Date of publication MAR-29, 2023, date of current version DEC-13, 2022.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.22.1.2879

Image Pair Comparison for Near-duplicates Detection

OLEKSII GOROKHOVATSKYI, OLENA PEREDRII
Simon Kuznets Kharkiv National University of Economics, Nayku Avenue, 9-a, 61166, Kharkiv, Ukraine

Corresponding author: Oleksii Gorokhovatskyi (e-mail: oleksii.gorokhovatskyi@gmail.com).

ABSTRACT The paper describes the search for a solution to the image near-duplicate detection problem. We assume that there are only two images to compare and classify whether they are near-duplicates. There are some traditional methods to match pair of images, and the evaluation of the most famous of them in terms of the problem is performed in this research. The effective thresholds to separate near-duplicate classes are found during experimental modeling using the INRIA Holidays dataset. The sequence of methods is proposed to make the joint decision better in terms of accuracy. It is shown also that the accuracy of binary classification of the proposed approach for the combination of the histogram comparison and ORB descriptors matching is about 85% for both near-duplicate and not near-duplicate pairs of images. This is compared to the existing methods, and it is shown, that the accuracy of more powerful methods, based on deep learning, is better, but the speed of the proposed method is higher.

KEYWORDS image; near-duplicates; non-near-duplicate; descriptors; similarity; accuracy; threshold; binary classification.

I. INTRODUCTION

SPREAD of smartphones made the creation of personal photo collections available for everyone. It is common for a lot of people to take multiple photos of the same life scene and choose the best one later. Some people will never make this in the future having a lot of similar pictures, which require a lot of memory capacity in smartphones, computers, and other devices. Automatic processing and detection of near-duplicates may simplify this process and allow the selection of only the best (in some context) images.

So, such images that contain the same scene but differ a bit are commonly called near-duplicates (ND), but the definition of the ND term varies from paper to paper [1]. Detection of ND is quite subjective [2-4], because the importance of differences between two images may be major for one person and minor for another. For instance, images may differ not only in scene, view angle, camera position, etc., but also from a technical point of view, having some artifacts, different contrast, compression, etc.

The search for near-duplicate and duplicate images is a special case of the other problem called Content-based Image Retrieval, which is about the search of similar images in the more common case.

Fig. 1 shows an example of a pair of images we call near-duplicates in this paper: the content is similar but the images are not full duplicates. It follows from the background that the camera had different positions as well as some other details in the images. This example also shows the main source of similar images, which is a collection of home pictures.

Figure 1. Example of near-duplicate images.

II. PROBLEM STATEMENT

The purpose of the paper is to investigate and develop a method for verifying whether a pair of images can be called near-duplicates or not. We want to develop a common method that has no prior information or metadata about images and can compare only two images without comparing the first image with the dataset, searching for the closest possible near-

duplicates. For some tasks (e.g., the analysis of frames in video sequences), it is impossible to compare one image with all existing ones.

We will use only a tiny portion of the training images to build the model.

Formally this is a binary classification problem: having a pair of images, we need to make a decision whether they are near-duplicates or not.

Let $M(I_1, I_2)$ be some method of comparison of the pair of images I_1 and I_2, and let $v = M(I_1, I_2)$ be some output value, that this method returns. Our aim is to find a threshold value v_T that maximizes classification accuracies for ND and non-near-duplicate (NND) classes, e.g., if $v \geq v_T \rightarrow$ND otherwise NND.

We also want to understand the effectiveness or accuracy of the method as well as its performance.

III. LITERATURE REVIEW

The most straightforward methods are pixel-based. The simplest of them includes iterating over both images pixel-by-pixel and calculating mean squared error (MSE) based on the difference between pixel intensities in the images being compared. There are also normalized modifications of this value (NMSE) that keep the output value within some specific range [0;1] or another.

Another method is based on similarity (Structural Similarity Index Measure, SSIM). It measures the difference between two similar images and returns a quality value, but it cannot find out which image is better without knowing this beforehand. During SSIM, the statistical features of two floating windows are compared [5].

