Scalable Image Infringement Evaluation and Detection System

1st Zherui Zhang
Department of Computer Science and Enginnering
Southern University of Science and Technology
Shenzhen, China
Student ID: 12133186

2th Luyan Yang
Institute for Quantum Science and Engineering
Southern University of Science and Technology
Shenzhen, China
Student ID: 12132862

***Abstract*—With the popularity of social networks and e-commerce, Internet media data is increasing at an alarming rate every day, and the value of images as an important content-bearing medium continues to increase, but with this comes increasingly frequent image infringement cases.**

This assignment focuses on researching image retrieval algorithms based on perceptual hashing, and has completed a set of image infringement detection system. The main tasks are as follows: 1. Defining infringing image problems, analyzing image retrieval technology; 2. Researching perceptual hashing algorithm, test results It shows that the perceptual hash algorithm has the problem of weak robustness in image retrieval, and it is difficult to deal with geometric tampering such as scale transformation; 3. Research on the perceptual hash algorithm based on SIFT, and obtain image perception by compressing image SIFT feature data. Hopefully, the robustness is improved; 4. In order to improve the search accuracy, ORB matching is added as the secondary search module. 5. The Faiss library is constructed to optimize the search speed.

Experimental results show that the system can quickly detect image infringement, and has certain advantages in robustness and scalability. The system can detect image tampering features such as image addition, blur processing, perspective transformation, rotation transformation, and image cutting. It has realized the retrieval of 10,000 pictures in 0.001s, and the use of this system can protect the interests of the owner of the content copyright.

***Index Terms*—infringement detection, similarity search**

I. Introduction

A. Background and Motivation

In 2014, Internet users reached 2.4 billion. By 2016, this number had grown to 3.4 billion, with an increase of 300 million Internet users in 2017. As of June 2019, there are now more than 4.4 billion Internet users. In just five years, the number of people using the Internet has increased 83 percent, and the increase in Internet users also means that a large amount of data is generated. It is estimated that by 2025, 463 EB of data will be created every day in the world.

People's lives in the information age are closely related to the Internet. With the continuous popularization of the Internet, various types of original design works have a certain degree of digitalization. Many of them are directly expressed in digital forms. The Internet has become the most important way for design creators to circulate digital products. Efficient medium. However, while the Internet provides convenience for circulation, it also reduces the difficulty of infringement, and digital copyright disputes follow. Digital copyright infringement forms are very diverse, concealed, and difficult to monitor. Digital products are naturally reproducible, easy to tamper with, and easy to embezzle. Infringers can achieve the purpose of stealing the creator's picture works by cutting, zooming, element swapping, rotating, adding and deleting watermarks, and splicing.

This system is oriented to the copyright protection of original picture works, and aims to solve the problem of picture infringement detection in actual scenes. There are two main technical challenges currently faced:

In reality, there are many types of original pictures, and there are many ways to infringe on the editing and deformation of pictures. How to identify various infringements? There are a large number of pictures on the Internet. How to find infringing pictures accurately and quickly?

In response to the above technical challenges, this work proposes an accurate and efficient image infringement detection algorithm based on perceptual hash, which effectively solves the problem of multiple types of image infringement methods in real scenes and realizes rapid detection.

B. Common Ways of Infringement and Tampering

There are mainly 5 common ways to change and modify:

Translation transformation refers to the translation of the target composition element in any direction to achieve the effect of changing the spatial position relationship between the composition elements.

Rotation transformation refers to the 0-360 degree rotation of the target composition element around a certain base point, which causes the angle of the target composition element to be changed.

Scale transformation, in the scene where the camera is used to shoot the object, the camera is far away from the object, and the photo presents more overview information of the object, which is more blurred; on the contrary, the photo presents more detailed information of the object and is clearer. The effect of scale transformation is similar to the above phenomenon, large-scale description of vague overview information, small-scale description of clear detailed information. In the implementation of the specific algorithm, the former is

completed by fuzzy processing, and the latter is completed by interpolation.

