Choosing the Method of Finding Similar Images in the Reverse Search System

Oleh Veres[1], Bohdan Rusyn[2], Anatoliy Sachenko[3], Ihor Rishnyak[4]

[1,4]Information Systems and Network Department, Lviv Polytechnic National University
[2,3]Kazimierz Pulaski University of Technology and Humanities in Radom, Poland
[2]Karpenko Physico-Mechanical Institute of the NAS Ukraine
[3]Reseach Institute for Intelligent Computer Systems, Ternopil National Economic University, Ternopil, Ukraine
Oleh.M.Veres@lpnu.ua[1], rishnyakiv@gmail.com[4]

Abstract. The article describes the research of image analysis methods. The methods of indexing images for the search of duplicate images, as well as methods for finding similar images based on the definition of key points are described. The prototype of the system was created, and testing of the described methods was carried out.

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

CEUR-WS.org/Vol-2136/10000099.pdf

1 Introduction

Graphic images are no less important than text, and sometimes it is impossible to reveal a topic without them. In addition, some types of images themselves are copyright objects and are protected by the copyright law of Ukraine. But this does not prevent you from copying or scanning images and publishing them for your own. In order to find duplicates or borrowing images in documents, it is necessary to determine which graphic elements are considered similar. Obviously, such are full duplicates that can in turn be reduced or stretched. When copying other people's images, a plagiarist can resort to various tricks, but the basic problem as with other forms of borrowing – do not visually similar to the original and keep its informative value. The modifications include changing the brightness, contrast, color gamut (putting the image in grayscale), etc. Among the modifications that affect the information content of the image, but can also be used in some cases is image cropping or gluing of several elements into one.

On the one hand, such images are not borrowings, although they are completely identical, and on the other hand, the value of an image may be precisely in the context of its use, if the author used this illustration in the possible set of solutions. A computer program is not able to assess the content of the image and make a conclusion about the licenses under which this image is licensed, so the final decision has to be made by an expert who checks the work using the program.

2 The Analysis of Recent Researches and Publications

Among approaches to the processing of graphic information into two main areas: the definition of key points on the image and the use of locally-sensitive hashing can be identified. These methods can be combined and generally give good results in the search for similar images. First, the key points in the image are determined, and then the image is divided into small fragments. By performing indexing of each fragment separately, an array of signatures that are responsible for the image as a whole is received. Using Hamming's measure [1], the same type of image was found, even with 90% cropping of the image. The described method covers the maximum number of possible modifications that may be affected by the image. However, there is one problem - the high probability of false results. The method finds an image that is partially similar to a given one, rather than a duplicate with the highest possible accuracy.

A. O. Biloshchitsky and O. V. Dichtyarenko [2] developed their own way of determining the key features of the image. Unlike the definition of key points, in this case the main features of the image were described using vectors. The resulting sets of vectors were the basis for creating an image signature; for the hash, a locally-sensitive minHash function was used. The method is named min-Hash and tf-idfWeighting. The main task is to quickly locate similar images in large data sets. This method finds similar images, even if it is different images of the same subject, but also has a lot of false positives.

The most popular are three methods for indexing images to find duplicate images: Average Hash; Difference Hash; Hash Perceptual.

To find similar images, the method is used to select key points. A key point, or point feature of an image, is a point whose placement stands out against the background of any other point. As features of the point of the image for most modern algorithms a square box is taken, the size of which is 5 by 5 pixels. The process of determining these points in the image is achieved using the method of using a detector and a descriptor. A detector is a method for determining a key point that allocates it to the background of an image. In turn, the descriptors should ensure the invariance of finding the correspondence between the key points of image transformation. A descriptor is a method which allows removing the key points of both images and comparing them with each other. In the case of modifications to research objects, the detector helps find the same key points on both objects [3].

Key points must have a number of features [4]: the difference – each point must be clearly distinguished from others and be unique in its area; invariance – the definition of a key point should be independent of affine transformations; stability – the allocation of such features should be resistant to noise and modifications; interpretation – key points should be allocated so that they can be used for the analysis of correspondences and extraction of the necessary information on their basis.

So, to find snippets of an image or similar content of the illustrations – it is necessary 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 in the construction of detectors and descriptors are FAST, SIFT, ORB, AKAZE, BRIEF, BRISK .

