Deep learning homework project: Waste classification

Anna Gergály, Péter Mészáros

May 15, 2022

1 Introduction

Lot of waste is produced all around the globe each year and this amount is increasing. Waste management on this scale poses a lot of challenges and risks, both economically and environmentally. One of the best solutions we know at the moment besides trying to reduce the amount of waste created is to recycle. This helps recover resources and decreases chance of pollution and mitigates energy usage.

Despite it's obvious benefits it is not practiced to a large enough extent. The reasons for this mainly stem from the same issue: the need to separate the different materials. At home recycling often fails due to the cumbersome nature of the activity and in-facility waste separation is a very difficult task: manual classification is almost impossible due to sheer quantity and automatic waste sorting usually only works for certain materials in certain conditions.

Our goal in this project was to create and review different solutions for a deep learning system that could provide automatic waste classification in such an environment.

2 Previous solutions

In [1] a complex hardware-software solution with a multilayer hybrid method was proposed to classify waste into recyclable and non-recyclable classes and they achieved over 90% mean accuracy. They cite previous solutions made with R-CNN, CNN and SVM classifiers with accuracy rates ranging from 23% to 77%. They trained the model with images of waste found in urban public areas, in a total of 22 classes.

The CNN they used in [1] was a modified version of AlexNet [2] that consisted of 5 convolution layers, 3 max-pooling layers and 3 dense layers at the end. The input image size was 240×240, the convolution kernel sizes were 11×11 for the first, 5×5 for the second and 3×3 for the last 3 convolutional layers. The first two dense layers had 512 neurons. The last dense layer had 22 neurons to match the number of classes. After this an additional layer was added during training with 1 neuron to classify into recyclable and non-recyclable.

They achieved accuracy and recall over 90% with only the CNN, but the metrics were even better when they added extra information by other sensors (e. g. weight).

In [3] a pre-trained ResNet-50 [4] network was used for recognition with a Support Vector Machine (SVM) classifier. The dataset is from [5] which turned out to be one of the datasets we used from Kaggle. (The dataset from Kaggle is just a subset of the original. [6]) They achieved 87% accuracy after 12 epochs trained with SGDM optimizer.

In [7] a pre-trained Inception-v3 was further trained to classify waste. The initial training was done on the ImageNet [8] dataset. The training and testing was done using a Raspberry Pi 3B, exceptional results were achieved with 92.5% with a similar training dataset to the previous one.

3 Dataset

We assembled our dataset from 3 online sources (2 from Kaggle and 1 from Github) both to have a larger pool of images and a more varied one. There are pictures that contain the discarded item in front of plain backgrounds and ones that show them in a more organic way: dirty and/or muddy on a floor.

We settled on having eight categories within the datasets. These (glass, paper, cardboard, trash, metal, plastic, e-waste, medical) were selected mainly on two grounds: the available data and practical considerations. Most of these are well-known, already used recycling categories (glass, paper, cardboard, metal, plastic) and we have a multiple classes for non-recyclables, too. The main one of these is trash, but having extra categories for e-waste and medical waste makes sense as these are more dangerous environmentally and public health-wise. To make our datasets conform to these categorical standards we rearranged some of the pre-labeled data: e.g. merged different colored glasses and deleted categories we deemed unnecessary.

The datasets together hold about 23,000 images which we decided to split in 14:3:3 ratio into training, validation and test sets. (Number of images per class on figure 2.) On the training set we decided to employ data augmentation: using the keras ImageDatasetGenerator we added zooming and shifting as well as both vertical and horizontal flips and 90° rotation to account for the fact that these items could be seen in any orientation in a waste management scenario.

Figure 1: Example images from the dataset.(From left to right: metal, paper, plastic, trash, cardboard, e-waste, glass, medical.)

