Advanced Machine Learning

Project work documentation

Hermán Judit
Kovács Kíra Diána
Nguyen Ba Phi
Varga Krisztián

1. Introduction

Machine learning, particularly deep learning, has become a cornerstone in advancing medical imaging and diagnostics. Convolutional neural networks (CNNs) and other deep learning models excel in analyzing medical images like MRI, CT, and X-ray scans by autonomously learning patterns without manual feature extraction. These techniques are widely applied to classify diseases, detect abnormalities, and segment anatomical structures across disciplines such as radiology, pathology, and cardiology. Despite their success, challenges like the need for large annotated datasets, computational demands, and model interpretability remain significant. [1] [2]

Additionally, deep learning aids in multi-modal data integration (e.g., combining imaging and genetic data), personalized medicine, and even generating synthetic medical images to augment datasets. Techniques like transfer learning are also being used to address data scarcity by adapting models trained on large general datasets for specific medical applications. [2] [3]

In the framework of this project we classified brain MRI images into nondemented, very mild demented, mild demented and moderate demented classes. We compared several different convolutional models using transfer-learning.

2. Data

The data we used is a Kaggle dataset [4] of brain MRI images and contains 4 classes: nondemented, very mild demented, mild demented and moderate demented. The dataset contains more than 35000 grayscale images in various sizes and their labels. As it can be seen in the next figure, with human eyes we cannot distinguish the classes based on these MRI pictures.

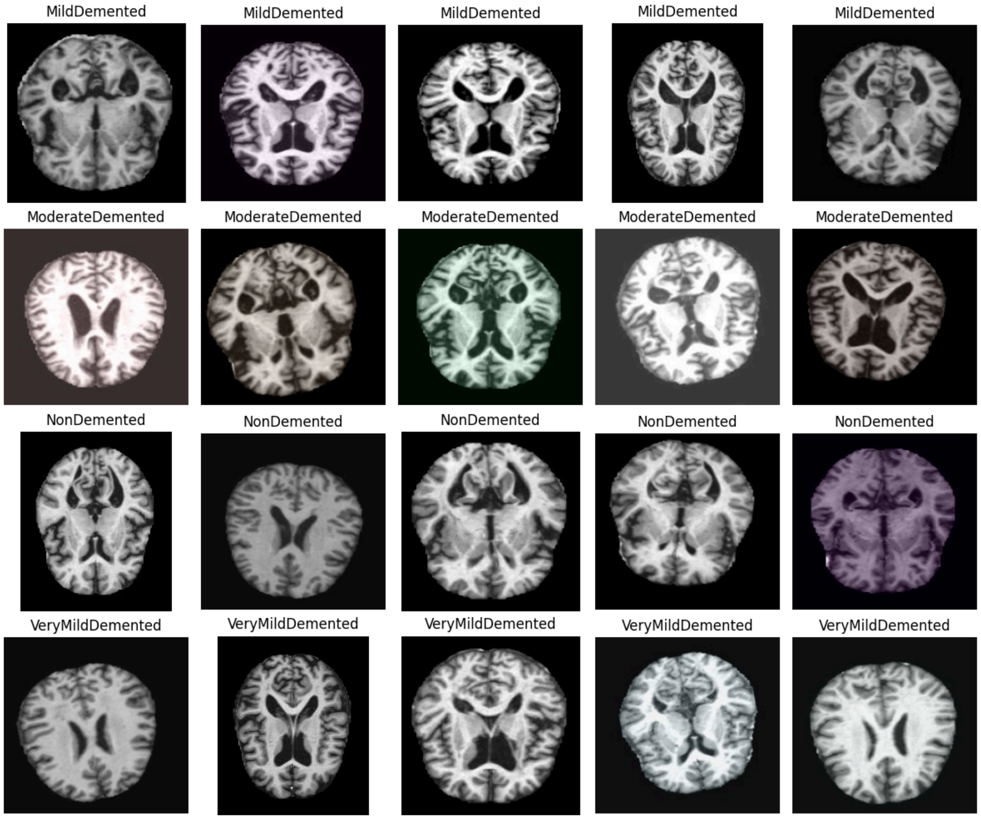


Figure 1.: Sample images from the 4 classes.

