УДК 008.5

Білощицький Андрій Олександрович
Доктор технічних наук, завідувач кафедри інформаційних технологій, *ORCID: 0000-0001-9548-1959*
Київський національний університет будівництва і архітектури, Київ
Діхтяренко Олександр Васильович
Аспірант кафедри основ інформатики, *ORCID: 0000-0001-9477-6149*
Київський національний університет будівництва і архітектури, Київ
Палій Сергій Володимирович
Кандидат технічних наук, доцент кафедри основ інформатики, *ORCID: 0000-0001-9742-1116*
Київський національний університет будівництва і архітектури, Київ

СПОСОБИ ПОШУКУ НЕТОЧНИХ ДУБЛІКАТІВ ЗОБРАЖЕНЬ В НАУКОВИХ РОБОТАХ

Анотація. *Наведено основні способи приховування запозичень. Зроблено аналіз популярних рішень, що використовуються для пошуку неповних дублікатів зображень. Описані можливі методи та підходи до пошуку дублікатів графічних елементів у наукових роботах. Розглянуто принципи роботи алгоритмів індексації зображень і випробувано один з найперспективніших методів. Отримані дані свідчать про відсутність готового рішення, яке задовольняє висунуті вимоги, а саме знаходження тільки дублікатів зображення, а не схожих на нього, а також нечутливість до таких модифікацій, як обрізання, віддзеркалення, поворот.*

Ключові слова: дублікати зображень; хешування зображень; averag ehash; difference hash; perceptual hash

Вступ

Графічні зображення у наукових роботах є не менш важливою складовою ніж текст, а інколи без них неможливо розкрити тему. Крім того, деякі типи зображень самі по собі є об'єктами авторського права [1] і захищаються законом України про захист авторських прав [1]. Але це ніяк не заважає копіювати чи сканувати зображення і видавати їх за свої. Для того щоб знайти дублікати чи запозичення зображень в документах, необхідно визначити, які саме графічні елементи вважаються схожими. Очевидно, що такими є повні дублікати, які можуть в свою чергу бути зменшені чи розтягнуті. При копіюванні чужих зображень, плагіатор може вдатися до різних прийомів, але основна задача як і при інших видах запозичення – зробити зображення візуально не подібним до оригіналу і зберегти при цьому його інформативну цінність. До даних модифікацій можна віднести зміну яскравості, контрасту, зменшення гамми кольорів (переведення зображення у відтінки сірого) тощо. Серед модифікацій, які впливають на інформативність зображення, але також можуть використовуватися в деяких випадках – обрізання зображення або склеювання з кількох елементів в один. Такі маніпуляції виконуються дуже просто, тому на це також необхідно звернути увагу.

Мета статті

Мета статті – перевірка запозичень не лише в текстових фрагментах наукових робіт, а й зображеннях також.

Виклад основного матеріалу

Зображення в загальному доступі

Якщо для одного автора ілюстрація в науковій роботі – це фото відомої картини, то інший автор може отримати майже ідентичне зображення зробивши знімок власноручно. Аналогічна ситуація із зображеннями, які знаходяться у вільному доступі та можуть використовуватися без жодних обмежень.

З одного боку такі зображення не є запозиченнями, хоча вони і повністю ідентичні, а з іншого – цінність зображення може бути саме в контексті його використання, якщо автор серед можливої множини рішень використав саме це зображення. Комп'ютерна програма не в змозі оцінити вміст зображення та зробити висновок щодо ліцензії, під якою розповсюджується дане зображення, тому остаточне рішення має робити експерт, який перевіряє роботу за допомогою програми.

Проте, існує можливість визначення зображень, які знаходяться у широкому доступі та використовуються без обмежень. Із великого масиву наукових робіт можна зробити вибірку однакових зображень. Зображення, які будуть повторюватися у багатьох роботах найчастіше, знаходяться у відкритому доступі. Тому за відсутності інших ознак плагіату, такі зображення можна пропускати і не позначати як збіги [9].

