Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für
Informatik und
Wirtschaftsinformatik

Waste classification - Boost performance with self-taken images

Project work

by

Ilyesse Hettenbach & Gabriel Schurr

Computer Science Bachelor

Mentor: Prof. Dr. rer. nat. Baier

Summer term 2022

Contents

Abstract - In order to create accurate, image-based classification models to solve important problems such as waste classification, it requires useful, adapted data. This proposal examines the influence of self-taken images compared to those from freely available sources. It creates a waste classification system based on household waste images as a contribution to awareness, education and support in successful waste separation, which preserves and protects living beings, resources and the environment. Furthermore, it works out approaches, concepts and solutions to successfully train image-based classification models.

1 Introduction

The responsibility for our fellow human beings, our planet, following generations, other species and ourselves rises. Not only because we are facing the climate change with all its consequences, especially due to the inequality in technological progress, wealth and privileges, we are more than ever before responsible to take care. Inadequate dumping takes place and with that various of horrific consequences [32, 16, 23]. To face this problem, many people, institutes, regions and countries are developing waste separation methods. "Waste separation is the process by which waste is separated into different elements" [33]. The classification of waste varies depending on the region, values, norms, laws, level of civilization and regulation of a society. An important prerequisite for a successful waste separation process is the existence of appropriate recycling and upcycling facilities. A functioning waste system is essential for the planet and thus for the environment with its inhabitants. It contributes to a cleaner and more aesthetic environment, where the risk from poisoning and diseases of species due to contamination is lower. This increases the standard of living and well-being. Consistent reuse of products and raw materials conserves our resources and protects our climate [24]. A major problem here is the unknowledge of citizens. According to [13], the majority of the content of residual waste in Germany does not actually belong in it. As a result, important resources are usually burned instead of entering the recycling loop [25]. Supporting citizens with waste classification procedures helps to maximize the proportion of reusable resources and serves to educate. Therefore conserve the consumption of finite resources and protect the environment.

2 Related Work

Other works mostly use the data from the Internet and do not use their own data.

Gary Thung and Mingxiang Yang [31] collected their data themselves and made it public afterwards. This dataset is known as TrashNet [15]. They divided the images into six different classes (plastic, metal, cardboard, paper, glass and general trash). The work was done in 2016. They compared a Support Vector Machine with a convolutional neural network (ConvNet) similar to AlexNet. Also, they did not work with transfer learning. They achieved a significantly better accuracy with the SVM (63%) than with ConvNet (22%).

The reason they give is that their hyperparameters may not work well or the architecture is too complex and simple, respectively.

Automatic Image-Based Waste Classification [27] uses the images of TrashNet. For the classification they compare different ConvNet architectures (VGG-16, VGG-19, Inception, ResNet, Inception-Resnet) and use the classes of TrashNet. With an accuracy of 88.66%, they achieve the best results with the ResNet architecture. However, the Inception-ResNet and Inception models achieve a very similar average accuracy.

Also "OscarNet: Using Transfer Learning to Classify Disposable Waste" [20] uses TrashNet as a data source. However, they added one to TrashNet's existing classes: non waste. This is an interesting approach when it comes to classifying whether a product might not belong in the trash. For this they used PASCAL VOC 2012 and Flowers dataset by Visual Geometry Group at the University of Oxford. They achieved an accuracy of 88.42% on the validation data with the VGG19 network after training 163 epochs.

3 Problem and goal

The goal is to train a deep neural network to classify waste images to use it on a website. A person should be able to upload images of waste and therefore receive a evaluation by the neural network in the form of a waste class. The special feature of this project is that the forecasts are based on the German waste system. This tool reduces uncertainty and provides information about waste separation regulations and laws in Germany. A common example of ignorance is baking paper. It seems logical to dispose of it in the blue garbage can[1], but due to its coating, baking paper belongs in the residual waste.

The technical goal of this project work is to find out whether we can improve the results of our predictions if we use self-taken images to extend existing datasets from the internet. So to achieve these goals, different models need to be trained, evaluated and compared. This procedure allows to identify the best model and use it in order to achieve the most accurate predictions on the website. Also, we shouldn't have to wait too long for a result when we upload an image, so the processing time also matters for later comparison.

[1] A common description for the bin for products such as paper or cartons.

4 Website

Our website [4] developed for this project offers a wide range of functions. Essentially, it was created for the application of image recognition. The results are presented as probabilities to the user in order to not only receive a simple, direct output, but a probability distribution, if the results from the neural network are not unique. The user should also have the option of analyzing multiple images at once and in this case the user only receives unique results[2] because a probability distribution for each individual image would be too cluttered and can be confusing. During the development of the project the website served another purpose. It was possible for the users to support us with their images in the form of uploading them to our data storage. All of these functions were and are available on mobile devices too.

5 Data processing

Data is one of the most important components in machine learning and therefore data acquisition, annotation and processing took a lot of time. On the one hand there are a bunch of datasets from publicly available sources, which anyone can receive free of charge and in large quantity, on the other hand there were also self-taken images. These two types of data form the basis for this project. A high diversity in the data is beneficial to receive good results in later predictions. In addition, they have to be sufficiently adapted to our target group, and thus, it must be clarified which target group the project is focused on. For this purpose, it is crucial to take enough time obtaining and analyzing the data in detail. Since this project is in our personal interest and is provided by our family and friends for testing, the images will in almost all cases contain garbage from Germany. There is, of course, waste, such as organic waste, that is not significantly different from location-independent. However, packaging waste in particular varies from country to country. It must also be considered that waste systems in other countries can differ greatly. Of course, this has a big influence on the labeling process of the images. In order to make predictions on our waste separation system, we created five classes. These are based on the local system of Karlsruhe, Germany [19], but are generally mostly handled in the whole country like the following:

- Glass: There are separate containers for glass and the aluminum lids can be disposed of as well.
- Organic: This is what belongs in the organic waste garbage can. This includes all vegetable kitchen waste as well as garden waste and products that are compostable.
- Paper: Cardboard and paper go into the waste paper container (blue garbage can).

[2] The class with highest probability

- Residual waste: This garbage is, as the name suggests, a trash for non-reusable material and goes into the black garbage can. It includes e.g. old clothes[3], diapers, tissues, baking paper, animal or medical products.
- Recyclable material: This includes recyclable and reusable waste such as plastic packaging, tetra packs, food cans and aluminum.