FAST (Features from Accelerated Segment Test). For a point-candidate P, using the Brezenham algorithm, a circle of 16 pixels is constructed. The point is an angle if there are N adjacent pixels on a circle whose intensity is greater than IP + t or the intensity of all less than IP - t, where IP is the intensity of the point P, it is the limiting value. Next, it is necessary to compare the intensity of the vertical and horizontal points on the circle with the intensity at the point P. If for 3 of these points the condition IPi > (IP + t) or IPi < (IP + t), i = 1, ... , 4, then a full test is conducted for all 16 points [5].

SIFT (Scale Invariant Feature Transform). A variable-size space is created, which calculates the functions LoG (Laplacian of Gaussian) with a different smoothing parameter. A point is considered key if it is a local extremum of the Hawsian difference. After the set of expected key points are specified (the points with a small contrast at the boundaries of objects are deleted) and their orientation is determined. For this purpose, a histogram of gradients is constructed in this area, the direction chosen corresponding to the maximum component of the histogram is selected. Points are assigned to all directions that correspond to the values of the components of the histogram, which are larger than the given threshold. Invariant with respect to landslides, rotations, changes in scale [6].

ORB (Oriented FAST and Rotated BRIEF). Uses FAST to find key points. FAST takes the threshold value of the intensity between the central pixel and the area around the pixels around it as a parameter. The ORB [7] uses the FAST-9 modification (the circle radius with the pixels around it is assumed to be 9), since it was the most efficient in terms of performance. After detecting potential key points, Harris's corner detector is used to refine them. To get N key points, first a low threshold in order to get more than N points is used, then they are arranged with the help of the Harris metric and the first N points are selected. To construct the descriptor of the points obtained, a modification of the BRIEF, invariant to the rotation due to additional transformations is used [8].

AKAZE (Accelerated KAZE). Searching for key points is based on non-linear image scaling using the FED (Fast Explicit Diffusion) scheme. As a binary descriptor, M-LDB (Modified-Local Difference Binary) is used. It is currently considered one of the best [9].

BRIEF (Binary Robust Independent Elementary Features). A descriptor that allows representing the original image in the form of binary strings is constructed for domains. The smoothed image is divided into sections and for them a unique set of points nd (x, y) is chosen. Then they compare intensity. As a result, we get a binary string of dimension ND (128, 256 or 512). The obtained descriptors are compared using Hamming's metric [10].

BRISK (Binary Robust Invariant Scalable Keypoints). Gaussian smoothing is applied to the circular areas of potential key points. To determine the direction of the key point, the amount of local gradients is used [11].

The main requirements for a method for finding identical images, more precisely for the results of his work, are the maximum accuracy and minimum errors. 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.

3 Analysis of methods for finding identical images

For the choice of methods for analysis and the search for identical images, it is necessary to explore the methods of indexing images; definition of a measure of similarity; methods used in the construction of detectors and descriptors. The result of the research is the basis for the design of an information reverse image search system [12-19]. 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 [20-27].

To obtain exactly identical, but not similar images, the method of hashing for the average value was analyzed, and for the determination of similarity dimensions, the Hamming distance was used. The main objective of the study is to determine the threshold function, which allows asserting that the images are complete duplicates.

A picture from the database is chosen and the result is shown in (Fig. 1-2).

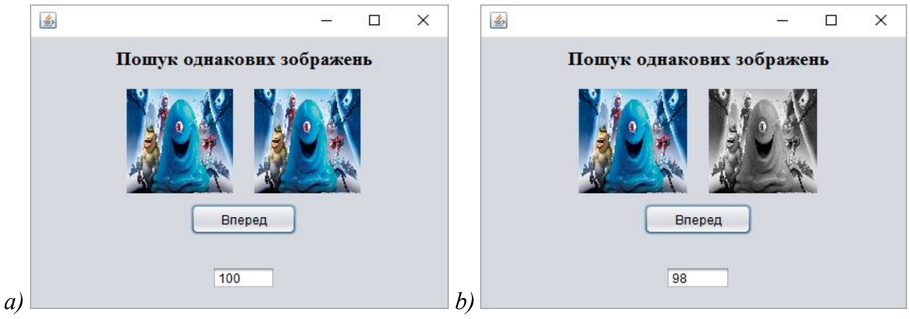

Fig. 1. The result of the processing: *a*) identical images; *b*) images with a color change

