Image Plagiarism Detection Pipeline for Vast Databases

Mariam Kaprielova, Andrey Grabovoy
Antiplagiat Company
Moscow, Russian Federation
kaprielova, grabovoy@ap-team.ru
Moscow Institute of Physics and Technology
Moscow, Russian Federation
FRC CSC RAS
Moscow, Russian Federation

Ksenia Varlamova, Ivan Potyashin
Antiplagiat Company
Moscow, Russian Federation
kvarlamova, potyashin@ap-team.ru
Moscow Institute of Physics and Technology
Moscow, Russian Federation

Yury Chekhovich, Aleksandr Kildyakov
Antiplagiat Company
Moscow, Russian Federation
chehovich, kildyakov@ap-team.ru

Abstract—Reuse detection in academic works is a relevant problem. There already are automatic systems to detect many kinds of violations of academic ethics in work texts, such as translation reuses, paraphrases, machine generation and many others. However, much less attention is paid to the image reuse problem. At the same time, the level of development of technical means of image processing makes it easy to falsify the results of scientific research or violate the principles of academic ethics in other ways. In order to address this problem, it is necessary to develop a image reuse detection system which would achieve high performance on large document collections. This paper presents an approach that is designed to search for image reuse in large collections of sources. The pipeline involves three steps: image conversion into a vector representation, candidate search, and similarity estimation between query image and each of candidates obtained at the previous step. The article presents results of experiments on quality and latency estimation of the developed system. We obtained Recall@1 = 98% quality for collection of images created without automatic drawing systems, 59% quality for images of handwritten essays and latency about 0.32 seconds per query for the collection of 59 million objects. The results show that the proposed system can be scaled up and used for industrial tasks that require quick verification of hundreds of thousands of images on a large number of potential sources of reuse.

I. Introduction

The task of identifying plagiarism in academic writings has become very important in recent decades. Modern information technology has greatly simplified the process of document search and copying. The significant increase in the number of cases of plagiarism in academic work was one of the negative results of technological development. At present, a large part of universities and scientific organizations use automatic systems for the detection of reuses, allowing to detect violations of academic ethics in the texts of works [1], [2], [3], [4], [5]. However, much less attention is paid to the problem of image reuse. At the same time, the level of development of the technical means of image processing makes it possible to quickly and efficiently make minor changes to the images and represent them as results of personal scientific research without any concern to be accused of plagiarism. The existence of an automatic image reuse detection system would make it possible to identify such violations. In this work we present a design of an automatic image reuse detection system for large collections of documents. The system consists of three stages. The first stage is the construction of an image embedding. The second stage, the candidate search, involves the search for candidates, which narrows the number of possible sources of reuse. The third stage is an accurate comparison, leaving a small number of candidates as likely sources of reuse. This approach allows not only to work with images of different structure, but also to scale the task of searching for efficient work on large collections of documents.

II. Related work

Quite a number of studies are devoted to the problem of finding violations of the principles of academic ethics. For example, industrial solutions have been proposed for the text reuse problem [3], translated reuses [1], [2], paraphrase. However, a similar problem in computer vision has received much less attention. The problem of finding borrowed images is a relevant task of computer vision. Thus, the work [6], [7] gives an overview of the main methods of changing images when reuse. In the article [6] one of the largest experts in academic ethics in the field of image reuse, Elizabeth Bick comments on her experience in this field, referring to reuse methods and fields of science in which borrowed images are more common than others. Research shows that the main manipulations used to borrow images are compression, grayscale conversion, scale change, rotation and mirroring.

