Development of a Reverse-search System of Similar or Identical Images

O. Veres, Ya. Kis, V. Kugivchak, I. Rishniak

Lviv Polytechnic National University, Lviv, Ukraine; e-mail: Oleh.M.Veres@lpnu.ua, rishnyakiv@gmail.com

Received February 02.2018: accepted April 05.2018

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. 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.

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. Such manipulations are performed very simply, so it is also necessary to pay attention to.

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

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. However, there is the possibility of defining images, which are widely available and used without limitation. From a large array of investigating works, a sample of identical images can be made. Images that will be repeated in most works most often are open access. Therefore, in the absence of other signs of plagiarism, such images can be skipped and not denoted as coincidences.

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 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.

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. This type of indexing is one of the simplest, but it is suitable for finding identical images where the use of borrowed graphics is not hidden.

Difference Hash. First, the size and number of colors are reduced, and the average color of the image is not taken into account. When forming a numeric signature, the value of the color of the previous pixel is taken into account, if the previous one was darker, is added to the signature 0, if on the contrary - the unit. In this way, only a local color change is taken into account without taking into account the general features of the image. This allows getting better duplicate recognition results if a non-linear color change applies to the copy. Hash, by contrast, like average hash, is well suited for defining identical images without significant changes, that is, when the authors do not use or attempt to hide copies of images.

Hash Perceptual. For this way of creating a signature, it is necessary to reduce the image to 8x8 pixels, and at least 32x32. Instead of bringing the image to a gray scale, a discrete cosine-like transformation is used, thus highlighting the characteristic features of the image. The binary value of the signature is created in the same way as the previous method. This method gives theoretically good results because it defines the features of the image and when hashing takes into account all these features (without splitting the image into fragments), which in theory should minimize the number of false positives of the algorithm.

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 $IPi < (IP + t)$, $i = 1, \dots, 4$, 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].

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 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. 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 such information image analysis system that could ideally identify identical or similar images based on a self-created image database. Therefore, the actual task is to design information, reverse image search systems. The reverse search image information system is designed to compare objects with objects in the database that are identical or similar to find information about the owners of these illustrations, or to search for their modified versions in order to prevent the use of plagiarism in user research.

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, 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].

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).

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).

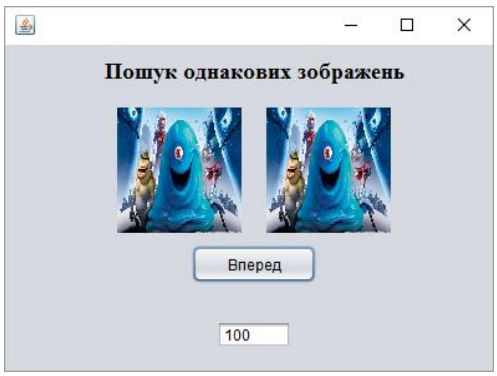


Fig. 1. The result of the processing of identical images

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. 2).

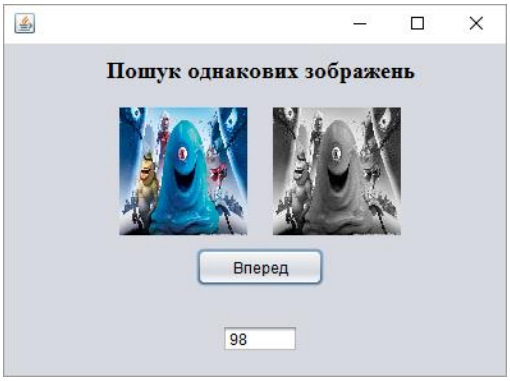


Fig. 2. The result of the processing of images with a color change

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. 3).

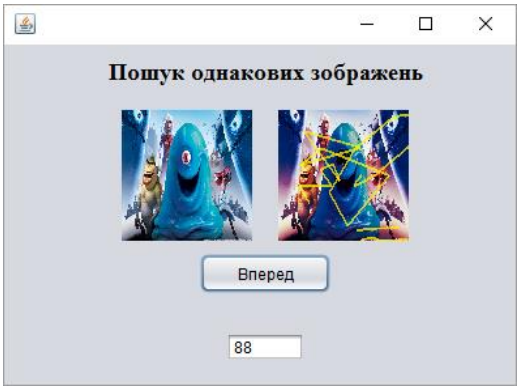


Fig. 3. The result of processing with the modification of the illustration

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. 4), 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.

