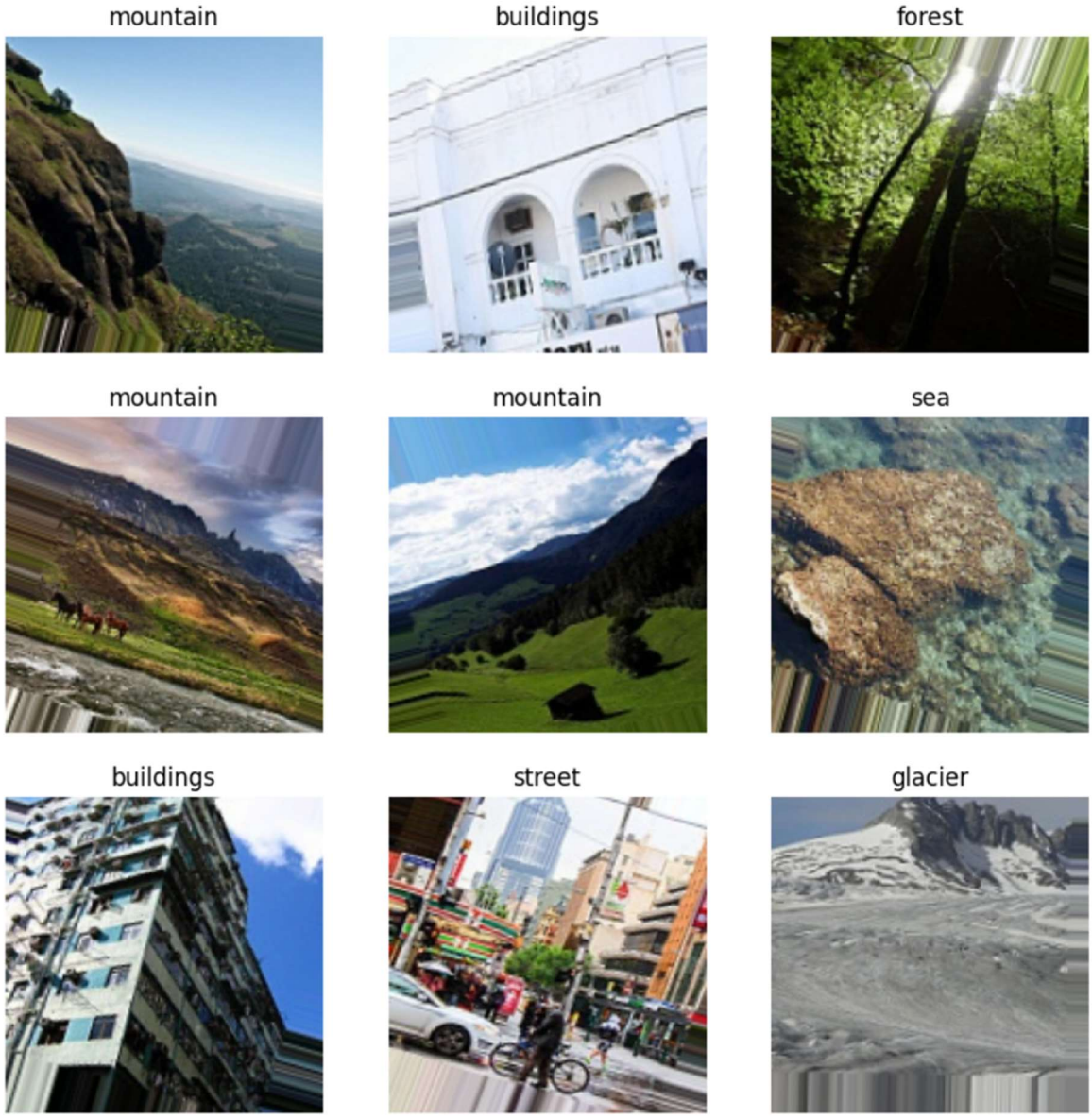
Image Classification Using Convolutional Neural Networks

1. Introduction

Images from the Intel Image Dataset are classified into six categories—buildings, forest, glacier, mountain, sea, and street—using Convolutional Neural Networks (CNNs). The goal of this project was to experiment with different CNN architectures, evaluate their performance, and understand how architecture changes affect model accuracy and predictions.

