Electronic Research Archive

ERA, 32(6): 4164–4180.
DOI: 10.3934/era.2024187
Received: 14 March 2024
Revised: 17 June 2024
Accepted: 19 June 2024
Published: 27 June 2024

http://www.aimspress.com/journal/ERA

Research article

An image filtering method for dataset production

Ling Li[1], Dan He[2,*] and Cheng Zhang[1]

[1] School of Computer Engineering, City Institute, Dalian University of Technology, Dalian 116600, China
[2] School of Business, Dalian University of Finance and Economics, Dalian 116600, China

* **Correspondence:** Email: hedanc@dlufe.edu.cn; Fax: +8613009424506.

Abstract: To address the issue of the lack of specialized data filtering algorithms for dataset production, we proposed an image filtering algorithm. Using feature fusion methods to improve discrete wavelet transform algorithm (DWT) and enhance the robustness of image feature extraction, a weighted hash algorithm was proposed to hash features to reduce the complexity and computational cost of feature comparison. To minimize the time cost of image filtering as much as possible, a fast distance calculation method was also proposed to calculate the similarity of images. The experimental results showed that compared with other advanced methods, the algorithm proposed in this paper had an average accuracy improvement of 3% and a speed improvement of at least 30%. Compared with traditional manual filtering methods, while ensuring accuracy, the filtering speed of a single image is increased from 9.9s to 0.01s, which has important application value for dataset production.

Keywords: DWT; hash; dataset; distance calculation

1. Introduction

Datasets play an important role in the field of image processing. The ImageNet [1] dataset, which was produced and released by Li Feifei's team in two and a half years in 2009, was the key to a new milestone in visual recognition technology. Since AlexNet [2] won the ImageNet large scale visual recognition challenge (ILSVRC) competition in 2012, far surpassing second place, the era of deep learning has arrived, and its application in the field of image processing is becoming increasingly widespread [3–6]. The effectiveness of neural network models is closely related to the scale and quality of training data, and the production of datasets has become an important factor in improving model

performance. Therefore, large-scale datasets such as MS COCO [7], VGG-Face [8], Places [9], etc., have also emerged. However, there is an inevitable problem, the production of the dataset is too time-consuming and labor-intensive.

In the process of creating datasets, there are two main methods: offline collection and online crawling. Offline collection is mostly based on shooting, but the acquisition channels for this method are limited. It is difficult for a team of only a few people to obtain tens of thousands of images in a short period through photography, and the demand for manpower is high. Web crawlers are currently the mainstream method, with the advantages of short time consumption, rich data, and considerable data scale. However, there are problems with complex data types, high repetitiveness, and a high proportion of bad data, which need to be screened manually according to the complex processing process, and the huge data scale undoubtedly reduces the efficiency and accuracy of data screening [10,11].

To reduce the time and labor costs of dataset production, researchers are also trying to use algorithms to assist in image filtering. For example, Deng and Cheng [12], Hua et al. [13] used the histogram method and perceptual hash algorithm to detect duplicate images in the production process of the dataset, respectively. However, the existing ordinary image similarity detection algorithms may not have strong specificity for data and cannot guarantee the accuracy of data filtering, which may lead to data waste or residual similar images, thereby reducing the quality of the dataset. Moreover, the filtering function is not perfect, and the data cleaning work in the early stage, such as checking the size, format, and number of channels, needs to be operated separately. At present, there is no specific algorithm for data cleaning and filtering during the dataset production process.

To solve the above problems, this paper proposes an image filtering algorithm based on the wavelet hash algorithm (wHash) for dataset generation, called image filter for dataset (IFD). Three major contributions of our works are:

1) DWT [14] was improved by objectively selecting wavelet basis functions to enhance the generalization performance of feature extraction, and enhance the expression ability of features through feature fusion.

2) A weighted hashing method was proposed to suppress the impact of noise on image hashing by assigning different weights to feature points and noise points.

3) Aiming at the specific scenario of dataset production, an image filtering algorithm is proposed, which greatly improves the speed of data filtering while ensuring detection accuracy, and has important application value for dataset production.

Furthermore, a distance measurement method that can reduce computational time overhead is proposed. The data processed by this algorithm can be used for subsequent annotation after simple rechecking, which greatly improves the efficiency of data filtering and realizes the automatic filtering of image data.

2. Related work

The function to be implemented in this paper is an image filtering algorithm for dataset production, which is based on image similarity detection. To obtain the similarity of the image, the central idea of image similarity calculation is to digitize the features of the image, and then choose appropriate methods to calculate the distance between features. There are also many works on image features, such as Chen et al. [15] proposed M^{3}FuNet to improve the extraction of hyperspectral image (HSI) spectral-spatial joint feature (FE). Traditional methods and convolutional neural network methods are the two

major classifications of image similarity calculation methods. With the development of convolutional neural networks, many image similarity detection tasks are also based on deep learning methods. Pinjarkar et al. [16] combined deep convolutional neural networks (DCNNs) with related feedback mechanisms to realize trademark image retrieval, which solves the semantic gap in the design and development of trademark retrieval system, and has achieved excellent results in accuracy and performance. Zeng et al. [17] proposed the concept of ontology semantic distance (OSD) based on the convolutional neural networks (CNN) model, and semantic tags are used to match the related images, which helps to collect and sort the data set of Dunhuang frescoes. Rajasenbagam and Jeyanthi [18] combined the Visual Geometry Group 16 algorithm (VGG16) with the K Nearest Neighbors algorithm (KNN) to create a method image retrieval system that can help radiologists quickly analyze computerized tomography (CT) images and generate treatment plans promptly. However, deep learning models require targeted training, and with different data categories, the model needs to be retrained, making it unsuitable for the work to be carried out in this paper. Therefore, traditional methods are used in this paper.

