Original Article

Image Capturing and Deleting Duplicate Images through Feature Extraction using Hashing Techniques

Prathima Ch [1], R. Swathi[2], K. Suneetha[3,] I. Suneetha[4], B. V. Suresh Reddy[5], Siva Kumar Depuru[6]

[1,6]*School of Computing, Mohan Babu University, (Erstwhile Sree Vidyanikethan Engineering College) Tirupati, India*
[2]*Department of CSA, Sri Venkateswara College of Engineering Tirupati Andhra Pradesh*
[3]*School of Computer Science & IT, Jain (Deemed-to-be University), Bangalore, Karnataka.*
[4]*ECE Department, Annamacharya Institute of Technology and Sciences, Tirupati*
[5]*Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.*

[2]*Corresponding Author : keerthisuni.k@gmail.com*

Received: 25 September 2023 Revised: 14 December 2023 Accepted: 23 December 2023 Published: 07 January 2024

Abstract *- Nowadays, a major issue in society is the duplication of all the objects in terms of images while capturing. For example, if a folder is on the PC, it allows the storage of the same image with different names. Here, the waste of memory is more. So, to avoid duplication of images, needed to scan the content inside the file and identify whether the images are duplicates or not. This process is known as CBIR (Content Based Image Retrieval). There are many techniques to find duplicate images. CNN (Convolution Neural Networks), Phash (perceptual hash), Block Truncation Technique etc., are used to find the image similarity identification and to avoid wastage of memory. The proposed work is to capture images and store them in a folder to identify duplicate images and delete them. Developed a user interface to delete duplicate images using a folder path for user-friendly usage. There are various techniques for image deduplication and all the techniques will not work effectively. There is a need for image deduplication in a wide range of domains like income tax, banks, and many other private organizations and private corporations where there store many images that are duplicates. Even some of the social media companies store images that are shared using the app. All these need to be deduplicated to save money, maintenance costs and storage space.*

Keywords *- Images, Convolutional Neural Network, Deep Learning, Deduplication.*

1. Introduction

In day-to-day life, we all deal with huge amounts of multimedia like images, videos, and songs. The same images are shared through multiple platforms like WhatsApp, Instagram, Facebook, and Telegram. All these images are stored on the phone or PC. This causes wastage of memory. A record or volume that is backed [2] up every week creates a huge amount of duplicate data. Even storing [5] the same image with different resolutions, sizes, and pixels causes duplicate data.

To store important data and access the data fast, have to clear all the duplicate data. However, clearing such a huge amount of duplicate data manually is difficult for everyone. All the users feel lazy about doing it manually. So, need some automatic systems to clear all this junk and duplicate data. To do this, there are many methods using artificial intelligence, deep learning, and ML. Some of them are CNN [1] (Convolution Neural Networks), Phash [4], Image Hashing [3], Block Truncation techniques.

Finding identical photos in your entire system is effort and difficult for us. You need to search among hundreds and thousands of images and then verify for each photo whether they are different or similar. Commonly, use the file names to differentiate the images that are stored in a folder. Images are generally derived from different sources, like mobile devices and social media apps, that make differences in name and size but also scaling, resolution, and compression brightness. So, HASH Functions are used to detect identical images.

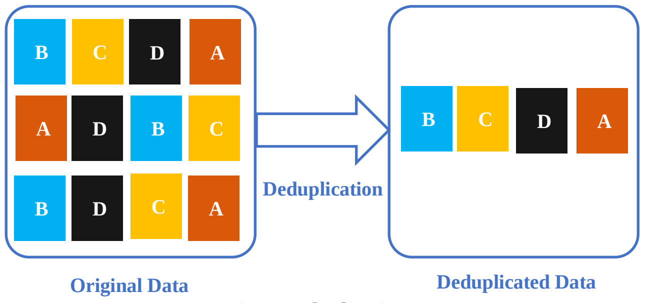


Fig. 1 Deduplication

Two images may look visually the same but numerically different. The existence of numerical differences is due to various reasons, such as the usage of different apps (social media apps), which may change the brightness, contrast, compression, resolution, or scaling. For example, sharing an image using Instagram leads to a decrease in resolution. The decrease in resolution may differ from device and user-defined settings. Visually, it is hard to see any changes between the source image and the Instagram image.