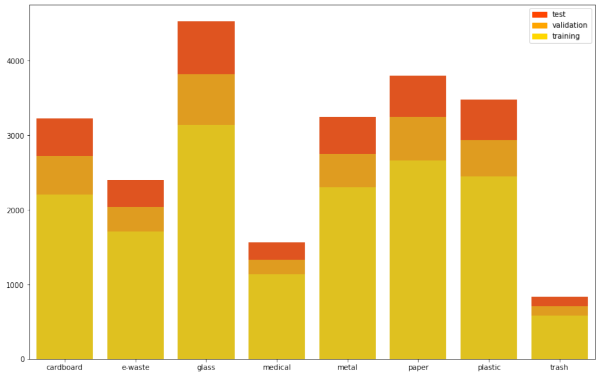

Figure 2: The size of the classes as divided into training, validation and test sets.

4 Proposed method

Our approach was to try out multiple pre-trained image classification CNNs by using transfer learning to prepare them for our specific task. The ones we use are EfficientNetB5 and MobileNetV2, both pre-trained on the imagenet dataset. To add a bit more interest and challenge to the task we also decided to try our hands at building our own CNN from scratch.

4.1 EfficientNetB5

We chose the B5 variant of the EfficientNet because of it's moderate size and fairly strong results. We wanted something that would be feasible to train given our limited GPU-resources, but had a high-chance of yielding acceptable results. We removed the top-layer and substituted it with a few fit for our own needs: global average pooling, dropout and a dense layer with 8-nodes and a softmax activation to serve as an output layer in the 8-class classification problem.

We trained this model, with only transfer learning using categorical crossentropy as our loss and categorical accuracy as our main metric, we used early stopping to stop the model from overfitting. The chosen batch size was 32, and 400 steps per epoch. We achieved a quite good result with this method (~87% categorical accuracy) and we decided to go on with fine-tuning this model.

This however, was not possible, due to our technical limitations: the Google Colab environment could not support training this large of a model. This was one of the reasons we decided on a MobileNet for the third model: we wanted a transfer learning solution where we could also employ fine-tuning.

4.2 Our own model

We built a CNN model comprising of Keras's Conv2D, MaxPool2D, AveragePooling2D, Dropout, Dense, Input and Flatten layers.

The model consisted of 4 convolutional blocks, each of which had 2 convolutional layers a max-pooling/average-pooling layer and a dropout layer. For the initial approach we stacked 2 convolutional layers with 5×5 kernel size and 10 filters and a max-pooling layer then a dropout. This section is repeated with 3×3 kernel size. This is repeated twice, but with only 5 filters, and instead of the max-pooling layer an average-pooling is used. The last layer is flattened and 3 dense layers are added with 1000 neurons and relu activation. A last dropout is added and a last dense layer with 8 neurons and softmax activation. The dropouts had a 0.4 dropout rate. This model didn't get us far, a high accuracy couldn't be achieved.

For the second approach we tweaked the model a bit, we decided to try adding skip connections to the model. The last dense layers were also reduced to have 512 neurons instead of 1000. Before the last convolutional block a skip connection was added from the average-pooling- layer. Before the last 3 dense layers a concatenation layer is added which concatenates the flattened output from the third block's average-pooling layer and the fourth block's average-pooling layer. With this approach, a 39% accuracy was achieved.

The last appoach was a slight modification of the second one. The third block's average-pooling layer's parameters were modified. The pool size became 3×3 with a stride of 3. On top of this an additional branch was added consisting of 2 convolutional blocks. The first block has 2 convolutional layers with 5×5 kernel size, 20 filters and 2 strides, after this a maxpool and a dropout layer is also there. (The strides were added to reduce the number of nodes and parameters.) The second block has 2 convolutional layers as well with a kernel size of 3×3 and with 10 and 5 filters respectively. A max-pooling layer is added with a pool size of 3 and strides 3 of 3. The output of the maxpool is flattened and concatenated to the output of the other branches flattened layers.

4.3 MobileNetV2

We wanted to try out a smaller model, both to improve inference time and to be able to fine-tune the model. With about 3000000 parameters MobileNetV2 seemed a good fit for the task. We prepared this model similar to the first model: removed the top-layer and substituted it with a few fit for our own needs: global average pooling, dropout and a dense layer with 8-nodes and a softmax activation to serve as an output layer in the 8-class classification problem.

When training this model we used the Adam optimizer and used categorical crossentropy as loss and categorical accuracy as our main metric. Batch size and steps per epoch were 32 and 400 respectively. This yielded about ~82.5% categorical accuracy.

