IMAGE CLASSIFICATIO N

Group Members

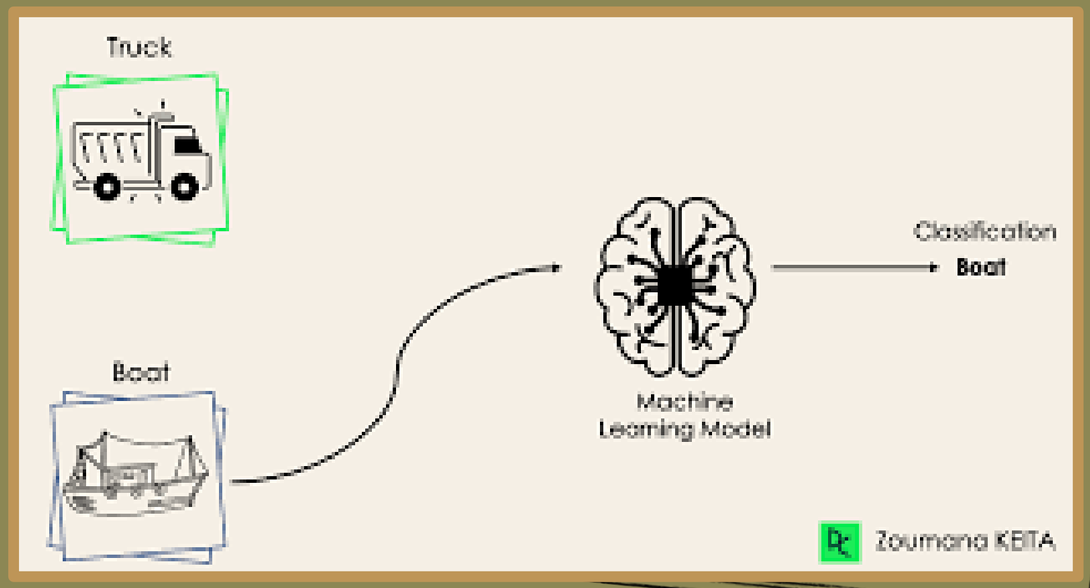
Bimsara Siman Meru Pathiranage

Ahmed Abdulrahim

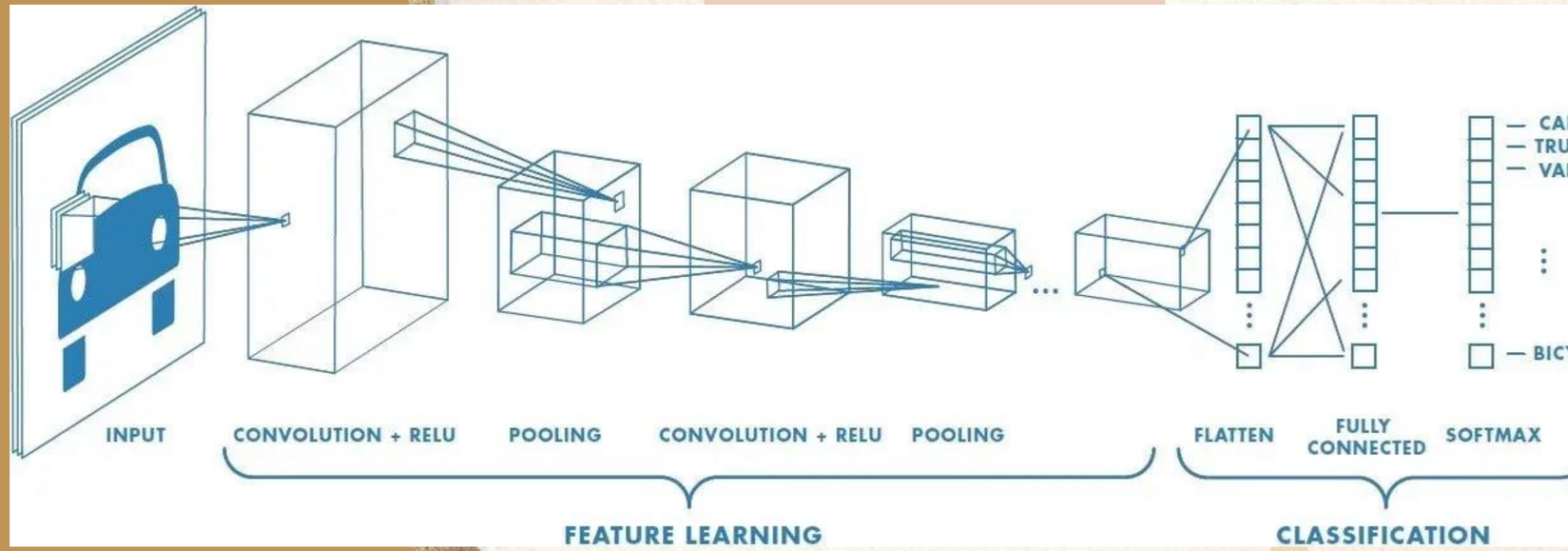
Simranjeet Kaur

Efemena Theophilus Edoja

Himanshu



Problem Definition



- Image classification is crucial across various fields.
- Categorizing natural scene images accurately is challenging due to landscape diversity.
- Our project explores advanced techniques like CNNs, logistic regression, and decision trees.