Deduplication [6] of images decreases storage costs as fewer disks are needed. It also Increases critical disaster recovery because there will be less data to propagate. Backup and archive data usually include a huge amount of duplicate data. It can differentiate the images by scanning them through pixels. Every image will have different hash functions [22]. This helps us to find duplicate images.

Nowadays, many private and public organizations are shifting to the cloud storage system. And this follows pay-as-you-use. They pay as they how much they used. This means they will pay like the rent. Users of cloud storage seek to ensure the confidentiality of their data, and both clients and servers want to utilise as little storage space as feasible. Existing deduplication methods [7] for regular files cannot be used since images must be detected based on their visual content. In this paper suggests a unique safe picture deduplication method based on clustering and a local binary pattern-based perceptual hash methodology.

Deep learning is a kind of machine learning that helps make Artificial Intelligence (AI) systems. This idea uses Artificial Neural Networks (ANN) [9] to analyze large amounts of data. The data is passed through layers of neurons to get a better understanding. Deep Neural Networks (DNN) [15,16] can be set up in many different ways. Deep networks called CNN or DCNN are used a lot to look for patterns in pictures and videos. DCNNs [18] are like computer brains. They are based on the brains of animals that see things. They work in a 3D way [19] to see things better than regular computer brains. Using deep learning computers to help do many things, like identifying objects and pictures, suggesting things which might like, and even understanding what is said.

Near-duplicate picture recognition [21] is easy to understand and has been studied a lot more than other problems in computer vision. Many ways to show pictures, like and, have been suggested. Find out how much two pictures or parts of a picture look alike. These features can handle noise and changes in images very well, making them good enough for identifying identical images [8]. Adding pictures to the Bag-of-Words can help find more pictures accurately when searching for a big collection of them. Made a way to find similar pictures in a group of pictures using SIFT BoW. Their experiments showed that their technique works well for as many as 1 million images with great precision.

2. Proposed Systems

2.1. Mohammed Hazim Alkawaz, Ghazali Sulong, Tanzila Saba & Amjad Rehman (2018), Detection of copy-move image forgery based on discrete cosine transform.

This article gives much information about image processing and image identification. It is hard to differentiate the two images with similar properties, but minute differences like changes in resolution and compression, brightness, contrast, etc. This article gave much information about the different methods to delete duplicate images and the advantages of deleting duplicates. It explained in detail about Discrete Cosine Transform (DCT) [21]. It will detect the tempered regions. The main idea of this paper is to identify the image which is best and original (source image) among all identical images. Because the human eye can't detect those images, this method is simple and gives accurate results with the best time complexity, and the efficiency of the algorithm is high.

2.2. Chen CC, Hsieh SL (2015), Using binarization and hashing for efficient SIFT matching.

The SIFT algorithm extracts the distinctive features for image retrieval. This method takes much time and slows down the whole process. The Euclidean distance is used to calculate the similarity of two features, which is expensive because it involves taking the square root. So, to enhance the SIFT matching, this paper introduces a new concept of the binary representation of the image retrieval process. Here, instead of calculating the distance, perform the bitwise operations so that retrieval time is decreased much. In this process, Hashing is used to retrieve similar images. Further, using the binarized features speeds up the image retrieval process. Here learned the process of finding similar images with their accuracy.

2.3. Dharshini T, Angelina JJR (2019), CT image denoising using NLM and correlation-based wavelet packet thresholding.

In the medical field, medical imaging with computed tomography is used to examine the internal structure of the human body parts using the tool called radiological diagnosis tool. If comparison between X-ray with X-ray CT, it adds the mathematical calculations to enhance the process. But it is harmful to patients. Using this method instead of the general process will increase the risk to children. Should try to avoid this method. To examine and innovate the internal structure of the body parts, one needs to scan the images properly and identify them. With the naked eye, it is impossible to see them. So, hashing techniques are used to identify similar structures among the many scans. It is necessary to find the images with their similarity and differentiate the structures. Having a different hashing technique for image processing, image retrieval, image identification and many more.

