ENSEEIHT

Machine learning

2023-2024

Land cover classification report

Oussakel Rachida

Contents

List of Figures

1 Introduction

The task of satellite image classification involves the identification and categorization of land cover types based on satellite imagery. This project focuses on leveraging machine learning techniques to classify various land cover types, including 'Forest,' 'River,' 'Highway,' and more, by analyzing the information captured in satellite images.
This report outlines the process of building and training a CNN model for land cover type classification using satellite images. It includes data loading and visualization, model building, training, evaluation, and inference steps.

2 Model selection

The model combines predictions from two different architectures: a custom CNN model tailored for the specific task and a pre-trained ResNet-50. By averaging predictions from these models, the approach aims to capitalize on their diverse architectures and learned representations to improve overall predictive performance.

The chosen **CNN model** comprises several convolutional layers, followed by max-pooling and batch normalization, culminating in fully connected layers for classification:

Layer Stacking : The progression from lower-level features to higher-level abstractions is achieved through a series of convolutional layers followed by pooling operations.

Batch Normalization : Included after each convolutional layer to accelerate training by normalizing inputs to each layer.

Global Pooling : Applied to condense feature maps into a vector for classification, capturing the most relevant information.

Dropout : Employed before the fully connected layers to prevent overfitting during training by randomly dropping units.

The **ResNet-50 model**, pretrained on ImageNet, is used for image feature extraction. The fully connected layer is replaced to suit the specific classification task, adapting to the desired number of output classes. Parameters are frozen to retain the pre-trained weights. This ResNet-50 variant's inclusion in the ensemble strategy aims to enhance overall predictive performance by leveraging its strong image representation capabilities in conjunction with a custom CNN model.

3 Data preparation

In the data preparation phase of our project, we employed a train-test split to partition our original dataset into training and validation sets. Subsequently, a series of image transformations were applied to convert the images into PyTorch tensors and normalize them. The normalization step, achieved through transforms.Normalize(mean=mean, std=std), improves the convergence and performance of deep learning models during training.

4 Training and parameters

Loss Function : Cross-entropy loss is used for multi-class classification tasks. It calculates the loss between the predicted probabilities and the ground truth labels.

Optimizer : Adam optimizeris employed with a learning rate of 0.001 and weight decay set to 1e-4. chosen for its adaptive learning rate method and efficient convergence.It updates model parameters using gradients computed during backpropagation.

Learning Rate Scheduler : Our training process incorporates a step-based learning rate scheduler, which decreases the learning rate by a factor of 0.9 every 5 epochs. This gradual reduction is designed to refine the learning process as the model approaches the optimum, allowing for finer adjustments in the later stages of training and potentially leading to better final performance.

