

Valerii Filatov<sup>1</sup>, Anna Filatova<sup>1</sup>, Anatolii Povoroznyuk<sup>1</sup>, Shakhin Omarov<sup>2</sup>

<sup>1</sup>National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine

<sup>2</sup>Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

## IMAGE CLASSIFIER FOR FAST SEARCH IN LARGE DATABASES

**Abstract. Relevance.** The avalanche-like growth in the amount of information on the Internet necessitates the development of effective methods for quickly processing such information in information systems. Clustering of news information is carried out by taking into account both the morphological analysis of texts and graphic content. Thus, an urgent task is the clustering of images accompanying textual information on various web resources, including news portals. **The subject of study** is an image classifier that exhibits low sensitivity to increased information in databases. **The purpose of the article** is to enhance the efficiency of searching for identical images in databases experiencing a daily influx of 10-12 thousand images, by developing an image classifier. **Methods used:** mathematical modeling, content-based image retrieval, two-dimensional discrete cosine transform, image processing methods, decision-making methods. **The following results were obtained.** An image classifier has been developed with low sensitivity to increased database information. The properties of the developed classifier have been analyzed. The experiments demonstrated that clustering information based on images using the developed classifier proved to be sufficiently fast and cost-effective in terms of information volumes and computational power requirements.

**Keywords:** information systems; content-based image retrieval; image classifier; large databases; two-dimensional discrete cosine transform.

### Introduction

Currently, much attention is paid to the collection and processing of information. A significant amount of information is contained in news feeds of electronic media. One of the ways to process such information is to cluster it according to its semantic load [1]. Typically, clustering of news information is carried out taking into account the morphological analysis of texts. It should be taken into account that in news feeds of electronic media, almost every news is accompanied by graphic content (photo materials), so it becomes possible to combine different news articles into clusters not only according to their content but also according to graphic accompanying information (the so-called “visual feature”). Image clustering plays the role of an additional feature in the text clustering algorithm, which sometimes plays an even more significant role than content clustering using keywords or other information features (cities, people, sports, or other characteristics that can be identified in the text and lead to the original word form).

It is important to take into account that the text can contain more than one image, which has both a negative and a positive effect. A negative effect may include the insertion of images that have little relevance to the analyzed texts (for example, company logos). The positive effect is that if the images correctly reflect the text, complex cross-clustering rules can arise that will allow clusters to be combined into digest groups containing the most complete information about the content (for example, an event or incident).

With such clustering, the following hypothesis can be formulated: the more unique the image, the higher the likelihood of “similarity” of the text content. In this case, headings, keywords, and other text information can be used as additional features for clustering.

Thus, the task of clustering images accompanying text information on various sites, including news feeds of electronic media, is relevant.

**Literature analysis.** Today there are many systems for image recognition. The most popular them are

TinEye, Google Similar Images, and AntiDupl.NET. A common disadvantage of these systems is the inability to load an image gallery and create your database (DB) for work [2]. At the same time, there are a large number of image search methods, which differ in varying complexity and efficiency.

In works [3, 4], the authors proposed an invariant model and a method for quickly searching for digital images in data warehouses. However, the proposed model [3] has several disadvantages: the model is considered only for halftone images; The model only takes into account the shape of the histogram, and it has not been proven that different halftone images cannot have histograms of the same shape. The authors also indicated that on average it takes 1.7 seconds to search for one image in a database that contains more than 100 thousand images [4]. This speed is unacceptable in an information system in which about 10-12 thousand images need to be classified per day. This classification may take more than 5 hours.

One of the modern approaches is content-based image retrieval (CBIR), which plays an important role in finding images similar to the query image by extracting visual features [5, 6]. This approach is based on converting images into low-level functions that describe the images being analyzed.

Fig. 1 shows a generalized block diagram of CBIR systems, which includes three stages: feature extraction from the query image, feature selection, and then similarity matching [7, 8]. The CBIR system generates multidimensional feature vectors that are compared with image vectors in the database. According to the method of information processing, the CBIR system is divided into online and offline subsystems, which have the same feature extraction block (Fig. 1, [7]). To compare vectors, the corresponding similarity measures or distance measures are used: Euclidean distance [9, 10], Manhattan distance (city block metric) [11, 12], cosine similarity [13, 14], Mahalanobis distance [15], Hamming distance [16] and others. The resulting similarity score is compared with a threshold value that is pre-determined by the CBIR system.

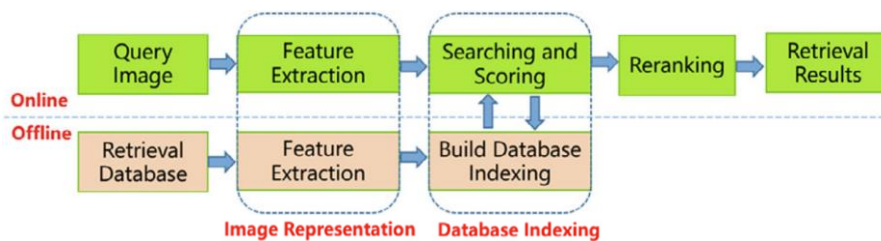


Fig. 1. Generalized block diagram of CBIR systems [7]

As a rule, the image characteristics that make up multidimensional feature vectors are divided into global and local [17, 18]. Global features such as color, texture, shape, and spatial information, for example, provide insight into the entire image and enable faster feature extraction and similarity calculations. Although global features do not allow searching for objects in an image, they are well-suited for image classification. Local features are used to search for objects in an image, including in problems of pattern recognition and labeling of objects in an image [19, 20].