Originally, the nondemented and very mild demented classes contained more than twice as many images as the mild demented class, but for the most efficient training we needed nearly equal class sizes. The most proper way of handling imbalanced data would be to oversample the smaller classes with data augmentation, but because of the limited computational capacity, we downsampled the other 3 classes to get as many images as the mild demented class.

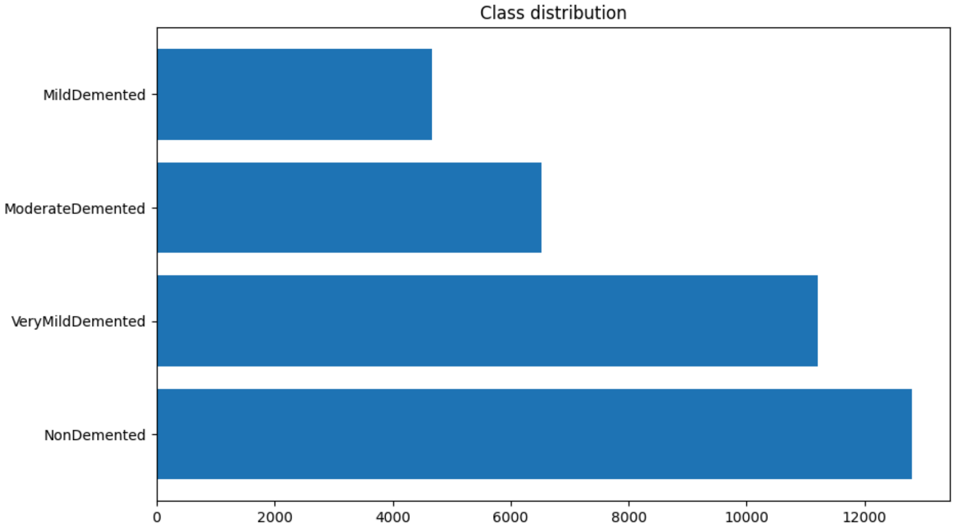


Figure 2: Imbalanced class sizes.

3. Methods of training

Our aim was to try out different deep learning models with several hyperparameters to learn how to use them, compare them, and find out which one is the most appropriate for this problem. We trained our models on the server of Kaggle using GPU, but the GPU time was limited, so we could not use hyperparameter optimization. We did the training in Pytorch and used CrossEntropyLoss.

3.1. Data preparation, hyperparameters

We used a learning rate of 10^{-3}, a batch size of 128 and Adam optimizer. Because we used pre-trained models which were trained on ImageNet, we used 224x224 image size, tripled the channel size because the ImageNet dataset contains colorful images and normalized them.

For most of the training we used CosineAnnealingLr as a learning rate scheduler, because it made the models better by our experience. [5]

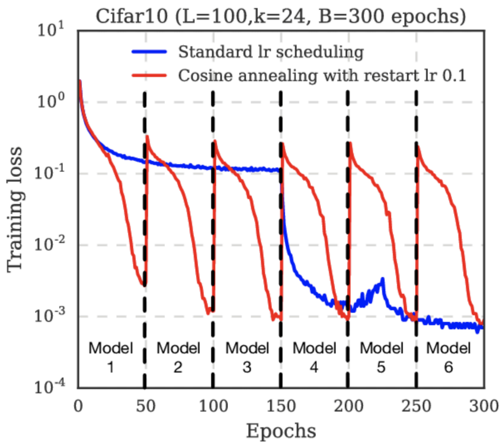


Figure 3: Cosine Annealing Scheduler.

Source: https://paperswithcode.com/method/cosine-annealing

3.2. Models

We tried out different pre-trained convolutional models: ResNet50 [6], VGG16 [7], EfficientNet-B0 [8] and MobileNet-V2 [9]. First, we froze the weights of the models and trained only the last layers through 25 epochs, then fine tuned them through 5 epochs in order to reach higher accuracy. We used early stopping with patience of 2 epochs and saved only the best states of the models.

