Assessment of the EuroSAT Dataset by applying Deep Learning Techniques

E. Emezina[a]

[a] Faculty of Geosciences and Environment (FGSE), Université de Lausanne (UNIL)

Institute of Earth Surface Dynamics *

Keywords: LULC; CNN; ResNet50; EuroSAT

Abstract

This project investigates the effects of hyper-parameters of two deep learning algorithms (i.e., an ANN baseline model and a CNN model)on the EuroSAT dataset. The dataset has 10 classes of Land Use and Land Cover (LULC) Classifications. Evidently, the pretrained ResNet50 model, performed better with an accuracy of 96.26% using the SGD optimizer. Fine-tuning the hyper-parameters on the training and validation datasets showed different results. This project is relevant for remote sensing tasks concerning LULC change from any satellite imagery with the RGB bands.

1 Introduction

Over the past decade, remote sensing (RS) computing has enabled users to instantly access and process vast amounts of data from traditional workstations with state-of-the-art (and often very expensive) hardware and RS software. Now we have moved to a cloud-based platform that enables easier access processed geo-data via user-friendly web-based interface and effective scripting languages (Tassi and Vizzari (2020)). The approach used was based on a study by (Helber, Bischke, Dengel, and Borth (2019)) which addresses the challenge of land use and land cover classification using Sentinel-2 satellite images. The satellite images obtained from Sentinel-2 are openly and freely accessible in the Earth observation program Copernicus. In a large region of interest, earth observation through satellite systems based on medium-resolution multi-spectral scanners is one of the most efficient and economical ways to obtain relevant information about terrestrial natural resources and the vegetation conditions (Mallmann, PRADO, and Pereira Filho (2015)). Therefore, the use of deep learning algorithms on satellite imagery of a given region can provide valuable information the land features.

This study uses a convolutional neural network model called ResNet50 which is effective for remote sensing tasks and further fine-tuning was applied to the model to see which verison of the model was the most efficient for the EuroSAT dataset.

2 Data

The dataset is based on Sentinel-2 satellite images covering 13 spectral bands and consisting out of ten LULC classes with a total 27,000 labeled and geo-referenced images. However, for this project only data from three bands of the Sentinel-2 satellite were used notably the Red, Green and Blue bands (link to the code at the end of the paper).

*Author correspondence: emmanuel.emezina@unil.ch

Moreover, the proposed dataset is based on open and freely accessible earth observation data, enabling a unique range of real-world applications. The patches included in the dataset are from cities spread over his 30+ countries in Europe, based on the coverage of the European Urban Atlas (Helber, Bischke, Dengel, and Borth (2018)). The ten classes in this data are as follows: Industrial , Residential,Annual Crop, Permanent Crop, River , Sea Lake, Herbaceous Vegetation, Highway , Pasture as well as Forest and they are illustrated in the figure 1 below

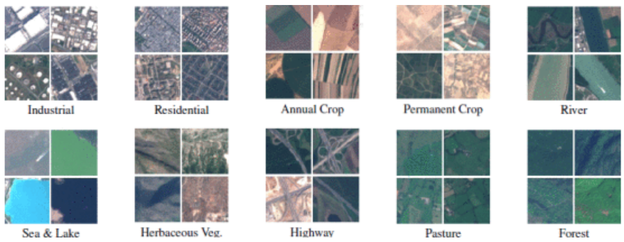


Figure 1: Illustration of a the ten classes used in EuroSAT dataset from Sentinel-2 images?helber2018introducing)

3 Methods

The method workflow consists of main steps which were written in the Python environment using Google Colab Interface (link to the code at the end of the paper)

1. Accessing the EuroSAT dataset via tensorflow dataset. The dataset is built-in in this module.
2. The next step involves the application of cross-validation. The dataset was divided into training (80%), validation (10%) and testing (10%) sets.
   The idea behind the cross-validation is to provide an estimate of the model's performance on data which has not been seen.We can then then obtain a better understanding of how well the model generalizes to new data.
3. The definition of certain variable used for the ResNet model; notably the number of epochs, the batch size, the buffer size and the image size which was constant through out the analysis.
4. The next step involved the application of data augmentation. The "tf.image.resize" function in the Tensorflow module was used to resize the target images. This technique is used to artificially increase the size of the dataset by generating modified versions of existing data in order to improve the performance of a model.
5. Building the ResNet50 involved the use of Keras which have pretrained models.
   These pretrained models were trained using the Imagenet

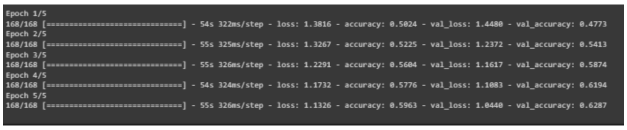
dataset(a large dataset of images that is commonly used for training computer vision models) which has weights from prior training. Its important to note that the model was altered with and without the weights to see which version performed better in the evaluation with the testing dataset.

