Finding Similar and Duplicate Images

Image searching using Image Hashing Technique

What is Image Hashing

1) Image hashing is the process of assigning a unique hash value to an image using an algorithm.
2) The hash value of duplicate copies of the picture is the same. The term 'digital fingerprint' is also used in certain circumstances.
3) Basically, the image hashing process is a way to reduce huge amounts of data into short numbers that can be used to identify the image.
4) Image hashing is the process of:-
    a) Examining the contents of an image.
    b) Constructing a hash value that uniquely identifies an input image based on the contents of an image

Difference in Images

Images that look identical to us, can be very different if you will just compare the raw bytes.

This can be due to:

- different formats
- minor noise, watermarks, artifacts
- change in brightness level
- rotation
- slightly different color gamma

Simple Hashing

The following functions map a single integer key (k) to a small integer bucket value h(k). m is the size of the hash table (number of buckets).

Division method (Cormen) Choose a prime that isn't close to a power of 2. h(k) = k mod m.

Steps in simple hashing:-

- Step 1: Represent the key in numerical form
- Step 2: Fold and Add
- Step 3: Divide by a prime number and use the remainder as the address.

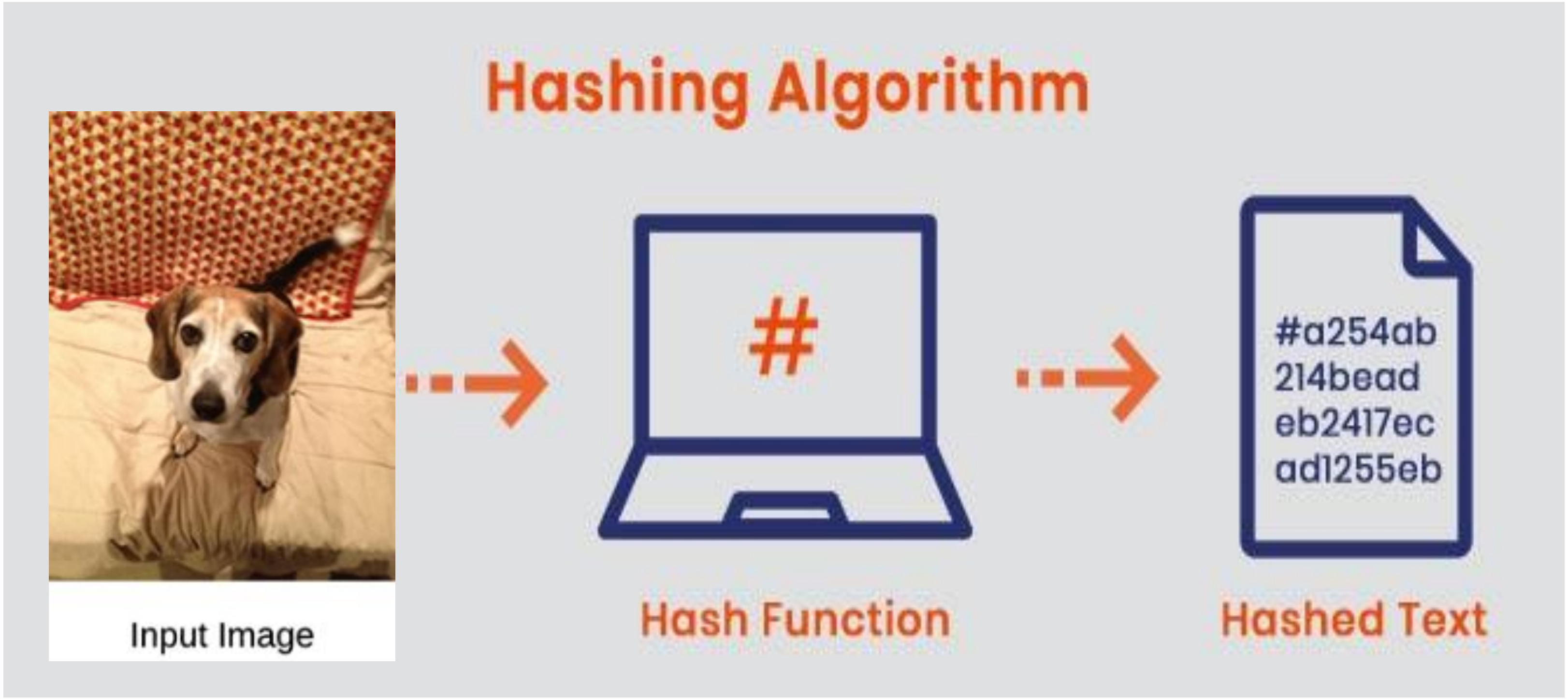
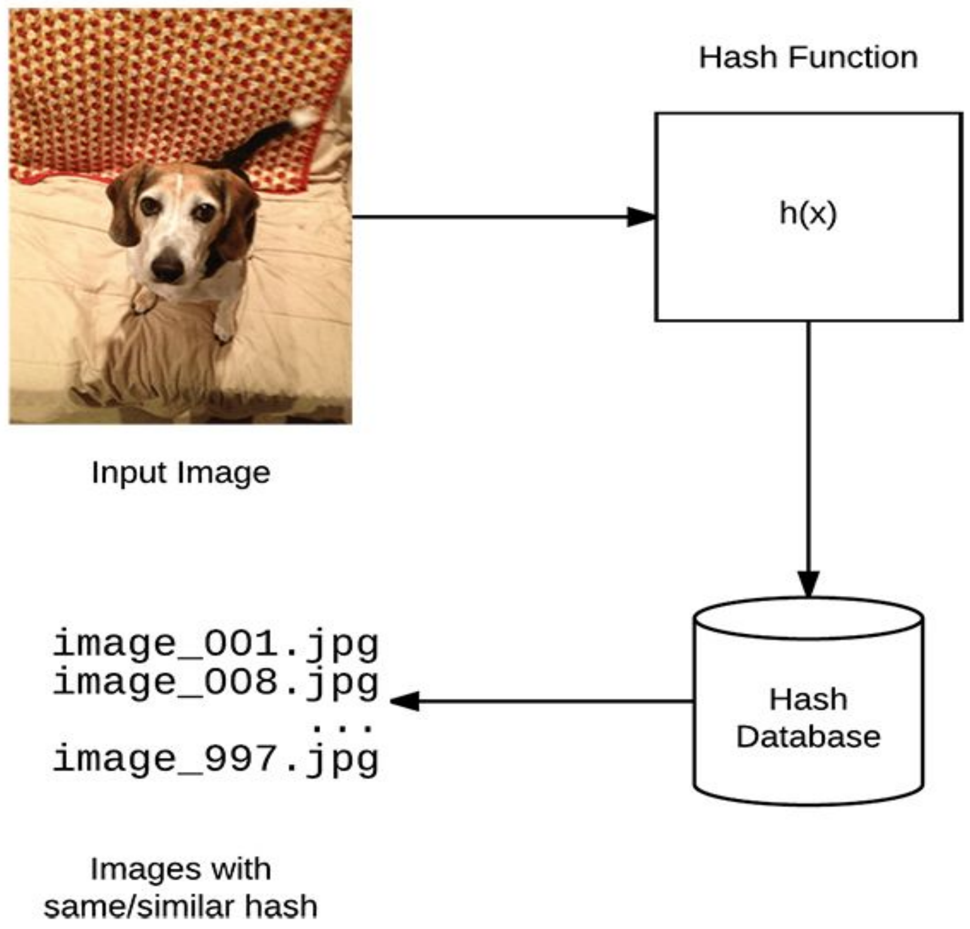

Hashing Algorithm
Input Image
#
Hash Function
#a254ab
214bead
eb2417ec
ad1255eb
Hashed Text