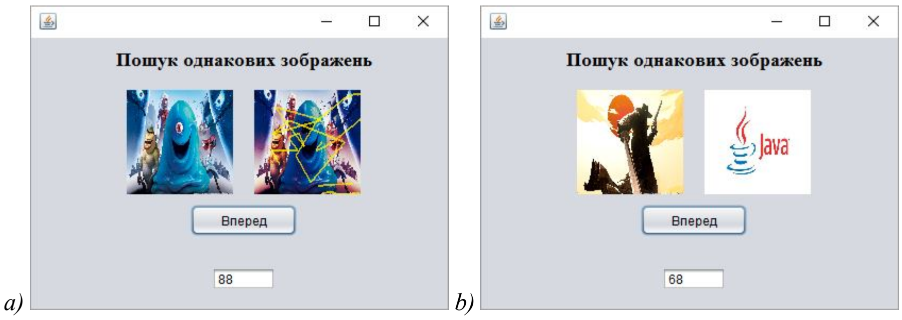

Fig. 2. The result of processing: *a*) with the modification of the illustration; *b*) different images

On the right there is the image of the user, and on the left - the image found in the database, the window below shows the Hamming distance in percentage terms. For identical images, Hemming's distance is 100% – the image is found and it is completely identical (Fig. 1*a*).

Now the task will be complicated, the drawing in black and white with shades of gray is completed, after which, the experiment is repeated (Fig. 1*b*). The system reported minor changes, but still chose the correct image, showing a deviation of only 2%.

Now the brightness and partially paint are changed (Fig. 2*a*). Hamming distance has decreased by 10% to 88%, but the system has found and correctly identified the need.

In the case of different images (Fig. 2*b*), as a result of the program's work, the most similar image was found, but the percentage of similarity has fallen by as much as 32%, indicating that it is impossible to speak of the image as identical.

Consequently, it has been experimentally proved that the threshold function should be set between 68-88%, so the smaller this figure, the more exact similarity is determined, (based on the average colors) of images, rather than full duplicates.

4 Analysis of methods working with control points

In order to find similar content in the image, a comparative analysis of the methods that work with the key points, namely: ORB, BRISK, AKAZE, FAST, respectively, based on the results of the classifier will be conducted. The size of the inbound images was compressed to 128, 256, and 512 pixels on each side. Input images are divided into three groups: 30 images with a lot of details (Tabl. 1); 30 images with a monitor set (Tabl. 2); 30 portrait photographs of people (Tabl. 3).

To submit the results, the following abbreviations are used:

- *PK* – a total number of key points found;
- T_{PK} – the total amount of time spent searching for key points (*ms*);
- T_D – descriptor time (*ms*);
- *S* – the average time to search for a single key point and calculate its descriptor, which looks like:

$$S = (T_{PK} + T_D) / PK \text{ (ms)}; \qquad (1)$$

- *T* – total time spent in the program in seconds (*s*).

Analysis of the results of the first group. All images in this group have a large number of parts located in different places. Information on evaluating methods for various image extensions is provided in Table 1.

The largest number of key points was found using the BRISK method, this number increases in geometric progression, respectively, the higher the resolution of the image under study, the more time will be needed for its processing.

Table 1. Estimation of methods for images of the first group

Method	*PK*	T_{PK}	T_D	*S*	*T*
128x128 pixels					
ORB	10444	247	5199	0,5214	18
BRISK	11768	12496	12533	2,1269	20
AKAZE	5041	972	11128	0,4166	12
FAST	6568	144	4141	0,6524	12
256x256 pixels					
ORB	12311	429	6129	0,5327	20
BRISK	26767	13096	12577	0,9591	44
AKAZE	7286	1872	1930	0,5218	18
FAST	15568	396	5643	0,3879	15
512x512 pixels					
ORB	15719	602	7626	0,5234	22
BRISK	78395	14087	12683	0,3415	58
AKAZE	8688	2777	3541	0,7272	25
FAST	32210	801	8111	0,2767	17

The ORB method was not too sensitive to resizing the image within the selected range, its complexity increases in arithmetic progression. The shortest execution time for the AKAZE descriptor. The FAST method takes the least time to search for similar images.

