Deep Learning - Land Cover project Report:

Authors: Haliloua Othmane, El Alout Ismail

3SN- HPC & BIG DATA

December 2023

Contents

1 Introduction:

Having a Data set of RGB images, each instance of size, representing 10 different types of landscapes: AnnualCrop, Forest, HerbaceousVegetation, Highway, Industrial, Pasture, PermanentCrop, Residential, River, SeaLake), we want to build a Deep Learning model capable of classifying Land images according to their appropriate classes. This computer vision problem would require a convolutional neural network, some of the challenges we could face in this project is the limited expressiveness of the data, for instance, if the percentages of instances for each class are vastly disproportionate, as this doesn't allow the model to learn as much patterns in some classes as good as we'd want, especially if the data in hand is very limited; From another perspective, the computational aspect comes to play just as significantly, we don't want to use a simplistic model (Less computational overload) and therefore cause underfitting, at the same time we don't want to use an overly complex model (More computational overload) that would cause overfitting, taking that into account, we should abide by a trade-off between the two extremes, lastly, since we are using a CNN there will be a a big overload on the RAM.

2 Model Selection

2.1 Architecture of the ResNet50 model

ResNets are known as deep neural networks which can achieve high accuracy for many computer vision tasks, such as image classification, object detection, medical imaging... The special point about ResNets is the residual blocks which were first introduced to solve the vanishing gradient problem. The figure below shows the architecture of ResNet50.

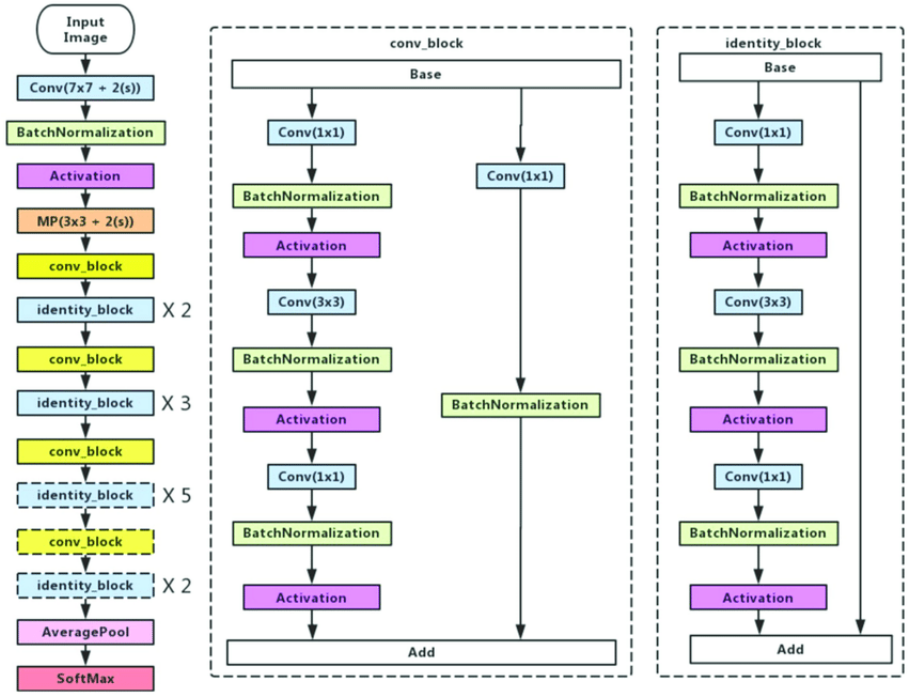

Figure 1: ResNet50 architecture

ResNet50 enables faster training for each layer as it uses a bottleneck design (1x1 convolutions) which reduces the number of parameters.

2.2 The pre-trained aspect of the used ResNet50

ResNet50 is pretrained on large image data sets such as ImageNet, which consists of millions of labelled images across thousands of classes. ResNet50 serves as a feature extractor and its pretraining on ImageNet enables transfer learning which we tried to apply in this project (fine-tuning on a specific dataset in order to improve performance).

3 Data Preparation

The provided training Data set consists of 20000 labeled images captured with satellites (64x64 RGB images). First, the Data set is split into a training set and validation set with 80% of the original data for the training set and 20% for the evaluation (validation). In addition to that, we performed data augmentation on these sets in order to enhance the model's generalization and improve its performance on unseen data. The data augmentation techniques used are image rotation (which helps the model become invariant to orientations), horizontal and vertical flips... [change if we add anything]. Then, we applied the normalization process on the images, which brings pixel values into a standard range centered around 0 accelerating the training of the network.