Traditional methods can be further divided into three categories. The first method is based on histograms. Aljanabi et al. [19] devised two new methods for image similarity and image recognition using information theory and joint histograms. To solve the problem of nonlinear radiation distortion and significant contrast difference in multi-modal remote sensing images (MRSI), Zhang et al. [20] proposed a new MRSI matching method — "histogram of the orientation of weighted phase" (HOWP). The second method is based on hashing, such as the average hash algorithm (aHash) and the perceptual hash algorithm (pHash). Drmic et al. [21] evaluated the robustness of the perceptual image hashing algorithm and illustrated that the image hashing algorithm is often used in image search and retrieval, looking for similar images, looking for a large number of repeated and near-repeated image sets, etc. The third method is based on feature extraction, such as scale-invariant feature transform (SIFT) [22] features and histogram of oriented gradient (HOG) [23] features. Sri et al. [24] proposed a SIFT algorithm to detect similarity between input images and calculate similarity scores for image matching. Naiemi et al. [25] proposed an improved HOG feature extraction method, and the HOG feature extraction method is used to improve the spam image optical feature recognition system. The histogram-based method is so simple that the accuracy of the calculated results will be significantly reduced when the image undergoes rotation and other deformations. The computational complexity of methods based on features such as SIFT and HOG is relatively high, and they do not have an advantage in processing time. So, we will focus our research on hash-based methods.

The image hash algorithm has a wide range of applications in the field of image processing, especially in image similarity detection [26–30]. By extracting image features and hashing them, complex image information is represented through dimensionality reduction [31,32]. Compared to pHash, wHash is not widely used. Because its computational complexity is relatively high, and pHash can also meet some of the requirements. However, due to the consideration of spatial information in the down-sampling process of DWT in wHash, it has better performance in image feature extraction and can ensure the accuracy of data filtering [33–35]. Furthermore, the selection of wavelet basis functions is somewhat difficult, and blindly down-sampling also leads to the loss of image features, so there is room for improvement, which is the direction of this paper's improvement.

3. IFD algorithm

When using crawlers to collect data, to ensure the richness and scale of the data, data are generally crawled from multiple sources, and with this comes the problems of data clutter, high repetitiveness, and junk data. The IFD algorithm aims to automate the image filtering part of the image dataset production process to replace manual filtering, reduce the time cost of image filtering, and improve the filtering quality. The algorithm flow is shown in Figure 1.

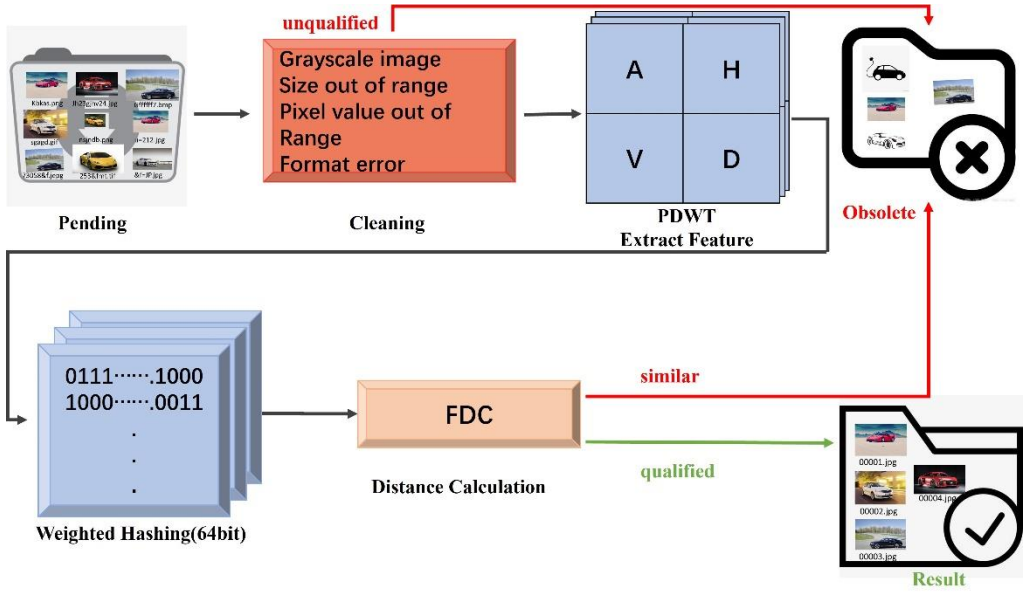

Figure 1. IFD algorithm process.

The processing of the method proposed in this paper mainly includes the following four parts. Firstly, perform personalized data cleaning. Customize the size, format, and pixel size range in the configuration file, and the algorithm can automatically filter out images that do not comply with the rules. Second, the improved discrete wavelet transform (PDWT) is used to extract features from the image. Adaptive selection of wavelet basis functions and fusion of shallow and deep features to enhance semantic information of features. Thirdly, a weighted hash algorithm is proposed to hash image features and reduce their complexity. Finally, the fast distance calculation algorithm (FDC) is used to calculate the distance between images, remove duplicate images, and realize automatic image filtering.

3.1. DWT

In machine learning based image processing algorithms, the discrete cosine transform algorithm (DCT) [36] is often chosen for image analysis, while the IFD algorithm proposed in this paper uses PDWT based on DWT. Both of DCT and DWT convert image signals from the spatial domain to the frequency domain and then decompose the frequency domain of the image into various sub-bands, analyzing each sub-band to obtain image information. The advantage of DWT is that the sub-bands preserve the spatial components of the image. Although the complex processing process may result in more time overhead, it has better performance and robustness. The formula for two-dimensional DWT

is as follows:

$$W_{\varphi}(0,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, \varphi_{0,m,n}(x,y), \tag{1}$$

$$W_{\psi}^{i}(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, \psi_{j,m,n}^{i}(x,y) \quad i = \{H,V,D\}. \tag{2}$$

Among them, $W_{\varphi}(0,m,n)$ is the w-frequency sub-band of the image, $W_{\varphi}^{i}(j,m,n)$ represents three high-frequency sub-bands in different directions, M,N are the length and width of the image, $\varphi(x,y)$ and $f(x,y)$ is the discrete form of the original image, $\varphi(x,y)$ and $\psi(x,y)$ are the scale and shift basis function.

