Machine Learning Project Report
Land Cover Classification

SOTORIVA LERMEN Thiago

November 17, 2023

Contents

1 Introduction

Land Cover Type Classification machine learning task of categorizing land cover types. The objective of this project is to develop a robust model capable of accurately classifying diverse land cover categories, including 'Forest,' 'River,' 'Highway,' and others, leveraging satellite images. The dataset provided for this project comprises labeled images, each associated with a specific land cover type. The ultimate aim is to train a machine learning model that can generalize well to predict the land cover type of unseen satellite images.

The main goals of this project are:

1. Develop a machine learning model for land cover type classification.
2. Explore and implement different model architectures using PyTorch.
3. Train the model on a labeled dataset and optimize its performance.
4. Evaluate the model's accuracy and generalization on both validation and test sets. In this case, the test metrics are done by the professors once the predictions were submitted.

While this project provides a flexible starting point with PyTorch for land cover classification, it is important to acknowledge certain limitations. The model's performance may be influenced by factors such as the quality and diversity of the training dataset, and the chosen model architecture and training parameters and techniques.

2 Model Selection

2.1 Overview of ML Models Considered

In the pursuit of an optimal solution for land cover type classification, various machine learning models were considered and experimented with. The following sections will explore the general idea of each model that was considered to be used.

2.1.1 Simple CNN Model

The first model examined was a simple CNN architecture with three layers. Despite its simplicity, this model failed to capture the intricate patterns present in the satellite images, leading to subpar classification accuracy. The limitations highlighted the need for more complex and expressive models.

2.1.2 Fine-tuning of Pre-trained Models

The approach of fine-tuning pre-trained models involves using established neural network architectures trained on large datasets (e.g. Imagenet) for general image recognition tasks and adapting them to the specific task of land cover type classification.

In this project, the following pre-trained architectures were considered to be used:

1. DenseNet: a densely connected convolutional network where the dense connections within the layers of DenseNet are designed to enhance feature reuse and facilitate gradient flow.
2. ResNet: the residual connections in ResNet models address the challenge of vanishing gradients, enabling the training of deeper networks.
3. EfficientNet: represents a scalable and efficient architecture that balances model depth, width, and resolution.

The model selection process involved iterative experimentation and evaluation, considering both model accuracy and computational efficiency. The following sections will delve into the chosen model architecture, detailing its structure and key features.

2.2 Model Choice and Architecture

After a careful exploration of various machine learning models for land cover type classification, the model of choice for this project is EfficientNetB0. The decision for EfficientNetB0 was influenced by several key factors, emphasizing both efficiency in training high-resolution images and promising results in terms of validation and test metrics. The architecture of EfficientNetB0 allows for quicker training times, making it well-suited for the constraints of the project environment, in the case of this project the Google Colab environment provides several limitations on the training process such as GPU and RAM memories. Also, the preliminary experiments with EfficientNetB0 demonstrated promising results in terms of model validation and test metrics.

3 Data Preparation

Data preparation is a crucial step in the machine learning pipeline, influencing the model's ability to learn and generalize effectively. The following sections will talk about the preprocessing steps applied to the dataset.

3.1 Data Format and Labels

The dataset is composed of satellite RGB images with a dimensional size of 64x64 split into two sets: training and validation set (20000 images) and validation set (7000 images). Each image in the dataset corresponds to one of the following classes: 'AnnualCrop', 'Forest', 'HerbaceousVegetation', 'Highway', 'Industrial', 'Pasture', 'PermanentCrop', 'Residential', 'River', 'SeaLake'. To facilitate model training, labels are transformed using one-hot encoding.

3.2 Data Preprocessing

Once retrieved, the data was sent to a preprocessing pipeline containing both normalization and augmentation steps. The normalization step is applied to ensure consistent and stable training across different images. The augmentation pipeline is employed to enhance the model's ability to generalize by exposing it to a variety of transformed versions of the training images. The augmentation techniques applied include random horizontal and vertical flips, random rotations, and resizing. All of these transformations were applied on the fly given a certain probability.

3.3 Data Splitting

The dataset is divided into training and validation sets to facilitate the training and evaluation of the machine learning model. The common practice of an 80-20 split is employed, where 16000 images are allocated to the training set, and the remaining 4000 images to the validation set.

4 Training Parameters

This section will present the main training parameters and techniques that were used to improve the model's training efficiency and accuracy.

4.1 Model Architecture

The chosen model architecture for this project is a CNN with an EfficientNet backbone. The main technique that was used on the model architecture was fine-tuning, and training the neural network with pre-trained weights to predict 10 classes. Also, to improve the model training efficiency and avoid overfitting, the weights of the first two bottleneck feature layers were frozen. It was done because the first layers of a neural network tend to work with high-spatial dimensions and also retrieve more general information. Thus, the idea was to improve the training efficiency to compute fewer gradients and focus on the deeper layers in the architecture which tend to extract more detailed information.

