ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 6, Num. 2, 2021

RESEARCH OF CONTENT-BASED IMAGE RETRIEVAL ALGORITHMS

Eduard Yakubchyk, Iryna Yurchak

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Author's e-mail*: eduard.yakubchyk.mki.2020@lpnu.ua*

Submitted on 13.05.2021

***Abstract*: Finding similar images on a visual sample is a difficult AI task, to solve which many works are devoted. The problem is to determine the essential properties of images of low and higher semantic level. Based on them, a vector of features is built, which will be used in the future to compare pairs of images. Each pair always includes an image from the collection and a sample image that the user is looking for. The result of the comparison is a quantity called the visual relativity of the images. Image properties are called features and are evaluated by calculation algorithms. Image features can be divided into low-level and high-level. Low-level features include basic colors, textures, shapes, significant elements of the whole image. These features are used as part of more complex recognition tasks. The main progress is in the definition of high-level features, which is associated with understanding the content of images.**

In this paper, research of modern algorithms is done for finding similar images in large multimedia databases. The main problems of determining high-level image features, algorithms of overcoming them and application of effective algorithms are described. The algorithms used to quickly determine the semantic content and improve the search accuracy of similar images are presented.

The aim: The purpose of work is to conduct comparative analysis of modern image retrieval algorithms and retrieve its weakness and strength.

***Index Terms*: image recognition, feature search, search algorithm, content-based image retrieval, text-based image retrieval.**

I. INTRODUCTION

The rapid development of multimedia technologies, the preservation of high quality images, the improvement of storage technologies contributes to the rapid growth of a large collection of images. This is primarily due to the widespread use of the Internet and portable devices to download digital images [1]. The development of many image retrieval systems requires effective search and browsing tools. Researchers are developing for new algorithms that can search for similar images in huge collections. Content-Based Image Retrieval (CBIR) systems are a popular trend, as traditional Text-Based Image Retrieval (TBIR) cannot satisfy modern users. CBIR has become a subject of wide interest and a source of fast and accurate search [2].

The last decade has seen the emergence of numerous works to Content-Based Image Retrieval. [3]

There are three key issues in Content-Based Image Retrieval: image representation, image organization, and image similarity measurement. Existing algorithms can be classified based on their impact on these key elements. An internal problem with content-based visual search is image comparison. Usually images are presented as one or more visual features [4]. The presentation is expected to be descriptive and discriminatory in order to distinguish between similar and dissimilar images. But there are always difficulties with the effect of the background and possible changes, such as translation, rotation, resizing, changing lighting, and so on [5].

Content-Based Image Retrieval is usually based on comparing low-level features, such as color, texture, or shape, that are automatically extracted from the images themselves [6].

Ideally, the similarity between images should reflect relevance in semantics, which is difficult to implement due to the problem of "semantic gap" in understanding the content of the image. Typically, the similarity of images when searching based on content is formulated based on the results of matching visual features with some weighing schemes. In addition, the formulation of image similarity in existing algorithms can also be considered as different cores of correspondence [7].

To solve the problem of determining the semantic features of the highest level today offer the use of modern approaches, the use of neural networks, genetic and natural algorithms [8]. The results presented in scientific works are embodied in practical implementations of systems of semantic search of similar images in huge multimedia bases.

The aim of this work is to research modern algorithms for finding similar image in multimedia databases. To do this, use combinations of Text Based Image Retrievals (TBIR) and Content-based image retrieval (CBIR) [9]. Each algorithm can be implemented by different algorithms. The choice of the appropriate algorithm provides higher search accuracy. Many algorithms can be used to determine low-level image features, SIFT and PCA-SIFT algorithms are selected for the research [10].

To find high-level functions, it is advisable to choose flexible natural algorithms. They are fast, resistant to noisy data and provide good results for multi-parameter tasks. The main representative of the group of evolutionary algorithms are genetic algorithms. Its combination with algorithms for detecting low-level functions will provide a fairly accurate and fast image search.

The purpose of this work is to conduct comparative analysis of modern image retrieval algorithms in multimedia databases. Compare ORB, BRISK, AKAZE and FAST algorithms to find their advantages and disadvantages.