And the most common reuse is in the biomedical field. The papers [8], [9], [10], [11] deal with methods of image reuse. The authors of [8] propose a solution based on classical computer vision algorithms. The article proposes to process images in several stages. First extract features of the classic algorithms of computer vision (SIFT [12], ORB [13], SURF [14], PCA-SIFT [15], FREAK [16] and KAZE [17]). Further, to search images in the database, the feature set is indexed using LSH [18]. Candidate images are searched for by the resulting index and then compared with each by the distance between the image histograms. The work presents the results of experiments for a collection of only about 6000 images. Therefore, it is impossible to assume latency of the proposed approach when used on large collection of objects. Article [9] offers a solution based on the representation of the original image and the image from the database by discrete functions and applies the F1 transform [19] to these functions. Thus, having simplified representations of images in the form of F1 transformation component matrices, it is possible to search for the original image by comparing the resulting matrices. The authors of [10] have developed a method based on the binarization of images followed by the calculation of the distance between their histograms. A major drawback of this approach is the inability to work with large collections of images, as the method is based on pairwise comparison. Article [11] focuses on one particular case of image manipulation - copy-paste attack, the essence of which is to replace a significant fragment of the image with the background, so that the real object in the picture is hidden. The algorithm is based on measuring the autocorrelation of small image fragments with each other. It is worth noting that the leading modern search engines have the function of searching similar images. Such algorithms are based on article [20] which introduced an approach based on teaching a ranking function on images based on triplets consisting of input, similar and different images. Thus, vector representations are trained to preserve the original properties of image similarity. Existing approaches to searching for borrowed images can be divided into several types:

1) Methods based on distance between image histograms. These methods aim at comparing and analysing the distribution of pixels in images to identify similarities or differences between them. However, such solutions have several drawbacks. In particular, low computational efficiency is one of the main limitations of this approach. Such methods require large computational resources to perform histogram comparison operations on images, which may be impractical when analyzing large volumes of data. In addition, methods based on analysis of the distance between histograms of images are usually unstable to various image transformations, such as lighting changes, rotations, scaling. This limits their applicability for real world data, where images can be altered in various ways.
2) Methods based on classical computer vision algorithms. Methods based on classical computer vision algorithms are traditional approaches to image processing that are based on the study and analysis of image features. However, because of their high computational complexity, these techniques have low efficiency, making them limited to the development of industrial image detection systems. Moreover, such approaches are often based on manually designed features, which makes it impossible to unify this group of methods for collections of images from different domains. When working with a large amount of image data, such methods can consume significant computing resources and require substantial processing time. Thus, low computational efficiency and feature design limitations make these approaches limited to the development of industrial image detection systems.
3) Neural network approaches. These techniques are a group of approaches that use artificial neural networks to solve the problem of finding reused images. They achieve high accuracy and efficiency in image reuse detection, making them most suitable for this task. However, it should be noted that neural network approaches require significant computational resources on both training and inference stages. Thus, neural network approaches are a powerful tool for image reuse detection, but their successful application requires competent design and optimization of image processing. The development and improvement of neural network methods in this area is an important task in order to improve the efficiency of the systems of reuse search and ensure high quality of the result.

III. Problem statement

Given a dataset of images:

$$Im_{query} = \{im^i_{query}\} \tag{1}$$

There is also given a set of images $Im_c = \{im^i_c\}$. We suppose that for each image from $im^i_{query} \in Im_{query}$ there is only one "source" image im^i_c:

$$g : Im_{query} \to Im_c \tag{2}$$

The major quality criterion for this task is Recall@K maximization where Recall@K is a ratio of relevant images in the most similar K images retrieved by the method

$$\begin{gathered} Recall@K = \frac{1}{|Im_{query}|} * \\ * \sum_{im^i_{query}} |f(im^i_{query})@K \cap \{g(im^i_{query})\}| \end{gathered} \tag{3}$$

where f is a image retrieval model, $f(im^i_{query})@K$ is a set top-k images from original set the most similar to the image im^i_{query}.

After the model found a probable text reuse source for the suspicious image, the source should be verified by the expert. In practice the expert can analyse only a small number of retrieved images, therefore the formal optimization task is to find a mapping, that maximizes Recall@1 for our dataset:

$$\hat{f} = \arg\max_{f \in F}(Recall@1(f, g, Im_c, Im_{query})) \quad (4)$$

where F is a family of considered retrieval models.