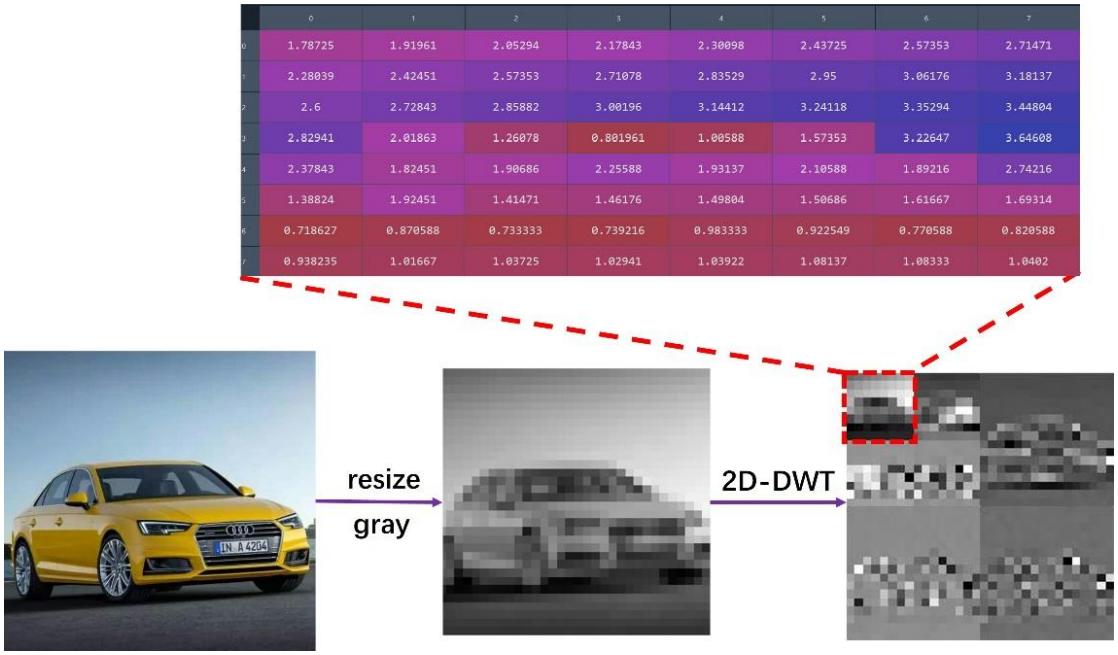

Figure 2. The process of feature extraction.

The process of extracting image features is shown in Figure 2. The original RGB image is used as input, and after resizing and grayscale processing, it is reduced to a size of 32 × 32 px to reduce computational complexity. We perform a two-layer two-dimensional discrete wavelet transform (2D-DWT) on the image, and after processing, the low-frequency component containing spatial information is in the lowest frequency sub-band of size 8 × 8 px in the upper left corner. Take "harr" as the wavelet basis function in Figure 2 as an example.

3.2. PDWT

The improvement of DWT by PDWT mainly lies in two aspects. One is to propose an objective method for selecting wavelet basis functions, to avoid the limitations of manual selection. The second is the introduction of feature fusion, which overlays the low-level features onto the high-level features, thereby improving the accuracy of the algorithm.

3.2.1. Selection of wavelet basis functions

Choosing the appropriate wavelet basis function can effectively analyze the characteristics of signals. On the contrary, if the selection of wavelet basis functions is incorrect, after applying wavelet transform to the signal, the projection coefficients of the signal on the wavelet function family are likely to overwhelm the characteristics of the signal.

In the discrete wavelet transform of images, there are six commonly used wavelet basis functions, as shown in Table 1.

Table 1. Commonly used wavelet basis functions.

Function	Orthogonality	Biorthogonality	Compactly supported	Symmetry
Harr	√	√	√	√
Daubechies	√	√	√	approximate
Symlets	√	√	√	approximate
Coifolets	√	√	√	approximate
Biorthogonal	×	√	√	×
ReverseBior	×	√	√	√

Two layers of discrete transformation were performed on the same image using 6 different wavelet functions, and the results are shown in Figure 3.

Figure 3. Transform results using different wavelet bases.

Abandoning the method of manually selecting wavelet bases, in PDWT, we adaptively select wavelet functions based on specific images. For the input dataset to be processed, PDWT randomly selects 1‰ images for processing such as noise addition, size change, and random cropping. After using the six wavelet basis functions mentioned above for DWT on these images, calculate the distance between features, and evaluate and analyze the calculation results. We choose the best-performing wavelet basis as the adaptive selection result for this dataset and use it for all image transformations. Although it may increase some time expenses, it has better robustness for feature extraction and enhances feature expression.

3.2.2. Feature fusion

The DWT of an image can also be understood as the down-sampling operation of the image, which uses a low-pass filter to extract the low-frequency components in the image, and this information precisely contains the features of the image.

To match the length of the hash sequence, sampling is usually stopped when the image is down-sampled to a size of 8 × 8 px. During the gradual down-sampling process, some key information may be lost, so we considered fusing the low-level features with the high-level features.

In PDWT, we performed two layers of discrete transformation on a 32 × 32 px image, up-sampling the obtained 8 × 8 px feature map, and fusing it with the 16 × 16 px features obtained from the first layer transformation. The fused image is then subjected to another discrete transformation to obtain the final result. The process is shown in Figure 4.