4 Training and Parameters

We employed two different training methods:

4.1 Cross Validation:

Although it isn't always feasible since it consists of retraining the model several times, K-fold cross-validation is an efficient way of tackling the challenges that we cited before in the introduction. It helps get efficient use of limited data by repeatedly splitting smaller samples of the data into training and validation sets, the model therefore the model's learning is intensified using small chinks of data. On another note the average performance across folds helps properly assess the bias (underfitting) and the variability in performance across folds helps us balance the other side by properly estimating the variance (Overfitting).

4.1.1 Early stopping callback and model selection

The early stopping callback is an approach employed during the training of machine learning models to prevent overfitting. It is involves monitoring a chosen performance metric: in our case we chose validation loss. If the validation loss ceases to decrease after a certain number of epochs (patience parameter which we can fix), then the training is halted to avoid overfitting the model. Another advantage of this technique is reducing the training time. For the model selection, we train a model during each fold and we save the model with the highest validation accuracy (or lowest validation loss). An important condition to get better results is to ensure the variability of the different subsets in each fold.

4.1.2 Optimizer and Parameters

The optimizer chosen for the training process is the Adam optimizer (Adaptive Moment Estimation), with cross entropy loss function (commonly used in classification tasks) which encourages the model to produce well-calibrated probability estimates. The learning rate is a crucial hyperparameter that controls the training process of the model, in our case 0.0001 is the learning rate that produces the best results. The weight decay parameter adds a regularization (L2 penalty) to our loss function based on the magnitudes of the weights (penalizes larges weights) and helps prevent overfitting and improves the generalization ability of the model. We also used the AMSGrad variant of the Adam optimize; its main advantage is that is helps maintain the stability of the learning rate, so potentially improving convergence.

4.2 Max accuracy training:

After each epoch we check if the new model reaches a new maximum for accuracy, if it does we save the model and use it as a starting point for the next epoch. The last saved model that recorded a maximum for accuracy is the model that we chose. It helps avoid the tediousness of k-fold cross-validation, as this method reaches similar promising results in a considerably fewer number of epochs.

5 Model Evaluation

5.1 Evaluation metrics

The weighted average of precision (0.95), recall (0.95) and F1 score (0.95) suggests that the model performs well across all classes, achieving both high precision and recall, with a balanced F1 score. Precision indicates how many of the predicted positive instances are relevant. Recall is the ratio of correctly predicted positive observations. F1 Score provides a balance between precision and recall and a higher F1 score suggests a strong balance between precision and recall which is the case for our model.

```
Class 0: Precision=0.93, Recall=0.96, F1 Score=0.95, Support=453
Class 1: Precision=0.99, Recall=0.98, F1 Score=0.99, Support=485
Class 2: Precision=0.93, Recall=0.93, F1 Score=0.93, Support=463
Class 3: Precision=0.96, Recall=0.90, F1 Score=0.93, Support=356
Class 4: Precision=0.89, Recall=0.99, F1 Score=0.94, Support=372
Class 5: Precision=0.97, Recall=0.95, F1 Score=0.96, Support=284
Class 6: Precision=0.88, Recall=0.94, F1 Score=0.91, Support=349
Class 7: Precision=0.99, Recall=0.89, F1 Score=0.94, Support=454
Class 8: Precision=0.95, Recall=0.94, F1 Score=0.95, Support=356
Class 9: Precision=0.99, Recall=0.96, F1 Score=0.98, Support=428
```

Figure 2: precision, recall, F1-score and support for each class. (K-fold cross validation) with K = 5

```
Class 0: Precision=0.94, Recall=0.94, F1 Score=0.94, Support=445
Class 1: Precision=0.98, Recall=0.98, F1 Score=0.98, Support=443
Class 2: Precision=0.94, Recall=0.81, F1 Score=0.87, Support=453
Class 3: Precision=0.95, Recall=0.80, F1 Score=0.87, Support=370
Class 4: Precision=0.96, Recall=0.95, F1 Score=0.96, Support=370
Class 5: Precision=0.94, Recall=0.91, F1 Score=0.93, Support=3P*
Class 6: Precision=0.79, Recall=0.93, F1 Score=0.86, Support=3.
Class 7: Precision=0.92, Recall=1.00, F1 Score=0.95, Support=445
Class 8: Precision=0.91, Recall=0.96, F1 Score=0.93, Support=362
Class 9: Precision=0.97, Recall=0.99, F1 Score=0.98, Support=441
```