IV. The image reuse detection method

The section has the following structure: in subsection A we describe the process of embedding generation, the subsection B is dedicated to vector search algorithm. In subsection C we focus on approaches we used for image similarity estimation. We offer a search system consisting of several blocks. The first block translates the image into a vector space. The second block is used to search for the closest candidates, and the third block is used to accurately compare candidates with the image-query. See Figure 1(a) for a diagram of the search engine. To fill the collection, a system consisting only of translation blocks into a vector space and a block of adding a vector representation to the index is used. The scheme of the indexing system is presented in Fig 1(b). Our research

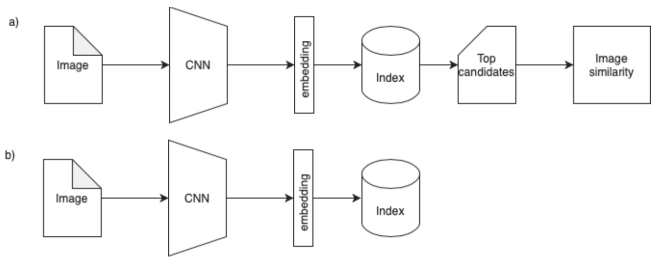

Fig. 1. The system pipeline: (a) search pipeline, (b) indexing pipeline

has shown that, depending on the type of images in the collection, it may be necessary to integrate additional stages into the system, such as specific image pre- and postprocessing, but the overall concept remains the same.

A. Vector representations

Since the system must have high performance, pairing the image query with all elements of the collection is not the optimal solution. Therefore, it is necessary to develop an algorithm for the search of candidate objects, among the limited number of which the original image is most likely to be found. We call an image a "candidate" if it is considered as a potential source of reuse. This approach to the problem allows us to significantly increase the performance of the system. In addition, the presence of a search phase allows for accurate comparisons to be made at later stages of the system, with better quality and more complex and resource-intensive models. The algorithm should be based on comparing the features and features of the images, taking into account different characteristics of the images, such as the shapes and structures of the objects in the image. Computer vision and image processing techniques, such as point-to-point matching algorithms, color and texture histogram comparison algorithms, and neural networks for image classification and comparison, are often used to solve similar problems. The search for candidate images requires the development of a method that transforms images into a vector representation in such a way that the structural characteristics and relationships between images are correctly reflected. This will solve the problem of image matching and comparison that often occurs when searching and analyzing large amounts of graphic data. The essence of this approach is to translate images into a multidimensional vector space, where each image will be represented as a unique vector, taking into account its structure and features. In this way, it is possible to compare the query image to the images from the collection more efficiently based on their vector representations. This will increase the accuracy and speed of searching for suitable images. There are several approaches to constructing the mapping of an object into a vector space. Conditionally they can be divided into two most prominent groups. The first group includes algorithms based on classical computer vision approaches such as SIFT [12] and HOG [21] and classical machine learning approaches. A significant disadvantage of such algorithms is that most solutions in this group require a lot of manual data processing and have a high specificity for a particular data domain. The second group may include approaches based on the use of neural network architectures. One of the most significant works in this field is the Deep Ranking model, introduced in 2014. The authors of the work proposed an architecture that was not simply intended to rank on the basis of the classes to which the objects belong, but on the basis of any characteristics of the objects, which may possess objects of the same class. At the stage of constructing vector representations, images are displayed in a metric vector space that preserves the structural relations between images. Therefore, it was decided to use the Deep Ranking [20] model to construct such a display. Studies have shown that the ResNet-50 [22] model is the most effective backup for the image classification problem.