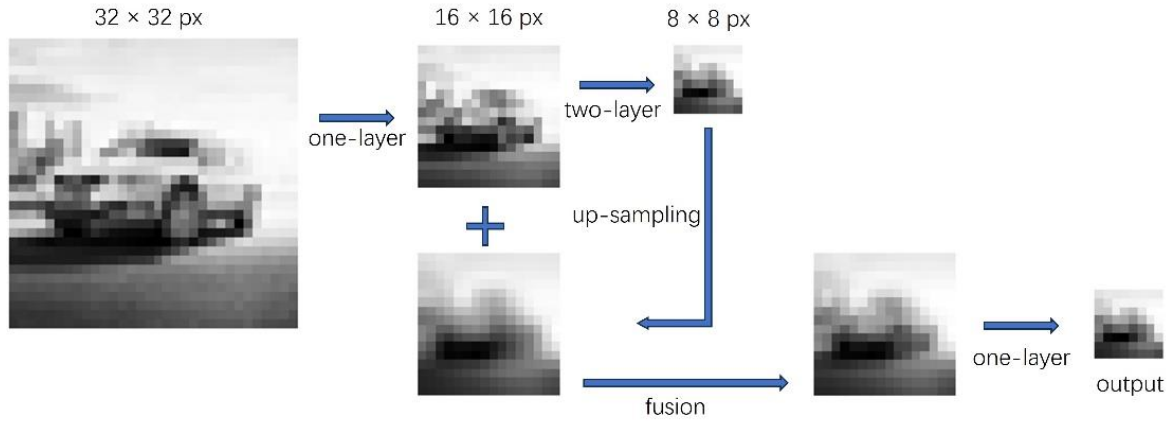

Figure 4. Schematic diagram of the feature fusion process.

3.3. Weighted hash

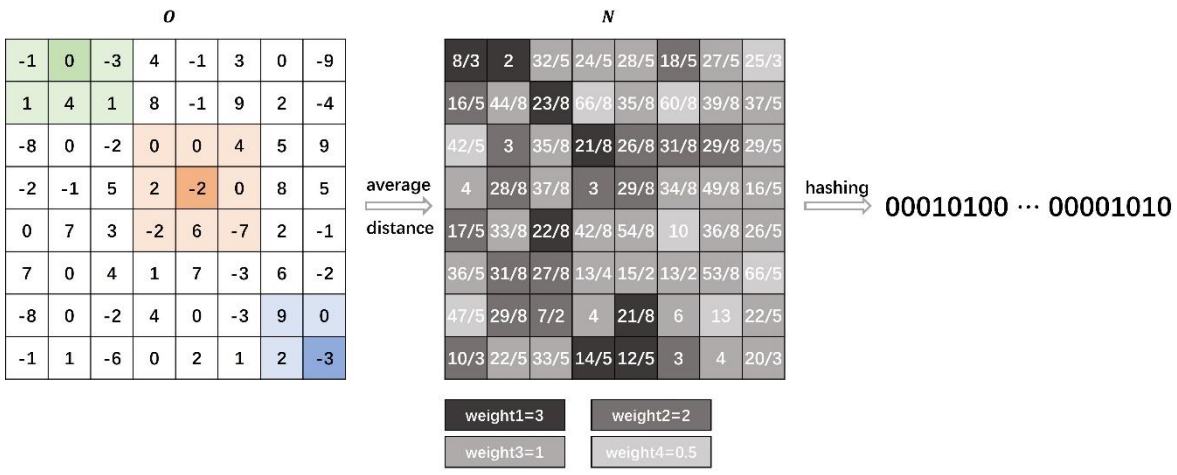

Figure 5. The calculation process of weighted hash.

In the hashing of image features, the method of average hashing is often used. Taking 64-bit data as an example, calculate the average value of the data, compare each bit of data with the average, mark

it as 1 if it is greater than the average, and mark it as 0 if it is less than the average, to obtain a hash sequence with a length of 64-bit. However, when there are extreme points in the image, the result of the average will be greatly affected. Therefore, this paper proposes a weighted hashing method, and its processing flow is shown in Figure 5.

Assuming the existence of a set of 64-bit data, calculate the average distance between each point and the surrounding data using the following formula.

$$N_j = \frac{\sum_{i=0}^{n-1}(|O_j - O_i|)}{n} \quad j \in [0,63], n \in [3,8]. \tag{3}$$

Among them, O_j represents one bit of the original 64-bit features, O_i is the point around O_j. The larger the average distance, the greater the difference between the point and the surrounding data, like high-frequency edge information. In hashing, more attention should be paid to low-frequency information. Therefore, assigning lower weights to points with larger average distances effectively avoids extreme points having a significant impact on the average. Sort N in ascending order, a divide the sorted data into 8, 16, 32, and 8 bits, assigning weights of 3, 2, 1, and 0.5, respectively, and then calculate the mean using the Eq (4).

$$\begin{gathered} mValue = \frac{\sum_{i=0}^{63} N_i \times weight_i}{weight1 \times 8 + weight2 \times 16 + weight3 \times 32 + weight4 \times 8}, \\ weight1 = 3, weight2 = 2, weight3 = 1, weight4 = 0.5 \end{gathered} \tag{4}$$

$$H_i = \begin{cases} 1, & N_i \geq mValue \\ 0, & N_i < mValue \end{cases}. \tag{5}$$

Finally, perform hashing according to Eq (5). Although this weighted hash may increase the computational cost slightly, it can make the hash sequence more expressive.

3.4. FDC

The Hamming distance [37–39] is usually used to represent the distance between two data and its formula is:

$$D(x, y) = \sum_{i=0}^{n-1} x_n \oplus y_n, \tag{6}$$

where x_n and y_n denote two n-bit strings and the $\oplus$ symbol indicates that the XOR operation is performed.

