PROBLEM STATMENT

World Wide Web contains billions of images. User browsing the internet will quickly encounter duplicate images in multiple locations. Duplicate image detection is done for reducing storage space, understanding behavior and interest of user and for copyrights. Duplicates can be exact duplicates, global duplicates or near duplicates.

In this Work a method has been proposed for detection of duplicate images using Hashing Algorithm.

Near duplicate images are detected based on selected user query image and retrieving the near duplicate image based on indexing. This process is achieved by four steps firstly Image enhancement, secondly Feature extraction by SIFT and finally it detects the duplicate images using hashing algorithm.

**Dataset Used:**

* Total number of images : 3000
* Image format: JPEG
* Images of Flower spices
* Downloaded from Kaggle

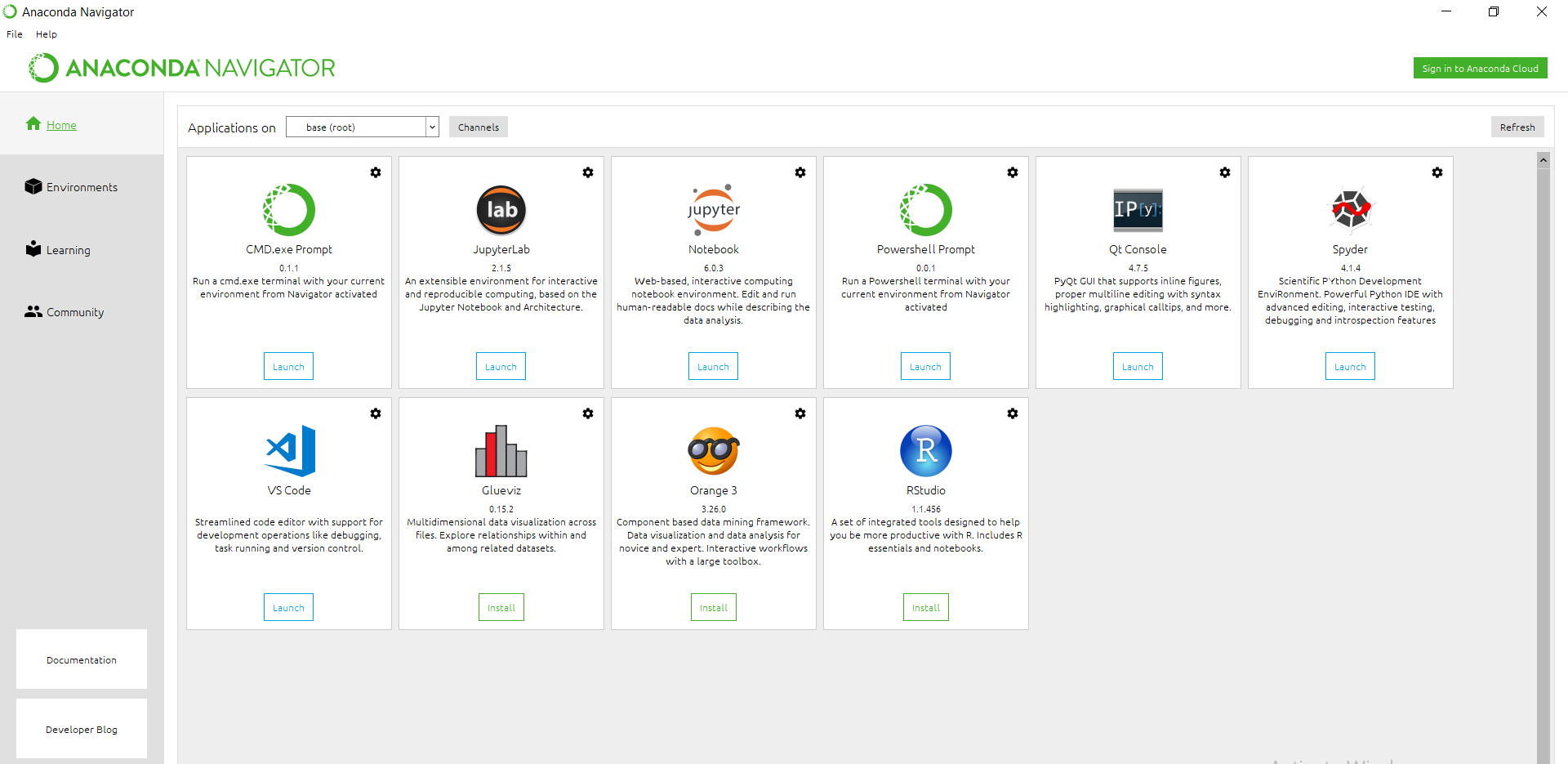
(https://www.kaggle.com/olgabelitskaya/flower-color-images)

