Deep Learning 2023 Course Project Report

Joona Syrjäkoski
University of Oulu
jsyrjako20@student.oulu.fi

Janne Yrjänäinen
University of Oulu
jyrjanai20@student.oulu.fi

Joonas Ojanen
University of Oulu
jojanen20@student.oulu.fi

GitHub Project link: https://github.com/jsyrjako/Deep-Learning-Project

Abstract

This paper explores the application of deep neural networks using transfer learning in image classification. The goal of this study is to see how to improve the performance in remote sensing applications using small datasets. The primary methods used in this study are the transfer learning technique *and fine-tuning the pre-trained model.*

1. Introduction

This project explores the application of transfer learning in remote sensing image classification, by pretraining the residual learning framework (Resnet) model using miniImageNet and then testing it on the EuroSAT dataset.

In this paper, we first explore the basics of deep learning and transfer learning, then we will go through the approach of the project and experiments and finally, there will be a short conclusion at the end.

Deep learning is a subfield of machine learning (ML), and its goal is to learn data representations using neural networks. In more detail, the purpose is to learn from large amounts of data (text/audio/video/picture) to make models that can predict or identify patterns in text, sounds, and other data. These models are used in tasks like speech recognition, image recognition and natural language processing (NLP).

The significance of deep learning lies in its automatic feature learning, its ability to process large amounts of complex data, and the performanceand applicability of the models.

There are also scenarios where basic deep learning methods are not enough and there are many reasons for this, for example, the deep learning model training could fail, if the data is insufficient, the data amount could be too low or too similar or it can be mislabelled. There can be overfitting, which can be caused by limited data or too complex neural networks. Deep learning model training can also be a very intensive and time-consuming process. And, if the tasks require complex relationships, basic deep learning may not be sufficient because it depends on patterns and algorithms.

In this project, we utilize a technique called transfer learning, it's a technique that utilizes a pre-trained model on a large dataset as a starting point for training a new model. The idea is to transfer the knowledge of an existing model to a new model (Zhuang et al., 2021).

Transfer learning is particularly useful when the new task has a limited amount of training data, as it can prevent overfitting and improve generalization. With transferred learning, the new model can learn faster and more efficiently than the model that was used in learning.

Our model had over 90% accuracy during pre-training, over 80% accuracy during validation, and over 80% accuracy during testing. High training accuracy indicates that the model has learned to identify most observations correctly. Lower evaluation accuracy indicates that it is less accurate at identifying new observations. Our fine-tuned model had an average validation accuracy of over 60%. In general, the model is likely to accurately identify most observations.

2. Approach

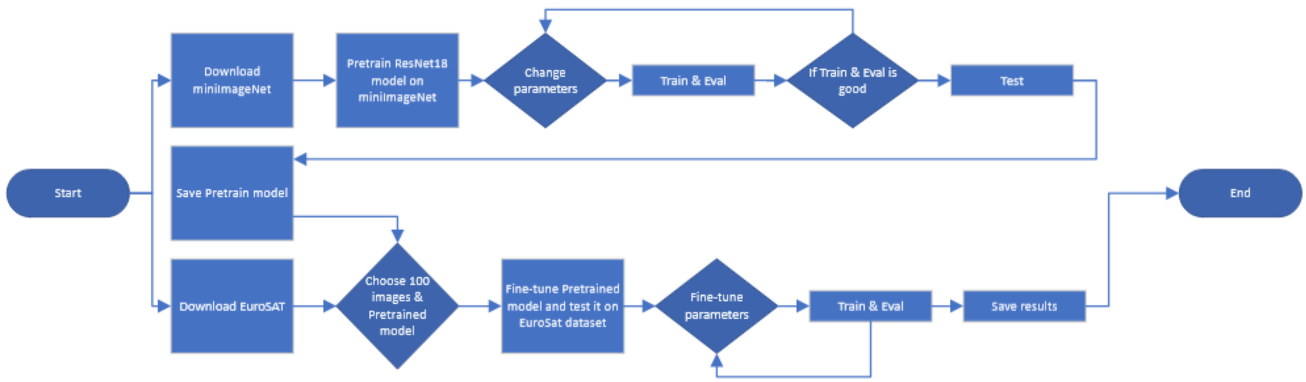

Figure 1: Project hierarchy.

2.1 First step

We are going to use Figure 1 as a baseline for our approach. First, we downloaded and read all the needed materials including miniImageNet, EuroSAT and, ResNet models. We used the ResNet models introduced in Deep Residual Learning for Image Recognition by He et al., 2015 with pre-trained weights. The pre-trained weights from Pytorch (Paszke et al., 2019) closely reproduce the results of the paper by He et al., 2015. Before we could pre-train the ResNet18 model we needed to split the train.tar data, which is from miniImageNet to train, validation and test data sets. For this we used a stratified split with a 0.7/0.3/0.3 split ratio, this part also includes image transformation and dataloader creation. For the loss and optimizer functions, we chose CrossEntropyLoss and Stochastic Gradient Descent or SGD in short (Murphy, 2022), the mathematics for those can be seen in section 2.4. After all this was done, we could start the training and evaluation process. And last we saved the pre-trained model for later use.

