Reverse-search System of Similar or Identical Images

Oleh Veres[1], Yaroslav Kis[2], Vladyslav Kugivchak[3], Igor Rishniak[4]

Information Systems and Network Department, Lviv Polytechnic National University,
Bandery str., 12, Lviv, Ukraine, 79013
Oleh.M.Veres@lpnu.ua[1], Yaroslav.P.Kis@lpnu.ua[2],
Vladyslav.Kuhivchak.MKN.2017@lpnu.ua[3], rishnyakiv@gmail.com[4]

Abstract. The article algorithms of image analysis on the number of errors of operation are investigated. The prototype of the system was created, and testing of the described methods was carried out. The result of the analysis became the basis for the information system project of reverse search of similar or identical images.

Keywords*:* analysis, detector, descriptor, image, key point, method, pixel, hashing.

1 Introduction

Graphic images are an important part of the information resource. Some types of images are subject to copyright and are protected by law. Therefore, it is necessary to determine exactly which graphic elements are considered similar. Obviously, these are full duplicates that can also be modified.

If, for one author, the illustration in the work under study is a photo of a known painting, then another author can get almost identical image by making a picture of himself. A similar situation with images that are freely accessible and can be used without any restrictions.

The computer program is not able to evaluate the content of the image and make a conclusion about the license, so the final decision is taken by the expert.

2 The Analysis of Recent Researches and Publications

Similar file search methods, can be used MD5 signatures search for completely identical files are locally sensitive hashing to find identical images. However, these methods are not feasible, since even changing the file format (for example, converting JPEG to PNG) completely changes the binary structure of the file. Therefore, the search methods based on image processing as binary files are inappropriate, unless absolutely identical ones are to be found. But in the case of using an image file hashing using the MD5 (or similar function) algorithm, even editing EXIF

COLINS'2018, Volume II: Workshop. Lviv, Ukraine, June 25-27, 2018, ISSN 2523-4013
http://colins.in.ua, online

information will result in files being considered differently. Therefore, it is necessary to use algorithms that consider the image as a graphic object, and not as a binary file. There are many types and image formats, but they can all be rendered in raster graphics and saved in one of the popular formats (PNG, TIFF, JPEG). Each image (in this case it is a raster graph) consists of individual points - pixels. Each pixel has its own color and position in the image. The most commonly used RGB (red, green, blue) model is the color scheme, where each color is created by combining three basic colors in different proportions. In this way, the color numerical values that are easy to manipulate can be specified. By accepting the maximum depth of 24 bits (~ 16 million colors), there will be the color values from RGB (0,0,0) to RGB (255,255,255), that is, one color may have an intensity from 0 to 255 units. This color descriptor model greatly simplifies image manipulation techniques.

There are two main areas for processing of graphic information: the definition of key points on the image and the use of locally sensitive hashing. These methods can be combined. They give good results in finding similar images. Using Hamming's measure [1], you can find the same type of image, even with 90% cropping the image. The method has a high probability of false results.

Own way of identifying key features developed image A. O. Biloshchitsky and O. V. Dichtyarenko [2]. The image is described using vectors. The method is named min-Hash and tf-idfWeighting.

The most popular search duplicate images are three indexing methods: Average Hash, Difference Hash, Perceptual Hash.

To find similar images, a method is used to select key points [3-4].

So, to find snippets of an image or similar content of the illustrations - you need to experiment with the methods of determining key points, each of which also has its own set of advantages and disadvantages.

The main methods used to construct detectors and descriptors are: FAST [5]; SIFT [6]; ORB [7, 8]; AKAZE [9]; BRIEF [10]; BRISK [11].

Today there are many systems for image recognition. The most popular ones are: TinEye, Google Similar Images, Yandex. Maps, AntiDupl.NET.

TinEye. This is the first mechanism for searching images on the Internet, not using key phrases or metadata, but according to a copy of the image. When downloading an image, this program creates a "unique and compact digital signature or imprint" and compares it with other indexed images in its own database of illustrations. This procedure allows recognizing even strongly altered versions of the original image, but usually does not return similar images in the results. Disadvantages: the service only works with file formats: JPG, PNG and GIF; image size not less than 100x100 pixels; file size is no more than 1 MB; Can not upload image gallery; the impossibility of creating one's own database (DB) for work.

Google Similar Images. Google uses the so-called "Reverse image search" mechanism, which eliminates the need to enter keywords and terms in the Google search field. Unlike TinEye, results may include similar images, web results, image pages, and various image permissions. Disadvantages: the impossibility of loading an

image gallery; Only file formats are supported: JPEG, GIF, PNG, BMP, TIFF or WebP; impossibility to create an own database; no flexible user setup for search criteria; it is impossible to perform a search on other search engines.