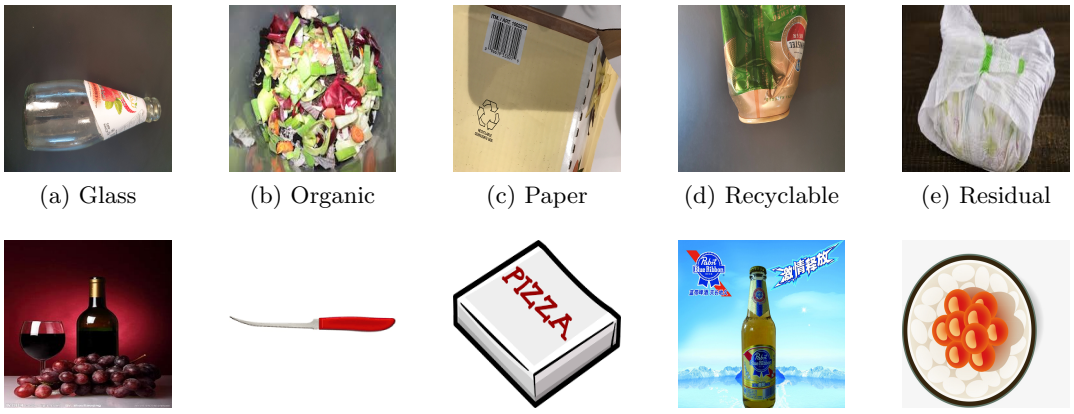
One difficulty was the existence of *residual waste* in Germany because many countries do not have this sort of waste separation and with that this sort of garbage in particular. Citizens in France, to give an example, dispose their diapers in the same trash can in which they dispose food (General Household waste [9]).

5.1 Images from the internet

We were able to get the largest amount of data on Kaggle [28]. With 22,500 RGB images the dataset, provided by Kaggle, offered the most. However, the images are divided into only two different classes (recyclable and organic), so these had to be manually annotated. Furthermore, we used the well-known dataset *TrashNet* [15]. It provides round about 2,500 images and in addition, these have great value because these are from household waste and thus adapt well to our target group. Finally, we used *Google Image Search* to get images.

After collecting the initial amount was 30,000 images. This number became a lot smaller because as shown in Figure 1 there were many images that we had to sort out. Among these images were for example (f)-(j). Furthermore there were several images showing large piles of waste from a garbage site. Unfortunately, these are unusable images for this project too because our recognition is supposed to work on images taken in households containing individual packages or products. In opposite, images (a)-(e) were suitable images for our problem and can be found in the final dataset. After sorting out images manually the individual classes had 18,749 images, which we exclusively obtained from the internet. It can be seen in the graph Figure 2 that the data is relatively well distributed across the classes except for recyclable material, which is still lacking in amount compared to the other classes. Residual waste is already representing many images. However, it mostly contains textiles such as shoes or clothes, which do not really belong to everyday waste, but still is part of it. For this reason, the focus has to be here to get images of other products such as diapers or tissues.

[3]If they are still usable, they should be disposed in the container for old clothes



(a) Glass (b) Organic (c) Paper (d) Recyclable (e) Residual

(f) Glass (g) Organic (h) Paper (i) Recyclable (j) Residual

Figure 1: Example images from online. (a)-(e) are images which correspond to our use case. (f)-(j) are sample images that had to be removed.

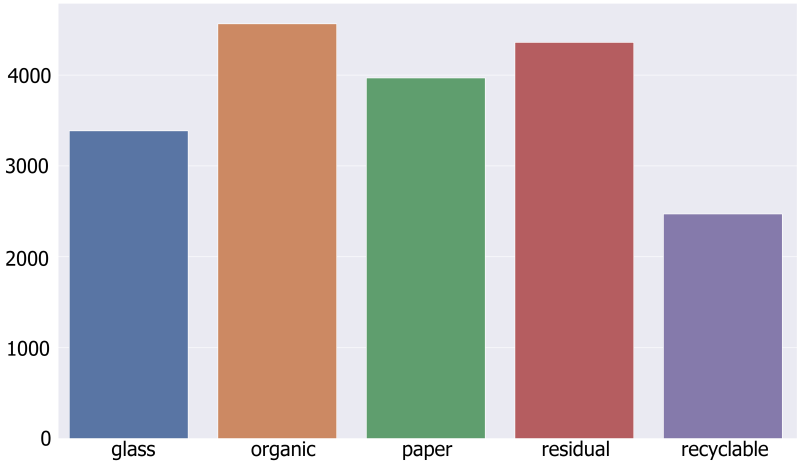
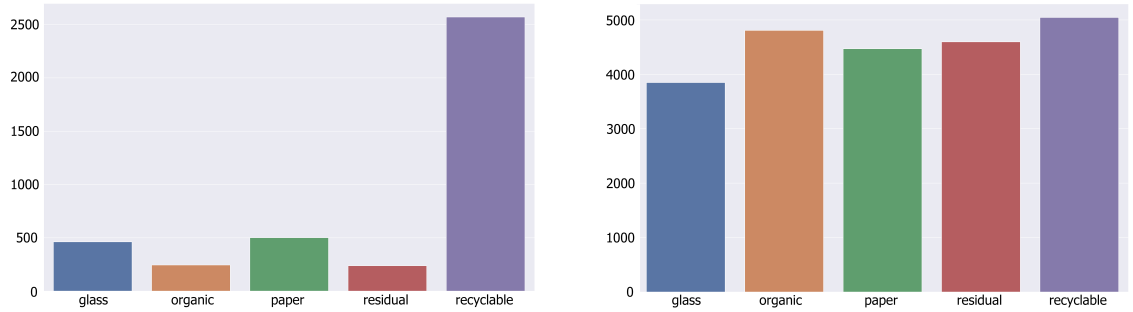


Figure 2: Images per class - online

5.2 Images from our website

Since we still needed pictures for the reasons already mentioned, we decided to create a website through which pictures can be uploaded. This allows the simple help of our family and friends to send us pictures to collect more representative data for our training. Through this more than 4000 images were collected (cf. Figure 3a), which are better suited for the later use case. We also collected another 1600 images for the evaluation of the trained models to test the performance on unknown data. Figure 3b shows, that we ended up with a total of 22,764 images for training. By collecting more pictures containing recyclable

waste, this gave us a better distribution throughout the classes. Finally, the images are divided among the classes as follows: Glass - 3848, Organic - 4808, Paper - 4469, Residual waste - 4596, Recyclable - 5043.