Both MSE and SSIM are similar, but their disadvantage is that they do not take into account the natural way of perception and comparison of images by humans. Such pixel comparison methods can only be applied to images of the same size, which is more or less suitable for finding exact duplicates of image collections but not near-duplicates because commonly images are of different sizes. Resizing images in turn distorts their features, which is critical for pixel comparison methods.

Another simple image comparison approach is the use of image histograms [6]. It includes the conversion of an image into HSV color space, building histograms for hue and saturation channels, normalization of histogram, and comparison, e.g., with the correlation coefficient.

The other known fast method for the search of near duplicates is LSH (Locality-Sensitive Hashing) [7, 8]. The common idea of hashing is the creation of fingerprints for images, which are similar for similar images. This is not cryptographic hashing when tiny changes in the input result in huge changes in the output.

There are different hash types, like average hash, difference hash, perceptual hash, etc. All these operate on a rescaled (frequently) image brightness matrix. Hashes are typically compared in their entire form, but they can also be split into parts following piecewise comparison.

Another class of image comparison methods is feature-based. It consists of searching for specific image features – typically descriptors – and matching them with a pair of images [9].

ORB (Oriented FAST and Rotated BRIEF) image descriptor detector was proposed in [10] as an alternative to previous known SIFT and SURF methods. This feature detector builds descriptor in binary form. Scaled Gaussian pyramid is built at the first stage of building descriptor. Next, brightness extrema are found at each scale. To do this, the FAST [11] algorithm is used, according to which for each point of the image a circle of a certain radius is formed, and the number of adjacent pixels lying on it and having values less or more than the brightness of its center is counted. If the quantity of such points exceeds 75% of their possible number, the circle center is considered a candidate point of interest.

BRISK (Binary Robust Invariant Scalable Keypoints) descriptor was proposed in [12]. It is a development of SURF in terms of further improvement of FAST and BRIEF components. This method provides different alternatives to mask shapes compared with ORBs to identify key points. A mask named 9-16 is used, which analyzes nine consecutive pixels in a 16-pixel circle to meet the FAST criterion so that they are sufficiently brighter or darker than the center. Other masks are also used for different scales.

Given the location of key points and the corresponding scale values, the BRISK descriptor composes a binary string descriptor by combining the results of comparative brightness tests. The characteristic direction of each key point is identified to obtain oriented-normalized descriptors and to ensure invariance to rotation.

The BRISK descriptor concept uses a template by analyzing points that are evenly distributed in circles concentric with a key point. This provides integrated analysis and high processing or storage speeds. The BRIEF descriptor here recognizes the same areas of the image taken from different points of view.

AKAZE (Accelerated KAZE) descriptor that uses the benefits of nonlinear scale spaces was proposed in [13]. Fast Explicit Diffusion (FED) embedded in a pyramidal framework to dramatically speed-up feature detection in nonlinear scale spaces was proposed, as was a Modified-Local Difference Binary (M-LDB) descriptor that is scale and rotation invariant and has low storage requirements.

Matching of binary feature descriptors is usually based on the Hamming distance between bit descriptor values.

Finally, the last type of methods includes artificial neural network approaches. One way is to train networks to get features for the image and compare them. The other is to train and use Siamese networks, which get the pair of images as input and share weights. Anyway, these methods require a lot of time for training and creating suitable architecture, but they are the most effective. In this work, we are going to develop a method that is more clear and easier to implement.

IV. METHODOLOGY

Our approach in the paper is experimentally driven, as the goal is to develop a method that allows us to find ND pairs effectively enough for the specific dataset.

A. DATASET

We have used INRIA Holidays dataset [14, 15] that contains 1491 images, making over 1.1 million image pairs in total.

2072 pairs are near-duplicates, while 1108723 are not. This creates a huge imbalance between the ND and NND classes.

