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.

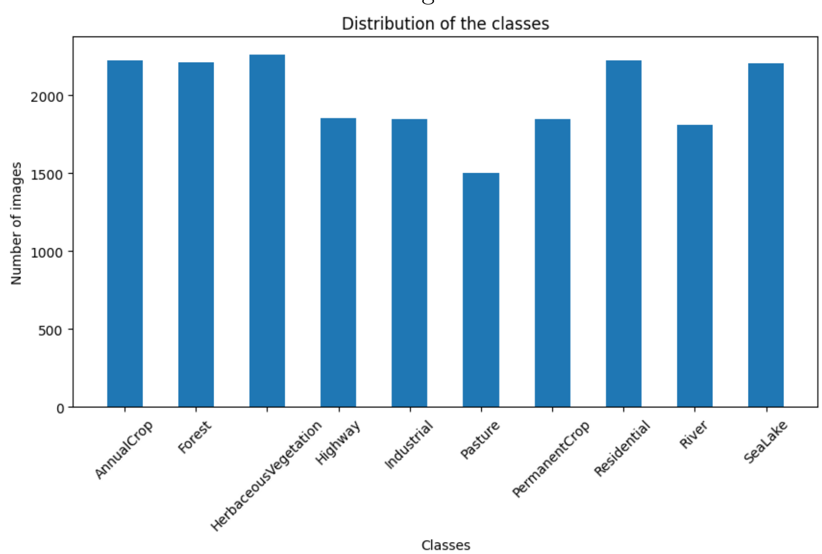
2 Model Selection

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.

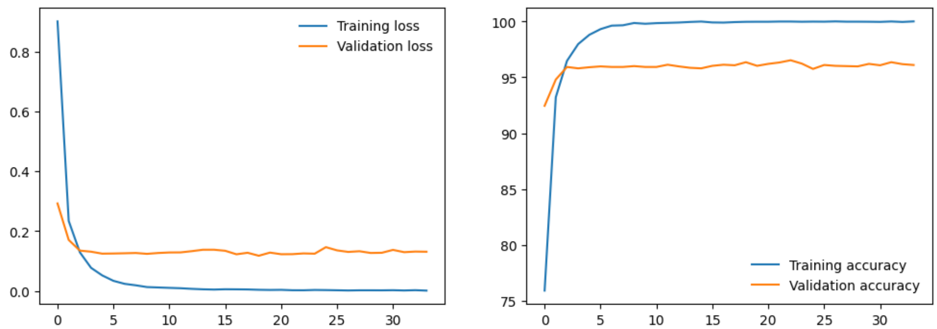
In order to adapt the images to the model input, with the same package, the images were resized to (224, 224). And then normalized.

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 0.00001 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 0.0001 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 :

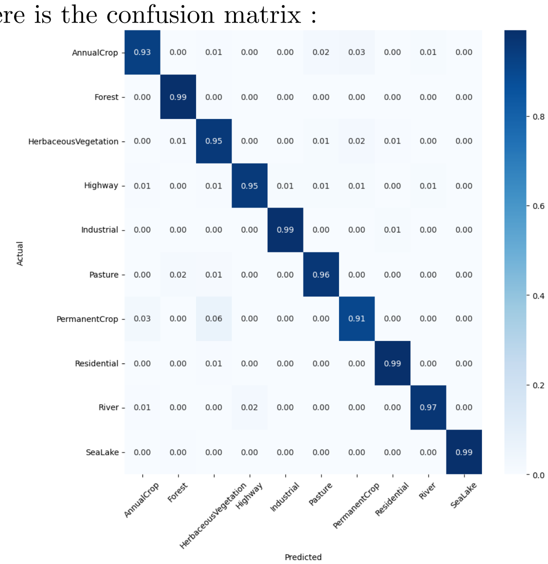


5 Model Evaluation

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 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. To enhance the model, we can explore parameter fine-tuning, more data augmentation, noise robustness and further analysis of misclassified samples. These steps are crucial for improving the model's reliability and overall performance.