**DESGIN AND IMPLEMENTATION**

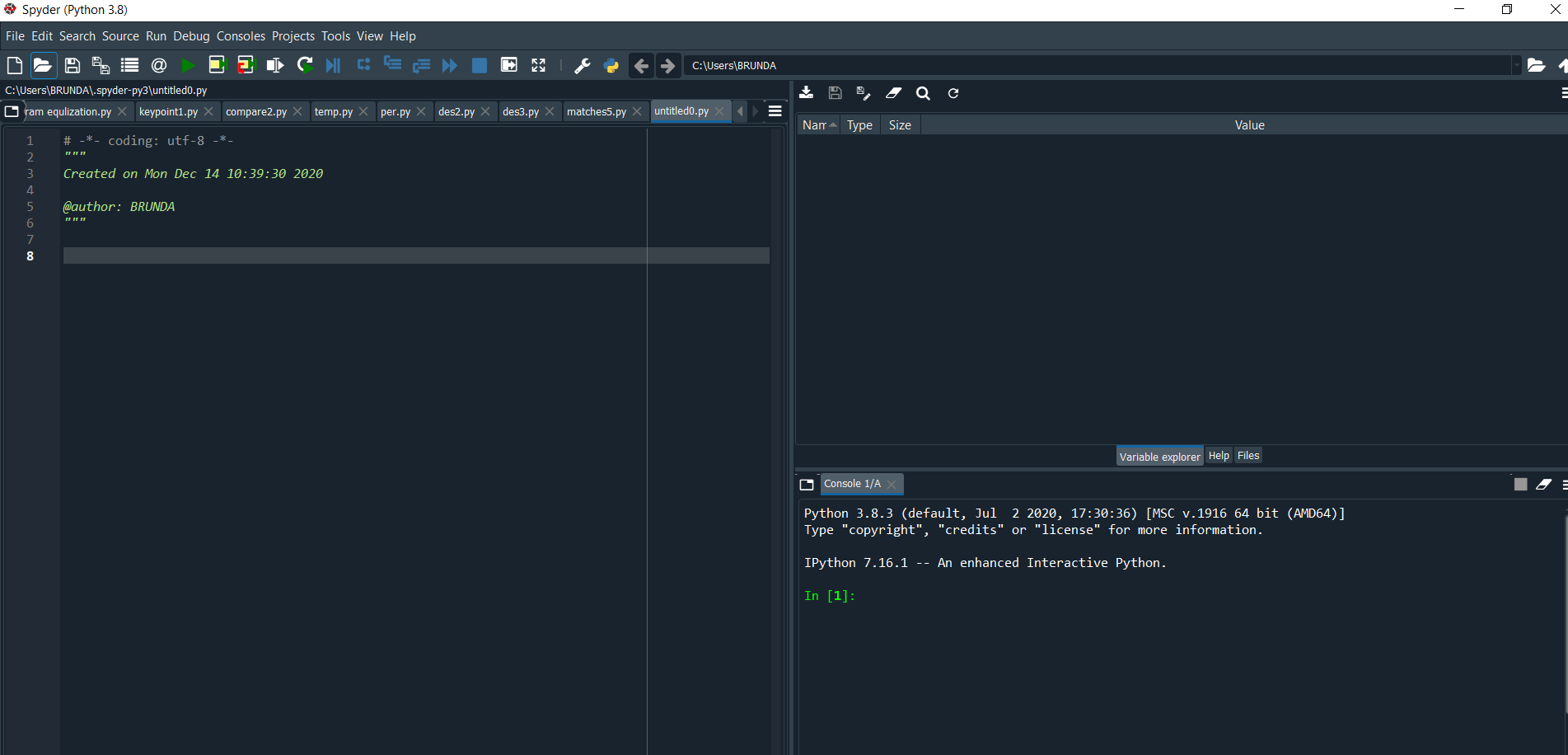
**5.1 Environmental setup:**

* Used Anaconda Navigator in that Spyder and