II. APPROACHES THAT USED FOR IMAGE RETRIEVAL

According to the search principle, all algorithms can be classified as follows:

- search by text attributes is only used for keywords that are used to search for notes in the image storage. Such systems use keywords to get and sort results based on matching. The logic can be configured to specify the degree of compliance (partial or exact);
- category search is used to access images categorized to facilitate quick search in storage based on categories that actually define groups for images in a large database;
- function search is used for images with letters, objects, shapes, and key points. The search operation is performed using this metadata, which allows us to restrict the search in the image storage;
- example search is used when passing the request image as input. It uses the request image to recognize objects/texts/objects. It also searches for similar images in the image store.

Multiclass image classification is one of the most popular image annotation algorithms that uses a huge vocabulary. Typically, annotation systems use machine learning techniques that generate keywords for images in the image repository.

In text models, text search works with text in the query and image storage. Logic is configured to determine the weight of each tag, which makes it easier to select specific images that can be mapped. In such systems, the word package algorithm is common. As it is shown in Fig. 1, images with their tags are displayed in the image store. This package of words allows the system to assign weights to different tags. Weight indicators determine the ability to select an image based on its weight.

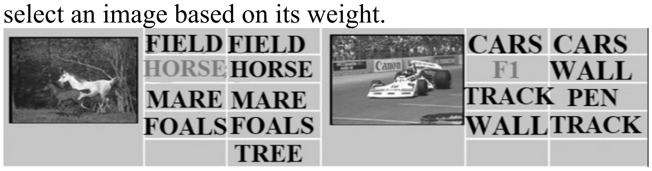

Fig. 1. Tagged images

Model Search is the search for a specific image from the image store using a model that organizes image tags. Using this model, we can generate weights and assign them to get images.

One such implementation is the Vector Space model, which uses an algebraic model to represent tags associated with images as vectors of identifiers. It is usually used for tag filtering, tag search, indexing, and relevance ranking.

An example of this approach is the calculation of the statistical indicator TF-IDF (term frequency-inverse document frequency – frequency of terms-inverse frequency of documents).

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

where n_{id} is the number of appearances of the corresponding tag in descriptions,

n_d -number of words,

N – number of images,

n_i – the number of images that contain the corresponding tag.

Images have corresponding tags as documents, and they are collected from the image store. We process them for a specific category of images, and then use the Vector Space model to assign them specific weights. A time-weighted document is used to filter out unimportant tags from these images in the image store and provides a better search experience.

Among the algorithms based on second-order derivatives, The Laplace operator is distinguished. This operator finds the limits at the places where the sign of the derivative of the brightness function changes. But Laplacian's cameraman is very sensitive to noise. In addition, its use leads to doubling of contours, which gives an undesirable effect and complicates segmentation. Therefore, Laplacian is often used in combination with smoothing, for example, using the Gaussian algorithm. Such combinations are called LaPlace Gaussian.

The filter mask is calculated using the formula

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right) \cdot e^{{x^2+y^2}/{2\sigma^2}}$$

Gaussian difference is a well-known feature enhancement algorithm that involves subtracting one blurry version of the original image from another, less blurry version of the original (Fig. 2). Blurry images are obtained by convolution of grayscale images with Gaussian nuclei with different standard deviations. In other words, the Gaussian difference is a bandpass filter that allows you to discard a large number of spatial frequencies that are present in the original image.

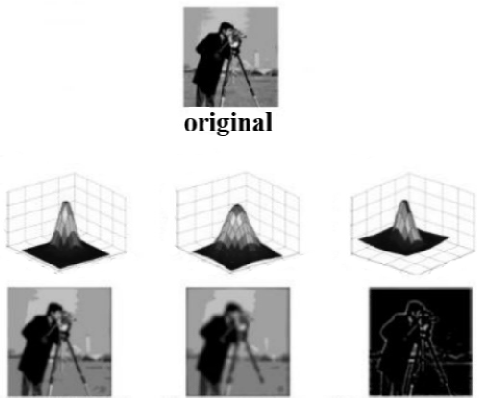

Fig. 2. Gaussian difference (SIFT)

Another approach is to use Euclidean distance (the distance between 2 points) [1]. This algorithm works on the basic principles of geometry, which allow you to map pixels to pixels. The algorithm compares two images by matching the distances of key points between them.

$$d(x,y_i)=\|x-y_i\|^2=\sum_{j=1}^{k}(x_j-y_{ij})^2$$

Where $x=(x_1,x_2,...,x_k)$ is the input vector,

$y_i=(y_{1i},y_{2i},...,y_{ki})$ - code word.