Способи визначення дублікатів зображень

Для знаходження дублікатів зображень можна використовувати методи пошуку схожих файлів, сигнатури MD5 для пошуку повністю ідентичних файлів або локально чутливе хешування (ssdeep). Однак дані способи недоцільні, оскільки навіть зміна формату файлу (наприклад, конвертація JPEG у PNG) повністю змінить бінарну структуру файлу. Тому методи пошуку, що базуються на обробці зображень як бінарних файлів недоцільні, хіба що ми хочемо знайти абсолютні дублікати. Але у випадку використання хешування файлів зображень за допомогою алгоритму MD5 (чи іншої подібної функції), навіть редагування EXIF інформації призведе то того, що файли вважатимуться різними. Тому необхідно використовувати алгоритми, що розглядають зображення як графічний об'єкт, а не бінарний файл. Існує багато типів та форматів зображень, але всі вони можуть бути приведені до растрової графіки і збережені в одному з популярних форматів (PNG, TIFF, JPEG). Кожне зображення (в даному випадку мова йде про растрову графіку) складається з окремих точок – пікселів. Кожен піксел має свій колір і позицію в зображенні. Для опису кольорів найчастіше використовується модель RGB (red, green, blue), в якій кожен колір створюється поєднанням трьох основних кольорів у різних пропорціях. Таким чином колір можна задати і числовими значеннями, якими легко маніпулювати. Прийнявши максимальну глибину кольору 24 біта (~ 16 мільйонів кольорів), матимемо значення кольорів від RGB(0,0,0) до RGB(255,255,255), тобто один колір може мати інтенсивність від 0 до 255 одиниць. Дана модель опису кольору значно спрощує методи маніпуляцій із зображеннями.

Головні вимоги до способу пошуку дублікатів, точніше до результатів його роботи, це максимальна точність і мінімум помилок. Алгоритм повинен знаходити всі явні дублікати (ті, в яких змінені лише кольори, розміри чи формат), але мінімізувати кількість роботи для оператора системи, тобто не показувати «подібні» зображення, а лише дублікати. Крім того, алгоритм має знаходити дублікати, які зазнали модифікацій, які легко виконуються: поворот, віддзеркалення, зміна кольорів, обрізання зображення.

Серед підходів до обробки графічної інформації можна виділити два основних напрямки: визначення ключових точок зображення та використання локально-чутливого хешування. Ці методи можуть бути поєднані і загалом дають добрі результати у разі пошуку схожих зображень. У роботах [3; 4] автори спочатку визначали ключові точки на зображенні (рис. 1), а потім ділили зображення на дрібні фрагменти. Виконуючи індексацію кожного фрагмента окремо, отримували масив сигнатур, який відповідав за зображення в цілому. Використовуючи міру Хеммінга [4], знаходили однотипні зображення, навіть при 90% обрізанні зображення.

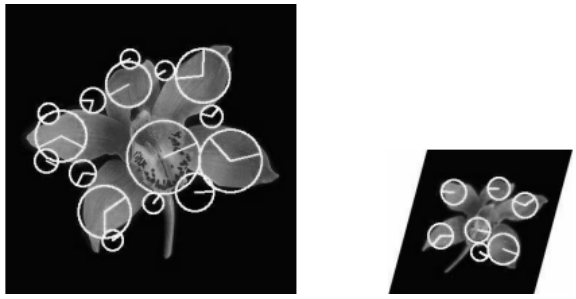


Рисунок 1 - Ключові точки на оригінальному та модифікованому зображеннях

Описаний спосіб охоплює максимальну кількість можливих модифікацій, які може зазнати зображення. Однак є одна проблема – велика ймовірність хибних результатів. Спосіб знаходить зображення, які «схожі» на задане, а нам потрібні лише дублікати з максимально можливою точністю. Маючи велику кількість помилкових спрацювань, ми перевантажуємо роботою експерта, який буде працювати із системою.

У роботі [5] автори розробили власний спосіб визначення ключових особливостей зображення. На відміну від визначення ключових точок, в даному випадку особливості зображення описували за допомогою векторів. Отримані набори векторів були основою для створення сигнатури зображення, для хешування використовувалася локально-чутлива функція minHash [6; 7]. Метод названий min-Hash and tf-idfWeighting. Основна задача, яка перед ним стояла, це швидке знаходження схожих зображень у великих масивах даних. Частину результатів, які отримали автори при тестуванні свого методу з використанням бази, що містить 10200 зображень, наведено на рис. 2. Метод знаходить схожі зображення, навіть якщо це різні зображення одного предмета, але також має багато помилкових спрацювань.