For 64-bit data, after exclusive OR (XOR), it takes 64 comparisons to obtain the distance value, while for large-scale image data, this is a significant time overhead. To reduce time overhead, the FDC is proposed in IFD, and its process is shown in Figure 6.

Figure 6 shows the process of FDC using 8-bit data as examples. XOR two equal-length data X and Y to obtain Z. When Z is not 0, loop through bit operations Z & (Z - 1) until it ends when Z is 0. The number of bit operations is the distance between X and Y. The distance between two data is the number of bit operations. In practical applications, after D++ operation, D will be compared with a pre-set threshold (usually set to 10). When the threshold is exceeded, it will be directly judged as a dissimilar image.

For 64-bit image hash values, it can greatly reduce the time cost of distance calculation, especially in the comparison of a large number of images, which can greatly improve the efficiency of image filtering.

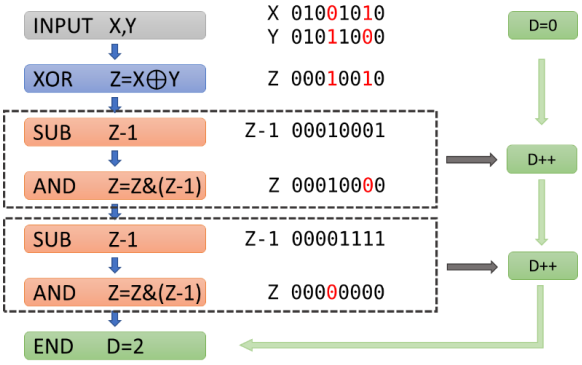

Figure 6. Schematic diagram of FDC.

4. Results

The experiment in this paper is mainly conducted from four aspects. The first aspect is to verify the effectiveness of feature fusion in the PDWT algorithm. Secondly, compared with other classic image feature extraction algorithms, evaluate them in terms of speed and accuracy. Thirdly, conduct component analysis experiments to verify the improved performance of the IFD algorithm. Fourthly, compare the time and accuracy of manually filtering images to verify the practical application effect of the IFD algorithm.

4.1. Experimental data preparation

To effectively evaluate the algorithm quantitatively and compare the tests, an experimental dataset with a known number of failed images and similar images was constructed. About 50,000 vehicle images were crawled from search engines by means of a web crawler as the raw data. 10,000 qualified images were manually selected as the base sample (named IFD-BASE). In addition, to prevent the impact of a single category of images on the experimental results, we also selected a total of 1000 images from different categories in the ImageNet dataset as the experimental subjects, which are referred to as IMGNET-BASE. All experiments in this paper will be based on these two datasets and meet the requirements of different comparative experiments by processing the images differently. The handling method is seen in Table 2.

Table 2. Experimental data processing.

Number	Methods	Parameter
1	Gray processing	-
2	Format conversion	png, bmp, tif, tiff
3	Size transformation	0.5, 0.8, 1.2, 1.4
4	Rotational transformation	10°, 20°, -10°, -20°
5	Gaussian	average value: 0 variance: 0.1
6	Poisson	-
7	Salt & Pepper	pollution level: 0.1
8	Speckle	variance: 0.04
9	Watermark	-

4.2. Comparison experiment of feature extraction

In the PDWT, we fused the feature map after the first layer discrete transformation (16 × 16 px) with the feature map after the second layer discrete transformation (8 × 8 px), referred to as PDWT_16_8 in the experiment. In this experiment, statistical feature-based evaluation methods were used to verify the effectiveness of feature fusion, including standard deviation (SD), average gradient (AG), and spatial frequency (SF). In addition, we also attempted to fuse the feature maps obtained from the second layer discrete transformation (8 × 8 px) with the feature maps obtained from the third layer discrete transformation (4 × 4 px), referred to as PDWT_8_4 in the experiment. The experimental results are shown in Table 3.

Table 3. Feature fusion evaluation results.

Performance metrics	DWT	PDWT_8_4	PDWT_16_8
SD	0.8	1.04	1.18
AG	0.503	0.389	0.587
SF	0.73	0.53	0.83

We randomly selected 200 images from the IFD-BASE dataset for the experiment, and the values in the table are the average results. From the results, PDWT_16_8 is significantly better than the other two methods in all three evaluation indicators. ForPDWT_8_4, due to the small size of the feature map, the fusion effect is significantly poor and inferior to the absence of the DWT algorithm.

4.3. Comparative experiment on distance calculation

In this section of the experiment, the PDWT algorithm was compared with DWT, feature extraction methods based on SIFT, and feature extraction methods based on HOG to demonstrate the performance of the algorithm.

4.3.1. Similarity comparison experiment

Due to the low texture complexity of car images and the fact that most of the main information is in the center of the image, a high-complexity landscape image was used in this section of the experiment. Perform the following five processing methods on the image: cropping, symmetry processing, adding Gaussian noise, adding watermark, and rotation, as shown in Figure 7. The experimental parameters are shown in Table 4.

Table 4. Experimental parameters.