Angle change, when we observe an object, from the front, it is the front view, and from the side, it is the side view. Based on common sense, the image of the object perceived by the front view and the measurement view is different. The perspective change is to achieve this effect through algorithm simulation. Changing the perspective of the target composition element will cause the modified composition element we perceive to be very different from the element perceived at the original angle.

Masking modification refers to adding some masks to the original image to partially or entirely cover the original image to achieve the goal of obtaining the difference between the original image and the infringing image, thereby avoiding infringement detection.

II. Image retrieval technology

A. Overview

Image retrieval can be divided into two categories according to the different ways of describing image content, one is TBIR (Text Based Image Retrieval), and the other is CBIR (Content Based Image Retrieval).

The text-based image retrieval method started in the 1970s. It uses text annotation to describe the content of the image, thereby forming keywords for each image to describe the content of the image, such as the objects and scenes in the image. , This method can be manual labeling, or semi-automatic labeling through image recognition technology. When searching, users can provide query keywords according to their own interests. The retrieval system finds out those pictures marked with the query keywords according to the query keywords provided by the user, and finally returns the query results to the user.

Content-based image retrieval, which uses a computer to analyze the image, establishes the image feature vector description and saves it in the image feature library. When the user inputs a query image, the same feature extraction method is used to extract the features of the query image to obtain the query vector. Then calculate the similarity between the query vector and each feature in the feature library under a certain similarity measurement criterion, and finally sort and output the corresponding pictures according to the similarity. This task is to build content-based image retrieval.

Content-based image retrieval technology has broad application prospects in industrial fields such as e-commerce, leather cloth, copyright protection, medical diagnosis, public safety, and street view maps. In terms of infringement detection, taking the leather textile industry as an example, leather fabric manufacturers can take samples into pictures. When clothing manufacturers need leather fabrics with a certain texture, they can search for the same or similar leather fabrics in the database, so that The copyright management of leather fabric samples is more convenient; in terms of copyright protection, service providers that provide copyright protection can apply image retrieval technology to verify whether the trademark has been registered or not.

B. Large-scale infringement image retrieval

Infringing image retrieval refers to inputting a picture and finding an image similar to the picture from the image database. An infringement detection system that can be used on a large scale should solve the three main problems of large image data, high feature dimensions, and short-time response.

(1) The amount of image data is large. Thanks to the development of multimedia information capture, transmission, storage and the improvement of computer computing speed, after more than ten years of development in content-based image retrieval technology, the scope of the applicable image scale has also been expanded from the original small image library to large-scale The image library is even a massive image data set, so image retrieval should meet the requirements of the big data era, and it should be scalable on large-scale image data sets.

(2) The feature dimension is high. The image feature is the cornerstone of directly describing the visual content of the image, and the quality of its feature expression directly determines the highest possible retrieval accuracy in the retrieval process. At the beginning of feature extraction, those relatively high-level features should be selected. If the local feature expression is also regarded as a kind of "high-dimensional", then the description ability of the feature has a greater correlation with the dimensionality of the feature. Therefore, large-scale image retrieval has obvious characteristics of high feature dimensionality in terms of feature description. Therefore, another typical feature of large-scale image data set retrieval is the high dimension of image feature description vector.

(3) Fast response speed is required. For user queries, the image retrieval system should have the ability to respond quickly to user queries. At the same time, due to the large amount of large-scale image data and high feature dimensions, it is difficult to directly use Brute Search indexing strategies (also called linear scanning) to satisfy the system. Real-time requirements, such as the average time spent on each query on the Oxford University building image data set. It can be seen that in the Oxford University building image set with only 4063 images, the query time is 100 words It takes about 1 second when the rearrangement depth is 1000, and the entire program is still running on a high-profile server. Therefore, large-scale image retrieval needs to solve the problem of real-time response of the system.

III. Key algorithms

A. ORB Feature Matching

Feature point extraction and matching is a very important link in computer vision. ORB is the abbreviation of Oriented Fast and Rotated Brief, which can be used to quickly create feature vectors for key points in the image. These feature vectors can be used to identify objects in the image.