The SURF algorithm is one of the content-based image comparison algorithms available today, which performs several operations on data to generate key points and compare points with each other to compare images.

SURF uses 3 steps:

- identifying key points
- description of key points
- matching key points

Key point detection is the process of selecting points in an image that are considered "good" in terms of image quality. Previous research on content - based image search techniques, such as SIFT, has identified key points with "good" features, and the key aspect is SURF, which returns high-quality "good" features.

The key point description deals with removing descriptors for key points that encode properties of functions, such as contrast with neighboring ones. Key point mapping works by comparing points in both images, and it will find the best points that match the points in the image. For this purpose, use the algorithms of searching for the nearest neighbors.

SURF processes recognize key points in the image and process its edges where the intensity of points changes. Points are classified to work on critical points related to images.

III. TEXT-BASED IMAGE SEARCH

There are two approaches that allow users to find images. First, it is text-based image search (TBIR); second, algorithms based on image context analysis (CBIR) [2].

TBIR search is based on the assumption that the surrounding text describes the image. It is believed that text surrounding the image, such as file names, captions, and "alt" tags in HTML, as well as paragraphs close to the image with possible corresponding text, provide key information about what exactly is shown in the image. In other words, in fact, Image Search is based on metadata that is considered relevant and relevant to reality. This approach has the following disadvantages:

- in TBIR, people are required to personally describe each image in the database, i.e. if there are a large number of images, this technique requires too much manual effort
- TBIR algorithms require a sufficiently large amount of metadata so that the search result is relevant and the output results do not contain too many records.
- the description of the content of an image is a subjective perception of a person, that is, different people can create different descriptions of the content of the same image.
- queries are performed mainly based on text information, and therefore execution strongly depends on the degree of correspondence between images and their text description.

To overcome these disadvantages, image comparison-based search algorithms are used, which can be divided into two types: accurate image search and approximate image search. Accurate image search can be called image recognition. It requires the images to match accurately (100%). An approximate search for suitable images is based on the content of the image. To solve this problem, many algorithms have been developed based on various statistical parameters of images. The purpose of these algorithms is to obtain more accurate image similarity results with high search performance [2, 3].

IV. CONTENT-BASED IMAGE RETRIEVAL

Image search algorithms combine both the visual features of the images you are looking for, which relate to more detailed aspects of perception, and the high-level semantic features that underlie the more general conceptual aspects of visual data.

Image search can be classified into the following types:

- getting an exact match: elements that perfectly match the request of a user who wants to identify a significant commonality of properties of two entities;
- low-level similarity search: low-level visual features such as color shape, texture, etc. are used;

- example search – an image is sent to the system input, and the system returns images that have functions similar to image properties. Image similarity is determined by values or similarity metrics that are specifically defined for each feature according to their physical meaning;
- high-level semantic search: the concept of similarity is not based on simple feature mapping, but is usually based on extended user interaction with the system. Such indexing algorithms provide descriptions using a fixed dictionary or high-level functions, also called semantic concepts [3, 4].

In general, the image search process is shown in (Fig. 3)

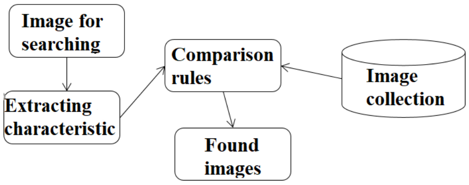

Fig. 3. General image search process

Image search algorithms are most often based on the following image functions [4]:

- color function. This algorithm does not search for exact color matches in images, but finds images with the corresponding pixel color information. This approach has proven very successful in image search, as there are simple concepts for measuring similarity based on color. And algorithms based on them are very easy to implement. In addition, this feature is resistant to noise and image rotation options. However, this function can only be used for global use, since global characteristics are taken into account, not local features in the image. For example, it is often difficult to determine the similarity between images of the same scene, but shot at different times and under different conditions [4, 5].
- Form Function. Natural objects are primarily recognized by their shape. For each object identified in each saved image, a number of features specific to the object's shape are calculated. In general, shape representations can be divided into two categories: border-based and Area-based. The first one uses only the outer borders of the shape, and the second one uses the entire area of the shape.