(a) Images for training

(b) Sum all images

Figure 3: Images per class - via our website

5.3 Data augmentation

It is usual that self-taken pictures are made from different angles and perspectives. Due to the major part of the data being from the internet and thus the lack of more individual, well adapted images, we used data augmentation. Data augmentation helps to robustify and enhance the model by applying transformations to the images in order to create more variation for training. In addition, it can prevent overfitting of the training data, since it virtually generates new data from the existing one. The images were all scaled to a size of 300×300. The interpolation method used for the transformation was bicubic interpolation because it uses 16 pixels as the calculation source, which makes the image look smoother after interpolation and contains fewer *interpolation artifacts* [6]. They were also randomly flipped (horizontally or vertically) and rotated by a value between -10 and 10 degrees. To avoid black borders or corners, they were cropped to the final size of 256×256. We chose this size because all the models we used for training can use images of this size as input.

6 Hardware and tools used

Since large, complex architectures are used, working on personal hardware is time inefficient and expensive. The HPO in particular was too memory and compute intensive, so we had to find a way to conduct the training. For this purpose, we used *high-performance GPU servers* for deep and machine learning at Karlsruhe University of Applied Sciences. In this lab, we have 8 x NVIDIA RTX A6000 with 48 Gigabyte (GB) each available. These are very well suited for the large amount of data we have been working with and also experienced maximum utilization due to training processes. This would not have been possible with the personal graphics cards (GeForce GTX 1660TI, GeForce GTX 1080). These were used instead for the data analysis and evaluation processes. It was important to run this on less powerful hardware to get a realistic impression of the processing power because in the application on the website it is similarly large. The code was based on Jupyter notebooks using the PyTorch framework for training, which allowed us to work in an agile and flexible way. Especially when analyzing data, it is advantageous to adapt and experiment quickly. The front-end of the website was realized with HTML, CSS and JavaScript. For the back-end the framework Flask for Python came into use based on personal interests and skills. Furthermore, a simple integration of the models was easy to implement. To store, organize and maintain the data, Google Drive was used as cloud storage. Furthermore, it was possible to pass images uploaded by site users directly to the Google Drive storage via an API in Python to later annotate them. Finally, Heroku [2] offers a hosting service to make the website independent and always available. This service is special because it is one of the few that offer such a service for Python.

7 Methodology

In the following, we will divide the data into two subsets. The first one, called *basic dataset* contains only images from the internet. The second one, which goes by *extended dataset* additionally contains self-taken images.

As evaulation and comparison criteria, we have focused on the accuracy and loss, which are two commonly used metrics to determine the performance of machine learning models. Whether to maximize the accuracy or minimize the loss depends on various of factors and therefore each of the following methods can vary in the metric focused on.

7.1 Hyperparameter optimization (HPO)

Training a machine learning model and especially a deep neural network requires hyperparameters. A good choice of hyperparameters improves the performance of the algorithm. So, in order to automate the training pipeline and face the issue of manually checking hyperparameter values, we performed a hyperparameter optimization to find a tuple of parameters that minimizes the given loss function [14]. For the HPO, *minimizing the loss* was a priority due to the use of pre-trained models and with that a already excellent accuracy. This step was proceeded with Optuna, "an open source hyperparameter optimization framework to automate hyperparameter search" [8] that consists of two counterparts of the optimization: the sampling strategy and the pruning strategy. The former can be understood as an approach where to look for better values of hyperparameters whereas the latter tries to accelerate the hyperparameter optimization by canceling those trials early that are likely to not improve the result. To start a HPO with Optuna, it is required to create a study which includes the direction of optimization[4] and both the sampler and pruner. The pruner further requires a number of start up trials, warm up steps and interval steps. The first two are used to control the number of initial trials and steps that are not affected by pruning, and the last is responsible for controlling the step rate of pruning checks [3]. These can be seen as *hyperhyperparameters* and are required to run the optimization process. "The Tree-structured Parzen Estimator (TPE) is a sequential model-based optimization approach" [5] that was used as the sampler in the implementation. It is based on Bayesian optimization with kernel fitting and uses the previous evaluated values to create an objective function which determines new values [10]. As pruner, the median pruner was used. It prunes "if the trial's best intermediate result is worse than median of intermediate results of previous trials at the same step" [3]. Both are commonly used algorithms in Optuna and have proven to be solid choices [1]. The selection of values used for the HPO was made on the basis of heuristic assumptions and frequently used values(cf. Table 1).

Two commonly used methods to mitigate the problem of optimizing the learning rate in

[4] maximization and minimization respectively
[5] Was used only for SGD optimizer
[6] Was used only in SSWQ

hyperparameter	values
learning rate	$[10^{-4}, 10^{-2}]$
batch size	$16, 32, 64, 128$
optimizer	SGD, Adam, Adagrad, RMSprop
momentum[5]	$[0.85, 0.95]$
dropout[6]	$0.3, 0.4, 0.5$
hidden layer 1[6]	$512, 1024, 2048$
hidden layer 2[6]	$512, 1024, 2048$

Table 1: Selected values for HPO

machine learning are Adagrad and Adam. Adagrad is an algorithm that adapts the learning rate over iterations by keeping track of the sum of previous gradients. Due to the sum of gradients always grow, especially in deeper neural networks the learning rate can become very small which will produce a dead neuron problem [26] and the progress stagnates. Adam extends Adagrad by using a decay rate in order to focus on recent gradients while also considering the first[7] and second momentum[8] [11] to mitigate the problem of Adagrad.

7.2 Model training

To train the models, the data was partitioned into a training, validation and test subset. The parameters were the same for all models, except for the hyperparameters shown in the previous chapter. Every model was trained in 20 epochs with a validation split of 15% and the best hyperparameters from the HPO. For this reason, the training only contained 20 epochs and used only the models where there was a significant improvement in training to avoid overfitting. To obtain better results we employed transfer learning, a method to apply already gained knowledge from a former to a new problem. Thus PyTorch provides pre-trained model versions trained on the ImageNet dataset [12]. To accomplish a high variance of models within a least amount of models, we chose the models ResNet-50, VGG16, EfficientNet-B3, EfficientNet-B4 and our own model SSWQ.