Our approach includes two stages. During the first one only 100 first images of the dataset were used in order to estimate the preliminary quality of the method and calculate the decision boundary. These 100 images include 4950 pair comparisons (105 near-duplicates and 4845 non-duplicates). At the second stage we performed near-duplicates detection for the entire dataset, which included 1491 images (2072 near-duplicate pairs and 1108723 non-duplicate).

B. FIRST STAGE

Our first goal was to detect whether it is possible to distinguish near-duplicate image pairs from non-duplicate ones using some threshold v_T for different existing methods M. During this stage we used only the first 100 images and pairs generated for them from INRIA Holidays dataset.

We tested various methods with different options and created a frequency histogram for decision value for pairs of images in order to calculate the threshold that allows us to split classes. The rough quality for this split was evaluated by the sum of accuracies for near-duplicate (ND) and non-near-duplicate (NND) classes, but keeping accuracies for both classes at least greater than 0.8.

C. SECOND STAGE AND SEQUENCE OF METHODS

We want to build an effective system using very few samples. 4950 pairs have been used at the first stage when the overall quantity of dataset include 1110795 pairs, so we used the first 0.4% of the data to build up the small comparison model and apply it to the entire dataset.

It is clear that using more data should make the performance of the first stage better but will require more time. Taking into account the variety of existing methods and parameters and our goal to test a lot of them, we decided to use only 100 images at the first stage.

The accuracy for both classes is also expected to decrease in the second stage compared to the first; the question is how much it will decrease.

When the accuracy provided by one method is not high it is possible to build up a cascade or ensemble of different methods. The cascade of methods includes the processing of the data in a waterfall manner, stage by stage, with various methods. The ensemble models combine the results of various methods to make some joint decisions via aggregation.

We will build a model, that is similar to both cascade and ensemble. Let $v_1 = M_1(I_1, I_2)$ be the value returned by the first method that makes the decision using threshold v_{T1}. Let L and H be the constants that define minimal and maximal values for the neighborhood near v_{T1} to estimate whether the decision is confident.

We claim the decision to be not confident if $v_1 \geq Lv_{T1}$ and $v_1 \leq Hv_{T1}$. In this case we use the decision made by another method $M_2(I_1, I_2)$ and the corresponding v_2 and v_{T2} values. Using additional methods and verifying their confidence is also an option.

Figure 2 depicts the entire approach, which includes method estimation at various stages.

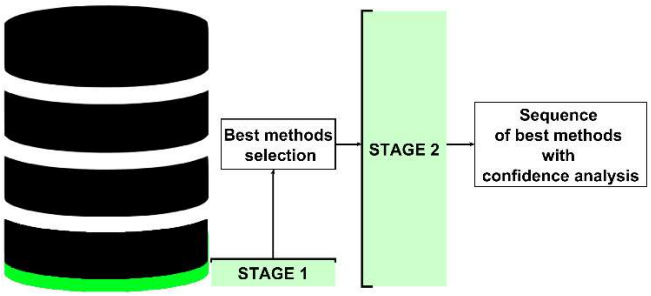

Figure 2. Two-stage proposed approach.

V. RESULTS AND EXPERIMENTS

The logic of performing experiments is the following.

We use each method M with various parameters to compare all pairs of images of the part of the dataset for the first stage and save $v = M(I_1, I_2)$ for ND and NND separately. Following that, we measure spread (reach) for all values for both classes and choose v_T value that maximizes the sum of accuracies for both classes. We also take into account the balance between accuracies, e.g., we prefer both accuracies to be 0.8 rather than 0.7 and 0.95. As a result of this stage, we have methods with the best accuracy.

During the second stage we estimate accuracies for both classes for these methods, using estimated thresholds v_T.

Finally, we propose the use of a combination of methods to achieve better results.

A. PIXEL-BASED METHODS

Firstly, we consider MSE, NMSE, and SSIM (implementations from skimage.metrics [16] were used). All these methods require the same size of images and have no other options, so we have performed resizing to the smaller image in pair.

