PRINCIPLES OF BIOLOGICAL VISION

Assignment-2

Suyog Gupta (B20AI059)

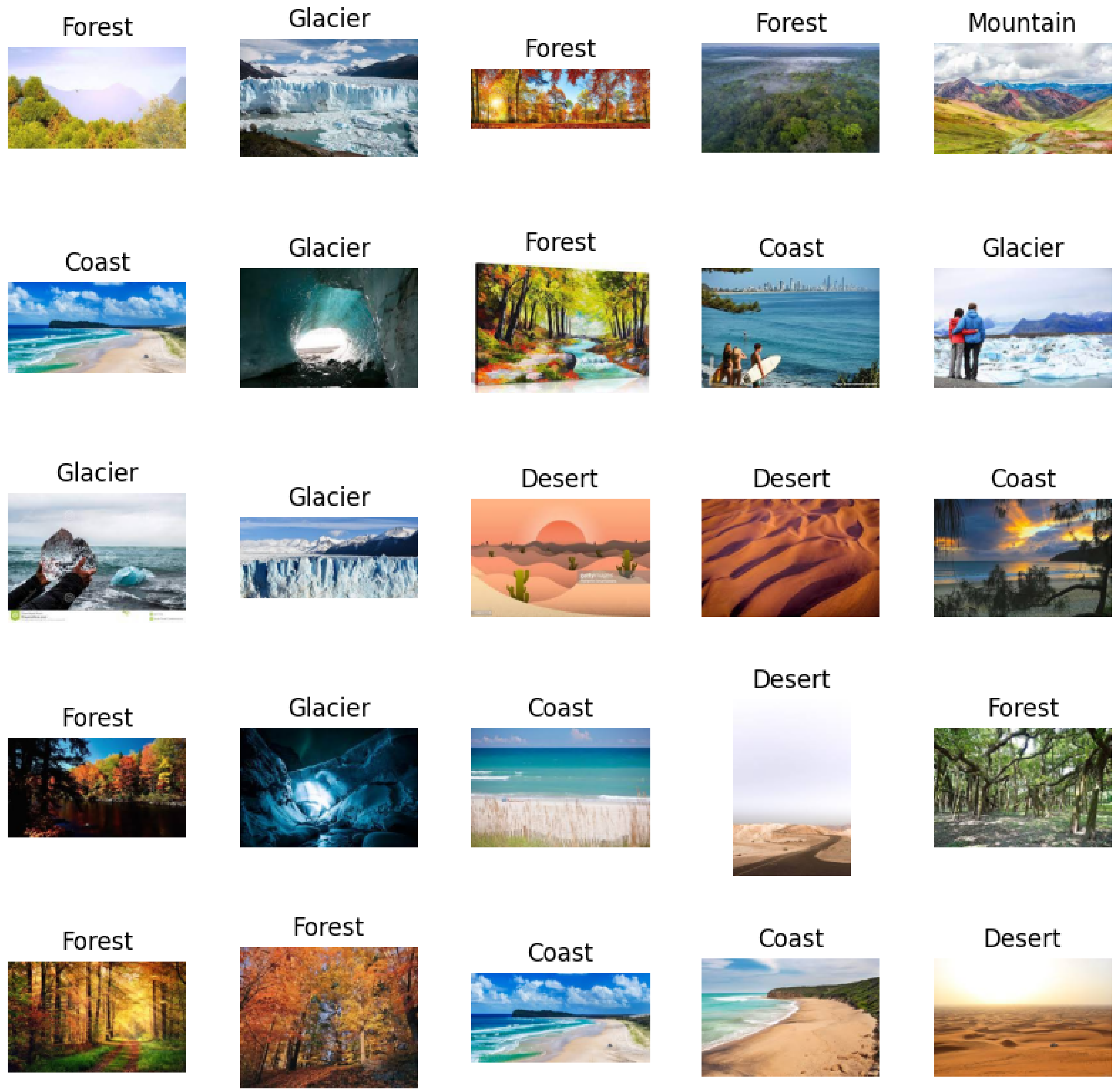
Objective

Landscape recognition is a challenging computer vision problem that involves categorizing images into various natural environment types. In this project, we aimed to develop a classifier capable of distinguishing between different landscape categories using GIST (Global Image Structure) features. GIST features capture the overall spatial layout and texture information in images, making them suitable for holistic context recognition.

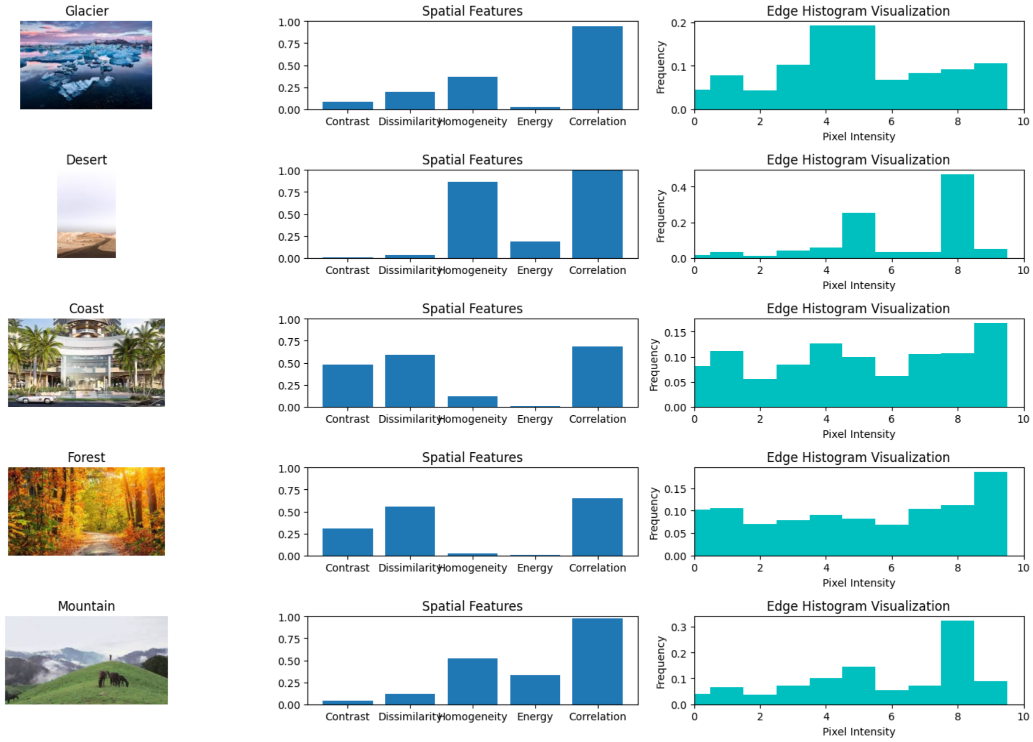
Data Exploration

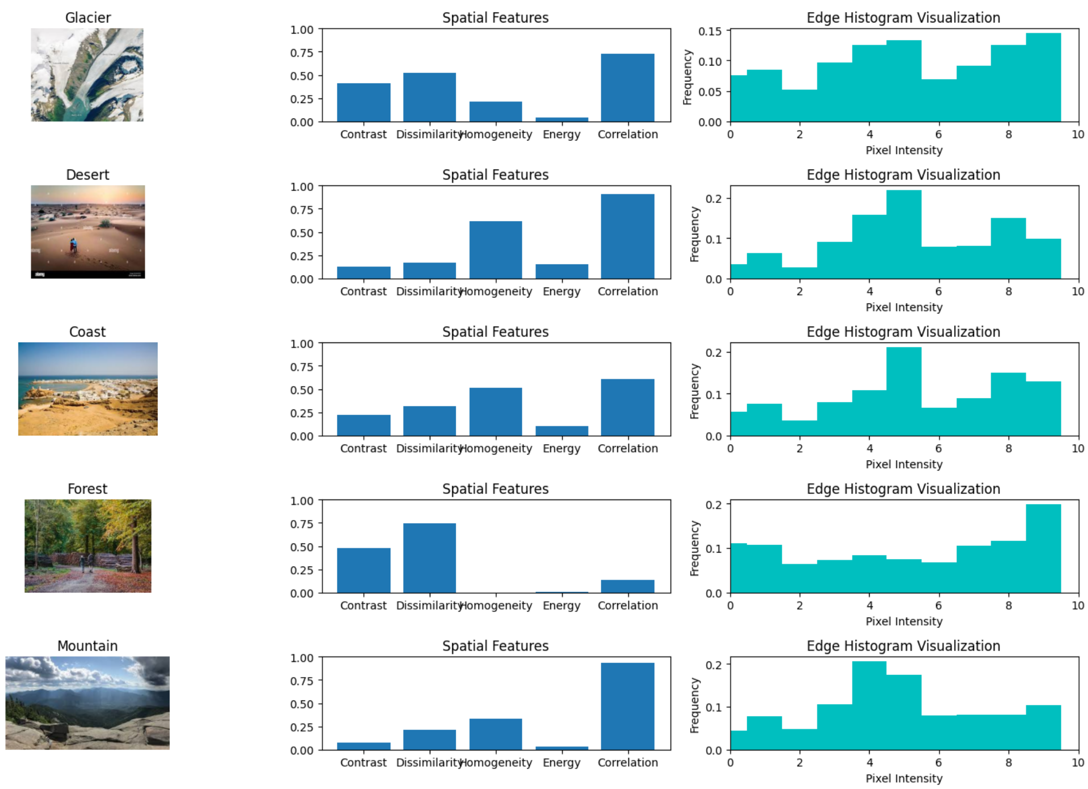
The first step in our project was to load and explore the provided landscape recognition dataset. We aimed to gain a better understanding of the data's characteristics and its distribution among different landscape categories. The dataset contains images of various landscapes, including coast, desert, forest, and more. Here's an overview of what we did:

