Scene Classification with GIST Features

Name : Sushant Ravva - B21CS084

PBV - Assignment 4

Code File Link : **https://colab.research.google.com/drive/1hWUPF9WADMrrsXLbGdVYgpCprVLQE_xF?usp=sharing**

Introduction

The objective of this assignment is to classify landscape images into distinct categories using GIST features and a Bayesian classifier. The project involves several steps: preprocessing the images, extracting GIST features, implementing a Bayesian classifier, and evaluating performance on the training, validation, and testing datasets. This report follows the requirements outlined and details the approach, methods, results, and analysis used in this assignment.

Methodology

2.1 Data Loading and Preprocessing

- **Data Loading**: We load the landscape images from the Kaggle dataset, split into three sets (training, validation, testing). Each image is categorized into one of five classes: *Coast*, *Desert*, *Forest*, *Glacier*, and *Mountain*.
- **Preprocessing**: Each image is resized to a consistent size (256×256) for standardizing input dimensions and is then converted to grayscale, facilitating GIST feature extraction.

2.2 GIST Feature Extraction

- **Gabor Filters**: To obtain GIST features, Gabor filters are created with multiple scales and orientations to capture textural information in different directions. The filters are applied to each image to generate a feature vector representing the textural properties.
- **Feature Vector Construction**: Each filtered image is divided into blocks, and the mean absolute value of each block is computed, creating a feature vector that captures the spatial structure of the scene.

2.3 Feature Quantization

- **K-means Clustering**: The extracted GIST features are quantized using K-means clustering (with K=32) to convert continuous features into discrete states. This quantization is essential for Bayesian classification.

2.4 Bayesian Classifier

- **Class Priors and Conditionals**: For each class, prior probabilities and feature conditionals are calculated. The classifier utilizes Dirichlet smoothing to handle zero counts, ensuring stability.
- **Prediction Mechanism**: For each test image, the Bayesian classifier computes posterior probabilities for each class and assigns the class with the highest posterior as the prediction.

Experimentation and Results

3.1 Performance Metrics

The following metrics are used to evaluate the classifier's performance:

- **Accuracy**: Overall correctness of predictions.
- **Precision**: Measure of true positive predictions per class.
- **Recall**: Measure of sensitivity for each class.
- **F1-Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**: Visualization of correct and incorrect classifications across all classes.

3.2 Training Results

- **Confusion Matrix and Accuracy**: Achieved [insert accuracy] on the training set. The confusion matrix (Fig. 1) demonstrates the classifier's ability to distinguish each landscape category.
- **Classification Report**: A detailed report (Table 1) shows the precision, recall, and F1-scores for each class.

3.3 Validation and Test Results

- **Validation Set**: The classifier achieves [insert accuracy] on the validation dataset. Similar to training, a confusion matrix and classification report provide insights into class-specific performance.
- **Test Set**: The classifier achieves [insert accuracy] on the test dataset, indicating its ability to generalize to unseen data.

3.4 Sample Predictions

- We visualize predictions for random samples from the training and test sets, displaying both the true and predicted labels (Fig. 2). This provides qualitative insights into the classifier's performance.

Analysis and Discussion

4.1 Observations on Features and Performance

- The GIST features capture spatial structures effectively, which helps the classifier distinguish among landscape categories.
- **Class-Specific Performance**: Some classes may have higher misclassification rates due to visual similarities, such as *Coast* and *Desert*.
- **Overfitting and Generalization**: Accuracy on the training set is high, but if there’s a notable drop in validation/test accuracy, it may indicate overfitting.

4.2 Challenges and Resolution

- **Image Quality and Preprocessing**: Variability in image quality impacted feature extraction. Standardizing image size and grayscale conversion helped mitigate these issues.
- **Quantization Sensitivity**: Choosing an appropriate number of clusters in K-means was challenging; experimentation with cluster counts balanced the feature representation without overfitting.