Yandex. Pictures. When searching for images using the appropriate section of the Yandex, a list of images which are similar to the one selected can be obtained. Duplicates found are not displayed in search results: the presence of duplicates can be seen on the preview page of the duplicate image. Disadvantages: It does not allow creating one's own image database for the search; cannot search in other search engines; is prohibited to use on the territory of Ukraine.

AntiDupl.NET. This program searches for the same and damaged images on a disk. As a rule, modern computer users have numerous image galleries in different formats, and not everyone wants to keep the same one, while taking the disk space. In order not to perform manual searches, the program is created which automates these actions. The search is based on a comparison of the contents of the file, so it is possible to search not only identical, but also modified (similar) images. Disadvantages: Does not allow creating separate sections in the database of images; with a large amount of data in the database (> 10,000) works extremely slowly or generally generates errors; the database is taken from the current computer; accordingly, all users have free access to it; looks for similar images, but does not structure them for the most similar, therefore it causes additional volume work for the user; the program does not process an image with defects (although the description says that it can process it); the program settings are not stored in the future; the program settings are not stored in the future; when the name of a picture in a database is changed, it issues an error and cannot process it.

The information system must not only find all explicit duplicates (those that have changed only the colors, sizes or format), but also "similar" images, while minimizing the amount of work for the system operator. Also, the system should find images that have undergone modifications that are easy to perform: rotation, reflection, color change, image cropping.

Today there is no information system of image analysis, which identifies the same or similar images from a database created images. The reverse search image information system must compare objects with identical or similar objects in the database to find information about the owners of the illustrations. You must also find modified images. This will prevent the use of plagiarism.

3 Research of algorithms for revealing the similarity of different images

To develop a reverse search information system project, a threshold function was searched for duplicate searches, using hashing on average and Hemming's measure. The threshold function should be chosen between 68-88%. The ORB method was

selected based on the analysis of test methods for constructing detectors and descriptors.

It is necessary to analyze the effectiveness of methods for images that have undergone modifications. The result of the research is the basis for the design of an information reverse image search system.

To conduct research, test modules for the program realization of the prototype of the information system of reverse pattern search was created based on their own data sample [12-18].

The algorithms to identify the similarity of different images will be selected, as well as check for errors in the work of each of the methods. To do this, three groups of images are created: identical images, similar images and different images. Each group contains two tests to verify the validity of each algorithm.

Same pictures. Test 1 – two completely identical images (Fig. 1, a). To determine the similarity of images as resources, two completely identical images were taken, each with key points defined and a descriptor created to remove these points and compare them to identity.

Test 2 – to determine the similarity of images as a resource, two completely identical images were taken (Fig. 1, *a* and *b*), one of which was rotated 90° (Fig. 1, *b*), after which for each of them the key points were determined and a descriptor is created to remove these points and compare them to each other for identity.

a) *b*)

c) *d*)

Fig. 1. Same pictures: *a*) original; *b*) reversed by 90 degrees; *c*) changed perspective; *d*) turned 45 degrees

COLINS'2018, Volume II: Workshop. Lviv, Ukraine, June 25-27, 2018, ISSN 2523-4013
http://colins.in.ua, online

Similar images. Test 1 – two similar images (Fig. 1, *a* and *c*), one of which was photographed from a different angle (the main object of the study - the book, is closer) (Fig. 1, *c*). Test 2 – two similar images, one of which was photographed from another angle (Fig. 1, *c*) and the main object of the study is at an angle of 45 degrees (Fig. 1, *d*).

Different images. Test 1 – two completely different images, one of which is an illustration of the radio site "Radiy", and the other – the main page site of the University of Lviv Polytechnic. Test 2 – two completely different images, one of which is a photograph of the National University "Lviv Polytechnic" (Fig. 2, *a*), and the other - "Ivan Franko Lviv National University" (Fig. 2, *b*).

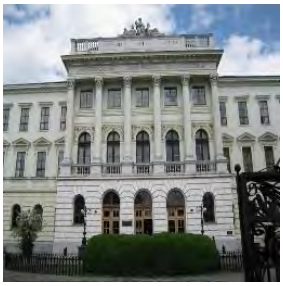
a)  *b*) 

Fig. 2. Different images for the Test 2: *a*) National University "Lviv Polytechnic"; *b*) Ivan Franko Lviv National University

For each image, key points are defined and a descriptor is created to remove these points and compare them to identity.