Jupyter notebook to run and implement the project

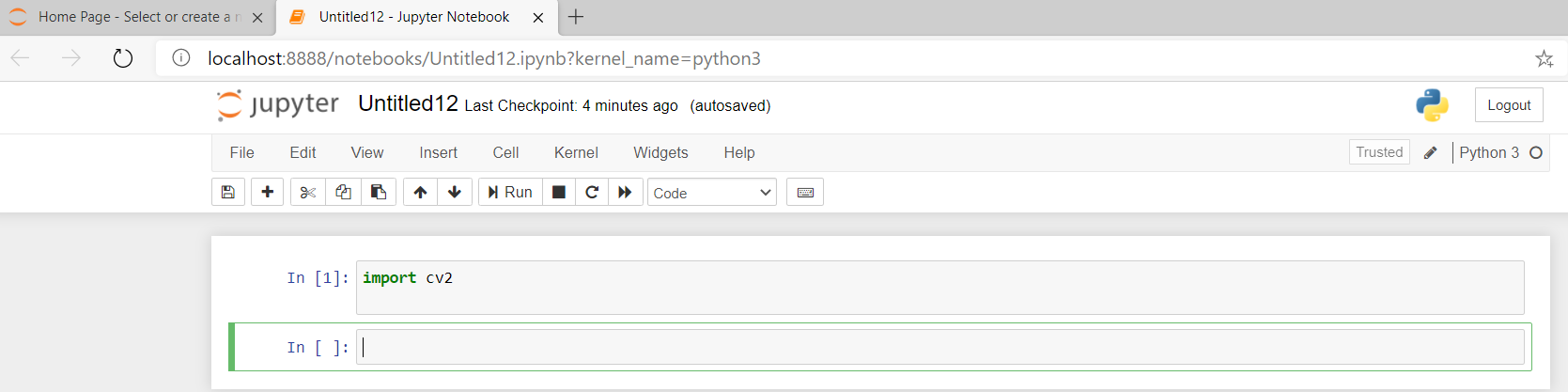
Use Python package and install other necessary libraries.



