free	$1.50	$1.00
Text Detection	Free	$1.50	$0.60
Document Text Detection	Free	$1.50	$0.60
Safe Search (explicit content) Detection	Free	Free with Label Detection, or $1.50	Free with Label Detection, or $0.60
Facial Detection	Free	$1.50	$0.60
Facial Detection - Celebrity Recognition	Free	$1.50	$0.60
Landmark Detection	Free	$1.50	$0.60
Logo Detection	Free	$1.50	$0.60
Image Properties	Free	$1.50	$0.60
Crop Hints	Free	Free with Image Properties, or $1.50	Free with Image Properties, or $0.60
Web Detection	Free	$3.50	Contact Google for more information
Object Localization	Free	$2.25	$1.50

Рисунок 2.1.1 — розцінка використання технологій платформи Google Cloud Vision API [4].

2. Обрані функції

Наразі Vision API пропонує такі категорії функцій [5]:

- Розпізнавання обличчя, емоцій та інших характеристик обличчя на фотографії
- Розпізнавання локації на зображенні та її координат
- Розпізнавання логотипу на зображенні
- Визначення змісту та імовірної назви зображення
- Розпізнавання тексту на зображенні
- Розпізнавання друкованого та рукописного тексту в документі
- Визначення домінантних кольорів та характеристик зображення
- Розпізнавання та опис об’єктів на зображенні
- Робота з оригіналами зображень та їхніми обрізаними копіями
- Розпізнавання відвертого змісту зображень
- Робота з веб-контентом, що має відношення до зображення (Web Entities)

Остання категорія знадобиться у подальшій роботі над застосунком, адже вона безпосередньо стосується бази зображень у мережі. Розглянемо алгоритм пошуку подібних зображень та оберемо необхідні функції.

Незважаючи на те, що Vision API найбільш відомий своїми інструментами розпізнавання за допомогою машинного навчання, він також має унікальну можливість виконувати алгоритм reverse image search (пошук за зображенням) у базі зображень Google Images [6]. За це відповідає категорія функцій Web Entities. При використанні цієї категорії, користувач отримує такі дані [5]:

- Веб-сутності (назви та описи подібних зображень в Інтернеті)
- Повні збіги зображень (список посилань на зображення будь-якої роздільної здатності, що повністю збігаються із вхідним зображенням)
- Часткові збіги зображень (список посилань на зображення будь-якої роздільної здатності, що частково збігаються із вхідним зображенням)
- Веб-сторінки, що мають збіги зображень (список посилань на веб-сторінки, що містять повні або часткові збіги зображень, та посилання власне на зображення)
- Візуально схожі зображення (список посилань на зображення будь-якої роздільної здатності, що є візуально схожими на вхідне зображення)
- Імовірна назва вхідного зображення

При недостатній ознайомленості з роботою функцій може здатися, що найбільш доцільним буде використання функції пошуку веб-сторінок, що мають збіги зображень, адже користувачу корисніше мати посилання саме на ресурси плагіату зображень, а не власне на зображення. Практика показала, що кожна функція категорії Web Entities, що повертає список посилань, показує лише 10 найперших збігів у мережі. Тому, аби не втратити релевантні посилання на віднайдені картинки, було вирішено знехтувати посиланнями на веб-ресурси і надалі використовувати при розробці функції пошуку повних збігів (full matching images), часткових збігів (partial matching images) та візуальної схожості (visually similar images), що у сумі дають максимум 30 посилань на подібні зображення в Інтернеті, на противагу лише 10 посиланням при використанні функції пошуку веб-сторінок.

Розділ 3: Алгоритми вимірювання схожості зображень та переваги алгоритму SSIM

1. Середня квадратична похибка (MSE)

MSE (Mean Squared Error) — традиційний метод вимірювання схожості зображень за допомогою вимірювання точності сигналу зображень [9].

Два сигнали попіксельно порівнюються зліва направо і зверху вниз рядками та стовпцями. Різниця між зображеннями відповідно обчислюється шляхом усереднення квадрату різниці між похибками зображень [7]. Формула різниці MSE між двома цифровими зображеннями:

$$MSE(X,Y) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - y_i\right)^2 \quad (3.1)$$

Де X, Y — сигнали відповідних зображень, N — кількість пікселів, а x_i та y_i — значення сигналу в кожному пікселі [9].