We wanted to experiment with different architecture principles while also testing historical, classical neural networks as ResNet or VGG in comparison to newer, upcoming models as EfficientNet. Especially interesting was the explicit competition between EfficientNet B3 and B4 to evaluate the outcome differences of slight disparities in the model architecture.

Residual neural networks (ResNet) [17] were a groundbreaking invention in the last few years. They face the degradation problem of convolutional neural networks by inserting skip connections to perform an identity mapping which is added to the outputs of the original stacked layer. Since then it was possible to build and train deeper neural networks

[7] the mean
[8] the uncentered variance

while still archiving great performance. Since ResNet is always evolving there are several variation such as ResNeXt [34], DenseNet [18] and BiT models [22] with layers of up to 152 [17]. For size and complexity related reasons we used ResNet-50 because we consider the model to be sufficiently complex and suitable for our problem.

VGG, which is an abbreviation for Visual Geometry Group, also refers to a ConvNet proposed in [29]. The model originally was developed for high-resolution image recognition. In comparison to existing ConvNet's, this model uses very small 3×3 filters in order to decrease the number of parameters per layer in order to increase the number of layers. Therefore the model is able to "incorporate three non-linear rectification layers instead of a single one, which makes the decision function more discriminative." [29].

EfficientNet denotes a currently aspiring ConvNet model family. A ConvNet has different dimensions of scaling as shown in Figure 4. As observed in [30], scaling up a network in

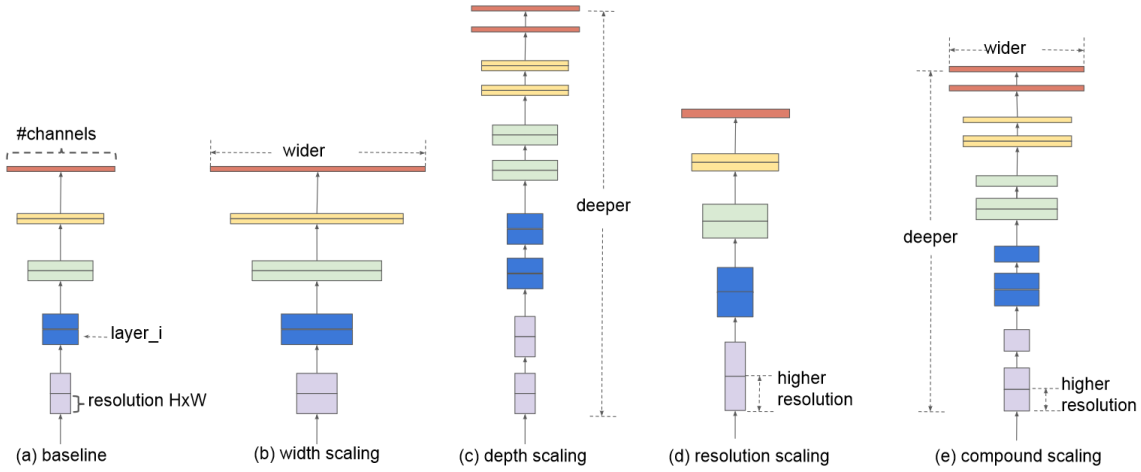


Figure 4: Model scaling methods. (a) is a base network example; (b)-(d) are model scalings in one dimension of either width, depth, or resolution. (e) is the compound scaling method [30]

.

any dimension improves accuracy in respect of the degradation problem or the Vanishing gradient problem and second, efficiency and accuracy requires to balance all dimensions of the network during ConvNet scaling. Due to this, [30] proposed a new *compound scaling method*, which scales depth, width and resolution of the network uniformly [7]. The training evaluation is based on accuracy and loss respectively, taking into account both validation and training outcome.

SSWQ	
Block	Layers
1	conv7-32
maxpool2	
2	conv3-64 conv3-64 conv3-64
maxpool2	
3	conv3-128 conv3-128
maxpool2	
4	conv3-256 conv3-256
maxpool2	
5	conv3-512 conv3-512 conv3-512
maxpool2	
dropout	
FC-hd2[9]	
dropout	
FC-5	
soft-max	

Figure 5: SSWQ architecture

We challenged ourselves by designing a own model through simple principles and thoughts. SSWQ, an abbreviation and homage to one favored artist, denotes a convolutional neural network. It takes input images of size $3 \times 256 \times 256$ which first are convoluted by a 7×7 kernel followed by a max pooling layer of 2×2. Every further block consists of two to three 3×3 convolutional layer stacking up the output layers from 32 in the beginning up to 512 in the fifth block and finishes with a max pooling layer of 2×2. After five convolutional blocks, the input runs through two fully connected layers, each with a dropout of 0.5 in between and finally a soft-max function to calculate the output (cf. Figure 5). In addition to the existing hyperparameters, SSWQ also takes a dropout rate and the amount of the fully connected layers.

In addition to the model training, we designed a separate evaluation process across all models and datasets to allow direct comparison of results.

8 Results and Discussion

In the following chapter, the results of the hyperparameter optimization, the model training and, in addition, of an independent overall evaluation are presented and discussed.

8.1 Hyperparameter optimization

The hyperparameter optimization results illustrated in Table 2 and Table 3 show the best tuple of hyperparameters for each model. Due to a strong deviation from the pre-trained models, SSWQ is not considered in the following evaluation and is covered separately. It is noticeable, that, independent of the dataset, every model chose Adagrad. It seems to optimize the parameters the best. This means that the HPO probably did not experience any dead neuron problems and the algorithm thus could unfold its high converging speed. Otherwise the missing performance of Adagrad is likely to cause the selection of another optimizer as the best.

Based on the self-taken images, there are correlations in the results. First, the batch size got smaller in most cases[10]. Smaller batch sizes generally result in faster, but more volatile learning. In addition, [21]brought up the observation that larger batches tend to give slightly worse results due to poorer generalization ability. Second, the learning rate got smaller in most cases[11]. A lower learning rate thus seems to favor the computation of the minimum which results in a faster training.