**Terminal (Spyder)**



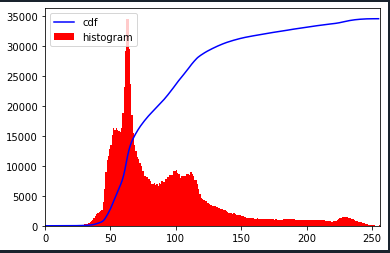
**Terminal (Jupyter)**



**Work-flow / Implementation Steps**

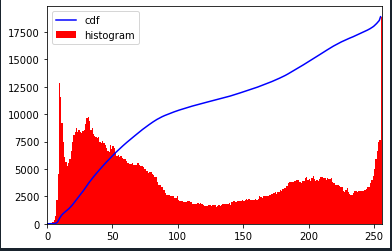
1. **Histogram equalization:**

* **Original image**





* **Duplicate image**

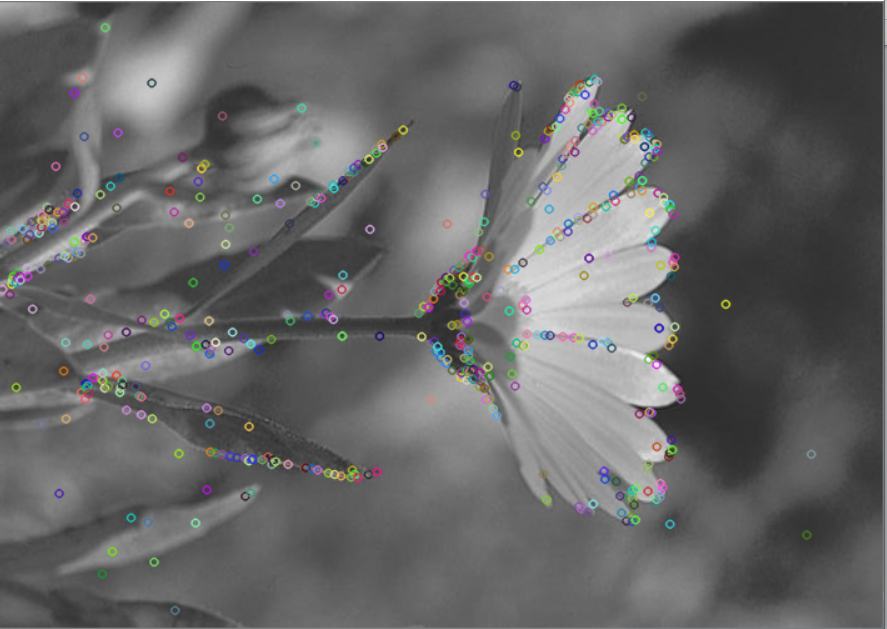




1. **SIFT**

* **Keypoints**

**Original and Smilar duplictae image**



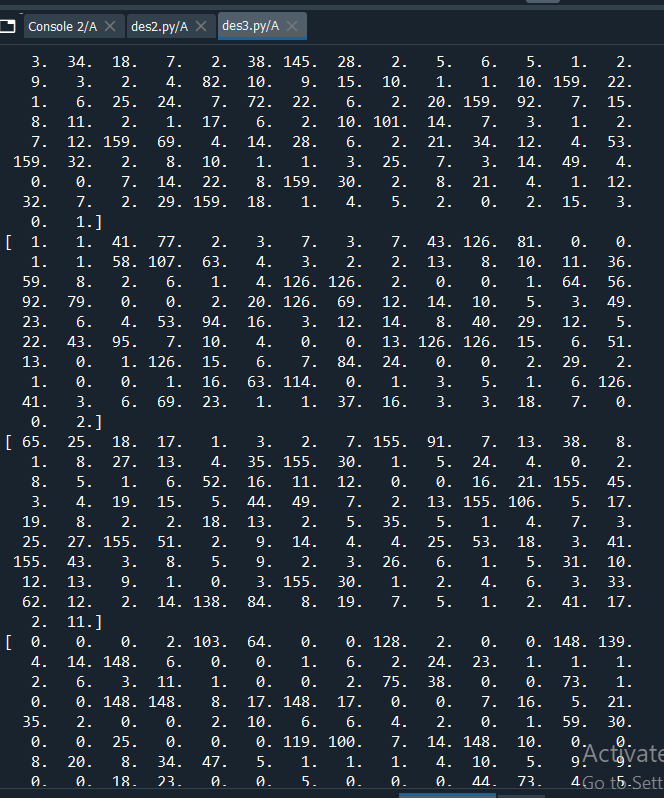
(original image) (Similar duplictae image)