This model was originally trained to learn similarity metrics between images; in our paper its main purpose is to create embeddings of images, s.t. similar images produce vectors that are close to each other in an embedding space and different images produce vector embeddings that are far apart. The model is trained on a set S of triplets of images. A triplet (Im, Im_p, Im_n) consists of a query image Im, a "positive" image Im_p of the same class c with Im and a "negative" image Im_n that is from another class $c_2 \neq c$. Each image of the triplet is fed into a deep convolutional neural network to form a vector embedding. Triplet loss is computed over vector embeddings to backpropagate the gradient flow through the network. A triplet loss function was defined in [23] as 5:

$$L(\mathrm{Emb}, \mathrm{Emb}_p, \mathrm{Emb}_n) = \max\{0, d(\mathrm{Emb}, \mathrm{Emb}_p) - d(\mathrm{Emb}, \mathrm{Emb}_n) + g\} \quad (5)$$

where $\mathrm{Emb}, \mathrm{Emb}_p, \mathrm{Emb}_n$ are vector embeddings of (Im, Im_p, Im_n), d is Euclidean distance and g is a gap parameter. In our work the dimension of vector embedding space was chosen to be $\dim = 1024$. For further details on training deepranking model, see [20].

B. Vector Search

At this stage, we find an extended list of similar images for the query image in the search system by searching the corresponding vectors that are closest to the vector of the input image. The most obvious way to do this is to compare the pairwise distances between the query vector and all vectors in the repository. However, given the large amount of data in the database (estimated volumes - hundreds of millions of objects), a complete search of all vectors is not the best option because of computational constraints. To speed up the process of finding similar images, different methods are often used to compare vectors effectively. These methods reduce the number of comparisons and speed up the search process, providing faster and more efficient search for similar objects. Thus, for efficient extraction of similar images based on vector representations, it is necessary to use specialized methods and algorithms that allow to quickly and accurately find the most suitable images in the database with a large amount of data. During the candidate search phase, all available images are compared against specific criteria to highlight the most similar images that may be potential sources of reuse. This process narrows the range of suspicious images and focuses on further analysis and verification of reuse. Thus, the development of the vector search algorithm is an important part of the task of image reuse detection and requires the application of specialized techniques to effectively and accurately identify candidates. At the candidate search stage, it is assumed that images are pre-selected without specifying which images have been borrowed. There is a wide range of methods based on index construction for approximate solution of the nearest object search problems, which allows to optimize the search and speed up computing processes. These methods are implemented using different algorithms and data structures designed to effectively represent and organize large amounts of information. Industrial solutions such as Annoy [24], Pinecone [25] and FAISS [26] are libraries specifically designed to implement such search methods. They offer various options for index construction and search algorithms, allowing you to adjust parameters to achieve optimal performance and accuracy of results. The use of such frameworks makes it possible to accelerate the process of searching for nearby objects and to increase the efficiency of computing operations when working with large volumes of data. Due to the variety of available methods and the ability to adjust parameters, users can choose the optimal solution depending on the specific task and requirements for computational efficiency. To implement our system, we used the FAISS [26] library. It contains implementations of the most popular approximate vector algorithms. The FAISS [26] library is a group of methods based on the idea of Locality-sensitive hashing (LSH). The idea of such hashing is quite simple and is based on the existence of a hash function that can convert similar objects into a single centroid. In this way, we can quickly localize all similar objects in the index and not waste computing resources searching for those that will be defined as irrelevant in the future. The FAISS [26] library offers many modifications of index construction, aimed at increasing performance and preserving the accuracy of vector search for collections of different sizes. In addition, the framework authors have also developed solutions for cases where the index cannot be fully loaded into RAM and other situations that can occur when working with large amounts of data. We chose the IVFPQ [27] algorithm which uses inverted index and quantification (effective vector compression). In this algorithm the vectors of images from the collection are clustered into N clusters and associates each vector in the base with some cluster. Then, for each cluster, the centroid is calculated. During the search, we find *nprobe* nearest cluster centers and then iterate over only those vectors that are inside these clusters. The index type was chosen based on the assumption of the collection size. It is worth noting that the method described in this section is suitable for images of different visual structure.