6. Subsequently the model was compiled using certain hyperparameters (notably loss, optimizer and metrics).
   The optimizer was altered between Adam and Stochastic Gradient Descent (SGD) to observe which version performed better during evaluation
7. The models' variants were fitted, run and plotted to view the train and validation accuracy and losses.
8. Finally the model was evaluated using the test dataset and tested to see its prediction.

4 Results

4.1 Significance of the weights

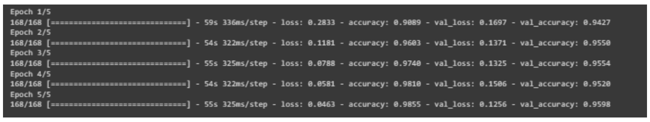
The first try to fine-tune the model was to see the effect of the weights on the training and validation sets.



```
Epoch 1/5
168/168 [==============================] - 54s 322ms/step - loss: 1.3816 - accuracy: 0.5024 - val_loss: 1.4480 - val_accuracy: 0.4773
Epoch 2/5
168/168 [==============================] - 55s 325ms/step - loss: 1.3267 - accuracy: 0.5225 - val_loss: 1.2372 - val_accuracy: 0.5413
Epoch 3/5
168/168 [==============================] - 55s 326ms/step - loss: 1.2291 - accuracy: 0.5604 - val_loss: 1.1617 - val_accuracy: 0.5874
Epoch 4/5
168/168 [==============================] - 54s 324ms/step - loss: 1.1732 - accuracy: 0.5776 - val_loss: 1.1083 - val_accuracy: 0.6194
Epoch 5/5
168/168 [==============================] - 55s 326ms/step - loss: 1.1326 - accuracy: 0.5963 - val_loss: 1.0440 - val_accuracy: 0.6287
```

Figure 2: Illustration of train and validation data metrics for their loss and accuracy after every epoch without the pre-trained weights.

Its also important to note that this involved the use of "adam" as optimizer, "accuracy" as metrics and "Sparse Categorical Cross-entropy" as loss.

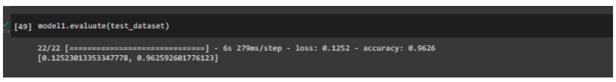


```
Epoch 1/5
168/168 [==============================] - 59s 336ms/step - loss: 0.2833 - accuracy: 0.9089 - val_loss: 0.1697 - val_accuracy: 0.9427
Epoch 2/5
168/168 [==============================] - 54s 322ms/step - loss: 0.1181 - accuracy: 0.9603 - val_loss: 0.1371 - val_accuracy: 0.9550
Epoch 3/5
168/168 [==============================] - 55s 325ms/step - loss: 0.0788 - accuracy: 0.9740 - val_loss: 0.1325 - val_accuracy: 0.9554
Epoch 4/5
168/168 [==============================] - 54s 322ms/step - loss: 0.0581 - accuracy: 0.9810 - val_loss: 0.1506 - val_accuracy: 0.9520
Epoch 5/5
168/168 [==============================] - 55s 325ms/step - loss: 0.0463 - accuracy: 0.9855 - val_loss: 0.1256 - val_accuracy: 0.9598
```

Figure 3: Illustration of train and validation data metrics for their loss and accuracy after every epoch with the pre-trained weights.

4.2 Significance of the hyperparameters

The choice of optimizer was alternated between Stochastic gradient descent (SGD)4 and Adam (Adaptive Moment Estimation)5. The ResNet50 model loss and accuracy when using the SGD and Adam on the test data are illustrated below.

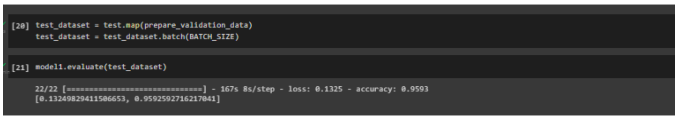


```
[49] model1.evaluate(test_dataset)

22/22 [==============================] - 6s 279ms/step - loss: 0.1252 - accuracy: 0.9626
[0.12523013353347778, 0.962592601776123]
```

Figure 4: Evaluation score on test data when using SGD optimizer.
Illustration the loss and accuracy

The final step involved the predictive accuracy of the ResNet50 model and it provided images of the different classes based on its accuracy level. The image 6 shows an example of classes predicted using the "Adam" optimizer. The red labelled class at the top right corner is a wrong prediction.



```
[20] test_dataset = test.map(prepare_validation_data)
     test_dataset = test_dataset.batch(BATCH_SIZE)

[21] model1.evaluate(test_dataset)

22/22 [==============================] - 167s 8s/step - loss: 0.1325 - accuracy: 0.9593
[0.13249829411506653, 0.9592592716217041]
```