Data Source

Dataset from Kaggle Intel Image Classification. Rich collection capturing diverse landscapes globally. The dataset comprises image data capturing natural scenes from various locations worldwide. It contains approximately 25,000 images, each sized 150x150 pixels, categorized into six distinct classes:

- Buildings (category 0)
- Forest (category 1)
- Glacier (category 2)
- Mountain (category 3)
- Sea (category 4)
- Street (category 5)

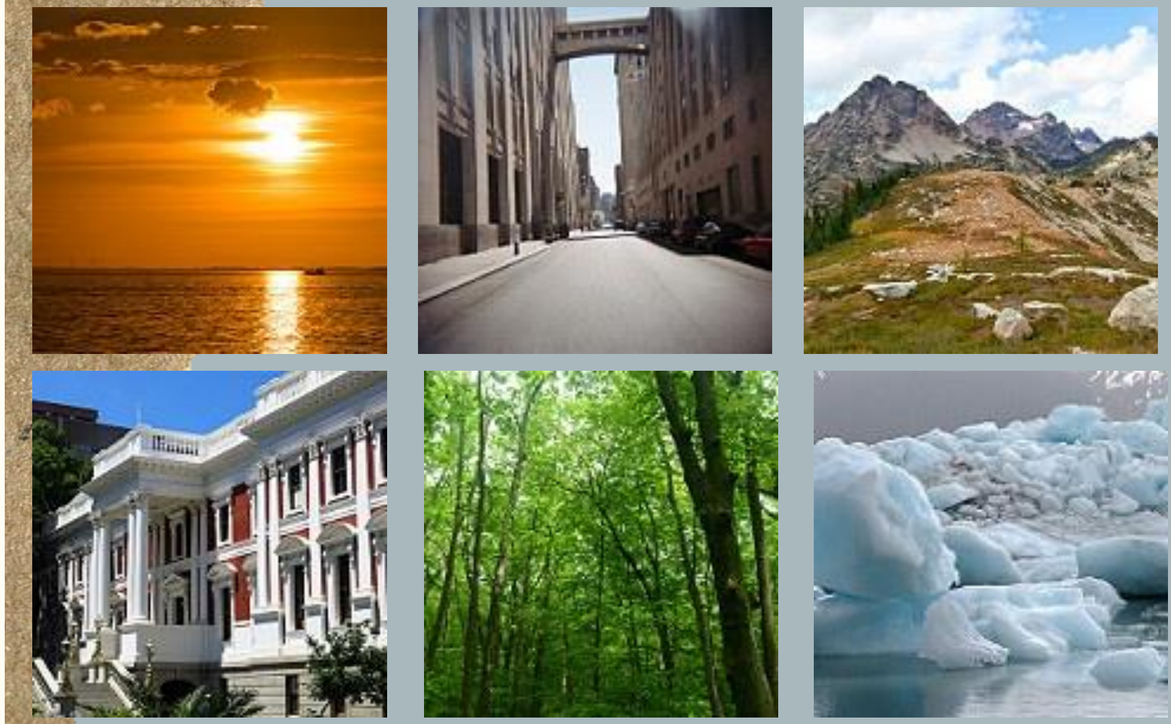
The dataset is divided into three main subsets:

14,000 images → Train set

Test set → 3,000 images

Prediction set→ 7,000 images

Each subset is stored in separate zip files.



Dataset URL : Intel Image Classification

Data Preprocessing

Dataset Augmentation

Used to artificially increase the diversity of the training dataset by applying various transformations to the existing images.

- Rescaling
- Zooming
- Horizontal and vertical shifts introduced
- Maintain image integrity

Feature Extraction

Features are extracted from images to prepare data for logistic regression and decision tree models.

Various image processing techniques applied:

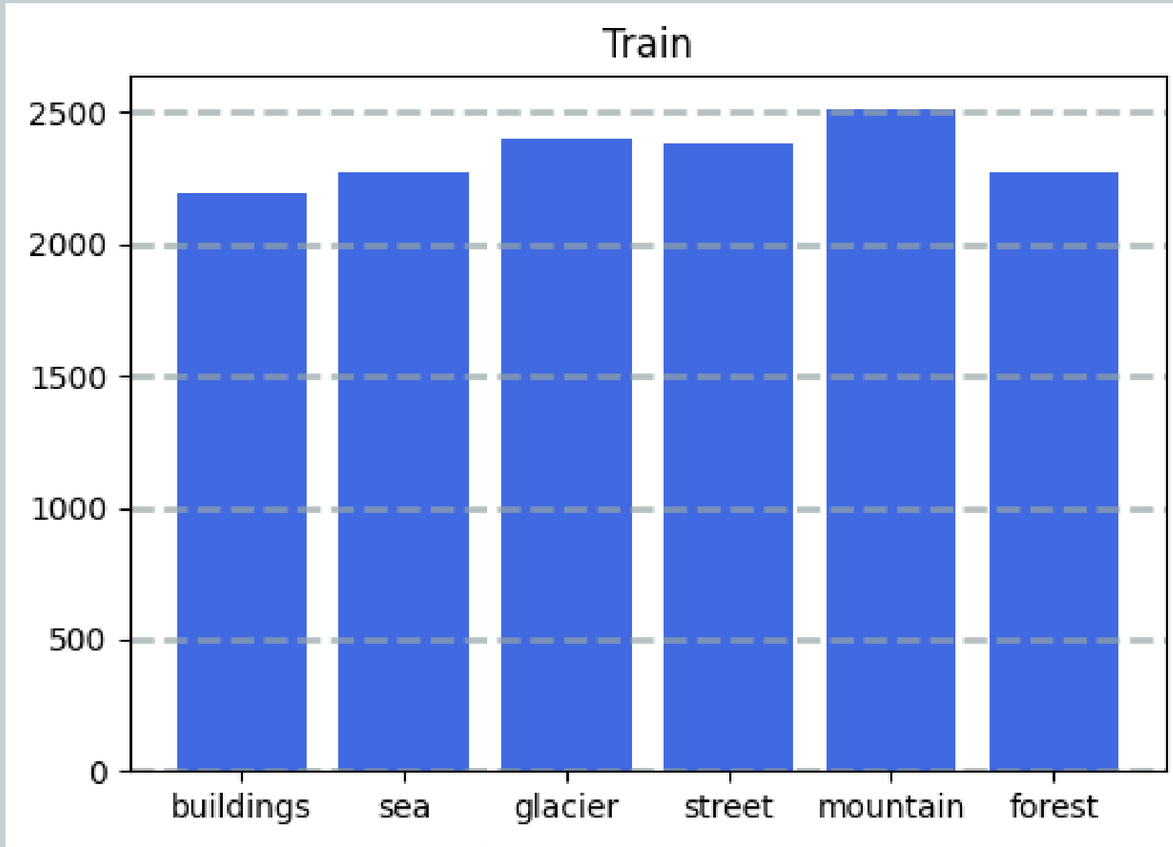
- Local Binary Pattern (LBP) for texture
- Sobel Edge Detection for edges
- Histogram of Oriented Gradients (HOG) for shape and texture
- Color histogram in RGB space for color information