SSWQ does not change its hyperparameters between the two datasets except for the learning rate, which, as with the other models, decreases slightly. It is interesting to discover a whole new tuple of hyperparameters that could arise due to its architecture and the fact that it is not pre-trained.

model name	ResNet-50	VGG16	EfficientNet-B3	EfficientNet-B4	SSWQ
learning rate	$4.82 \cdot 10^{-4}$	$3.27 \cdot 10^{-4}$	$2.15 \cdot 10^{-3}$	$1.50 \cdot 10^{-3}$	$1.32 \cdot 10^{-4}$
batch size	64	64	128	64	16
optimizer	Adagrad	Adagrad	Adagrad	Adagrad	Adam
dropout	-	-	-	-	0.5
hidden layer 1	-	-	-	-	512
hidden layer 2	-	-	-	-	2048

Table 2: Hyperparameter optimization results of basic dataset

[10]Except for EfficientNetB4, which remained the same.
[11]Except for VGG16, which got bigger.
[11]hidden layer 2

model name	ResNet-50	VGG16	EfficientNet-B3	EfficientNet-B4	SSWQ
learning rate	$3.48 \cdot 10^{-4}$	$3.70 \cdot 10^{-4}$	$2.00 \cdot 10^{-3}$	$1.44 \cdot 10^{-3}$	$1.20 \cdot 10^{-4}$
batch size	32	32	64	64	16
optimizer	Adagrad	Adagrad	Adagrad	Adagrad	Adam
dropout	-	-	-	-	0.5
hidden layer 1	-	-	-	-	512
hidden layer 2	-	-	-	-	2048

Table 3: Hyperparameter optimization results of extended dataset

There were no correlations between the individual models and the two data sets regarding hyperparameter importance. Therefore, it will not be discussed further in the following.

8.2 Training

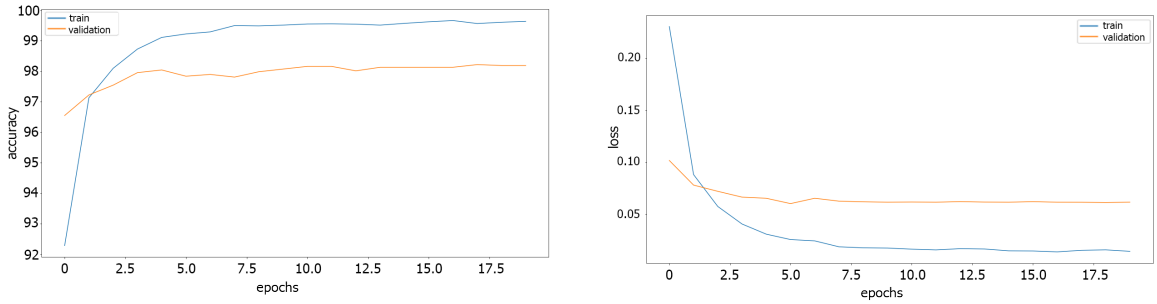
The results were obtained on $3,400$ random validation images. As can be seen in Table 4, the accuracy of the pre-trained models is excellent at over 98% except for VGG16, which only is around 96%. This means, the models have almost assigned all images to the correct class after training. The loss only differs minimally too. For the training of the basic dataset, the EfficientNet-B4 model has the best accuracy and loss. One exception is the extended dataset. Here, B3 has a slightly better accuracy. However, it should be noted that both models do nearly not differ in loss and thus it is hard to detect whether B3 really outperformed B4. The time difference between both datasets are approx. 56 seconds on average with a standard deviation of 16.51. That is, the time difference between the two data sets depends on several factors that will not be discussed further here. The time comparison is not as interesting as later when it comes to the time required by the models for evaluating images the same images.

	Online images			**Extended with own images**		
model name	Accuracy	Loss	∅ time	Accuracy	Loss	∅ time
ResNet-50	98.22%	0.063	$95s$	97.07%	0.1	$165s$
VGG16	96.05%	0.16	**94.5s**	95.49%	0.17	$159.32s$
EfficientNet-B3	98.54%	0.056	$107s$	**97.92**%	0.074	**149.3s**
EfficientNet-B4	**98.76**%	**0.048**	$132s$	97.83%	**0.073**	$166.9s$
SSWQ	78.14%	0.77	$102.6s$	75.7%	0.81	$172.2s$

Table 4: Training results

The performance of SSWQ reaches almost 80% accuracy although it is not pre-trained or optimized. Nevertheless, a comparison between the pre-trained models and SSWQ is not

proportionate and purposeful. For this reason, our focus is naturally on the strong, pre-trained models, as they have achieved excellent results. In the direct comparison of the accuracy, SSWQ is noticeable worse in the extended dataset. This is probably due to the self-taken images, which widely differ from the basic dataset images. Interestingly, VGG16 has not changed significantly. This could mean that this model is better able to adapt to a wider variety of image features.

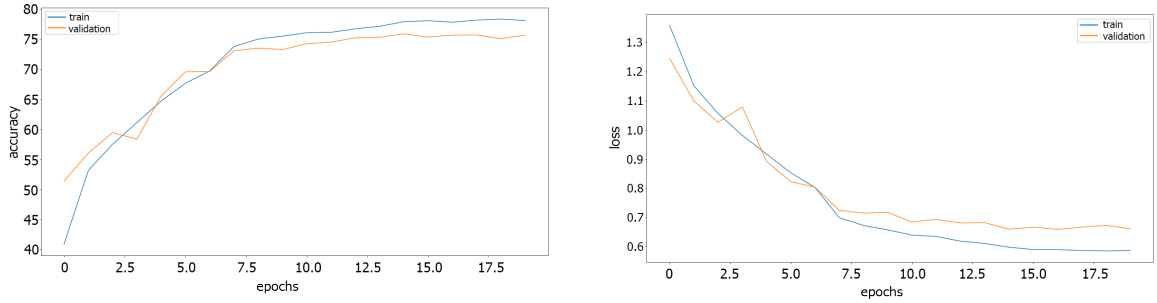


(a) Accuracy (b) Loss

Figure 6: Training history of EfficientNet-B4.

The training process was characterized by a rapid improvement of accuracy and loss in the first few epochs as shown in Figure 6, which is representative for all pre-trained models. Further, it becomes clear that loss and accuracy do not change significantly after a few epochs.

In the training history of our model, an improvement over the epochs is more recognizable (cf. Figure 7). However, even here a plateau is reached after a few epochs and no significant improvements are achieved. It is possible that after many more epochs there would have been more improvement again. However, as mentioned earlier, we wanted to train all models with the same number of epochs.