2. Dataset Overview:

- Dataset source: Kaggle – Intel Image Classification



- Number of classes: 6
- Structure: Training and validation folders with subfolders for each class.
- Preprocessing steps:
    - Resizing images to 150×150 pixels
    - Normalizing pixel values to [0,1]
    - Converting labels to one-hot encoding.

3. CNN Architectures

3.1 Architecture 1 – Basic CNN

- Model details:
    - 3 Conv2D layers (32 → 64 → 128) with MaxPooling
    - Flatten → Dense(128) → Dropout(0.5) → Output layer
- Reason for choosing: Simple architecture to act as a baseline, fast to train
- Training details: 10 epochs, optimizer: Adam, loss: categorical_crossentropy

Results:

Validation Accuracy: 83%

Observations:

Captures basic features well

Some individual predictions were incorrect

Quick and stable training

3.2 Architecture 2 – Deeper CNN with Batch Normalization

- Model details:
    - 4 Conv2D layers (32 → 64 → 128 → 256)
    - BatchNormalization after each convolutional layer
    - GlobalAveragePooling instead of Flatten
    - Dense(256) → Dropout(0.5) → Output layer
- Reason for change: To improve feature extraction, generalization, and handle more complex patterns

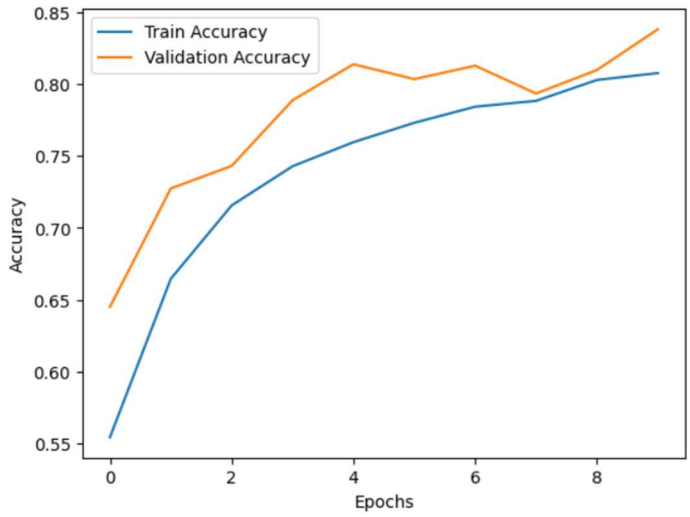
- Training details: 20 epochs, optimizer: Adam, loss: categorical_crossentropy

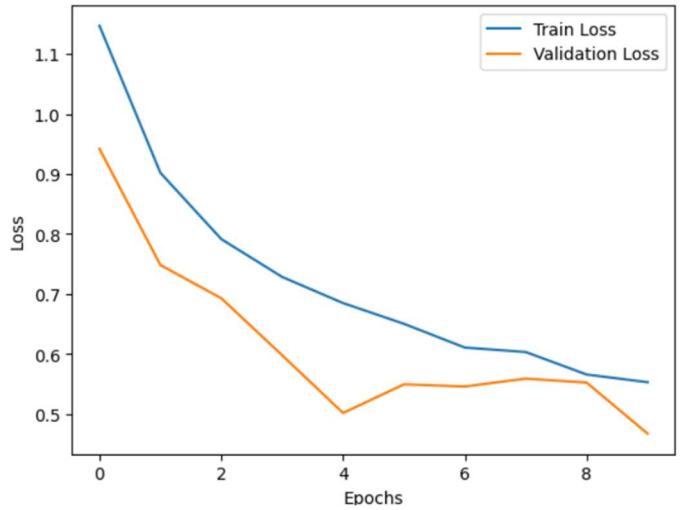
Results:

- Validation Accuracy: 75%
- Observations:
    - Sample predictions were all correct for selected images
    - Overall accuracy was lower, possibly due to overfitting or insufficient tuning
    - Demonstrates that deeper networks don’t always guarantee higher validation accuracy