C. Image Similarity

At this stage we perform an accurate comparison, leaving the most likely candidates out of a small number. Our research has shown that, at this stage, it is important to consider the features of the images being processed. Thus, one of the most famous and common approaches to this problem is the use of Siamese Neural Networks [28]. This solution is great for the case in which the collection contains a large number of images with different visual structure [29]. An example of such a collection would be a collection of images created without using automatic drawing systems. However, if all the images in the collection have a similar structure, using Siamese Neural Networks [28] does not provide the desired system performance. An example of such a case is a collection of handwritten texts represented as images. Our research has shown that these types of collections are more effectively solved if treating the problem as a modification of Image Matching task. Such approaches can be summarized as follows. The main idea of the keypoint-extraction-based approach was to use different keypoint extractor algorithms (SIFT [12], ORB [13] or a neural network) to select keypoints and compute their descriptors for each image from the collection and each query image. A descriptor is an object that contains some information about a key point on the basis of which it is possible to infer a similarity or difference between two key points. Similarity is defined as the Euclidean distance between the corresponding key point descriptors. We look for the closest and second closest keypoint descriptors in the candidate image for every keypoint descriptor in the query image. Let us denote the corresponding distances as d_1 and d_2. Two keypoints (one from query image and its closest one from the candidate image) make matches of two types:

- type 1 if $d_1 < \theta_1$
- type 2 if $d_1 < d_2\theta_2$

for fixed thresholds θ_1, θ_2. For each match type we compute similarity between two images as a portion of keypoints in a query image that have a match. Finally, a query image is considered a reuse if both values of similarities exceed some fixed thresholds τ_1, τ_2.

There are many algorithms for solving the problem of finding key points, both in classical computer vision and neural networks. The first group includes such algorithms as SIFT [12], ORB [13]. The second group includes the implementations described in the works [30], [31]. Unlike the majority of computer vision tasks, the gap in quality between classical computer vision solutions and deep learning methods for image matching is not very impressive [32]. Until recently, most neural network approaches to key point matching required two separate architectures. One neural network was just detecting the key points, and then the other was comparing them. However, a few years ago, with the advent of Transformer-type architectures into computer vision, solutions emerged that combined these two stages. Lately Transformers became a popular solution in a wide range of Computer Vision problems. For example, PoseFormer [33] and TransPose [34] for pose estimation task, DETR for object detection [35], Visual Transformer [36] for classification task, ViT [37] for image recognition and so on. One of the tasks where such architectures showed high efficiency is Image Matching. Authors of LoFTR [38] represented a novel architecture based on Transformer which became a new SOTA for Image Matching task. Inspired by their work, we modified LoFTR [38] architecture to meet the specific requirements of our task. Our research has shown that the most effective solution is to use Transformer-based architectures.

V. Experiment

Experiments were carried out for two collections: images created without the use of automatic drawing systems and a collection of handwritten texts presented as images.

A. Search

This section presents the results of experiments on the quality and performance of the designed image reuse retrieval system.

B. Metrics

Recall@K and FalsePositiveRate(FPR) functions are used to evaluate the system performance.

$$Recall@1 = \frac{TP}{TP + FN}, \tag{6}$$

$$FPR = \frac{FP}{FP + TN}, \tag{7}$$

FPR is the ratio of the number of false positives to the sum of the number of false positives and correctly classified as false samples. The choice of these functions is determined by the specificity of the task of detecting violations of academic ethics: in addition to high quality search, it is necessary to minimize the proportion of false positives of the system. Based on this principle, the FPR target for industrial image retrieval is low.

C. Images created without automatic drawing systems

We used open source Internet data for the experiment. The bulk of the data are academic papers, such as graduate qualifications and theses, as well as scientific articles and literature. At the time of the experiment, the collection contains about 59 million. images. For this experiment we formed three groups of test sets. The first one contained 5,000 images that had been indexed in the system at the time of search. These images did not undergo any further changes. The second group contained the same 5,000 images, which underwent minor changes. We will consider minor changes as:

- Image crop within 5% of original size.
- 25% resize of image.
- Grayscale.
- Gaussian Blur.

