

Building a CNN Model for Image Classification

Objective

The goal of this assignment is to **design, implement, and evaluate a Convolutional Neural Network (CNN)** model using a real-world dataset. Students will gain hands-on experience in data preprocessing, CNN architecture design, training, evaluation, and model interpretation.

Dataset

Use the [Intel Image Classification Dataset](#) (available on Kaggle).

- Link: Intel Image Classification
 - It contains **25,000+ natural scene images** across **6 classes**:
 - Buildings
 - Forest
 - Glacier
 - Mountain
 - Sea
 - Street
-

Tasks

Part A: Data Understanding and Preprocessing

1. Load the dataset and explore its structure (number of classes, sample images, image sizes).
2. Perform **data preprocessing** steps:

- Resize all images to a common size (e.g., 150×150).
- Normalize pixel values.
- Split dataset into **train, validation, and test sets**.
- Apply **data augmentation** (rotation, flip, zoom, shift, etc.) to improve generalization.

Part B: CNN Model Development

3. Build a **Convolutional Neural Network (CNN)** from scratch using TensorFlow/Keras or PyTorch.
 - Minimum 3 convolutional layers.
 - Use activation functions (ReLU, softmax).
 - Apply pooling layers (max pooling/average pooling).
 - Add fully connected layers and dropout for regularization.
4. Train the CNN model on the training dataset and validate it on the validation set.
 - Plot training vs. validation **loss and accuracy curves**.
 - Use early stopping and model checkpointing if necessary.

Part C: Model Evaluation and Interpretation

5. Evaluate the model on the test set and report performance metrics:
 - Accuracy
 - Precision, Recall, F1-score per class
 - Confusion Matrix
6. Visualize predictions for at least **10 random test images** with predicted vs. actual labels.