Fig. 3 shows the spread of MSE values for ND and NND pairs of images. As one can see, they are intersecting. Values v for ND pairs vary from 549 to 20817; values for NND images are in the [1300; 30700] range. We built histograms for these values and plotted them as graphs in Fig. 3. After that, we found the threshold that allows us to distinguish ND from NND classes. Let us assume that if MSE is greater or equal to 1000, the pair of images represents ND. In this case we will successfully classify 98% of the ND cases but 0% for NND. So, for all methods we try to get a threshold that provides the best accuracies for two classes.

For MSE we were able to achieve only 32% accuracy for ND and 33% for NND using v_T=5400 threshold. NMSE allows us to get pretty similar results; normalization does not improve the situation.

SSIM frequency values for ND are in the range [0.06; 0.77] and for NND between 0.06 and 0.7. Despite such strong intersection (Fig. 4), one can see that more ND pairs exceed the 0.4 threshold and more NND pairs don't. The effective 0.4 threshold allows us to properly classify 59% of ND and 76% of NND, while the v_T=0.35 threshold allows reaching 66% accuracy for both classes.

LSH is one of the ways to reduce data dimensionality and build short signatures (hashes) of the image. Hash values should be similar for similar images. The hashing process includes rescaling an image, grayscaling, and building a binary image mask to generate a hash. The difference hash we used

here includes a comparison of pixel rows and assigning black (or white) to the pixels with higher intensity. Here, we adopted this to increase the stability of the results and include a minimum gap of 20 intensity points to ensure the difference between pixels is strong enough.

Signatures are compared by splitting them into bands and looking for partial similarity. So, the usage of LSH requires size of hash (approximate size to downscale initial image to) and quantity of bands to be set up. We used the implementation of LSH available in [7], excluding the search of the signature of the first image in all other images in the dataset but just building and comparing two signatures for the pair of images being considered.

We performed experiments with different parameters and the best result we achieved was 53% accuracy for ND and 65% accuracy for NND for hash size 16 and 32 bands.

The last method we tested is the comparison of histograms [6]. This method consists of transforming the image to HSV color space, splitting the ranges for H (0-180) and V (0-256) into 40 bins each, normalizing and comparing them. The full implementation is available in [6].

Testing on the first 100 dataset images is promising, the similarity values for ND vary from 0.06 to 0.99, corresponding values for NND are in range (-0.11; 0.96), decision threshold v_T=0.35 allows us to classify correctly 85% both ND and NND representatives. The visualization of frequencies for similarity values is shown in Fig. 5. An additional ND curve with magnified amplitude is added for better trend visibility.

B. FEATURE-BASED METHODS

The work of feature-based methods includes three stages: using point detectors to get points of interest, building descriptors for these points, and finally matching descriptors. Different detectors may be combined with different descriptors. In this paper we used BRISK detector and descriptor, ORB descriptor was tested with Harris and FAST detectors.

There are also different known descriptors matching techniques, in this work we used brute force and K−nearest neighbor search. Additionally, Lowe test ratio as well as regular filtering of matches were applied.

A lot of methods belonging to this class have various parameters that justify their work. Additionally, there are different methods to compare descriptors and filter them.

We tested BRISK, AKAZE, and ORB with different options.

The best result for BRISK we achieved with the detection threshold of 60 according to [17]. Matches with a Hamming distance only less than 64 were used after a brute force match between all descriptors. If the quantity of matched points is at least 4, we will conclude that the pair of images contains near-duplicates, and this threshold yields 84% ND and 85% NND correct classifications.

For AKAZE we used the upright KAZE descriptor [18] and filtered out matches with a distance greater than 50. Making the decision by threshold 4 makes it possible to classify correctly 88% ND and 85% NND pairs.

A lot of parameters for ORB were tested. The best combination includes FAST keypoint detector, 10000 features, filtration of matches to use ones with distances less than 64, and KNN matcher with Lowe test ratio 0.75 [19]. 86% of ND have at least 52 matches and 89% of NND have less, than 52. Probably, these values may be better when using more features, but this will require more time.