We also tried to fine-tune the model with various very small learning rates, ranging from 0.01 to 0.00001 but these have all produced worse results than the previous one, so we decided to keep the original imagenet weights.

5 Evaluation method

For evaluating the trained models, we used a few metrics, these were calculated using sklearn. The calculated metrics are accuracy, balanced accuracy, precision (weighted), recall (weighted), f1 score (weighted) and top2 and top3 accuracy. Of course the top2 and top3 accuracies are less important because of the small number of classes.

6 Results

The best results were achieved with the EfficientNetB5, the accuracy on the test dataset is 89.9% as can be seen on figure 3. The accuracy achieved with MobileNetV2 was 83.5% (figure 6). On our own model the accuracy was about 37-39% (figure 4, 5). The confusion matrices are on figure 7, 8, 9, 10.

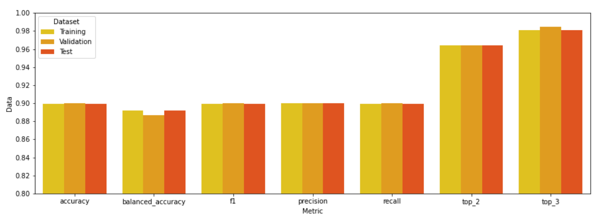

Figure 3: The evaluated metrics on the first model.

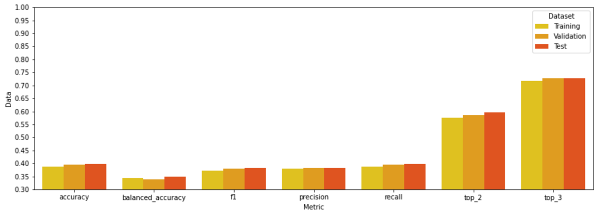

Figure 4: The evaluated metrics on the second model, with two branches.

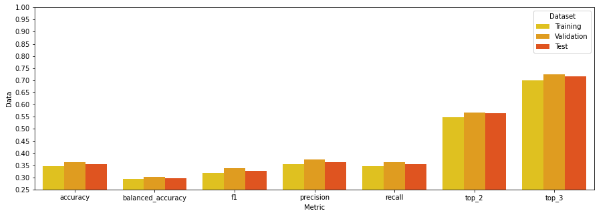

Figure 5: The evaluated metrics on the second model, with three branches.

7 Discussion

The results with the best models are comparable to the results which we've seen and covered in section 2. Our own model had the worst results, but they are still better than random chance. The main reason could be the small number of images compared to the ImageNet dataset which the other models were pre-trained on. The model complexity and size can also be a hurdle, but as we've seen in [1], great results can be achieved which smaller (in terms of layer number) models. The best results were achieved with the EfficientNetB5 which is not surprising because it's the largest model out of the three with about 30 million parameters. With the MobileNetV2 the accuracy was worse, but that model has about 3 million parameters and the inference time on a GPU is more than 6 times better (according to the Keras site).

Overall, we found this method for waste classification quite effective and we believe that with further research into this subject involving segmentation and not just classification it could be used in a modern automatic waste sorting system.

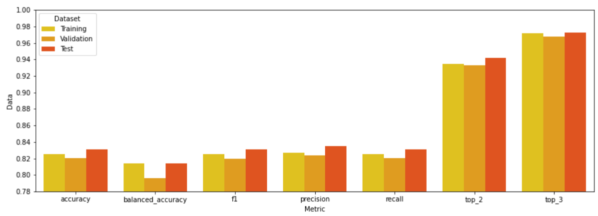

Figure 6: The evaluated metrics on the third model.

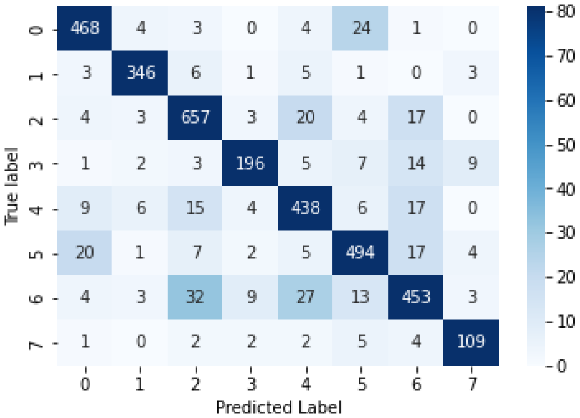

Figure 7: Confusion matrix of the first model.