Among them, FAST and BRIEF are feature detection algorithm and vector creation algorithm respectively. ORB will

first look for special areas in the image, called key points. Key points are small areas that stand out in the image, such as corner points. For example, they have the characteristic that the pixel value changes sharply from light to dark. Then the ORB will calculate the corresponding feature vector for each key point. The feature vector created by the ORB algorithm contains only 1 and 0, which is called a binary feature vector. The order of 1 and 0 will vary depending on the specific key point and the pixel area around it. This vector represents the intensity pattern around the key point, so multiple feature vectors can be used to identify a larger area or even a specific object in the image.

The first step of ORB feature detection is to find the key points in the image. In this step, the FAST algorithm is used.FAST is the abbreviation of Features from Accelerated Segments Test. You can quickly select key points. The algorithm steps are as follows:

1) Determine the value of the threshold parameter h of the selected feature point.
2) For any pixel p on the image, FAST compares the 16 pixels in the circle with the point p as the center. If the gray value of the circle is less than lp-h (lp is the gray value of p If there are more than 8 pixels in total, or the gray value is greater than lp + h, the pixel p is selected as the key point.

The reason FAST is so efficient is that it only compares p to the 4 equally spaced pixels in the circle. This method has proven to be the same as comparing 16 surrounding pixels. If there is at least one pair of consecutive pixels whose grayscale is higher than lp + h or lower than lp-h, then p is selected as the key point. This optimization shortens the time to search for key points in the entire image by a factor of four.

BRIEF is the abbreviation of Binary Robust Independent Elementary Features. Its function is to create a binary feature vector based on a set of key points, also known as a binary feature descriptor, which is a feature vector containing only 1 and 0. Each key point in BRIEF is described by a binary feature vector, which is generally a string of 128-512 bits, which contains only 1 and 0.

The biggest advantage of the BRIEF algorithm to generate binary feature descriptors is that they can be stored in memory very efficiently and can be calculated quickly, and it can run on devices with very limited computing resources (such as smart phones).

The specific steps of the BRIEF algorithm to generate feature descriptors are as follows:

1) Use Gaussian to smooth the given image to prevent the descriptor from being too sensitive to high-frequency noise.
2) Then for a given key point, a pixel is extracted from the Gaussian distribution centered on the key point, and this point is called the number one point of the key point below, the standard deviation is σ.
3) A pixel is extracted from the Gaussian distribution centered on the first point. This point is called the second point of the key point, and the standard deviation is $\sigma/2$, which is because of experience shows that this choice improves the feature matching rate.
4) Construct a binary feature descriptor for the key points by comparing the grayscale values of the first and second points obtained in (2) and (3). If point one is lighter than point two, the value 1 is assigned to the corresponding bit in the descriptor, otherwise the value 0 is assigned.
5) Then select new No. 1 and No. 2 points for the same key point, compare their gray levels and assign 1 or 0 to the next bit in the feature vector. (Jump to (2) repeat loop)
6) In order to generate feature descriptors with specific dimensions, the BRIEF algorithm will repeat (2) (4) corresponding times to generate feature descriptors of specified length and repeat the above algorithm for each feature point.

ORB is characterized by very fast speed, and to a certain extent it is not affected by noise and image transformations, such as rotation and scaling transformations.

B. Perceptual Hash

pHash (perceptual hash) is also known as perceptual hash. The perceptual hash value can represent the fingerprint of a picture. By comparing the fingerprints, it can be judged whether two pictures are similar.