The results of the tests carried out are presented in the table 1-4. The basic parameters of test results:

- *PK* – number of key points;
- PK_S – a number of similar key points between two images;
- K_S – the percentage of shared key points is the similarity between two images.

Table 1. Results of the study by the ORB method

Key points	*Same pictures*				*Similar images*				*Different images*			
	Test 1		**Test 2**		**Test 1**		**Test 2**		**Test 1**		**Test 2**	
	Fig1	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**
PK	416	416	416	415	416	419	416	396	423	416	455	454
PK_S	416		406		167		73		27		27	
K_S(%)	100		98		40		19		6		6	

The ORB method is well suited to all tests, since the percentage of shared key points decrease according to less similar images. The number of key points obtained

is practically equal in each of the experiments, taking this into consideration, it can be said that this method proves to be stable.

In the percentage ratio, the AKAZE method shows the results at the level with the ORB, but the number of generated key points here is much smaller and not even, so it can be said that the method is stable in the results, but unpredictable as to the number of creating the main points in the image.

Table 2. Results of the study by the AKAZE method

Key points	Same pictures				Similar images				Different images			
	Test 1		**Test 2**		**Test 1**		**Test 2**		**Test 1**		**Test 2**	
	Fig1	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**
PK	33	33	33	33	33	36	33	35	114	129	326	376
PK_S	33		32		13		8		7		16	
K_S(%)	100		97		36		23		5		4	

Table 3. Results of the study by the BRISK method

Key points	Same pictures				Similar images				Different images			
	Test 1		**Test 2**		**Test 1**		**Test 2**		**Test 1**		**Test 2**	
	Fig1	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**
PK	363	363	363	373	363	389	363	492	369	1009	1330	1444
PK_S	361		333		160		72		11		26	
K_S(%)	99		89		41		15		1		2	

The results of the tests with the use of the BRISK method indicate that this algorithm also coped with its task, but in relation to previous methods, it showed the worst result in finding similar and identical images, but was able to clearly distinguish between different illustrations in the tests. The number of key points is not stable and increases with increasing detail in the image.

Table 4. Results of the study by the FAST method

Key points	Same pictures				Similar images				Different images			
	Test 1		**Test 2**		**Test 1**		**Test 2**		**Test 1**		**Test 2**	
	Fig1	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**	**Fig1**	**Fig2**
PK	566	566	566	566	566	560	566	658	411	742	1777	1863
PK_S	566		23		282		33		27		39	
K_S(%)	100		4		50		5		4		2	

The FAST method was the leader in determining the speed of finding key points and deducting the descriptors for them, but did not cope with the tests, and although the number of its key points was much larger than its predecessors, it did not allow them to recognize identical images when they rotated 90 degrees and similar images when rotated 45 degrees.

4 Structural elements of the system reverse-search image

The analysis of the conducted research allows formulating requirements and executing the design of information system of reverse image search.

To find the same images, we use the average value of the hash method. To deduct a measure of similarity is the Hamming distance. The user, at the beginning of the work, downloads the image from the external media, after which the system reads it and reduces to 100 by 100 pixels. Next, the number of colors is reduced, turning it into a black and white image with shades of gray. After receiving a new image, the average pixel color for it is looked for. To do this, it is necessary to go through all the pixels, adding all their colors individually for each RGB and dividing them by their number. Knowing the average color of the image, we pass through each pixel of the image is passed through, comparing it to the average. A signature in the image: for each step is created, if the pixel is darker than the average, a single signature is added and the pixel value to the black color is assigned, if the light is 0 and white. So a bitmap image that can also be displayed as a binary number is assigned, and a completely black and white image without shades of gray is obtained. The same is done with all images of our own database. All the necessary data from using Hamming distance will be obtained. The Hamming distance is defined as the number of bits that differ between the two corresponding input vectors of a fixed length. The larger this distance, the more different the image. If the Hamming distance is zero, this means that the images being studied are completely identical.

The detector is used to determine the key points used. To search for similar key points there is a descriptor. For this, the ORB algorithm is used. To find the similarity between images, that is, the definition of key points and their comparison, the OpenCV open access library is used.

The design of the system is accomplished by means of structural modeling. For better understanding of the relationship between the system and the external environment, data flow diagrams [12] are constructed. The software component of the system is implemented by means of the programming language [13, 14]: Java - as the main language for the development of business logic and user interface, as well as SQL-for work with the database. The analysis of existing technologies of work with information resource is conducted [15-17]. MySQL is selected to create and work with the database of user images.