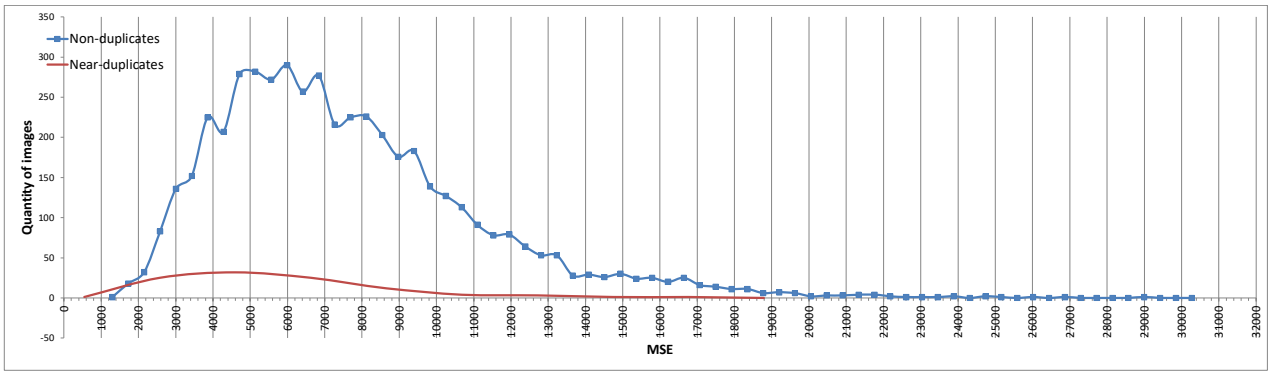

Figure 3. MSE frequency values.

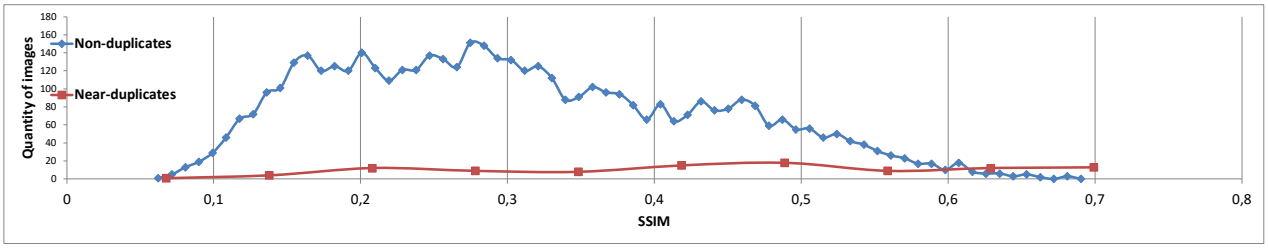

Figure 4. SSIM frequency values.

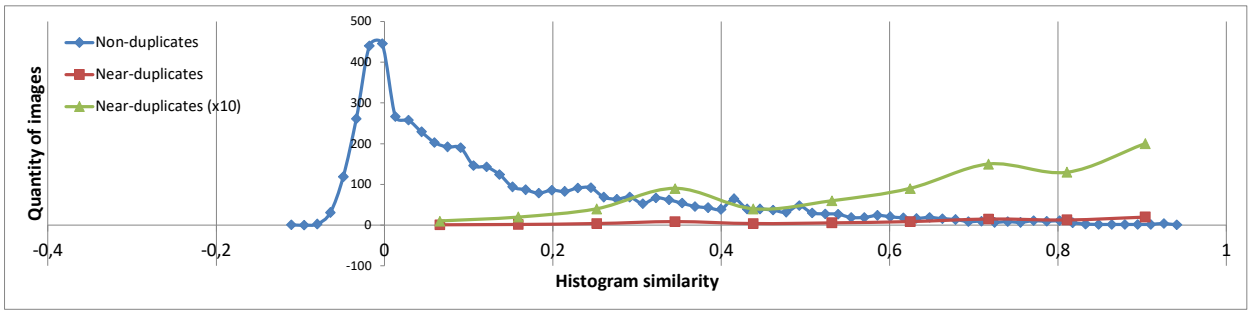

Figure 5. Histogram frequency values.