Цей алгоритм набрав велику популярність через свою простоту обчислення та використання малих об’ємів пам’яті [9]. Однак недоліком алгоритму є досить мале співпадіння із людським сприйняттям зображення. Візуально зовсім різні зображення можуть мати однакове значення MSE.

2. Пікове співвідношення сигналу до шуму (PSNR)

PSNR (Peak Signal-to-Noise Ratio) — популярний метод вимірювання схожості зображень за допомогою обчислення співвідношення максимально можливого значення сигналу та потужності спотворюючого шуму на зображенні [8]. Схожість зображень вимірюється у децибелах.

Значення PSNR обчислюється на основі MSE і є, по суті, зворотнім децибельним значенням MSE [9]. Формула обчислення PSNR між двома цифровими зображеннями:

$$PSNR(X,Y) = 10log_{10}\left(\frac{MPP^2}{MSE(X,Y)}\right) \quad (3.2)$$

Де X, Y — сигнали відповідних зображень, N — кількість пікселів, x_i та y_i — значення сигналу в кожному пікселі, а MPP — максимальне значення пікселю (зазвичай 255) [9].

Незважаючи на популярність, цей алгоритм не є досконалим через відсутність аналізу структурних компонентів зображення [10].

3. Індекс структурної схожості (SSIM) та його переваги

Алгоритм SSIM (Structure Similarity Index Method) вважається покращеною версією та продовженням роботи над алгоритмами MSE та PSNR. Він заснований на сприйнятті людиною зображень [8]. Обчислюється за допомогою порівняння між трьома складовими: освітленість, контрастність та структура [9].

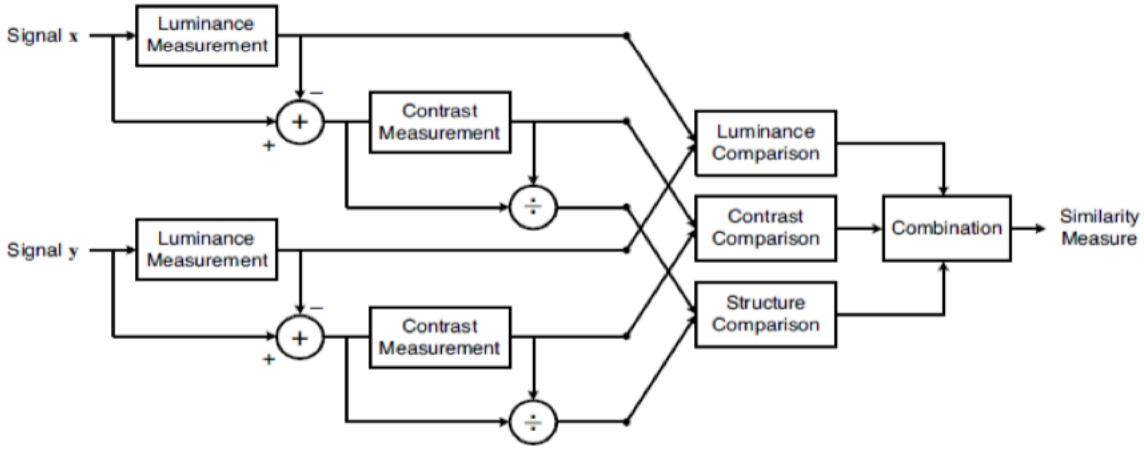


Рисунок 3.3.1 — діаграма системи обчислення за алгоритмом SSIM [9]

Обчислення індексу структурної схожості набагато складніше за обчислення MSE чи PSNR та є більш ресурсозатратним. Однак цей алгоритм дає неперевершені результати у співпадіннях із людським сприйняттям зображення, адже містить у собі не тільки попіксельні обчислення різниці, а й аналіз структурних компонентів картинки.

З огляду на те, що при розробці застосунку необхідно враховувати порівняння вхідного зображення із видозміненими та відредагованими картинками, а отже, виконувати структурний аналіз, для подальшої роботи було обрано алгоритм SSIM.

