



Expt_10

Pooja Naskar-30

Aim: To Create Program to perform a retrieving Image and Searching.

Objective:

The fundamental need of any image retrieval model is to search and arrange the images that are in a visual semantic relationship with the query given by the user.

Most of the search engines on the Internet retrieve the images based on text-based approaches that require captions as input.

Theory:

Creating an image retrieval and search program involves a number of important steps. Make sure your image dataset is well-structured for access before gathering and organizing it. Next, choose which characteristics—such as texture features, color histograms, or deep learning-based features—you wish to employ to represent photos. After these features are extracted from your information, create a database to hold them and any relevant metadata, like file paths or image descriptions. Provide a querying interface where users may enter a search image and have it preprocessed to extract features

To find out how similar the characteristics in your database are to those in the query image, use the appropriate similarity metrics. Sort the images based on how similar they are to the search image to determine the top k images. If you want your system to be user-friendly, think about creating a user interface, which could be a desktop application, online application, or command-line tool. The results, together with the top k similar photos and their corresponding metadata, ought to be visible to users.

When working with large datasets, it is extremely important to maximize the performance of your system by using indexing or approximate search algorithms. To verify the efficacy of your system, thoroughly test and evaluate it using a range of query photos utilizing evaluation metrics such as precision, recall, and F1-score. Think about.



Code:

```
import cv2

import numpy as np

import os

from IPython.display import Image, display

def calculate_histogram(image):

    hist = cv2.calcHist([image], [0, 1, 2], None, [8, 8, 8], [0, 256, 0, 256, 0, 256])

    hist = cv2.normalize(hist, hist).flatten()

    return hist

def chi_squared_distance(hist1, hist2):

    return 0.5 * np.sum(((hist1 - hist2) ** 2) / (hist1 + hist2 + 1e-10))

def build_image_database(image_folder):

    image_database = {}

    for image_filename in os.listdir(image_folder):

        image_path = os.path.join(image_folder, image_filename)

        image = cv2.imread(image_path)

        if image is not None:

            hist = calculate_histogram(image)

            image_database[image_filename] = hist

    return image_database

def search_similar_images(query_image_path, image_database, top_k=5):
```



```
query_image = cv2.imread(query_image_path)

query_hist = calculate_histogram(query_image)

similarities = {}

for image_filename, hist in image_database.items():

    similarity = chi_squared_distance(query_hist, hist)

    similarities[image_filename] = similarity

similar_images = sorted(similarities, key=similarities.get)[:top_k]

return similar_images

image_folder = '/content/sample_data/Dataset'

image_database = build_image_database(image_folder)

query_image_path = '/content/sample_data/Dataset/0001.png'

query_image_path = '/content/sample_data/cat1.jpg'

similar_images = search_similar_images(query_image_path, image_database, top_k=5)

for idx, image_filename in enumerate(similar_images):

    similar_image_path = os.path.join(image_folder, image_filename)

    display(Image(filename=similar_image_path))
```



Output:

Input_Image_1



Output_1





Input_Image_2



Output_2





Conclusion:

The primary goal of an image retrieval and search program is to enable users to find images in a dataset that visually resemble a query image. Numerous techniques, including text-based search, content-based image retrieval (CBIR), and combinations of the two, can be used to accomplish this. One may build a basic image retrieval and search engine using a color histogram and basic similarity measures.

We were able to successfully implement the program for retrieving and searching images in this experiment.