Although many CBIR methods have been developed, they all suffer from the disadvantage of the “semantic gap,” which is the gap between low-level features describing images (machine-parsed pixels) and high-level semantics (human perception), resulting in irrelevant image retrieval [21, 22]. One of the main disadvantages of CBIR methods is their low performance (i.e., memory usage, scalability, speed, accuracy), due to the use of multi-dimensional functions for translating visual image content into numerical form and complex decision-making algorithms, for example, such as machine learning [23, 24] or genetic algorithms [25, 26]. The authors in [7, 27] noted that the most widely known clustering methods, such as k-means or nearest neighbor pairing [28, 29], are too slow for digital image processing tasks. In addition, in these methods it is necessary to pre-specify the number of classes, which makes them inappropriate for the problem of image clustering in large databases in which information is constantly being added.

In addition to feature extraction methods and decision-making algorithms, the performance of CBIR systems is also affected by the way images are searched in the database, especially in large ones with tens of thousands of records. Methods for searching a database for a multidimensional feature vector are usually divided into two types [7]: direct search and search using functions. In direct database search methods, the feature vector is directly an index of the original image [30]. Feature retrieval methods map high-dimensional floating-point feature vectors to low-dimensional vectors or binary vectors, reducing the computational complexity of distance or similarity measures as well as storage costs. The second group of methods often uses hashing [31], which allows the use of simple metrics, such as the Hamming distance among hash codes. It should be noted that practical implementations of CBIR systems often combine both types of database search methods.

The analysis showed that existing methods for searching images in databases are of little use for solving the problem of clustering images accompanying text information on various sites, due to their low

performance when it is necessary to search in large databases that are constantly updated.

**Problem statement and purpose of the study.** The purpose of the research is to improve the performance of searching for “similar” images in large databases, in which the rate of adding information reaches 10-12 thousand images per day.

To achieve this goal, it is necessary to solve the following tasks:

- develop an image classifier with low sensitivity to increased database information.
- carry out a study of the properties of the developed image classifier.

### Development of an image classifier

The general model of an image classifier can be represented as a tuple

$$IC = \langle I, C, S, R, O \rangle, \quad (1)$$

where  $I = \{I_1, I_2, \dots, I_p\}$  is the set of images that need to be classified (collection of images);  $C = \{C_1, C_2, \dots, C_K\}$  is a set of clusters (image classes), while  $C_i \cap C_j = \emptyset \forall i \neq j$ ;  $S = \{\vec{S}_1, \vec{S}_2, \dots, \vec{S}_L\}$  – is a set of image signatures;  $R \subset C \times S$  – relationship between clusters and signatures;  $O: I \rightarrow C$  – is a clustering operation, which consists of transforming images, after which either an image  $I_n \in I$  with a signature  $\vec{S}_i \in S$  belongs to an existing cluster  $C_k \in C$ , or a conclusion is made about the need to create a new cluster  $C_{K+1} \in C$  to which this image can be assigned, while one image can be assigned to only one cluster.

By image signature we mean a vector of feature values used for unambiguous image classification.

The relation  $R$  has the following property:  $\forall C_i \in C \exists \vec{S}_j \in S: (C_i, \vec{S}_j) \in R$ .

Based on the proposed model (1), we can conclude that the speed of the classifier is primarily affected by the method of calculating the image signature. The paper proposes to use the two-dimensional discrete cosine transform (2D-DCT) to compute the image signature.

In general, the 2D-DCT  $F[u, v]$  ( $0 \leq u \leq N-1$ ;  $0 \leq v \leq N-1$ ) of an image  $\mathbf{A}$  (with dimensions  $N \times N$  pixels) is defined as follows:

$$F[u, v] = (2/N) \cdot \alpha[u] \alpha[v] \times \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left( A[x, y] \cos \frac{\pi(2x+1)u}{2N} \cos \frac{\pi(2y+1)v}{2N} \right); \quad (2)$$

$$\alpha[u] = \begin{cases} 1/\sqrt{2}, & \text{if } u = 0; \\ 1, & \text{otherwise,} \end{cases} \quad (3)$$

where  $F[u, v]$  are the coefficients of the 2D-DCT ( $0 \leq k \leq N-1$ ;  $0 \leq l \leq M-1$ );  $\alpha[u]$ ,  $\alpha[v]$  are normalizing coefficients calculated using expression (3).

The inverse 2D-DCT can be calculated as follows

$$A[x, y] = (2/N) \times \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left( \alpha[u] \alpha[v] F[u, v] \times \cos \frac{\pi(2x+1)u}{2N} \cos \frac{\pi(2y+1)v}{2N} \right), \quad (4)$$

where  $\alpha[u]$ ,  $\alpha[v]$  are normalizing coefficients calculated using expression (3).

The 2D-DCT has a number of useful properties that can be used when calculating an image signature. Let's look at these properties.

As a result of calculating the 2D-DCT of a matrix  $\mathbf{A}$ , the lowest frequencies will be located in the upper left part of the resulting matrix  $\mathbf{F}$ , and the highest frequencies will be located in the lower right part. Since the amplitude of low-frequency components is usually greater than that of high-frequency components, the upper left part of the display contains relatively large values that carry basic information about the image, and the lower right contains small values that carry information about details. Therefore, to increase the speed of signature calculation, it is advisable to use only a given small number of low-frequency components of the matrix  $\mathbf{F}$ . If the matrix  $\mathbf{A}_{my}$  is a mirror image of the matrix  $\mathbf{A}$  relative to the vertical axis, i.e.,  $A_{my}[x, y] = A[x, N-1-y]$ , then the 2D-DCT of the mirror image  $\mathbf{A}_{my}$  has the following form

$$F_{my}[u, v] = (2/N) \cdot \alpha[u] \alpha[v] \times \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left( (-1)^y A[x, y] \cos \frac{\pi(2x+1)u}{2N} \cos \frac{\pi(2y+1)v}{2N} \right).$$