ResNet reached the validation accuracy of 0.68 with frozen weights and during fine tuning it became higher than 0.9.

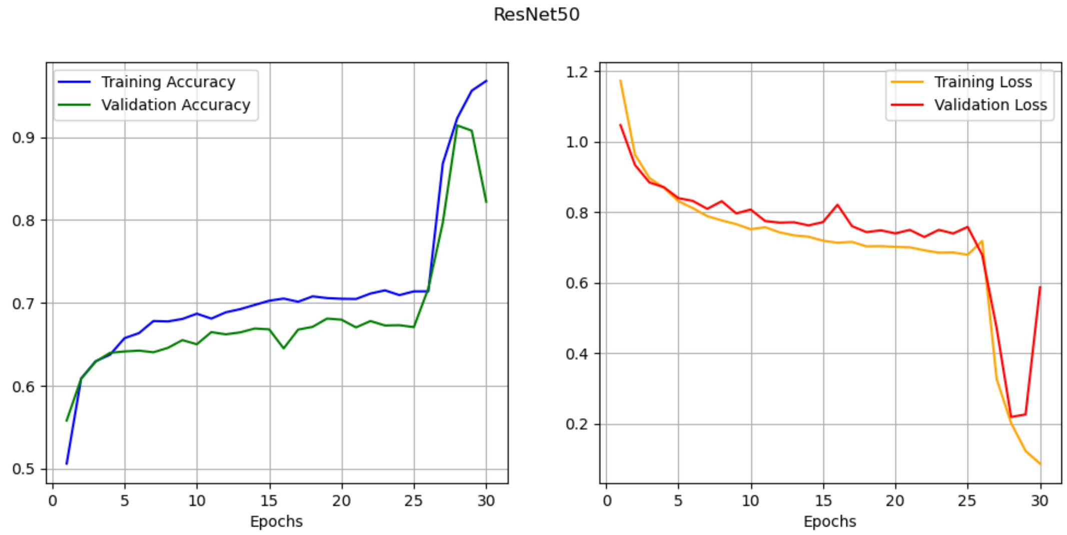


Figure 4: Training process of ResNet50.

VGG16 did not perform well with this number of epochs and got only worse with fine tuning, but since the training of this model was very slow and our computational capacity was limited, we did not increase the number of epochs and optimize the hyperparameters for this model.

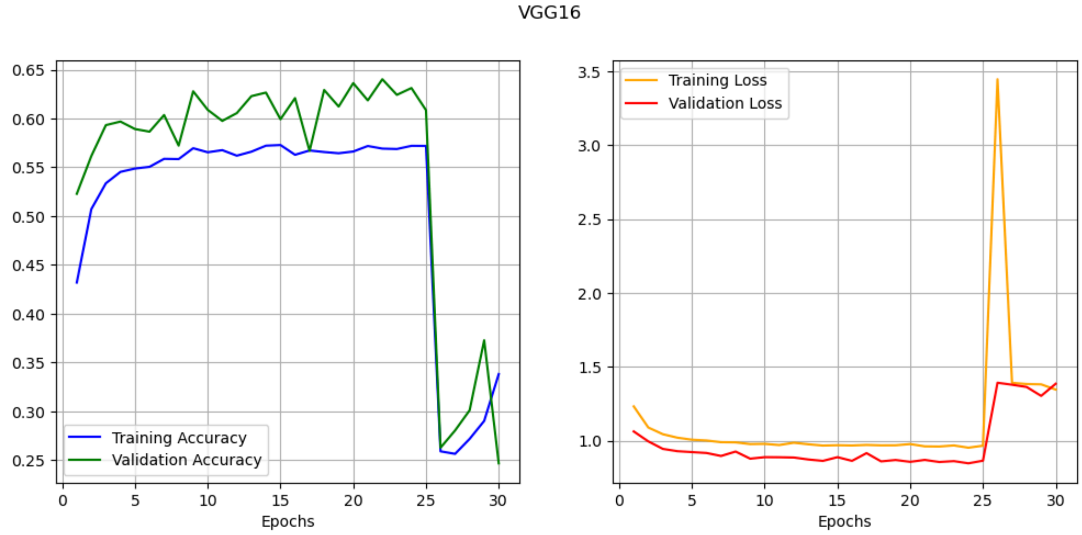


Figure 5: Training process of VGG16.