(a) Accuracy (b) Loss

Figure 7: Training history of our Model SSWQ.

8.3 Overall evaluation

This section analyses an additional evaluation across all models in both versions to provide a basis for comparison. This procedure was based on 1634 new images that were never used before and were considered representative of the mapping of the problem.

Table 5 summarizes the overall results of a selection of ConvNet's comparing the two datasets. The fastest models are VGG16 and SSWQ, both between 270 and 280 seconds whereas the slowest model requires around 295 to 300 seconds. Of course, the evaluation time within a model was neither increased nor decreased because every model is trained on the same data. Slight variances in time could be due to inconsistencies in data processing. This leads us to the first observation:

Observation 1 - There is no large time difference between the model evaluations.

That said, each studied model is applicable for image classification on the website.

The most accurate models are the EfficientNet's B3 and B4 with a accuracy of around 72 to 73% and 87%, respectively(cf. Table 5).

	Basic dataset		**Extended dataset**	
model name	Accuracy	time	Accuracy	time
ResNet-50	67.81%	296.51s	82.69%	293.96s
VGG16	67.68%	279.13s	77.20%	**270.10s**
EfficientNet-B3	72.03%	289.81s	**87.77**%	290.14s
EfficientNet-B4	**73.37**%	297.33s	87.65%	300.17s
SSWQ	40.13%	**270.80s**	48.60%	273.31s

Table 5: Training results

SSWQ is, as expected, the worst when it comes to accuracy. Due to it's a priori disadvantage of not being pre-trained plus being relatively simple constructed, our expectations regarding the accuracy were low. The hope was, due to simplicity, to register a medium amount of time saving. It should be mentioned, of course, that our intention was not to built a competitive model, but to our disappointment the model could not perform sufficiently. As a matter of fact, it is even worse in runtime than VGG16.It should be mentioned, of course, that our intention was not to built a competitive model, but to our disappointment the model could not perform sufficiently, but, as a matter of fact, it is even worse in runtime than VGG16. ResNet-50 has a better accuracy than VGG16, especially with the extended dataset, which can be traced back to the idea of residual neural networks [17]. When it's about runtime, VGG16 is naturally faster than ResNet-50 due to it's simple, convolutional architecture of 16 layers. The difference in accuracy between the EfficientNet's B3 and B4 are vanishingly small whereas the time saving of B3 is around 10 seconds, probably due to lower scaling

factors and thus simpler architecture. According to [30], it is not surprising to see that the EfficientNet's deliver the best results.

Observation 2 - Self-taken pictures have a major influence on the accuracy of the trained networks.

This is in line with the training results in Section 8.2 where increased classification difficulty can be detected due to self-taken pictures. From this, a correlation between the performance of the neural network and the type of data used can be deduced. Moreover, data is an significantly important factor, if not the most important factor to train a good neural network. In fact, This significant impact meets the expectations, as it is apparently clear that self-taken images fit the problem better than online images from the basic dataset. Pre-trained networks [12] on ImageNet data have shown adequate results too. This is hardly surprising, as the weights are especially well optimized based on over 14 million images and 20 thousand classes, but it works well as a basis for building a competitive classification model on an individual problem. The countplots present the true and false predictions by the corresponding model (cf. Section 10.1). Overall a moderate increase in accuracy can also be observed due to the countplots. Especially glass and recyclable material tremendously increased by around 30%. Paper mostly remained the same, so the own images did not influence the accuracy there. The overall accuracy from Table 5 is reflected in the countplots. Thus the EfficientNet's record a vast increase while SSWQ is still suffering.

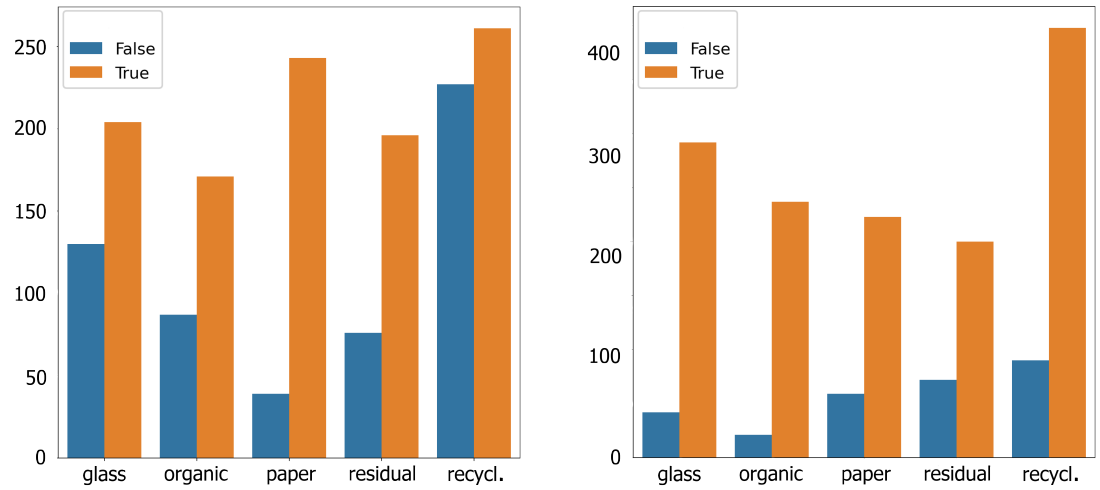
It should be noted that although the trained model is sufficiently accurate, it cannot perfectly detect all wastes due to the lack of data. Therefore, it is likely that there will be problems in predicting uncommon or rare wastes. Furthermore, we only compared a small selection of neural networks. Consequentially, other models could provide better results in order to attack this classification problem. Lastly, the data do not fully represent global or even German average household waste due to a lack of data collection sources.

9 Conclusion

In conclusion, we proposed a machine learning model that is able to classify images of waste. It separates them into the five classes of the German waste system assisting the waste cycle and preventing avoidable pollution and the waste of resources. Perserving a clean environment for successive generations is a priority thus this paper aims to support that. According to the results, the EfficientNet-B3 model trained on extended data will be used on the website to classify waste from now on. The model still acts as an advisor and assistant rather than a fully autonomous decision maker yet. To improve the results, we could have collected more data, especially own data through our website. The classification could be extended to more and detailed classes such as electric waste, hazardous materials and basic resources as metal or wood. The most effective extension would be a real time object segmentation which then could also be used in waste sorting systems.

