Trash Material Classification via Deep Learning

Abdulellah Abualshour
King Abdullah University of Science and Technology

abdulellah.abualshour@kaust.edu.sa

Abstract

Trash material classification is an interesting task that helps save the environment in its applications. In my project, I tackle the material classification problem using deep learning in search of the best ways to classify trash material for environmental purposes. A good convolutional model can help create useful application for sorting trash material without the need of human contact with recycling equipment. In my experiments, I will try out several methods and networks, including different optimization techniques and different network architectures in order to find the best and most accurate convolutional models for the trash material classification task on benchmark datasets. I will rely mostly on fine-tuning existing models in order to spend more time exploring and comparing different methods.

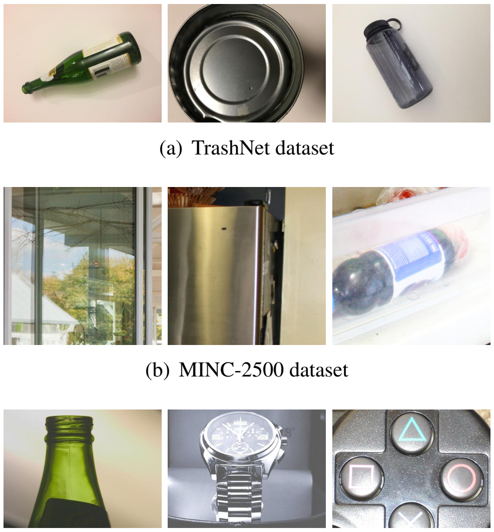
1. Introduction

1.1. Motivation

Waste management is a very important task to preserve the environment. Environment and natural materials played an important function in the development of human societies and in history on the whole [1]. Therefore, trash material classification helps a lot with achieving the goal of a better environment by applying deep learning methods to help us build useful applications dedicated for categorizing different types of materials given suitable datasets.

1.2. Problem Statement

In this project, I try to find the best and most efficient approaches to image classification suitable for trash material classification, applied on benchmark datsets that will be introduced in a later section. My hope in this project is to implement and develop very accurate and efficient classifiers for the task of material classification.



(a) TrashNet dataset

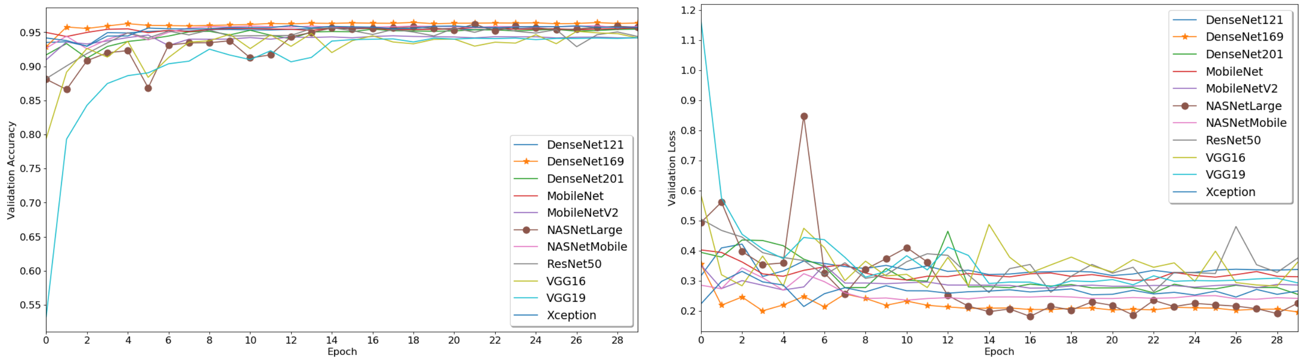
(b) MINC-2500 dataset

(c) FMD dataset

Figure 1. Samples from benchmark datasets used. Columns from left to right: glass, metal, plastic.