EfficientNet turned out to be the best model out of these four. It exceeded the validation accuracy of 0.69 during the training with frozen weights and reached 0.96 in the fine tuning phase.

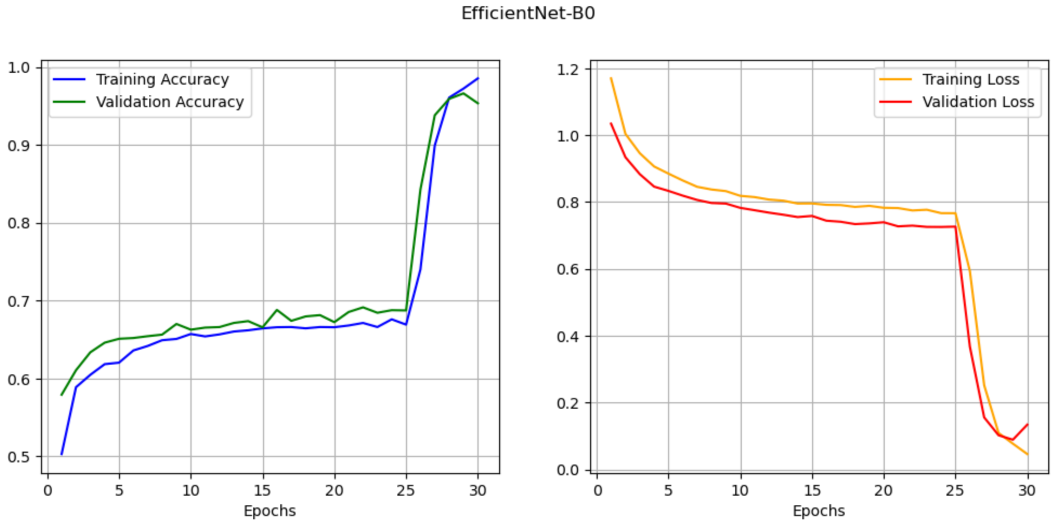
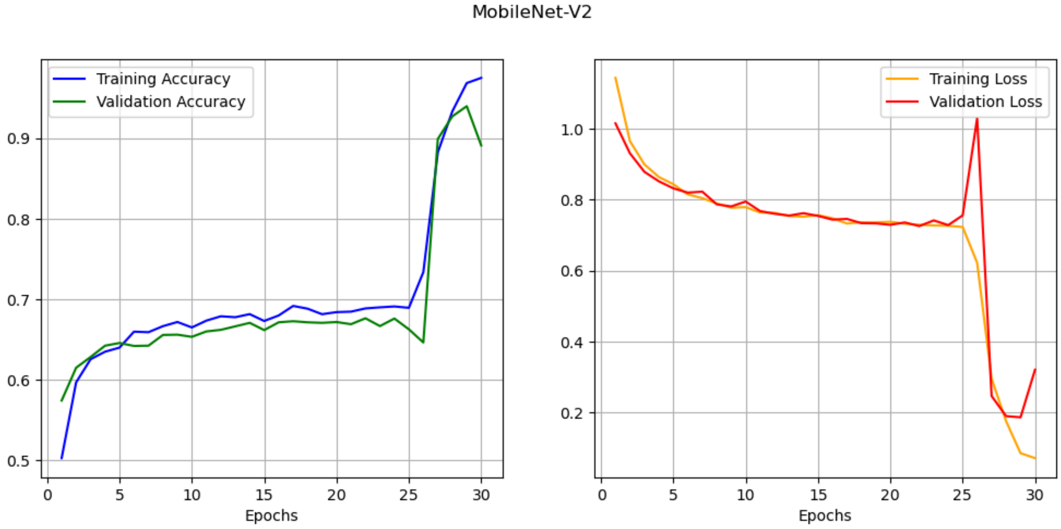


Figure 6: Training process of EfficientNet-B0

We also tried a MobileNet-V2 network to find out how it performs on this data. Without fine tuning it made until 0.67 validation accuracy and after fine tuning it also exceeded 0.9.