The third sub-sample consists of the same 5,000 images that have undergone a more significant transformation. These include:

- Image crop within 15% of original size.
- 50% resize of image.
- 180 degree rotation.

The augmentations were implemented using the Pillow package [39].

D. Quality study

The quality of the system was measured using three sub-samples: unchanged, with minor changes and with more serious augmentations. The results are shown in the Table I.

TABLE I. QUALITY STUDY RESULTS FOR IMAGES CREATED WITHOUT AUTOMATIC DRAWING SYSTEMS

Metrics	No changes	Minor changes	Significant changes
Recall@1	100%	98%	52.5%
FPR	0%	7.5%	21.5%

E. Search latency

Performance was measured on a virtual machine with 8 cores of the 128 x AMD EPYC 7532 32-Core Processor 2.3 GHz processor and 32 GB RAM. To estimate the performance we used 5,000 images included in the collection and 1,000 images which are not in the collection. The performance of the system is about 0.32 seconds per query. As the search pipeline almost doesn't vary for different domains of images, the obtained latency estimation can be considered as relevant for collection of images of handwritten essays.

We suppose the obtained quality and latency on a real-life dataset is high enough to establish the efficiency of the proposed design for collection of images created without automatic drawing systems.

F. Images of handwritten essays

The system was evaluated using the HWR200 [40], which was specialized for the problem of finding duplicate manuscripts and handwritten works text reuse. The basis of the dataset is 35 different unique texts further referred to as originals. They are used to generate most of the dataset: texts further referred to as reuses. The reuses consist of two types of sentences: sentences that appeared in original texts and unique sentences. In total, 2650 reuses are generated. In addition, there are 35 more unique texts further referred to as fprs. Each handwritten text in the dataset was translated into the image format in three different ways: scanning, photography in good light without objects in the frame, and photography in poor light. All three images required that the text be fully visible in the frame. An example of all three types of images is presented in the Fig. 2.

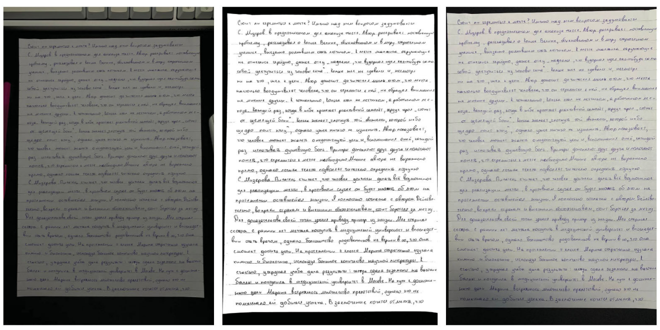

Fig. 2. Three types of images: (left) photographed in poor light and with other objects, (center) scanned, (right) photographed in good light and without other objects. [40]

G. Quality study

For this experiment we used two types of images from the HWR200 [40] dataset: dark and light photos. Recall@1 and FPR were calculated as mean Recall@1 and FPR for two cases. The first case is when collection consisted of light photos and query images were dark photos and the second case was vice versa. The results are shown in the Table II. The results of our experiment show

TABLE II. QUALITY STUDY RESULTS FOR IMAGES OF HANDWRITTEN ESSAYS

Metrics	Results
Recall@1	59%
FPR	5.5%

that although images of handwritten essays significantly differ from images created without automatic drawing systems, the developed method still performs quite well. The quality of the proposed approach may be improved by introducing additional pre- and postprocessing stages, specifically designed for the domain.