Розділ 4: Розробка системи та алгоритм її роботи

1. Застосовані при розробці інструменти

Для розробки системи перевірки зображень у PDF-документах на плагіат було обрано:

- Мову програмування Python
- Бібліотеку OpenCV для роботи з зображеннями
- Бібліотеку scikit-image для застосування функції structural_similarity, що має в основі обраний алгоритм SSIM
- Бібліотеку PySimpleGUI для побудови інтерфейсу програми
- Бібліотеку google.cloud.vision для під'єднання до обраного сервісу Vision API
- Бібліотеку csv для запису результатів роботи програми у окремий текстовий файл
- Службовий пакет poppler-utils для роботи з зображеннями у PDF-документах
- Інші службові бібліотеки

2. Алгоритм роботи системи

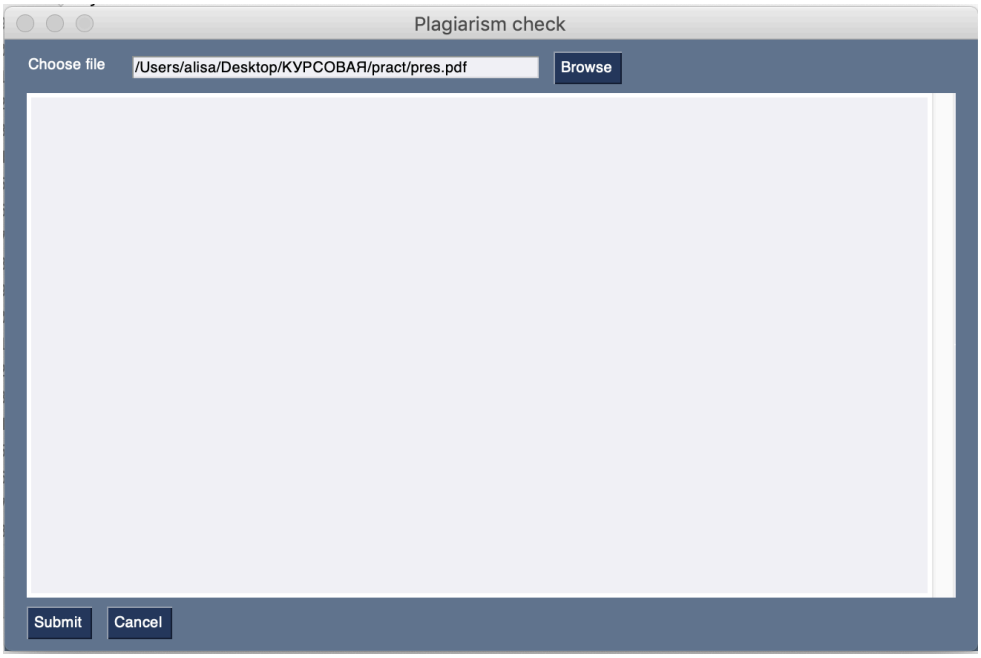


Рисунок 4.2.1 — користувач натискає кнопку «Browse» і обирає з локального сховища потрібний PDF-документ