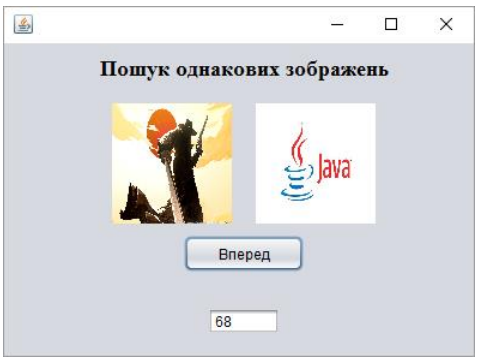


Fig. 4. The result of processing different images

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.

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. 1-3); 30 images with a monitor set (Tabl. 4-6); 30 portrait photographs of people (Tabl. 7-9).

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: $S = (T_{PK} + T_D) / PK$ (*ms*);
- 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. 1-3.

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 measuring 128x128 pixels

Method	PK	T_{PK}	T_D	S	T
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

Table 2. Estimation of methods for images measuring 256x256 pixels

Method	*PK*	T_{PK}	T_D	*S*	*T*
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

Table 3. Estimation of methods for images measuring 512x512 pixels

Method	*PK*	T_{PK}	T_D	*S*	*T*
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. 4-6).

Table 4. Estimation of methods for images measuring 128x128 pixels

Method	*PK*	T_{PK}	T_D	*S*	*T*
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

Table 5. Estimation of methods for images measuring 256x256 pixels

Method	*PK*	T_{PK}	T_D	*S*	*T*
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

Table 6. Estimation of methods for images measuring 512x512 pixels

Method	*PK*	T_{PK}	T_D	*S*	*T*
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. 7-9).

Table 7. Estimation of methods for images measuring 128x128 pixels

Method	*PK*	T_{PK}	T_D	*S*	*T*
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

Table 8. Estimation of methods for images measuring 256x256 pixels

Method	*PK*	T_{PK}	T_D	*S*	*T*
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

Table 9. Estimation of methods for images measuring 512x512 pixels

Method	*PK*	T_{PK}	T_D	*S*	*T*
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

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.

RESEARCH OF ALGORITHMS FOR REVEALING THE SIMILARITY OF DIFFERENT IMAGES

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. 5, 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.

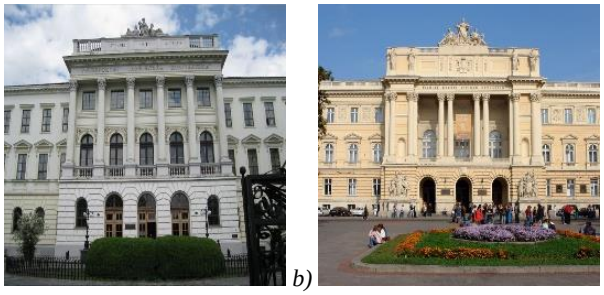
Test 2 – to determine the similarity of images as a resource, two completely identical images were taken (Fig. 5, *a* and *b*), one of which was rotated 90° (Fig. 5, *b*), after which for each of them the key points were determined and a descriptor is created to remove these points and compare them to each other for identity.

a) *b)* *c)* *d)*

Fig. 5. Same pictures: *a*) original; *b*) reversed by 90 degrees; *c*) changed perspective; *d*) turned 45 degrees

Similar images. Test 1 – two similar images (Fig. 5, *a* and *c*), one of which was photographed from a different angle (the main object of the study - the book, is closer) (Fig. 5, *c*). Test 2 – two similar images, one of which was photographed from another angle (Fig. 5, *c*) and the main object of the study is at an angle of 45 degrees (Fig. 5, *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. 6, *a*), and the other - "Ivan Franko Lviv National University" (Fig. 6, *b*).



a) *b)*

Fig. 6. 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 10-13. 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 10. Results of the study by the ORB method

Key points	Test 1		Test 2	
	Fig 1	Fig 2	Fig 1	Fig 2
Same pictures				
PK	416	416	416	415
PK_S	416		406	
K_S (%)	100		98	
Similar images				
PK	416	416	416	415
PK_S	416		406	
K_S (%)	100		98	
Different images				
PK	416	416	416	415
PK_S	416		406	
K_S (%)	100		98	

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.

Table 11. Results of the study by the AKAZE method

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

Table 12. Results of the study by the BRISK method

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

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.

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 13. Results of the study by the FAST method

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

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 [13-18]: 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 [19-24]. 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.

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.