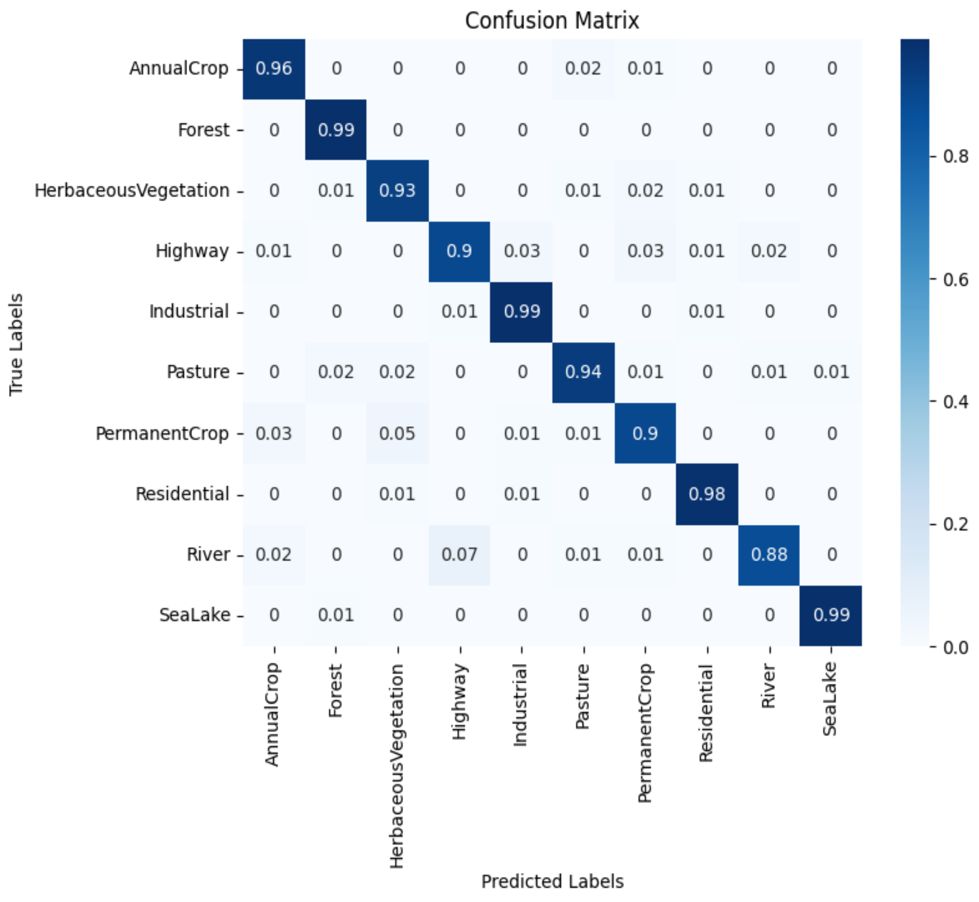
Training Loop: The model is trained over a course of 30 epochs, which is determined to be sufficient for convergence without overfitting, as per our validation checks. We process the data in batches of 128 samples, striking a balance between computational efficiency and the ability to generalize across the dataset. This batch size is selected to make the most of our available computational resources while still providing robust gradient estimates for the optimizer.

5 Model Evaluation

recall and F1-score :

With a precision of 94.64% , the model accurately identifies the specified class, showcasing remarkable correctness in positive predictions. Simultaneously, a recall score of 94.65% highlights its proficiency in capturing the vast majority of instances belonging to that class. These high precision and recall values, both around 94%, signify a well-balanced model, ensuring accurate identification while minimizing false positives and negatives. With an F1-score of 94.62%, this model exemplifies a robust balance between precision and recall, cementing its reliability in accurately classifying instances of the specified class.

Confusion matrix :