If the matrix  $\mathbf{A}_{mx}$  is a mirror image of the matrix  $\mathbf{A}$  relative to the horizontal axis, i.e.,  $A_{mx}[x, y] = A[N-1-x, y]$ , then the 2D-DCT of the mirror image  $\mathbf{A}_{mx}$  has the following form

$$F_{mx}[u, v] = (2/N) \cdot \alpha[u] \alpha[v] \times \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left( (-1)^x A[x, y] \cos \frac{\pi(2x+1)u}{2N} \cos \frac{\pi(2y+1)v}{2N} \right).$$

Rotate an image on  $90^\circ$  is equivalent to the simultaneous transposition and mirror reflection of the matrix  $\mathbf{A}$  relative to the horizontal axis, i.e.,  $\mathbf{A}_{m90} = \mathbf{A}'_{mx}$ . Then the 2D-DCT rotated by  $90^\circ$  the matrix  $\mathbf{A}_{m90}$  has the following form

$$F_{m90}[u, v] = (2/N) \cdot \alpha[u] \alpha[v] \times \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left( (-1)^x A[y, x] \cos \frac{\pi(2x+1)u}{2N} \cos \frac{\pi(2y+1)v}{2N} \right).$$

Thus, the property of rotated and mirror images can be formulated as follows

$$|F[u, v]| = |F_{mx}[u, v]| = |F_{my}[u, v]| = |F_{m90}[v, u]|. \quad (5)$$

Property (5) is advisable to use to generate signatures that are invariant to rotations and mirror reflections of the image along any of the axes.

There are two different ways to implement the 2D-DCT. The first method is based on the direct calculation of monetary policy coefficients using expressions (2) – (4). However, this method requires significant computational resources, especially for large images. The second method uses a square discrete cosine transform matrix  $\mathbf{T}$ , which is calculated as follows

$$T[k, l] = \begin{cases} 1/\sqrt{N}, & \text{if } k = 0, 0 \leq l \leq N-1; \\ \sqrt{2/N} \cos(\pi(2l+1)k/2N), & \text{if } 1 \leq k \leq N-1, 0 \leq l \leq N-1. \end{cases} \quad (6)$$

Then the 2D-DCT of the matrix  $\mathbf{A}$  is calculated as follows

$$\mathbf{F} = \mathbf{T} \times \mathbf{A} \times \mathbf{T}' \quad (7)$$

The inverse 2D-DCT of the matrix  $\mathbf{F}$  is calculated as  $\mathbf{A} = \mathbf{T}' \times \mathbf{F} \times \mathbf{T}$ .

The main advantage of this method of determining the 2D-DCT is the speed of calculation. Here, a time-consuming operation is to calculate the matrix  $\mathbf{T}$  using expression (6). If all processed images have the same size, then the matrix  $\mathbf{T}$  only needs to be calculated once, and matrix multiplication operations can be performed quite quickly. Therefore, the work proposes to use the calculation of the 2D-DCT using the second method using expressions (6) and (7).

Since electronic media news feeds use both color and black-and-white (grayscale) images, the work proposes to use the following image model

$$Img = \langle \mathbf{A}_{gs}, \mathbf{A}_r, \mathbf{A}_g, \mathbf{A}_b \rangle, \quad (8)$$

where  $\mathbf{A}_{gs}, \mathbf{A}_r, \mathbf{A}_g, \mathbf{A}_b$  are the square matrices of the grayscale and color components of the image, respectively (RGB color model).

For halftone images, the matrix of color components  $\mathbf{A}_r, \mathbf{A}_g, \mathbf{A}_b$  are filled with zeros. For color images, the grayscale component matrix  $\mathbf{A}_{gs}$  is calculated from the color component matrices  $\mathbf{A}_r, \mathbf{A}_g, \mathbf{A}_b$ .

Let us denote the 2D-DCT of each component of the original image as  $\mathbf{F}_{gs}, \mathbf{F}_r, \mathbf{F}_g, \mathbf{F}_b$ . Taking into account the properties of DCT coefficients described above, the work proposes to quantize the coefficients in order to analyze only those frequency components that exceed a given threshold. Let us set the DCT coefficients for each component  $\mathbf{F}_{gs}, \mathbf{F}_r, \mathbf{F}_g, \mathbf{F}_b$  of the quantization matrix  $\mathbf{Q}_{gs}, \mathbf{Q}_r, \mathbf{Q}_g, \mathbf{Q}_b$ . Then quantization is performed by element-wise dividing each matrix of DCT coefficients  $\mathbf{F}_{gs}, \mathbf{F}_r, \mathbf{F}_g, \mathbf{F}_b$  into the corresponding quantization (weighting) matrix  $\mathbf{Q}_{gs}, \mathbf{Q}_r, \mathbf{Q}_g, \mathbf{Q}_b$ , the values of the elements of which increase as they move away from the upper left corner and approach the lower right corner:

$$F_{(q)}[u, v] = F[u, v] / Q[u, v].$$

Taking into account the placement of frequency components in the matrix  $F_{(q)}$ , we transform each of the matrices  $F_{(q)gs}, F_{(q)r}, F_{(q)g}, F_{(q)b}$  into the corresponding one-dimensional vectors  $\vec{f}_{gs}, \vec{f}_r, \vec{f}_g, \vec{f}_b$  (zigzag scanning of the matrix, starting from the upper left corner).

From each frequency component one can construct the following vector  $\vec{f}_i = (f_{gs}[i], f_r[i], f_g[i], f_b[i])$ ,  $i = \overline{0, M-1}$ , where  $M \ll N^2$ , i.e., it is proposed to take into account only the first  $M$  quantized frequency components. Then the length of the vector  $\vec{f}_i$  is calculated as  $|\vec{f}_i| = \sqrt{f_{gs}^2[i] + f_r^2[i] + f_g^2[i] + f_b^2[i]}$ .