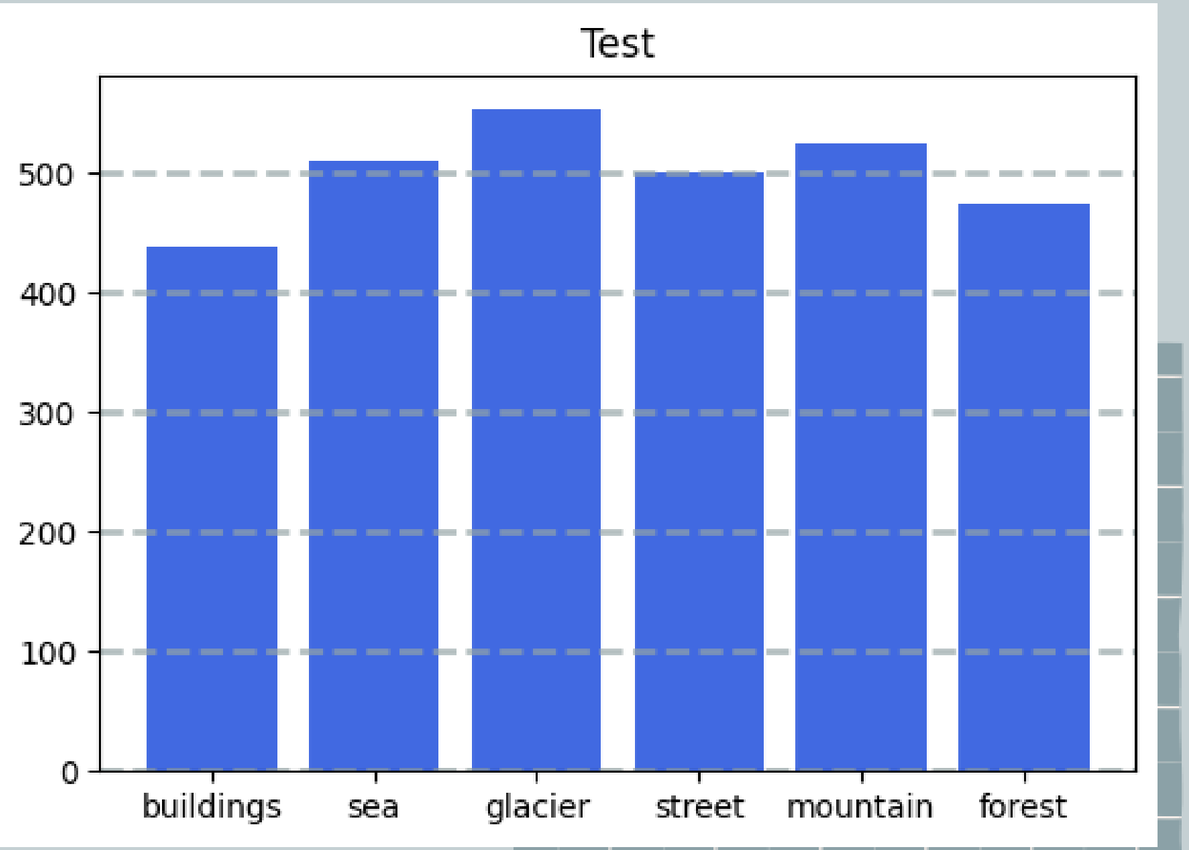
Extracted features stored for future use.

Data preparation steps:

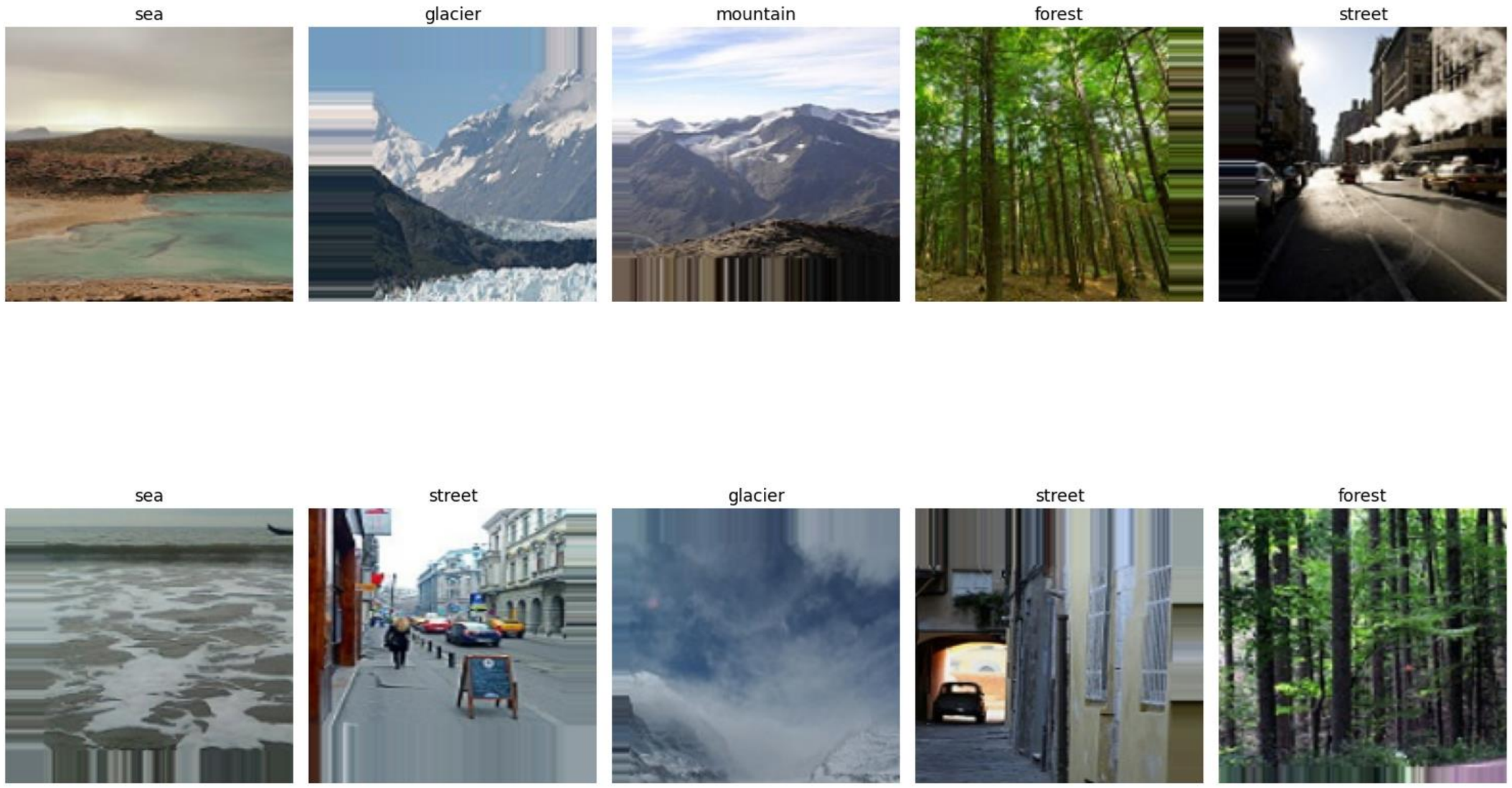
- Features and labels converted to NumPy arrays.
- One-hot encoded labels converted to integer labels