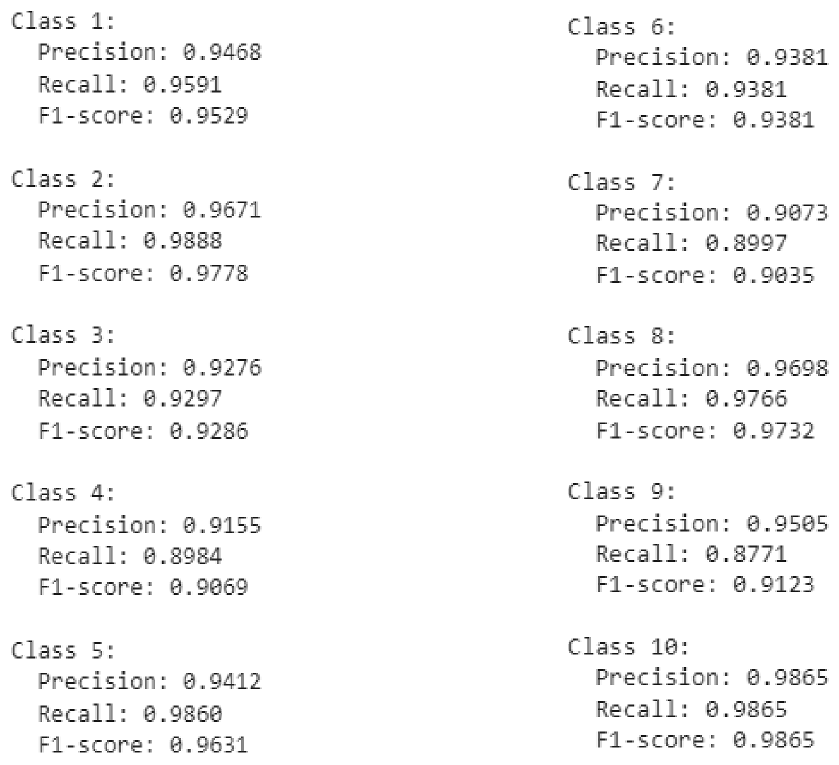
	AnnualCrop	Forest	HerbaceousVegetation	Highway	Industrial	Pasture	PermanentCrop	Residential	River	SeaLake
AnnualCrop	0.96	0	0	0	0	0.02	0.01	0	0	0
Forest	0	0.99	0	0	0	0	0	0	0	0
HerbaceousVegetation	0	0.01	0.93	0	0	0.01	0.02	0.01	0	0
Highway	0.01	0	0	0.9	0.03	0	0.03	0.01	0.02	0
Industrial	0	0	0	0.01	0.99	0	0	0.01	0	0
Pasture	0	0.02	0.02	0	0	0.94	0.01	0	0.01	0.01
PermanentCrop	0.03	0	0.05	0	0.01	0.01	0.9	0	0	0
Residential	0	0	0.01	0	0.01	0	0	0.98	0	0
River	0.02	0	0	0.07	0	0.01	0.01	0	0.88	0
SeaLake	0	0.01	0	0	0	0	0	0	0	0.99

Figure 1: Confusion matrix

The model exhibits strong diagonal values, indicating high accuracy for most classes. Classes like 'Forest' and 'SeaLake' have particularly high true positive rates, denoting that the model is very effective at classifying these types correctly.

6 Results analysis

6.1 Class-wise Performance



```
Class 1:
  Precision: 0.9468
  Recall: 0.9591
  F1-score: 0.9529

Class 2:
  Precision: 0.9671
  Recall: 0.9888
  F1-score: 0.9778

Class 3:
  Precision: 0.9276
  Recall: 0.9297
  F1-score: 0.9286

Class 4:
  Precision: 0.9155
  Recall: 0.8984
  F1-score: 0.9069

Class 5:
  Precision: 0.9412
  Recall: 0.9860
  F1-score: 0.9631

Class 6:
  Precision: 0.9381
  Recall: 0.9381
  F1-score: 0.9381

Class 7:
  Precision: 0.9073
  Recall: 0.8997
  F1-score: 0.9035

Class 8:
  Precision: 0.9698
  Recall: 0.9766
  F1-score: 0.9732

Class 9:
  Precision: 0.9505
  Recall: 0.8771
  F1-score: 0.9123

Class 10:
  Precision: 0.9865
  Recall: 0.9865
  F1-score: 0.9865
```

Figure 2: Metrics for each class

Based on the metrics above, we notice that the model exhibits strong performance across various classes, with particularly high precision and recall values. Notably, Class 2 showcases exceptional precision (96.71%) and recall (98.88%), resulting in an impressive F1-score of 97.78%. The overall robustness of the model is highlighted by consistent F1-scores above 90% for the majority of classes, indicating a well-balanced trade-off between precision and recall. Although Class 9 shows slightly lower recall (87.71%), the precision-recall-F1 balance remains satisfactory. Class 10 stands out with near-perfect precision, recall, and F1-score values at 98.65%. The model effectively discriminates between diverse land cover types.

6.2 Misclassification

Based on the confusion matrix in Figure 1 :

- There is some misclassification between similar classes such as 'River' and 'SeaLake', which is understandable given that both classes pertain to water bodies and might share similar features.
- The model appears to confuse 'AnnualCrop' with 'PermanentCrop' more than with other classes. This could be due to the similarity in the spectral signatures of the crops or the seasonal variations not being distinctly captured by the model.
- 'Highway' is often misclassified as 'Industrial', which could be due to the presence of roads in industrial areas, leading the model to learn an incorrect association.

