Transfer Learning-based Image Classification

Cantore Andrea
University of Oulu
Andrea.Cantore@student.oulu.fi

Moricoli Davide
University of Bologna
davidemoricoli@gmail.com

Manex Sorarrain Agirrezabala
University of the Basque Country
msorarrain001@ikasle.ehu.eus

Abstract

The aim of the project is to try to overcome the challenge of limited data availability. For this reason has been used the method of transfer learning with the goal of improving classification performance. The target dataset is EuroSAT, a popular resource for Earth observation. The models have been pretrained on miniImageNet, a dataset commonly used in few-shot learning tasks. The project begins by processing the miniImageNet dataset with two distinct architectures that have been employed, ResNet18 and a Vision Transformer. The pre-trained models are evaluated and tested on the validation and test sets of miniImageNet. Subsequently, a small amount of images from EuroSAT has been selected to fine-tune the pretrain models and test them. To ensure robustness, fine-tuning is performed multiple times, and average results are reported. The report presents a comparative analysis of different models, including ResNet18 and Vision Transformer, evaluating their performance in the context of small datasets.

1. Introduction

Artificial intelligence is concerned with building systems that simulate intelligent behavior. A deep neural network is a type of machine learning model wielding its prowess in extracting intricate patterns and representations from vast datasets. It focuses on the development of artificial neural networks in order to perform complex tasks such as image recognition, natural language processing, and others. The term "deep" refers to the use of multiple layers (deep architecture) in these neural networks. One of the key components of deep learning is the training. In this phase, the model learns from data and adjusts its parameters based on the input data and the desired output. Obviously, there are both positive aspects to the use of these algorithms and negative aspects. Deep learning excels in the representation of data (feature representation) and it has been demonstrated to outperform some classical machine learning approaches in different tasks. It is characterized by a wide scope of applicability and usually the larger the datasets are, the better the performance is. On the other hand, there are some issues such as the need for large amounts of labeled data and computational resources. One thing that must be avoided and which neural networks are very prone to do is overfitting. In this case, the model learns the training data too well and fails to generalize new unseen data. Training robust models with a small amount of labeled data is one of the main challenges in this sector. This leads us to the following question: What happens when basic deep learning methods fall short, and how can we transcend these limitations? Enter transfer learning—an innovative approach that seeks to leverage knowledge gained from one task to enhance performance on another. In practice, the learning process consists of fine-tuning a pre-trained model on the target task. Transfer learning is a valuable approach in scenarios with limited sample sizes thanks to its capability to use a pre-trained model's early layers as a feature extractor and its capability to converge in a small amount of time since the model has already some knowledge. Overfitting is avoided as the range of data already observed during the initial training phase is wide, but at the same time, the source and the target domains must be related. The subsequent sections of this paper will be about the development of a transfer learning approach to enhance the classification performance on the EuroSAT dataset. The proposed methodology involves pretraining a deep neural network (ResNet18) on the miniImageNet dataset, followed by fine-tuning it on a small subset of EuroSAT images. Has been also implemented also a vision transformer to compare the results. While the initial accuracy of ResNet18 might appear modest, subsequent fine-tuning with a minimal set of images demonstrated its effective generalization to different tasks. The lower accuracy on the challenging miniImageNet dataset compared to

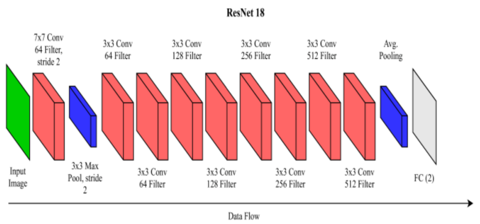

Figure 1. ResNet18 architecture. [1]

fine-tuning underscores the model's adaptability. The results highlight the model's promising capability to generalize and handle diverse problems.

2. Approach

In several domains, obtaining large datasets for the training of deep neural networks is quite difficult. For this reason, we tried to transfer the knowledge learned from a source task to enhance the performance of a different target task. The primary objective of this project is to improve the classification performance in remote sensing applications, specifically focusing on the EuroSAT dataset, which is known for its limited size.

2.1. First step

The first step was to download the miniImageNet dataset in order to pretrain a ResNet18 model on it. The architecture of the ResNet18 can be seen in figure 1.