Because the orientation of the original image is unknown (the image can be mirrored and/or rotated by  $90^\circ$ ), then taking into account property (5) to calculate a signature invariant to rotations and mirror reflections of the image along any of the axes, the paper proposes a vector  $\vec{s} = (s_0, s_2, \dots, s_{M-1})$  whose elements are calculated as follows expression

$$s_i = (|\vec{f}_i| + |\vec{f}_i^r|) / 2, \quad i = \overline{0, M-1}. \quad (9)$$

Then the  $\vec{S}_i = (S_{i0}, S_{i2}, \dots, S_{iM-1})$  image signature  $I_n$  with model (8) can be calculated as follows

$$S_{ii} = \text{round}(w_i s_i / |\vec{s}|), \quad i = \overline{0, M-1}, \quad (10)$$

where  $w_i$  are the weighting coefficients.

It is proposed to use the metric of city blocks as a

measure of the similarity of two vectors  $\vec{a}$  and  $\vec{b}$

$$d(\vec{a}, \vec{b}) = \sum_i |a_i - b_i|. \quad (11)$$

Then a new image  $I_n \in I$  with a signature  $\vec{S}_n \in S$  belongs to an existing cluster  $C_k \in C$  if there is such a signature  $\vec{S}_l$ , the distance to which is minimal among all signatures and less than a given threshold  $\varepsilon_k$ . Otherwise, a new cluster  $C_{K+1}$  with the image  $I_n \in I$  is created. Thus, the decision rule of the classifier has the following form

$$I_n \in \begin{cases} C_k, & \text{if } \exists (C_k, \vec{S}_l) : d(\vec{S}_n, \vec{S}_l) = \\ & = \min_{\substack{p \neq n, \\ p \in I, P}} d(\vec{S}_n, \vec{S}_p) < \varepsilon_k, 1 \leq k \leq K; \\ C_{K+1}, & \text{otherwise.} \end{cases} \quad (12)$$

### Study of the properties of the developed classifier

To develop a fast image classification algorithm using the proposed method, let's consider some of its properties. For experiments, we will take into account all DCT coefficients, i.e., quantization matrices  $Q_{gs}, Q_r, Q_g, Q_b$  are matrices of units.

Characteristics of the data set for experiments: number of images – 804724; image sizes are different; all images are in color; the number of image clusters is 445999, with 172344 (38.64%) clusters containing 2 images each, 2 clusters containing 1792 (0.22%) and 6828 (0.85%) images, the number of images in the remaining clusters is shown in Fig. 2.

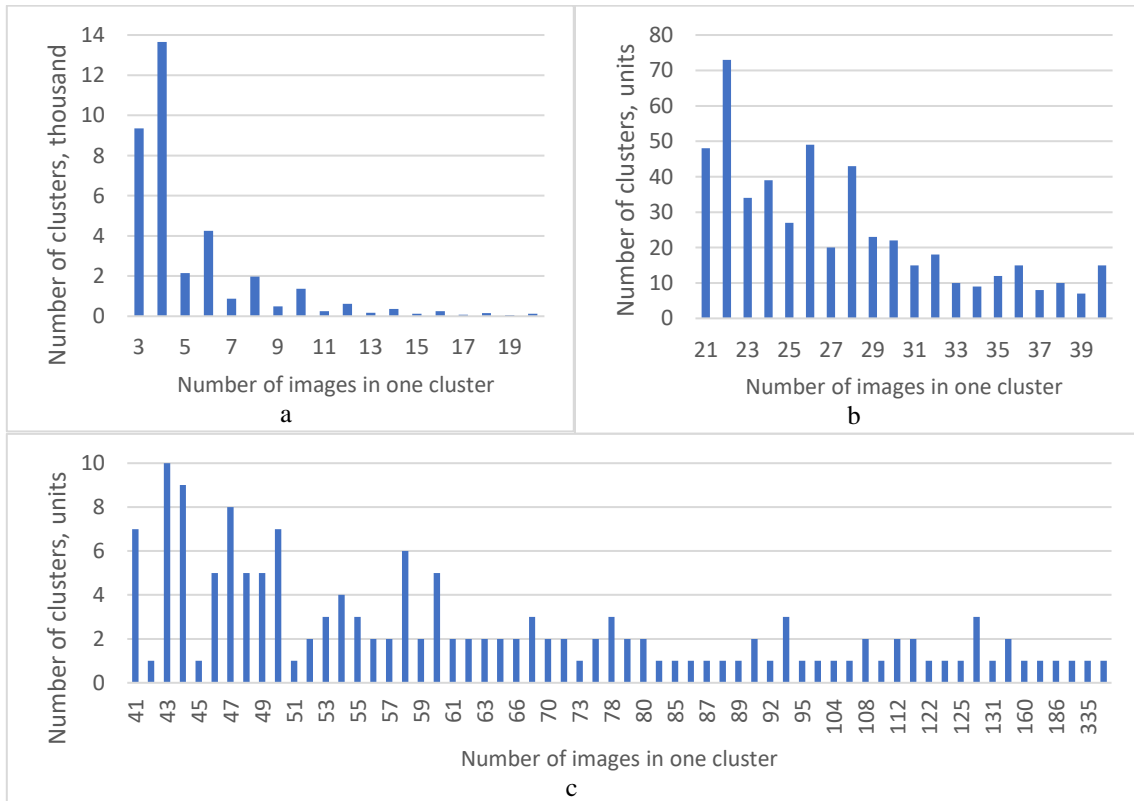


Fig. 2. Structure of the data set, number of clusters containing: a – from 3 to 20 images; b – from 21 to 40 images; c – more than 40 images