Exploratory Data Analysis





Sample Training Images



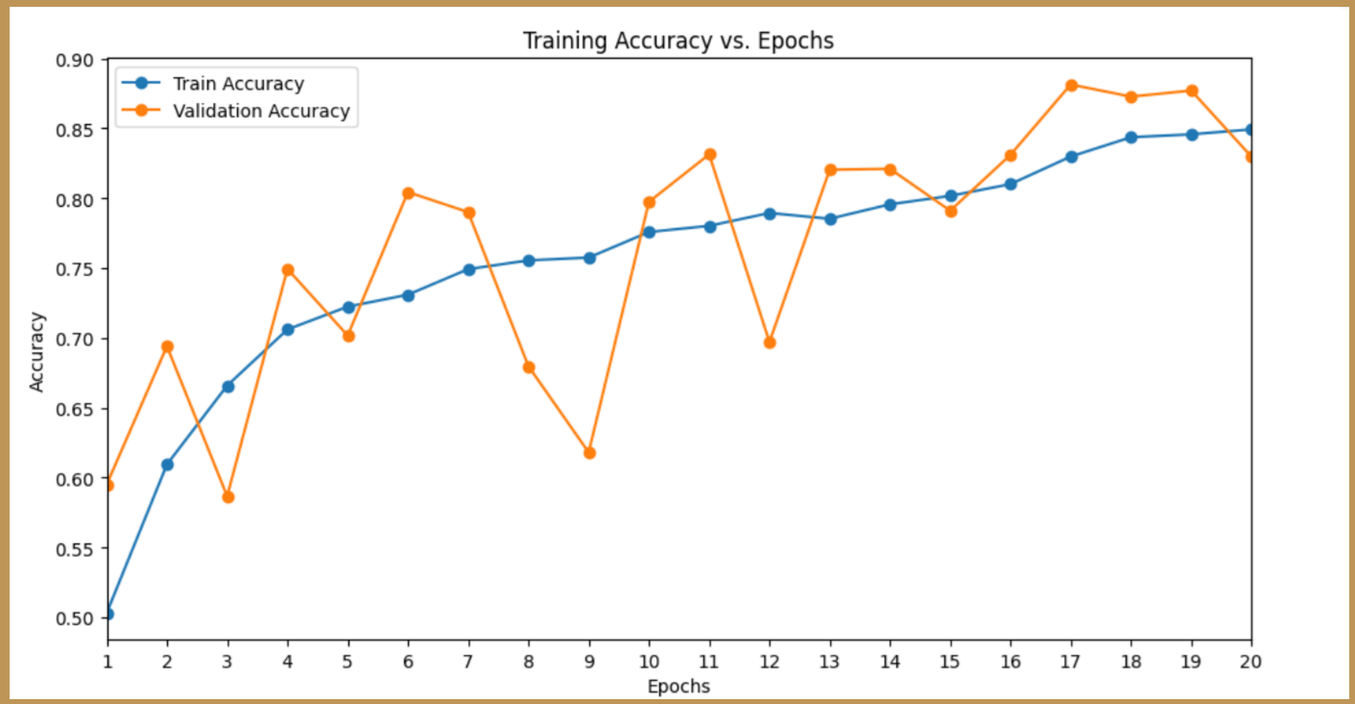
CNN Model

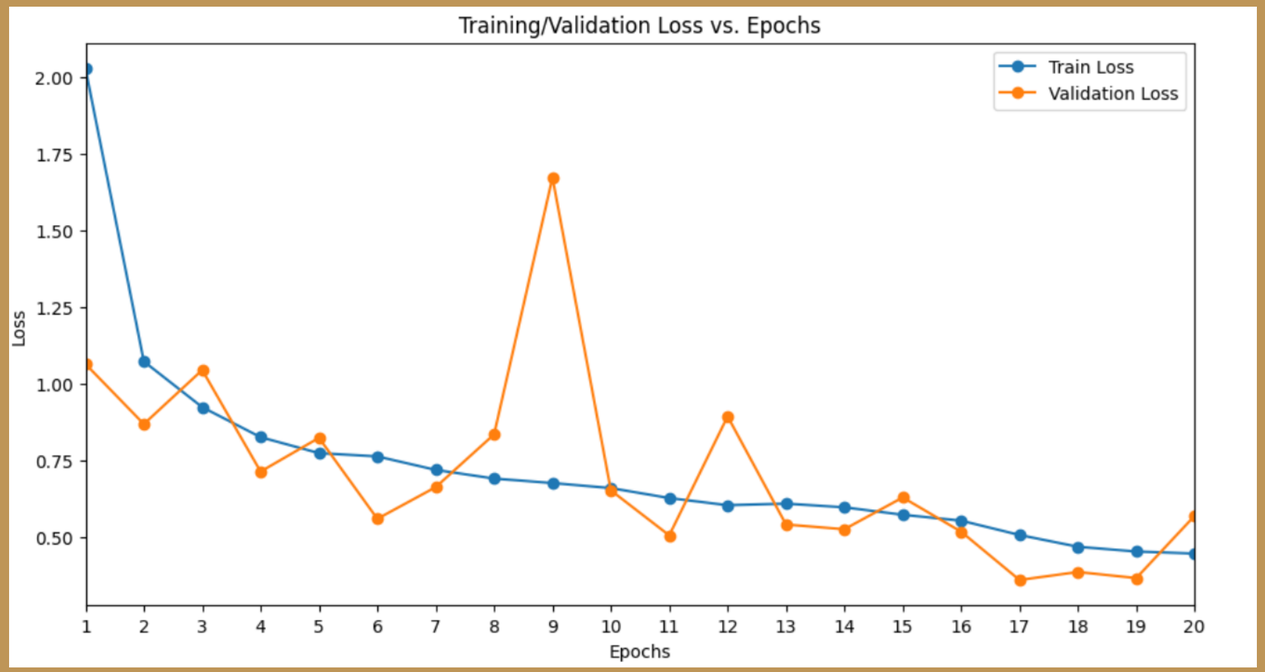
- CNN model with convolutional layers, activation functions, pooling layers, batch normalization, dropout, and dense layers.
- Learning Rate Reduction: Implemented ReduceLROnPlateau callback to reduce learning rate when validation loss plateaus.
- Optimizer: Used Adam optimizer with a learning rate of 0.001.
- Compiled the model with Categorical Cross Entropy loss function and accuracy metric.
- Trained the model for 20 epochs on the training dataset.

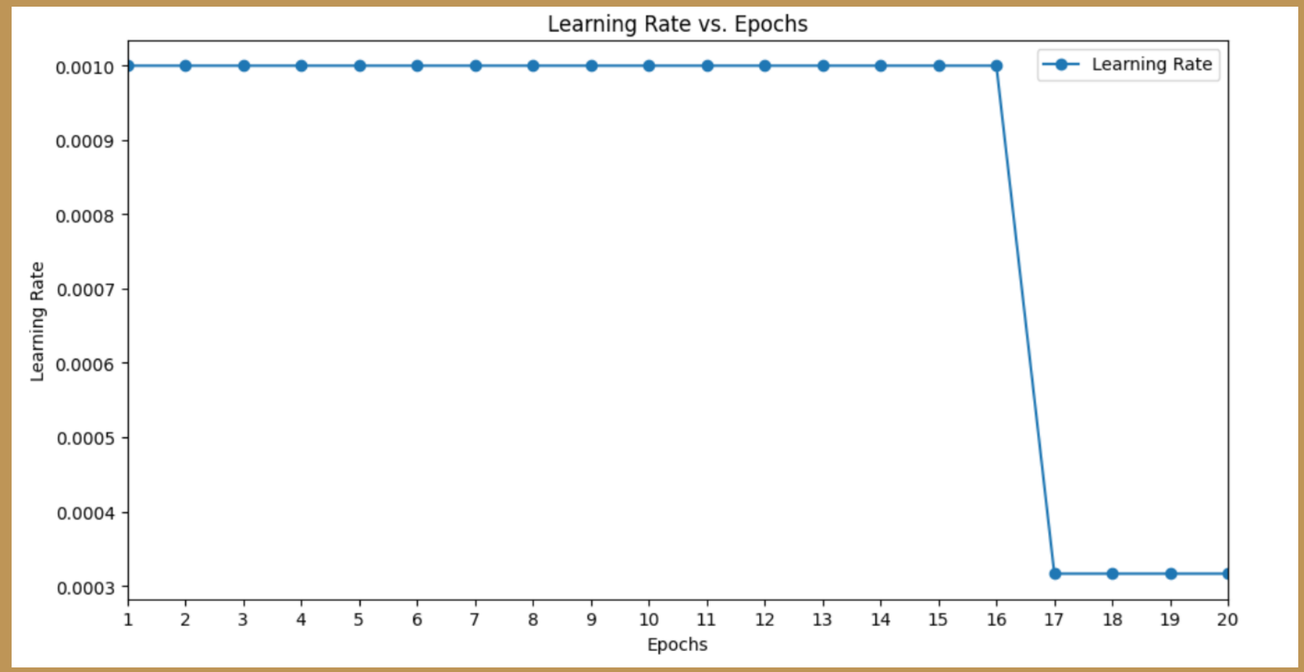
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 146, 146, 128)	9,728
activation (Activation)	(None, 146, 146, 128)	0
max_pooling2d (MaxPooling2D)	(None, 73, 73, 128)	0
batch_normalization (BatchNormalization)	(None, 73, 73, 128)	512
conv2d_1 (Conv2D)	(None, 71, 71, 64)	73,792
activation_1 (Activation)	(None, 71, 71, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 35, 35, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 35, 35, 64)	256
conv2d_2 (Conv2D)	(None, 33, 33, 32)	18,464
activation_2 (Activation)	(None, 33, 33, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 32)	0
batch_normalization_2 (BatchNormalization)	(None, 16, 16, 32)	128
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 256)	2,097,408
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 6)	1,542