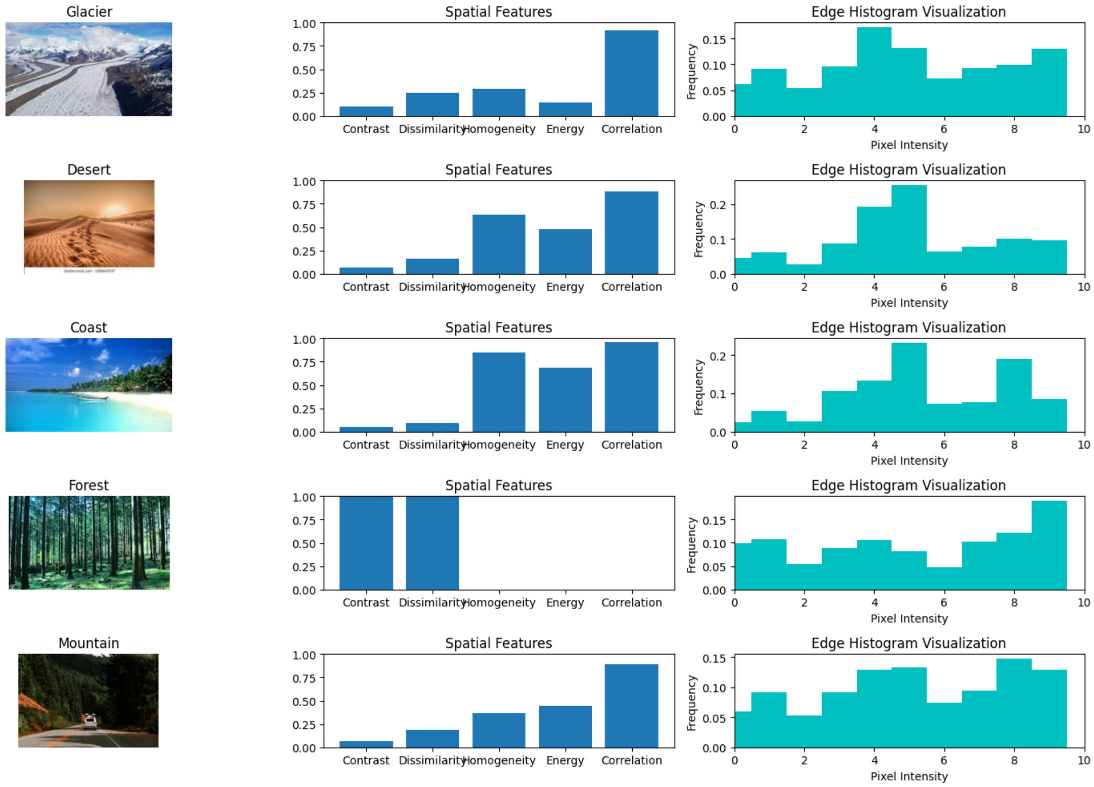
Loading the Dataset: We loaded the dataset into the colab, ensuring that we could access the images and their corresponding labels.



Exploratory Data Analysis (EDA): We conducted an EDA to understand the dataset's structure. This involved checking the number of images in each category, assessing class imbalances, and visualizing sample images from different categories.We also visualized the edge histograms and the spatial features.ie **Contrast, Dissimilarity, Homogeneity, Energy and Correlation** corresponding to image of each class to gain insights.


Glacier
Spatial Features
Contrast
Dissimilarity
Homogeneity
Energy
Correlation
Edge Histogram Visualization
Frequency
Pixel Intensity
Desert
Spatial Features
Edge Histogram Visualization
Coast
Spatial Features
Edge Histogram Visualization
Forest
Spatial Features
Edge Histogram Visualization
Mountain
Spatial Features
Edge Histogram Visualization


Glacier
Spatial Features
Contrast
Dissimilarity
Homogeneity
Energy
Correlation
Edge Histogram Visualization
Frequency
Pixel Intensity
Desert
Spatial Features
Edge Histogram Visualization
Coast
Spatial Features
Edge Histogram Visualization
Forest
Spatial Features
Edge Histogram Visualization
Mountain
Spatial Features
Edge Histogram Visualization



Implementing the Paper

The GIST descriptors were found as described in the paper which follows this methodology-

1. Convolve the image with 32 Gabor filters at 4 scales, 8 orientations, producing 32 feature maps of the same size of the input image.
2. Divide each feature map into 16 regions (by a 4x4 grid), and then average the feature values within each region.
3. Concatenate the 16 averaged values of all 32 feature maps, resulting in a 16x32=512 GIST descriptor.

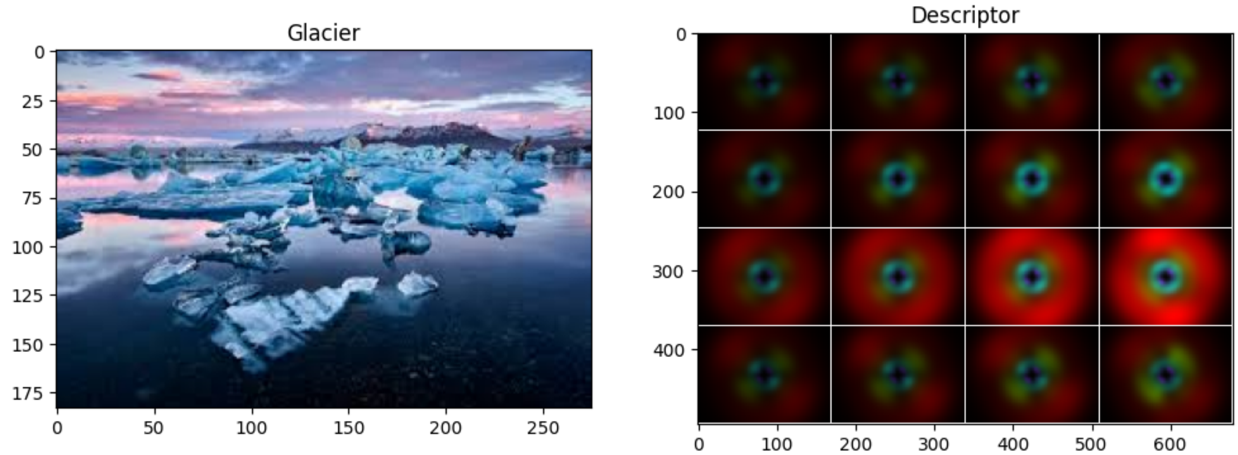
Intuitively, GIST summarizes the gradient information (scales and orientations) for different parts of an image, which provides a rough description (the gist) of the scene.