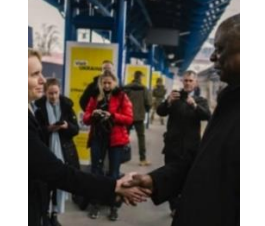


The matrix  $\mathbf{T}$  needs to be calculated once using expression (6). Research has shown that when an image is reduced to  $64 \times 64$  pixel sizes (using the cubic interpolation method), the signatures of the original and reduced images do not change (Fig. 3, Table 1). In all experiments, the signature length is  $M = 10$ . As can be seen from the vector values  $\vec{s} = (s_0, s_2, \dots, s_9)$  in Table 1, the zero value is an order of magnitude greater than the values of all other elements of the vector, therefore the following vector was used as weighting coefficients in

formula (10)  $\vec{w} = (0.1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$ .

Experiments have shown that further reducing the size of the images leads to the fact that the signatures begin to differ slightly (Fig. 3).

Therefore, at the first stage of image preprocessing, the work proposes to bring all images to the same  $64 \times 64$  pixel size. At the same time, the signatures retain their ability to identify specific images, for example, the signatures of images No. 1 and No. 2 differ significantly (Table 1)  $d(\vec{S}_1, \vec{S}_2) = 103$ .

Table 1 – Examples of signatures for images of different sizes

No.	Image	Image dimensions, pixels	Unnormalized signature vector $\vec{s} = (s_0, s_2, \dots, s_9)$ according to (9)	Signature $\vec{S} = (S_0, S_2, \dots, S_9)$
1		$1000 \times 1500$	(302204, 26798, 26798, 23164, 14941, 23164, 7439, 13837, 13837, 7439)	(47, 42, 42, 36, 23, 36, 12, 22, 22, 12)
		$64 \times 64$	(1579 2, 1401, 1401, 1210, 782, 1210, 388, 722, 722, 388)	
2		$448 \times 669$	(151201, 7120, 7120, 14308, 5998, 14308, 10071, 6079, 6079, 10071)	(47, 22, 22, 44, 18, 44, 31, 19, 19, 31)
		$64 \times 64$	(17680, 833, 833, 1667, 699, 1667, 1177, 709, 709, 1177)	
3		$400 \times 360$	(4888 0, 11665, 11665, 2642, 4948, 2642, 6838, 9627, 9627, 6838)	(20, 47, 47, 11, 20, 11, 28, 39, 39, 28)
		$64 \times 64$	(824 4, 1969, 1969, 445, 836, 445, 1153, 1625, 1625, 1153)	
4		$400 \times 360$	(1602 3 3, 4984, 4984, 7209, 509, 7209, 3515, 3097, 3097, 3515)	(75, 23, 23, 34, 2, 34, 16, 15, 15, 16)
		$64 \times 64$	(2 6998, 835, 835, 1216, 89, 1216, 595, 523, 523, 595)	
5		$465 \times 620$	(142646, 21929, 21929, 17459, 5352, 17459, 5958, 3118, 3118, 5958)	(33, 50, 50, 40, 12, 40, 14, 7, 7, 14)
		$64 \times 64$	(17005, 2612, 2612, 2082, 636, 2082, 710, 371, 371, 710)	

If in the decision rule (12)  $\varepsilon_k = 0$ , then completely identical images or images that cannot be visually distinguished will be collected in the same clusters. Fig. 4 shows graphs of the dependence of the number of clusters  $K(\varepsilon)$  and the number of image classification errors  $CE(\varepsilon)$  on the threshold value  $\varepsilon$  (in each of the experiments the threshold value for all clusters  $\varepsilon_k$  was set the same).

Analysis of the graphs shown in Fig. 4 showed that increasing the threshold above 40 leads to a sharp increase

in image clustering errors, while the number of clusters sharply begins to decrease (i.e., clusters begin to merge).

Image analysis showed that at a threshold,  $\varepsilon_k < 10$  similar images fall into one cluster. Examples of similar images are the images in Fig. 5.

In the case of fuzzy clustering, such images can be combined into one cluster using the developed classifier, because, in the future, text content is clustered not only based on images, but also other text parameters, such as news date, headline, keywords, and so on.

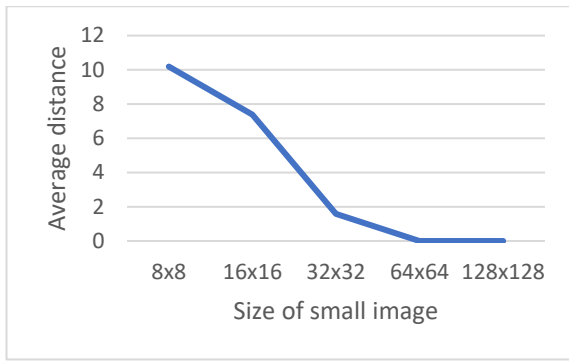


Fig. 3. Graph of the dependence of the average distance between the original image and the reduced one on the size of the reduced image

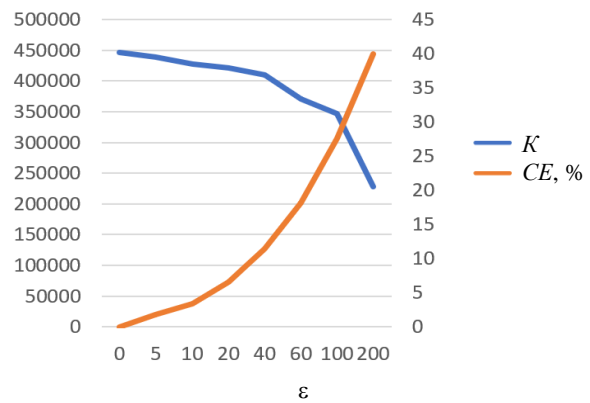


Fig. 4. Charts  $K(\epsilon)$  and  $CE(\epsilon)$



(46, 31, 31, 45, 9, 45, 20, 23, 23, 20) (46, 30, 30, 45, 9, 45, 20, 24, 24, 20) (46, 30, 30, 45, 9, 45, 21, 23, 23, 21)

