2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE) | 978-1-6654-9260-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IITCEE57236.2023.10091044

Quality Improvement of Image Datasets using Hashing Techniques

Aditi Joshi
Student,
Department of AI & ML,
BNM Institute of Technology, Bangalore
Email Id:
aditi.v2702@gmail.com

Aman V Shet
Student,
Department of AI & ML,
BNM Institute of Technology, Bangalore
Email Id:
amanshet711@gmail.com

Adarsh S Thambi
Student,
Department of AI & ML,
BNM Institute of Technology, Bangalore
Email Id:
adarshsthambi@gmail.com

Dr. Sunitha R
Associate Professor,
Department of AI & ML, BNM Institute of Technology, Bangalore
Email Id:
sunithar@bnmit.in

Abstract -Image processing has become extremely important, with the consequences of real-time image processing failures being severe; thus, research and study in real-time image processing methods are extremely important. Some images contain incorrect information, requiring the use of techniques to improve the image and make it more understandable. Others require some pre-processing for the machine to understand and make important decisions about the image on its own, without any manual intervention. Part of pre-processing includes doing a clean-up of the dataset by removal of the bias that could appear. This paper presents the framework for detecting duplicates and near duplicates aiming towards making the dataset more efficient. This will in turn help in training the ML model to be better. This is achieved by implementing the Difference Hash (dHash), Average Hash (aHash), and Perceptual Hash (pHash). Hash functions are ideal to detect (near-)identical photos because of the robustness against minor changes, while also minimizing the number of false positive collisions. The proposed model is well-suited for various real-time image processing applications.

Keywords- Image processing, Quality improvement, d-Hash, p-Hash, a-Hash.

I. Introduction

In machine learning applications, the most critical requirement for their work is the data sets they're trained on. It is essential that the data set is unbiased to ensure the efficiency of the ML model. When the model is being trained initially, it is mandatory that the training data set is not biased towards any single piece or group of data and has a uniform equal distribution of data of all entities. Automatic quality check systems could replace manual quality inspections [3-4].

In unsupervised learning, the data sets are not labelled and we only specify the desired and undesired, and the model understands and categorizes the features accordingly. If there are multiple samples of the same entity, the model may prioritize the features of the particular entity and may be biased towards it. So, when these 'biased' features are present in another entity which is not desired, the ML model may categorise it as desired. This contradicts the entire purpose of the ML model and can reduce the model's efficiency. Therefore, there must be no similar images or samples in the data set that can bias the model towards a particular feature.

Consider a model which is used to select a particular type of fruit. Let us consider the model is trained using unsupervised learning, where we only state the desired and undesired data according to the features without necessarily naming them. Then, the model has to automatically categorize the features which may be desired and undesired.

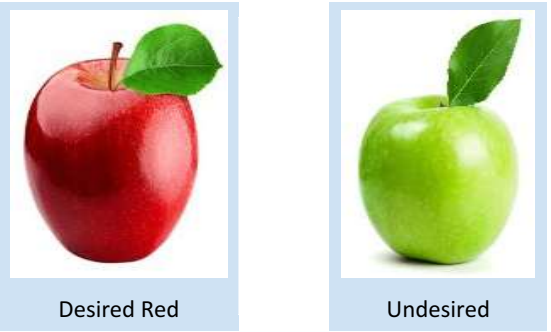

Figure 1. Initial data set with desired and undesired training samples

For example, consider the above Figure 1 having two images in which we state the image on the left is desired because of its red colour and the image on the right is undesired because of its green colour. Suppose in the data set, there are several similar images of the undesired(green) image, the model may prioritize any images(fruits) with the colour green as undesired and can even reject images that are desired.

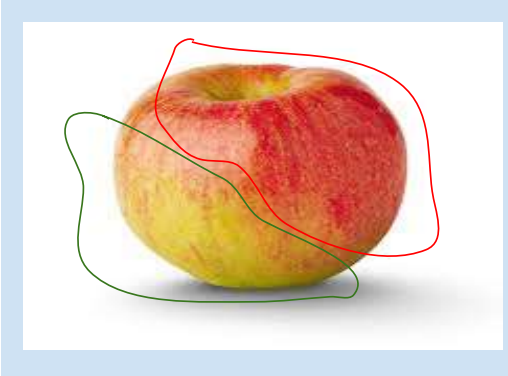

Figure 2. Apple with equal parts red and green which are to be categorized as desired

For example, consider the above Figure 2, in which the apple is required to be categorized as desired since it is equally red and green. But since the model is biased towards rejecting the green colour more than accepting the red colour, it will reject this image and therefore is not effective in its purpose.