7 Results visualisation

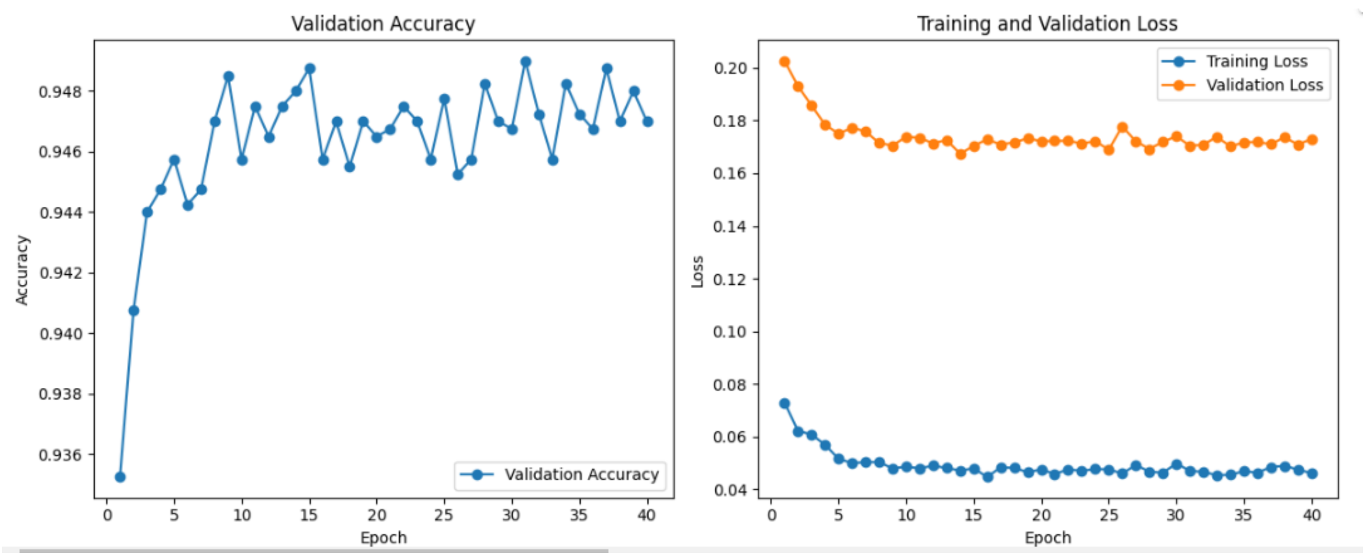


Figure 3: Learning curves

Validation Accuracy: The graph shows a rapid increase in accuracy initially, followed by fluctuations within a narrow range. The accuracy stabilizes quickly, indicating that the model reaches its effective capacity early.
Training and Validation Loss: There is a steep decline in training loss, demonstrating rapid learning in the initial epochs. The training loss plateaus, which is typical as the model begins to converge. Meanwhile, the validation loss decreases to a point and then remains relatively constant, with a slight but consistent gap above the training loss, indicating that the model may be starting to overfit as it is not improving on the validation set as much as on the training set.

8 Conclusion

The model showed particular strength in distinguishing certain land cover types, with class-wise performance metrics indicating robustness across various categories. However, there were notable instances of misclassification, particularly between similar classes like 'River' and 'SeaLake', suggesting limitations in the model's ability to differentiate between closely related features.
One key limitation of the model is its sensitivity to noise and potential overfitting, as indicated by the discrepancies between training and validation accuracy. Additionally, the misclassification between certain classes could be attributed to insufficient feature discrimination or class imbalance in the training dataset.
For future improvements, enhancing the dataset with more varied examples, especially for underrepresented classes, could address the class imbalance issue.