Fig. 5. Examples of similar images that can be combined into one cluster and their signatures

### Quick searching in large databases

When a new image enters an information system (for example, a news portal), the following basic steps are performed for classification:

- 1) reducing the image size to  $64 \times 64$ ;
- 2) calculation of the signature using expression (10) taking into account expressions (6)-(9);
- 3) classification according to the decision rule (12) taking into account the metric (11).
- 4) writing a new image to the database indicating an existing cluster, if one is found, or a new cluster, if the image is unique.

To proceed with step 3, the first task is to attempt to find an existing cluster in the database. We will consider the case when in the decision rule (12) for all clusters  $\epsilon_k = 0$ .

Modern relational database management systems allow for fast searching through indexed fields. Then, to implement a quick search, it is necessary to perform indexing in the relational database by the field containing the signature. According to the proposed classifier model (1), a signature is a vector containing several values, so it is necessary to convert this vector into one field by which the database will be indexed and subsequently searched.

The work proposes to convert a vector  $\vec{S}_i = (S_{i_0}, S_{i_2}, \dots, S_{i_{M-1}})$  into a string of the following form  $S_{i_0} : S_{i_2} : \dots : S_{i_{M-1}}$ , i.e., write sequentially the

elements of the vector, separated by a colon. For example, the value of the string field for image signature No. 1 from Table. 1 will look like 47:42:42:36:23:36:12:22:22:12. This representation makes it possible to record the signature in one field, regardless of the value of  $M$  (the number of elements in the vector). Then the number of the cluster  $N_{kcluster}$  with images whose signature  $imsig$  matches the new image signature  $newimsig$  can be obtained by a simple query to the table `news_images`:

```
SELECT Nkcluster FROM news_images
WHERE imsig=newimsig
```

This search will be completed in the shortest possible time for tables with an index field. Experiments showed that on average the classifier speed was 0.03 sec/image. Thus, when up to 10-12 thousand images are received per day, up to 5-6 minutes are spent on their clustering.

### Conclusions

A general model of a content-based image classifier has been developed, which is based on transforming the image to its signature – a vector of feature values used for unambiguous image classification. Based on the proposed model, it is shown that the speed of the classifier is primarily influenced by the method of calculating the image signature. The paper proposes to use a two-dimensional discrete cosine transform to calculate the image signature. Based on a study of the

properties of the two-dimensional discrete cosine transform when analyzing the grayscale and color components of images, respectively (RGB color model), mathematical expressions for calculating the components of a signature that are invariant to rotations and mirror reflections of the image along any of the axes are obtained.

A method has been developed for classifying images by comparing their signatures, using the metric of city blocks as a measure of similarity.

The resulting method is the scientific novelty of this research.

A study of the properties of the developed image classifier was carried out, as a result of which the

conversion of all images to the same  $64 \times 64$  pixel size was justified.

At the same time, their signatures retain their ability to identify specific images, and the matrix of DCT coefficients is calculated once for all images, which significantly reduces the complexity of the method.

The experiments showed that clustering information from images turned out to be quite fast and low-cost in terms of information volumes and computing power requirements.

Further research is aimed at finding the optimal parameters of the proposed classifier, as well as studying the possibility of using the classifier for fuzzy clustering, as well as fuzzy search in the database.