The purpose of this paper is to study and propose a similarity index-based approach for identifying similar images to support the virtual restoration of damaged images.

This paper is further structured in the following way: the various existing literature on duplicate image detection using various machine learning and other mechanisms are done in section II, and the proposed method with different algorithms for duplicate image removal will be discussed in section III. The experimentation details and the results are discussed in the following section IV. The overall work is concluded in section V followed by references.

II. Literature Survey

This section presents various existing literature on duplicate image detection mechanisms, hashing mechanisms, image processing techniques, and similarity indexing mechanisms implemented to solve various real-time problems.

Thyagharajan et al. [1] presented a novel approach to finding nearly duplicate images that can be extended to audio and video fingerprinting research. For this study, various hashes were used, such as a-hash, p-hash, and d-hash. Because averaging pixels takes longer than taking the difference in d-hash, a-hash takes longer to load and scale images. In terms of accuracy, p-hash produced the best results. However, in terms of performance, it falls short of d-hash. D-hash may produce some false positives that can be ignored with human intervention.

Ibrahim Masri a, et al. [2] provide an overview of current image processing research, techniques, and methods. This paper focuses on the methods and algorithms used in image processing (enhancement, restoration and compression). Also describes new Real-Time processing techniques such as image segmentation, edge detection, corner detection, and so on.

Yudong Cao et al. [3] proposed a method for detecting duplicate images using the SIFT feature. The Affine-SIFT is more susceptible to affine distortions than the MSER and Hessian-Affine. The Affine-SIFT-based algorithm can detect the majority of near duplicate images with large-scale changes, viewpoint changes, occlusion, or small spatial deformation.

Anna Syberfeldt et al. [4] created a deep neural network model to detect quality issues in paper bag production. They have hired specialised personnel to conduct quality inspections, which involve comparing one or more product characteristics to product specifications. To avoid releasing low-quality products, products that do not meet the specifications are rejected or returned for improvement.

A study on finding flaws in the production of emulsion pumps was reported by Zhu et al. They look at the viability of replacing the current manual inspection of the pumps with CNNs. High-quality image sets were essential because the lack of images of damaged products posed the biggest challenge to their adoption. Slant correction was applied as a pre-treatment to enhance the image quality. The CNN had a 97% prediction accuracy with a mean detection rate of 0.18 seconds when evaluated on fresh images.

An investigation of the textile sector by Jing et al. focuses on finding fabric flaws. The objective of this work, like that of Zhu et al., was to substitute CNNs for manual inspections. Using a dataset of several fabrics with a range of colours and recurring patterns, the scientists instructed CNN to identify six types of typical faults. The fabric patch sizes were determined automatically by the network, improving accuracy and enabling the detection of extremely minor flaws. Over 97% accuracy was attained on average by CNN.

Amerini et al. presented the Scale Invariant Feature Transform (SIFT) to filter, sort, and categorize key point pairs for copy-move defect detection. Li et al. try to speed up DAR and FPR by segregating a worrisome image into non-overlapping patches to speed up similar region-matching times. A key points-based image passive detecting technique based on the Harris detector and region growth technology was presented by Wang et al. It is immune to gamma correction, brightness enhancement, and JPEG compression.

Li et al. suggested a fuzzy clustering approach based on an optimally stable colour region detector and Zernike moments to extract all critical point features. A technique for fragmenting a suspicious image into irregular superpixels labelled as smooth, texture, and strong texture was presented by Wang et al.. The stable image key points can be extracted from each superpixel. The detection of copy-move fraud has advanced thanks to the existing techniques. However, they were unable to improve DAR and FPR in a way that would have cut down on matching times. Adjusting the contrast ratio, brightness, hue, and hybrid operations, as well as other post-processing procedures, encounter less opposition.

A sort of one-way translation from multimedia presentations to perceptual hashes depending on the perceptual content is known as "perceptual hashing." It is frequently used for cataloguing, finding, and verifying multimedia content. Perceptual hash algorithms are used to create biometrics for digital photos and then compare them, much like target tracking and image searching.

Q. Cheng et al. [16] created a semantic-preserving deep hashing model for multi-label remote sensing image retrieval. They created this model by combining a hashing mechanism with a machine-learning mechanism. Qian et al. [15] presented an image copy-move forgery detection mechanism based on Perceptual Hashing. To identify duplicates, they primarily used a perceptual hashing mechanism while storing and processing images.