Рисунок 4.2.2 — PDF-документ, що обрано для перевірки коректності роботи системи

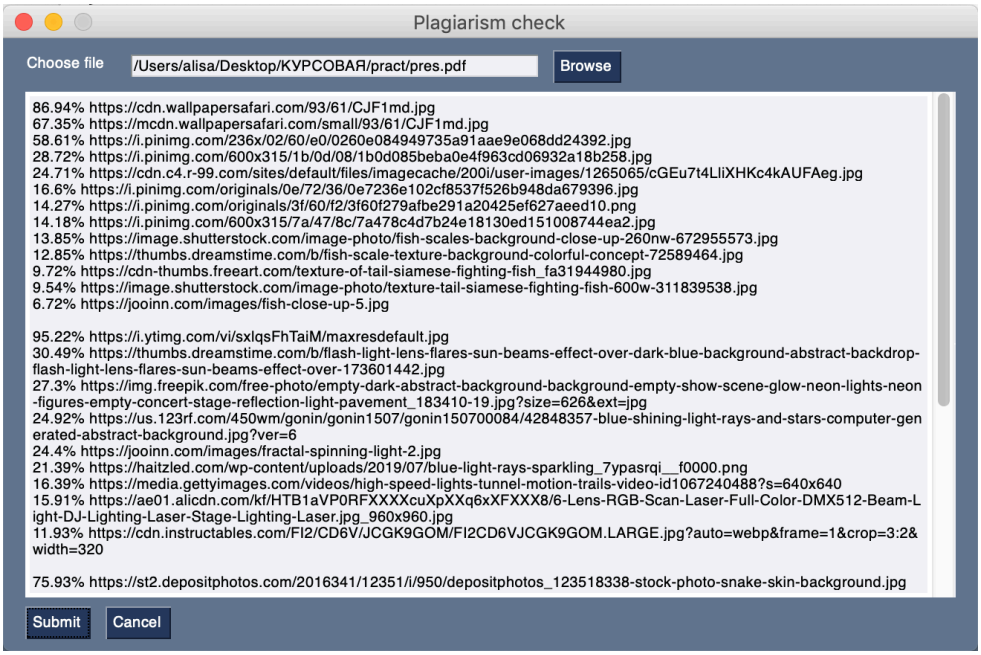


Рисунок 4.2.3 — користувач натискає кнопку «Cancel» аби вийти із програми або кнопку «Submit» аби розпочати процес пошуку та перевірки зображень у мережі на плагіат. У текстовому полі вікна відображається список посилань на результуючі зображення, відсортований за відсотком подібності

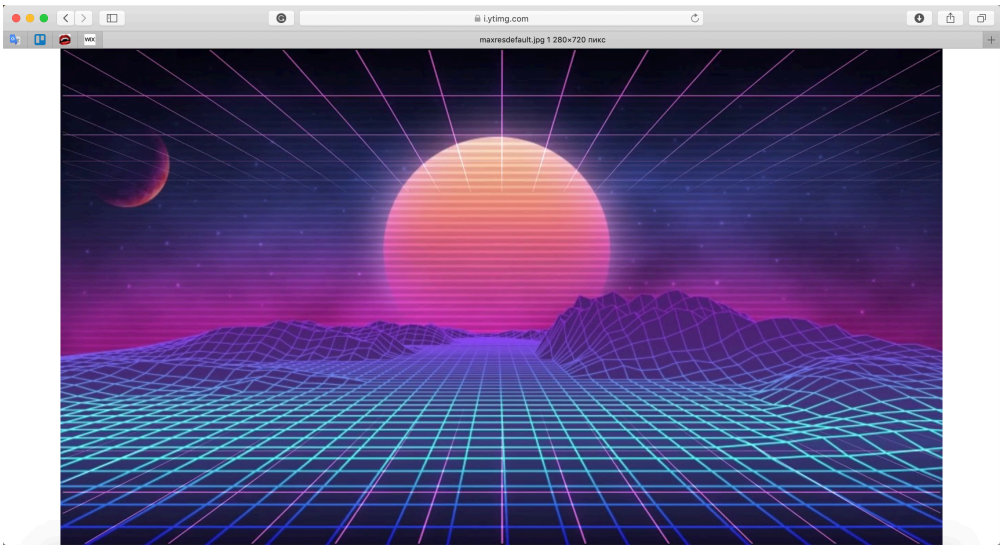


Рисунок 4.2.4 — результуюче зображення, коефіцієнт схожості якого становить 95.22%

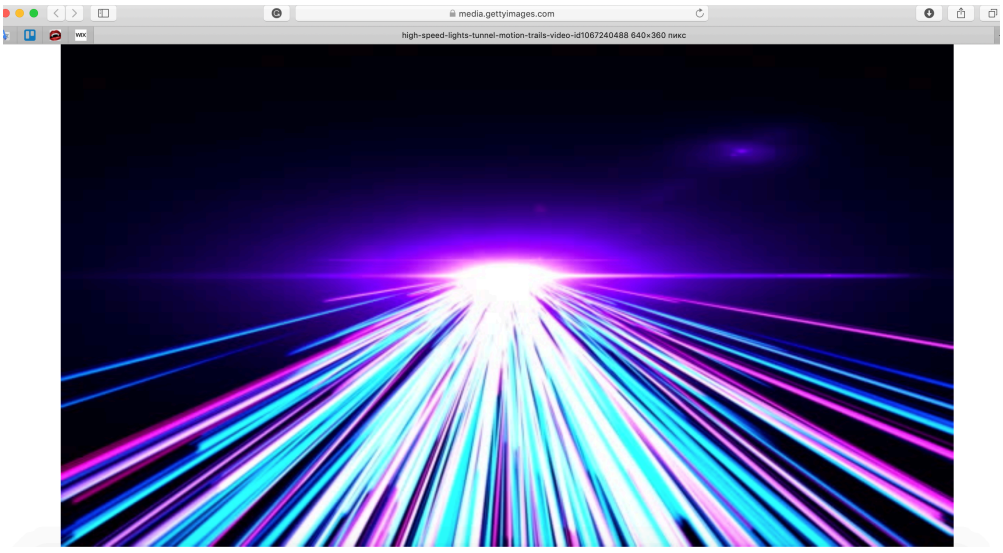


Рисунок 4.2.5 — результуюче зображення, коефіцієнт схожості якого становить 16.39%