2.2 Second step

In the second step, we also use Figure 1 as a baseline. In the second step, we started the process by selecting 100 images from 5 different categories each including 20 samples from the EuroSAT dataset, 25 (5/5/5/5/5) images for training, and 75 (15/15/15/15/15) images for validation. Next, we trained the pre-trained ResNet18 model with the 25 training images and validated the model with the 75 validation images. These steps were done in a loop multiple times so we could fine-tune the model several times with different images to get an average result.

2.3 Third step

In the third step, we compared how different models like ResNet18 and ResNet34 differed and tried to investigate which optimization strategy was best. We also evaluated the CUB dataset so we could see how the pre-trained ResNet18 model handles it.

2.4 Mathematics

Loss:

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{C} e^{x_{ij}}} \qquad (1)$$

and

$$loss_i = -\sum_{j=1}^{C} y_{ij} * \log\left(softmax(x_i)_j\right) \qquad (2)$$

SGD:

$$\theta_{t+1} = \theta_t - \eta_t \nabla L(\theta_t, z_t) = \theta_t - \eta_t g_t \qquad (3)$$

The softmax function (1) helps in predicting the different class probabilities. Softmax function calculates probabilities for each class (Goodfellow et al., 2016). Loss is then calculated with cross-entropy (2). The cross-entropy measures the difference between predicted and actual class (Murphy, 2012). The SGD optimizer function (3) updates the model's parameters. It aims to update the parameters to minimize the loss and thus refine the prediction of classes (Murphy, 2022).

3. Experiments

The model was first trained against miniImageNet (Vinyals et al., 2016), which contains 100 classes from the ILSVRC-12 dataset. ILSVRC-12 is also known as ImageNet (Russakovsky et al., 2015).

We used the EuroSAT dataset created by Helber et al., 2017. The dataset includes 27000 images in total, which are taken by Sentinel-2 satellite.

The model was evaluated against the CUB-200 2011 dataset created at the California Institute of Technology by Welinder et al., 2010. The 2011 dataset consists of 11788 images of birds in 200 different categories.
In our test setup, we first train our pre-model with training data for 10 iterations and then evaluate it with evaluation data. Lastly, we test the pre-model with the test data. After that, we fine-tune our pre-model with the EuroSAT training data, after which we evaluate our model with the the EuroSAT evaluation data. Since EuroSAT dataset is small, the loss value expected to fluctuate while training and validating against the dataset.

Our pre-training model had an accuracy of about 96% and a validation accuracy of about 86%. Testing accuracy was about 85%. Training accuracy is high, indicating that the model has learned to correctly identify most of the observations in the training data. Evaluation accuracy is slightly lower than training accuracy, indicating that the model is not as good at identifying new observations. Average validation accuracy for out fine-tuned model was about 64%.

Using ResNet with more layers (e.g. 34 or 50 layers) did not improve accuracy and training took significantly more time. Changing the model to a VGG model also resulted in poorer accuracy and longer training time. The best accuracy and the most reasonable training time was obtained by using the ResNet18 model.

We evaluated our models against CUB dataset, and the results were not good. The validation accuracy was 0.8% when using ResNet18 model.

4. Conclusion

Overall, the results are good and the model can probably correctly identify most observations. More layers do not mean better accuracy and usually just leads to longer training times. A comparison of the models against different dataset revealed that they do not perform well with different kind of datasets.

4.1. Contribution

Contributor	Percentage
Joona Syrjäkoski	33%
Janne Yrjänäinen	33%
Joonas Ojanen	34%

References

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. The MIT Press.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*.

Helber, P., Bischke, B., Dengel, A., & Borth, D. (2017). *EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification*. https://doi.org/10.1109/JSTARS.2019.2918242

Murphy, K. P. (2012). *Machine Learning : A Probabilistic Perspective*. The MIT Press. https://pc124152.oulu.fi:9443/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=480968&site=ehost-live&scope=site

Murphy, K. P. (2022). *Probabilistic Machine Learning: An introduction*. MIT Press. probml.ai

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Vinyals, O., Blundell, C., Lillicrap, T., kavukcuoglu, koray, & Wierstra, D. (2016). Matching Networks for One Shot Learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 29). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., & Perona, P. (2010). *Caltech-UCSD Birds 200* (Issue CNS-TR-2010-001).

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, *109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555