.Figure 7: Training process of MobileNet-V2

We can see in the figures that at the end of fine tuning the models started to overfit, but we loaded the states with the best validation loss after the training and evaluated it on the test set.

4. Evaluation on the test set

We evaluated our models on the test set and got the following results: the best performing model was Efficientnet with 95% test accuracy (which is quite good for 4 classes). The second best model was MobileNet with almost 93% test accuracy. Not far behind, the ResNet model almost reached 90% accuracy, but the VGG model performed very poorly on the test set, it has only 38% test accuracy after fine tuning.

We compared not only the top1 accuracies, but checked what proportion of the test data has the true label among the top 2 or 3 classes with the highest probability according to the models. It is visible in the figure that EfficientNet has all true labels among the top2 predicted classes, and the MobileNet and ResNet models' top2 accuracies are also very near to 1. VGG is very bad considering the top 2 and 3 accuracies, probably it would have needed more training epochs before the fine tuning or more optimal hyperparameters, but we did not have the computational capacity to do this.

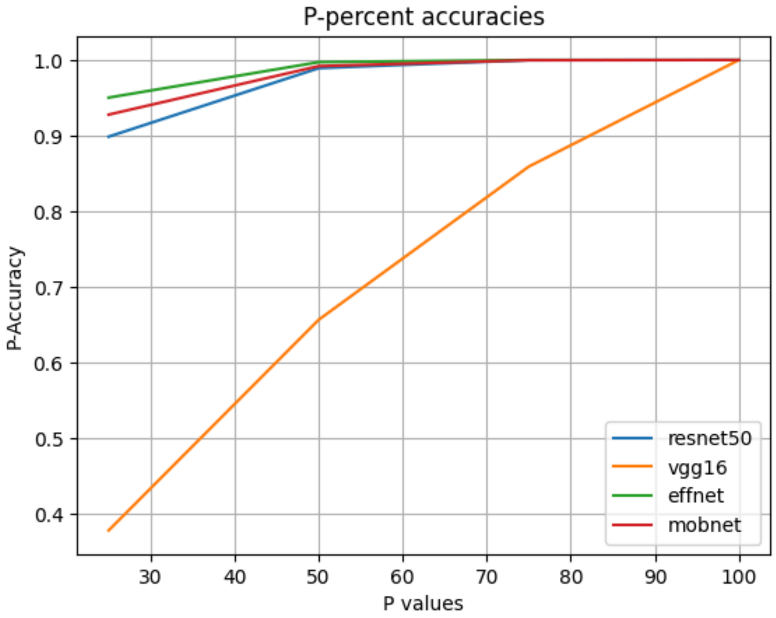


Figure 8: top2 and top3 accuracies of the models

We visualized the confusion matrices of the models. The best model performs well in identifying ModerateDemented and MildDemented cases, as the high values along the diagonal indicate, but in some cases it tends to confuse NonDemented with VeryMildDemented. It is also visible in the figure that it correctly identified all the pictures belonging to the ModerateDemented class and misclassified only a few images of the MildDemented class.