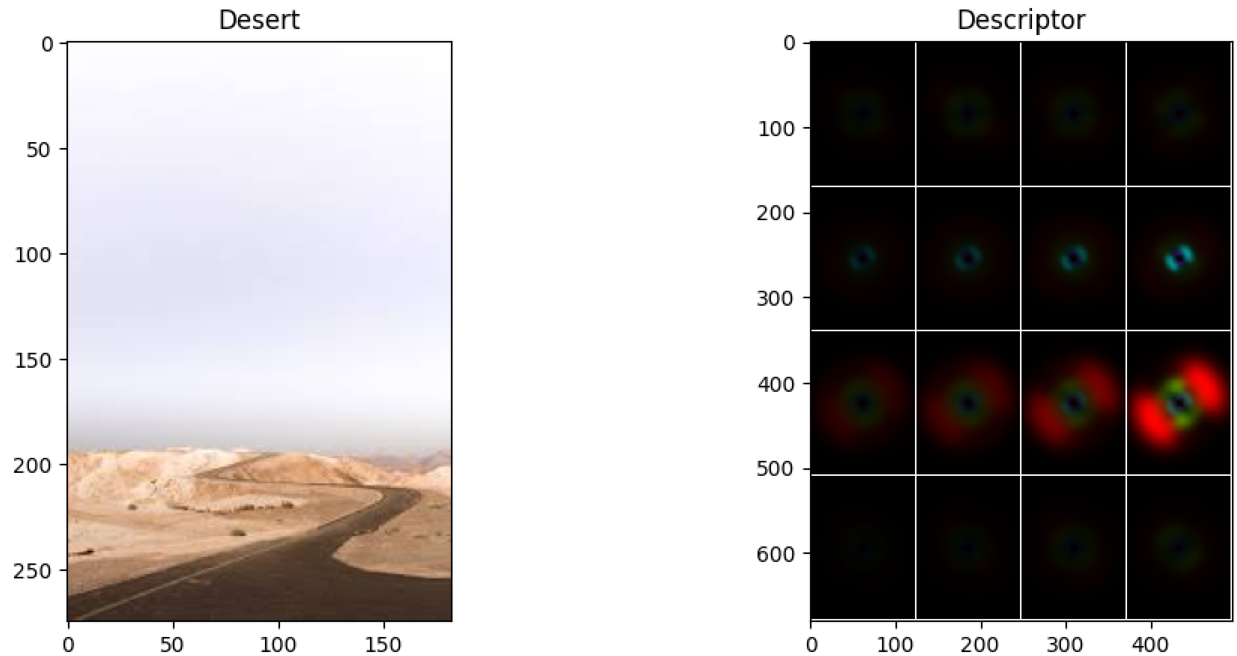
$$I(x, y, f_x, f_y) = \sum_{x', y'=0}^{N-1} i(x', y')\, h_r(x' - x, y' - y) e^{-j\, 2\pi (f_x x' + f_y y')} \tag{2}$$

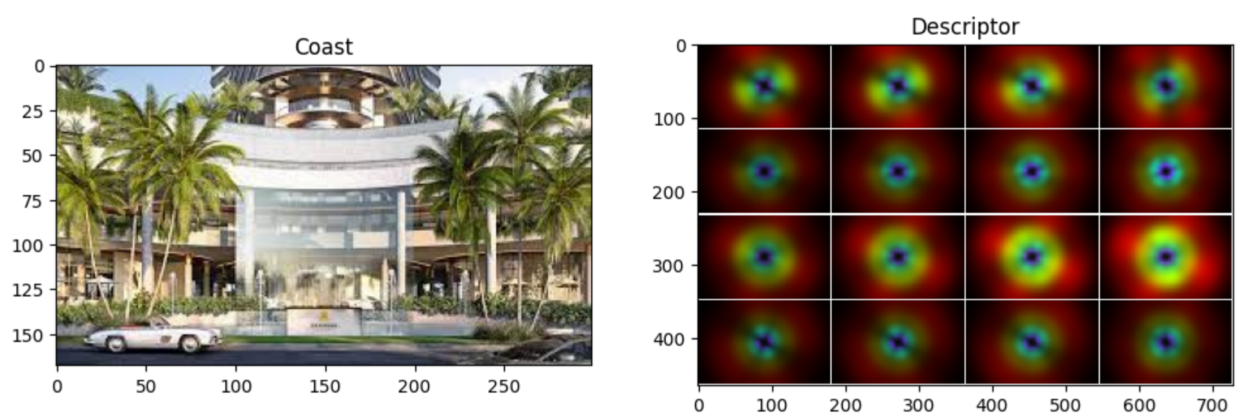
$$\begin{aligned} I(f_x, f_y) &= \sum_{x, y=0}^{N-1} i(x, y) h(x, y) e^{-j\, 2\pi (f_x x + f_y y)} \\ &= A(f_x, f_y)\, e^{j \Phi(f_x, f_y)} \end{aligned} \tag{1}$$

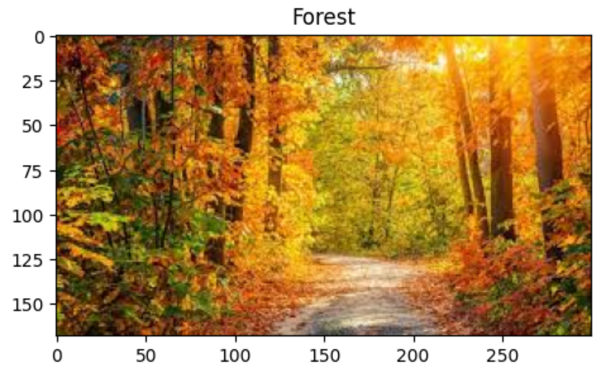
Convolving the images with gabor filters used to get the windowed fourier transform as described in the paper, which highlights the spatial distribution of spectral information.

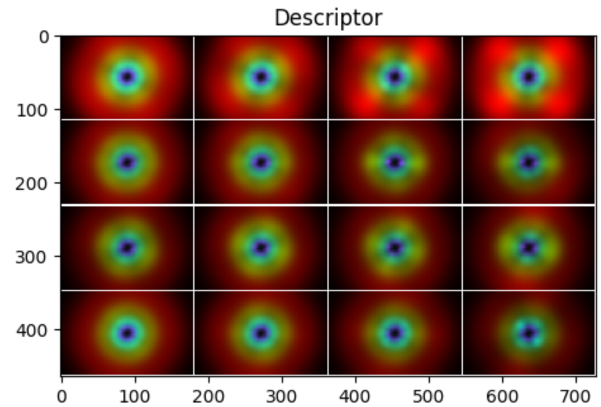
The principal components of the spectrogram of real-world scenes. The spectrogram is sampled at 4 × 4 spatial location for a better visualization. Each sub image corresponds to the local energy spectrum at the corresponding spatial location.

