

HW5 Report

1.

Browser address bar: https://colab.research.google.com/drive/1VoCGT99m1L_DMw_iEMykhGp3dnW5B1-g#scrollTo=j-YBYQYtiYOX

Colab interface: Bengisu_Sahin_152120191064_HW5.ipynb

Navigation: Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi

Code editor: + Kod + Metin

Output: 5 dk.

Image: Predicted

Class label of test sample is: BACKGROUND_Google

Right sidebar: sahin.bengisu00@gmail.com

Merhaba Bengisu!

[Google Hesabınızı yönetin](#)

Diğer hesapları göster

Gizlilik Politikası • Hizmet Şartları

Bottom status bar: ✓ 5 dk. 46 sn. tamamlanma zamanı: 19:10

2. This code performs image classification using SIFT (Scale-Invariant Feature Transform) features and a Support Vector Machine (SVM) classifier. It begins by loading necessary libraries and connecting to Google Drive. The `load_dataset` function loads image file paths and their corresponding labels from the specified directory. The dataset is divided into training and test sets, and category names are extracted from the folder names. The images are preprocessed and converted to grayscale, with SIFT features extracted for each image. These features are padded to ensure uniform length and then reduced in dimensionality using PCA. The SVM model is trained on the processed training features and used to predict labels for the test set. The code calculates and prints the accuracy, a detailed classification report, and visualizes the confusion matrix using a heatmap. Finally, a sample image prediction is displayed along with its predicted class label.

```
Bengisu_Sahin_152120191064_HW5.ipynb ☆
Dosya Düzenle Göster Ekle Çalışma zamanı Araçlar Yardım Tüm değişiklikler kaydedildi

+ Kod + Metin

from skimage.feature import SIFT
from sklearn.svm import SVC
from sklearn.feature_selection import SequentialFeatureSelector
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.decomposition import PCA
from google.colab import drive
import cv2
import seaborn as sns
import pandas as pd

drive.mount('/content/drive')

# Define function to load train and test datasets
def load_dataset(path):
    data = load_files(path)
    condition_files = np.array(data['filenames'])
    print(len(condition_files))
    condition_targets = np.array(data['target'])
    return condition_files, condition_targets

# Load train and test datasets
path_org = '/content/drive/MyDrive/PatternRecognition/HW5/'
train_files, y_train = load_dataset(path_org + 'train')
test_files, y_test = load_dataset(path_org + 'test')

# Extract condition names from file paths
condition_names = [os.path.basename(os.path.dirname(item)) for item in sorted(glob(path_org + 'train/*'))]
print(condition_names)

# Print statistics about the dataset
print('There are %d total categories.' % len(condition_names))
print('There are %d training images.' % len(train_files))
print('There are %d test images.' % len(test_files))

def path_to_tensor(img_path):
    img = cv2.imread(img_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (224, 224))
    img_lab = rgb2lab(img)
    img_gray = rgb2gray(img_lab)
    return img_gray

def extract_sift_features(img_path):
    img_gray = path_to_tensor(img_path)
    sift = SIFT()
    sift.detect_and_extract(img_gray)
    descriptors = sift.descriptors
    if descriptors is None:
        return np.zeros((1, 128))
    return descriptors

def output_features(img_path):
```