Analysis of the results of the second group. 30 illustrations of a monitor image are taken, each of which will represent images in different windows of different programs. This group for various image extensions will be analyzed (Tabl. 2).

Table 2. Estimation of methods for images of the second group

Method	*PK*	T_{PK}	T_D	*S*	*T*
128x128 pixels					
ORB	1409	27	422	0,3187	12
BRISK	2178	2917	3014	2,7231	15
AKAZE	995	202	316	0,5206	8
FAST	1024	44	623	0,6514	9
256x256 pixels					
ORB	1661	47	497	0,3278	13
BRISK	4954	3057	3025	1,2276	33
AKAZE	1438	389	541	0,6465	12
FAST	2427	121	849	0,3996	11
512x512 pixels					
ORB	2121	66	619	0,3229	15
BRISK	14509	3288	3050	0,4369	44
AKAZE	1715	577	992	0,915	17
FAST	5022	245	1220	0,2917	13

The number of key points in the sum of all images significantly decreased compared to the first group, which affected the program's run time, the descriptor, and the cost, respectively, the less the key points generate any algorithm, the less time it takes to process them. All time costs are proportional to the number of key points. The results of the algorithms practically do not differ from the previous group, this indicates that their work does not depend on the input data.

Analysis of the results of the third group. For the last group, portraits of 30 people were selected (Tabl. 3).

The average number of generating key points is greater than the same average number of points in the second group, and less than the first. If each algorithm is taken separately, their results for each group are proportional. Consequently, the program's running time depends on the method selected and the number of key points it detects.

Table 3. Estimation of methods for images measuring Estimation of methods for images of the third group

Method	***PK***	T_{PK}	T_D	***S***	***T***
128x128 pixels					
ORB	5306	123	2516	0,3761	13
BRISK	5504	6082	6135	1,914	14
AKAZE	2382	463	570	0,3698	8
FAST	2996	74	1880	0,5145	8
256x256 pixels					
ORB	6254	213	2966	0,5083	20
BRISK	12518	6375	6157	1,0011	44
AKAZE	3443	892	975	0,5424	18
FAST	7101	204	2562	0,3895	15
512x512 pixels					
ORB	8864	332	4097	0,4996	22
BRISK	37030	6925	6271	0,3564	58
AKAZE	4557	1469	1986	0,7582	25
FAST	16309	458	4087	0,2787	17

5 Conclusion

To develop a reverse search information system project, a threshold function was searched for duplicate searches, using hashing on average and Hemming's measure. On the basis of the experimental path, it can assumed that the threshold function should be chosen between 68-88%, accordingly, the smaller this indicator, the more it will determine exactly the similar (based on average colors) of images, rather than full duplicates. Also similar images based on key points were tested by algorithms for finding similar ones. The main element in this study was the time taken to find the key points and compare them to the similarity of the methods: ORB, BRISK, AKAZE and FAST. The worst was the BRISK algorithm, because the number of points generated by them was considerably larger, which led to a rapid increase in processing time. Experimentally it was discovered that the image size of 256x256 pixels is the most optimal for its processing.