Form-based image search algorithms take the input image provided by the user and output a set of (possibly ranked) system database images, each of which must contain query-like forms. There are two main types of possible queries: example queries and sketch queries. When searching based on a shape, it is quite difficult to analyze isolated objects, because to compare them with the query, they must first be localized in the image. Shape localization is a non-trivial problem, since it involves solving the problem of separating certain objects from the background. The second problem is the need to deal with an inaccurate match between a stylized sketch and a real image. It is possible that the detailed form contained in the image will need to take into account possible differences between these two forms when comparing them [4, 5].

Texture function. Texture is an important characteristic in many types of images. Despite its importance, there is no official definition of texture. If an image has a wide variety of tonal primitives, the dominant property of that image is texture. Texture is a spatial relationship that manifests itself in gray levels in a digital image. Spatial relations between pixels, spatial indicators related to indicators mainly obtained from Spatial Statistics and used mainly in geospatial applications to characterize and quantify spatial models and processes [5].

A useful approach to texture analysis is based on a histogram of the intensity of the entire image or part of it. Common features of a histogram include: moments, entropy variance, mean (an estimate of the average intensity level), variance (the second point is a measure of the variance of the intensity of a region), square mean or average energy, skew (the third point indicates histogram symmetry), and kurtosis (cluster severity).

One of the easiest ways to get statistical characteristics of an image is to use the probability distribution of the amplitude of a quantized image, which can be determined in this way.

$$P(b) = P_R^{(F(j,k)=r_b)}$$

where r_b determines the level of quantized amplitude for 0, b, and L-1. the First-Order histogram simply estimates P (b)

$$P(b) = \frac{N(b)}{M}$$

where M is the total number of pixels in an adjacent window of a certain size centered approximately (j, k),

b-Gray level in the image,

N (b) is the number of pixels of the r_b amplitude in the same window.

Performance detection algorithms consist of two main categories:

- feature-based algorithms, such as a color histogram and a shape or border detector.
- texture-based algorithms, such as scale invariant function transformation (SIFT), reliability function acceleration (SURF) and analysis of the main components of PCA-SIFT.

Let's look at the main characteristics of these algorithms.

Algorithms based on color histogram functions are based on determining the signature for each image based on its pixel values and image comparison rules. However, only the color signature is used [5].

Existing general-purpose Color Image search engines roughly fall into three categories depending on the signature creation approach, namely histograms, color placement, and Area-based search. Histogram-based search algorithms are studied in two different color spaces. A color space is defined as a model for representing a color in terms of intensity values. As a rule, the color space defines a one-to four-dimensional space. Three-dimensional color spaces such as RGB (Red, Green, and blue) and HSV (hue, saturation, and value) are explored.

The disadvantage of this algorithm is that information about the location, shape, and texture of the object is discarded. They also use color histogram options with rotation, zoom, light changes, and image noise without human perception.

The features of the accelerated segment Test (FAST) algorithm are based on the Harris angle detector, which aims to introduce a new algorithm for detecting and determining specific points or angles. The Harris angle detector is a popular special point detector due to its stability in relation to rotation, scale, and image noise using the autocorrelation function [5].

When developing this algorithm, an algorithm was developed to detect reliable features in any image that meet the basic stability requirements. But this algorithm only detected angles, and there were no special point connections, which is the main limitation for obtaining basic level descriptors (for example, surfaces and objects).

The algorithm aims to identify distinctive invariant features of images that can later be used to reliably match different types of objects or scenes. This definition uses two key concepts: distinctive invariant features and reliable correspondence [6, 7].