As a result of the first stage, three methods were chosen to perform full-scale modeling using the entire dataset. These methods and corresponding accuracy values are shown in Table 1. As one can see, the accuracies are much worse now. It is obvious that comparing histograms is much faster and it better classifies ND than NND.

Table 1. Results of the modeling on the entire dataset

Method and description	Stage 1 accuracies (ND, NND)	Stage 2 accuracies (ND, NND)	Average time per image pair, s.
Histogram 40x40 bins for H and V channels, v_T =0.35	84.76%, 84.52%	83.97%, 79.83%	0.3
ORB 10k features, FAST corner detector, max allowed distance is 64, KNN with Lowe test ratio 0.75, v_T =52.	85.71%, 88.59%	67.86%, 86.08%	1.7
BRISK Point threshold detection is 60, max allowed distance is 60, brute force matching, v_T =4	83.81%, 84.69%	65.54%, 83.43%	1.7

C. SEQUENCE OF METHODS

Let us look at ways to improve accuracy by using a combination of several methods. As the first method, it is possible to choose the comparison of histograms. If the decision on it is not certain (close to the threshold value), it is possible to use the result of classification by another method (for example, ORB). The simulation results showed that the combination of these methods allowed one to correctly classify 85% of pairs for both the ND class and the NND, which is a good result.

The method based on the comparison of histograms with the decision threshold v_T =0.35 was chosen as the first method. Thus, 885064 (79.93%) pairs that were not similar and 1740 (83.98%) ND were correctly classified. The decision was considered confident if the value for its adoption was outside the 30% deviation of v_T i.e, greater than 0.455 or less than 0.245. There were 908851 such values (82.97%). For pairs of images whose histogram comparison values ranged from 0.245 to 0.455, the comparison of key points using the ORB method was used. This allowed us to correct the results of the classification for 76 pairs of ND, while 45 correctly classified pairs were lost. The corresponding quantity for corrected pairs composed of NND images was 70688, while 16343 previously correctly classified pairs received an incorrect class label. After such a correction, we have about 85% of the correctly classified pairs both for ND (1771 of 2072) and NND (939409 of 1108723) classes. The average time per pair of images processed with this correction is 0.6 sec.

An example of such corrections is shown in Fig. 6. A pair depicted in Fig. 6 is known to be the pair of near-duplicates. A comparison of histograms shows a similarity value of 0.27, which is less than the decision threshold of 0.35, so this pair is classified as NND using histograms. But the ORB comparison finds 409 matched descriptors which is significantly higher than threshold 52. So, the final classification is ND.

Of course, this is true – a second check may change the initially correct classification result after the first method.

Figure 6. Near-duplicate images.

VI. DISCUSSION

The main approach for ND searching covered in most scientific papers relates to the search for the most similar (in some terms) image in the entire dataset – Content-based Image Retrieval problem. So, one image, typically in the form of its features, is compared to all existing ones.

This is not what we do here, because we compare only two of images. Commonly, we can say that the task of searching for the most similar image can be represented as a comparison of each image with all the others in pairs, measuring the similarity score for each pair and comparing them. The methods developed for this approach are not always suitable for comparison of the independent images on a single pair.

It is also shown in the Results and Experiments section that a lot of known image comparison methods do not work successfully enough to detect ND pairs in the scope of the massive experiment.

Here we are going to compare our results with one of the best methods based on the neural networking CLIP (Contrastive Language-Image Pre-Training) approach. It is a model trained on the (image, text) pairs and it can be used to predict text based on the image. This model maps images and text to the same numerical vector space. We used sentence

transformers implementation and the "clip-ViT-B-32" CLIP model here [20, 21].