**Original and Dissimilar image**

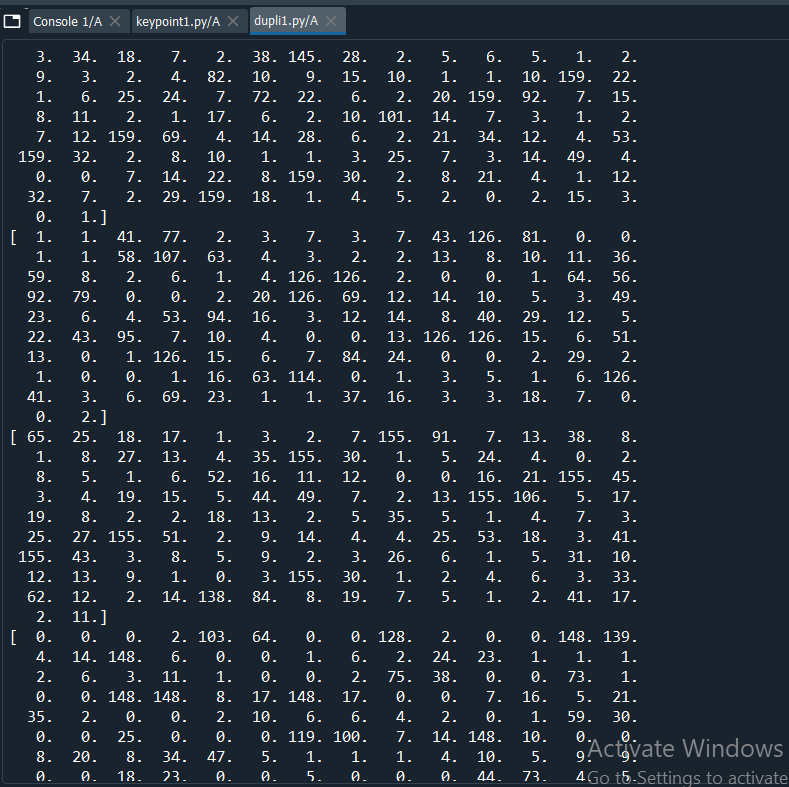
 

(Original image) (**Dissimilar** image)

* **Descriptors for Original and Similar duplicate images**

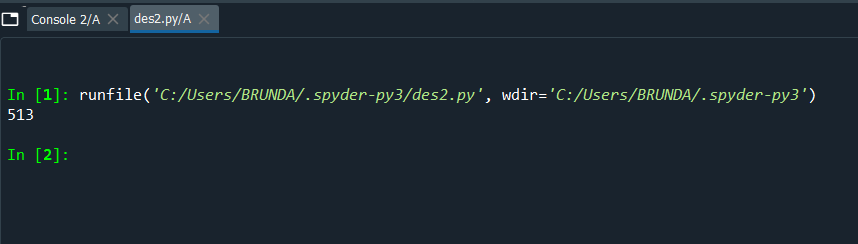


* **Descriptors for Original and Dissimilar images**



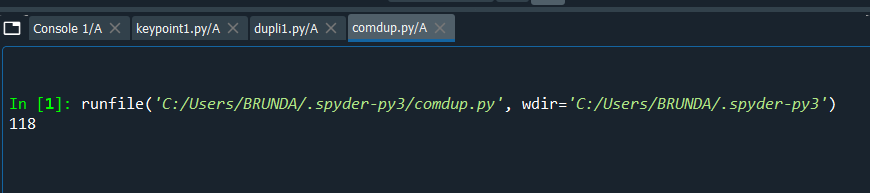
* **Number of Matches for Original and Similar duplicate images**