We are deploying the machine learning mechanism along with hashing techniques to detect and remove duplicate images based on a survey we conducted on duplicate image detection for various purposes. We employ the Structural Similarity Index Measurement (SSIM) concept [14]. SSIM is a quality evaluation index for full reference images that measure image similarity based on brightness, contrast, and structure. The SSIM scale runs from 0 to 1. The greater the value, the less image distortion. Rather than calculating the similarity of two images, this method is typically used to measure the distortion of a compressed image.

III. Proposed Solution

To improve the efficiency of the training data set and remove any bias, we propose a multilevel model in which all the duplicates and similar images of an image are removed. At each level, we use a hashing algorithm to remove a particular type of a duplicate. After the final level of removing similar images, we obtain an unbiased and efficient data set that can be used to train the model. The overall design of the proposed model is depicted in the figure 3.

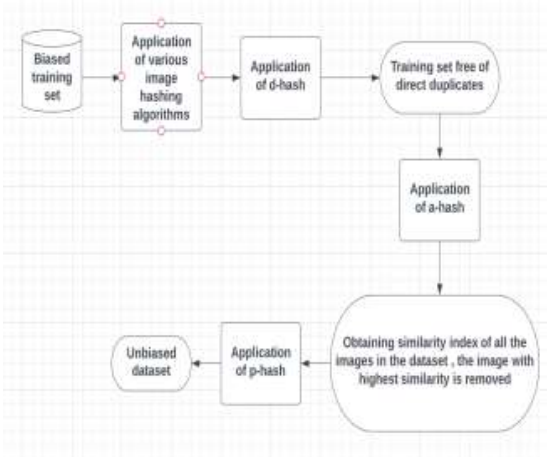

Figure 3. Design diagram of the multilevel model

Testing data is used to evaluate the performance of the model. The images used in the above example are used as the testing dataset. Initially this test dataset is a biased one, hence leading to a poor performance of the ML model. In order to make the

model unbiased we use a multilevel implementation of a few hashing function. The first stage includes the d-hash algorithm that is applied on the dataset. As a result, the 'direct duplicate' images are removed from the dataset.
In comparison to the other two hashing algorithms, d-hash, difference hash, is found to be the most accurate in terms of eliminating the direct duplicates and is the fastest. The second stage of our multilevel implementation involves using the average hash or a-hash algorithm which is applied to the resultant dataset that is free of direct duplicates. The similarity index of all the images in the dataset is obtained after applying the a-hash algorithm. The similarity index of one image is compared to the similarity index of all images. Images with a higher index value are removed. There are no near duplicate images in the resulting dataset. Finally in the third and the final stage , the p-hash algorithm is applied to the dataset which is obtained in the previous step, yielding the unbiased 'final' dataset. This unbiased dataset has a higher performance and can be used to train any ML model.

D-Hash (Difference Hash):

dHash is a type of hashing algorithm that is used to identify similar and duplicate images. The input image is first converted to grayscale by the difference hash technique. Working with the difference between neighbouring pixels, the dHash algorithm indicates the direction of the relative gradient. Then the hamming distance of the adjacent pixels is calculated. If the hamming distance is less than 5, then the images are similar/duplicates.

A-Hash (Average Hash):

Each pixel block is compared to the average (as the name implies) of all the pixel values in the image after the decolorizing and scaling steps. We'll create a 64-bit hash; thus, the image will be resized to 8x8 pixels. When we apply the average hash function to the 10 altered photos, we can observe that there are only slight variations in the image hash.

P-Hash (Perceptual Hash):

Perceptual hashing is a class among one translation from visual materials to perceptual hash values in terms of perceptual content. It is commonly used to retrieve, identify, and authenticate multimedia material. The proposed technique efficiently locates and identifies many duplicated regions in digital images that may have been warped by modifying contrast ratio, brightness, hue, and their hybrid operations. It does this by using perceptual hash algorithms. It's possible that these modifications introduced white Gaussian noise and Gaussian blurring.

IV. Results and Discussion

The model is designed using python programming and executed. The dataset is collected by us from different sources. The model is tested with 31 images for testing it. Here, we are going to find similar images using distance using two concepts: dHash and hamming distance. The data set contains a list of duplicates.

The code is as:
We import all the required dependencies. Then we import all the images of the sample data set. Initially, we convert the image into a grey image. The next step is to resize the image. The higher the size of the image, the more precise fingerprints we get. Out of the many algorithms available, here we flatten the images, row by row and column by column. We find the intensities of the gradient images and assign 0s and 1s using the flattened values. We obtain a basic thumbnail between the two images. Then using all these values, we obtain all the difference hash values and store them in a dictionary. Then, we check these hash values with all the other images in the set and filter the images as original or duplicate. In our test set, we take 31 images to check whether the results are per the algorithm.