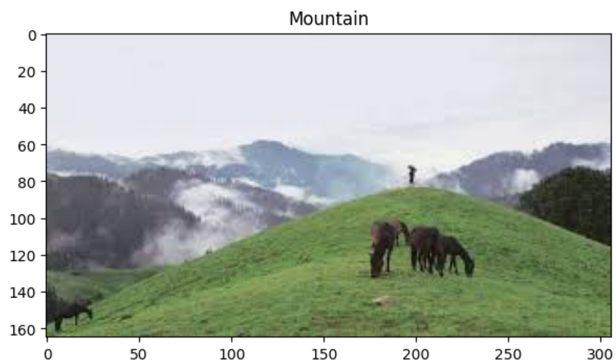

Glacier
Descriptor

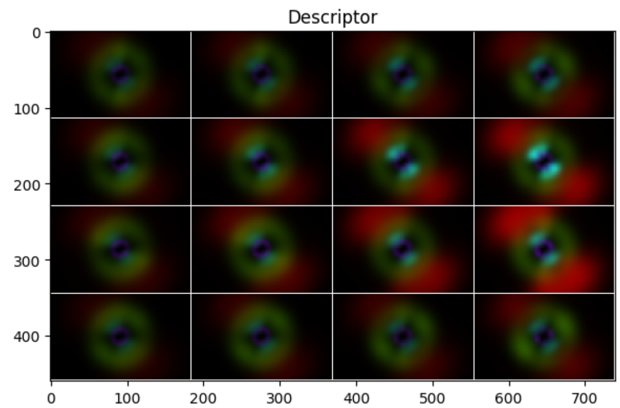

Desert
Descriptor

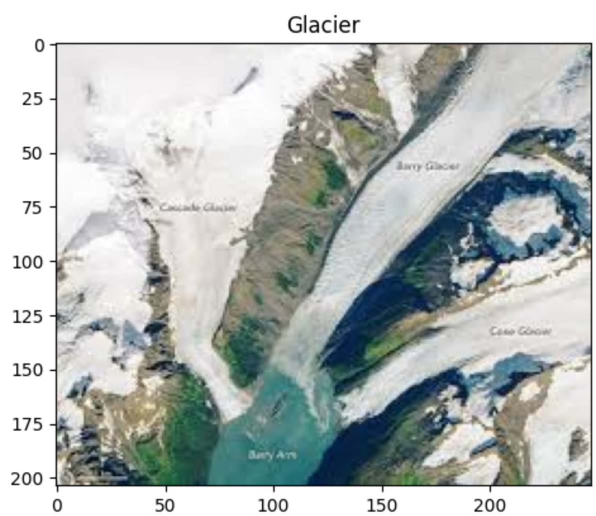

Coast
Descriptor

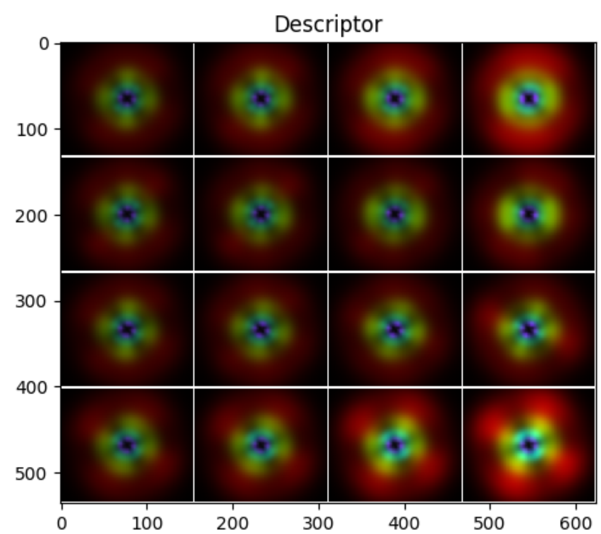

Forest
0
25
50
75
100
125
150
0
50
100
150
200
250


Descriptor
0
100
200
300
400
0
100
200
300
400
500
600
700

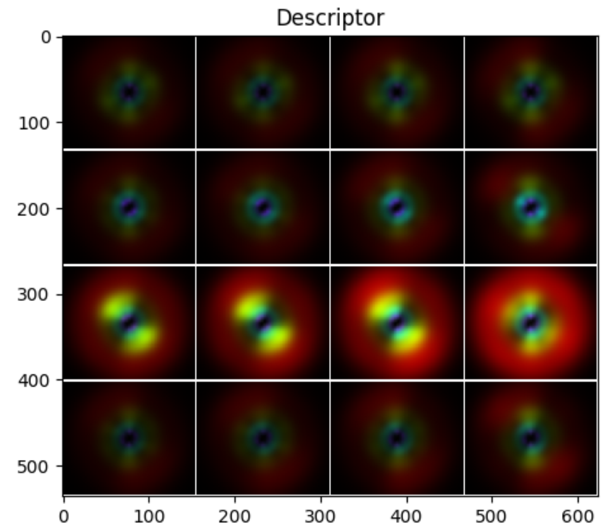

Mountain
0
20
40
60
80
100
120
140
160
0
50
100
150
200
250
300

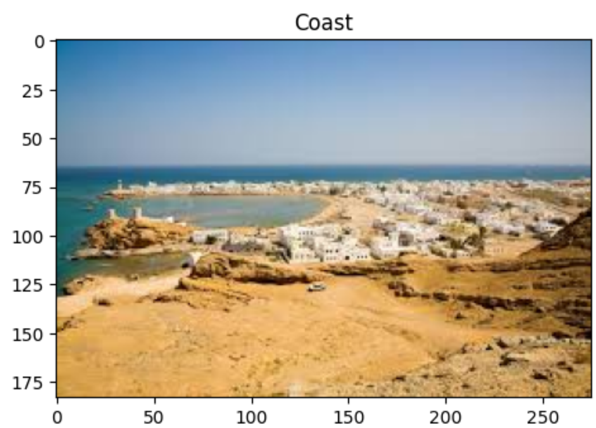

Descriptor
0
100
200
300
400
0
100
200
300
400
500
600
700

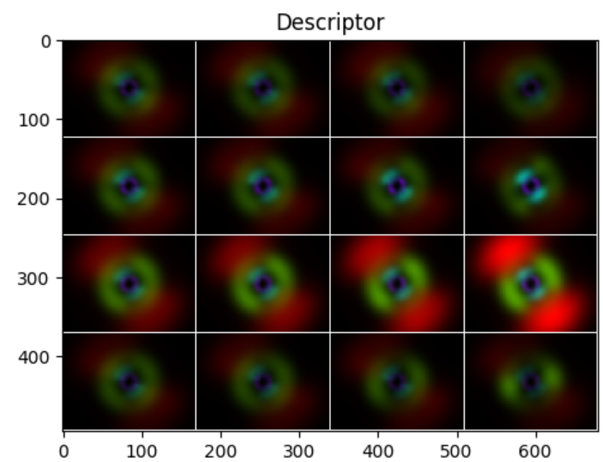

Glacier
0
25
50
75
100
125
150
175
200
0
50
100
150
200


Descriptor
0
100
200
300
400
500
0
100
200
300
400
500
600

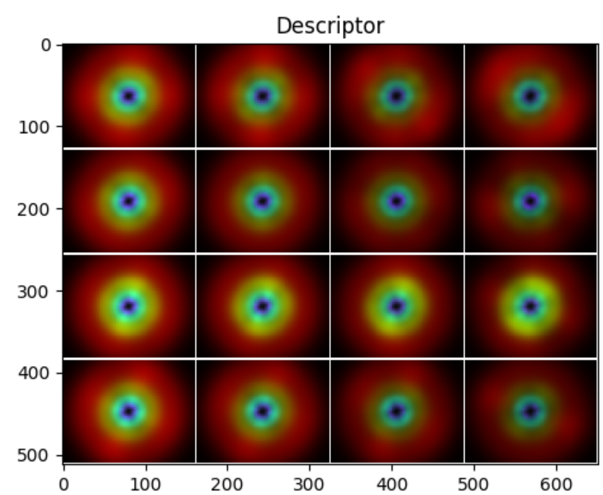

Desert

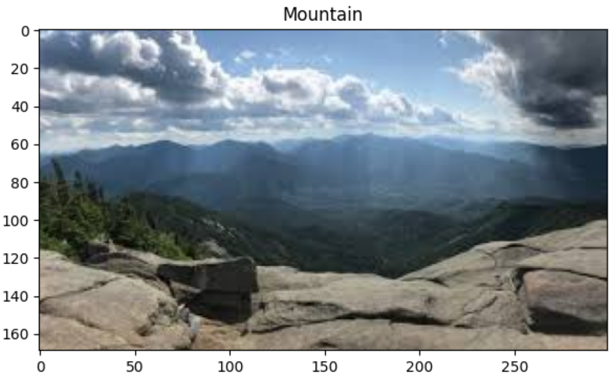

Descriptor

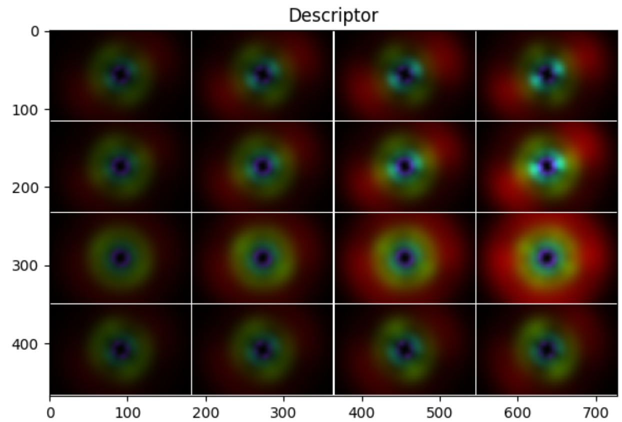

Coast

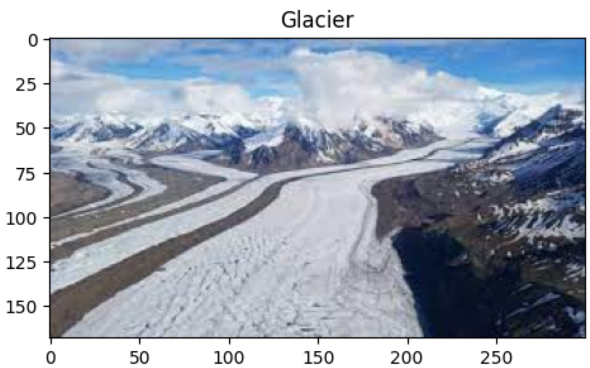

Descriptor

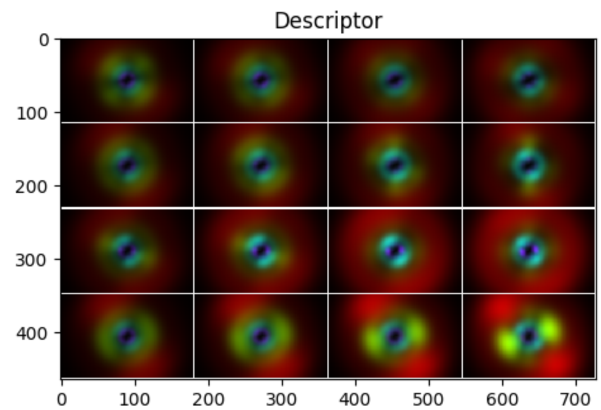

Forest


Descriptor

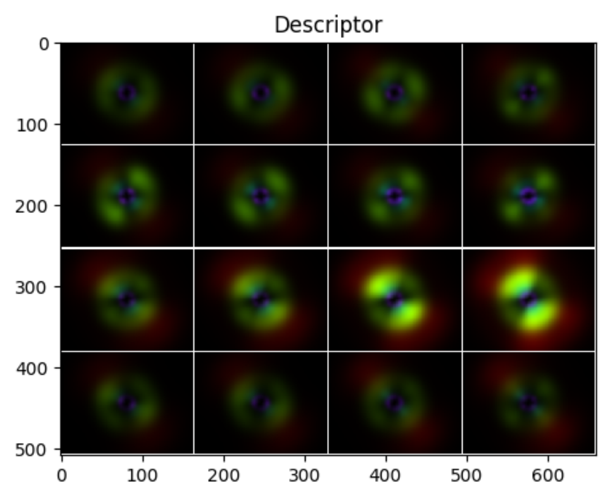

Mountain

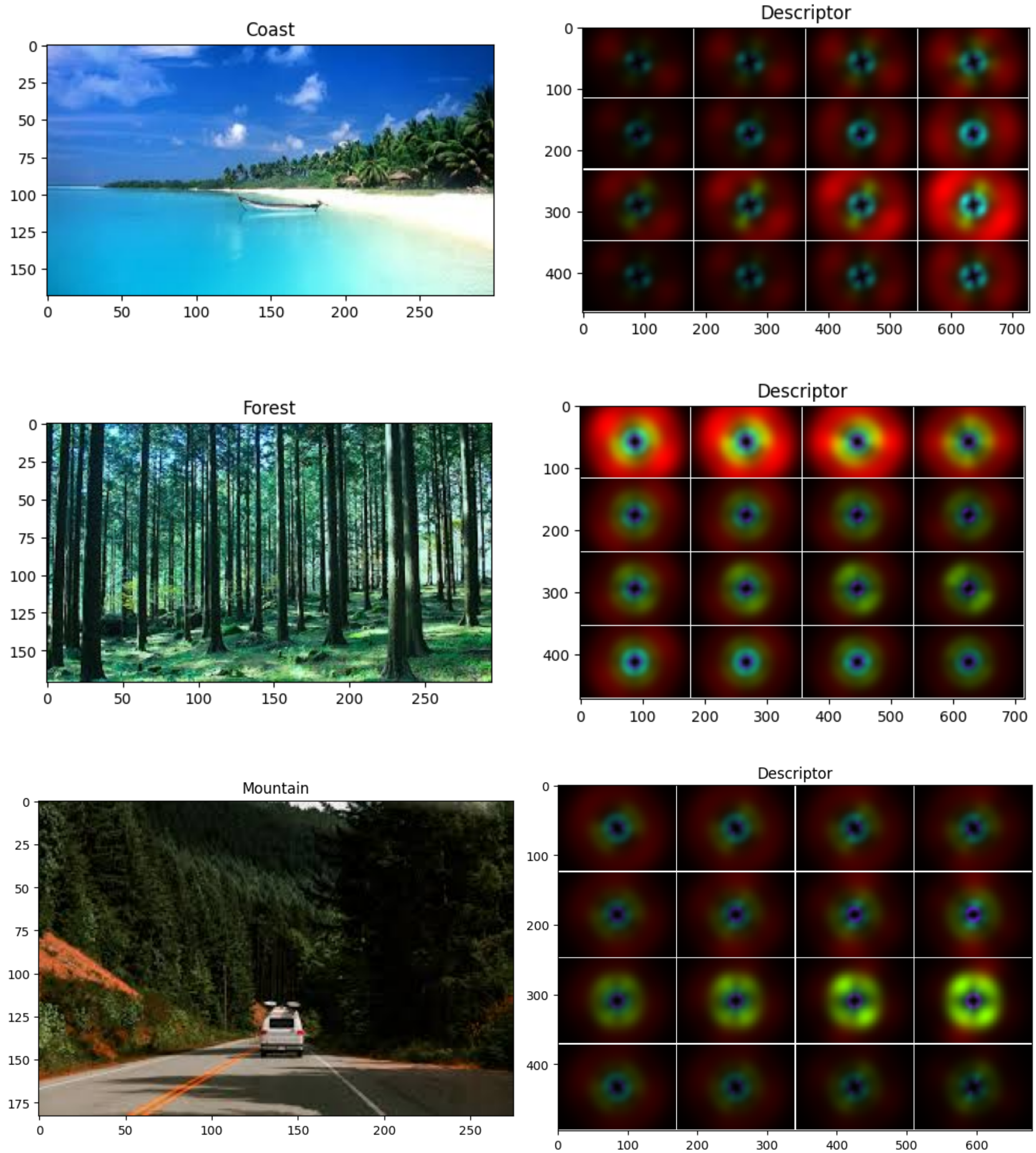

Descriptor


Glacier


Descriptor


Desert
shutterstock.com · 1200643537


Descriptor


Coast
Descriptor
Forest
Descriptor
Mountain
Descriptor

Data Preprocessing for Classification

Effective preprocessing is crucial for image analysis tasks. In this project, we focused on the following preprocessing steps:

Feature Extraction: For feature extraction, we considered two primary options: GIST and edge histogram features. GIST features were extracted to capture the global spatial layout and texture information, while edge histogram features were considered to capture more detailed information about edges in the images. The choice between the two depended on the problem requirements and the classifier's performance.

0	1	2	3	4	5	6	7	8	9	...	16	17	18	19	20	21	22	23	24	25
