Machine Learning Project Report

Jed Moutahir

November 2023

1 Introduction

The task at hand involves classifying land cover types, such as 'Forest' 'River' 'Highway' and more, based on satellite images. The dataset consists of images labeled with their corresponding land cover types. The goal is to train a machine learning model that can accurately predict the land cover type of new, unseen images.

The labeled data available is 20000 images for 10 classes :

- 'AnnualCrop'
- 'Forest'
- 'HerbaceousVegetation'
- 'Highway'
- 'Industrial'
- 'Pasture'
- 'PermanentCrop'
- 'Residential'
- 'River'
- 'SeaLake'

At first hand, it seems that having an accuracy over 95% might be quite hard. In fact, some classes seem to be quite close, for example 'Annual Crop' and 'Permanent Crop'.

Multiple models were tested but the selected one is a modified version of ResNet.

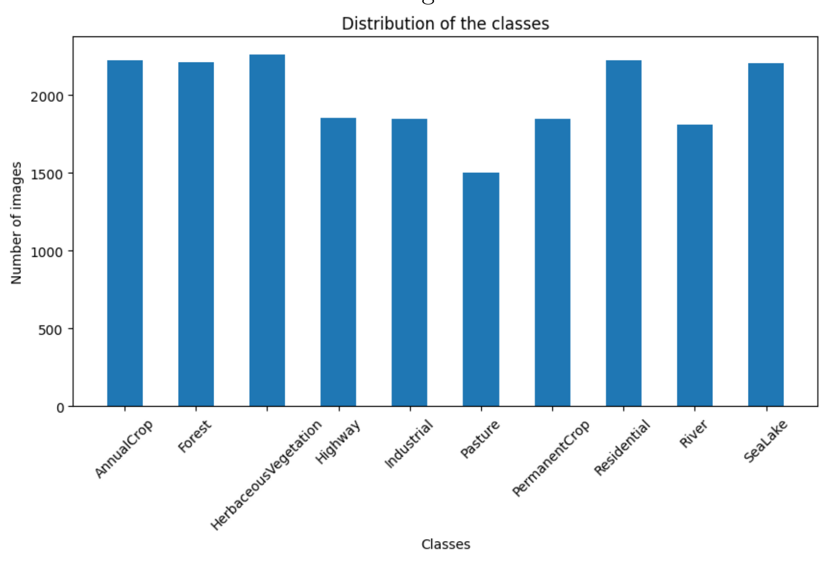
2 Model Selection

The chosen model is a modified version of ResNet for the classification task based on several key considerations.

ResNet is a deep convolutional neural network architecture known for its exceptional performance in image classification tasks. Its residual learning framework allows for the training of very deep networks without encountering the vanishing gradient problem, which is crucial for capturing complex features in images. Moreover, ResNet models are pre-trained on large datasets like ImageNet, making them adept at learning general image features and patterns, which can provide a substantial head start for our classification task. By fine-tuning a pre-trained ResNet on our specific dataset, we are able to harnessed the power of transfer learning, allowing the model to adapt to the unique characteristics of our images and the 10 target classes more efficiently. This approach not only saves training time and data but also often results in better classification performance.

3 Data Preparation

The data available is 20000 labeled images from 10 classes :



The data seems quite evenly distributed. Hence, no pre-processing were done for this.

The labels were encoded using $OneHotEncoder$ from the package $sklearn.preprocessing$.

The data was separated in 2 categories :

- Training (80%) used for training the Neural Network
- Validation (20%) used for validation during training

In order to augment the data available, data augmentation techniques were used using $transforms$ from the package $torchvision$: Random horizontal flips, Random vertical flips, Random rotations.

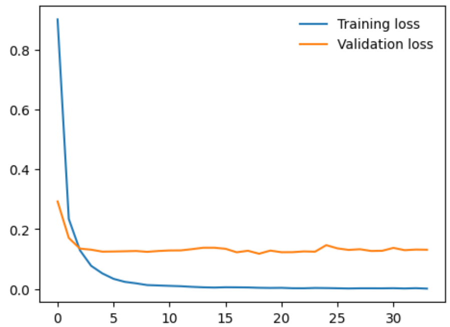
In order to adapt the images to the model input, with the same package, the images were resized to (224, 224). And then normalized.