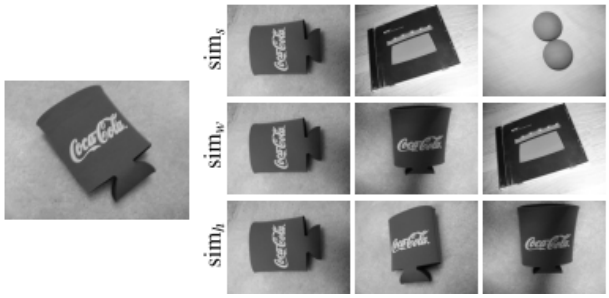


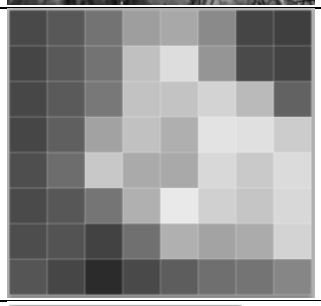
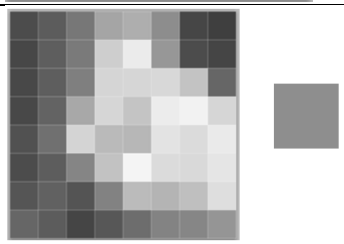
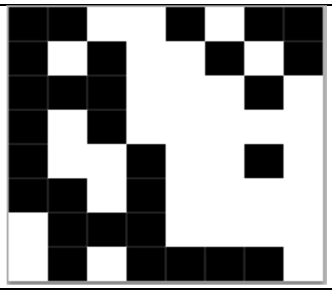
Рисунок 2 - Результати роботи алгоритму min-Hash and tf-idfWeighting

Розглянемо три інші можливі методи індексації зображень.

Хешування по середньому значенню (Average Hash)

Даний тип індексації є одним з найпростіших, але він підходить для пошуку однакових зображень там, де використання запозиченої графіки не намагаються приховати. Сам алгоритм доволі простий, тому для демонстрації створення локально чутливого хешування використаємо саме його та випадкове зображення. Послідовність виконання операцій та отримані результати наведені у табл. 1.

Таблиця 1 – ***Послідовність індексації зображення***

№	Зображення на етапі	Коментарі
1		Вхідне зображення може бути будь-якого розміру
2		Зображення приведене до розміру 8x8 пікселів та переведене у відтінки сірого
3		Визначений середній колір зображення
4		Результат порівняння кожного пікселя зображення з середнім кольором

Перший крок – це зменшення роздільної здатності картинки, наприклад до 8x8 пікселів, відповідно зміна пропорцій сторін до 1:1. Отримуємо прямокутник 64-ох точок. Далі зменшуємо кількість кольорів у зображенні, перевівши його у відтінки сірого. Будь-який сірий колір в моделі RGB має однакові значення всіх трьох кольорів (наприклад, RGB(75,75,75)), тому для розрахунків вже можна брати лише одне число з трьох. Маємо зображення з 64 пікселів, кожен з яких має значення кольору від 0 до 255, і тепер можемо підрахувати середнє значення кольору. Наступна операція – переведення зображення у бітовий формат.

Знаючи середній колір зображення, проходимо по кожному пікселю зображення, порівнюючи його із середнім. Створюємо сигнатуру зображення: по кожному кроку, якщо піксел темніший середнього, додаємо в сигнатуру одиницю, якщо світліший – 0. Таким чином отримуємо рисунок в бітовому форматі (табл. 1, пункт 4), який можна також відобразити у вигляді двійкового числа. Отримане число можна перевести в будь-яку іншу систему числення. Отримані сигнатури різних зображень, їх можна порівнювати між собою за допомогою метрики Хеммінга, визначаючи відсоток подібності. Це дуже зручно, оскільки отримані сигнатури можна зберігати в базі для того щоб не виконувати дані операції кожного разу. Таким чином, індексація кожного зображення відбувається лише один раз.