Total params: 2,201,830 (8.40 MB)
Trainable params: 2,201,382 (8.40 MB)
Non-trainable params: 448 (1.75 KB)

CNN Model Training







Logistic Regression Model

- Dataset split into training and testing sets with an 80-20 ratio.
- Logistic Regression model initialized with max iteration set to 1000 and random state as 42.

Random Forest Classifier

- Initialized Random Forest Classifier with specific parameters

  n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, max_features='auto', random_state=42, n_jobs=-1

- **Second Iteration of Random Forest Model**

  Created a second Random Forest model with a reduced max_depth parameter to mitigate overfitting.

Decision Tree Classifier

- **Initialized Decision Tree Classifier with specific parameters including**

  criterion='gini', splitter='best', max_depth=20, min_samples_split=20, min_samples_leaf=10, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.01, random_state=42

Model Tuning with PCA

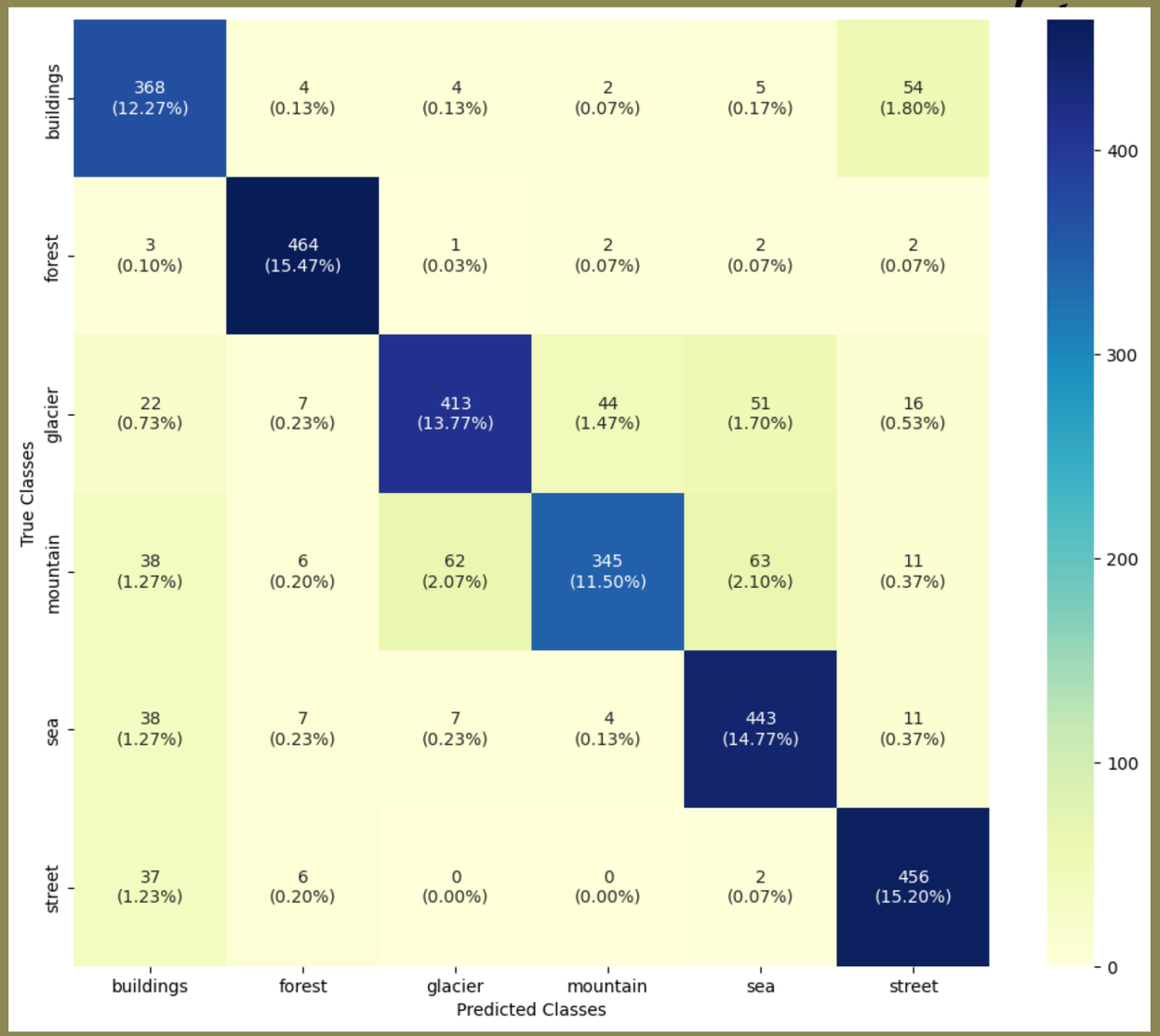
- Applied Principal Component Analysis to reduce dimensionality while preserving most of the variance.
- Helps in improving performance and reducing overfitting.
- n_components=0.95 specifies retaining principal components explaining at least 95% of the variance.