SIFT is divided into four main computational stages:

- detection of extreme scales in scale space: the first stage of calculation performs a search at all scales and locations in the image. It is effectively implemented by using the Gaussian difference function to determine potential points of interest that are scale-and orientation-invariant.
- localization of key points: this step attempts to remove points from the list of key points by searching for those that have low contrast or are poorly localized at the border.
- orientation assignment: one or more orientations are assigned to each key point location based on the directions of the local image gradient. All future operations are performed on these images that have been transformed relative to the assigned orientation, scale, and location for each function, thus ensuring that these transformations are invariant.
- key point descriptor: local image gradients are measured at the selected scale in the area around each key point. They turn into a representation that allows you to achieve significant levels of local shape distortion and light changes.

In the SIFT algorithm, "there is no need to analyze the entire image", but only interesting key points can be used to describe the image. Unfortunately, the disadvantage of the algorithm is that SIFT considers it the slowest texture-based algorithm, difficult to calculate, and consumes a lot of resources.

Principal Component Analysis (PCA-SIFT algorithm) is a large – scale invariant transformation of functions. PCA is a standard dimensionality reduction technique and is applicable to a wide range of computer vision problems, including function selection and object recognition. Although this algorithm suffers from a number of disadvantages, such as implicit assumption of Gaussian distributions, limited to orthogonal linear combinations, it remains popular due to its simplicity. The idea of applying PCA to parts of images is not new [8].

PCA is well suited for correcting key points (after they have been converted to Canonical scale, position, and orientation). This view significantly improves the efficiency of Sift matching. PCA-SIFT is significantly more accurate and much faster than the standard local sift descriptor.

The main representative of the group of evolutionary algorithms is genetic algorithms. Their combination with standard algorithms based on characteristic detection will provide a fairly accurate and fast image search [9, 10].

The general structure of an image search system based on a genetic search optimization algorithm is shown in (Fig. 4)

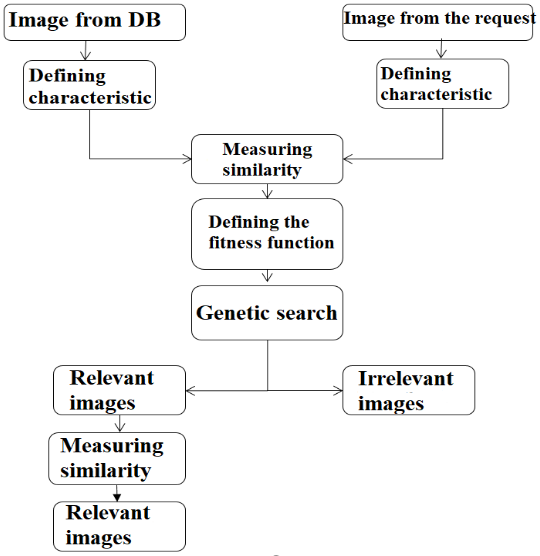

Fig. 4. Generalized image search scheme using genetic algorithms

To find similar images, use the algorithm of selecting key points. A key point, or point feature of an image, is a point whose location stands out against any other point. As a feature of the image point for most modern algorithms take a square window that is 5 by 5 pixels in size. The definition of these points in the image is achieved by using a detector and a descriptor. The detector is a algorithm of determining the key point that highlights it against the background of the image, and descriptors must ensure the invariance of the correspondence between the key points in terms of image transformations. A descriptor is a algorithm that allows you to delete the key points of both images and compare them with each other. In the case of modifications to the study objects, the detector helps to find the same key points on both objects.

The main algorithms used in the construction of detectors and descriptors: FAST (Features from Accelerated Segment Test), SIFT (Scale Invariant Feature Transform), ORB (Oriented FAST and Rotated BRIEF), AKAZE (Accelerated KAZE), BRIEF (Binary Robust Independent Elementary Features), BRISK (Binary Robust Invariant Scalable Keypoints).

In order to find similar images, we will perform a comparative analysis of algorithms that work with key points, namely: ORB, BRISK, AKAZE, FAST, respectively, based on the results of the classifier. The size of the input images is considered in compressed form to 128, 256 and 512 pixels on each side. Input images are divided into three groups: 30 images with a large number of details (Table 1); 30 images with a monitor image (Table 2).