#### REFERENCES

1. Amons, O. A., Yanov, Yu. O. and Bezpalyi, I. O. (2008), "Clustering of documents based on statistical proximity of terms", *Visnyk NTUU «KPI» Informatyka, upravlinnia ta obchysluvalna tekhnika*, No 49, pp. 55–62, available at: <https://ela.kpi.ua/handle/123456789/6114>
2. Veres, O. M., Kis, Ya. P., Kuhivchak, V. A. and Rishniak I. V. (2018), "Choose methods to find new or similar images", *Visnyk Natsionalnoho universytetu "Lvivska politekhnika"*, Seriya: *Informatsiini systemy ta merezhi*, No 887, p. 43–50, available at: [http://nbuv.gov.ua/UJRN/VNULPICM\\_2018\\_887\\_8](http://nbuv.gov.ua/UJRN/VNULPICM_2018_887_8)
3. Smeliakov, K. S., Chupryna, A. S., Sandrkin, D. L., Vakulik, Ye. V. and Drob, Ye. M. (2021), "Development of an invariant digital image model for quick search of data collections", *Zbirnyk naukovykh prats Kharkivskoho natsionalnoho universytetu Povitrianykh Syl*, No 2(68), pp. 108–115, doi: <https://doi.org/10.30748/zhups.2021.68.14>
4. Smeliakov, K. S., Sandrkin, D. L., Tovchyrechko, D. O., Vakulik, Ye. V. and Drob, Ye. M. (2021), "Exploration of the method of quick search of digital images in the collections of data", *Systemy obrobky informatsii*, No 2(165), pp. 54–63, doi: <https://doi.org/10.30748/soi.2021.165.07>
5. Ali, F. and Hashem, A. (2020), "Content Based Image Retrieval (CBIR) by statistical methods", *Baghdad Science Journal*, vol. 17(2 SI), pp. 694–700, DOI: [http://dx.doi.org/10.21123/bsj.2020.17.2\(SI\).0694](http://dx.doi.org/10.21123/bsj.2020.17.2(SI).0694)
6. Salih, F.A.A. and Abdulla, A.A. (2021), "An Efficient Two-layer based Technique for Content-based Image Retrieval", *UHD Journal of Science and Technology*, vol. 5(1), pp. 28–40, doi: <https://doi.org/10.21928/uhdjt.v5n1y2021.pp28-40>
7. Xiaoqing, Li, Jiansheng, Yang and Jinwen, Ma (2021), "Recent developments of content-based image retrieval (CBIR)", *Neurocomputing*, vol. 452, pp. 675–689, doi: <https://doi.org/10.1016/j.neucom.2020.07.139>
8. Salih, S.F. and Abdulla, A.A. (2021), "An Improved Content Based Image Retrieval Technique by Exploiting Bi-layer Concept", *UHD Journal of Science and Technology*, vol. 5(1), pp. 1–12, doi: <https://doi.org/10.21928/uhdjt.v5n1y2021.ppl-12>
9. Kashif, M., Raja, G. and Shaukat, F. (2020), "An efficient content-based image retrieval system for the diagnosis of lung diseases", *Journal of Digital Imaging*, vol. 33, pp. 971–987, doi: <https://doi.org/10.1007/s10278-020-00338-w>
10. Garg, M. and Dhiman, G. (2021), "A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants", *Neural Computing and Applications*, vol. 33, pp. 1311–1328, doi: <https://doi.org/10.1007/s00521-020-05017-z>
11. Ashraf, R., Ahmed, M., Jabbar, S., Khalid, S., Ahmad, A., Din, S. and Jeon, G. (2018), "Content based image retrieval by using color descriptor and discrete wavelet transform", *JMS*, vol. 42, pp. 1–12, doi: <https://doi.org/10.1007/s10916-017-0880-7>
12. Kenchappa, Y.D. and Kwadiki, K. (2022), "Content-based image retrieval using integrated features and multi-subspace randomization and collaboration", *International Journal of System Assurance Engineering and Management*, vol. 13, pp. 2540–2550, doi: <https://doi.org/10.1007/s13198-022-01663-9>
13. Mistry, Y., Ingole, D.T. and Ingole, M.D. (2018), "Content based image retrieval using hybrid features and various distance metric", *Journal of Electrical Systems and Information Technology*, vol. 5(3), pp. 874–888, doi: <https://doi.org/10.1016/j.jesit.2016.12.009>
14. Lee, K., Lee, Y., Ko, H.-H. and Kang, M. (2022), "A Study on the Channel Expansion VAE for Content-Based Image Retrieval", *Applied Sciences*, vol. 12(18), 9160, doi: <https://doi.org/10.3390/app12189160>
15. Bu, H.H., Kim, N.C. and Kim, S.H. (2023), "Content-based image retrieval using a fusion of global and local features", *ETRI Journal*, vol. 45(3), pp. 505–518, doi: <https://doi.org/10.4218/etrij.2022-0071>
16. Abdullah, Sura Mahmood and Jaber, Mustafa Musa (2023), "Deep learning for content-based image retrieval in FHE algorithms", *Journal of Intelligent Systems*, vol. 32(1), 20220222, doi: <https://doi.org/10.1515/jisys-2022-0222>
17. Datta, R., Joshi, D., Li, J. and Wang, J. Z. (2008), "Image retrieval", *ACM Computing Surveys*, 40(2), pp. 1–60, doi: <https://doi.org/10.1145/1348246.1348248>
18. Zalevska, O., Miroshnychenko, I., Smakovskiy, D., Haharin, O. and Palamar, I. (2023). "Improvement of the image clustering method", *Suchasni problemy modeliuвання*, vol. 24, pp. 79–86, DOI: <https://doi.org/10.33842/2313-125X-2022-24-79-86>
19. Halawani, A.H., Teynor, A., Setia, L., Brunner, G. and Retrieval, C.I. (2006), "Fundamentals and Applications of Image Retrieval: An Overview", *Datenbank-Spektrum*, vol. 18, pp. 14–23, available at: <https://api.semanticscholar.org/CorpusID:45584192>
20. Bansal, M., Kumar, M. and Kumar, M. (2020), "2D object recognition techniques: State-of-the-art work", *Archives of Computational Methods in Engineering*, vol. 28(3), pp. 1147–1161, doi: <https://doi.org/10.1007/s11831-020-09409-1>
21. Ibtihal, M. Hameed, Sadiq, H. Abdhussain and Basheera, M. Mahmmod (2021), "Content-based image retrieval: A review of recent trends", *Cogent Engineering*, vol. 8, 1927469, doi: <https://doi.org/10.1080/23311916.2021.1927469>
22. Sikandar, S., Mahum, R. and Alsaman, A. (2023), "A Novel Hybrid Approach for a Content-Based Image Retrieval Using Feature Fusion", *Applied Sciences*, vol. 13(7), 4581, doi: <https://doi.org/10.3390/app13074581>