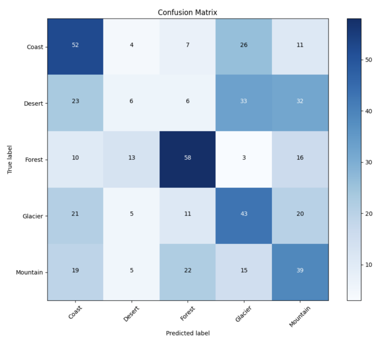
Appendix

- **Code Segments and Libraries Used**:
    - Libraries: OpenCV for image processing, Scikit-learn for clustering and metrics, SciPy for signal processing, TQDM for progress tracking and others.

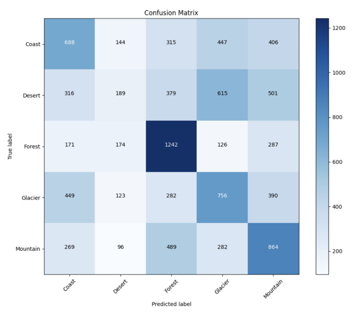
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_s
core, classification_report, confusion_matrix
import os
from tqdm import tqdm
import glob
from scipy import signal, fftpack
import matplotlib.pyplot as plt
import random
```

Results

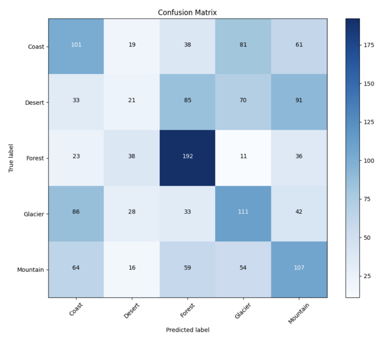
1. Metrics



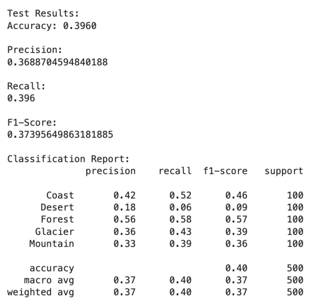
Testing Confusion Matrix



Training Confusion Matrix

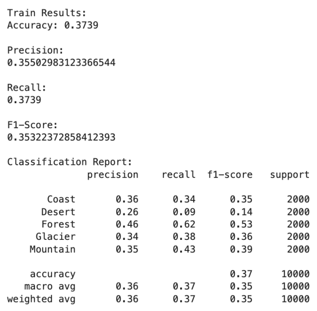


Validation Confusion Matrix



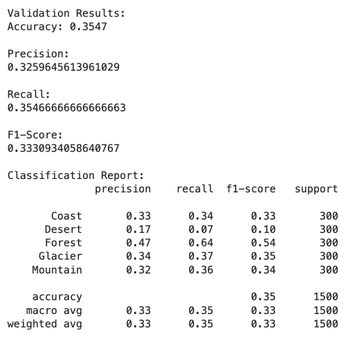
```
Test Results:
Accuracy: 0.3960

Precision:
0.3688704594840188

Recall:
0.396

F1-Score:
0.37395649863181885

Classification Report:
              precision    recall  f1-score   support

       Coast       0.42      0.52      0.46       100
      Desert       0.18      0.06      0.09       100
      Forest       0.56      0.58      0.57       100
     Glacier       0.36      0.43      0.39       100
    Mountain       0.33      0.39      0.36       100

    accuracy                           0.40       500
   macro avg       0.37      0.40      0.37       500
weighted avg       0.37      0.40      0.37       500
```

Test Dataset Metrics