Figure 3: precision, recall, F1-score and support for each class. (Max accuracy training)

5.2 Confusion Matrix

Figure 4: Confusion matrix (K-fold cross validation) with K = 5

Figure 5: Confusion matrix (Max accuracy training)

5.3 Results analysis

The evaluation metrics and the confusion matrix above show that the model selected through K-fold cross validation had a good performance in overall, with an accuracy higher than 0.92 for 9 classes. The confusion matrix shows where our model struggles a bit: Pasture (0.90).

5.4 Results Visualisation

5.4.1 Method 1 : Cross validation

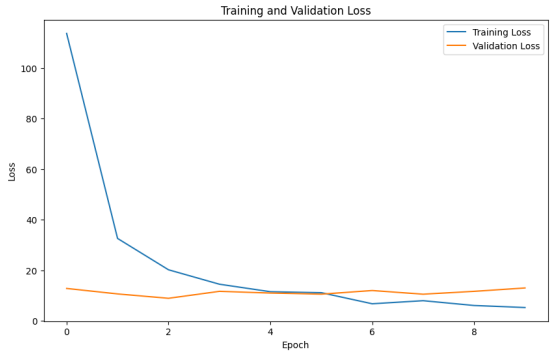

Figure 6: Evolution of the training and validation loss (the mean losses across all folds)

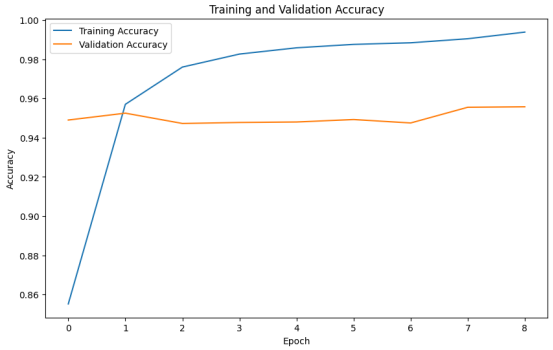

Figure 7: Evolution of the training and validation accuracy (the mean accuracies across all folds)

5.4.2 Method 2 : Max accuracy training

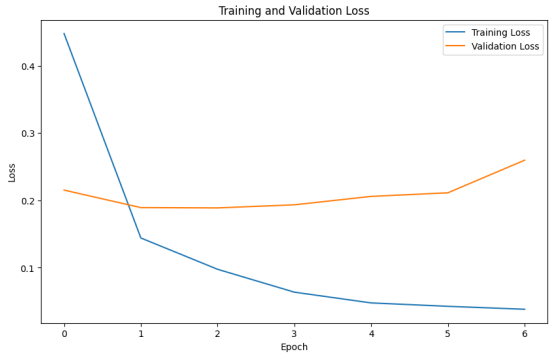

Figure 8: Evolution of the training and validation loss

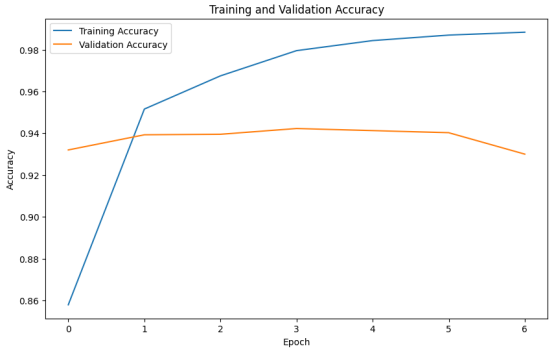

Figure 9: Evolution of the training and validation accuracy

6 Conclusion

The selected model from K-fold cross validation reaches 95.9% accuracy. This suggests that the model's performance is good in overall. Yet, we can consider other aspects in order to see if we can get a better accuracy : for example the the class with the lowest accuracy for this model was the "PermanentCrop" class, we can adapt the transformations applied in the data preparation step in order to help our model explore other features in the augmented data.

7 Work sharing:

There wasn't a specific list of tasks for each member, however, we collectively implemented the base code, and then we recurrently ameliorated it, as we alternated sessions for code modifications.