Хеш по різниці (Differencehash)

Даний метод мало відрізняється від попереднього. Перші перетворення здійснюються за таким же принципом: спочатку зменшення розмірів та кількості кольорів, а середній колір зображення не враховується. При формуванні числової сигнатури враховується значення кольору попереднього пікселя, якщо попередній був темніший, додаємо до сигнатури 0, якщо навпаки – одиницю. Таким чином враховується лише локальна зміна кольору без врахування загальних особливостей зображення. Це дозволяє отримувати кращі результати розпізнавання дублікатів, якщо до копії була застосована нелінійна зміна кольорів. Хеш по різниці, як і хеш по середньому значенню, добре підходить для визначення дублікатів без значних змін, тобто коли автори робіт не використовують або не намагаються приховати копії зображень.

Перцептивний хеш (Perceptual Hash)

Даний метод відрізняється від описаних двома основними факторами. Для даного способу створення сигнатури необхідно зменшувати зображення не до 8x8 пікселів, а як мінімум до 32x32. Замість приведення зображення до сірої гамми використовується дискретне косинусне перетворення, таким чином виділяючи характерні особливості зображення. Двійкове значення сигнатури створюється так само, як і за попереднім методом. Цей метод дає теоретично добрі результати, оскільки він визначає особливості зображення і при хешуванні враховуються всі ці особливості (без поділу зображення на фрагменти), що теоретично має мінімізувати кількість помилкових спрацювань алгоритму.

Перевірка можливостей PerceptualHash

Для оцінки можливостей бібліотеки, була створена колекція зображень, частина з яких були дублікатами, але з використанням різних модифікацій. Тестування виконувалося за допомогою бібліотеки phashion [7]. Для оцінки розбіжності зображень використовувався досить простий код:

```
require 'phashion'
img1 = Phashion::Image.new(filename1)
img2 = Phashion::Image.new(filename2)
img1.distance_from(img2)
```

До дублікатів оригінального зображення, застосовувалися такі модифікації: нелінійна зміна кольорів, накладання тексту, обрізання (5, 20, 50%), віддзеркалення та поворот (5°, 90°, 180°). Також було взято два зображення, які не були дублікатами. У табл. 2 наведені результати роботи алгоритму, а саме отримана різниця (міра Хеммінга) між початковим зображенням та модифікованим. За отриманими даними можна зробити висновки, що метод підрахунку перцептивного хешу дає непогані результати лише на незначних модифікаціях. Такі модифікації, як зміна кольорів, накладання тексту та незначне обрізання зображення (до 5% площі) були виявлені. З нахиленими чи повернутими зображеннями алгоритм працює значно гірше. Можна виявляти зображення, якщо вони нахилені не більш ніж на 5°, але й це потребує збільшення допуску, максимальної різниці, за якої зображення вважаються дублікатами, що в свою чергу може призвести до помилкових спрацювань.

Таблиця 2 – ***Результати роботи алгоритму Perceptual Hash з різними зображеннями***

№	Зображення	Опис модифікацій	Різниця
1		Оригінальне зображення приведене у відтінки сірого	0
2		Нелінійна зміна кольорів зображення	2
3		Накладений поверх зображення текст	8
4		Незначне обрізання зображення, 5% від низу зображення	8
5		Обрізання приблизно 20% зображення	32
6		Обрізання 50% зображення	34
7		Віддзеркалене по горизонталі зображення	34
8		Поворот на 5°	14
9		Поворот на 90°	30
10		Поворот на 180°	30
11		Несхоже зображення взяте для прикладу	30
12		Несхоже зображення, взяте для прикладу	34

Висновки

Кожен із описаних методів має свої недоліки. Хеш по середньому значенню добре справляється з пошуком дублікатів зображень, що були модифіковані за допомогою лінійної зміни кольорів і працює швидше інших описаних методів, але він дуже чутливий до локальних модифікацій зображення, до нелінійної зміни кольорів та модифікації зображення методом обрізання чи склеювання. У таких випадках різниця між сигнатурами виходить достатньо велика, щоб не помітити дублікати.

Хеш по різниці працює так само швидко і усуває один з недоліків – чутливість до нелінійної зміни кольорів. Але решта недоліків притаманна і цьому методу. Крім того, обидва методи мають достатньо велику ймовірність помилкових спрацювань, що обмежує можливість їх використання при опрацюванні великої кількості зображень. І лише перцептивний хеш нечутливий до незначних локальний модифікацій, до зміни кольорів, але все одно не дає можливості знаходити фрагменти чи склейки.

