AIT challenge: Separate waste efficently with deep learning

Introduction

Waste separation based on computer vision (CV) models can help improve work processes. I approached the task at hand by firstly getting familiar with the dataset. In figure 1 below you can see an example image of each class.

Figure 1. Example images of each class

Secondly, I took a look at previous literature. This search was lead by the idea that fine-tuning an existing deep learning model on the dataset might yield good results already. I found a paper by Malik et al. (2022) which achieves good accuracy on waste separation tasks using the EfficientNet-B0 model. An open-source implementation of EfficientNet-B0 by Luke Melas-Kyriazi is available at `https://github.com/lukemelas/EfficientNet-PyTorch`. Thus, the approach was clear: Build a machine learning pipeline to fine-tune the EfficientNet-B0 model on the dataset.

Methodology

I separated the dataset into training and validation with roughly 85-15% split. Since the classes are not balanced, I made sure that training contains around 85% of each image class and validation the remaining 15%. The images are resized to fit the input dimensions of EfficientNet-B0 (224x224) and normalized with values from the ImageNet dataset. Then, the EfficientNet-B0 model (pretrained on the ImageNet dataset) was finetuned on the training data. Training took around one hour.

Results

Figures 2 and 3 show the training progress on training and validation set (loss and accuracy, respectively). Training loss goes down while both accuracies tend to go up. It seems reasonable that by using appropriate hyperparameters, an accuracy of around 90% on the validation set is achievable by this approach.

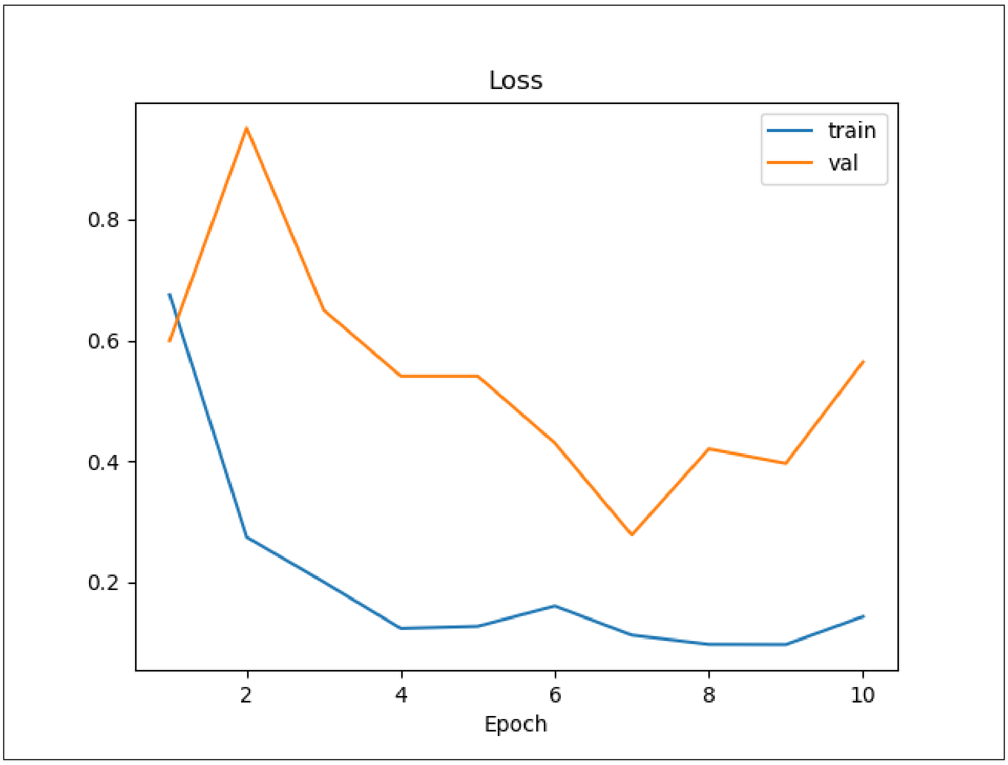

Figure 2. Loss throughout training

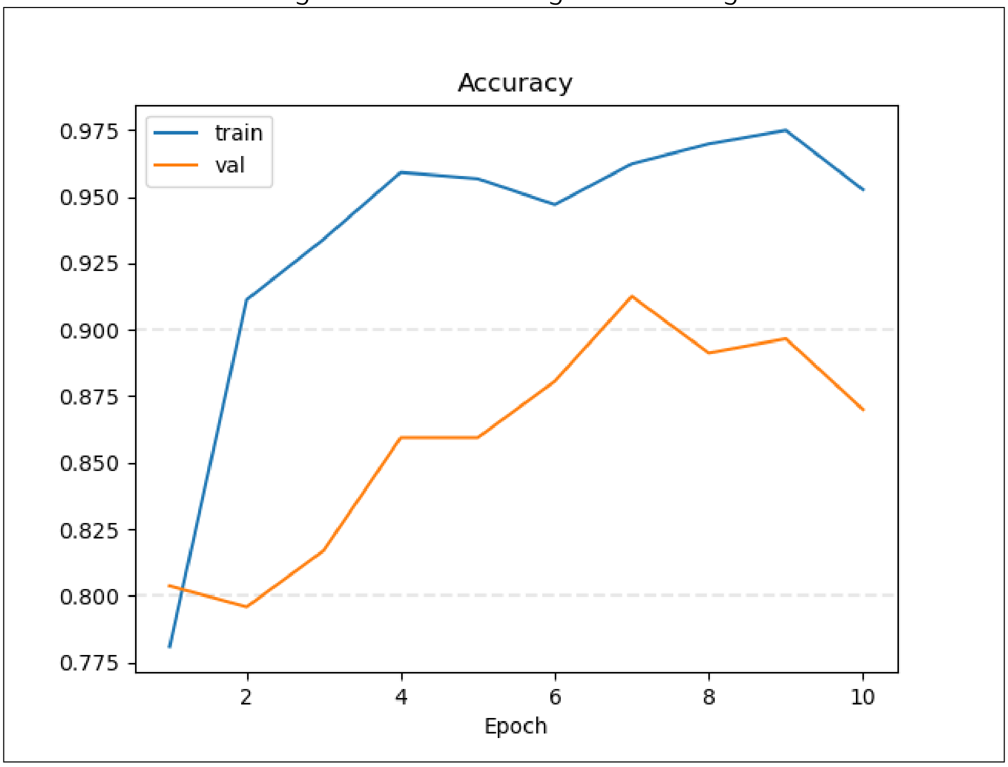

Figure 3. Accuracy throughout training

Conclusion

The solution to the challenge presented here shows that pre-trained models are very powerful and often able to address machine learning tasks with good results. Personally, I see this implementation as a proof of concept. I am content with the results and the model's good performance. However, the main constraint was time. Familiarization with the task, implementation, documentation and writing the report took seven hours plus an additional two hours of training time, as I had to run the model twice. As such, this solution can be expanded upon in a number of ways. Here are some of the next steps I would like to take if I had more time to spend on this task:

- Conduct a thorough literature review
- Expand the dataset. This can be done in multiple ways:
  - Search further image datasets for waste separation
  - Use data augmentation techniques (e.g. rotation, zoom, brightness, contrast)
- Implement a different train/validation/test split (e.g. cross validation)
- Try more models
- Hyperparameter tuning

I would be happy to discuss potential approaches to this problem in a personal interview.