6.006838	-2.398132	8.187969	1.701154	1.889051	-5.231250	1.422649	7.629479	3.594717	1.447268	...	0.066872	0.086812	0.049791	0.094379	0.137537	0.137100	0.072016	0.091698	0.118471	0.145323
2.334625	8.994803	16.479666	-6.025523	-7.303590	-4.889502	6.372811	-6.210614	1.358121	2.113750	...	0.044670	0.078271	0.042981	0.102096	0.191972	0.192548	0.066786	0.082563	0.091923	0.106190
7.403770	-0.607260	4.298734	0.785822	-1.919728	-3.916264	-1.176495	-3.336758	-0.758777	6.010840	...	0.084287	0.092836	0.047516	0.079774	0.127706	0.132971	0.064892	0.096853	0.116584	0.156580
3.293411	-3.002451	-4.516487	0.210527	5.866006	-4.159185	3.003882	4.466413	1.109296	3.218502	...	0.052896	0.078371	0.043000	0.082206	0.106706	0.178798	0.063646	0.085961	0.193502	0.114913
-6.284536	14.778725	-6.899095	2.509909	1.384817	1.848903	5.196611	-4.720983	2.672344	4.727325	...	0.033325	0.068642	0.025004	0.108895	0.152353	0.222806	0.088529	0.104893	0.116878	0.078675

Quantization: To facilitate the development of a Bayesian classifier, we quantized the extracted features into discrete feature states. This step involved converting continuous feature values into a limited number of bins (here 7) or categories. The choice of quantization depends on the specific classifier configuration and the number of landscape categories to be recognized.

	0	1	2	3	4	5	6	7	8	9	...	17	18	19	20	21	22	23	24	25	Class
0	4	2	6	4	4	0	4	6	6	5	...	4	4	4	4	2	5	4	2	5	3
1	4	6	6	0	0	0	6	0	4	5	...	3	3	5	6	4	3	2	0	2	3
2	6	3	5	4	2	1	2	1	2	6	...	4	4	2	3	2	2	5	2	5	3
3	4	2	1	3	6	1	5	6	4	6	...	3	3	2	1	4	2	2	5	2	3
4	2	6	0	4	4	4	6	0	5	6	...	1	0	6	4	5	6	6	2	0	3

Quantization may involve dividing feature values into intervals or using clustering techniques to group similar values together. The goal is to simplify the feature space, making it more amenable to probabilistic modeling using a Bayesian network.