2. Related Work

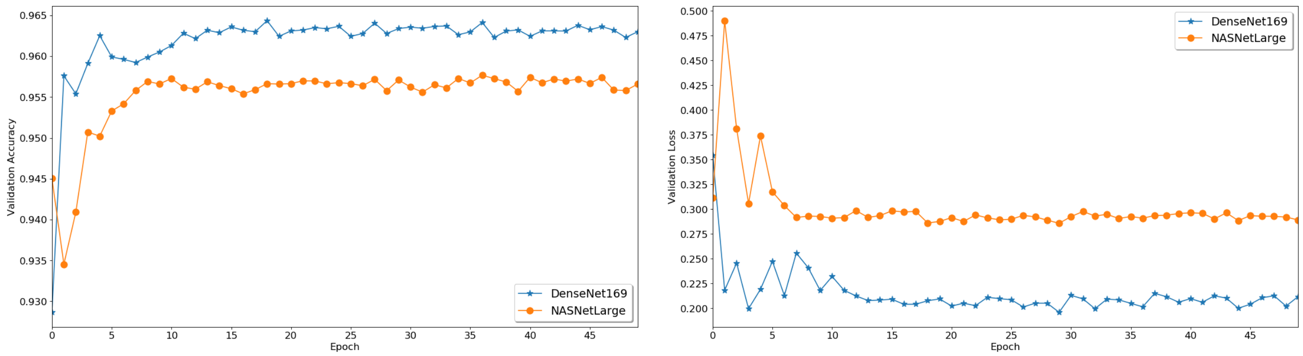
Schwartz et al. [13] explored the difference between different material types and their properties from an image recognition point of view, illustrating how different materials show different textures and smoothness and how their features can be useful in image recognition tasks. Another interesting paper is by students from Stanford University [17] where they explored computer vision and deep learning approaches to classify trash material using a dataset that they acquired by hand for the purpose of their project. This dataset is used as a benchmark dataset in this project. Bircanoglu et al. [3] introduce their own RecycleNet architecture dedicated for recycling material classification and also some experimintations with other networks. In addition to these related papers, I explore different networks as base models in my experiments, such as ResNet50 [8], DenseNet [10], and others in hopes of finding the most suitable clas-



(a) Validation Accuracy

(b) Validation Loss

Figure 2. **Validation accuracy and loss during training using different fine-tuned architectures on the TrashNet [16] dataset.** NASNetLarge and DenseNet169 seem to perform similarly as the epoch number increases during training. When evaluating the models during the test phase, NASNetLarge perfroms the best on average.



(a) Validation Accuracy

(b) Validation Loss

Figure 3. **Validation accuracy and loss during training the state-of-the-art beating models on TrashNet [16] dataset.** Both DenseNet169 and NASNetLarge beat state-of-the-art accuracy.

sifier for trash material classification.