After the first stage, this method allows us to get 0.91 accuracy for both classes with v_T=0.8 (meaning if the similarity ratio for a pair of images is greater than or equal to 0.8, we classify it as ND). Testing on the entire dataset shows the NND detection accuracy to be more than 99% and almost 92% for ND pairs. The average classification time per pair of images is about 2.1 seconds, of which 1.3 are spent for loading the CLIP model.

The other approach we compared our results against is the direct matching of the outputs of the pretrained neural network. We used ResNet50 with pretrained ImageNet weights without top classification layers, followed by the comparison of output features values for pair candidates with the cosine similarity measure [22, 23]. The results of modeling after the first stage with the effective threshold v_T=0.2 were 90% for ND and 87% for NND classes and after the second stage they were 91% and 90%, respectively. The average processing time is 0.8 seconds, and additionally, loading the ResNet model requires 1.1 sec.

We also tested the image similarity measurement module from the DeepRanking approach [24, 25] but it seems not to be applicable to our methodology or dataset as the distances for ND and NND are significantly intersecting (the accuracies for both classes at the first stage are lower than 30%).

As a summary for this section, it is obvious that the approaches based on the artificial neural networks could reach better accuracy (as expected), but they require more time (firstly for the loading of the bigger models) compared to the approach discussed in this paper.

VII. CONCLUSION

The enormous quantity of images stored on the different devices and occupying a lot of physical memory requires automatic methods to process and analyze them. One of the problems with such processing is the search for near-duplicate images.

The scientific novelty of the proposed paper includes the method of image pair comparison that uses the sequence of different image matching methods to detect near-duplicates with known accuracy and time.

The practical significance of the results includes a ready-to-use method for comparing two images and searching for near-duplicates. Additionally, the effective thresholds and parameters for different image comparison methods are found as a result of experimental modeling. It is also shown, that the proposed method has better performance compared to the more effective neural network models.

Prospects for further research relate to the improvements in two key indicators of the proposed methods, namely accuracy and processing time.

Acknowledgements

We dedicate this paper to the Armed Forces of Ukraine and all its branches, to medics, volunteers, and everyone who stands for Ukraine. This work would be impossible to complete without their strong and bravery resistance to external war aggression.

References

[1] L. Morra, F. Lamberti, "Benchmarking unsupervised near-duplicate image detection," *Expert Systems with Applications*, vol. 135, pp. 313-326, 2019. https://doi.org/10.1016/j.eswa.2019.05.002

[2] A. Jaimes, S. Chang and A. Loui, "Detection of non-identical duplicate consumer photographs," *Proceedings of the Fourth International Conference on Information, Communications and Signal Processing,* Singapore, December 15-18, 2003, vol. 1, pp. 16-20. https://doi.org/10.1109/ICICS.2003.1292404

[3] J. Chum, J. Philbin and A. Zisserman, "Near duplicate image detection: min-Hash and tf-idf weighting," *Proceedings of the British Machine Vision Conference*, Leeds, UK, September 1-4, 2008, pp. 1-10. https://doi.org/10.5244/C.22.50

[4] A. Jinda-Apiraksa, V. Vonikakis and S. Winkler, "California-ND: An annotated dataset for near-duplicate detection in personal photo collections," *Proceedings of the 5th Intenational Workshop on Quality of Multimedia Experience*, Klagenfurt, Austria, July 3-5, 2013. https://doi.org/10.1109/QoMEX.2013.6603227

[5] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, № 4, pp. 600-612, 2004. https://doi.org/10.1109/TIP.2003.819861

[6] OpenCV Compare Images, 2022, [Online]. Available at: https://www.delftstack.com/howto/python/opencv-compare-images/

[7] LSH for near-duplicate image detection, 2021, [Online]. Available at: https://github.com/mendesk/image-ndd-lsh

[8] Fingerprinting Images for Near-Duplicate Detection, 2020, [Online]. Available at: https://realpython.com/fingerprinting-images-for-near-duplicate-detection/