result

https://cdn.wallpapersafari.com/93/61/CJF1md.jpg	86.94
https://mcdn.wallpapersafari.com/small/93/61/CJF1md.jpg	67.35
https://i.pinimg.com/236x/02/60/e0/0260e084949735a91aae9e068dd24392.jpg	58.61
https://i.pinimg.com/600x315/1b/0d/08/1b0d085beba0e4f963cd06932a18b258.jpg	28.72
https://cdn.c4.r-99.com/sites/default/files/imagecache/200i/user-images/1265065/cGEu7t4LiXHKc4kAUFAeg.jpg	24.71
https://i.pinimg.com/originals/0e/72/36/0e7236e102cf8537f526b948da679396.jpg	16.6
https://i.pinimg.com/originals/3f/60/f2/3f60f279afbe291a20425ef627aeed10.png	14.27
https://i.pinimg.com/600x315/7a/47/8c/7a478c4d7b24e18130ed151008744ea2.jpg	14.18
https://image.shutterstock.com/image-photo/fish-scales-background-close-up-260nw-672955573.jpg	13.85
https://thumbs.dreamstime.com/b/fish-scale-texture-background-colorful-concept-72589464.jpg	12.85
https://cdn-thumbs.freeart.com/texture-of-tail-siamese-fighting-fish_fa31944980.jpg	9.72
https://besthqwallpapers.com/Uploads/4-5-2019/90265/thumb2-fish-scales-texture-fish-skin-scales-background-fish-gray-scales-background.jpg	8.34
https://jooinn.com/images/fish-close-up-5.jpg	6.72
https://i.ytimg.com/vi/sxlqsFhTaiM/maxresdefault.jpg	95.22
https://png.pngtree.com/thumb_back/fw800/background/20190223/ourmid/pngtree-tech-purple-light-new-year-party-background-material-meetingannual-summary-meetingannual-image_71405.jpg	27.99
https://img.freepik.com/free-photo/empty-dark-abstract-background-background-empty-show-scene-glow-neon-lights-neon-figures-empty-concert-stage-reflection-light-pavement_183410-19.jpg?size=626&e	27.3
https://us.123rf.com/450wm/gonin/gonin1507/gonin150700084/42848357-blue-shining-light-rays-and-stars-computer-generated-abstract-background.jpg?ver=6	24.92
https://www.chakras.info/wp-content/uploads/Energy-Healer-1.jpg	22.07
https://qph.fs.quoracdn.net/main-qimg-76d559bc1837f1b848b99ed09d0b205e	20.79
https://media.gettyimages.com/videos/high-speed-lights-tunnel-motion-trails-video-id1067240488?s=640x640	16.39
https://ae01.alicdn.com/kf/HTB1aVP0RFXXXXcuXpXXq6xXFXXX8/6-Lens-RGB-Scan-Laser-Full-Color-DMX512-Beam-Light-DJ-Lighting-Laser-Stage-Lighting-Laser.jpg_960x960.jpg	15.91
https://cdn.instructables.com/FI2/CD6V/JCGK9GOM/FI2CD6VJCGK9GOM.LARGE.jpg?auto=webp&frame=1&crop=3:2&width=320	11.93
https://st2.depositphotos.com/2016341/12351/i/950/depositphotos_123518338-stock-photo-snake-skin-background.jpg	75.93
https://t4.ftcdn.net/jpg/02/54/80/37/240_F_254803758_MPeb1GY74JSj58xfICMJWdD2AEYXGGxT.jpg	45.72
https://images.designtrends.com/wp-content/uploads/2016/10/24182723/Pink-Flowers-Pattern.jpg	3.12
https://media.fabfab.net/images/products/popup/cotton-jersey-scale-pattern-pink--104_poso_b20_236.jpg	3.1
https://thumbs.dreamstime.com/b/background-pattern-fleece-cotton-weaving-fabric-braid-crafts-thread-flowers-ornament-lace-geometric-decor-vintage-design-176040911.jpg	2.96
https://us.123rf.com/450wm/euroshot/euroshot1312/euroshot131200019/24454445-rings-optical-illusion-vector-seamless-pattern-background-some-stereoscopic-effect-appears-.jpg?ver=6	2.68
https://thumbs.dreamstime.com/b/seamless-pattern-cute-cartoon-colorful-fish-hand-drawn-scribbles-sea-animals-kids-background-fabric-baby-clothes-textile-156547015.jpg	2.57
https://l450v.alamy.com/450v/2aw2amb/this-is-a-heart-shape-fair-isle-pattern-suitable-for-valentines-day-print-designs-fashion-textiles-knitwear-etc-2aw2amb.jpg	2.54
https://l450v.alamy.com/450v/ra7k8w/seamless-red-pattern-with-hearts-vector-illustration-ra7k8w.jpg	2.24
https://l450v.alamy.com/450v/ff941r/complex-intricate-geometric-ethnic-style-seamless-pattern-in-magenta-ff941r.jpg	1.97
https://image.shutterstock.com/image-vector/black-seamless-lace-floral-pattern-600w-497322103.jpg	1.72
https://img.freepik.com/free-vector/batik-seamless-pattern_8306-370.jpg?size=338&ext=jpg	1.52