**513 matches**



* **Number of Matches for Original and Dissimilar images**

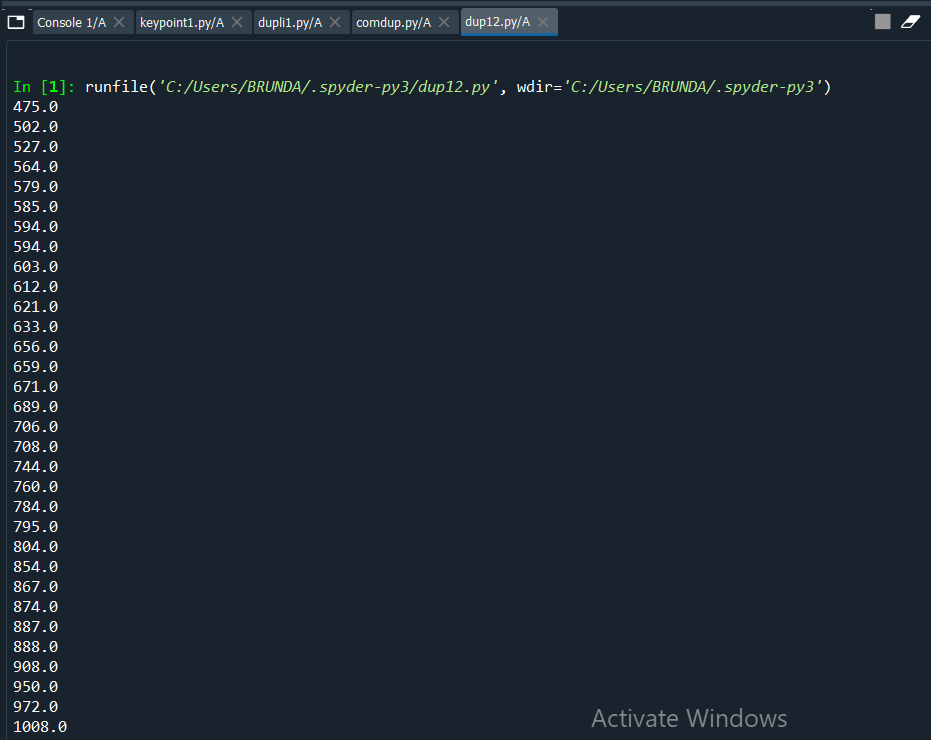
**118 matches**



* **Matches for Original and Similar duplicate image**

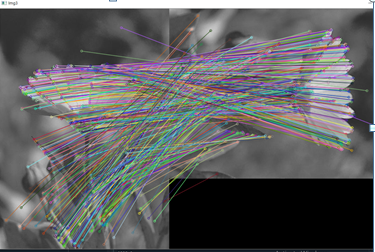


* **Matches for Original and Dissimilar image**



* **Matching for Original and Similar duplicate image**

**513 matches**



* **Matching for Original and Dissimilar image**

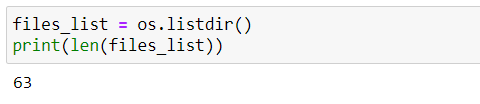
**118 matches**



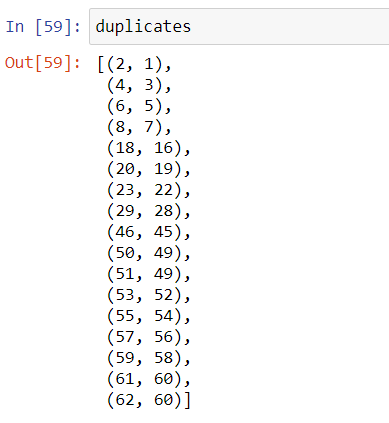
**Results and Discussion**

**OutPut:**

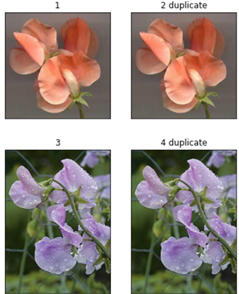
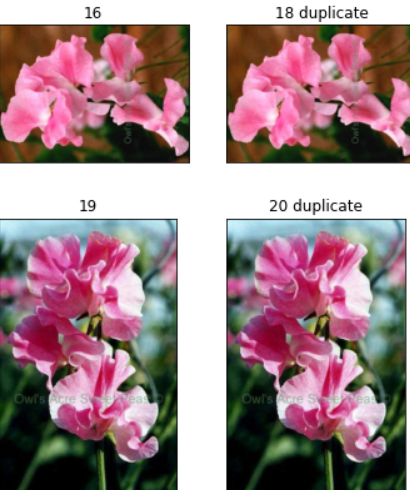
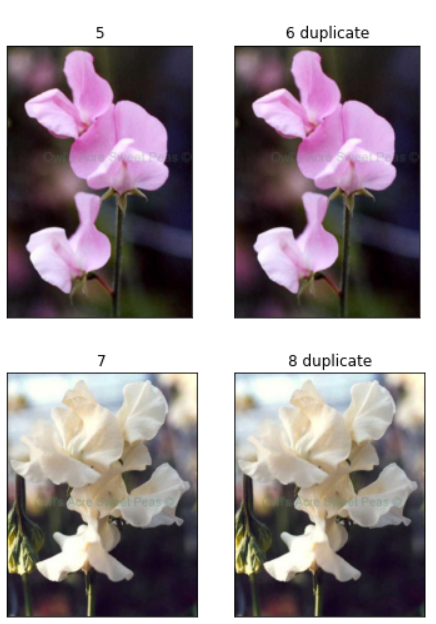
* **Number of Images in a File:**