Figure 5: Evaluation score on test data when using Adam optimizer.
Illustration the loss and accuracy

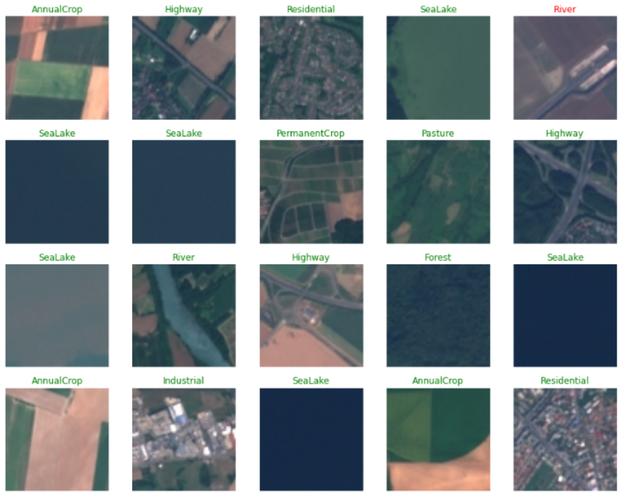


Figure 6: Evaluation score on test data when using Adam optimizer. // Illustration the loss and accuracy

5 Discussion

In general, it is recommended to use pre-trained weights as a starting point when training a CNN for image classification.In image 2 the weights were ignored leading to a poor accuracy after 5 epoch while the inclusion of the weights in image 3.
These weights are typically trained on large datasets such as ImageNet. The imagenet dataset contains a large number of images labeled with thousands of different object classes.Pre-trained weights also act as strong initializations, they help to avoid overfitting and they can also be used as a form of transfer learning . The use of pretrained models also assume that the images are preprocessed in a specific way and several cases they may expect the inputs to be scaled from 0 to 1, or –1 to 1 (Géron (2022)).
The use of ResNet for this project was ideal becauseit was used to win the ILSVRC 2015 challenge. Moreover, the layout of its architecture enables it to possess large number of layers which enables it to learn very complex patterns in the data, which may be necessary for accurately classifying satellite images. Another strenght of the resnet is its use of residual connections allowing the model to learn the residual (i.e., the difference between the input and the desired output) instead of the actual output. Figure 7 shows how the residual learning occurs in the ResNet structure.

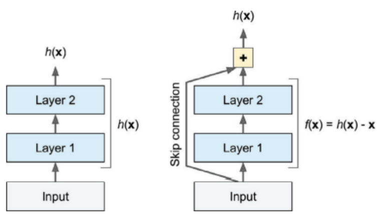


Figure 7: Illustration of the residual learning of ResNet.(Géron (2022))

The application of the ResNet50 on the EuroSAT dataset by (Helber et al. (2019)) had an accuracy of over 98% which is higher than our based accuracy score using SGD optimizer. However, this could be due the use of the dataset containing all 13 spectral bands.

6 Conclusion

Using ResNet50 for satellite image classification could be a powerful and effective deep learning model for this task as ResNet50 can effectively learn and classify features from satellite images.The use of SGD 4 as optimizer and the inclusion of the weights provided an overall accuracy of an accuracy of 96.26% which is acceptable .

It has been demonstrated on various satellite imagery datasets and can learn and classify features at multiple scales. This is important for accurately classifying objects in satellite imagery. Moreover, using ResNet50 improves the accuracy and efficiency of satellite image classification, making it a valuable tool for various applications. It important to point out that other models can obtain similar or better score and it all depends on the dataset and the type of fine-tuning that was carried out for said model.

References

Géron, A. (2022). *Hands-on machine learning with scikit-learn, keras, and tensorflow*. " O'Reilly Media, Inc.".

Helber, P., Bischke, B., Dengel, A., & Borth, D. (2018). Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *Igarss 2018-2018 ieee international geoscience and remote sensing symposium* (pp. 204–207).

Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Mallmann, C., PRADO, D. d. A., & Pereira Filho, W. (2015). Índice de vegetação por diferença normalizada para caracterização da dinâmica florestal no parque estadual quarta colônia, estado do rio grande do sul-brasil. *Revista Brasileira de Geografia Física, Recife*, *8*(5), 1454–1469.

Tassi, A., & Vizzari, M. (2020). Object-oriented lulc classification in google earth engine combining snic, glcm, and machine learning algorithms. *Remote Sensing*, *12*(22), 3776.