Рисунок 4.2.6 — CSV-файл із списком посилань на результуючі зображення, відсортованим за відсотком подібності

Висновки

Виконана курсова робота є багатокомпонентною складною системою, що надає дуже зручний функціонал для тих користувачів, що бажають перевірити PDF-документ на наявність неунікальних зображень. Проаналізувавши пропозиції вільного доступу в Інтернеті, можна зробити висновок, що сервіс є достатньо унікальним на предмет функціоналу. Під час дослідження, було встановлено, що саме PDF-документ — один з найчастіших видів файлів, що подаються на розгляд при виконанні роботи, зокрема, наукової. Тому було надзвичайно важливо пропрацювати саме цей формат, але саме з погляду унікальності зображень, а не лише з текстового аспекту. Під час виконання роботи вдалося подолати дуже складну перешкоду — створити правильний алгоритм для парсування PDF-документів, щоб коректно забезпечити функціонал застосунку. Крім того, дуже важливою ланкою системи є налагодження зв'язку з Google Cloud Vision API, що було новим досвідом при написанні коду програми та надало нові знання з теми інтеграції в розробку онлайн-плагінів. Не менш важливим здобутком цієї роботи є також досягнення правильного та коректного виводу для зручності користувачів, це було досягнено з допомогою CSV-файлу результатів аналізу, що є зручним для проглядання у Microsoft Excel.

program.py

```
import os
import PySimpleGUI as sg
import cloud_vision
import cv2
import glob
import csv

layout = [
    [sg.Text('Choose file'), sg.InputText(key="file"), sg.File-
Browse(file_types=("PDF Files", "*.pdf"))],
    [sg.Output(size=(100, 30), key='Output')],
    [sg.Submit(), sg.Cancel()]
]

window = sg.Window('Plagiarism check', layout)

while True:
    event, values = window.read()
    if event in (None, 'Exit', 'Cancel'):
        break
    if event == 'Submit':
        f = open('result.csv', 'w')
        wr = csv.writer(f, delimiter=';', lineterminator='\n')

        query = 'pdfimages ' + values['file'] + ' pdf_images/i'
        os.system(query)
        s = ""
        counter = 0
        for file in glob.glob('pdf_images/*'):
```

```
            i = cv2.imread(file)
            name = "pdf_images/" + str(counter) + ".jpg"
            cv2.imwrite(name, i)
            counter += 1

        files = glob.glob('pdf_images/*.ppm')
        for f in files:
            os.remove(f)

        for file in glob.glob('pdf_images/*'):
            res = cloud_vision.main_method("/Users/alisa/Desktop/
КУРСОВАЯ/pract/" + file)
            for obj in res:
                s += str(obj[1]) + "% " + str(obj[0]) + "\n"
                wr.writerow(obj)
            wr.writerow("")
            s += "\n"

        window['Output'].update(s)

        files = glob.glob('pdf_images/*.jpg')
        for f in files:
            os.remove(f)
```