10 Appendix

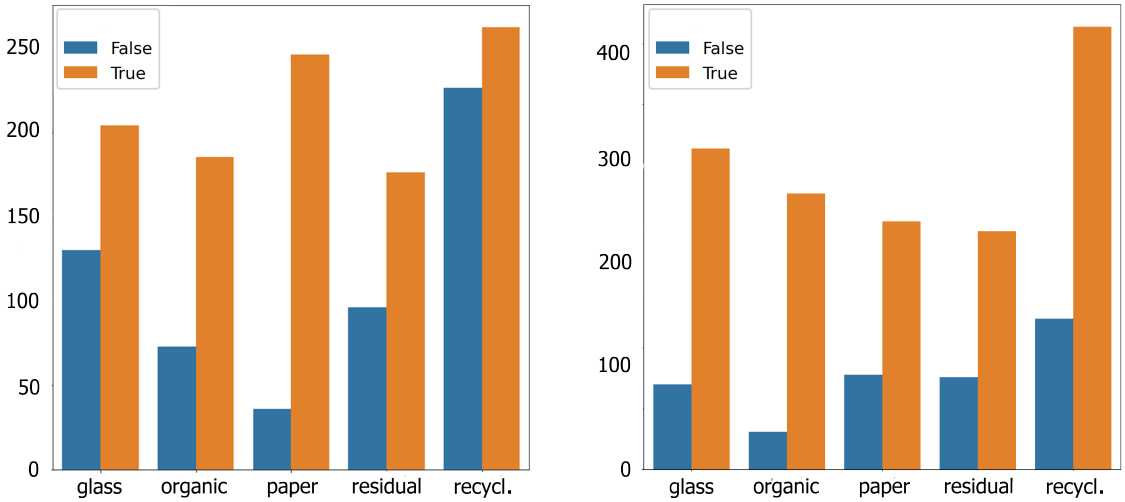
10.1 Countplots



(a) ResNet-50 without own images

(b) ResNet-50 with own images

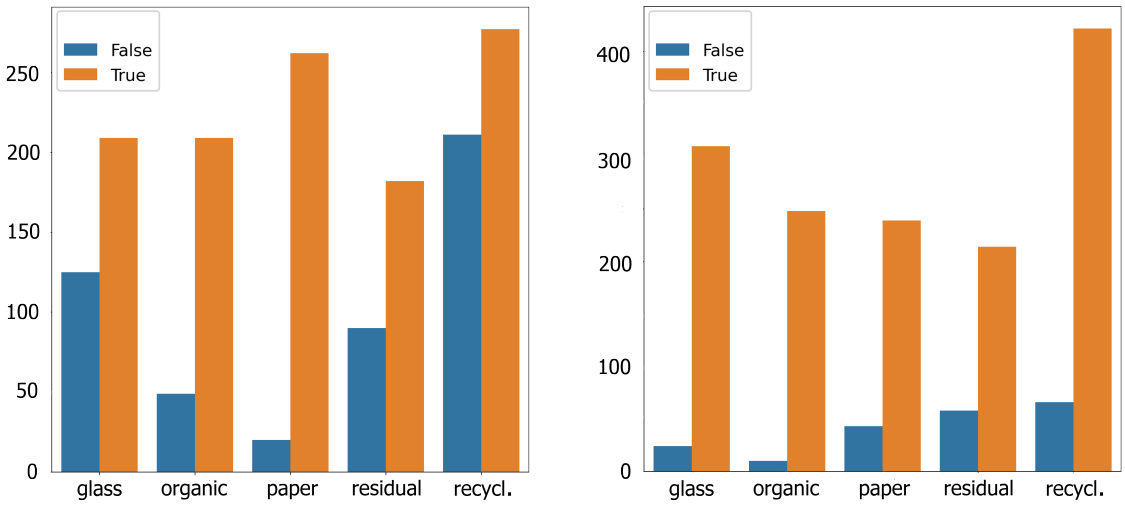
Figure 8: Countplots ResNet-50



(a) VGG16 without own images

(b) VGG16 with own images

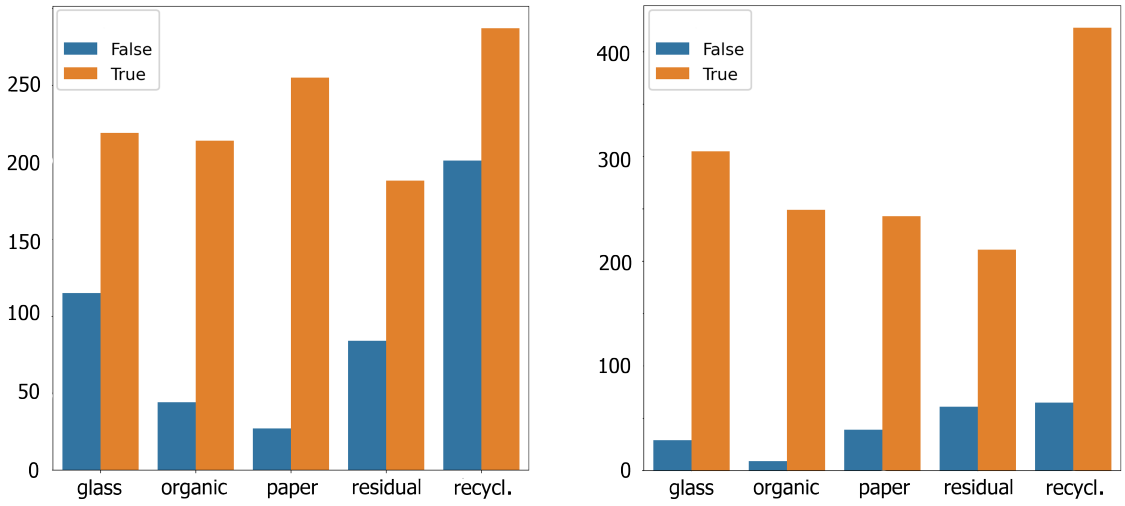
Figure 9: Countplots VGG16



(a) EfficientNet-B3 without own images

(b) EfficientNet-B3 with own image

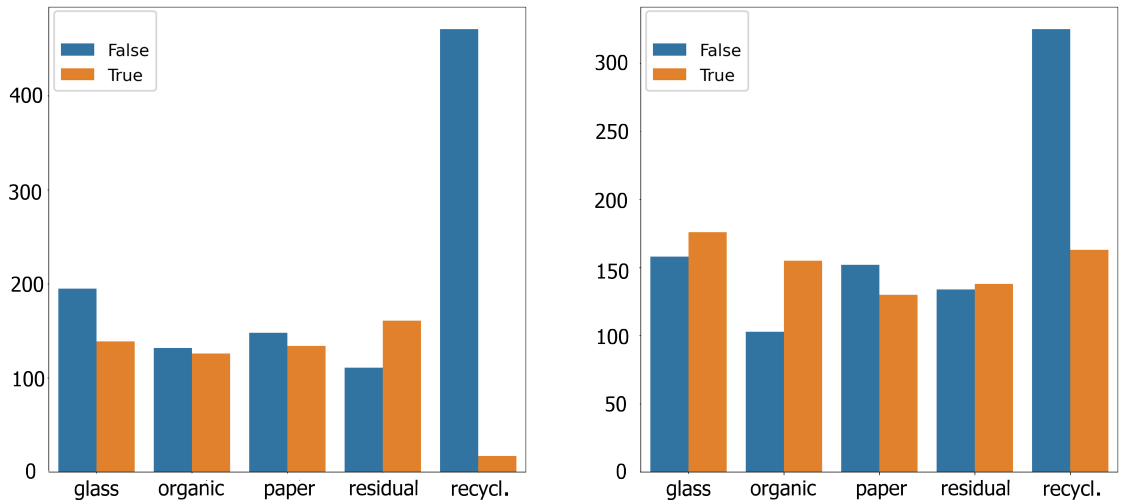
Figure 10: Countplots EfficientNet-B3



(a) EfficientNet-B4 without own images

(b) EfficientNet-B4 with own images

Figure 11: Countplots EfficientNet-B4



(a) SSWQ without own images

(b) SSWQ with own images

Figure 12: Countplots SSWQ

References

[1] Efficient optimization algorithms. https://optuna.readthedocs.io/en/stable/tutorial/10_key_features/003_efficient_optimization_algorithms.html. Accessed July 30th, 2022.

[2] Heroku - hosting service for our website. https://www.heroku.com/python.

[3] optuna.pruners.medianpruner. https://optuna.readthedocs.io/en/stable/reference/generated/optuna.pruners.MedianPruner.html/. Accessed July 30th, 2022.

[4] Our app hosted by heroku. https://waste-classificater.herokuapp.com/.

[5] Solver overview. https://optunity.readthedocs.io/en/latest/user/solvers.html. Accessed July 30th, 2022.

[6] Bicubic interpolation. https://en.m.wikipedia.org/wiki/Bicubic_interpolation, 10 2021. Accessed August 24th, 2022.

[7] Vardan Agarwal. Complete architectural details of all efficientnet models. https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142, May 2020.

[8] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[9] Angloinfo. Refuse collection and recycling in france. https://www.angloinfo.com/how-to/france/housing/setting-up-home/recycling. Accessed July 27th, 2022.

[10] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[11] Jason Brownlee. Gentle introduction to the adam optimization algorithm for deep learning. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/, January 2021.

[12] Torch Contributors. Models and pre-trained weights. https://pytorch.org/vision/stable/models.html.

[13] Dr. Heinz-Josef et al. Dornbusch. Vergleichende analyse von siedlungsrestabfällen aus repräsentativen regionen in deutschland zur bestimmung des anteils an problemstoffen und verwertbaren materialien. June 2020.

[14] Matthias Feurer and Frank Hutter. *Hyperparameter Optimization*, pages 3–33. Springer International Publishing, Cham, 2019.

[15] Garythung. Trashnet. https://github.com/garythung/trashnet, 2017. Accessed July 27th, 2022.