Methods	Parameters
Cropping	0.6
Rotation	-15°
Gaussian	average value: 0 variance: 0.01

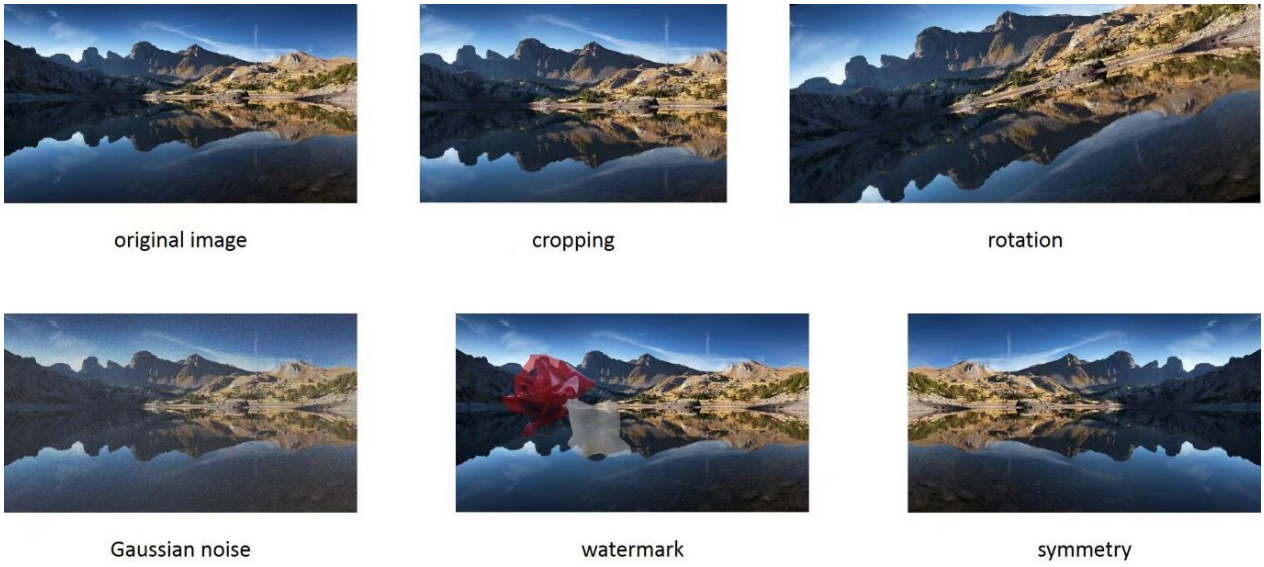

Figure 7. Image processing.

Using the PDWT+weighted hash+FDC algorithm and wHash, SIFT, and HOG methods, calculate the similarity between the original image and the processed image. The results are presented in percentage form, and the specific data is shown in Table 5. The experimental results show that the PDWT algorithm has good performance in similarity detection of the above five image transformations and has good robustness for various situations.

Table 5. Feature fusion evaluation results.

Processing category	Similarity (%)			
	wHash	IFD	SIFT	HOG
Cropping	97	91	20	83
Symmetry	44	56	67	77
Gaussian	92	95	100	82
Watermark	86	89	50	89
Rotation	72	78	0	74

4.3.2. Comparison experiment with existing methods

Randomly select 2000 images from the IFD dataset, use all the images in IMGNET-BASE, transform them according to Table 2, expanding them into IFD-5000 containing 5000 images, IFD-1000 containing 10,000 images and IMGNET-5000 containing 5000 images. The performance of the algorithm is evaluated from both speed and accuracy using classic methods–wHash, HOG, SIFT, and advanced deep learning method–VGG16.

Evaluate according to the evaluation in object detection. Data prediction generally has the following concepts: TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative). Precision refers to the proportion of all retained images that should truly be retained. The recall rate refers to the proportion of images that should be retained and have been retained. The calculation formulas for precision and recall are as follows:

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{7}$$

$$\text{Recall} = \frac{TP}{TP+FN}. \tag{8}$$

We use average accuracy (AP) as an evaluation metric for similarity detection accuracy, which is the area of the curves of precision × recall. The experimental results are shown in Table 6.

Table 6. Performance evaluation experiment results.

Method	Dataset	AP	Time (s)
wHash	IFD-5000	89.56	50.46
	IFD-10000	88.73	117.57
	IMGNET-5000	91.67	48.39
SIFT	IFD-5000	86.45	94.62
	IFD-10000	84.23	908.79
	IMGNET-5000	89.22	100.65
HOG	IFD-5000	93.80	48.09
	IFD-10000	93.49	450.16
	IMGNET-5000	93.08	50.53
VGG16	IFD-5000	93.16	159.43
	IFD-10000	93.70	328.99
	IMGNET-5000	94.09	149.08
IFD	IFD-5000	93.71	35.89
	IFD-10000	93.68	81.31
	IMGNET-5000	94.14	33.72

The experimental results show that the IFD algorithm has excellent performance in both speed and accuracy. Although the accuracy of HOG-based feature extraction algorithms is comparable to that of IFD, the advantage of IFD algorithms in detection speed becomes more apparent as the amount of data increases. Deep learning methods also have excellent performance in accuracy, but their detection speed is not as fast as the method proposed in this paper. This speed does not include the time for model training. Therefore, overall, the algorithm in this article has excellent filtering performance.

4.4. Component analysis experiment

To analyze the impact of various improvements on accuracy and speed in the IFD algorithm, we conducted component analysis experiments. Use IFD-5000 as the test dataset and wHash as the base. The specific experimental results are shown in Table 7.

The experimental results of the component analysis show that using the PDWT algorithm can effectively improve the accuracy of image similarity detection. Although weighted hashing leads to an increase in detection time, the FDC algorithm completely compensates for this overhead and has a significant speed improvement compared to the original distance calculation measurement method. This section of the experiment demonstrates the effectiveness of each component of the IFD algorithm.

Table 7. Component analysis experiment results.

Component	wHash					IFD
Wavelet selection						√
Feature fusion		√		√	√	√
Weight hash			√	√	√	√
FDC					√	√
AP	89.56	91.06	90.29	92.94	92.94	93.71
Time (s)	50.46	53.25	62.88	66.73	34.12	35.89

4.5. IFD algorithm evaluation

The original intention of the algorithm proposed in this paper is to face the specific scenario of dataset production, mainly to achieve automated filtering of raw images crawled and to reduce the cost of dataset production. Therefore, in this experiment, we compared IFD with manual screening to demonstrate the practical application value of the algorithm.

Table 8. Performance evaluation experiment results.