Незважаючи на певні переваги використання перцептивного хешу над іншими методами, він все одно не дає результат, який задовольняє наші вимоги. Зокрема він не знаходить фрагменти зображення, повернуті та віддзеркалені зображення. Останні дві проблеми вирішуються досить легко. Необхідно будувати хеш не лише для оригінального зображення, а й для його варіантів. Тобто створюється хеш для звичайного зображення, для зображення повернутого на 90°, 180°, віддзеркаленого і так далі. Таким чином отримуємо замість однієї сигнатури цілий набір сигнатур. При перевірці двох зображень перевіряються їхні набори, а у випадку збігу хоча б одного елемента, ці зображення вважаються дублікатами. Для знаходження фрагментів зображення необхідно експериментувати з методом визначення ключових точок та шукати компроміс між точністю та шириною вибірки.

Можна зробити висновок, що спосіб знаходження дублікатів знайдено і при деяких оптимізаціях його можна використовувати. Як напрямок для подальших досліджень можна взяти метод визначення ключових точок на зображенні та алгоритми пошуку дублікатів, що базуються на ключових точках. Необхідно оцінити точність алгоритму з урахуванням специфіки зображень, що перевірятимуться (графіки, діаграми тощо).

Список літератури

1. *Закон України «Про авторське право і суміжні права» № 3729-12 від 05.12.2012, підстава 5460-17*
2. *Білощицький, А.О. Ефективність методів пошуку збігів у текстах / А.О. Білощицький, О.В. Діхтяренко // Управління розвитком складних систем. – 2013. – № 14.– С. 144 – 147.*
3. *Ke, Y., Sukthankar, R., Huston, L., Ke, Y., & Sukthankar, R. (2004, October). Efficient near-duplicatedetectionand sub-imageretrieval. In ACM Multimedia (Vol. 4, No. 1, p. 5).*
4. *Lv, X., & Wang, Z. J. (2012). Perceptual image hashing based on shape contexts and local feature points. Information Forensics and Security, IEEE Transactions on, 7(3), 1081-1093.*
5. *Hamming, R. W. (1950). Error detecting and error correcting codes. Bell System technical journal, 29(2), 147-160.*
6. *Chum, O., Philbin, J., &Zisserman, A. (2008, September). NearDuplicateImageDetection: min-Hash and tf-idfWeighting. In BMVC (Vol. 810, pp. 812-815).*
7. *Leskovec, J., Rajaraman, A., &Ullman, J. D. (2014). Miningofmassivedatasets. CambridgeUniversityPress..*
8. *Platter W., Phashion, (2014), GitHubrepository, https://github.com/westonplatter/phashion*
9. *Білощицький А.О. Оптимізація системи пошуку збігів за допомогою використання алгоритмів локально чутливого хешування наборів текстових даних/ А.О. Білощицький, О.В. Діхтяренко // Управління розвитком складних систем. – 2014. – № 19.– С. 113 – 117.*
10. *Богунский, В.Д. Обоснование закона о конкурентных свойствах проектов / В.Д. Богунский, С.В. Руденко, П.А. Тесленко // Управління розвитком складних систем. –2011. – № 8. – С. 14 – 16.*
11. *Оборський, Г.О. Стандартизація і сертифікація процесів управління якістю освіти у вищому навчальному закладі / Г.О. Оборський, В.Д. Богунський, О.С. Савельєва // Тр. Одес. политехн. ун-та. –2011. – № 1(35). – С. 251 – 255.*
12. *Колесникова, Е.В. Моделирование слабо структурированных систем проектного управления / Е.В. Колесникова // Тр.Одес. политехн. ун-та. – 2013. - № 3 (42). – С. 127 – 131.*
13. *Колесникова, Е.В. Трансформация когнитивных карт в модели марковских процессов для проектов создания программного обеспечения / Е.В. Колесникова, А.А. Негри // Управління розвитком складних систем. - 2013. – №15. – С. 30 – 35.*
14. *Vaysman, V. A. The planar graphs closed cycles determination method / V. A. Vaysman, D. V. Lukianov, K. V. Kolesnikova // Тр. Одес. политехн. ун-та. – 2012. – № 1(38). – С. 222 – 227.*
15. *Burkov, V. N., Biloshchytskyi, A. A., & Gogunsky, V. D. (2013). Options citation of scientific publications in scientometric databases. Management of development of difficult systems. Kyiv, Ukraine: KNUCA, 15, 134 - 139.*
16. *Gogunsky, V. D., Kolyada, A. S., & Iakovenko, V. O. (2014). Scientometric data scientific publication "Management of development of difficult systems. Management of development of difficult systems. Kyiv, Ukraine: KNUCA, 19, 6 – 11.*
17. *Vlasenko, O. V., Lebed' V. V., & Gogunsky, V. D. (2012). Markov model of communication processes in international projects. Management of development of difficult systems. Kyiv, Ukraine: KNUCA: 12, 35 - 39.*
18. *Gogunsky, V. D., Iakovenko, V. O., & Kolyada, A. S. (2014). Application of Latent Dirichlet allocation for the analysis of scientometric publications database. Proc. of Odes. Polytechnic. Univ. Odessa, Ukraine, ONPU: 1 (43), 186 – 191.*