Then we split each class into train, validation, and test sets and defined a data transformations to resize the images and normalized them. After defining the architecture of the model, we performed hyperparameter search to find out the best parameters for the training. At this point the ResNet18 has been pretrained on the training set of miniImageNet, and then evaluated and tested on the validation and test set. Finally the pretrained model has been saved. Has been decided to compare the results of the ResNet18 model with a Vision Transformer. (Should we said it? We achieve bad results)

2.2. Second step

In the second step of the project, the focus shifted to the EuroSAT dataset. The goal was to fine-tune the previously trained ResNet18 model on a subset of EuroSAT images and evaluate its performance. To proceed with fine-tuning, 100 images were randomly selected from EuroSAT, covering 5 different categories with 20 samples each. From these 100 images, 25 were randomly chosen as the training set, ensuring a diverse representation from the 5 categories.

For the fine tuning process, all the layers of the model have been frozen except the last two convolutional layers. Freezing a layer means that its weights will not be updated in the fine tuning process. The last fully connected layer is changed according to the number of classes (in this case 5), and trained, followed by a softmax layer for the classification.

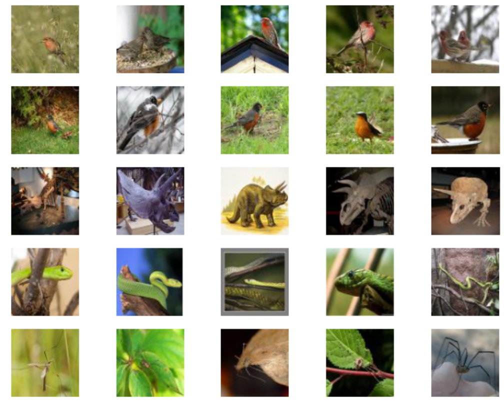

Figure 2. Some images of miniImageNet dataset.

The fine-tuning process involved updating the model's weights based on the EuroSAT data while retaining the knowledge gained during the pretraining on miniImageNet. To enhance robustness and capture variability in performance, the fine-tuning process was repeated several times on different sets of 25 EuroSAT training images. This iterative approach allowed for a more comprehensive evaluation of the model's adaptability to different subsets of EuroSAT.

3. Experiments

3.1. Datasets

The datasets that were used are miniImageNet and EuroSAT(RGB).

MiniImageNet is a widely used dataset in machine learning and image analysis. It is designed for image classification and object recognition, specifically focusing on few-shot learning tasks. The dataset is a smaller version of the larger ImageNet dataset, containing a subset of classes selected from ImageNet. In this case only the subfolder Train was used. This folder is divided into 64 subfolders each containing 600 images 84x84 pixels. In figure 2 some images are represented.

EuroSAT (RGB) is a dataset of satellite images designed for land cover classification. The images are captured by the Sentinel-2 satellites as part of the European Space Agency's Copernicus Earth observation program. The "(RGB)" indicates that the images are in the Red-Green-Blue color representation, common in color images. The dataset includes 13 land cover classes such as forests, agricultural areas, meadows, and cities. It serves as a benchmark for evaluating the performance of image classification algorithms, particularly those dealing with satellite data for applications like envi-

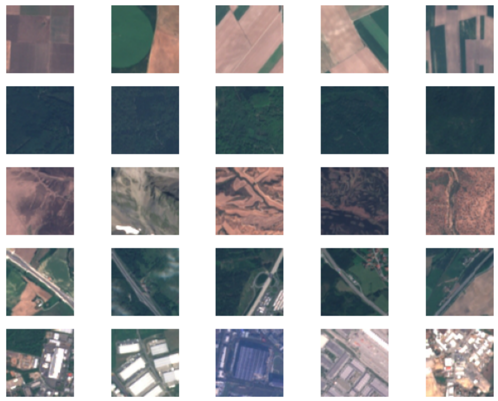

Figure 3. Some images of EuroSAT (RGB) dataset.

ronmental monitoring, urban planning, and precision agriculture. In 3 some images are represented.

3.2. Training Setups

3.2.1 Pretraining

The entire miniImageNet dataset was divided into a training set, test set and validation set. To achieve this each folder of the dataset has been divided properly. The training set consists of 70% of the total images, and the test and validation set correspond to 15% each. The training has been performed utilizing 10 as the number of epochs and the CrossEntropyLoss as loss function. In details, below are shown the hyperparameters that have been ispected with ResNet18.