6 References

1. L. Shapiro, G. Stockman. Computer vision. Washington University. – 752 p. (2006).
2. A. Biloshchyts′kyy, O. Dikhtyarenko. The effectiveness of methods for finding matches in texts. Managing the development of complex systems, 14, P. 144-147. (2013).
3. N. S. Shozda. Searching for textures in large databases. Informatics, Cybernetics and Computing. Donetsk. Ukraine. n. 39, - P. 182 - 187. (2002). (in Ukrainian).
4. A. Biloshchyts′kyy, O. Dikhtyarenko. Optimize the match search system by using algorithms for locally sensitive hashing of text data sets. Managing the development of complex systems. — № 19. – P. 113 – 117. (2014). (in Ukrainian).
5. P. F. Alcantarilla, J. Nuevo, A. Bartoli. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. British Machine Vision Conference (BMVC), (2013).
6. S. Grewenig, J. Weickert, C. Schroers, A. Bruhn. Cyclic Schemes for PDEBased Image Analysis. International Journal of Computer Vision. (2013).
7. E. Rublee, V. Rabaud, K. Konolige, G. Bradski. ORB: an efficient alternative to SIFT or SURF, Computer Vision. (ICCV), IEEE International Conference, 2564 – 2571. (2011).
8. E. Rosten, T. Drummond. Machine learning for high-speed corner detection. 9th European Conference on Computer Vision (ECCV). – P. 430 – 443. (2006).
9. X. Yang, K. T. Cheng. LDB: An ultra-fast feature for scalable augmented reality. In IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR). – P. 49 – 57. (2012).
10. M. Calonder, V. Lepetit, C. Strecha, P. Fua. BRIEF: Binary Robust Independent Elementary Features. 11th European Conference on Computer Vision (ECCV), 778– 792. (2010).
11. S. Leutenegger, M. Chli, R. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. Zurich. — P. 2548 – 2555. (2011).
12. Lytvyn, V., Vysotska, V., Peleshchak, I., Rishnyak, I., Peleshchak, R.: Time Dependence of the Output Signal Morphology for Nonlinear Oscillator Neuron Based on Van der Pol Model. In: International Journal of Intelligent Systems and Applications, 10, 8-17 (2018)
13. Vysotska, V., Hasko, R., Kuchkovskiy, V.: Process analysis in electronic content commerce system. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT 2015, 120-123 (2015)
14. Vysotska, V., Chyrun, L.: Analysis features of information resources processing. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 124-128 (2015)
15. Teslyuk, V., Beregovskyi, V., Denysyuk, P., Teslyuk, T., Lozynskyi, A.: Development and Implementation of the Technical Accident Prevention Subsystem for the Smart Home System. In: International Journal of Intelligent Systems and Applications, 10(1), 1-8 (2018)
16. Tkachenko, R., Tkachenko, P., Izonin, I., Tsymbal, Y.: Learning-based image scaling using neural-like structure of geometric transformation paradigm. In: Studies in Computational Intelligence, 730, Springer Verlag, 537–565 (2018)
17. Peleshko, D., Rak, T., Izonin, I.: Image Superresolution via Divergence Matrix and Automatic Detection of Crossover. In: International Journal of Intelligent Systems and Application, 8(12), 1-8 (2016)

18. Rashkevych, Y., Peleshko, D., Vynokurova, O., Izonin, I., Lotoshynska, N.: Single-frame image super-resolution based on singular square matrix operator. In: IEEE 1th Ukraine Conference on Electrical and Computer Engineering (UKRCON), 944-948 (2017)
19. Rusyn, B., Lutsyk, O., Lysak, O., Lukeniuk, A., Pohreliuk, L.: Lossless Image Compression in the Remote Sensing Applications In: Proc. of the IEEE First Int. Conf. on Data Stream Mining & Processing (DSMP), 195-198 (2016)
20. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: The Risk Management Modelling in Multi Project Environment.. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 32-35 (2017)
21. Chen, J., Dosyn, D., Lytvyn, V., Sachenko, A.: Smart Data Integration by Goal Driven Ontology Learning. In: Advances in Big Data. Advances in Intelligent Systems and Computing. – Springer International Publishing AG 2017. P. 283-292 (2017).
22. Su, J., Vysotska, V., Sachenko, A., Lytvyn, V., Burov, Y.: Information resources processing using linguistic analysis of textual content. In: Intelligent Data Acquisition and Advanced Computing Systems Technology and Applications, Romania, 573-578, (2017)
23. Eckel B. Philosophy Java: Programmer's Library. St.Petersburg: Peter. - 980 p. (2001).
24. Mashnin T. JavaFX 2.0. Development of RIA applications. - BHV-Petersburg, 320 p. (in Russia). (2012).
25. Bernard V. H. JDBC: Java and Databases. M .: Izd. Lori. - 324 p. (in Russia) (1999).
26. Gamma E. Methods of object-oriented design. Design Patterns. St. Petersburg: Publishing House "Peter". 366 p. (in Russia) (2007).
27. G Shildt., Holmes D. The Art of Programming on JAVA . - M .: Izd. House Williams. 336 p. (in Russia) (2005).