cloud_vision.py

```
import os, io
from google.cloud import vision
from google.protobuf.json_format import MessageToJson
import json
import requests
import glob
import ssim


def algorithm(images, filepath):
    data = {}
    counter = 1
    for link in images:
        try:
            path = "/Users/alisa/Desktop/КУРСОВАЯ/pract/images/f"
+ str(counter) + ".jpg"
            r = requests.get(link)
            with open(path, 'wb') as outfile:
                outfile.write(r.content)
            data[link] = round(ssim.SSIM_method(filepath,
path)*100, 2)
            counter += 1
        except:
            continue
    return data


def main_method(link):
    os.environ["GOOGLE_APPLICATION_CREDENTIALS"] =
r'vision_key.json'
    client = vision.ImageAnnotatorClient()
```

```
    with io.open(link, 'rb') as image_file:
        content = image_file.read()
    image = vision.types.Image(content=content)
    response = client.web_detection(image=image)
    print(response)
    response = json.loads(MessageToJson(response,
preserving_proto_field_name=True))
    webdetection = response["web_detection"]

    urls = []
    try:
        full_matching = webdetection["full_matching_images"]
        for obj in full_matching:
            urls.append(obj['url'])
    except:
        print("no full_matching")

    try:
        partial_matching = webdetection["partial_matching_images"]
        for obj in partial_matching:
            urls.append(obj['url'])
    except:
        print("no partial_matching")

    try:
        visually_similar = webdetection["visually_similar_images"]
        for obj in visually_similar:
            urls.append(obj['url'])
    except:
        print("no visually_similar")

    files = glob.glob('/Users/alisa/Desktop/КУРСОВАЯ/pract/images/
*')
    for f in files:
```

```
        os.remove(f)


    stats = algorithm(urls, link)
    result = sorted(stats.items(), key=lambda x: x[1],
reverse=True)
    return result
```

ssim.py

```
from skimage.metrics import structural_similarity
import cv2


def SSIM_method(link1, link2):
    img1 = cv2.imread(link1)
    img2 = cv2.imread(link2)
    (height, width, c) = img1.shape
    img2 = cv2.resize(img2, (width, height))
    (score, res) = structural_similarity(img1, img2,
multichannel=True, full=True)
    return score
```

Список використаних джерел

1. Розподіл ринку пошукових систем. Режим доступу: https://netmarketshare.com/search-engine-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Desktop%2Flaptop%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Trend%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22searchEngine%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22searchEngines-Desktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222019-05%22%2C%22dateEnd%22%3A%222020-04%22%2C%22segments%22%3A%22-1000%22%7D

2. Сервіс TinEye. Режим доступу: https://tineye.com/faq#what

3. Сервіс Pixsy. Режим доступу: www.pixsy.com

4. Дані про умови користування Google Cloud Vision API. Режим доступу: https://cloud.google.com/vision/pricing

5. Дані про функції Google Cloud Vision API. Режим доступу: https://cloud.google.com/vision/docs/features-list?hl=ru

6. Reverse Image Search. Режим доступу: https://www.forbes.com/sites/kalevleetaru/2019/06/01/using-google-vision-ais-reverse-image-search-to-richly-catalog-television-news/#450f554e1d59

7. MSE vs. SSIM / S.A. Gandhi, C.V. Kulkarni. Режим доступу: https://www.ijser.org/researchpaper/MSE-Vs-SSIM.pdf

8. Image Quality Assessment through FSIM, SSIM, MSE and PSNR — A Comparative Study / U. Sara, M. Akter, M.S. Uddin. Режим доступу: https://pdfs.semanticscholar.org/955f/e7d24bf21963137db8603a18aba11b2f139b.pdf?_ga=2.237945152.1708071393.1589208148-45286010.1589067277

9. ANALYTICAL RELATION & COMPARISON OF PSNR AND SSIM ON BABBON IMAGE AND HUMAN EYE PERCEPTION USING MATLAB / M. Goyal, Y. Lather, V. Lather. Режим доступу: https://pdfs.semanticscholar.org/

8 6 e 3 / 1 e d e 1 a 9 c c 7 4 d b 7 f 0 7 f 7 0 d 8 2 b 2 1 6 a 8 d 2 c a f b 3 . p d f ? _ga=2.166186238.1708071393.1589208148-45286010.1589067277

10. Image quality metrics: PSNR vs. SSIM / A. Hore, D. Ziou. Режим доступу: https://www.researchgate.net/publication/220931731_Image_quality_metrics_PSNR_vs_SSIM

11. Mystery behind similarity measures MSE and SSIM / G. Palubinskas. Режим доступу: https://www.researchgate.net/publication/301412008_Mystery_behind_similarity_measures_mse_and_SSIM