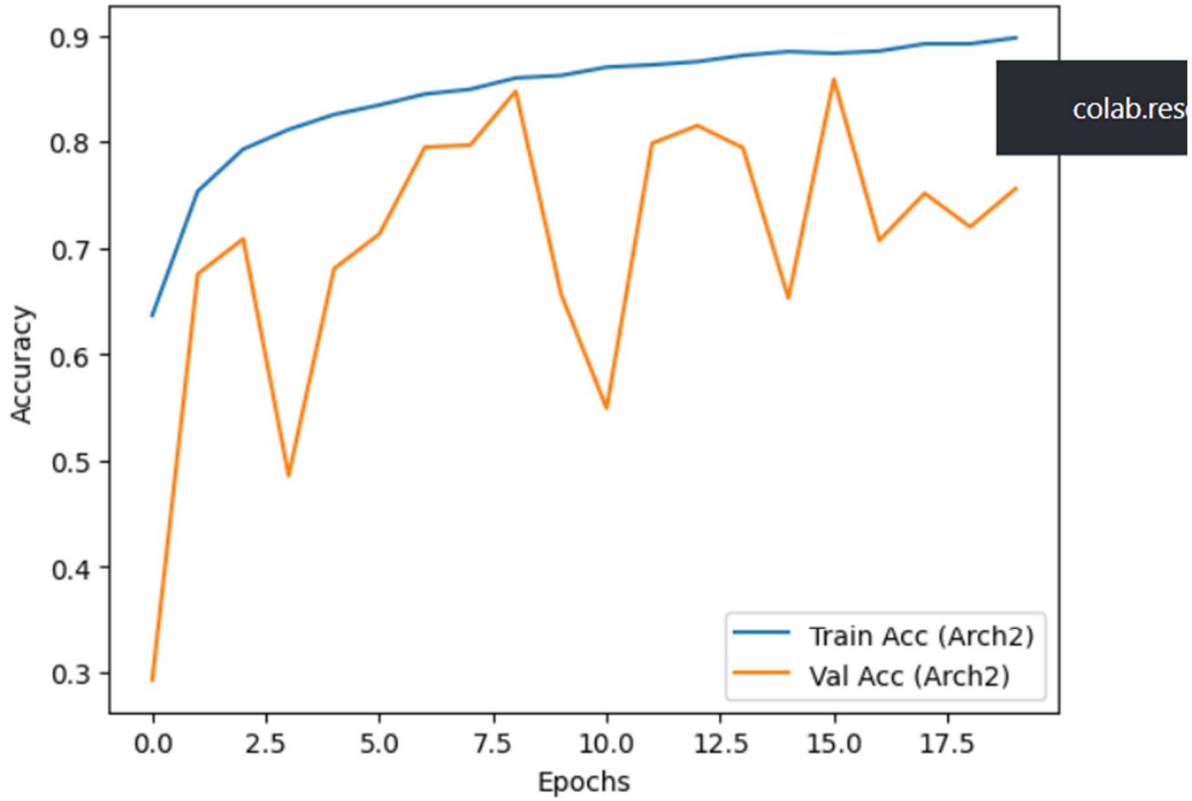
Training Curves:

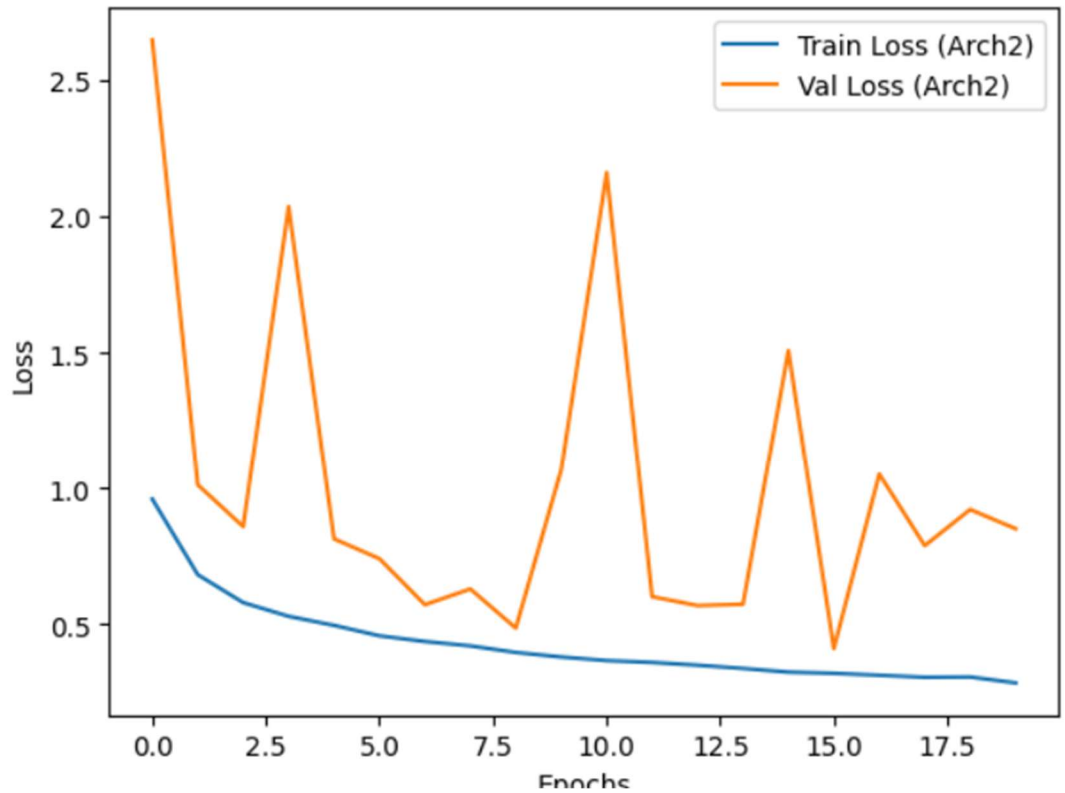
ARCHITECTURE-1:





ARCHITECTURE-2:

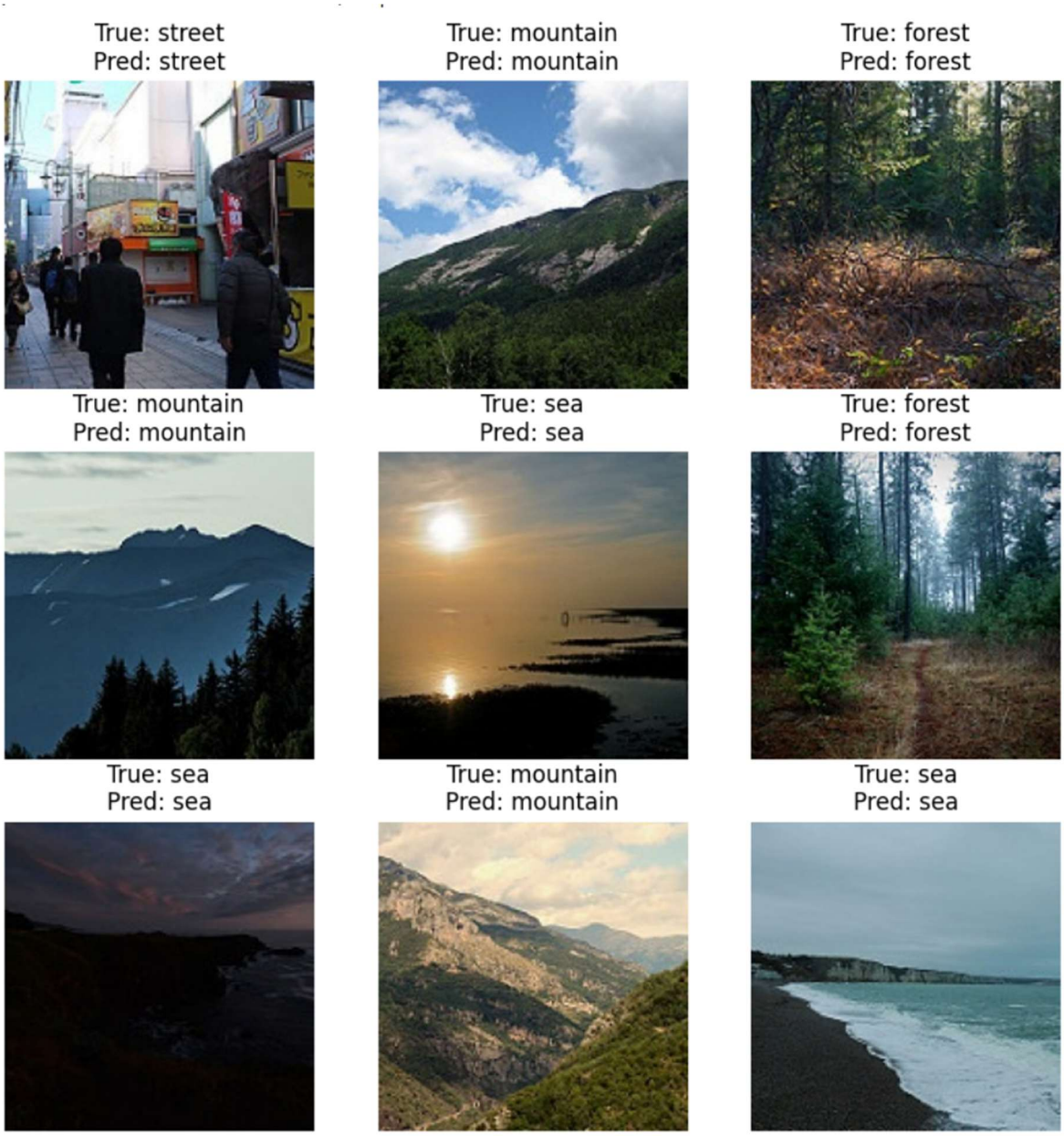




TRUE VS PREDICTED LABELS:

Architecture-1:

Architecture-2:



Choosing Initial Architecture

- Architecture 1 (Initial Choice):
  - 3 Conv2D layers (32 → 64 → 128) with MaxPooling
  - Flatten → Dense(128) → Dropout(0.5) → Output
- Reason for choosing:
  - Simple baseline to quickly test performance
  - Easier to train and observe initial results
- Evaluation Metrics Used:
  - Validation accuracy: 83%
  - Loss curves (training vs validation)
- Observations:
  - Worked well overall; captured basic features
  - Some individual predictions were wrong
  - Limitations: May not capture complex patterns due to simplicity

Changing Architecture

- Architecture 2 (Modified Choice):
  - Added 1 extra Conv2D layer (32 → 64 → 128 → 256)
  - Added BatchNormalization after each convolution
  - Replaced Flatten with GlobalAveragePooling
  - Dense(256) → Dropout(0.5) → Output
- Reason for Change:
  - Improve feature extraction and generalization
  - Reduce overfitting using BatchNormalization
  - Handle more complex patterns in the images
- Evaluation Metrics Used:
  - Validation accuracy: 75%
  - Loss curves (training vs validation)

    - Sample predictions to see qualitative performance
- Observations:
    - All sample prediction images were predicted correctly.
    - Validation accuracy decreased compared to Architecture 1 (likely overfitting or more tuning needed).
    - Demonstrates that deeper networks may require more epochs, data, or hyperparameter tuning to outperform simpler models.

Conclusion:

In this project, two CNN architectures were tested to classify images from the Intel Image Dataset into six categories. The initial simple CNN achieved a validation accuracy of 83%, capturing basic features well but making some mistakes on individual images. The deeper CNN with batch normalization and GlobalAveragePooling had a lower validation accuracy of 75%, but all sample predictions were correct, showing it learned better feature representations for certain images. This demonstrates that deeper networks do not always guarantee higher overall accuracy, and evaluating both quantitative metrics and qualitative predictions is essential. The project highlights the importance of iterative architecture design, evaluation, and improvement in multi-class image classification.