November 2024

AIT challenge: Separate waste efficently with deep learning

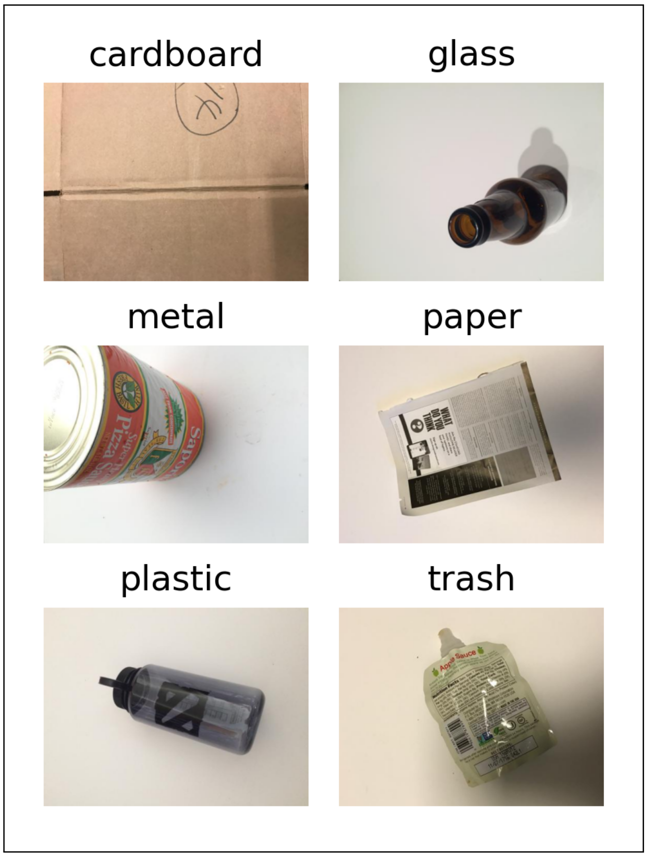
Lukas Till Schawerda

Abstract
Waste separation based on computer vision (CV) models can help improve efficiency and automate tasks. The goal of this challenge is to implement a machine learning model to tackle this problem. A quick look at the literature lead to the decision of fine-tuning the EfficientNet-B0 model to solve this issue. A machine learning pipeline was built and achieved around 90% accuracy on the validation set.

Key words: Computer vision – Waste separation – EfficientNet

1 Introduction

The first step in addressing this challenge was to get familiar with the dataset. To this end, example images of each of the six classes (cardboard, glass, metal, paper, plastic, trash) were visualized:



The second step was to develop a plan for classifying these images into their respective classes. Building convolutional neural networks (CNNs) with limited data and time can be a difficult task. As such, the decision was made to make use of the power of pretrained machine learning models and their existing capabilities. Pre-trained models are often able to collect salient features from images. From previous experience, it seemed reasonable that fine-tuning an existing model on the task at hand might already yielt good results. With this in mind, a cursory literature review was undertaken. A paper by Malik et al. (2022) claimed high accuracy on waste separation by using the EfficientNet-B0 model proposed by Tan & Le (2020). An open-source implementation by Luke Melas-Kyriazi is available at `https://github.com/lukemelas/EfficientNet-PyTorch`

2 Methodology

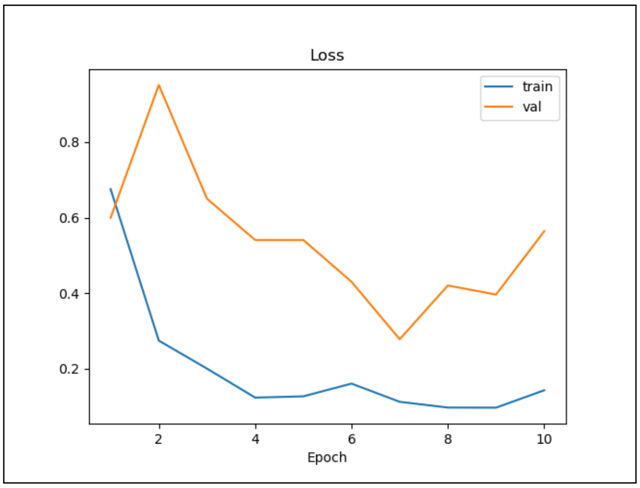
EfficientNet is a group of model architectures that combine different width, depth and resolution scaling methods. EfficientNet-B0 is the smallest of these models. The version used in this task was pre-trained on the ImageNet dataset.

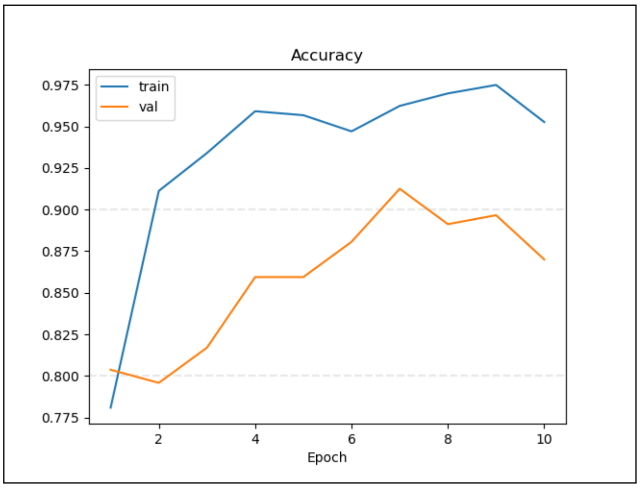
The dataset was split into training and validation with roughly 85-15 split. Since the classes are not balanced, it was made sure that training contains around 85% of each image class and validation the remaining 15%. Thus, the training set contains 2150 observations (343 cardboard, 426 glass, 349 metal, 505 paper, 410 plastic, 117 trash) and the validation set contains 377 observations (60 cardboard, 75 glass, 61 metal, 89 paper, 72 plastic, 20 trash). Each image has dimensions 384x512.

The images are resized to fit the input dimensions of EfficientNet-B0 (224x224) and normalized with values from the ImageNet dataset. Training images were augmented by applying random horizontal flipping. Then, the EfficientNet-B0 model (pretrained on the ImageNet dataset) was finetuned on the training data. Training took around one hour. The code used in this challenge can be found at `https://github.com/schawerda/AIT-challenge`

3 Results

The results show that loss goes down while both training and validation accuracy tend to go up (see figures below). It seems reasonable that by using appropriate hyperparameters, an accuracy of around 90% on the validation set is achievable with this approach. The highest validation accuracy in this specific training run was 0.9125, which is already satisfactory. Further steps which could be taken to improve these results are discussed in the conclusion.





4 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 eight 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.

References

Malik M., Sharma S., Uddin M., Chen C.-L., Wu C.-M., Soni P., Chaudhary S., 2022, Sustainability, 14

Tan M., Le Q. V., 2020, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (arXiv:1905.11946), https://arxiv.org/abs/1905.11946