The basic steps of Phash work:

1) **Rescale the picture**
   32 * 32 is a better size, which is convenient for DCT calculation
2) **Convert to grayscale image**
   Convert the zoomed image into a 256-level grayscale image.
3) **Calculate the DCT**
   The collection of the rate at which DCT separates the image into components
4) **Reduce DCT**
   The matrix after DCT calculation is 32 * 32, and 8 * 8 in the upper left corner is retained, which represents the lowest frequency of the picture.
5) **Calculate the average value**
   Calculate the average value of all pixels after the DCT is reduced.
6) **Further reduce the DCT**
   Record as 1 if greater than the average value, otherwise record as 0.
7) **Obtain information fingerprint**
   Combine 64 information bits, the sequence is arbitrary to maintain consistency.

The basic principle of pHash is to reduce the picture to a range that can be calculated, and then filter the main features of the image through the DCT algorithm to obtain data that can reflect the characteristics of the picture to a certain extent, and finally output the hash value of the picture. It is understood as a picture fingerprint, and then the similarity of multiple

pictures is calculated through the contrast difference of the hash value. The method often used is the Hamming distance.

C. Facebook AI similarity search

Faiss is an open source search library for clustering and similarity by the Facebook AI team. It provides efficient similarity search and clustering for dense vectors, supports search for billions of vectors, and is currently the most mature approximate nearest neighbor search library. It contains a variety of algorithms for searching vector sets of any size (note: the size of the vector set is determined by RAM memory), as well as supporting codes for algorithm evaluation and parameter adjustment. Faiss is written in C++ and provides a Python interface that perfectly connects with Numpy. In addition, GPU implementations are provided for some core algorithms.

Through the introduction of Faiss document, we can understand that the main function of Faiss is similarity search. As shown in the figure below, taking picture search as an example, the so-called similarity search is to find the K pictures most similar to the target I specified in a given bunch of pictures, which is also referred to as the KNN (K Nearest Neighbor) problem for short.

In order to solve the KNN problem, it is necessary to realize the storage of the existing gallery in the project. When the user specifies to retrieve the pictures, he needs to know how to find the most similar K pictures from the stored picture library. Based on this, we speculate that Faiss has the add function and search function in the application scenario, and the corresponding modification and deletion functions will follow. From the above analysis, Faiss is essentially a vector (vector) database.

For databases, time and space optimization are two eternal themes, that is, how to store more information with less space in storage, and how to search for more accurate information at a faster speed in search. How to reduce the time required for searching? The most common operation in the database is to add various indexes, and encapsulate the functions of various search algorithms or the strategy of changing space and time into various indexes to meet various reference scenarios. In this task, we used Faiss to build a class library to optimize retrieval speed.

D. Fast Library for Approximate Nearest Neighbors

The full name of the FLANN library is Fast Library for Approximate Nearest Neighbors, which is currently the most complete (approximate) nearest neighbor open source library. It not only implements a series of search algorithms, but also includes a mechanism for automatically selecting the fastest algorithm.

IV. Experiments and results

A. System structure design

This system adopts image retrieval technology of perceptual hash. The specific framework can be divided into five steps: feature extraction, hash coding, Hamming distance sorting, Judgement and ORB matching.

1) **Feature extraction**
   Use the ORB algorithm to extract the features of the images in the image database one by one, and add them to the feature database in a one-to-one correspondence between the image file name and the image feature. In order to make sure the efficiency of our system, we only use 40 key points for each image, with a 128-dimensional vector of feature descriptor for each feature point.
2) **Hash encoding**
   Use the Phash algorithm to get the hash values of the images in the image database one by one, and add them to the hash value database in a one-to-one correspondence between the image file name and the image hash values. The hash value we get for each image is 256 bit.
3) **Hamming distance sorting**
   In the Hamming distance sorting stage, for a given query image, the Hamming distance between the hash code corresponding to the query image and the other hash codes is calculated one by one, and then the similarity is sorted from small to large to get search result.
4) **Judgement**
   For the result smallest Hamming distance, we judge if it is over 80, which we determine by examining the experimental result, we consider this image can't be detected by Phash. Then we use ORB matching.
5) **ORB matching**
   In the ORB matching stage, for a given query image, we calculate the number of similar key points (similarity between the feature descriptors larger than 70%) to the other images one by one. Then we sort the number of similar feature points from large to small to determine the final detection result.