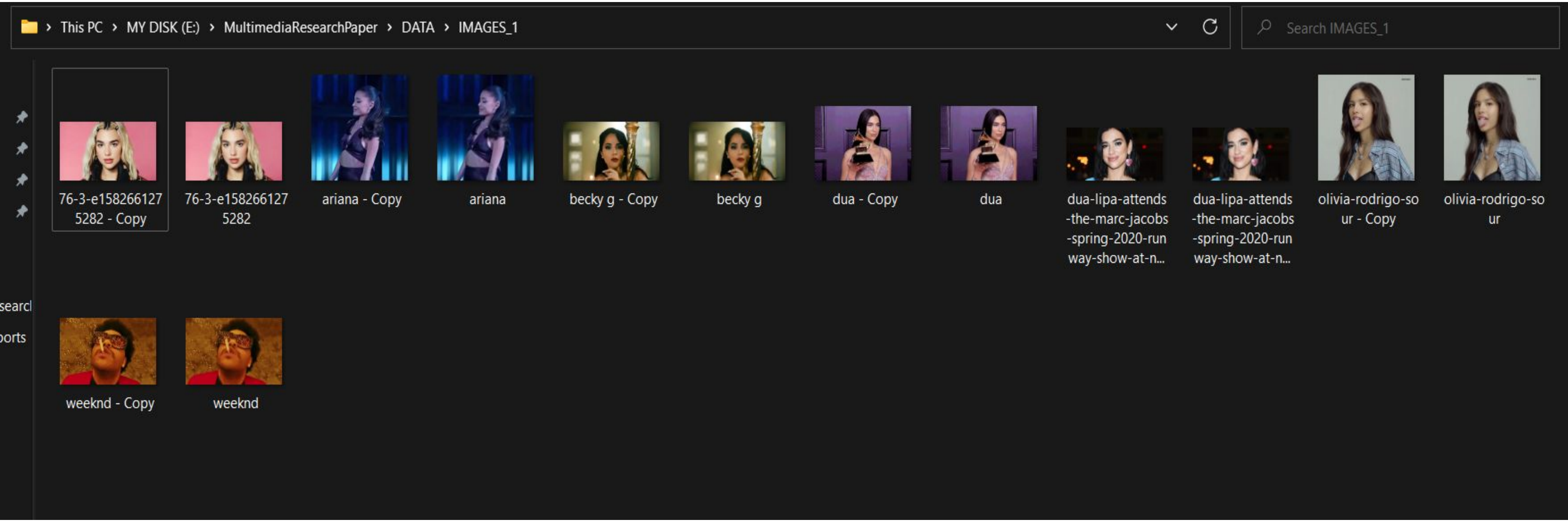
Image Searching using image hashing

An image search engine, you present a query image (not a textual word/phrase). The image search engine then returns similar image results based solely on the contents of the image. To find near-duplicate images, our original image hashing method would require us to perform a linear search, comparing the query hash to each individual image hash in our dataset.