2.4. Singh, S., Bhatnagar, G. & Singh, A. (2021). A New Robust Reference Image Hashing System.

This paper gives details about the perceptual hashing technique. The protection and authentication of multimedia data is a very difficult and challenging task for us. In this challenging task, we can use the Perceptual Hashing method to overcome the issues. This paper explains a novel system for generating an image hash. In this process, it utilizes the local and global features of the hashing techniques. The local features can be obtained in many ways, like non-linear scale-space-based KAZE features. These local features are stable. This process involves normalizing the image into coordinates, and then, with the help of this local feature, global features of the image are extracted. We also have many Python libraries to perform all these hashing techniques. Can use them effectively to do all these tasks. There are many hashing methods in Python, like average hashing, image hashing, perceptual hashing and many more. Hashing is an algorithm that works by calculating a fixed-size bit string value from an image or file. In this hashing technique, hash tables and hash functions play an important role. This paper helped us a lot to understand the hashing mechanism [11], how to apply it in day-to-day life, and its massive applications in society.

3. Survey

This survey was done to know the opinions of the people from different sections, like a student, faculty, lab admin and the public regarding issues raised by duplicate images.

3.1. Student

Surveyed a student who was facing major issues with the same images but with different file names in his phone gallery. Whenever this student reaches his Instagram or any social media, various images are generated over those platforms. In turn, those repeated or the same images have been stored in the gallery of students, which reduces the storage space in the device and also makes it work slowly.

3.2. Faculty

A faculty has also been surveyed regarding the issues of duplicate images. In general, faculty would be circulating various images regarding syllabi, circulars, and exam schedules. They also need to deal with the images with respect to the profession. Also, being busy in the profession, it is almost impossible to delete the duplicate images stored in the gallery. Probably, the space recovered by deleted duplicate images could be utilized for useful purposes.

3.3. Lab Administrator

The work done by an administrator, especially in labs, faces major problems of duplicated images in their systems. Even this might lead to the loss of much greater storage space in their databases or even in the individual systems because many of the students who are working over systems are going in the same flow. That is, the majority of them are following the instructions of their mentors and technicians. Here, the same images would be seen at last in a huge number which couldn't be deleted manually.

3.4. Neighbors

Even the common public near and around us are also facing major issues with duplicate images. Particularly images generated in the WhatsApp groups, social media, and general usage; many of the duplicate images are saved in the galleries. This reduces the storage capacity and also makes them work slowly; moreover, the reduction of duplicate images is necessary as it is impossible to search for similar images in the storage space. Also, the fast-running lifestyle got used to better technologies to do their work efficiently.

4. Existing System

Improvements in image deduplication are taking place in day-to-day life. A huge number of models and techniques are discovered to find duplicate images. Few methods and techniques will work effectively and are not. There are different types of deduplications methods that suit only particular situations. It means there are different techniques for various situations.

The main deduplication methods are developed using Convolution neural Networks. Few techniques are completely for cloud storage [8,10], and few are using clustering-based, and many other methods are developed using virtual techniques. These deep CNN techniques work based on the pattern in the images. First, they will scan the image completely and store the image patterns. Then, if they find any same or similar pattern, then it is found to be a duplicate.

5. Problem Definition

The major issue is duplicate or redundant data and storage problems [17]. This is everyone's issue who uses phones and computers. Even in big data centers, 25 percent of duplicate data is stored. While talking about images, the same images are stored with different names in the same folder or directory.

Now, the problem is to clear these duplicate images. Feel lazy to clear this manually. Even clears duplicate data if the data size is in a small range. It is difficult to delete duplicates from huge data sets. Here, manual work is late and complex to complete. In this running world, nobody will spend more time doing this.

To overcome this problem, storage needs an automatic process.There's a picture store, an advanced hacking gadget, and a watermark encoder within the framework. A computerized picture hashing gadget computes a hash esteem for each computerized picture, conglomerating outwardly comparable photographs into the same hash esteem and outwardly isolated pictures into various values.