In the second study, attention was focused on which of the algorithms has the least number of errors in operation. 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. **L. Shapiro, G. Stockman. 2006.** Computer vision. Washington University. – 752 p.
2. **A. Biloshchyts′kyy, O. Dikhtyarenko**. **2013.** The effectiveness of methods for finding matches in texts. Managing the development of complex systems. — № 14. – P. 144 – 147. (in Ukrainian).
3. **N. S. Shozda. 2002.** Searching for textures in large databases. Informatics, Cybernetics and Computing. Donetsk. Ukraine. n. 39, - P. 182 - 187. (in Ukrainian).
4. **A. Biloshchyts′kyy, O. Dikhtyarenko**. **2014.** 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. (in Ukrainian).
5. **P. F. Alcantarilla, J. Nuevo, A. Bartoli**. **2013.** Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. British Machine Vision Conference (BMVC),
6. **S. Grewenig, J. Weickert, C. Schroers, A. Bruhn**. **2013.** Cyclic Schemes for PDEBased Image Analysis. International Journal of Computer Vision.
7. **E. Rublee, V. Rabaud, K. Konolige, G. Bradski. 2011.** ORB: an efficient alternative to SIFT or SURF, Computer Vision. (ICCV), IEEE International Conference. – P. 2564 – 2571.
8. **E. Rosten, T. Drummond**. **2006.** Machine learning for high-speed corner detection. 9th European Conference on Computer Vision (ECCV). – P. 430 – 443.
9. **X. Yang, K. T. Cheng**. **2012.** LDB: An ultra-fast feature for scalable augmented reality. In IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR). – P. 49 – 57.
10. **M. Calonder, V. Lepetit, C. Strecha, P. Fua. 2010.** BRIEF: Binary Robust Independent Elementary Features. 11th European Conference on Computer Vision (ECCV). – P. 778 – 792.
11. **S. Leutenegger, M. Chli, R. Siegwart. 2011.** BRISK: Binary Robust Invariant Scalable Keypoints. Zurich. — P. 2548 – 2555.
12. **V.V. Litvin, N. B. Shakhovska. 2000.** Designing Information Systems: Teaching. Manual. Lviv: Novyy svit. 380 p. (in Ukrainian).
13. **Eckel B. 2001.** Philosophy Java: Programmer's Library. St.Petersburg: Peter. - 980 p. (in Russia).
14. **Mashnin T. 2012.** JavaFX 2.0. Development of RIA applications. - BHV-Petersburg. - 320 p. (in Russia).
15. **M. Davis, J. Phillips. 2008.** Studying PHP and MySQL. - M .: Symbol-Plus. - 442 p. (in Russia)
16. **Bernard V. H. 1999.** JDBC: Java and Databases. M .: Izd. Lori. - 324 p. (in Russia)
17. **Gamma E. 2007.** Methods of object-oriented design. Design Patterns. St. Petersburg: Publishing House "Peter". 366 p. (in Russia)
18. **G. Shildt, D. Holmes**. **2005.** The Art of Programming on JAVA . - M .: Izd. House Williams. 336 p. (in Russia)
19. **Shakhovska N.B. 2011.** Formal presentation of data space as an algebraic system. System Research and Information Technologies / National Academy of Sciences of Ukraine, Institute for Applied Systems Analysis. – Kyiv, 2011. – № 2. – C.128 – 140. (in Ukrainian).
20. **Shakhovska N.B., Bolubash Yu.Ja., Veres O.M. 2015.** Big Data Federated Repository Model // The Experience of Designing and Application of CAD Systems in Microelectronics (CADMS'2015) Proc. of the XIII-th Int. Conf., (Polyana-Svalyava (Zakarpattya), Ukraine, 24-27 February, 2015). – Lviv: Publishing Lviv Polytechnic, 2015.– P. 382-384. (in Ukrainian).
21. **N. Shakhovska, O. Veres, Y. Bolubash, L. Bychkovska-Lipinska. 2015.** Data space architecture for Big Data managering. Xth International Scientific and Technical Conference "Computer Sciences and Information Technologies" (CSIT'2015). pp. 184-187, Lviv, 2015. DOI: 10.1109/STC-CSIT.2015.7325461
22. **O. Veres, N. Shakhovska. 2015.** Elements of the formal model big date. XI International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH'2015), pp. 81-83, Lviv, 2015. (in Ukrainian).
23. **N. Shakhovska, O. Veres, M. Hirnyak. 2016.** Generalized formal model of Big Data. ECONTECHMOD: an international quarterly journal on economics of technology and modelling processes. Vol. 5, no. 2, pp. 33–38, 2016.
24. **N. Shakhovska, O. Veres, Y. Bolubash. 2015.** Big Data Model "Entity and Features". ECONTECHMOD: an international quarterly journal on economics of technology and modelling processes. Vol. 4, no. 2, pp. 51–58, 2015. .