4.2 Training Loop and Hyperparameters

For the training loop, the choice of an appropriate loss function and optimizer is critical for model training. In this project, the CrossEntropyLoss is utilized, and the Adam optimizer with AMSGrad optimization is employed. Additionally, class weights are applied to address the class imbalance.

For the training loop, the following hyperparameters were defined:

1. Number of training epochs: 100
2. Batch size: 32
3. Learning rate: 3×10^{-3}
4. Weight decay: 10^{-5}

In addition, to prevent overfitting and improve training efficiency, an Early Stopping technique is employed. For that validation accuracy monitoring was implemented and if the accuracy does not improve for a specified number of consecutive epochs (the early stopping patience parameter was defined as 15 epochs) the training is finished.

Also, a custom learning rate scheduler is implemented to adjust the learning rate during training. The scheduler lowers the learning rate at specified epochs, aiming to refine the model's convergence. The scheduler adjusts the learning rate at defined intervals as follows:

1. If the current epoch < 20 then learning rate is multiplied by 10^{-1}
2. Else if the current epoch < 30 then learning rate is multiplied by 10^{-2}
3. Else the learning rate is multiplied by 10^{-3}

5 Results Analysis

This section will examine the main metrics for the two trained model architectures: EfficientNet-B0 and ResNet-50 using the following metrics for the analysis: Precision, Recall, F1-Score, Accuracy, Macro-average, and Weighted Average.

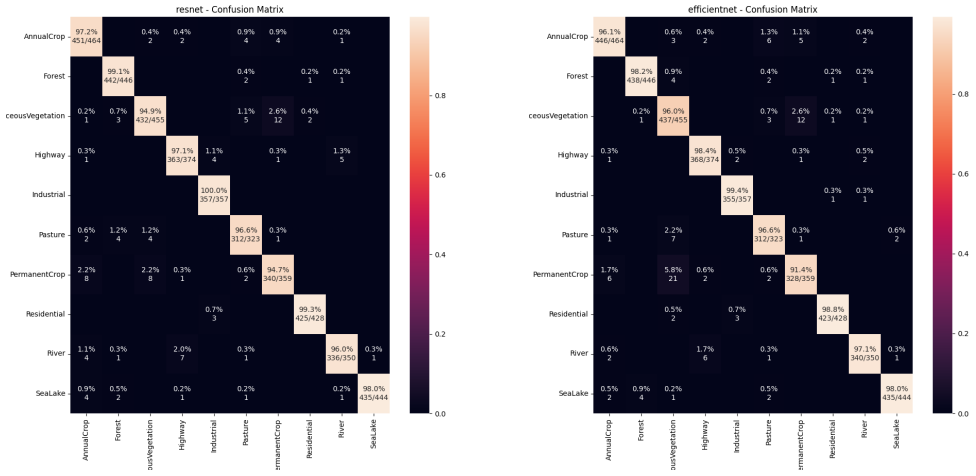
5.1 Evaluation Metrics

Table 1 presents the performance metrics of two trained models. As we can see, in terms of macro and weighted average both models indicate balanced performances across all classes reflecting their ability to well generalize not-seen data. In terms of accuracy, ResNet-50 marginally outperformed EfficientNet-B0 by 0.2%. This suggests that both architectures are effective in capturing the complexities of the proposed classification task.

Model	**ResNet-50**				**EfficientNet-B0**			
Class	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Accuracy**
AnnualCrop	0.96	0.96	0.96	0.972	0.97	0.96	0.97	0.961
Forest	0.98	0.99	0.99	0.991	0.98	0.98	0.98	0.982
HerbaceousVegetation	0.96	0.95	0.95	0.949	0.93	0.96	0.94	0.960
Highway	0.96	0.98	0.97	0.971	0.97	0.98	0.97	0.984
Industrial	0.99	0.99	0.99	1.000	0.99	0.99	0.99	0.994
Pasture	0.97	0.97	0.97	0.966	0.95	0.96	0.96	0.966
PermanentCrop	0.93	0.95	0.94	0.947	0.94	0.92	0.93	0.914
Residential	0.99	0.99	0.99	0.993	0.99	0.99	0.99	0.988
River	0.98	0.95	0.97	0.960	0.97	0.97	0.97	0.971
SeaLake	1.00	0.98	0.99	0.980	0.99	0.98	0.99	0.980
Macro-Average	0.97	0.97	0.97		0.97	0.97	0.97	
Weighted Average	0.97	0.97	0.97		0.97	0.97	0.97	
Accuracy				**0.972**				0.970

Table 1: Main metrics comparison between ResNet-50 and EfficientNet-B0

For further analysis, Figure 1 shows the confusion matrix of both models. As we can see, ResNet-50 has problems classifying the class 'HerbaceousVegetation' and 'PermanentCrop' with an accuracy of 94.9% and 94.7% respectively, classifying it wrongly and confusing between these two classes. On the other hand, EfficientNet can better generalize all classes with the exception of class 'PermanentCrop' with an accuracy of 91.4%, confusing mostly with the classes 'HerbaceousVegetation' and 'AnnualCrop'.