The developed information system provides the user with the opportunity to make a selection of images based on the input data, to perform their revision, to find the same and similar images, to add new images to the database, to view the data of the owner of the illustration in order to determine the plagiarism [18-49].

5 Conclusion

The algorithms of image analysis on the number of errors of operation are investigated for the development of the information system project. For this purpose, groups of identical images, similar images and totally different images were created. The FAST algorithm did not cope with this task, and therefore, despite its best results in image processing, this method cannot be used. ORB and AKAZE algorithms showed the best test results for all indicators.

According to the results of the tests, ORB method is chosen for implementation in the reverse image search information system, since it generates much more key point per unit time than the AKAZE method.

The chosen method is implemented in the prototype of the reverse-search information system. The system is designed to compare objects with objects in the database that are identical or similar.

Further work will be devoted to the improvement of the prototype of the information system software and the development of an intelligent component for efficient image searching.

References

1. Shapiro, L., Stockman, G.: Computer vision. Washington University (2006)
2. Biloshchyts′kyy, A., Dikhtyarenko, O.: The effectiveness of methods for finding matches in texts. In: Managing the development of complex systems, 14, 144–147 (2013)
3. Shozda, N.: Searching for textures in large databases. In: Informatics, Cybernetics and Computing. Donetsk. Ukraine, 39, 182 - 187 (2002)
4. Biloshchyts′kyy, A., Dikhtyarenko, O.: Optimize the match search system by using algorithms for locally sensitive hashing of text data sets. In: Managing the development of complex systems, 19, 113-117 (2014)
5. Alcantarilla, P., Nuevo, J., Bartoli, A.: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. British Machine Vision Conference (BMVC), (2013)
6. Grewenig, S., Weickert, J., Schroers, C., Bruhn, A.: Cyclic Schemes for PDEBased Image Analysis. In: International Journal of Computer Vision (2013)
7. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF, Computer Vision. (ICCV). In: IEEE International Conference, 2564-2571 (2011)
8. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: 9th European Conference on Computer Vision (ECCV), 430-443 (2006)
9. Yang, X., Cheng, K.: LDB: An ultra-fast feature for scalable augmented reality. In: IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR), 49-57 (2012)
10. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary Robust Independent Elementary Features. In: 11th European Conference on Computer Vision, 778-792 (2010)
11. Leutenegger, S., Chli, M., Siegwart, R.: BRISK: Binary Robust Invariant Scalable Keypoints. Zurich, 2548-2555 (2011)
12. Litvin, V., Shakhovska, N. Designing Information Systems: Teaching. Lviv (2000)
13. Mashnin, T.: JavaFX 2.0. Development of RIA applications. BHV-Petersburg. (2012)

14. Davis, M., Phillips, J.: Studying PHP and MySQL. M .: Symbol-Plus (2008)
15. Shakhovska, N., Bolubash, Y., Veres, O.: Big Data Federated Repository Model. In: The Experience of Designing and Application of CAD Systems in Microelectronics. (2015)
16. Shakhovska, N., Veres, O., Bolubash, Y., Bychkovska-Lipinska, L.: Data space architecture for Big Data managering. In: Xth International Scientific and Technical Conference "Computer Sciences and Information Technologies" (CSIT), 184-187 (2015)
17. Veres, O.: Shakhovska, N. Elements of the formal model big date. XI International Conference on Perspective Technologies and Methods in MEMS Design, 81-83 (2015).
18. Vysotska, V., Chyrun, L., Lytvyn, V.: Methods based on ontologies for information resources processing. Germany: LAP LAMBERT Academic Publishing (2016).
19. Vysotska, V.: Tekhnolohiyi elektronnoyi komertsiyi ta Internet-marketynhu. Saarbrücken, Germany: LAP LAMBERT Academic Publishing (2018)
20. Vysotska, V., Lytvyn, V.: Web resources processing based on ontologies. Saarbrücken, Germany: LAP LAMBERT Academic Publishing (2018)
21. Vysotska, V., Shakhovska, N.: Information technologies of gamification for training and recruitment. Saarbrücken, Germany: LAP LAMBERT Academic Publishing (2018)
22. Vysotska, V.: Internet systems design and development based on Web Mining and NLP. Saarbrücken, Germany: LAP LAMBERT Academic Publishing (2018)
23. Vysotska, V.: Computer linguistics for online marketing in information technology : Monograph. Saarbrücken, Germany: LAP LAMBERT Academic Publishing (2018)
24. Lytvyn, V., Vysotska, V., Chyrun, L., Smolarz, A., Naum O.: Intelligent System Structure for Web Resources Processing and Analysis. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 56-74 (2017)
25. Lytvyn, V., Vysotska, V., Wojcik, W., Dosyn, D.: A Method of Construction of Automated Basic Ontology. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 75-83 (2017)
26. Lytvynenko, V., Lurie, I., Radetska, S., Voronenko, M., Kornilovska, N., Partenjucha, D.: Content analysis of some social media of the occupied territories of Ukraine. In: 1st Inter. Conference Computational Linguistics and Intelligent Systems, COLINS, 84–94 (2017)
27. Shepelev, G., Khairova, N.: Methods of comparing interval objects in intelligent computer systems. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 100–109 (2017)
28. Orobinska, O., Chauchat, J.-H., Sharonova, N.: Methods and models of automatic ontology construction for specialized domains (case of the Radiation Security). In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 95–99 (2017)
29. Hamon, T., Grabar, N.: Unsupervised acquisition of morphological resources for Ukrainian. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 20–30 (2017)
30. Grabar, N., Hamon, T.: Creation of a multilingual aligned corpus with Ukrainian as the target language and its exploitation. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 10–19 (2017)
31. Hamon, T.: Biomedical text mining. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, http://colins.in.ua/wp-content/uploads/2017/04/2017COLINS-THAMON-keynote.pdf (2017)
32. Lande, D., Andrushchenko, V., Balagura, I.: An index of authors' popularity for Internet encyclopedia. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 47–55 (2017)

