Indian Institute of Information Technology Dharwad

A

Project report On

“VIRTUAL HACKATHON”

For The Course Introduction

To Algorithms SUBMITTED BY

K.BHARATH(18BEC023)

K.YASHWANTH(18BEC020)

SRINIVAS B NAYAKA(18BEC046)

G.HARSHIL(18BEC014)

B.PRAMOD(18BEC007)

ADITI(18BEC002)

SUBMITTED TO

PROF. UMA.S IIIT

DHARWAD

FOR ACADEMIC YEAR

INTRODUCTION

The main objective of the project is to write an algorithm to sort the images using any of the sorting algorithms and find its uses in the real world. This type of project helps in sorting or searching of an image or object in many areas.

DEVELOPMENT TOOLS :

- Python software or google collab.
- Libraries : import os ; import cv2 ; import time ;

SYSTEM REQUIREMENTS :

- Operating system(any).
- Web browser.
- Python programming software.

FEATURES :

- User friendly.
- Can be used for multiple images.
- Faster.
- Depends only on object size.

Methodology

1. Uploading all the images to colab as we are using that here.
2. Finding its vectorized form, size and name and appending it into a list.
3. **Quick sorting** the images in the list in ascending order of size.
4. Showing the sorted list.
5. Using cv2.split and cv2.subtract functions to compare the pixel values of searched image with images in list using **linear search**.

Brief introduction to application.

1. Finding duplicate image:

Let’s say there are multiple products in a store with same id number (which should not be like that), we can take picture of all products and find the pair of products with same id number. As we are using cv2.split function to split them into RGB and then compare them with each other if they have same pixel values. It exactly finds pixel values and hence the problem is solved.

2. Sorting helps in time management

Let’s say we have to share a photo of a product. If we have less data photo it can be easily send to different parts of the company with less time to process.

3. Finding fake medicines: Let’s think that there is a fake medicine producing factory which supplies products using similar logo. We can

check logo and compare it to find out whether they are fake or not.

Time complexity for sorting:

Worst-case performance	O(n2)
Best-case performance	O(n log n)
Average performance	O(n log n)

Time complexity for linear search:

O(n) for all the cases.

Extra initiative

We have added a function which calculates the pixel values and finds the difference of pixel values between the searching images. If the difference is zero, we found the
image. We have also added time function, to calculate execution time of search and sort.

Conclusion

By using this code we can efficiently sort and search any number of images, without any drawback.