Table 1

Dimensionality incoming images, pixels	**Algorithm**	**General number found key points**	**General amount of time spent on search key points, ms**	**Work time descriptor, ms**
128x128	ORB	10444	247	5199
128x128	BRISK	11768	12496	12533
128x128	AKAZE	5041	972	11128
128x128	FAST	6568	144	4141
256x256	ORB	12311	429	6129
256x256	BRISK	26767	13096	12577
256x256	AKAZE	7286	1872	1930
256x256	FAST	15568	396	5643
512x512	ORB	15719	602	7626
512x512	BRISK	78395	14087	12683
512x512	AKAZE	8688	2777	3541
512x512	FAST	32210	801	8111

According to the (Table. 1) accounting 512 × 512 dimension images we can say that ORB is the fastest algorithm in searching key points, we can also say that ORB algorithm is 23.4 times faster than BRISK algorithm by dividing search time of BRISK algorithm to search time of ORB algorithm – 14087 / 602 = 23.4 times.

All images in this group contain numerous details located in different places. Information on algorithm estimates for different extensions of illustrations (Table. 1). The largest number of key points is found using the BRISK algorithm, this number increases exponentially.

Accordingly, if the resolution of the subject increases image, it takes much longer to process. The ORB algorithm was not very sensitive to changing the image size within the selected limits, its complexity increases in arithmetic progression. The shortest execution time of the descriptor in the AKAZE algorithm. The FAST algorithm spends the least time on a general search for similar images.

Let's take 30 illustrations of the monitor image, each of which will present images in different windows of different programs. Let's analyze this group for different extensions of illustrations (Table. 2).

Table 2

Dimensionality incoming images, pixels	**Algorithm**	**General number found key points**	**General amount of time spent on search key points, ms**	**Work time descriptor, ms**
128x128	ORB	1409	27	422
128x128	BRISK	2178	2917	3014
128x128	AKAZE	995	202	316
128x128	FAST	1024	44	623
256x256	ORB	1661	47	497
256x256	BRISK	4954	3057	3025
256x256	AKAZE	1438	389	541
256x256	FAST	2427	121	849
512x512	ORB	2121	66	619
512x512	BRISK	14509	3288	3050
512x512	AKAZE	1715	577	992
512x512	FAST	5022	245	1220

The number of key points in the sum of all images decreased significantly compared to the first group. This affected the running time of the program, the descriptor and the costs. Accordingly, the fewer key points generated by any algorithm, the less time it spends on their processing. All time costs are proportional to the number of key points. The results of the algorithms are almost no different from the previous group, which indicates that their work does not depend on the input data.

The ORB algorithm performed well in all tests, as the percentage of common key points decreases accordingly for less similar images. The AKAZE algorithm shows results at the ORB level, but the number of key points generated by it is much smaller and uneven, so we can say that the algorithm is stable in the results, but unpredictable in terms of the number of created main image points. BRISK algorithm – this algorithm also performed its task, but showed worse results in finding similar and identical images, but was able to clearly distinguish different illustrations in the tests. The FAST algorithm is one of the leaders in the speed of detecting key points and calculating descriptor values for them, but failed the tests, and although the number of its key points is much higher than its predecessors, it did not allow it to recognize identical images rotated 90 degrees, and similar images when rotated 45 degrees.

V. CONCLUSIONS

The paper presents a comparative analysis of modern image retrieval algorithms in multimedia databases. Today, a popular trend is to combine search algorithm: Text Based Image Retrievals (TBIR) and Content-based image retrieval (CBIR). These algorithms complement the results and the search accuracy increases.

Content-based image retrieval algorithms are aimed at determining the essential properties of images of low and higher semantic level. Based on them, a vector of features is built, which will be used in the future to compare pairs of images. Each pair always includes an image from the collection and a sample image that the user is looking for. The result of the comparison is a quantity called the visual relevance of the images. Image properties are called features and are evaluated by calculation algorithms.

Algorithms of image recognition based on low-level features (color, texture and shape) are analyzed. These are well-designed algorithms that give good results. These algorithms are used for image pre-processing.

The main element of this study was the time spent finding key points and comparing them to similarity algorithms: ORB, BRISK, AKAZE and FAST.

The BRISK algorithm turned out to be the worst, because the number of points generated by it is very large, which led to a rapid increase in processing time. It has been experimentally found that the image size of 256 × 256 pixels is the most optimal for its processing.