33. Lande, D.: Creation of subject domain models on the basis of monitoring of network information resources. In: 1st Inter. Conference Computational Linguistics and Intelligent Systems, COLINS, http://colins.in.ua/wp-content/uploads/2017/04/Lande.pdf (2017)
34. Protsenko, Y.: Intuition on modern deep learning approaches in computer vision. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, http://colins.in.ua/wp-content/uploads/2017/04/protsenko.pdf (2017)
35. Kolbasin, V.: AI trends, or brief highlights of NIPS 2016. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, http://colins.in.ua/wp-content/uploads/2017/04/CoLlnS_TuS.pdf (2017)
36. Kersten, W.: The Digital Transformation of the Industry – the Logistics Example. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, http://colins.in.ua/wp-content/uploads/2017/04/CoLlnS_TuS.pdf (2017)
37. Shalimov, V.: Big Data – Revolution in Data Storage and Processing. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, http://colins.in.ua/wp-content/uploads/2017/04/BigData_eng.pdf (2017)
38. Hnot, T.: Qualitative content analysis: expertise and case study. In: 1st Inter.l Conference Computational Linguistics and Intelligent Systems, COLINS, http://colins.in.ua/wp-content/uploads/2017/04/Qualitative-content-analysis_expertise-and-case-study.pdf (2017)
39. Romanyshyn, M.: Grammatical Error Correction: why commas matter. In: 1st Inter. Conf. Computational Linguistics and Intelligent Systems, COLINS, http://colins.in.ua/wp-content/uploads/2017/04/Grammatical-Error-Correction-why-commas-matter.pdf. (2017)
40. Yukhno, K., Chubar, E.: Gamification: today and tomorrow. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 139–140 (2017)
41. Pidpruzhnikov, V., Ilchenko, M.: Search optimization and localization of the website of Department of Applied Linguistics. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 137–138 (2017)
42. Olifenko, I., Borysova, N.: Analysis of existing German Corpora. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 135–136 (2017)
43. Kolesnik, A., Khairova, N.: Use of linguistic criteria for estimating of wikipedia articles quality. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 133–134 (2017)
44. Kirkin, S., Melnyk, K.: Intelligent data processing in creating targeted advertising. In: 1st Inter. Conf. Computational Linguistics and Intelligent Systems, COLINS, 131–132 (2017)
45. Hordienko, H., Ilchenko, M.: Development and computerization of an English term system in the fields of drilling and drilling rigs. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 129–130 (2017)
46. Gorbachov, V., Cherednichenko, O.: Improving communication in enterprise solutions: challenges and opportunities. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 127–128 (2017)
47. Didusov, V., Kochueva, Z.: Statistical methods usage of descriptive statistics in corpus linguistic. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 125–126 (2017)
48. Verbinenko, Yu.: Discursive units in scientific texts. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 120–123 (2017)
49. Titova, V., Gnatchuk, I.: Evaluation ofa formalized model for classification of emergency situations. In: 1st International Conference Computational Linguistics and Intelligent Systems, COLINS, 110–119 (2017)