Remote Sensing Image Classification Using Down Sampled RSI-CB256

Project Description:

The objective of this capstone project is to develop and train a deep learning models for classifying remote sensing images using a down sampled version of the RSI-CB256 dataset. The dataset will be reduced to 15 classes and around 12,500 image samples to facilitate ease of implementation and accommodate available compute power. This project will leverage convolutional neural networks (CNN's) to accurately classify various land cover types, showcasing the practical application of deep learning techniques in remote sensing and geospatial analysis.

Additionally, to provide a comparison, we will implement an existing open-source model, ResNet-18, and evaluate its performance against my custom CNN model. This comparison will highlight the strengths and weaknesses of each approach and provide insights into model selection for remote sensing image classification tasks.

Why is it Good?

1. Practical Relevance: This project addresses real-world challenges in urban planning, environmental monitoring, disaster management, and agriculture by providing a robust classification model that aids efficient resource management and decision-making.

2. Technological Advancement: By applying cutting-edge deep learning techniques to high-resolution satellite imagery, the project demonstrates the potential of AI in processing and analyzing complex geospatial data, contributing to technological advancements in the field of remote sensing.

3. High-Quality Dataset: Utilizing the down sampled RSI-CB256 dataset with 15-20 classes and 12,500 images ensures a balanced and comprehensive dataset for training and evaluating deep learning models.

How Will You Do It?

1. Dataset Selection: The RSI-CB256 dataset will be down sampled to 15-20 classes and 12,500 image samples.

2. Data Preprocessing: Steps include resizing images to a consistent resolution (256x256 pixels), normalizing pixel values, and splitting the dataset into training, validation, and test sets. Data augmentation techniques such as rotation, flipping, and scaling will be applied to increase the diversity of the training data.

3. **Model Selection and Architecture**:

- **Custom CNN**: A convolutional neural network (CNN) will be designed and implemented using PyTorch, featuring multiple convolutional layers, pooling layers, and fully connected layers to capture spatial hierarchies. Regularization techniques like dropout will be used to prevent overfitting.
- **ResNet-18**: A pre-trained ResNet-18 model from the PyTorch torchvision.models library will be adapted for my classification task.

4. **Model Training**:

- **Custom CNN**: The model will be trained on the training dataset with optimized hyperparameters such as learning rate, batch size, and the number of epochs. The training process will be monitored using metrics like loss and accuracy, with early stopping considered to avoid overfitting.
- **ResNet-18**: The ResNet-18 model will be fine-tuned on the same training dataset, following a similar training process.

5. Hyperparameter Tuning: Hyperparameters Tuning will be experimented with to optimize the model's performance, recording and analyzing the impact of each.

6. Evaluation: The trained model's will be evaluated on the validation dataset using metrics such as accuracy, precision, recall, and F1-score. Visualizations of the model's predictions and misclassifications will be created to assess performance.

7. Fine-Tuning and Iteration: Based on evaluation results, the model architecture or hyperparameters will be adjusted and retrained as necessary to improve accuracy and generalization.

8. Final Model Testing: The final models will be tested on the held-out test dataset to assess generalization to unseen data.

9. Documentation and Reporting: A comprehensive project report will summarize the dataset, model architecture, training process, evaluation results, and insights gained, including visualizations and explanations.

10. Presentation: A brief presentation will showcase key findings and outcomes, sharing experiences, challenges faced, and lessons learned.

11. Conclusion: The project will conclude by summarizing achievements and suggesting future work or improvements.

Data to be Used:

The RSI-CB256 dataset, down sampled to 15-20 classes and 12,500 high-resolution satellite images, each with a resolution of 256x256 pixels. This ensures detailed and high-quality data for training deep learning models.

Evaluation of System Performance:

1. Accuracy: Measure the percentage of correctly classified images out of the total number in the test set.

2. Precision and Recall: Precision will measure the accuracy of positive predictions, while recall will measure the model's ability to find all relevant instances.

3. F1-Score: Calculate the mean of precision and recall to provide a balanced performance metric.

4. Confusion Matrix: Generate a confusion matrix to visualize the classification performance across different classes.

5. Visualization: Visually look at the model's predictions to understand strengths and weaknesses, and analyze misclassified examples to identify improvement areas.

This proposal outlines my comprehensive plan to utilize the down sampled RSI-CB256 dataset for developing an advanced remote sensing image classification model using deep learning techniques. By implementing and comparing a custom CNN with an established model like ResNet-18, this project aims to provide a thorough evaluation of deep learning approaches for remote sensing image classification, contributing valuable insights to the field.