- learning rates = [0.001, 0.01, 0.1]
- weight decays = [0.0001, 0.001, 0.01]
- optimizer names = ['Adam', 'SGD']

In details, below are shown the hyperparameters that have been ispected with Vision Transform.

- learning rates = [0.001, 0.01]
- weight decays = [0.0001, 0.001]
- optimizer names = ['Adam', 'SGD']

3.2.2 Fine tuning

When fine tuning models with this few data, the hyperparameters we choose are more important due to the lack of the data, and it is easier for the model to overfit.

In this case, we decided to use the Stochastic gradient descent optimizer with momentum 0.9 and weight decay of 0.002, with a learning rate of 0.0005. The models were trained for 100 epochs.

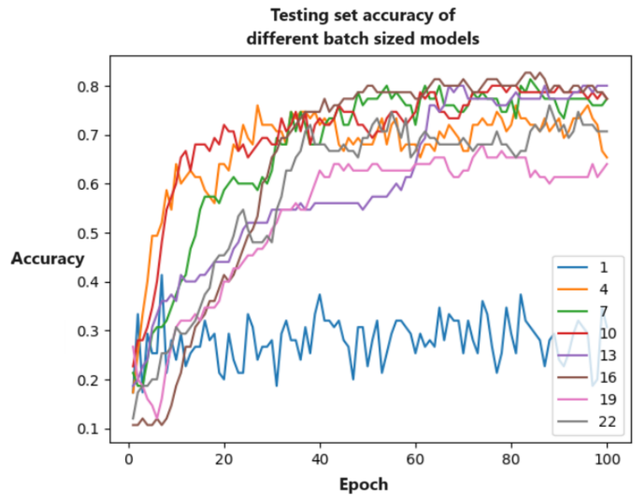

Figure 4. Batch size value exploration.

Optimizer	Learning rate	Weight decay	Validation acc.
Adam	0.001	0.001	39.27%
Adam	0.001	0.01	13.12%
SGD	0.001	0.001	**42.48%**
SGD	0.01	0.001	41.27%
SGD	0.01	0.0001	39.51%
SGD	0.1	0.0001	37.48%

Table 1. ResNet18 results.

As the training set had only 25 images, we noticed that the batch size we used had an important impact in the outcome, so we set up an experiment to came up with the most suitable option. We trained eight different models on the same data, each one with a different batch size, the plot of the training process can be found in figure 4. After this experiment, we decided to use the batch size value 13 because it has a good learning curve with not much downfalls.

As mentioned earlier, we fine tuned the model in 10 randomly selected 100 image sets of the EuroSAT dataset, and then got the average results for better understanding of the performance of the model.

3.3. Results

In Table 1 are shown the results achieved regarding the validation accuracy using ResNet18.

Among the configurations tested, the best validation accuracy was achieved with the combination of SGD optimizer, learning rate of 0.001, and weight decay of 0.001, resulting in an accuracy of 42.48%. This suggests that the choice of optimizer and hyperparameters significantly influences the model's performance, with SGD outperforming

Optimizer	Learning rate	Weight decay	Validation acc.
Adam	0.001	0.0001	**9.46%**
Adam	0.001	0.001	3.35%
Adam	0.01	0.0001	7.45%
Adam	0.01	0.001	1.56%
SGD	0.001	0.0001	2.67%
SGD	0.001	0.001	2.10%
SGD	0.01	0.0001	5.68%
SGD	0.01	0.001	2.22%

Table 2. Vision Transform results.

Accuracy	Avg
68%	**71%**
68%	
73%	
69%	
85%	
65%	
74%	
68%	
69%	
69%	

Table 3. Fine tuning test accuracies

Adam in this particular experiment.

In Table 2 are shown the results achieved regarding the validation accuracy using Vision Transform.

Surprisingly, the results show significantly lower validation accuracies compared to expectations and to ResNet18. The best-performing configuration achieved only a 9.46% accuracy. Unlike ResNet18, Vision Transform prefers Adam as optimizer even though a detailed analysis cannot be done with these results.

As mentioned, we trained 10 model instances on 10 different sets, obtaining the accuracy values in the test set shown in table 3. With an average test accuracy of 0.71%.

To get a better understanding of this model, a training curve of each instance is plotted in figure 5. As the dataset is really small and there is no validation set, the accuracy of the model has been calculated for each epoch in both training and testing sets.