23. Tzelepi, M. and Tefas, A. (2018), “Deep convolutional learning for content based image retrieval”, *Neurocomputing*, vol. 275, pp. 2467–2478, doi: <https://doi.org/10.1016/j.neucom.2017.11.022>
24. Sezavar, A., Farsi, H. and Mohamadzadeh, S. (2019), “Content-based image retrieval by combining convolutional neural networks and sparse representation”, *Multimedia Tools and Applications*, vol. 78(15), pp. 20895–20912, doi: <https://doi.org/10.1007/s11042-019-7321-1>
25. Phadikar, B.S., Phadikar, A. and Maity, G.K. (2018), “Content-based image retrieval in DCT compressed domain with MPEG-7 edge descriptor and genetic algorithm”, *Pattern Analysis and Applications*, vol. 21(2), pp. 469–489, DOI: <https://doi.org/10.1007/s10044-016-0589-0>
26. Alsmadi, M.K. (2020), “Content-based image retrieval using color, shape and texture descriptors and features”, *Arabian Journal for Science and Engineering*, vol. 45(4), pp. 3317–3330, doi: <https://doi.org/10.1007/s13369-020-04384-y>
27. Rusovych, S.Y. and Ponomarenko, N.N. (2012), “Study of the effectiveness of clustering methods in the formation of codebooks in digital image processing problems”, *Radioelektronni i kompiuterni systemy*, No 3, pp. 122–125, available at: [http://nbuv.gov.ua/UJRN/recs\\_2012\\_3\\_20](http://nbuv.gov.ua/UJRN/recs_2012_3_20)
28. Sampathila, N. and Martis, R.J. (2022), “Computational approach for content-based image retrieval of K-similar images from brain MR image database”, *Expert Systems*, vol. 39(7), e12652, doi: <https://doi.org/10.1111/exsy.12652>
29. Monowar, M.M., Hamid, M.A., Ohi, A.Q., Alassafi, M.O. and Mridha, M.F. (2022), “A Self-Supervised Spatial Recurrent Network for Content-Based Image Retrieval”, *Sensors*, vol. 22(6), 2188, doi: <https://doi.org/10.3390/s22062188>
30. Junjie, Cai, Qiong, Liu, Francine, Chen, Dhiraj, Joshi and Qi, Tian (2014), “Scalable Image Search with Multiple Index Tables”, *Proceedings of International Conference on Multimedia Retrieval (ICMR '14)*, Association for Computing Machinery, New York, NY, USA, pp. 407–410, doi: <https://doi.org/10.1145/2578726.2578780>
31. Cheng, S., Wang, L. and Du A. (2019), “An Adaptive and Asymmetric Residual Hash for Fast Image Retrieval”, *IEEE Access*, vol. 7, pp. 78942–78953, doi: <https://doi.org/10.1109/ACCESS.2019.2922738>

Received (Надійшла) 18.01.2024

Accepted for publication (Прийнята до друку) 10.04.2024

#### ABOUT THE AUTHORS / ВІДОМОСТІ ПРО АВТОРІВ

**Філатов Валерій Володимирович** – аспірант кафедри комп’ютерної інженерії та програмування, Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна;

**Valerii Filatov** – PhD Student of Computer Engineering and Programming Department, National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine;

e-mail: [filatov@mail.com](mailto:filatov@mail.com); ORCID ID: <https://orcid.org/0009-0007-7762-1517>;

**Філатова Ганна Євгенівна** – доктор технічних наук, професор, професор кафедри комп’ютерної інженерії та програмування, Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна;

**Anna Filatova** – Doctor of Technical Sciences, Professor, Professor of Computer Engineering and Programming Department, National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine;

e-mail: [filatova@gmail.com](mailto:filatova@gmail.com); ORCID ID: <https://orcid.org/0000-0003-1982-2322>;

Scopus ID: <https://www.scopus.com/authid/detail.uri?authorId=56448583600>

**Поворознюк Анатолій Іванович** – доктор технічних наук, професор, професор кафедри комп’ютерної інженерії та програмування, Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна;

**Anatolii Povoroznyuk** – Doctor of Technical Sciences, Professor, Professor of Computer Engineering and Programming Department, National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine;

e-mail: [ai.povoroznyuk@gmail.com](mailto:ai.povoroznyuk@gmail.com); ORCID ID: <https://orcid.org/0000-0003-2499-2350>;

Scopus ID: <https://www.scopus.com/authid/detail.uri?authorId=55225664000>

**Омаров Шахін Анвер огли** – доктор економічних наук, доцент, професор кафедри комп’ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківський національний університет радіоелектроніки, Харків, Україна;

**Shakhin Omarov** – Doctor of Economic Sciences, associate Professor, Professor of Computer-Integrated Technologies, Automation and Robotics Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;

e-mail: [shakhin.omarov@nure.ua](mailto:shakhin.omarov@nure.ua); ORCID ID: <https://orcid.org/0000-0002-2887-9083>.

#### Класифікатор зображень для швидкого пошуку у великих базах даних

В. В. Філатов, Г. Є. Філатова, А. І. Поворознюк, Ш. А. Омаров

**Анотація. Актуальність.** Лавиноподібне зростання кількості інформації в Інтернеті потребує розробки ефективних методів швидкої обробки такої інформації в інформаційних системах. Кластеризація новинної інформації проводиться як з урахуванням морфологічного аналізу текстів, так і графічного контенту. Таким чином, актуальним завданням є кластеризація зображень, що супроводжують текстову інформацію на різних веб-ресурсах, включаючи портали новин. **Предмет дослідження:** класифікатор зображень, що малочутливий до зростання кількості інформації в базах даних. **Метою дослідження** є підвищення продуктивності пошуку однакових зображень у базах даних, у яких швидкість додавання інформації досягає 10-12 тисяч зображень на добу, шляхом розробки класифікатора зображень. **Методи, що використовуються:** математичне моделювання, пошук зображень на основі контенту, двовимірне дискретне косинусне перетворення, методи обробки зображень, методи прийняття рішень. **Отримані результати.** Розроблено класифікатор зображень, що малочутливий до зростання кількості інформації в базах даних. Виконано аналіз властивостей розробленого класифікатора. Проведені експерименти показали, що кластеризація інформації за зображеннями за допомогою розробленого класифікатора виявилася досить швидкою та маловитратною з погляду обсягів інформації та вимог до обчислювальної потужності.

**Ключові слова:** інформаційні системи; пошук зображень на основі контенту; класифікатор зображень; великі бази даних; двовимірне дискретне косинусне перетворення.