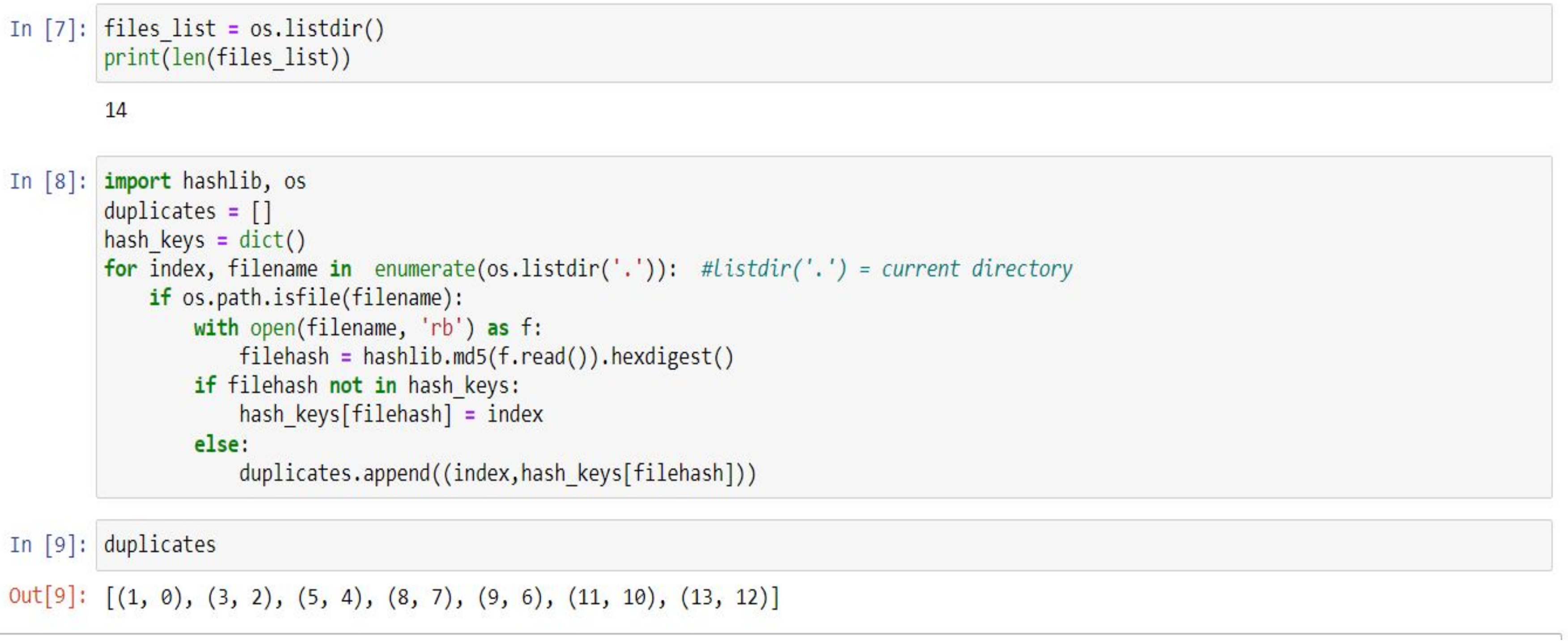
To build a search engine, we must implement the following steps;

1. Setting up an image database.
2. Index images using hashing.
3. Fetch similar images by comparing hash values of a query image and images in the database. Similar images are fetched based on the similarity score of hashes.


Hash Function
h(x)
Input Image
Hash Database
image_001.jpg
image_008.jpg
...
image_997.jpg
Images with same/similar hash

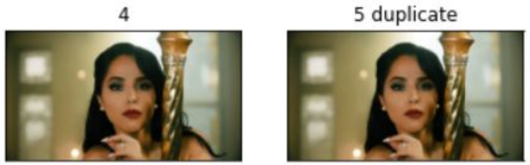

This PC › MY DISK (E:) › MultimediaResearchPaper › DATA › IMAGES_1
Search IMAGES_1
searcl
ports

76-3-e1582661275282 - Copy
76-3-e1582661275282
ariana - Copy
ariana
becky g - Copy
becky g
dua - Copy
dua
dua-lipa-attends-the-marc-jacobs-spring-2020-runway-show-at-n...
dua-lipa-attends-the-marc-jacobs-spring-2020-runway-show-at-n...
olivia-rodrigo-sour - Copy
olivia-rodrigo-sour
weeknd - Copy
weeknd



```
In [7]: files_list = os.listdir()
        print(len(files_list))

        14

In [8]: import hashlib, os
        duplicates = []
        hash_keys = dict()
        for index, filename in  enumerate(os.listdir('.')):  #listdir('.') = current directory
            if os.path.isfile(filename):
                with open(filename, 'rb') as f:
                    filehash = hashlib.md5(f.read()).hexdigest()
                if filehash not in hash_keys:
                    hash_keys[filehash] = index
                else:
                    duplicates.append((index,hash_keys[filehash]))

In [9]: duplicates

Out[9]: [(1, 0), (3, 2), (5, 4), (8, 7), (9, 6), (11, 10), (13, 12)]
```

```
In [9]: for file_indexes in duplicates[:30]:
            try:

                plt.subplot(121),plt.imshow(imread(files_list[file_indexes[1]]))
                plt.title(file_indexes[1]), plt.xticks([]), plt.yticks([])

                plt.subplot(122),plt.imshow(imread(files_list[file_indexes[0]]))
                plt.title(str(file_indexes[0]) + ' duplicate'), plt.xticks([]), plt.yticks([])
                plt.show()

            except OSError as e:
                continue
```