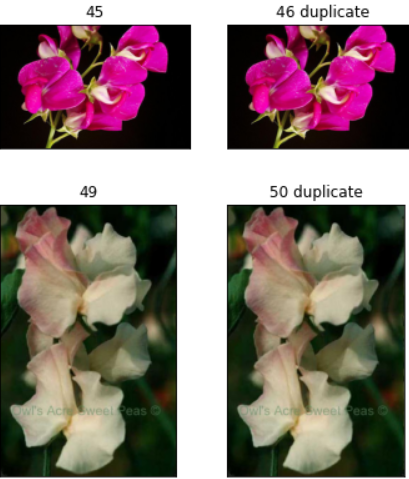
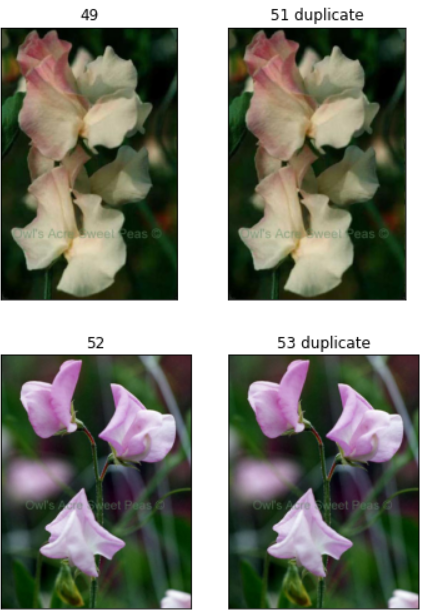


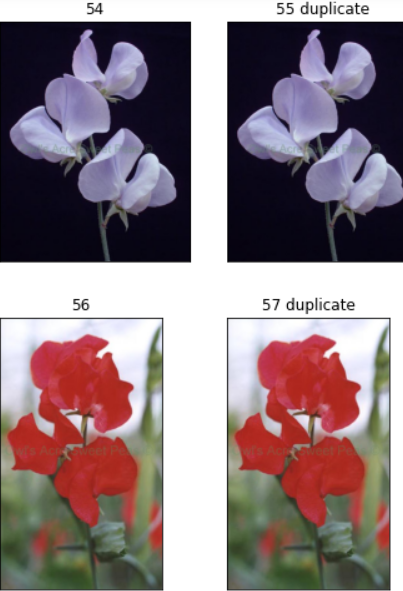
* List of Duplicate images



* Duplicate images found

**CONCLUSION**

The overall work here is detecting near duplicate images and identifying those images from a collection of dataset. In this work, methodology is presented to perform identifying and displaying of near-duplicate images using Hashing algorithm. Initially, the query is passed by the user to the search engine and the search engine results set of query related images. These images are collection of duplicate as well as near-duplicate images. Here we concentrate in detecting near-duplicate images and identify those images. This is done using following steps – initially enhance the user query image and then features are extracted. After features are extracted similarity is measured and finally identified the near

Duplicate images. We conclude that our approach is highly effective for collections of up to a few hundred images.