Once we run the algorithm on the test set, we can see that the algorithm lists out the set of originals and their duplicates as shown in Figure 4 and 5. This completes the initial removal of duplicates.

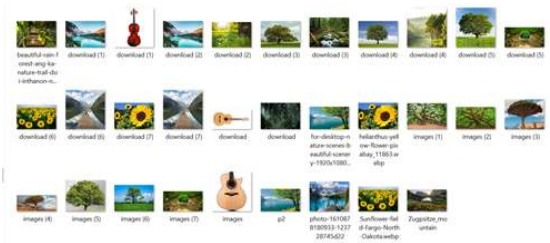

Figure 4. Input image dataset

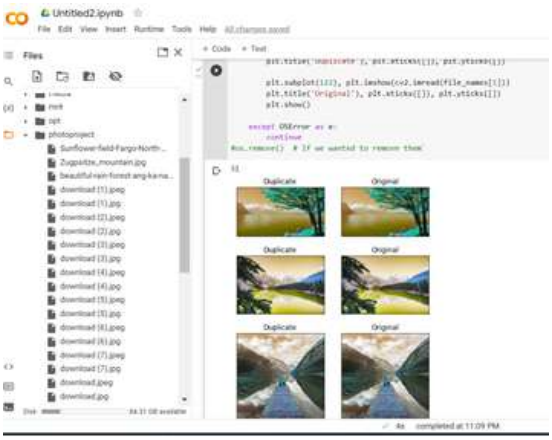

Figure 5. Output of the model

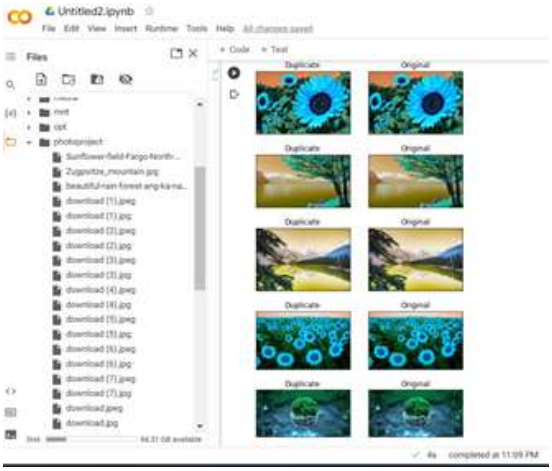

Figure 6. Output of the model

Table 1 is taken from [14] as reference for choosing the dHash hashing algorithm to display the results.

Contrast Images		Time (s)			Accuracy		
Image1	Image2	aHash	dHash	pHash	aHash	dHash	pHash
original	original	0.31	0.09	4.04	100.00%	100.00%	100.00%
original	light	0.31	0.09	4.08	98.44%	96.88%	97.66%
original	resize	0.28	0.07	3.96	93.75%	95.31%	96.88%
original	contrast	0.28	0.07	4.21	98.44%	98.44%	99.61%
original	sharp	0.32	0.07	4.15	82.81%	85.94%	83.20%
original	blur	0.29	0.07	4.15	76.56%	73.44%	95.28%
original	color	0.28	0.07	3.91	98.44%	100.00%	100.00%
original	rotate	0.29	0.07	4.07	56.25%	57.81%	57.81%

Table 1. Comparison among aHash, dHash, and pHash

From Table 1, it is observed that the time taken by dHash is less compared to the other two algorithms. The pHash algorithm is superior to the aHash method, whereas the dHash algorithm falls short. But is best for detection of exact duplicates. ahash has the average time and average accuracy among the three. pHash has the highest accuracy. But has the least execution time i.e. pHash is slow compared to the other two algorithms. The pHash algorithm can ignore the effects of gamma correction and colour histogram modification and can tolerate images with rotation and size changes of less than 25%. From the experimentation, it is observed that dHash is the most efficient among all the three. But pHash is the most accurate one. Although if it has the least execution time.

This solution can be applied to a wide range of ML models, where the principal training data set is images. For example, Image classification, Images for Weather Recognition, Recursion Cellular Image Classification, Architectural Heritage Elements, Coast set Image Classification Dataset. These are just some specific applications where the training data sets are images, but the solution can be easily applied to any general ML model application using images as their training set.