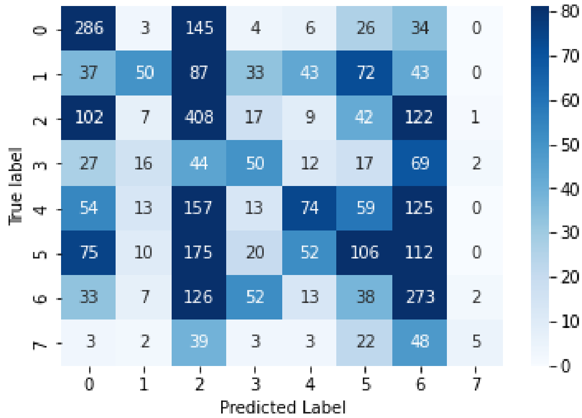

Figure 9: Confusion matrix of the second model with three branches.

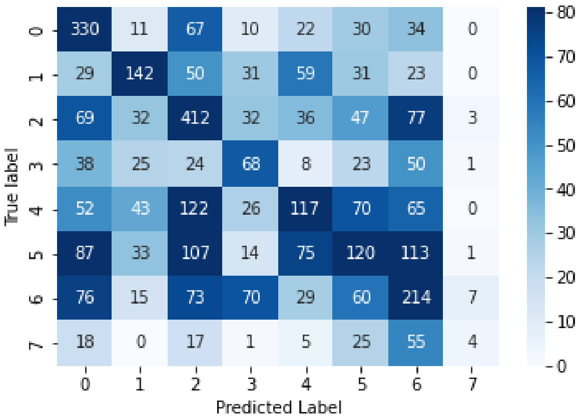

Figure 8: Confusion matrix of the second model with two branches.

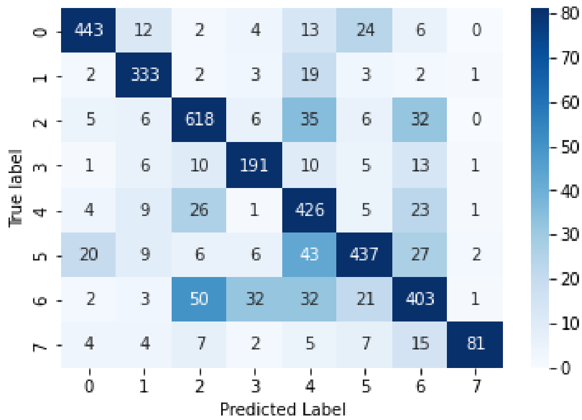

Figure 10: Confusion matrix of the third model.

References

[1] Yinghao Chu, Chen Huang, Xiaodan Xie, Bohai Tan, Shyam Kamal, and Xiaogang Xiong. Multilayer hybrid deep-learning method for waste classification and recycling. *Computational Intelligence and Neuroscience*, 2018:5060857, Nov 2018.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[3] Olugboja Adedeji and Zenghui Wang. Intelligent waste classification system using deep learning convolutional neural network. *Procedia Manufacturing*, 35:607–612, 2019. The 2nd International Conference on Sustainable Materials Processing and Manufacturing, SMPM 2019, 8-10 March 2019, Sun City, South Africa.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[5] Mindy Yang and Gary Thung. Classification of trash for recyclability status. *CS229 project report*, 2016:3, 2016.

[6] Cchangcs. Garbage classification, Nov 2018. Accessed on 05.14.2022.

[7] Fatin Amanina Azis, Hazwani Suhaimi, and Emeroylariffion Abas. Waste classification using convolutional neural network. In *Proceedings of the 2020 2nd International Conference on Information Technology and Computer Communications*, ITCC 2020, page 9–13, New York, NY, USA, 2020. Association for Computing Machinery.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.