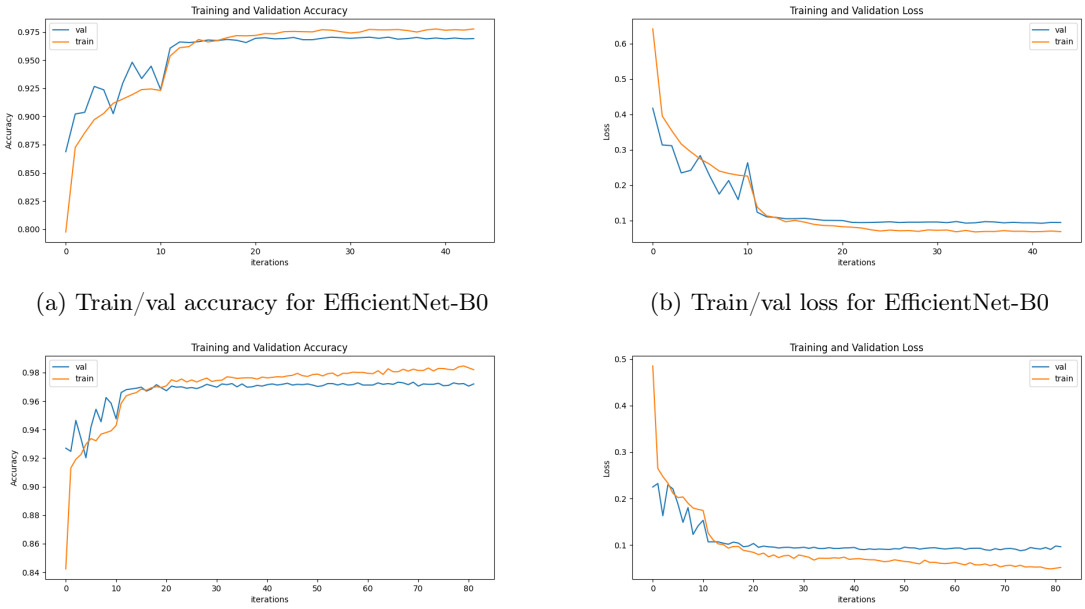
(a) CM for ResNet-50 (b) CM for EfficientNet-B0

Figure 1: Confusion Matrices for ResNet-50 and EfficientNet-B0

5.2 Visualizing the training process

Figure 2 shows the evolution of the accuracy and loss for both trained models. As we can see, EfficientNet-B0 demonstrates a normal training progression, with both training and validation accuracies steadily increasing and training and validation losses decreasing. The model exhibits stability around epoch 20, indicating that the model has converged to a reasonable solution without overfitting. The training process is efficient, with noticeable improvements in accuracy and loss during the early epochs.

On the other hand, ResNet-50, with its larger number of parameters, takes a longer time to train. The training and validation accuracies continue to increase, and the training and validation losses decrease consistently throughout the training process. ResNet-50 reaches stability around epoch 45, indicating that the model is effectively learning the underlying patterns in the data. Similar to EfficientNet-B0, there is no evidence of overfitting, and the model continues to improve its performance.



(a) Train/val accuracy for EfficientNet-B0 (b) Train/val loss for EfficientNet-B0

(c) Train/val accuracy for ResNet-50 (d) Train/val loss for ResNet-50

Figure 2: Train and validation loss and accuracies per epoch for ResNet-50 and EfficientNet-B0

6 Conclusion

In concluding the land cover type classification project, both ResNet-50 and EfficientNet-B0 demonstrated strong performance across precision, recall, F1-score, and overall accuracy metrics. While ResNet-50 achieved a slightly higher accuracy of 97.2%, EfficientNet-B0, with 97.0% accuracy, emerged as the preferred choice due to its faster training time, ease of training, and lighter model size.

EfficientNet-B0 reached stability around epoch 20, showcasing rapid convergence without overfitting, whereas ResNet-50 required a longer training period, attributed to its larger parameter count. The marginal difference in accuracy is outweighed by the practical advantages of EfficientNet-B0 in terms of computational efficiency.

In real-world applications, where speed and resource efficiency are critical, EfficientNet-B0's benefits make it the preferred model for deployment, striking a balance between accuracy and practical considerations.