Стаття надійшла до редколегії 23.12.2014

Рецензент: д-р техн. наук, проф. С.Д. Бушуєв, Київський національний університет будівництва і архітектури, Київ.

Белощицкий Андрей Александрович
Доктор технических наук, профессор кафедры информационных технологий, *ORCID: 0000-0001-9548-1959*
Киевский национальный университет строительства и архитектуры, Киев
Дихтяренко Александр Васильевич
Аспирант кафедры основ информатики, *ORCID: 0000-0001-9477-6149*
Киевский национальный университет строительства и архитектуры, Киев
Палий Сергей Владимирович
Кандидат технических наук, доцент кафедры основ информатики, *ORCID: 0000-0001-9742-1116*
Киевский национальный университет строительства и архитектуры, Киев

СПОСОБЫ ПОИСКА НЕТОЧНЫХ ДУБЛИКАТОВ ИЗОБРАЖЕНИЙ В НАУЧНЫХ ТРУДАХ

Аннотация. *Приведены основные способы скрытия заимствований. Сделан анализ популярных решений, которые используются для поиска неточных дубликатов изображений. Описаны возможные методы и подходы к поиску дубликатов графических элементов в научных трудах. Рассмотрены принципы работы алгоритмов индексации изображений и испытан один из самых перспективных методов. Полученные данные свидетельствуют об отсутствии готового решения, которое может удовлетворить поставленные требования, а именно нахождение только дубликатов изображений, а не похожих на него, а также нечувствительность к таким модификациям, как обрезка, отражение и поворот.*

Ключевые слова: дубликаты изображений, хеширование изображений, average hash, difference hash, perceptual hash

Biloshchytskyi Andrii
Doctor of Technical Sciences, Head of information technologies, *ORCID: 0000-0001-9548-1959*
Kyiv National University of Construction and Architecture, Kiev
Dikhtiarenko Oleksandr
Postgraduate at the Department of computer science fundamentals, *ORCID: 0000-0001-9477-6149*
Kyiv National University of Construction and Architecture, Kiev
Paliy Serhiy
Ph.D., assistant professor of computer science, *ORCID: 0000-0001-9742-1116*
Kyiv National University of Construction and Architecture, Kiev

SEARCHING FOR PARTIAL DUPLICATE IMAGES IN SCIENTIFIC WORKS

Abstract. *Graphic elements in scientific works are often as much important as the text content. Unfortunately, because of fast growing of modern methods of communication and transmission of information, there are cases of abuse of free access to information, namely attempts to grant someone else's work for their own. Graphic images, like text information, is very easy to copy. In order to hide the fact of copying, can be used different techniques as basic resizing and more complex ways of image editing. In this paper we highlight the main ways to hide the facts of copying. Analysis of popular solutions, which are used to search for partial duplicate images, was made. We described the possible methods and approaches to search for duplicates of graphical elements in scientific works. The operation principles of algorithms for image indexing, were described also, and we tested one of the most promising methods. The obtained data indicate the absence of ready-made solutions that can meet the set requirements, namely to find duplicate images only, not the similar images, and insensitivity to such modifications as trimming, flipping and rotation.*

Keywords: duplicates of images, image hashing, average hash, difference hash, perceptual hash

References

1. The law of Ukraine on copyright and related rights № 3729-12 on 05.12.2012

2. Biloshchytskyi, A., & Dikhtiarenko, O. (2013). The effectiveness of methods for finding matches in texts. Management of complex systems, 14, pp. 144 – 147.