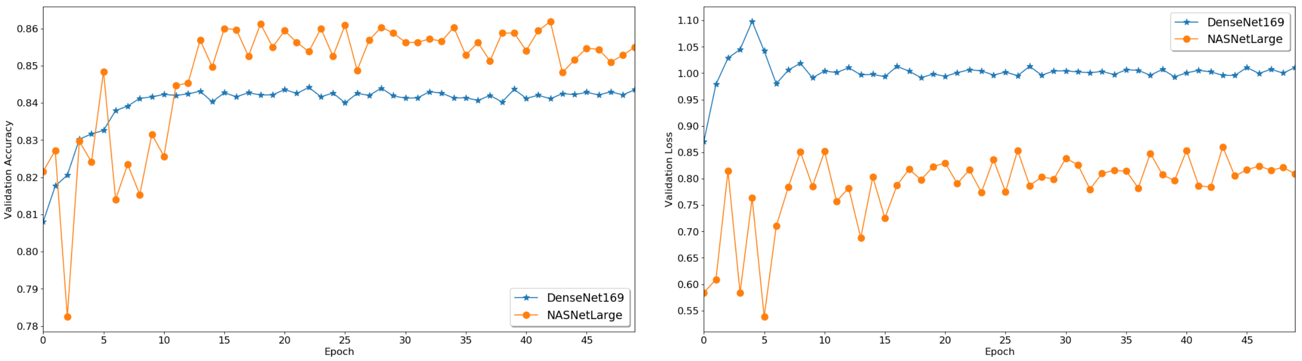
3. Datasets

In this project, I focus on trash material classification. In order to look for a suitable model of classification for such material, I collected three datasets that have been used for similar tasks. Benchmark datasets used in this project include the TrashNet dataset [16], the Flickr Material Database (FMD) [14] and the Materials in Context Database (MINC-2500) [2]. These datasets will be used to experiment with different convolutional models. The TrashNet dataset contains 2527 images belonging to 6 classes (paper, plastic, cardboard, glass, metal, and uncategorized). FMD contains 1000 images belonging to 10 classes (paper, plastic, glass, metal, fabric, foliage, leather, stone, water, and wood). Finally, MINC-2500 dataset contains 57500 images belonging to 23 classes (brick, carpet, ceramic, fabric, foliage, food, glass, hair, leather, metal, mirror, painted, paper, plastic, polished stone, skin, sky, stone, tile, wallpaper, water, wood, and uncategorized).

4. Methodology

4.1. Convolutional Neural Networks

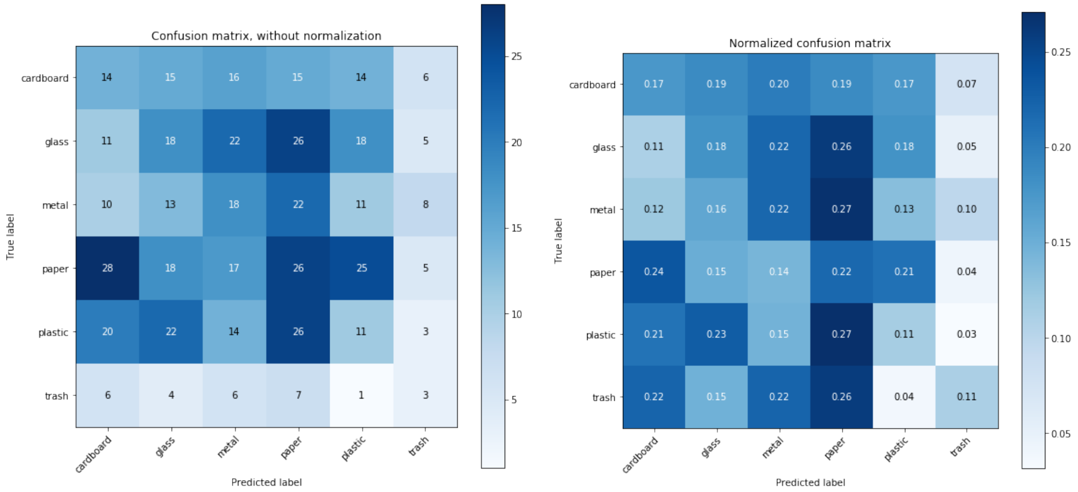
I rely mostly on fine-tuning existing models in the deep learning industry. In this case, I take different existing architectures pre-trained on ImageNet [5] and fine-tune them to do the task of trash material classification. Networks to be considered along with their tests results are shown in Table 1. I also implement my own model, but it does not perform as good as fine-tuned models when trained from scratch. Therefore, I have stopped further experiments using the model I implemented in order to focus on fine-tuning and optimizing results.



(a) Validation Accuracy

(b) Validation Loss

Figure 4. **Validation accuracy and loss during training the state-of-the-art beating models on MINC-2500 [2] dataset.** Both DenseNet169 and NASNetLarge beat state-of-the-art accuracy.



(a) Non-normalized Confusion Matrix

(b) Normalized Confusion Matrix

Figure 5. **Confusion matrices illustrating the errors in prediction on the TrashNet test set.** Uses the best performing model.

4.2. Data Augmentation

Augmentation to the data is applied on the training set and the test set to increase classification accuracy. Augmentation techniques initially used include shifting images horizontally and vertically, rotation by a 45 degree angle, horizontal flipping, shearing, and zooming. However, I've decided to eliminate shifting, shearing and zooming and only keep a rotation by 15 degree angle and horizontal flipping in order to replicate augmentation done by state-of-the-art results on TrashNet. After further observation, it can be concluded that some augmentations contribute to the loss of some features the resulted in bad training trials. After observing some false negatives and true negatives, the model seemed to think paper was cardboard more often using the initial augmentation approach. However, it performed better on other classes on average, but overall the performance peeked when using the final augmentation approach of only rotating by a 15 degree angle and flipping horizontally.