3.4. Analysis

The results presented in the tables showcase the outcomes of experiments conducted with two distinct models, ResNet18 and Vision Transform, employing various combinations of optimizers, learning rates, and weight decay. For ResNet18, the findings indicate a significant impact of the choice of optimizer and hyperparameters on the validation accuracy. Notably, the best accuracy, at 42.48%, was achieved using the SGD optimizer with a learning rate of 0.001 and weight decay of 0.001. Surprisingly, Adam did not perform as well. In contrast, the results for Vision Transform reveal lower validation accuracies across the board, with the best-performing configuration achieving only 9.46%. Unlike ResNet18, Vision Transform appears to prefer the Adam optimizer, though the low accuracies make it challenging to draw definitive conclusions about the optimizer's influence. The surprising difference in how well ResNet18 and Vision Transform performed has us wondering if Vision Transform is the right choice for this task or dataset.

On the other hand, we think that the results obtained in the fine tuning of the ResNet18 model are good, obtaining an average accuracy in the testing set of 71%. We have to remember, that this results have been achieved with a training set of 25 images. As we can see in figure 5, in some cases the model has slightly overfitted but we don't see it as a major issue as in most of the cases it has generalized well.

4. Conclusion

When looking at the ResNet18 results, the obtained accuracy of 42.48% may seem poor and one may think that the model has learnt nothing. But the results obtained in the fine tuning process prove that wrong, just training with 25 images the model has been able to somehow generalize and get acceptable results in a completely different task, showing that indeed it has learnt how to represent an image with all its important attributes in an efficient way.

The low accuracy obtained in the miniImageNet dataset might be due to the difficulty of the task (64 classes) compared for example with the fine tuning task, where there are only 5 classes.

We are happy with the obtained results, as the model has shown capability of generalization to new problems.

References

[1] Muhammed Enes Atik and Z. Duran. *Deep Learning-Based 3D Face Recognition Using Derived Features from Point Cloud*, pages 797–808. 02 2021. 2

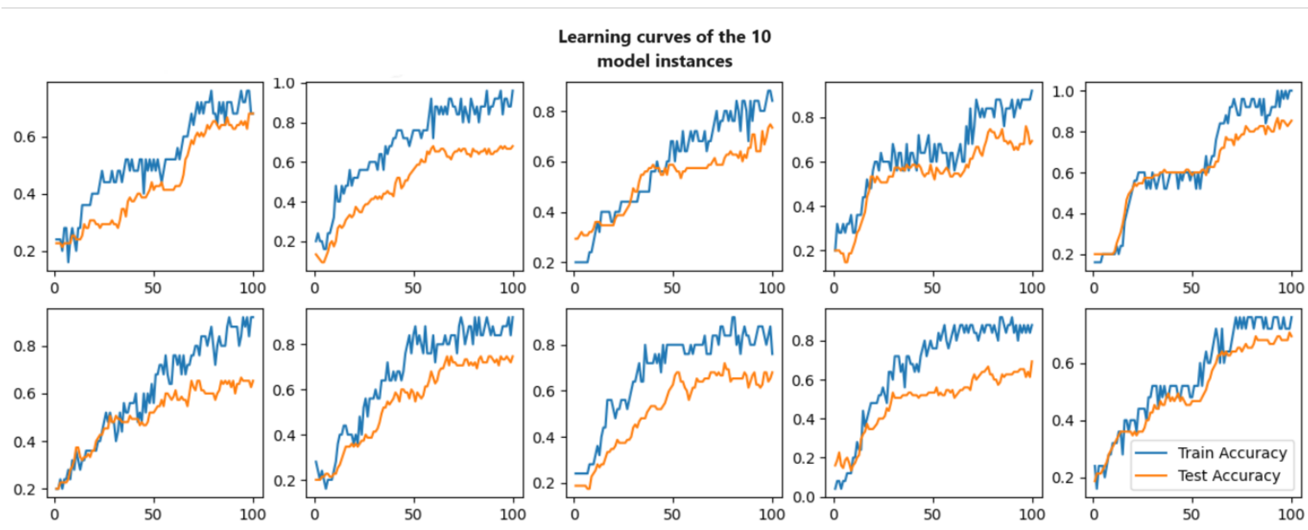

Figure 5. Training curves of the instances.