3. Ke, Y., Sukthankar, R., Huston, L., Ke, Y., & Sukthankar, R. (2004, October). Efficient near-duplicate detection and sub-image retrieval. In ACM Multimedia (Vol. 4, No. 1, p. 5).

4. Lv, X., & Wang, Z. J. (2012). Perceptual image hashing based on shape contexts and local feature points. Information Forensics and Security, IEEE Transactions on, 7(3), 1081-1093.

5. Hamming, R. W. (1950). Error detecting and error correcting codes. Bell System technical journal, 29(2), 147-160.

6. Chum, O., Philbin, J., & Zisserman, A. (2008, September). Near Duplicate Image Detection: min-Hash and tf-idf Weighting. In BMVC (Vol. 810, pp. 812-815).

7. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). Mining of massive datasets. Cambridge University Press..

8. Platter W., & Phashion, (2014), GitHub repository, https://github.com/westonplatter/phashion

9. Biloshchytskyi, A., & Dikhtiarenko, O. (2014). Optimization of Matching algorithms by using local-sensitive hash sets of text data. Management of complex systems, 19, pp. 113 – 117.

10. Gogunsky, V. D., Rudenko, S. V., & Teslenko, P. A. (2012). Justification law on competitive properties of projects. Management of development of difficult systems. Kyiv, Ukraine, KNUCA: 8, 14 - 16.

11. Oborsky, G. A., Gogunsky, V. D., & Saveleva O. S. (2011). Standardization and certification processes of the quality management education in higher education. Proceedings of Odes. Polytechnic. Univ, 1 (35), 251 – 255.

12. Kolesnikova, K. V. (2013). Modeling weakly structured project management systems. Proceedings of Odes. Polytechnic. Univ, 3 (42), 127 – 131.

13. Kolesnikova, K. V., & Negri, A. A. (2013). Transformation of cognitive maps in the model of Markov processes for projects creating software. Management of development of difficult systems. Kyiv, Ukraine: KNUCA, 15, 30 – 35.

14. Vaysman, V. A. Lukianov, D. V. & Kolesnikova, K. V. (2012). The planar graphs closed cycles determination method. Proceedings of Odes. Polytechnic. Univ, 1(38), 222 – 227.

15. Burkov, V. N., Biloshchytskyi, A. A., & Gogunsky, V. D. (2013). Options citation of scientific publications in scientometric databases. Management of development of difficult systems. Kyiv, Ukraine: KNUCA, 15, 134 - 139.

16. Gogunsky, V. D., Kolyada, A. S., & Iakovenko, V. O. (2014). Scientometric data scientific publication "Management of development of difficult systems. Management of development of difficult systems. Kyiv, Ukraine: KNUCA, 19, 6 – 11.

17. Vlasenko, O. V., Lebed' V. V., & Gogunsky, V. D (2012). Markov model of communication processes in international projects. Management of development of difficult systems. Kyiv, Ukraine: KNUCA: 12, 35 - 39.

18. Gogunsky, V. D., Iakovenko, V. O., & Kolyada, A. S. (2014). Application of Latent Dirichlet allocation for the analysis of scientometric publications database. Proc. of Odes. Polytechnic. Univ. Odessa, Ukraine, ONPU: 1 (43), 186 – 191.

Посилання на публікацію

APA *Biloshchytskyi, A., Dikhtiarenko, O., & Paliy, S. (2015). Searching for partial duplicate images in scientific works. Management of Development of Complex Systems, Issue 21, P. 149 – 155 [in Ukrainian].*

ГОСТ *Білощицький А.О. Способи пошуку неточних дублікатів зображень в наукових роботах [Текст] / А.О. Білощицький, О.В. Діхтяренко, С.В. Палій // Управління розвитком складних систем. – 2015. - № 21. – С. 149 - 155.*