B. Dataset

We use the Totally-Looks-like dataset which is a dataset and benchmark challenging machine-learned representations to reproduce human perception of image similarity. The dataset contains in total 12032 images, and we choose the first 10000 images of it in order to standardize our experiment.

1) Test dataset generation: We use the open-source image augmentation library AugLy to generate our test dataset. Automatic transformations we used are classified into the following broad categories:

- **Overlays**
  Text and emoji
- **Color transformations**
  Changes in brightness or saturation, grayscale
- **Pixel-level transformations**
  Blur, random noise
- **Spatial transformations**
  Cropping, rotation, padding, aspect ratio change, perspective transformation

There are in total 12 types of image transformation methods. For each type of the transformation method, we randomly choose 10 images from the dataset of 10000 images, and perform corresponding transformation on it to get the query image. All in all, there are 120 images in the test dataset, and each one with a corresponding original image in the original dataset with 10000 images.

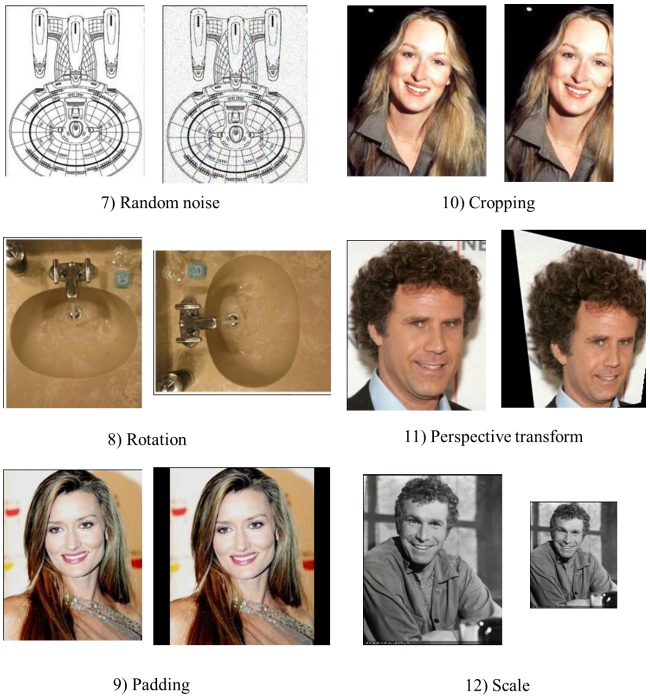
We show the example image of each image transformation method in Fig 1 and Fig 2.

1) Overlay emoji 4) Brightness

2) Overlay text 5) Saturation

3) Gray scale 6) Blur

Fig. 1. Example images of transformation 1 to 6.



7) Random noise 10) Cropping

8) Rotation 11) Perspective transform

9) Padding 12) Scale

Fig. 2. Example images of transformation 7 to 12.

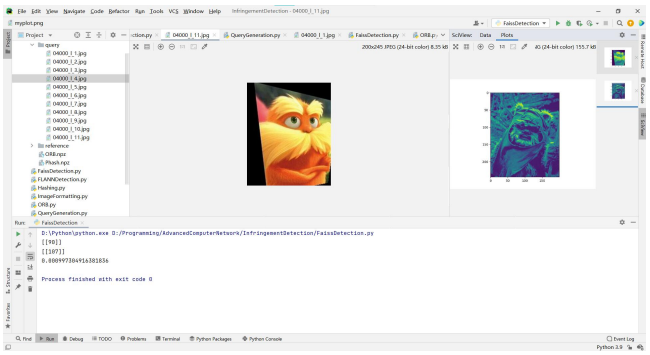


Fig. 3. System demo on Pyhcarm IDE.

C. Experimental results

The Phash module, ORB feature extraction module, FLANN module and FAISS feature matching module in this system implementation are implemented in C/C++ language and developed in PyCharm IDE; The overall process control of the system is realized by Python language with version 3.9.0.