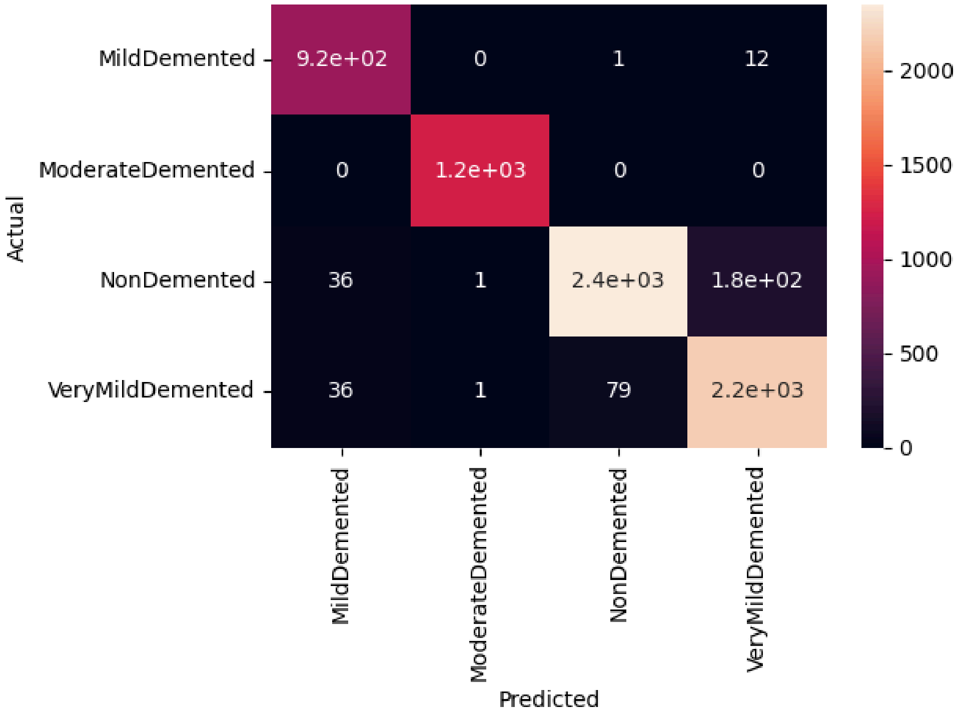


Figure 9: Confusion matrix of EfficientNet on the test data

5. Conclusions

This project demonstrated the application of transfer learning and deep learning models in classifying brain MRI images into four categories: NonDemented, VeryMildDemented, MildDemented, and ModerateDemented. Among the models evaluated, EfficientNet-B0 emerged as the most effective, achieving a test accuracy of 95%. The MobileNet-V2 and ResNet50 models also performed well, achieving accuracies of 93% and 90%, respectively. However, the VGG16 model significantly underperformed, likely due to inadequate training epochs and suboptimal hyperparameters.

Our study highlights the advantages of using pre-trained models, such as EfficientNet-B0, to overcome challenges like limited computational resources and imbalanced datasets. The approach of freezing weights initially and later fine-tuning provided a clear strategy for optimizing performance. Additionally, the use of metrics beyond top-1 accuracy, such as top-2 and top-3 accuracies and confusion matrices, gave a comprehensive view of the models' classification abilities. The analysis also revealed areas for improvement, such as addressing confusion between NonDemented and VeryMildDemented categories.

Maybe instead of doing a 4-class classification, first doing a binary classification into NonDemented and Demented classes, and then trying to distinguish the 3 demented classes can improve the identification of these two classes. Another future direction could be the usage of weight decays and doing a grid search or Bayesian optimization to fine-tune hyperparameters like learning rates and batch sizes. Addressing the issue of imbalanced datasets by employing advanced augmentation techniques or generating synthetic data could enhance model training, but for this more computational capacity is needed.

Usage of LLMs

We used the help of ChatGPT to write the introduction, conclusion and to search related scientific papers.

References

[1] Aimina Ali Eli and Abida Ali. *Deep Learning Applications in Medical Image Analysis: Ad-vancements, Challenges, and Future Directions*. arXiv, 2024.

[2] Li, Mengfang and Jiang, Yuanyuan and Zhang, Yanzhou and Zhu, Haisheng. *Medical image analysis using deep learning algorithms*. Frontiers in Public Health, 2023.

[3] Ker, Justin and Wang, Lipo and Rao, Jai and Lim, Tchoyoson. *Deep Learning Applications in Medical Image Analysis*. IEEE Access, 2018.

[4] https://www.kaggle.com/datasets/arjunbasandrai/medical-scan-classification-dataset

[5] https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.CosineAnnealingLR.html

[6] https://pytorch.org/vision/0.18/models/generated/torchvision.models.resnet50.html

[7] https://pytorch.org/vision/main/models/generated/torchvision.models.vgg16.html

[8] https://pytorch.org/vision/main/models/generated/torchvision.models.efficientnet_b0.html

[9] https://pytorch.org/hub/pytorch_vision_mobilenet_v2/