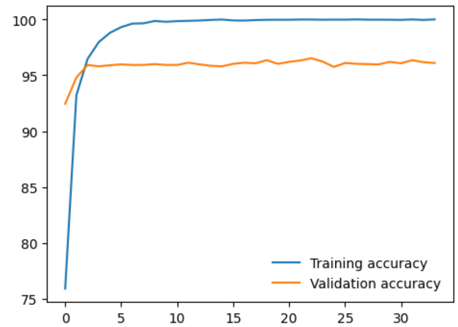
4 Training and Parameters

The loss used is the Cross Entropy (from the package *torch.nn*). It is a usual loss used for classification problems.
The optimizer used is Adam :
Adam is a widely adopted optimization algorithm that combines the benefits of both the RMSprop and Momentum algorithms. It is well-suited for training deep neural networks and offers several advantages. Adam adapts the learning rate for each parameter during training, which can lead to faster convergence and better optimization compared to fixed learning rates. The momentum term helps the optimizer overcome local minima and accelerate convergence.
The choice of a learning rate of 10^{-5} is a hyperparameter setting that balances the trade-off between convergence speed and stability. It was found great to be this small to avoid unstability.
A weight decay of 10^{-4} is included as a regularization technique to prevent overfitting during training. Weight decay (or L2 regularization) adds a penalty term to the loss function that discourages large weight values. This helps the model generalize better to unseen data by preventing it from fitting the training data too closely, which can lead to overfitting.

An early stop was incorporated as a way to select the best model and avoid overfitting.

Here is the evolution of the loss and the accuracy during training :



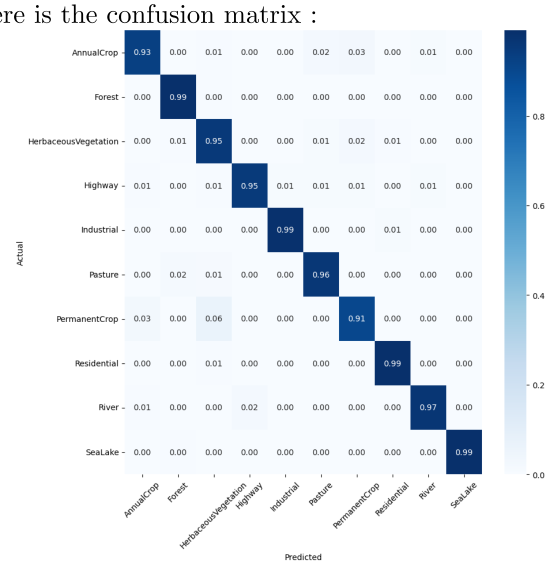


5 Model Evaluation

The model was evaluated on the Test dataset. Hence, it was unseen data.

The accuracy is 96%

Here is the confusion matrix :



The results show that the model is really good for every classes. However, it seems that some classes are confusing the model, for example : 'Annual Crop', 'Permanent Crop' and 'Herbaceous Vegetation'. After displaying the wrong classification for these classes, we can see that multiple reason could justify these results : These images really could blend for all 3 classes, The labelisation should not be taken as perfect and it could be a source of error, The limit between as 3 classes is quite hard to identify with such low resolution $(64 * 64)$

6 Failed attempts at fixing getting better results

6.1 Freezing layers

The attempt to employ freezing layers, a technique where certain layers of a neural network remain static during training, encountered significant challenges leading to its failure. The misjudgment of layer dependencies and the inadequacy of the chosen frozen layers disrupted the model's adaptability and hindered its capacity to learn new, discriminative features. Additionally, the dynamic nature of the dataset contributed to the failure, as the initially frozen layers were unable to encapsulate the evolving features necessary for accurate classification. In summary, the failure of the freezing layers method highlights the importance of accurately assessing layer dependencies, selecting appropriate frozen layers, and ensuring adaptability to dynamic datasets.