1) Experimental configuration: The configuration of the machine running the experiment is as follows:

- CPU: Intel Core i5-9300H
- Memory: 16G
- Operating system: Windows 10

2) Demo description: The C-end rendering of the infringement detection system is shown in the figure. The C-end of our system is in the Pycharm IDE. The system will output the ID and the probability of an image in the original dataset that might face the problem of infringement by a query image. Our system demo is depicted in Fig 3 and Fig 4.

3) Final result: The final result of our system can detect a query image with the accuracy rate of 115/120 = 95.8%, with average detection time of 0.33s per image.

V. Conclusion

A. Advantages of traditional CV technology

This part will introduce in detail the reasons why traditional feature-based methods can effectively improve performance in

TABLE I
Final Result

	Phash	ORB	All	Time(s)
True	83	32	115	40.17
False	0	5	5	

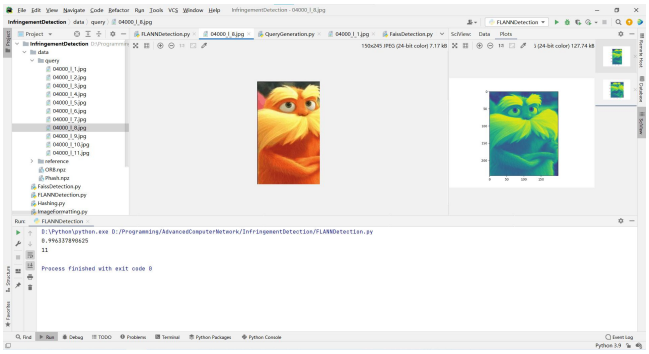


Fig. 4. Detect successfully on Pyhcarm IDE.

CV tasks. These traditional methods include: Scale Invariant Feature Transform, SIFT; Speeded Up Robust Feature, SURF; Features from Accelerated Segment Test, FAST; Hough transform; economic hashing.

Traditional CV technology is usually a general algorithm, and the features learned by deep neural networks are specific to training data. In other words, if there is a problem with the construction of the training data set, the network will not perform well on image processing other than the training data set.

Traditional CV technology is a simpler and faster technology when dealing with some problems. For example, classify two types of products on an assembly line conveyor belt, one is red and the other is blue. Deep neural networks need to collect sufficient training data first. However, the same effect can be achieved using a simple color threshold method.

Traditional CV technology has sufficient transparency, and people can judge whether the solution works effectively outside the training environment. When the training data set is limited, the neural network may be over-fitting and cannot be effectively generalized. Manual tuning is very difficult, because DNN has millions of parameters and the relationship between them is intricate. For this reason, deep learning models have been criticized as black boxes. CV engineers understand the problems their algorithms can be migrated to, so that if something goes wrong, they can perform parameter adjustments so that the algorithm can effectively process a large number of images.

Nowadays, traditional CV technologies are often used to solve simple problems, so that they can be deployed on low-cost microprocessors, or by highlighting specific features in data, enhancing data, or auxiliary data set annotations to limit the problems that deep learning technologies can solve. Later in this article, we will discuss how many image transformation techniques can be used in neural network training. Finally, there are many more challenging problems in the CV field, such as robotics, augmented reality, automatic panorama stitching, virtual reality, 3D modeling, motion estimation, video stabilization, motion capture, video processing, and scene understanding. These problems cannot be passed Deep learning is easy to implement, but it can benefit from traditional CV techniques.

There are some problems in the CV field, such as robotics, augmented reality, automatic panorama stitching, virtual reality, 3D modeling, motion estimation, video stabilization, motion capture, video processing, and scene understanding. They are difficult to differentiate in a differentiable way through deep learning. Easy to achieve, but requires the use of other ”traditional” technologies.