VI. DISCUSSION

The experiments show that our approach can be used as an industrial solution for image reuse detection task as it shows decent quality and latency of search for big collections of objects. Our implementation of the system reaches Recall@1 = 98% quality for the most wide-spread ways of image editing for image reuse in academic field and Recall@1 = 52.5% for less common ones. As the experiment was held using only one image processing library, the results might vary depending on the augmentation tool used. This issue can be addressed by finetuning separate parts of the system. The results obtained for the collection of images of handwritten essays show that our method is applicable for different domains of images. To increase the quality of our approach for some specific domain of data it may be necessary to add extra pre- and postprocessing stages to the general pipeline. For existing antiplagiarism solutions one of the crucial features is a report which shows why exactly an image is considered to be a reuse. Highlighting similar fragments of query and source is necessary and leaves field for future work. It would also be interesting to estimate how more complicated augmentations as watermarking [41] would influence the results of the experiment.

VII. CONCLUSION

In this work we introduce a design of an image reuse detection system which would achieve high performance on large document collections. The system solves the problem of finding reused images in several stages. The first stage is the construction of the image embedding. The second stage, candidate search, involves a search for objects that narrows down the number of possible sources of reuse.

The final stage is an accurate comparison, leaving a small number of candidates which are considered likely to be the source of the reuse. This approach allows not only to work with images of different structure, but also to has proved to be effective for the image reuse detection task on large collections of objects. We conducted experiments for collections containing images of two types: images created without the use of automatic drawing systems and a collection of handwritten texts presented as images. We obtained 98% quality for images created without the use of automatic drawing systems, 59% quality for images of handwritten essays and latency about 0.32 seconds per query for the collection of 59 million objects. The results suggest that the proposed system can be used as an industrial solution.

References

[1] O. Bakhteev, A. Ogaltsov, A. Khazov, K. Safin, and R. Kuznetsova, “Crosslang: the system of cross-lingual plagiarism detection,” in Workshop on Document Intelligence at NeurIPS 2019, 2019.

[2] K. Avetisyan, G. Gritsay, and A. Grabovoy, “Cross-lingual plagiarism detection: Two are better than one,” Programming and Computer Software, vol. 49, no. 4, pp. 346–354, 2023.

[3] M. V. Kuznetsova, O. Y. Bakhteev, and Y. V. Chekhovich, “Methods of cross-lingual text reuse detection in large textual collections,” Informatika I Ee Primeneniya [Informatics and Its Applications], vol. 15, no. 1, pp. 30–41, 2021.

[4] G. Gritsay, A. V. Grabovoy, A. S. Kildyakov, and Y. V. Chekhovich, “Artificially generated text fragments search in academic documents,” Doklady Rossijskoj akademii nauk. Matematika, informatika, processy upravleniâ, vol. 514, no. 2, pp. 308–317, 2023.

[5] G. Gritsay, A. Grabovoy, and Y. Chekhovich, “Automatic detection of machine generated texts: Need more tokens,” in 2022 Ivannikov Memorial Workshop (IVMEM). IEEE, 2022, pp. 20–26.

[6] H. Shen, “Meet this super-spotter of duplicated images in science papers,” Nature, vol. 581, pp. 132–136, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218605470

[7] K. F., “Image manipulation and image plagiarism – what’s fine and what’s not?” ndian J Dermatol Venereol Leprol, vol. 83, pp. 519–521, 2017.

[8] S. Srivastava, P. Mukherjee, and B. Lall, “implag: Detecting image plagiarism using hierarchical near duplicate retrieval,” 2015 Annual IEEE India Conference (INDICON), pp. 1–6, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:22244017

[9] P. Hurtík and P. Hodáková, “Ftip: A tool for an image plagiarism detection,” 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR), pp. 42–47, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:2776165

[10] S. Akshay, B. Chaitanya, and R. Kumar, “Image plagiarism detection using compressed images,” IJITEE, vol. 8, pp. 1423–1426, 2019.

[11] H. Huang, W. Guo, and Y. Zhang, “Detection of copy-move forgery in digital images using sift algorithm,” in 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, vol. 2. IEEE, 2008, pp. 272–276.