Bayesian Network Classifier

We will discuss the implementation of a Bayesian Network Classifier from scratch. A Bayesian Network Classifier is a probabilistic graphical model that represents the probabilistic relationships between features and class labels. The classifier uses Bayes' theorem to calculate the posterior probability of a class given the observed features.

Implementation Steps:

Model Configuration: We designed a Bayesian network with nodes representing the landscape categories (e.g., coast, desert, forest) and edges representing probabilistic dependencies between classes and features.

Classifier Logic: The classifier logic was implemented to calculate the posterior probabilities of each class given the observed features. This involved using Bayes' theorem to estimate the likelihood of observing the features for each class and the prior probabilities of each class.

From the training data, we estimated the parameters of the Bayesian Network using the following steps:

Class Prior Probabilities (P(C)): We estimated the prior probabilities of each landscape category. These prior probabilities represent the likelihood of each class occurring in the dataset.

Feature Likelihoods (P(X|C)): For each feature, such as GIST or edge histogram, we estimated the likelihood of observing a specific value given a class. This involved calculating the conditional probabilities of feature values for each class.

Parameter estimation is essential to make the Bayesian Network Classifier operational and ready for testing.

Evaluation and Interpretation

Accuracy: We calculated accuracy as a measure of the classifier's overall correctness in classifying landscape images.

Test-

```
Accuracy: 0.68
```

Val-

```
Accuracy: 0.63
```

Precision, Recall, and F1-Score: These metrics were calculated for each individual class to understand the classifier's performance for each landscape type.

Test-

```
Precision: 0.73
Recall: 0.68
F1 Score 0.67
```

Val-

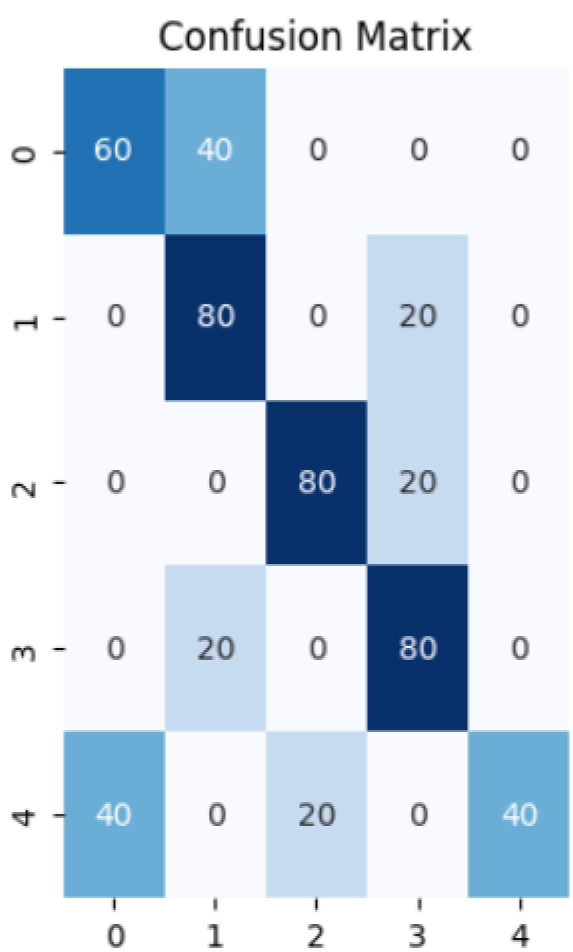
```
Precision: 0.64
Recall: 0.63
F1 Score 0.62
```

Interpretation and Justification: We noticed not so good performance as compared to other work on the Kaggle, we can account for computing the less GIST Descriptors as compared to paper because of limited computing power as well as other image processing approaches like smoothening, cropping, augmentation could be used.

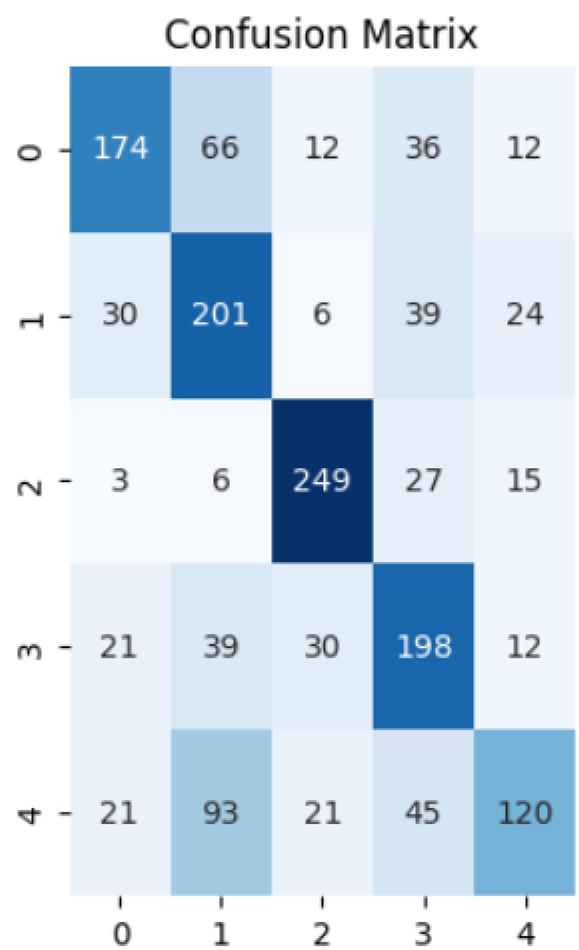
Accuracy measures the overall correctness of the classifier, while precision, recall, and F1-score provide insights into the classifier's performance for individual classes.

Confusion Matrix Visualization: We generated confusion matrices using the training and test sets and visualized them as heatmaps for better interpretation.

Test-



Val-



Class-Specific Metrics: Accuracy, precision, recall, and F1-score were calculated for each class. This allowed us to identify how well the model performed for individual landscape categories.

Test-

```
For Coast
Accuracy: 0.6, Precision: 0.6, Recall: 0.6, F1-Score: 0.6

For Desert
Accuracy: 0.8, Precision: 0.57, Recall: 0.8, F1-Score: 0.67

For Forest
Accuracy: 0.8, Precision: 0.8, Recall: 0.8, F1-Score: 0.8

For Glacier
Accuracy: 0.8, Precision: 0.67, Recall: 0.8, F1-Score: 0.73

For Mountain
Accuracy: 0.4, Precision: 1.0, Recall: 0.4, F1-Score: 0.57
```