The hash esteem is spared within the picture's hash table and associated with the initial picture through the table. Pictures can be put away in this picture hash table. The watermark encoder utilizes the hash esteem and mystery to calculate the watermark. Utilizing both values makes the watermark flexible to BORE (Break Once, Run All over) ambushes since an assailant still requires the hash esteem of each picture to assault, indeed, on the off chance that the ordinary watermark mystery is uncovered.

6. Methodology

6.1. User Interface using Tkinter

Tkinter is indeed the Python interface to the Tk GUI toolkit, and it's used for creating graphical user interfaces. Tk, on the other hand, is a separate library that Tkinter is built upon. Tk is a GUI toolkit written in the Tcl programming language, and it provides the basic building blocks for creating graphical user interfaces.

Tkinter wraps these Tk functionalities for use in Python. Tkinter is available on most Unix platforms, including Linux, and it's also available on Windows. It's one of the standard libraries included with Python, so you don't need to install it separately.

To check if Tkinter is properly installed on your system and view the Tk version, you can open a Python terminal and run the following command: import tkinter as tk, print(tk.TkVersion), This will print the version of Tk that Tkinter is using. Tk is indeed implemented in C, and it's used as a library by both Tcl and Python (via Tkinter). It provides a set of widgets and functions for creating GUIs.

Tk is a Tcl module that is developed in C language, which adds custom user commands to create and change the GUI interface. All Tk object embeds Tcl interpreter objects with Tk loaded into it. Tk interfaced widgets are very customizable, though the cost of appearance.

6.2. Tinker Modules

The Tkinter application is accessed and served across several modules or libraries.

Most of all, the applications need the main tkinter module and the tkinter. ttk module, which gives the latest themed widget and API:

```
from Tkinter import *
from Tkinter import ttk
```

Tk 8.5 invented the latest set of themed user interface components along with a new and latest API to use them easily. All APIs are available and useful. In most of the documentation, you will find online still using the legacy API.

Detection and deletion of duplicates:

```
Modules used in the work are:
        From PIL import Image
                import imagehash
                import os
```

6.3. About PIL Module

Python Imaging Library (PIL) is a free and open-source library for the Python programming language that supports opening, editing, and saving different image or photo formats. PIL are the de facto image processing library for Python language. PIL incorporates a lightweight image or photo processing tool that helps in editing, creating and saving images. These aids for Python Imaging Library have stopped in 2011, but a work application named Pillow is the original PIL module and added Python3 support for it. Pillow is announced as another name for PIL for future usage and development. Pillow supports a huge number of image file formats like PING, JPEG, BMP, TIFF and many more.

Expected Outcomes: Removes duplicate images, reduces storage space, Redundancy of data is avoided, User friendly interface, Faster deletion of duplicates.

6.4. About Image Hash

Python supports an image hash library. ImageHash package contains Average Hashing (AV), Perceptual Hashing (PH), Difference Hashing (DH), Wavelet Hashing (WH), HSV color hashing (colorhash), Crop Resistant Hashing (CRH).

But here, the average hashing technique is used. Average hashing works very effectively. The results are accurate. Even it will not take more time to run the application. To import the image hash library, first need to install the anaconda prompt and then run the code in the virtual environment to avoid dependency and version issues.

The virtual environment supports installing and running the different versions. The hash function turns the entered data into a short string that is like a unique signature. This is called the hash value of the image. So, to create a good hash function have to rely entirely on the input data, which in this case is an image.

There are different ways to summarize things, like images or text. Some of these methods are called average hashes, perceptual hashes, difference hashes, Haar Wavelet Hashes (HWH), Daubechies Wavelet Hashes (DWH), HSV color hashes, and crop-resistant hashes. Each hash function has different properties that protect it against changes in darkness, darkness, lightness, contrast, image quality and size.

For some time now, the pre-processing steps have been fundamental to creating a hash of a newly finished image: 1. Color removal, 2. Standardization of pixel values, and 3. Image scaling. The background for color removal is that the information needed to "perceive" the image is quickly available in the gray scale. Also more computationally interesting is the drop from 24 bits per pixel in RGB to 8 bits per pixel, both in terms of speed and memory [12-14]. At this point, the image is reduced/scaled to a smaller size.