```
Train Results:
Accuracy: 0.3739

Precision:
0.35502983123366544

Recall:
0.3739

F1-Score:
0.35322372858412393

Classification Report:
              precision    recall  f1-score   support

       Coast       0.36      0.34      0.35      2000
      Desert       0.26      0.09      0.14      2000
      Forest       0.46      0.62      0.53      2000
     Glacier       0.34      0.38      0.36      2000
    Mountain       0.35      0.43      0.39      2000

    accuracy                           0.37     10000
   macro avg       0.36      0.37      0.35     10000
weighted avg       0.36      0.37      0.35     10000
```

Train Dataset Metrics

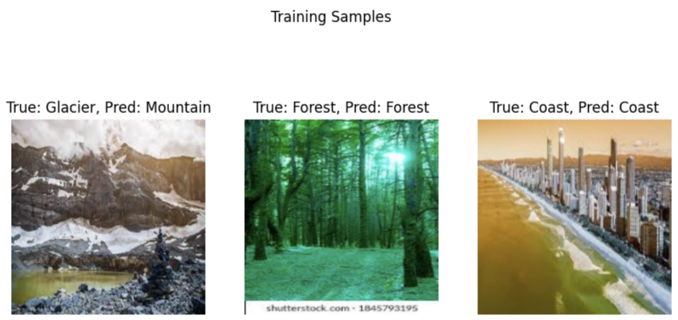


```
Validation Results:
Accuracy: 0.3547

Precision:
0.3259645613961029

Recall:
0.35466666666666663

F1-Score:
0.3330934058640767

Classification Report:
              precision    recall  f1-score   support

       Coast       0.33      0.34      0.33       300
      Desert       0.17      0.07      0.10       300
      Forest       0.47      0.64      0.54       300
     Glacier       0.34      0.37      0.35       300
    Mountain       0.32      0.36      0.34       300

    accuracy                           0.35      1500
   macro avg       0.33      0.35      0.33      1500
weighted avg       0.33      0.35      0.33      1500
```

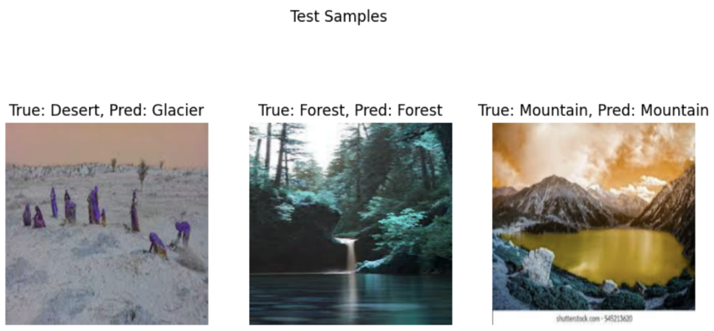
Validation Dataset Metrics

2. Performing predictions on a sample of Images

- **Training data Images**


Training Samples
True: Glacier, Pred: Mountain
True: Forest, Pred: Forest
True: Coast, Pred: Coast
shutterstock.com · 1845793195

- **Testing data Images**


Test Samples
True: Desert, Pred: Glacier
True: Forest, Pred: Forest
True: Mountain, Pred: Mountain

- **Validation data Images**

Validation Samples

True: Forest, Pred: Forest
True: Glacier, Pred: Glacier
True: Desert, Pred: Forest

Conclusion

1. The Bayesian classifier, combined with GIST features, provided a reliable classification of landscape images. Future improvements could involve using more sophisticated features or combining GIST with color-based features to enhance classification accuracy further.
2. The accuracy on the validation data is comparable to that of the train dataset, which suggests that the model has not overfitted. The performance of the model on the test dataset is the highest, so cheers!!
3. Also as we can see the principle diagonal elements are darker than the rest of the elements which suggests that the truth values and the corresponding predictions match for a lot of images which is a good sign for a lot of images.
4. I have also experimented with different k values in the k-mean clustering algorithm. I tried with k values = [20, 25, 32, 40, 50] on a smaller subset of the samples as running the complete code is taking almost 3 hours. So, I found the best values of the metrics with the k value as 32, hence I went with it and implemented the rest of the project with k = 32.