

Department of Computer Engineering

56-Sahil Pawar

To create program to perform a retrieving Images and Searching

Date of Performance: 09/10/2023

Date of Submission: 16/10/2023

NA-VARDHIA

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

56-Sahil Pawar

Aim: To create program to perform a retrieving Images and Searching

Objective: The fundamental need of any image retrieval model is to search and arrange the images that arc 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:

Image Retrieval is a fundamental and long-standing computer vision task that involves finding images similar to a provided query from a large database. It's often considered as a form of fine-grained, instance-level classification. Not just integral to image recognition alongside classification and detection, it also holds substantial business value by helping users discover images aligning with their interests or requirements, guided by visual similarity or other parameters.

Code:-

import os

import numpy as np

from keras.applications.vgg16 import VGG16, preprocess_input

from keras.preprocessing import image

from sklearn.metrics.pairwise import cosine_similarity

import matplotlib.pyplot as plt



Department of Computer Engineering

from PIL import Image, ImageDraw, ImageFont

```
def extract features(image path):
  model = VGG16(weights='imagenet', include top=False)
  img = image.load img(image path, target size=(224, 224))
  img array = image.img to array(img)
  img array = np.expand dims(img array, axis=0)
  img array = preprocess input(img array)
  features = model.predict(img_array)
  features = features.flatten()
  return features
def find similar images(query features, dataset features):
  similarities = {}
  for filename, features in dataset features.items():
     similarity = cosine similarity([query features], [features])[0][0]
     similarities[filename] = similarity
  return similarities
```



Department of Computer Engineering

```
def plot images with similarity(images, similarity ratios, query image path):
  # Load the query image
  query img = Image.open(query image path)
  # Plotting setup
  fig. axs = plt.subplots(1, len(images) + 1, figsize=(15, 5))
  axs[0].imshow(query img)
  axs[0].axis('off')
  axs[0].set title('Query Image')
  # Load and annotate similar images
  for i, (filename, ratio) in enumerate(zip(images, similarity ratios), 1):
     img path = os.path.join(dataset path, filename)
     img = Image.open(img path)
     axs[i].imshow(img)
     axs[i].axis('off')
    axs[i].set title(f'{filename}\nSimilarity: {ratio:.4f}')
  plt.show()
```



Department of Computer Engineering

```
# Path to your dataset
dataset path = "/content/IMG"
# Extract features for all images in the dataset
feature vectors = {}
for filename in os.listdir(dataset path):
  if filename.endswith(".jpg") or filename.endswith(".png"):
     image path = os.path.join(dataset path, filename)
     features = extract_features(image_path)
     feature vectors[filename] = features
# Path to your query image
query image path = "/content/Hyundai-Grand-i10-Nios-200120231541.jpg"
query features = extract features(query image path)
# Find similar images
similarities = find similar images(query features, feature vectors)
# Sort the results by similarity
sorted similarities = sorted(similarities.items(), key=lambda x: x[1], reverse=True)
```



Department of Computer Engineering

Extract filenames and similarity ratios for plotting filenames, similarity ratios = zip(*sorted similarities)

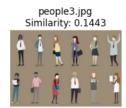
Plot images with similarity ratios
plot_images_with_similarity(filenames, similarity_ratios, query_image_path)

Output:-











Conclusion: In conclusion, the aim of studying the image retrieval using SIFT (Scale-Invariant Feature Transform) descriptors. It extracts distinctive features from a query image and a dataset of images, matches these features, and ranks images in the dataset based on the number of matching features. The program can be useful for applications like content-based image search and recommendation. While the code here uses SIFT, more advanced methods and deep learning can enhance retrieval accuracy.