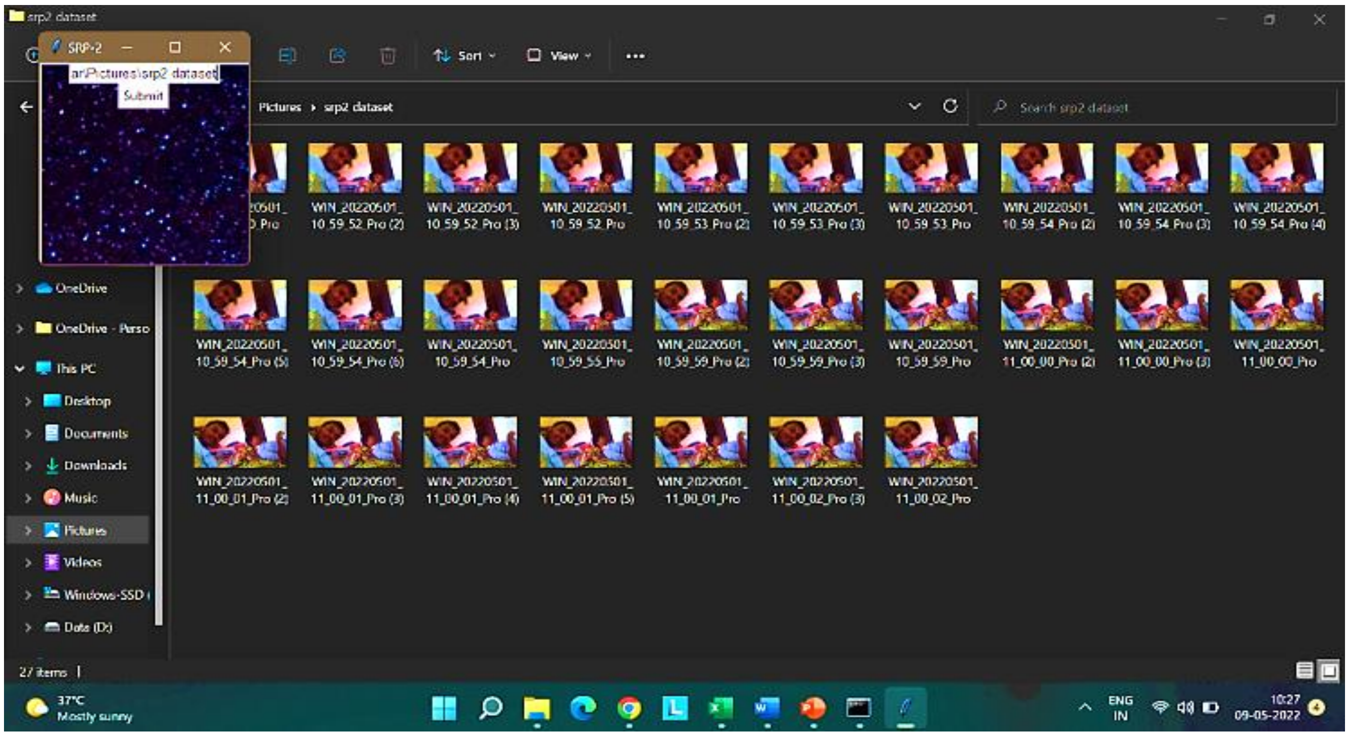


Fig. 2 Live image capturing

The most commonly used is the 64-bit hash, which basically infers that the image has been reduced to 8×8 pixels. Below is an example of calculating the hash of an image using Python for different hash powers. To create a virtual environment in anaconda prompt need to use the following command:

Conda create –name virtualenv_name

After executing the above command, it will install all the required commands. To use the virtual environment, need to activate it. To activate the virtual environment, need to use the following command:

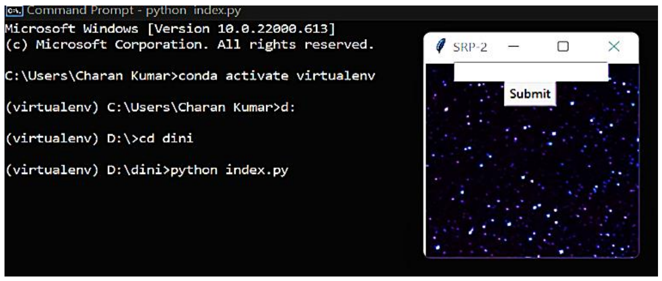
Conda activate virtualenv_name

7. Results and Discussion

First, open your camera and take some photos in the same position. If you take photos using a laptop camera, you should take them in such a way that the images should be similar. Store all the images in the same folder so that the folder contains some duplicate images. Even you can take any image dataset from Google or take photos with your mobile and store it on your laptop or PC. Open the Command prompt. Create a virtual environment in the command prompt. Then run the index.py file. Here, connect the image hashing code to the user interface.

The following picture contains the user interface for easy access.The user interface was developed using Tkinter, which is a Python modle. The interface contains a submit button and an input field. The input field is for pasting a folder path in which the duplicates are present, and the submit button is for submitting the input field value, i.e. folder path. To run the program after creating the virtual environment, navigate to the folder where the program is stored on the PC or laptop.

The type ‘python file_filename.py’ and runs the program. Now copy the folder path that contains the images and make a count of images before executing the code. While copying the folder path, make sure to copy the whole path without any leading or lagging spaces in the path. For example, the path is “C:\Users\Charan Kumar\Pictures\Camera Roll”. Here I copied the whole path, including the directory name. Even if you miss the directory name, accessing the folder is not possible. In the folder now, there are a total of thirteen images. All the image is similar. Now, enter the folder path in the entry field of the website and click on the submit button. After submitting, duplicate images are successfully deleted. Now, count the images in the folder to differentiate the number of images deleted in the folder. To find the total number of duplicate images deleted need to subtract both the count.



```
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Charan Kumar>conda activate virtualenv

(virtualenv) C:\Users\Charan Kumar>d:

(virtualenv) D:\>cd dini

(virtualenv) D:\dini>python index.py
```
Fig. 3 Running command prompt in virtual mode

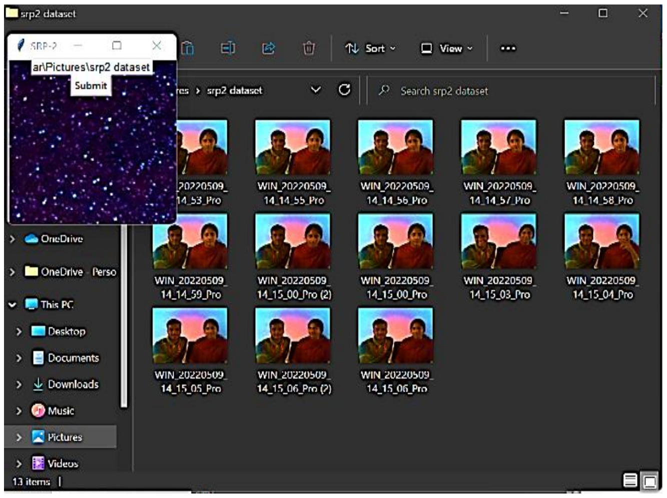

srp2 dataset
SRP-2
ar\Pictures\srp2 dataset
Submit
Sort
View
srp2 dataset
Search srp2 dataset
20220509_ 14_14_53_Pro
WIN_20220509_ 14_14_55_Pro
WIN_20220509_ 14_14_56_Pro
WIN_20220509_ 14_14_57_Pro
WIN_20220509_ 14_14_58_Pro
WIN_20220509_ 14_14_59_Pro
WIN_20220509_ 14_15_00_Pro (2)
WIN_20220509_ 14_15_00_Pro
WIN_20220509_ 14_15_03_Pro
WIN_20220509_ 14_15_04_Pro
WIN_20220509_ 14_15_05_Pro
WIN_20220509_ 14_15_06_Pro (2)
WIN_20220509_ 14_15_06_Pro
OneDrive
OneDrive - Perso
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
13 items