5. Results

5.1. Evaluation

For performance evaluation, I decided to use the following metrics: categorical accuracy, categorical cross-entropy loss, mean squared error, mean absolute error, and mean squared logarithmic error. Test results for these metrics are shown in Table 1. Figure 5 shows the confusion matrices resulting from evaluating the test set of TrashNet using the best performing model (DenseNet169). It can be observed that paper often interestingly gets misclassified as cardboard, which can be justified given how cardboard im-

Architecture	$\delta \uparrow$	cce$\downarrow$	mse$\downarrow$	mae$\downarrow$	msle$\downarrow$	Parameters	Batch Size
ResNet50 [8]	0.880	1.088	0.036	0.040	0.016	25m	32
VGG16 [15]	0.852	1.090	0.042	0.050	0.020	15m	32
VGG19	0.859	1.062	0.041	0.049	0.020	20m	32
MobileNet [9]	0.875	1.258	0.039	0.044	0.019	4m	32
MobileNetV2	0.866	1.062	0.039	0.046	0.019	3m	32
DenseNet121 [10]	0.870	1.146	0.041	0.045	0.020	7m	8
DenseNet169	0.885	1.123	0.035	0.039	0.017	13m	8
DenseNet201	0.881	0.945	0.035	0.040	0.017	18m	8
Xception [4]	0.880	0.806	0.033	0.042	0.016	23m	8
NASNetMobile [19]	0.877	0.987	0.035	0.041	0.017	5m	32
NASNetLarge	**0.901**	**0.791**	**0.031**	**0.038**	**0.015**	88m	4

Table 1. **Comparisons of test results of different fine-tuned architectures on the TrashNet dataset using Adam as an optimizer.** Rows from left to right: architecture, test accuracy, categorical cross-entropy, mean squared error, mean absolute error, mean squared logarithmic error, number of trainable parameters in millions, and batch size. The reported numbers are real results from the experiments.

Architecture	$\delta \uparrow$	cce$\downarrow$	mse$\downarrow$	mae$\downarrow$	msle$\downarrow$	Parameters	Batch Size
ResNet50 [8]	0.864	1.202	0.046	0.050	0.025	25m	32
VGG16 [15]	0.820	1.310	0.053	0.063	0.033	15m	32
VGG19	0.829	1.290	0.051	0.061	0.030	20m	32
MobileNet [9]	0.845	1.302	0.042	0.049	0.031	4m	32
MobileNetV2	0.844	1.112	0.043	0.049	0.027	3m	32
DenseNet121 [10]	0.850	1.226	0.052	0.055	0.032	7m	8
DenseNet169	0.873	1.243	0.046	0.040	0.020	13m	8
DenseNet201	0.870	1.145	0.040	0.038	0.018	18m	8
Xception [4]	0.866	0.906	0.035	0.049	0.022	23m	8
NASNetMobile [19]	0.852	1.023	0.040	0.043	0.019	5m	32
NASNetLarge	**0.891**	**0.802**	**0.034**	**0.038**	**0.017**	88m	4

Table 2. **Comparisons of test results of different fine-tuned architectures on the TrashNet dataset using SGD as an optimizer.** Rows from left to right: architecture, test accuracy, categorical cross-entropy, mean squared error, mean absolute error, mean squared logarithmic error, number of trainable parameters in millions, and batch size. The reported numbers are real results from the experiments.

Architecture	$\delta \uparrow$
Zhang et al. (Deep-TEN) [18]	0.813
Ours (DenseNet169)	0.842
Ours (NASNetLarge)	**0.862**

Table 3. **Our results compared to state-of-the-art accuracy on MINC-2500 dataset.** We beat state-of-the-art.

ages and paper images in TrashNet often produce similar features.