Results

Convolutional Neural Network (CNN):

- Test Loss: 0.57
- Test Accuracy: 0.83

	precision	recall	f1-score	support
buildings	0.73	0.84	0.78	437
forest	0.94	0.98	0.96	474
glacier	0.85	0.75	0.79	553
mountain	0.87	0.66	0.75	525
sea	0.78	0.87	0.82	510
street	0.83	0.91	0.87	501
accuracy			0.83	3000
macro avg	0.83	0.83	0.83	3000
weighted avg	0.83	0.83	0.83	3000



Logistic Regression Model

Accuracy for training: 0.7840919212612452
Accuracy for testing: 0.6583541147132169

Precision: 0.6577111522267143
Recall: 0.6583541147132169
F1 Score: 0.657681999378986

Classification Report:

	precision	recall	f1-score	support
0	0.67	0.63	0.65	431
1	0.89	0.92	0.91	454
2	0.59	0.59	0.59	480
3	0.53	0.57	0.55	488
4	0.58	0.54	0.56	470
5	0.70	0.71	0.71	484
accuracy			0.66	2807
macro avg	0.66	0.66	0.66	2807
weighted avg	0.66	0.66	0.66	2807

Decision Tree Model

Accuracy for training: 0.44909592945577625
Accuracy for testing: 0.4346277164232277

Precision: 0.4088810115915435
Recall: 0.4346277164232277
F1 Score: 0.40972820708332475

Classification Report:

	precision	recall	f1-score	support
0	0.23	0.21	0.22	431
1	0.66	0.76	0.71	454
2	0.24	0.07	0.11	480
3	0.36	0.51	0.42	488
4	0.49	0.44	0.46	470
5	0.47	0.60	0.53	484
accuracy			0.43	2807
macro avg	0.41	0.43	0.41	2807
weighted avg	0.41	0.43	0.41	2807

Random Forest Classifier Model

Accuracy for training: 1.0
Accuracy for testing: 0.1923762023512647

Precision: 0.8348217421523807
Recall: 0.1923762023512647
F1 Score: 0.2670414343211963 7

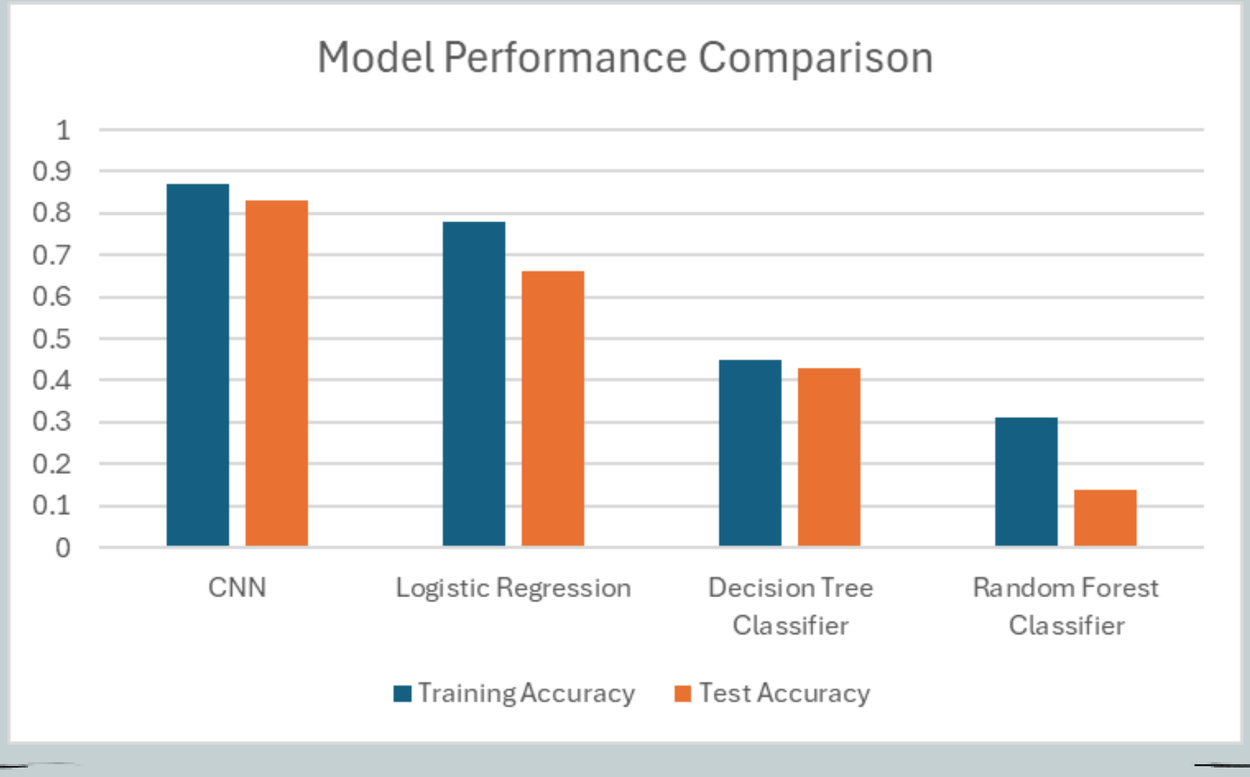
Classification Report:

	precision	recall	f1-score	support
0	0.86	0.01	0.03	431
1	0.94	0.67	0.78	454
2	0.82	0.12	0.20	480
3	0.72	0.05	0.09	488
4	0.93	0.08	0.15	470
5	0.76	0.23	0.36	484
micro avg	0.87	0.19	0.32	2807
macro avg	0.84	0.19	0.27	2807
weighted avg	0.83	0.19	0.27	2807
samples avg	0.19	0.19	0.19	2807

Accuracy for training: 0.3137970962857397
Accuracy for testing: 0.14463840399002495
Precision: 0.7211732505215358
Recall: 0.14463840399002495
F1 Score: 0.1973403710339356
Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	431
1	0.93	0.65	0.76	454
2	0.85	0.06	0.11	480
3	0.76	0.03	0.05	488
4	0.91	0.04	0.09	470
5	0.81	0.10	0.18	484
micro avg	0.90	0.14	0.25	2807
macro avg	0.71	0.15	0.20	2807
weighted avg	0.72	0.14	0.20	2807
samples avg	0.14	0.14	0.14	2807