[12] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in 2011 International conference on computer vision. Ieee, 2011, pp. 2564–2571.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346–359, 2008, similarity Matching in Computer Vision and Multimedia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314207001555

[15] Y. Ke and R. Sukthankar, “Pca-sift: a more distinctive representation for local image descriptors,” in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., vol. 2, 2004, pp. II–II.

[16] A. Alahi, R. Ortiz, and P. Vandergheynst, “Freak: Fast retina keypoint,” in 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 510–517.

[17] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, “Kaze features,” in Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12. Springer, 2012, pp. 214–227.

[18] K. Zhao, H. Lu, and J. Mei, “Locality preserving hashing,” in AAAI Conference on Artificial Intelligence, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:11130464

[19] I. Perfilieva, P. Hodáková, and P. Hurtík, “Differentiation by the f-transform and application to edge detection,” Fuzzy Sets Syst., vol. 288, no. C, p. 96–114, apr 2016. [Online]. Available: https://doi.org/10.1016/j.fss.2014.12.013

[20] 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,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1386–1393.

[21] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 1, 2005, pp. 886–893 vol. 1.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[23] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, “Learning local feature descriptors with triplets and shallow convolutional neural networks.” in Bmvc, vol. 1, no. 2, 2016, p. 3.

[24] “Annoy,” Available at: https://github.com/spotify/annoy.

[25] “Pinecone,” Available at: https://github.com/pinecone-io.

[26] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, “The faiss library,” arXiv preprint arXiv:2401.08281, 2024.

[27] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with GPUs,” IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535–547, 2019.

[28] I. Melekhov, J. Kannala, and E. Rahtu, “Siamese network features for image matching,” in 2016 23rd International Conference on Pattern Recognition, ICPR 2016. United States: IEEE, 2016, pp. 378–383, international Conference on Pattern Recognition, ICPR ; Conference date: 04-12-2016 Through 08-12-2016.

[29] O. Bakhteev, Y. Chekhovich, E. Finogeev, T. Gorlenko, M. Kaprielova, A. Kildyakov, and A. Ogaltsov, “Image reuse detection in large-scale document scientific collection,” Concurrent Sessions 12, p. 107, 2021.

[30] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2018, pp. 224–236.

[31] A. Barroso-Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, “Key. net: Keypoint detection by handcrafted and learned cnn filters,” in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 5836–5844.

[32] D. Mishkin, “Local features: from paper to practice,” in CVPR. Computer Vision and Pattern Recognition 2020, 2020.

[33] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, “3d human pose estimation with spatial and temporal transformers,” 2021. [Online]. Available: https://arxiv.org/abs/2103.10455

[34] S. Yang, Z. Quan, M. Nie, and W. Yang, “Transpose: Keypoint localization via transformer,” 2020. [Online]. Available: https://arxiv.org/abs/2012.14214

[35] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and

S. Zagoruyko, “End-to-end object detection with transformers,” 2020. [Online]. Available: https://arxiv.org/abs/2005.12872

[36] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, “Visual transformers: Token-based image representation and processing for computer vision,” 2020. [Online]. Available: https://arxiv.org/abs/2006.03677

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[38] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, “LoFTR: Detector-free local feature matching with transformers,” CVPR, 2021.

[39] A. Clark, “Pillow (pil fork) documentation,” 2015. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf

[40] I. Potyashin, M. Kaprielova, Y. Chekhovich, A. Kildyakov, T. Seil, E. Finogeev, and A. Grabovoy, “Hwr200: New open access dataset of handwritten texts images in russian,” COMPUTATIONAL LINGUISTICS AND INTELLECTUAL TECHNOLOGIES”, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:267505062

[41] V. Gorbachev, E. Kaynarova, A. Makarov, and E. Yakovleva, “Digital image watermarking using dwt basis matrices,” in 2017 21st Conference of Open Innovations Association (FRUCT), 2017, pp. 127–133.