B. Advantages of Deep Learning

Deep learning can help traditional CV engineers achieve higher accuracy in tasks such as image classification, semantic segmentation, target detection, and simultaneous localization and mapping (SLAM). Since the neural network used in deep learning is trained rather than programmed, the application of this method requires less expert analysis and fine-tuning, and can handle the massive amount of available video data in the current system. Deep learning also has excellent flexibility, because for any use case, CNN models and frameworks can be retrained with a custom data set, which is different from the CV algorithm, which is more domain specific.

Deep learning can automatically extract features. Traditional CV methods are more troublesome for large-scale data. Selecting important features from each image is a necessary step. As the number of categories increases, feature extraction becomes more and more troublesome. Determining which characteristics best describe the different target categories depends on the judgment and long-term trial and error of the CV engineer. In addition, each feature definition also needs to deal with a large number of parameters, all parameters must be adjusted by the CV engineer. Deep learning introduces the concept of end-to-end learning, that is, each image in the image data set provided to the machine has been labeled with a target category. Therefore, the deep learning model is ”trained” based on the given data, in which the neural network finds the underlying pattern in the image category and automatically extracts the most descriptive and salient features for the target category.

C. Traditional CV + deep learning = better performance

There is a clear trade-off between traditional CV technology and deep learning methods. The classic CV algorithm is mature, transparent, and optimized for performance and energy efficiency; deep learning provides better accuracy and versatility, but consumes more computing resources.

The hybrid method combines traditional CV technology and deep learning, and has the advantages of both methods. They are especially suitable for high-performance systems that need to be implemented quickly.

The mixture of machine learning metrics and deep networks has become very popular because it can generate better models. The implementation of hybrid vision processing can bring performance advantages, and reduce the multiplication and accumulation operations to one-third of the 130-1000 of the deep learning method, and the frame rate is 10 times higher than that of the deep learning method. In addition, the hybrid

method uses only half of the memory bandwidth of the deep learning method, and consumes much less CPU resources.

REFERENCES

[1] The augly data augmentation library. https://github.com/facebookresearch/AugLy.

[2] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky (2014) Neural codes for image retrieval.In Proc. ECCV, pages 584–599. Springer

[3] M. Berman, H. J´egou, A. Vedaldi, I. Kokkinos, and M. Douze (2019) Multigrain: a unified image embedding for classes and instances. arXiv preprint arXiv:1902.05509

[4] Matthijs Douze and Giorgos Tolias and Ed Pizzi and Zoë Papakipos and Lowik Chanussot and Filip Radenovic and Tomas Jenicek and Maxim Maximov and Laura Leal-Taixé and Ismail Elezi and Ondřej Chum and Cristian Canton Ferrer. The 2021 Image Similarity Dataset and Challenge.

[5] Daniel Sáez Trigueros , Li Meng , Margaret Hartnett (2018) Face Recognition: From Traditional to Deep Learning Methods

[6] Nash W, Drummond T, Birbilis N (2018) A Review of Deep Learning in the Study of Materials Degradation. npj Mater Degrad 2:37. https://doi.org/10.1038/s41529-018-0058-x

[7] Bonaccorso G (2018) Machine Learning Algorithms Popular Algorithms for Data Science and Machine Learning, 2nd Edition. Packt Publishing Ltd

[8] Diligenti M, Roychowdhury S, Gori M (2017) Integrating Prior Knowledge into Deep Learning. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp 920–923

[9] Fernandez-Labrador C, Perez-Yus A, Lopez-Nicolas G, Guerrero JJ Layouts from Panoramic Images with Geometry and Deep Learning

[10] Wang J, Ma Y, Zhang L, Gao RX (2018) Deep learning for smart manufacturing: Methods and applications. J Manuf Syst 48:144–156. https://doi.org/10.1016/J.JMSY.2018.01.003

[11] Zheng L, Yang Y,Tian Q SIFT Meets CNN: A Decade Survey of Instance Retrieval

[12] AlDahoul N, Md Sabri AQ, Mansoor AM (2018) Real-Time Human Detection for Aerial Captured Video Sequences via Deep Models. Comput Intell Neurosci 2018:1–14. https://doi.org/10.1155/2018/1639561