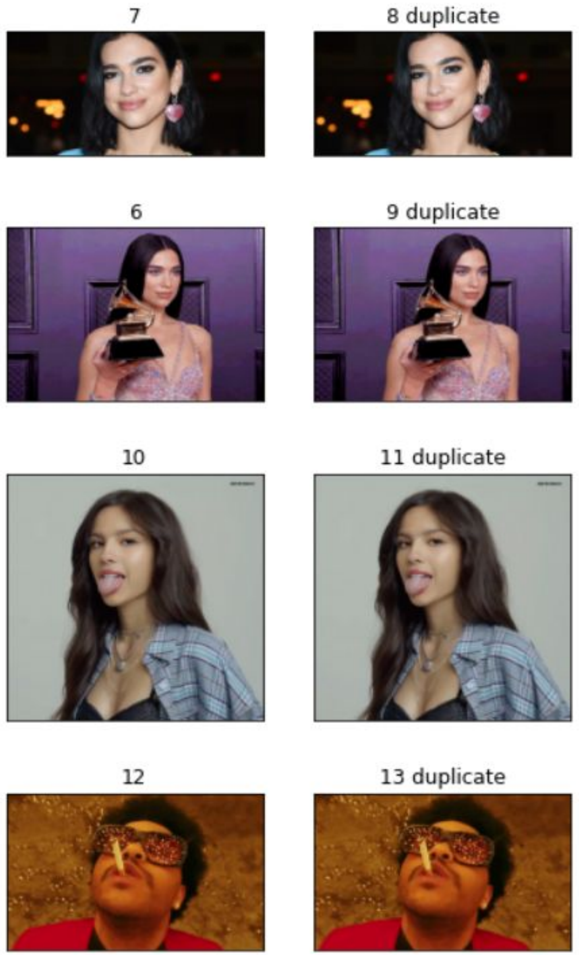
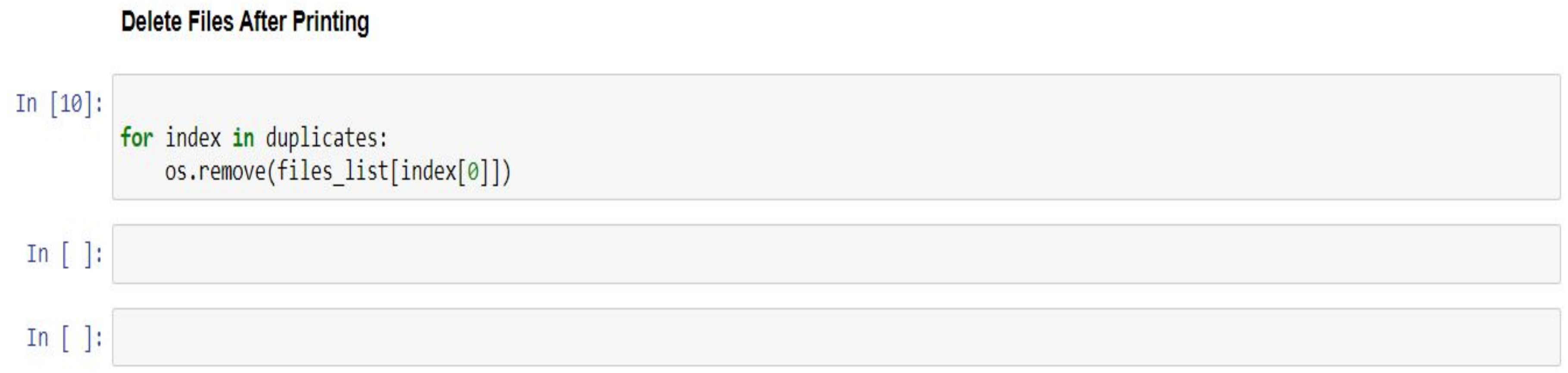