The second study focused on determining which of the algorithms had the fewest errors. To do this, groups of identical images, similar images and completely different images were created. The FAST algorithm did not cope with this task, so, despite its best results in image processing, this algorithm cannot be used. The best test results for all indicators in the algorithms ORB and AKAZE.

We can conclude that ORB algorithm takes the least time spent on searching key points, in comparison with other algorithms it is 23.4 times faster than the slowest BRISK algorithm, but BRISK algorithm can find the greater number of key points. So in order to find greater number of key points it is suggested to use BRISK algorithm, but is speed is more important then ORB would be better choice.

References

[1] Ibtihaal M. Hameed, Sadiq H. Abdulhussain & Basheera M. Mahmmod | D T Pham (Reviewing editor) (2021) Content-based image retrieval: A review of recent trends, Cogent Engineering, 8:1, DOI: 10.1080/23311916.2021.1927469.

[2] Narendra Kumar Rout, Mithilesh Atulkar and Mitul Kumar Ahirwal. A review on content-based image retrieval system: present trends and future challenges. Published Online:August 19, 2021pp., 461-485, Available at: https://www.inderscienceonline.com/doi/abs/10.1504/IJCVR.2021.117578, (Accessed: 18 November 2021).

[3] Alsmadi, M.K. Content-Based Image Retrieval Using Color, Shape and Texture Descriptors and Features. Arab J Sci Eng 45, 3317–3330 (2020), https://doi.org/10.1007/s13369-020-04384-y.

[4] Zenggang, X., Zhiwen, T., Xiaowen, C. et al. Research on Image Retrieval Algorithm Based on Combination of Color and Shape Features. J Sign Process Syst 93, 139–146 (2021), https://doi.org/10.1007/s11265-019-01508-y.

[5] Juan Luo, Oubong Gwun. "A comparison of sift, pca-sift and surf." International Journal of Image Processing (IJIP) (2009): 143-152, Available at: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.7041&rep=rep1&type=pdf, (Accessed: 18 November 2021).

[6] Ghrabat, M.J.J., Ma, G., Maolood, I.Y. et al. An effective image retrieval based on optimized genetic algorithm utilized a novel SVM-based convolutional neural network classifier. Hum. Cent. Comput. Inf. Sci. 9, 31 (2019), https://doi.org/10.1186/s13673-019-0191-8.

[7] William La Cava, Sara Silva, Kourosh Danai, Lee Spector, Leonardo Vanneschi, Jason H. Moore. Multidimensional genetic programming for multiclass classification, Swarm and Evolutionary Computation, Volume 44, 2019, Pages 260-272, ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2018.03.015.

[8] Haseena Sikkandar , Revathi Thiyagarajan. Soft biometrics-based face image retrieval using improved grey wolf optimization ISSN 1751-9659 Received on 7th March 2019 Revised 11th September 2019 Accepted on 14th October 2019 E-First on 27th January 2020, doi: 10.1049/iet-ipr.2019.0271, www.ietdl.org.

[9] Xushan Peng, Xiaoming Zhang, Yongping Li, Bangquan Liu, Research on image feature extraction and retrieval algorithms based on convolutional neural network, Journal of Visual Communication and Image Representation, Volume 69, 2020, 102705, ISSN 1047-3203, https://doi.org/10.1016/j.jvcir.2019.102705.

[10] F. Luo, L. Zhang, X. Zhou, T. Guo, Y. Cheng and T. Yin, "Sparse-Adaptive Hypergraph Discriminant Analysis for Hyperspectral Image Classification," in IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 6, pp. 1082-1086, June 2020, doi: 10.1109/LGRS.2019.2936652.

Eduard Yakubchyk Received a bachelor's degree in Computer Engineering in 2020. Since 2020 he has been studying for a Master's degree. Now he is a second-year computer engineering student at Lviv Polytechnic National University. His research interests include image processing, analyzing and searching, operating systems.

Iryna Yurchak received the Ph.D. degree in automated control systems and advanced information technologies from State Research Institute of Information Infrastructure, Lviv, in 1999.

Her research interests: artificial intelligence systems, intelligent computing systems, neural networks, genetic algorithms, fuzzy logic, recognition systems, prediction problems, computer graphics, computer modeling and animation, web design.