Amount	Method	AP	Time
2000	Manual	93.49	5.5 h
20,000	IFD	92.52	214 s

The IFD-BASE was expanded according to the method in Table 2 to obtain IFD-20000 containing 20,000 images. The IFD algorithm was used to perform similarity detection on the images in this dataset, and the detection time and accuracy were recorded. Due to the high cost of manual screening, we randomly selected 2000 images from IFD-20000 and had two people screen them separately, recording the average screening time and accuracy. The experimental results are shown in Table 8.

The processing time for manually filtering 2000 images has reached 5.5 hours, while using the IFD algorithm to process 20,000 images takes less than 4 minutes, demonstrating the algorithm's application value in detection speed. Although the accuracy of manually filtering 2000 images is slightly higher than that of the IFD algorithm, the fact that people have a certain impression of the data has improved the accuracy and speed of filtering to a certain extent. When the amount of data is larger and the data is brand new, the accuracy and speed of manual filtering will inevitably decrease by a greater extent.

5. Conclusions

The quality of the dataset is of great significance for algorithm research, but there are problems with insufficient standardization and low automation in the production of the dataset. To improve the standardization and automation of image filtering during dataset production, this paper proposes an image filtering method IFD based on PDWT and weighted hashing. IFD completes all the work of dataset filtering through a "one-stop service" approach. The IFD-BASE dataset is constructed as the data support for the algorithm validation. The performance of the method in terms of data processing

was verified by several sets of comparison experiments. The time and performance of the algorithm were compared with the traditional manual processing process. The experimental results have proven that the method proposed in this paper has excellent performance in filtering speed and accuracy. The method proposed in this paper can promote the automation and standardization of dataset production, and has certain value.

The dataset construction (data collection, data filtering, data annotation) is moving toward the direction of automated processing, and the work done in this paper is an automated method for data filtering, while there is no breakthrough in the automated processing of data annotation yet. Based on this, our subsequent work will be oriented towards the automated processing of data annotation, considering the use of convolutional neural networks to achieve semi-automation of data annotation, transforming manual annotation work into manual inspection work, reducing the labor cost of dataset production, and improving the efficiency of dataset production. This also has important implications for the development of algorithms based on deep learning for image processing.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments (All sources of funding of the study must be disclosed)

This study was supported by the Basic Scientific Research Project of Liaoning Province Education Department (Grant 202263).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, F. F. Li, Imagenet: A large-scale hierarchical image database, in *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, (2009), 248–255. https://doi.org/10.1109/CVPR.2009.5206848
2. A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, **60** (2017), 84–90. https://doi.org/10.1145/3065386
3. A. V. Emchinov, V. V. Ryazanov, Research and development of deep learning algorithms for the classification of pneumonia type and detection of ground-glass loci on radiological images, *Pattern Recognit. Image Anal.*, **32** (2022), 707–716. https://doi.org/10.1134/S1054661822030105
4. H. Tang, Research progress and development of deep learning based on convolutional neural network, in *2021 2nd International Conference on Computing and Data Science (CDS)*, Stanford, CA, USA, (2021), 259–264. https://doi.org/10.1109/CDS52072.2021.00052
5. H. Luo, J. Luo, R. Li, M. Yu, Optimization algorithm design of laser marking contour extraction and graphics hatching based on image processing technology, *J. Phys. Conf. Ser.*, **2173** (2022), 012078. https://doi.org/10.1088/1742-6596/2173/1/012078

6. L. Zhang, Y. P. Sui, H. S. Wang, S. K. Hao, N. B. Zhang, Image feature extraction and recognition model construction of coal and gangue based on image processing technology, *Sci. Rep*., **12** (2022), 20983. https://doi.org/10.1038/s41598-022-25496-5
7. X. L. Chen, H. Fang, T. Y. Lin, R. Vedantam, S. Gupta, P. Dollar, et al., Microsoft COCO captions: Data collection and evaluation server, preprint, arXiv:1504.00325. https://doi.org/10.48550/arXiv.1504.00325
8. O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in *BMVC 2015 - Proceedings of the British Machine Vision Conference 2015*, Swansea, UK, (2015), 1–12.
9. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, T, Antonio, Places: A 10 million image database for scene recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, **40** (2018), 1452–1464. https://doi.org/10.1109/TPAMI.2017.2723009
10. M. Kumar, A. Bindal, R. Gautam, R. Bhatia, Keyword query based focused Web crawler, *Procedia Comput. Sci.*, **125** (2018), 584–590. https://doi.org/10.1016/j.procs.2017.12.075
11. G. Lin, Y. Liang, A. Tavares, Design of an energy supply and demand forecasting system based on web crawler and a grey dynamic model, *Energies*, **16** (2023), 1431. https://doi.org/10.3390/en16031431
12. Q. C. Deng, K. Cheng, Collection and semi-automatic labeling of custom target detection dataset (in Chinese), *Soft. Guide*, **21** (2022), 116–122.
13. M. Z. Hua, L. M. Wang, J. W. Jiang, Construction of large-scale coral dataset based on web resources (in Chinese), *J. North. Nor. Univer.*, **55** (2023), 72–79. https://doi.org/10.16163/j.cnki.dslkxb202209230003
14. M. J. Shenza, The discrete wavelet transform: wedding the a trous and mallat algorithms, *IEEE Trans. Signal Process.*, **40** (1992), 2464–2482. https://doi.org/10.1109/78.157290
15. H. Y. Chen, H. Y. Long, Y. J. Song, H. L. Chen, X. B. Zhou, W. Deng, M³FuNet: An unsupervised multivariate feature fusion network for hyperspectral image classification, *IEEE Trans. Geosci. Remote. Sens.*, **62** (2024),1–15. https://doi.org/10.1109/TGRS.2024.3380087
16. L. Pinjarkar, M. Sharma, S. Selot, Deep CNN combined with relevance feedback for trademark image retrieval, *J. Intell. Syst.*, **29** (2020), 894–909. https://doi.org/10.1515/jisys-2018-0083
17. Z. Zeng, S. Sun, J. Sun, J. Yin, Y. Shen, Constructing a mobile visual search framework for Dunhuang murals based on fine-tuned CNN and ontology semantic distance, *Electron. Lib.*, **40** (2022), 121–139. https://doi.org/10.1108/EL-09-2021-0173
18. T. Rajasenbagam, S. Jeyanthi, Semantic content-based image retrieval system using deep learning model for lung cancer CT images, *J. Med. Imaging Health Inf*., **11** (2021), 2675–2682. https://doi.org/10.1166/jmihi.2021.3859
19. M. A. Aljanabi, Z. M. Hussain, S. F. Lu, An entropy-histogram approach for image similarity and face recognition, *Math. Probl. Eng.*, **2018** (2018), 1–18. https://doi.org/10.1155/2018/9801308
20. Y. Zhang, Y. Yao, Y. Wan, W. Liu, W. Yang, Z. Zheng, et al., Histogram of the orientation of the weighted phase descriptor for multi-modal remote sensing image matching, *J. Photogramm. Remote Sens.*, **196** (2023), 1–15. https://doi.org/10.1016/j.isprsjprs.2022.12.018
21. A. Drmic, M. Silic, G. Delac, K. Vladimir, A. S. Kurdija, Evaluating robustness of perceptual image hashing algorithms, in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics*, Opatija, Croatia, (2017), 995–1000. https://doi.org/10.23919/MIPRO.2017.7973569