7
8 duplicate
6
9 duplicate
10
11 duplicate
12
13 duplicate

Image Deletion

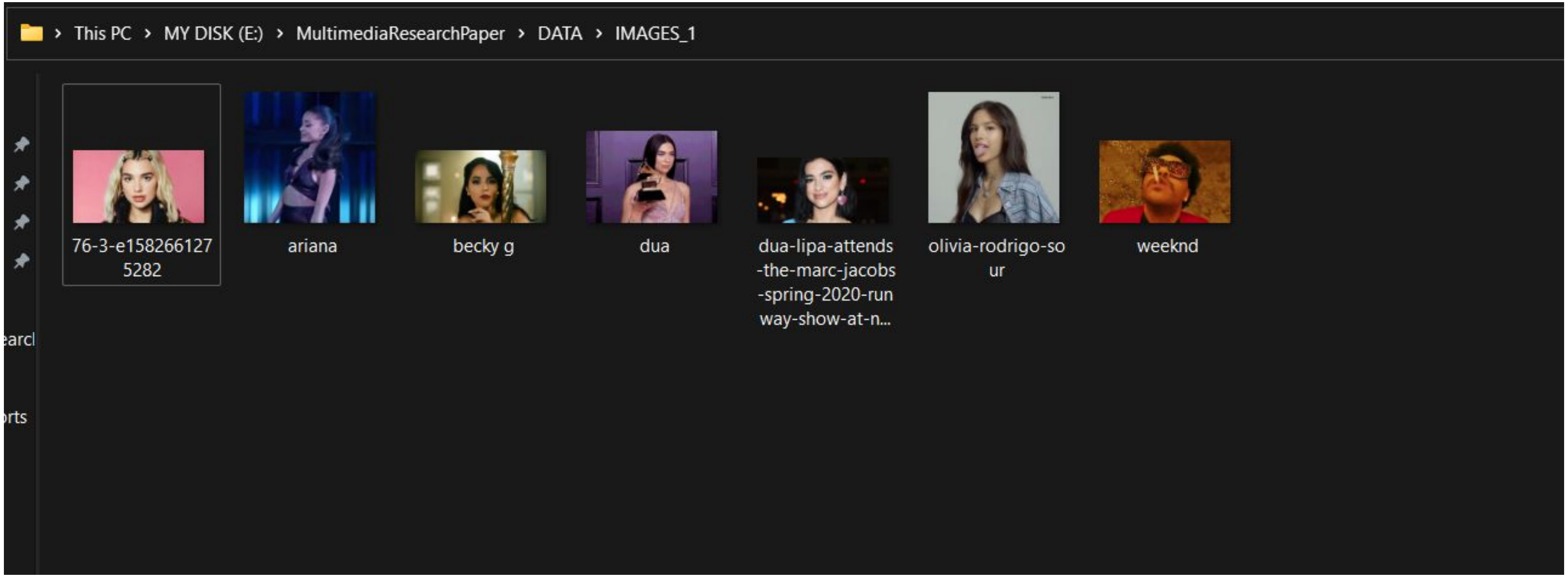
Two images can be visually similar but numerically different. Numerical differences can be caused by various reasons such as the use of social media apps which may change the brightness, contrast, gamma corrections, compression, resolution, and/or scaling.

From a visual perspective, it is difficult to observe the changes between original image and a duplicate image. So, we use hashing to find the duplicated photos and delete the copies or duplicates of the original photo.



Delete Files After Printing

```
In [10]:
for index in duplicates:
    os.remove(files_list[index[0]])

In [ ]:

In [ ]:
```



This PC › MY DISK (E:) › MultimediaResearchPaper › DATA › IMAGES_1

76-3-e1582661275282

ariana

becky g

dua

dua-lipa-attends-the-marc-jacobs-spring-2020-runway-show-at-n...

olivia-rodrigo-sour

weeknd

The End

Akhil Balineni, Ananth Sai Valluru, Kalyan Reddy Bezawada, Chaitanya Nitin Suda, Bharath Chandra Kongara