5.2. Experiments

I have tested the architectures using several optimizers: SGD [12], Adam [11], RMSprop [7], Adamax [11] and Nadam [6]. Among all of these optimizers, Adam performed the best, as shown in Table 1. The second best optimizer was SGD, and Table 2 illustrates the results obtained. I have I also implemented learning rate reduction on plateau to help the training process get a little boost if no real progress was made for 5 epochs during training. All experiments for TrashNet dataset are run for 30 epochs with a batch size of 32 except for dense networks due to hardware limitations such as NASNetLarge in which I used a batch size of 4, Xception and DenseNet in which I used a batch size of 8, in all my initial experiments. However, I got access to the IBEX cluster at KAUST and was able to use a Tesla v100 node for experiments, which allowed me to train using a batch size of 32. The final results are illustrated in Tables 3 and 4.

5.3. User Interface

I have implemented a simple graphical user interface (GUI) that can be used to test my best performing model for trash material classification. The user can browse their computer to choose an image they took or an image from the test set of TrashNet in order to classify the object. It is recommended to take a picture of an object with a white background and avoiding shadows in order to achieve the best prediction results possible. A white background is recommended because the dataset used in training has a white background for all objects in the samples. This is okay because in a real world application, a recycling bin can be designed with well lighted interior and white backgrounds to ensure accurate prediction.

6. Future Work

6.1. FMD Investigation

For some unknown reason, the FMD dataset did not yield satisfying results during experiments. I intend to delve deeper into the issue and dig into related work in which FMD was used to see if I can improve the performance of training using the FMD dataset.

6.2. GUI Improvement

I intend to improve the GUI implemented in this project. I also intend to use the best performing model in a future project where it can be applied to be used in smart containers for recycling.

6.3. Model Implementation

The work done inside a convolutional neural network is still a mystery to the deep learning industry. No one knows exactly what contributes to such good results when it comes to classification. However, I intend to research CNNs even more especially in the field of material classification and recognition to see if distinctive features of different materials contribute to the difference between these materials when it comes to training a network. A CNN should not be treated like a black box, like it is usually used by many deep learning researchers and enthusiasts.

7. Conclusion

I was able to beat state-of-the-art results on two of the three benchmark datasets (TrashNet and MINC-2500). For optimization, Adam performed the best among all tested optimizing methods. NASNetLarge seemed to perform the best among all architectures used in MINC-2500 as shown in Table 3. DenseNet169 on the other hand was better as an architecture for TrashNet, as shown in Table 4. These better results were achieved by providing better hardware for training, that resulted in my ability to increase batch size during training. Elimination of some augmentations also helped increase the accuracy due to the nature of the datasets and how some augmentation techniques make the samples lose some features.

Architecture	$\delta \uparrow$
Bircanoglu et al. (DenseNet121) [3]	0.95
Ours (NASNetLarge)	0.955
Ours (DenseNet169)	**0.964**

Table 4. **Our results compared to state-of-the-art accuracy on TrashNet dataset.** We beat state-of-the-art.

References

[1] S. Barles. *History of waste management and the social and cultural representations of waste*, volume 4. Springer, 2014.

[2] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3479–3487, 2015.

[3] C. Bircanoglu, M. Atay, F. Beser, O. Genc, and M. A. Kizrak. Recyclenet: Intelligent waste sorting using deep neural networks. *2018 Innovations in Intelligent Systems and Applications (INISTA)*, pages 1–7, 2018.

[4] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] T. Dozat. Incorporating nesterov momentum into adam. In *ICLR*, 2015.

[7] A. Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[10] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[12] H. Robbins and S. Monro. A stochastic approximation method. In *The Annals of Mathematical Statistics*, 2007.

[13] G. Schwartz and K. Nishino. Recognizing material properties from images. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[14] L. Sharan, R. Rosenholtz, and E. H. Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, vol. 14, no. 9, article 12, 2014.

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[16] G. Thung. Trashnet. `https://github.com/garythung/trashnet`, 2016.

[17] M. Yang and G. Thung. Classification of trash for recyclability status. *Stanford University*, 2016.

[18] H. Zhang, J. Xue, and K. J. Dana. Deep ten: Texture encoding network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2896–2905, 2017.

[19] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018.