6.2 "Superclasses"

Since after observation a couple classes were found quite closely, an attempt at creating so called "superclasses" was done. With one general model and a model to differienciate each images in these classes theoritacally, this method should be more efficient in handling the similarities between closely related classes. By establishing superclasses, the goal is to leverage shared features and characteristics among these classes, enabling a more streamlined and generalized approach in the overall classification process. The general model at the superclass level can capture common patterns and traits that are consistent across the closely related classes. Meanwhile, the individual models dedicated to differentiating within each superclass can focus on the nuances and distinctive attributes specific to each class, providing a fine-grained analysis. This hierarchical structure aims to strike a balance between capturing overarching similarities and preserving the unique details of each class. Additionally, it offers the advantage of reducing redundancy in the learning process, as the general model can learn and generalize from a broader perspective, while the subclass models can fine-tune the classification based on the finer details. The effectiveness of this method is contingent upon the careful curation of superclasses and the proper assignment of images to each category within the hierarchy, ensuring a cohesive and meaningful organization of the dataset.

Unfortunately it failed at being better than the actual model :

- Oversimplification of the Problem: The attempt failed due to an initial oversimplification of the image classification problem. Assuming that a single general model could effectively capture the diverse features of closely related classes proved to be overly optimistic.

- Inadequacy of General Model: The general model designed for superclasses struggled to capture the intricate details and subtle variations within each closely related class. This limitation hindered the model's ability to provide accurate and specific classifications.

- Insufficient Hyperparameter Tuning: The attempt fell short in achieving optimal performance due to a lack of emphasis on fine-tuning hyperparameters. Achieving an accuracy threshold of over 96% requires a meticulous optimization of parameters such as learning rates, batch sizes, and regularization techniques. Neglecting this crucial step hindered the model's capacity to reach the desired level of accuracy.

6.3 Other models

Simpler models like small CNNs were trained but their accuracy never went over 90%. This is mainly because Resnet is a deep and highly expressive architecture that captures intricate features, while simpler models like small CNNs lack the depth and capacity to represent complex patterns. The superior performance of ResNet highlights the importance of model complexity in handling the intricacies of the dataset. The attempt with smaller CNNs underscores the significance of selecting models that strike a balance between simplicity and expressiveness to achieve optimal accuracy in image classification tasks.

6.4 More data Augmentation

Adding more data augmentation techniques was counterproductive. It ended up confusing the model more than helping training. This i maily due to the fact that they added images that had nothing to do with the actual dataset and made it deviate from its intended learning patterns. The inclusion of excessive data augmentation introduced unrelated variations, causing the model to struggle with discerning relevant features. This counterproductive effect is attributed to the introduction of images that diverged significantly from the characteristics of the actual dataset. Instead of enhancing the model's generalization ability, the additional augmentation complexities hindered the learning process, emphasizing the importance of judiciously selecting and applying augmentation techniques that align with the inherent patterns of the dataset.

7 Conclusion

In summary, the model achieved an impressive 96% accuracy in classifying images, demonstrating its effectiveness. However, it's important to acknowledge that there are limitations, as some misclassifications remain. However the problem at hand is quite complex since getting to 95% is "quite easy". However from 95% to 96% is a more formidable challenge, requiring a nuanced understanding of the dataset's intricacies. While the model showcased its effectiveness with a commendable 96% accuracy, the acknowledgment of remaining misclassifications underscores the inherent complexity of the task. Moving from 95% to 96% accuracy represents a significant leap, demanding fine-tuning and optimization efforts. The recognition of this challenge highlights the diminishing returns and increased difficulty in achieving higher accuracies as the model approaches its performance ceiling. Reaching accuracy levels beyond 96% poses an increasingly challenging endeavor in image classification. The pursuit of higher accuracy demands a meticulous approach, involving sophisticated model architectures, advanced optimization techniques, and potentially novel strategies. Incremental gains beyond 96% require a deep understanding of subtle patterns within the dataset, fine-tuning hyperparameters to their optimal values, and exploring state-of-the-art methodologies. Moreover, the law of diminishing returns becomes more pronounced, making each percentage point a harder-fought achievement. Balancing model complexity with interpretability and avoiding overfitting becomes crucial. As the quest for higher accuracy continues, researchers must navigate the trade-offs and complexities inherent in pushing the boundaries of image classification performance.