[16] Go Greener. The consequences of poor waste management. https://www.gogreenerltd.co.uk/the-consequences-of-poor-waste-management/.

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

[18] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[19] Stadt Karlsruhe. Abfallentsorgung in karlsruhe. https://www.karlsruhe.de/stadt-rathaus/service-buergerinformation/abfallwirtschaft/abfallentsorgung. Accessed August 24th, 2022.

[20] T Kennedy. Oscarnet: Using transfer learning to classify disposable waste; cs230 report: Deep learning, 2018.

[21] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima, 2016.

[22] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Large scale learning of general visual representations for transfer. *CoRR*, abs/1912.11370, 2019.

[23] R. Losurdo. Consequences of poor industrial waste disposal methods/services. https://mcfenvironmental.com/poor-industrial-waste-disposal-methods/.

[24] Naturschutzbund Deutschland (NABU). Abfallverbrennung ist nicht klimaneutral: mit echter kreislaufwirtschaft klima und ressourcen schützen. https://www.nabu.de/imperia/md/content/nabude/konsumressourcenmuell/211102_abfallverbrennung_policypaper.pdf.

[25] Naturschutzbund Deutschland (NABU). Das schlummernde potenzial in der schwarzen tonne. https://www.nabu.de/umwelt-und-ressourcen/abfall-und-recycling/kreislaufwirtschaft/29148.html.

[26] Luthfi Ramadhan. Neural network: The dead neuron. https://towardsdatascience.com/neural-network-the-dead-neuron-eaa92e575748, November 2021.

[27] Victoria Ruiz, Ángel Sánchez, José F. Vélez, and Bogdan Raducanu. Automatic image-based waste classification. In José Manuel Ferrández Vicente, José Ramón Álvarez-Sánchez, Félix de la Paz López, Javier Toledo Moreo, and Hojjat Adeli, editors, *From Bioinspired Systems and Biomedical Applications to Machine Learning*, pages 422–431, Cham, 2019. Springer International Publishing.

[28] Sashaank Sekar. Waste classification data. https://www.kaggle.com/datasets/techsash/waste-classification-data, 2019. Accessed July 27th, 2022.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[30] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.

[31] Gary Thung and Mingxiang Yang. Classification of trash for recyclability status. 2016.

[32] Unknown. Illegal dumping: Causes, effects and solutions to huge piles of wastes. https://www.conserve-energy-future.com/causes-effects-solutions-illegal-dumping.php.

[33] Wikipedia. Waste sorting — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Waste%20sorting&oldid=1092911241, 2022. Accessed August 2nd, 2022].

[34] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016.