Model Performance Comparison

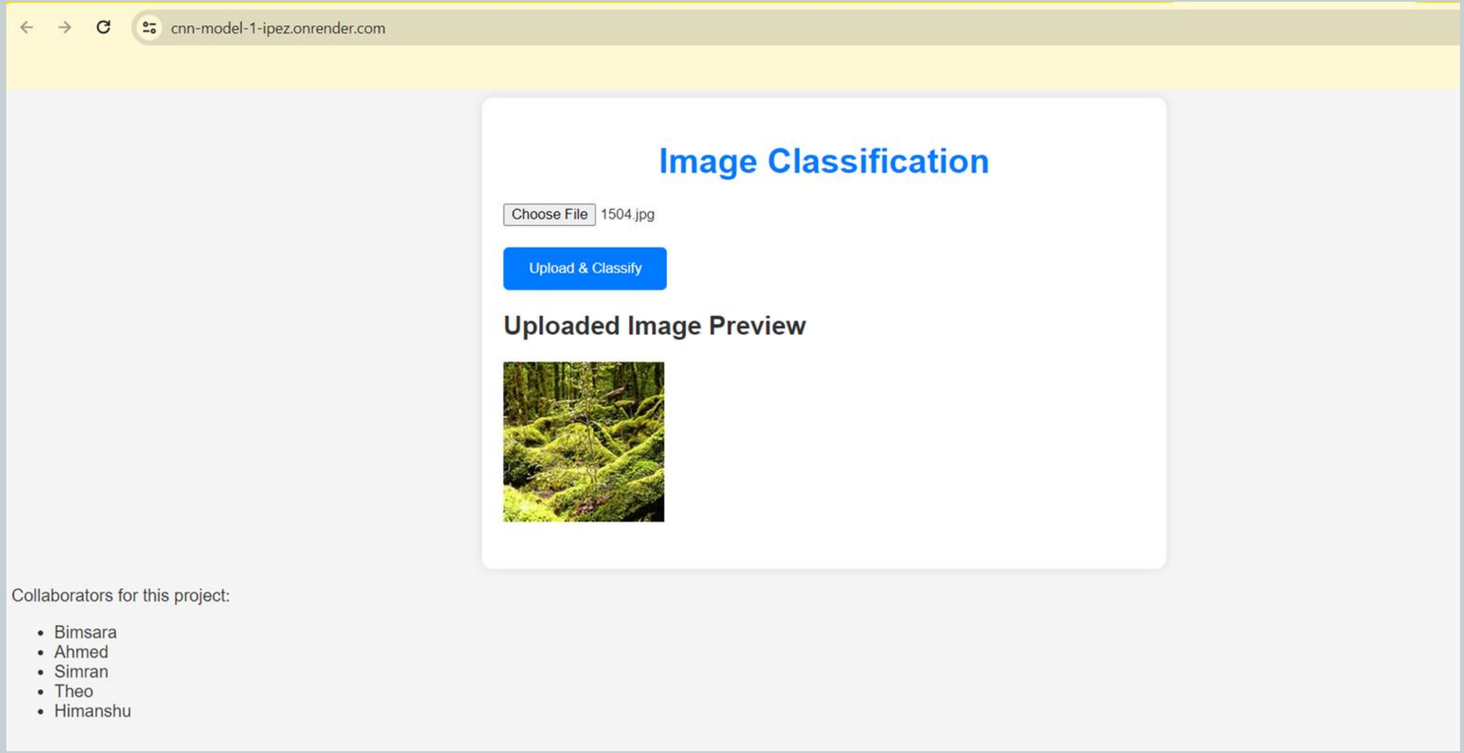


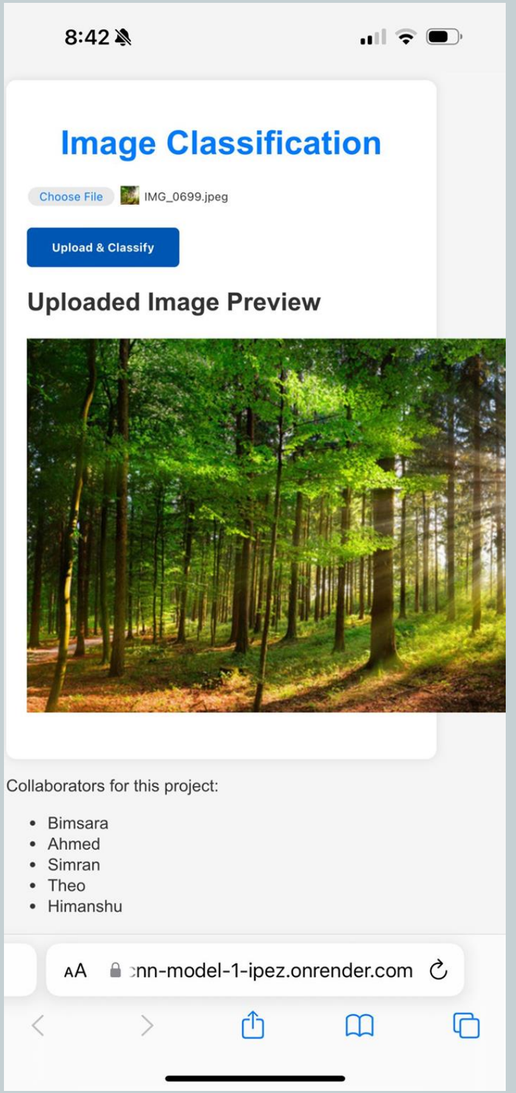
Misclassified Images

UI Development

Web Service Interface

https://cnn-model-1-ipez.onrender.com/





Thank you !