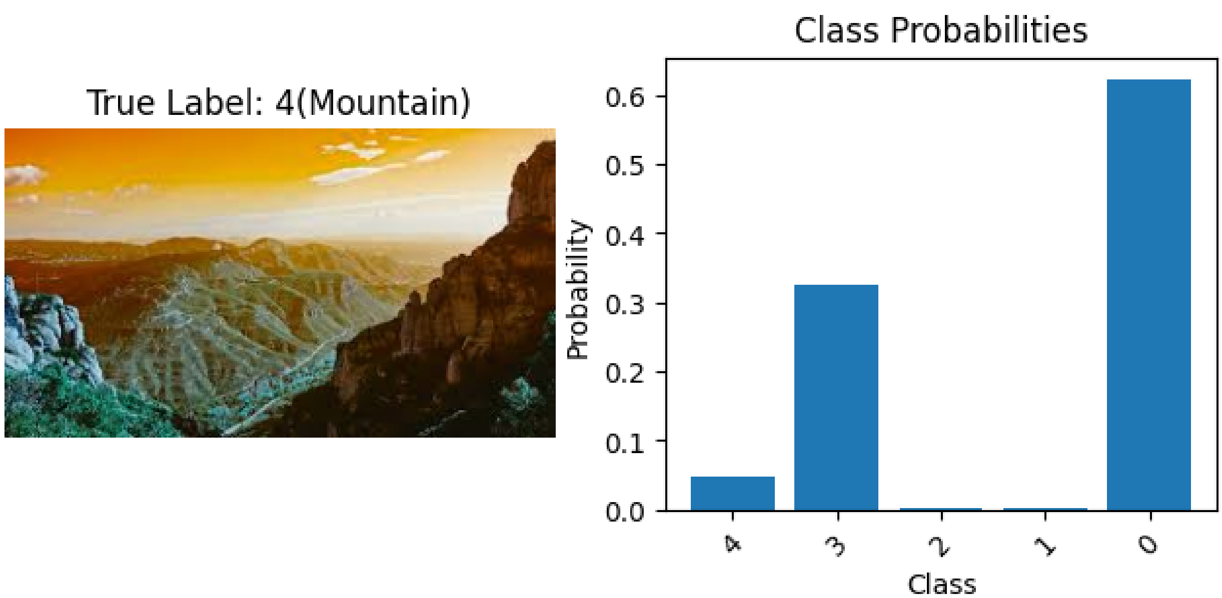
Val-

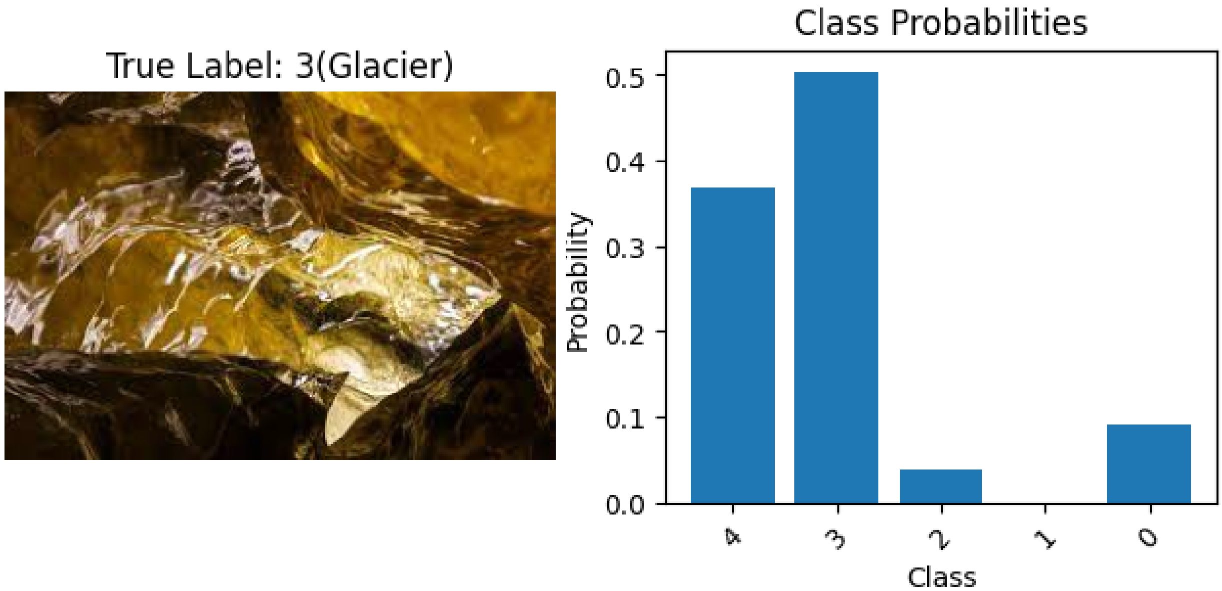
```
For Coast
Accuracy: 0.58, Precision: 0.7, Recall: 0.58, F1-Score: 0.63

For Desert
Accuracy: 0.67, Precision: 0.5, Recall: 0.67, F1-Score: 0.57

For Forest
Accuracy: 0.83, Precision: 0.78, Recall: 0.83, F1-Score: 0.81

For Glacier
Accuracy: 0.66, Precision: 0.57, Recall: 0.66, F1-Score: 0.61

For Mountain
Accuracy: 0.4, Precision: 0.66, Recall: 0.4, F1-Score: 0.5
```

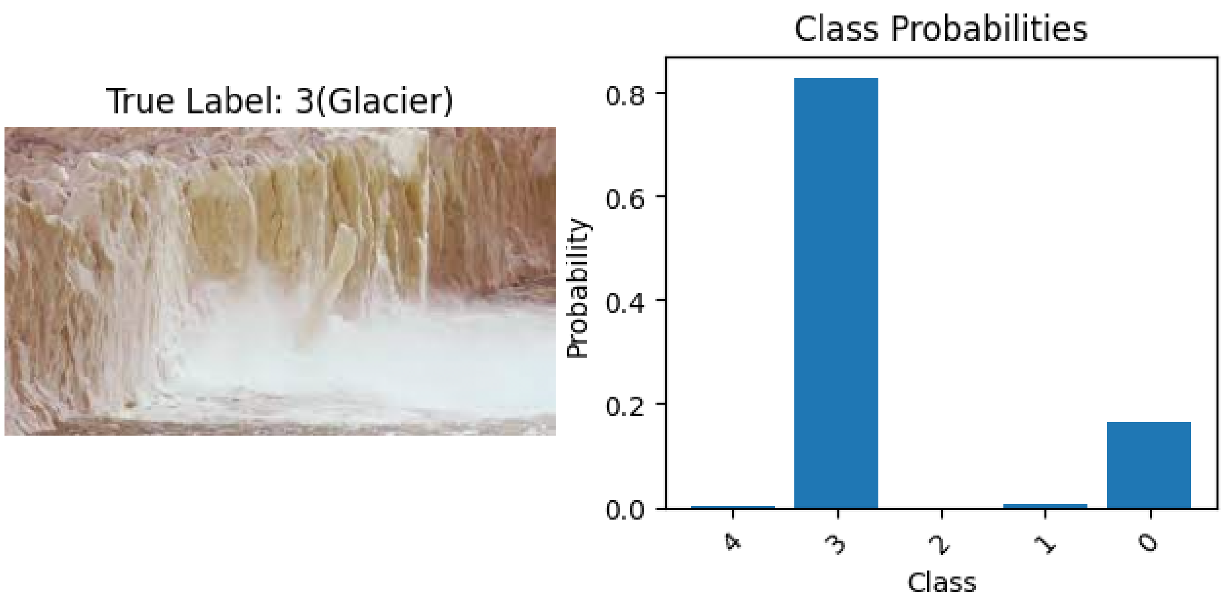
Visualize Prediction

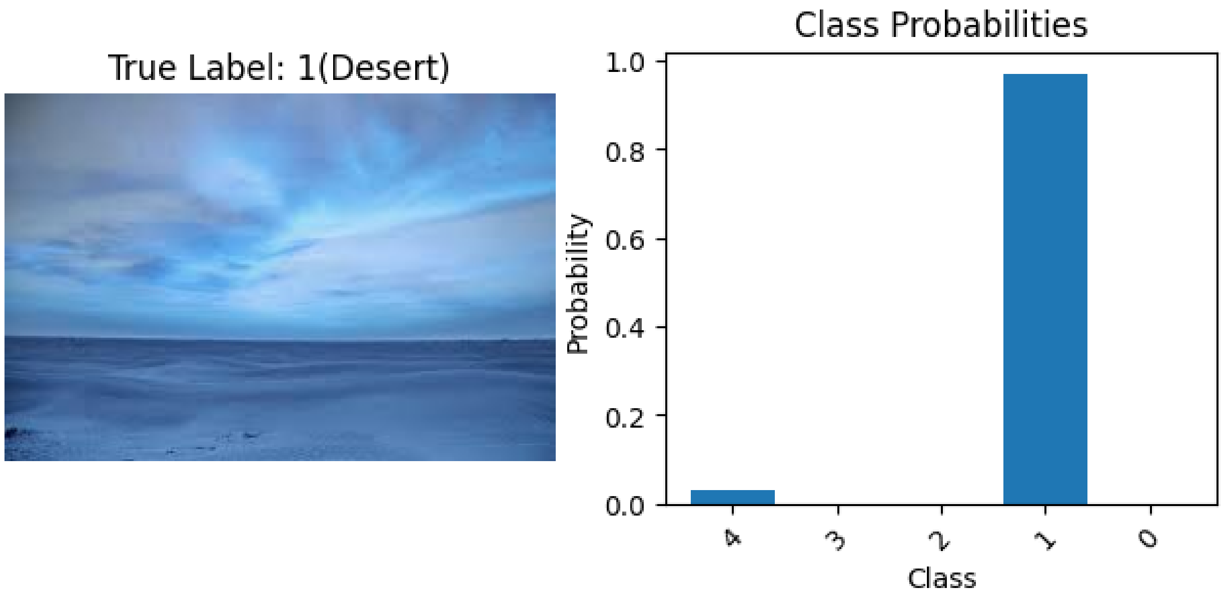
We randomly selected a subset of test images and visualized the model's predictions along with the ground truth labels.