[9] V". Gorokhovatsky, D. Pupchenko, K. Solodchenko, "Analysis of properties, characteristics and results of the use of advanced detectors to determine the specific points of the image," *Control, Navigation and Communication Systems*, vol. 1, issue 47, pp. 93–98, 2018. https://doi.org/10.26906/SUNZ.2018.1.093

[10] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proceedings of the International Conference on Computer Vision*, Barcelona, Spain, November 3-16, 2011, pp. 2564-2571. https://doi.org/10.1109/ICCV.2011.6126544

[11] E. Rosten, T. Drummond, "Machine Learning for High-Speed Corner Detection," *Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science*, vol. 3951, pp. 430-443, 2006. https://doi.org/10.1007/11744023_34

[12] S. Leutenegger, M. Chli and R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," *Proceedings of the International Conference on Computer Vision*, Barcelona, Spain, November 3-16, 2011, pp. 2548-2555. https://doi.org/10.1109/ICCV.2011.6126542

[13] P. Alcantarilla, J. Nuevo and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *Proceedings of the British Machine Vision Conference*, Bristol, UK, September 9-13, 2013, pp. 13.1-13.11. https://doi.org/10.5244/C.27.13

[14] H. Jegou, M. Douze, C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," *Computer Vision – ECCV 2008. ECCV 2008. Lecture Notes in Computer Science*, vol. 5302, pp. 304-317, 2008. https://doi.org/10.1007/978-3-540-88682-2_24

[15] INRIA Holidays dataset, 2008, [Online]. Available at: http://lear.inrialpes.fr/~jegou/data.php

[16] Scikit-image, 2022, [Online]. Available at: https://scikit-image.org/

[17] cv::BRISK Class Refeence, 2022, [Online]. Available at: https://docs.opencv.org/4.x/de/dbf/classcv_1_1BRISK.html

[18] cv::AKAZE Class Reference, 2022, [Online]. Available at: https://docs.opencv.org/3.4/d8/d30/classcv_1_1AKAZE.html

[19] cv::ORB Class Reference, 2022, [Online]. Available at: https://docs.opencv.org/3.4/db/d95/classcv_1_1ORB.html

[20] Sentence Transformers: Multilingual Sentence, Paragraph, and Image Embeddings using BERT & Co., 2022, [Online]. Available at: https://github.com/UKPLab/sentence-transformers

[21] clip-ViT-B-32, 2021, [Online]. Available at: https://huggingface.co/sentence-transformers/clip-ViT-B-32

[22] Image Similarity in Percentage, 2020, [Online]. Available at: https://github.com/XingLiangLondon/Image-Similarity-in-Percentage

[23] P. Kasnesis, R. Heartfield, X. Liang, L. Toumanidis, G. Sakellari, C. Patrikakis, G. Loukas, "Transformer-based identification of stochastic information cascades in social networks using text and image similarity,"

Applied Soft Computing, vol. 108, 2021. https://doi.org/10.1016/j.asoc.2021.107413

[24] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen and Y. Wu, "Learning fine-grained image similarity with deep ranking," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, USA, June 23-28, 2014, pp. 1386-1393. https://doi.org/10.1109/CVPR.2014.180

[25] Image Similarity using Deep Ranking, 2018, [Online]. Available at: https://github.com/akarshzingade/image-similarity-deep-ranking

***OLEKSII GOROKHOVATSKYI** is an Associate Professor at S. Kuznets KNUE, received Ph.D. degree in Artificial Intelligence Systems and Tools from Kharkiv National University of Radio Electronics in 2010. Research interests include computer vision, image processing, pattern recognition, artificial intelligence.*

***OLENA PEREDRII** is an Associate Professor at S. Kuznets KNUE, received Ph.D. degree in Mathematical Modelling from Kharkiv National University of Radio Electronics in 2012. Research interests are about image processing, perspective and affine distortions normalization, license plate detection.*