Fig. 4 Folder before deleting duplicates

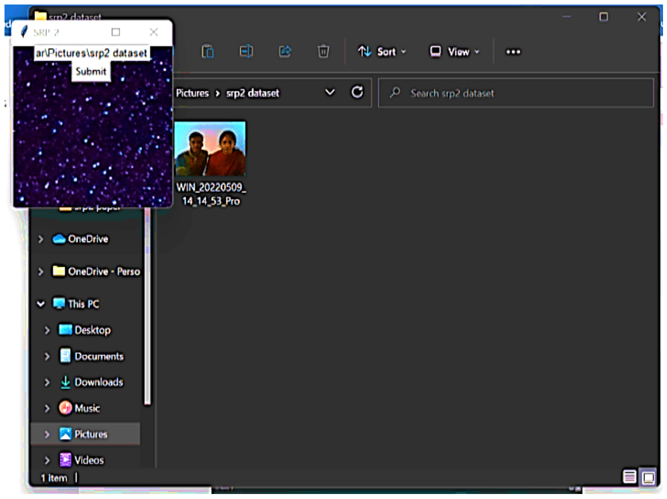


Fig. 5 After deleting duplicates

Table. 1 Image similarities

S. No	All Images	Duplicates	Similarity
1.	Img (1).jpeg	Img (1).jpeg	99
2.	Img (2).jpeg'	Img (2).jpeg'	93
3.	Img (3).jpeg'	Img (3).jpeg'	91
4.	Img (4).jpeg'	Img (4).jpeg'	93
5.	Img (5).jpeg'	Img (5).jpeg'	93
6.	Img (6).jpeg'	Img (6).jpeg'	92
7.	Img (7).jpeg'	Img (7).jpeg'	94
8.	Img (8).jpeg'	Img (8).jpeg'	95
9.	Img (9).jpeg'	Img (9).jpeg'	97
10.	Img (10).jpeg'	Img (10).jpeg'	97
11.	Img (11).jpeg'	Img (11).jpeg'	96

Finally, there is only one image in the folder; out of 13 images, 12 images are deleted, and only one image is not deleted. To run the above program, every image is compared with each other. If its similarity is greater than or equal to the given threshold similarity, then the images are considered duplicate images. In the above case, the similarity is given as 90%. Tested this algorithm using 20 images; it yields 16 comparisons in total, with a total time of 1 minute thinking. My CPU usage is 100% all that time, which makes me think that this thing can be used as a stress test for my CPU. This algorithm is tested not only for 20 images, it has been tested for 11, 15, 30, 50, and 100 images. Also, heavily constraint this test by storing the image files on a RAM Disk, so Python won't have a problem when opening the files. The only bottleneck is the total number of comparisons.

8. Conclusion

The application is quite useful for detecting and deleting duplicate images. This results in saving storage and space in the devices and also enhances the working of the device. Even though it is a heavy task to manually detect and delete duplicate images, with the help of this application, hard work is converted into smart work. Various sections of people use this according to their needs. Mainly, laptops and personal computer user will get benefit from this application to remove duplicate images from their folders. If the application is converted into apk then all the users of mobiles can be utilized it. For each hash work, the discoveries were or maybe steady. While using a genuine dataset, it may also be seen that a hash estimate of 8 (64-bit) makes a part of collisions for both the normal hash and the wavelet. Besides, it clusters various nearly indistinguishable photographs together. Differential comes about were both precise and cautious.

The observational change, on the other hand, created solid comes about but is less cautious. When the hash measure was expanded to 16 (256 bits), none of the hash calculations made collisions, but the normal hash and the wavelet created nearly comparative pictures. There were no collisions or about comparative pictures in either the seen hash or the distinction hash. The scattering of perceptions is additionally less cautious in this case.