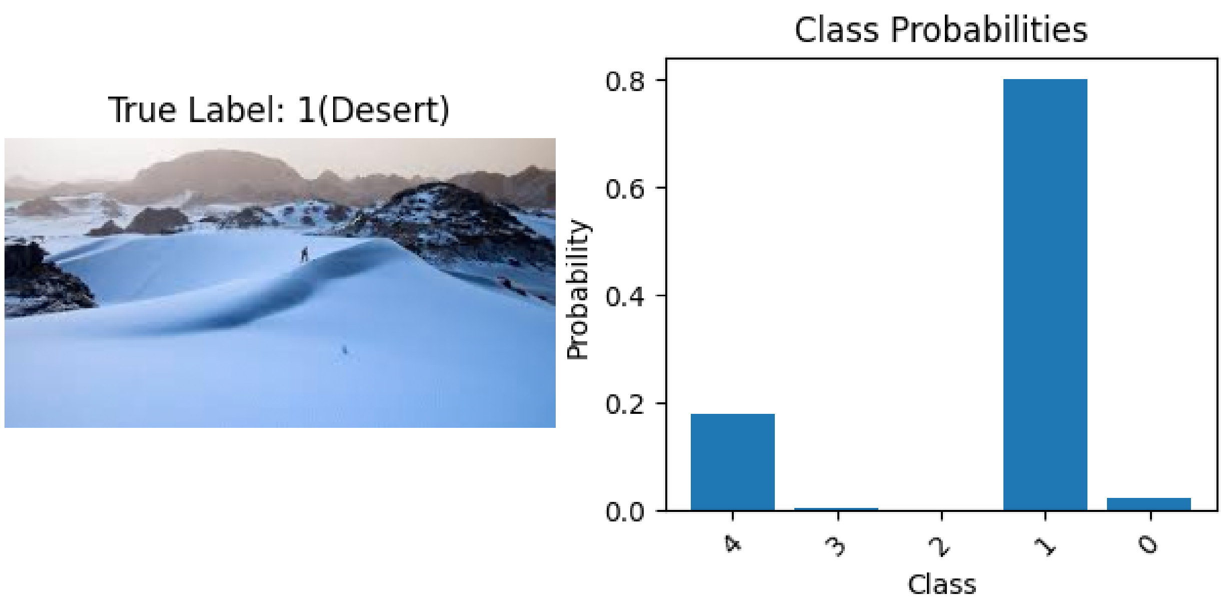
For each image, we showed the posterior probability values for the predicted class, as well as the class and posterior probability for which the second-highest posterior probability was achieved.

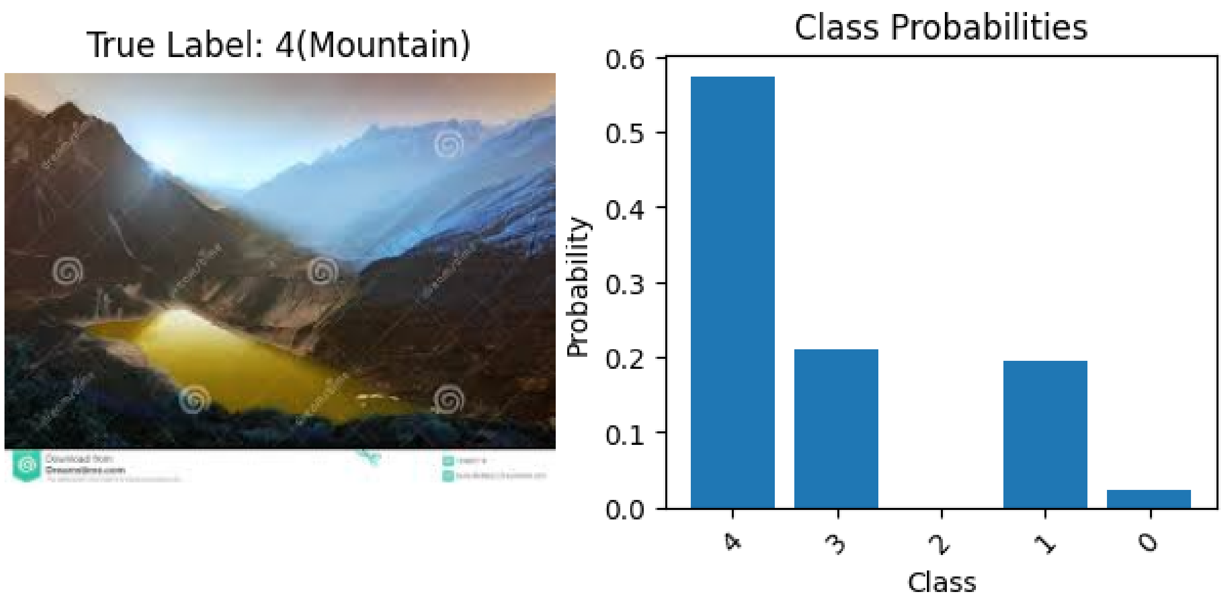


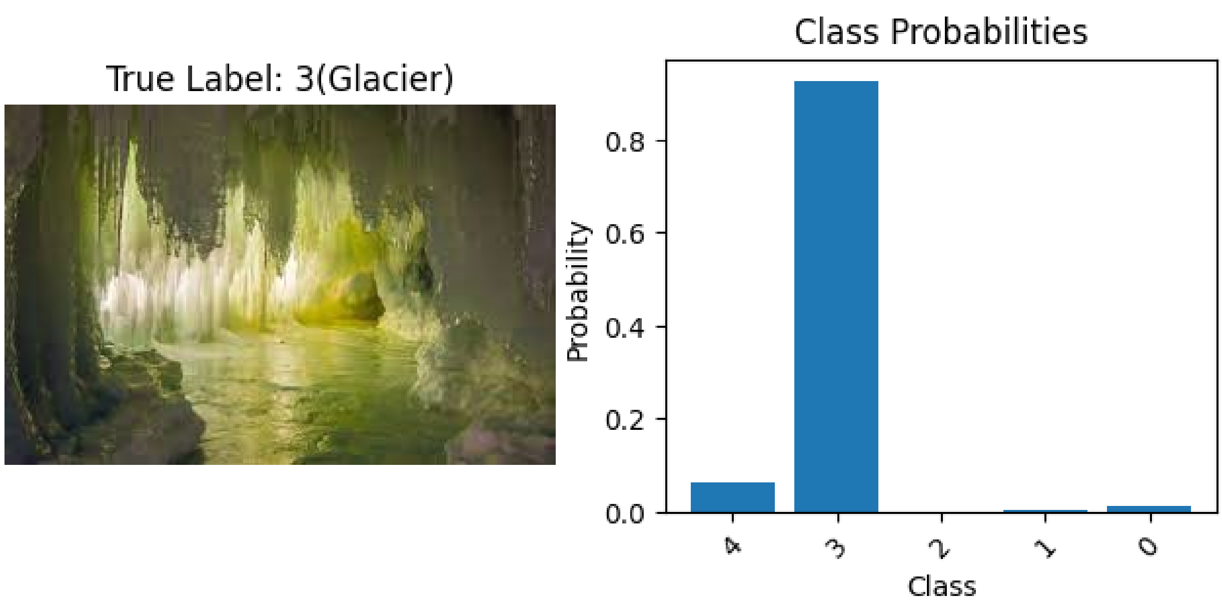

True Label: 3(Glacier)
Class Probabilities
Probability
0.0
0.1
0.2
0.3
0.4
0.5
4
3
2
1
0
Class


True Label: 3(Glacier)
Class Probabilities
Probability
0.0
0.2
0.4
0.6
0.8
4
3
2
1
0
Class


True Label: 1(Desert)
Class Probabilities
Probability
Class
0.0
0.2
0.4
0.6
0.8
1.0
4
3
2
1
0


True Label: 1(Desert)
Class Probabilities
Probability
Class
0.0
0.2
0.4
0.6
0.8
4
3
2
1
0


True Label: 4(Mountain)
Class Probabilities
Probability
0.0
0.1
0.2
0.3
0.4
0.5
0.6
4
3
2
1
0
Class


True Label: 3(Glacier)
Class Probabilities
Probability
0.0
0.2
0.4
0.6
0.8
4
3
2
1
0
Class

References -

1. The code and implementation is based on the MATLAB code provided by the authors Link
2. The concept is based on the paper- Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope

Note - The work is based on the implementation given by authors, that's why it could be similar to other students also.