V. Conclusion

Duplicate Fraudsters sometimes employ counterfeiting as a popular method of altering digital images. To accomplish the target of hiding some targets or accentuating some key things, some parts in a digital image are copied and then pasted into other sections in the same image. Here we are proposing a multilayer approach in which every duplicate and related picture of a given image is eliminated. We employ a hashing method to eliminate a certain kind of duplication at each level. We get a fair and effective data set that can be utilised to train the model after the last level of similar image removal. The results show the good performance of the proposed model in detecting the duplicate objects in the various image processing applications. This model is well suited for various real-time applications.

References

[1] Thyagharajan, K. K., & Kalaiarasi, G. (2020). A Review on Near Duplicate Detection of Images using Computer Vision Techniques. arXiv. https://doi.org/10.1007/s11831-020-09400-w.

[2] Masri, Ibrahim & Erdal, Erdal. (2019). Review Paper on Real Time Image Processing: Methods, Techniques, Applications.

[3] Y. Cao, H. Zhang, Yanyan Gao and Jun Guo, "An efficient duplicate image detection method based on Affine-SIFT feature," 2010 3rd IEEE

International Conference on Broadband Network and Multimedia Technology (IC-BNMT), 2010, pp. 794-797, doi: 10.1109/ICBNMT.2010.5705199.

[4] Anna Syberfeldt, Fredrik Vuoluterä, Image Processing based on Deep Neural Networks for Detecting Quality Problems in Paper Bag Production, Procedia CIRP, Volume 93, 2020, Pages 1224-1229, ISSN 2212-8271, https://doi.org/10.1016/j.procir.2020.04.158.

[5] Zhu, C., Zhou, W., Yu, H. & Xiao, S. Defect Detection of Emulsion Pump Body Based on Improved Convolutional Neural Network. Proceedings of the 2019 International Conference on Advanced Mechatronic Systems. Kusatsu, Shiga, Japan 26-28 August 2019, 2019;349-352.

[6] Jing J-F., Ma H., Zhang H-H. Automatic fabric defect detection using a deep convolutional neural network**,** Coloration Technology., 135 (3) (2019), pp. 213-223

[7] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," IEEE Transactions on Information Forensics and Security, vol. 6, no. 3, pp. 1099–1110, 2011.

[8] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo, "Counter-forensics of SIFT-based copy-move detection by means of keypoint classification," EURASIP Journal on Image and Video Processing, pp. 1–17, 2013.

[9] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," IEEE Transactions on Information Forensics and Security, vol. 10, no. 3, pp. 507–518, 2015.

[10] X. Wang, G. He, C. Tang, Y. Han, and S. Wang, "Keypoints-Based Image Passive Forensics Method for Copy-Move Attacks," International Journal of Pattern Recognition and Artificial Intelligence, vol. 30, no. 3, Article ID 1655008, 2016.

[11] J. Li, F. Yang, W. Lu, and W. Sun, "Keypoint-based copy-move detection scheme by adopting MSCRs and improved feature matching," Multimedia Tools and Applications, vol. 76, no. 20, pp. 20483–20497, 2017.

[12] X.-Y. Wang, S. Li, Y.-N. Liu, Y. Niu, H.-Y. Yang, and Z.-L. Zhou, "A new keypoint-based copy-move forgery detection for small smooth regions," Multimedia Tools and Applications, vol. 76, no. 22, pp. 23353–23382, 2017.

[13] X. M. Niu and Y. H. Jiao, "An overview of perceptual hashing," ACTA Electronica Siniica, vol. 36, no. 7, pp. 1405–1411, 2008.

[14] Hua, Wei & Hou, Miaole & Qiao, Yunfei & Zhao, Xuesheng & Xu, Shishuo & Li, Songnian. (2021). Similarity Index Based Approach for Identifying Similar Grotto Statues to Support Virtual Restoration. Remote Sensing. 13. 1201. 10.3390/rs13061201.

[15] Cheng, Q.; Huang, H.; Ye, L.; Fu, P.; Gan, D.; Zhou, Y. A Semantic-Preserving Deep Hashing Model for Multi-Label Remote Sensing Image Retrieval. *Remote Sens.* 2021, *13*, 4965. https://doi.org/10.3390/rs13244965

[16] Qian, Zhenxing, Wang, Huan, Wang, Hongxia, Perceptual Hashing-Based Image Copy-Move Forgery Detection, Security and Communication Networks, Hindawi, 1939-0114, https://doi.org/10.1155/2018/6853696.