22. D. G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision*, **60** (2004), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94
23. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, (2005), 886–893. https://doi.org/10.1109/CVPR.2005.177
24. K. H. Sri, G. T. Manasa, G. G. Reddy, S. Bano, V. B. Trinadh, Detecting image similarity using SIFT, in *Expert Clouds and Applications: Proceedings of ICOECA 2021*, Singapore, **209** (2022), 561–575. https://doi.org/10.1007/978-981-16-2126-0_45
25. F. Naiemi, V. Ghods, H. Khalesi, An efficient character recognition method using enhanced HOG for spam image detection, *Soft Comput.*, **23** (2019), 11759–11774. https://doi.org/10.1007/s00500-018-03728-z
26. Y. L. Liu, G. J. Xin Y. Xiao, Robust image hashing using Radon transform and invariant features, *Radioengineering*, **25** (2016), 556–564. https://doi.org/10.13164/re.2016.0556
27. N. Hussein, M. Ali, M. E. Mahdi, Detecting similarity in color images based on perceptual image hash algorithm, in *IOP Conference Series: Materials Science and Engineering*, Istanbul, Turkey, **737** (2020), 012244. https://doi.org/10.1088/1757-899X/737/1/012244
28. M. Hori, T. Hori, Y. Ohno, S. Tsuruta, H. Iwase, T. Kawai, A novel identification method using perceptual degree of concordance of occlusal surfaces calculated by a Python program, *Forensic Sci. Int.*, **313** (2020), 110358. https://doi.org/10.1016/j.forsciint.2020.110358
29. M. Fei, J. Li, H. Liu, Visual tracking based on improved foreground detection and perceptual hashing, *Neucomputing*, **152** (2015), 413–428. https://doi.org/10.1016/j.neucom.2014.09.060
30. D. M. Mo, W. K. Wong, X. J. Liu, Y. Ge, Concentrated hashing with neighborhood embedding for image retrieval and classification, *Int. J. Mach. Learn. Cybern.*, **13** (2022), 1571–1587. https://doi.org/10.1007/s13042-021-01466-7
31. A. Jose, D. Filbert, C. Rohlfing, J. R. Ohm, Deep hashing with hash center update for efficient image retrieval, in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, (2022), 4773–4777. https://doi.org/10.1109/ICASSP43922.2022.9746805
32. W. J. Yang, L. J. Wang, S. L. Cheng, Y. M. Li, A. Y. Du, Deep hash with improved dual attention for image retrieval, *Information*, 1**2** (2021), 285. https://doi.org/10.3390/info12070285
33. C. Tian, M. Zheng, W. Zuo, B. Zhang, Y. Zhang, D. Zhang, Multi-stage image denoising with the wavelet transform, *Pattern Recognit.*, **134** (2023), 109050. https://doi.org/10.1016/j.patcog.2022.109050
34. J. Bhardwaj, A. Nayak, Haar wavelet transform-based optimal bayesian method for medical image fusion, *Med. Biol. Eng. Comput.*, **58** (2020), 2397–2411. https://link.springer.com/article/10.1007/s11517-020-02209-6
35. R. Ranjan, P. Kumar, An improved image compression algorithm using 2D dwt and pca with canonical huffman encoding, *Entropy*, **25** (2023), 1382. https://doi.org/10.3390/e25101382
36. G. Strang, The discrete cosine transform, *SIAM Rev.*, **41** (1998), 135–147. https://doi.org/10.1137/S0036144598336745
37. M. Norouzi, A. Punjani, D. J. Fleet, Fast exact search in hamming space with multi-index hashing, *IEEE Trans. Pattern Anal. Mach. Intell.*, **6** (2014), 1107–1119. https://doi.org/10.1109/TPAMI.2013.231

38. H. W. Zhang, Y. B. Dong, J. Li, D. Q. Xu, An efficient method for time series similarity search using binary code representation and hamming distance, *Intell. Data Anal.*, **25** (2021), 439–461. https://doi.org/10.3233/IDA-194876
39. F. Rashid, A. Miri, I. Woungang, Secure image deduplication through image compression, *J. Inf. Secur. Appl.*, **27** (2016), 54–64. https://doi.org/10.1016/j.jisa.2015.11.003

AIMS Press