References

[1] S. Bhattacharjee, and M. Kutter, "Compression Tolerant Image Authentication," *Proceedings of International Conference on Image Processing*, ICIP98 (Cat. No.98CB36269), Chicago, IL, USA, vol. 1, pp. 435-439, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[2] Lu Chen, Feng Xiang, and Zhixin Sun, "Image Deduplication Based on Hashing and Clustering in Cloud Storage," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 4, pp. 1448-1463, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] R. Venkatesan et al., "Robust Image Hashing," *Proceedings of International Conference on Image Processing*, (Cat. No.00CH37101), Vancouver, BC, Canada, vol. 3, pp. 664-666, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[4] Ling Du et al., "An Image Hashing Algorithm for Authentication with Multi-Attack Reference Generation and Adaptive Thresholding," *Algorithms*, vol. 13, no. 9, pp. 1-17, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Chun-Shien Lu, and H.Y.M. Liao, "Structural Digital Signature for Image Authentication: An Incidental Distortion Resistant Scheme," *IEEE Transactions on Multimedia*, vol. 5, no. 2, pp. 161-173, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[6] S.S. Kozat, R. Venkatesan, and M.K. Mihcak, "Robust Perceptual Image Hashing via Matrix Invariants," *International Conference on Image Processing*, ICIP '04., Singapore, vol. 5, pp. 3443-3446, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[7] V. Monga, A. Banerjee, and B.L. Evans, "Clustering Algorithms for Perceptual Image Hashing," *3rd IEEE Signal Processing Education Workshop, IEEE 11th Digital Signal Processing Workshop*, Taos, NM, USA, pp. 283-287, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jian Shen et al., "Block Design-Based Key Agreement for Group Data Sharing in Cloud Computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 6, pp. 996-1010, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Wen Xia et al., "A Comprehensive Study of the Past, Present, and Future of Data Deduplication," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1681-1710, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[10] Yinjin Fu et al., "Application-Aware Big Data Deduplication in Cloud Environment," *IEEE Transactions on Cloud Computing*, vol. 7, no. 4, pp. 921-934, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart, "Message-Locked Encryption and Secure Deduplication," *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 7881, pp. 296-312, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[12] H.S. Gunawi et al., "Deconstructing Commodity Storage Clusters," *Proceedings of the 32nd International Symposium on Computer Architecture*, pp. 60-71, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[13] William J. Bolosky et al., "Single instance storage in Windows® 2000," *Proceedings of the 4th Conference on Usenix Windows Systems Symposium*, pp. 13-24, 2000. [Google Scholar] [Publisher Link]

[14] Tim D. Moreton, Ian A. Pratt, and Timothy L. Harris, "Storage, Mutability and Naming in Pasta," *Proceedings of International Conference on Research in Networking*, vol. 2376, pp. 215-219, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[15] L.L. You, K.T. Pollack, and D.D.E. Long, "Deep Store: An Archival Storage System Architecture," *21st International Conference on Data Engineering*, Tokyo, Japan, pp. 804-815, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[16] Athicha Muthitacharoen, Benjie Chen, and David Mazières, "A Low-Bandwidth Network File System," *Proceedings of the 18th ACM Symposium on Operating Systems Review*, vol. 35, no. 5, pp. 174-187, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[17] Lei Zhang, and Jun Ma, "Image Annotation by Incorporating Word Correlations into Multi-Class SVM," *Fifth International Conference on Natural Computation*, Tianjian, China, pp. 516-520, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[18] Chenggang Yan et al., "Deep Multi-View Enhancement Hashing for Image Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 4, pp. 1445-1451, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Chenggang Yan et al., "3D Room Layout Estimation from a Single RGB Image," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 3014-3024, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Chenggang Yan et al., "Depth Image Denoising Using Nuclear Norm and Learning Graph Model," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 16, no. 4, pp. 1-17, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Subiman Chatterjee, and Kishor Sarawadekar, "An Optimized Architecture of HEVC Core Transform Using Real-Valued DCT Coefficients," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 12, pp. 2052-2056, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] Ch. Prathima, and L.S.S. Reddy, "A Survey on Efficient Data Deduplication in Data Analytics," *Soft Computing and Medical Bioinformatics, Springer Briefs in Applied Sciences and Technology*, Springer, Singapore, pp. 103-113, 2019. [CrossRef] [Google Scholar] [Publisher Link]