Waste Classification using Convolutional Neural Networks

Abstract:

Segregation of garbage at home helps municipalities manage and sort solid waste generated in the country and can have huge positive impacts on the environment. However, this can be a confusing task for many consumers. This paper tries to solve this problem by building a model that helps consumers segregate their trash with ease in the future. To achieve this, we experimented with the state-of-the-art network VGG-16 [1] to implement our own model inspired by it (Model 1), and built another model from scratch (Model 2). We tested 3 regularization techniques - batch normalization [2], dropout [3], and full data augmentation [4] - to find the most efficient model. We found that different techniques work best for the two models and that the Model2 gives better results overall. The VGG-16 inspired model (Model 1) achieved 70.5% accuracy with the three regularization techniques while the Model2 achieved 79.8 % with only data augmentation.

Introduction:

Improper disposal of waste by households is a major problem for waste management municipalities - it makes the task of sorting of waste more difficult and can have severe consequences on the environment and public health. There is a need to automate this process such that it is accessible and helpful for consumers. To this end, we implement 2 models that, given an image of trash as input, classifies it into 6 categories - cardboard, glass, metal, paper, plastic and trash. Our problem statement is a supervised multiclass classification problem. We implemented 2 models, one inspired by VGG-16 and the other from scratch, and experimented with regularisation techniques to find the best performance.

We evaluated our results on the basis of accuracy of our train and test data. We chose this metric because the nature of our problem does not require the calculation of precision or recall, additionally, our dataset is pretty balanced out [5], [6], [7], [8]. This is also supported by our experiments where we found that the F1 score and the accuracy of our model comes out to be very similar.

Background/Related Work:

One of the prevalent works in waste classification has been that of Yang and Thung [9], [10] who collected the TrashNet dataset, one of the most used datasets for this problem, and worked with SVM and AlexNet to classify the data. They were unsuccessful in achieving good results, with an accuracy of 22% by their CNN model compared to 63% by their SVM baseline model. Their work, however, is considered one of the most comprehensive contributions in this problem area. It displays the lack of an exhaustive dataset in academia, which our research also faced.

In [11], He et al. modified the AlexNet model by removing its last two layers, and tried various techniques like adding dropout, different loss and activation functions, Softmax and SVM classifiers, and full and partial augmentation of data. They received an accuracy of about 70% to 80% in their experiments.

The work of Ciresi and Natu [12] tests three types of transfer learning methods using ResNet and MobileNet for classification of data into seven classes using an augmented TrashNet data. Chu et al. [13] worked on a multilayer hybrid deep learning system that leverages a CNN model based on AlexNet that was able to achieve accuracies greater than 90% when tested on a dataset they created on their own.

In their work, Bircanoğlu et al. [14] used transfer learning techniques, attaining accuracies as high as 95% on models like DenseNet121, Inception-v4, and Inception-Resnet. They fabricated their own model RecycleNet that achieved an accuracy of 81% while significantly reducing the parameters by 4 million.

Approach:

The task of our project is to build a multilabel image classifier that segregates the input image into one of the six waste classes. For this, in milestone 1 we explored and experimented with the VGG-16 [1] model to create our own model. We compared it's performance with VGG-16 on the TrashNet dataset and found that our model gave equivalent test accuracy with much lower training time. The sparse categorical cross entropy loss function of our model with softmax as the activation for the last fully connected layer is given as,

$$J(w) = -\frac{1}{N}\sum_{i=i}^{N}[y_i.\log(y'_i)]$$

where,

w refers to the model parameters, e.g. weights of the neural network
y_i is the true label
y'_i is the predicted label

We came up with different results for every iteration of our model by experimenting with numerous regularisation and data augmentation techniques on our base Convolution Neural Network model. For all the models we used Adam optimizer [15] and trained for 50 epochs. We also used a learning rate of 0.0005 with dropout of p=0.6 and 3 batch normalization layers. We assessed our model based on the training and testing accuracy. The model's final architecture is described below:

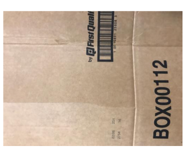
- 2 x convolution layer of 32 channel of 3x3 kernel and same padding and relu activation
- 1 x batch normalisation layer
- 1 x maxpool layer of 2x2 pool size
- 2 x convolution layer of 64 channel of 3x3 kernel and same padding and relu activation
- 1x batch normalisation layer
- 1 x maxpool layer of 2x2 pool size
- 2 x convolution layer of 128 channel of 3x3 kernel and same padding and relu activation
- 1 x maxpool layer of 2x2 pool size
- 1 x Flatten layer
- 1 x Batch normalization layer
- 1 x Dropout Layer with p=0.6
- 1 x Dense layer of 128 units and relu activation
- 1 x Dense Softmax layer of 6 units

The second CNN architecture that we implemented from scratch showed more promising results than model 1. The network used Adam gradient descent optimization along with learning rate scheduling to achieve better accuracy. Data augmentation was also performed where techniques like random brightness change, rotation, scaling, translation and shearing of the images were included. We experimented with various regularisation techniques including dropout and batch normalization, and included the best combination in the final model. The model was trained till 100 epochs and achieved the training accuracy of 90% and validation accuracy of 79.8%. The second model's final architecture is described below:

- 1 x convolution layer of 256 channels of 3x3 kernel and same padding and relu activation
- 1 x maxpool layer of 2x2 pool size
- 1 x batch normalisation layer
- 1 x convolution layer of 128 channels of 3x3 kernel and same padding and relu activation
- 1 x maxpool layer of 2x2 pool size
- 1 x convolution layer of 64 channels of 3x3 kernel and same padding and relu activation
- 1 x maxpool layer of 2x2 pool size
- 1 x convolution layer of 32 channels of 3x3 kernel and same padding and relu activation
- 1 x maxpool layer of 2x2 pool size
- 1 x Flatten layer
- 1 x Dense layer of 64 units and relu activation
- 1 x Dropout Layer with p=0.4
- 1 x Dense layer of 32 units and relu activation
- 1 x Dropout Layer with p=0.4
- 1 x Dense Softmax layer of 6 units

Experiments:

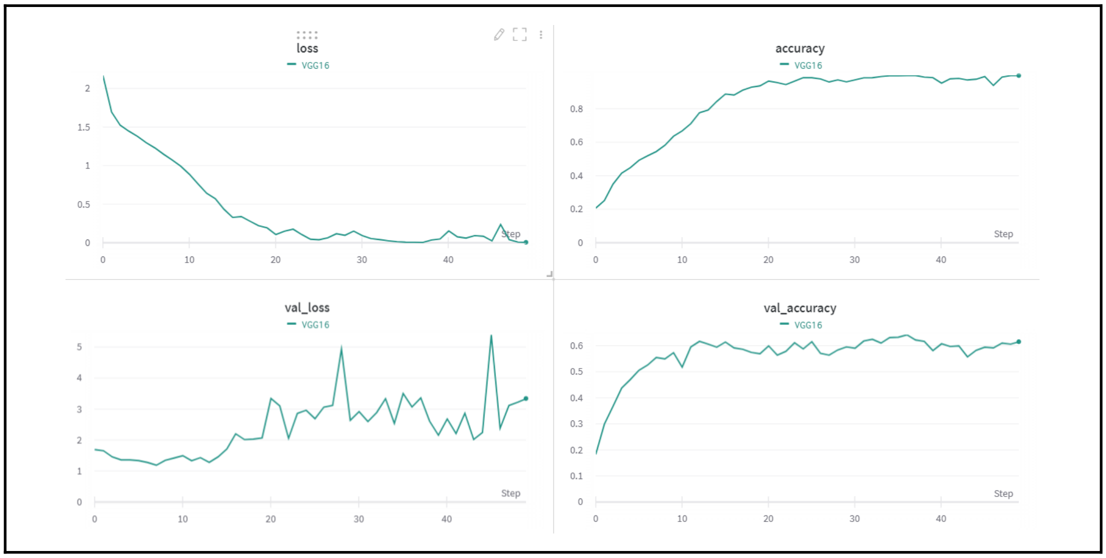
We are using the TrashNet [1] dataset containing approximately 2500 images belonging to 6 classes as shown in the image below. These are all colored images of waste with each image containing one object belonging to one of the 6 classes on a plain white background. We increase the size of the dataset with data augmentation techniques like zooming, varying the brightness, and scaling.

					
cardboard (403)	glass (501)	metal (410)	paper (594)	plastic (482)	trash (137)

Samples from training set of TeshNet dataset(Fig-1)

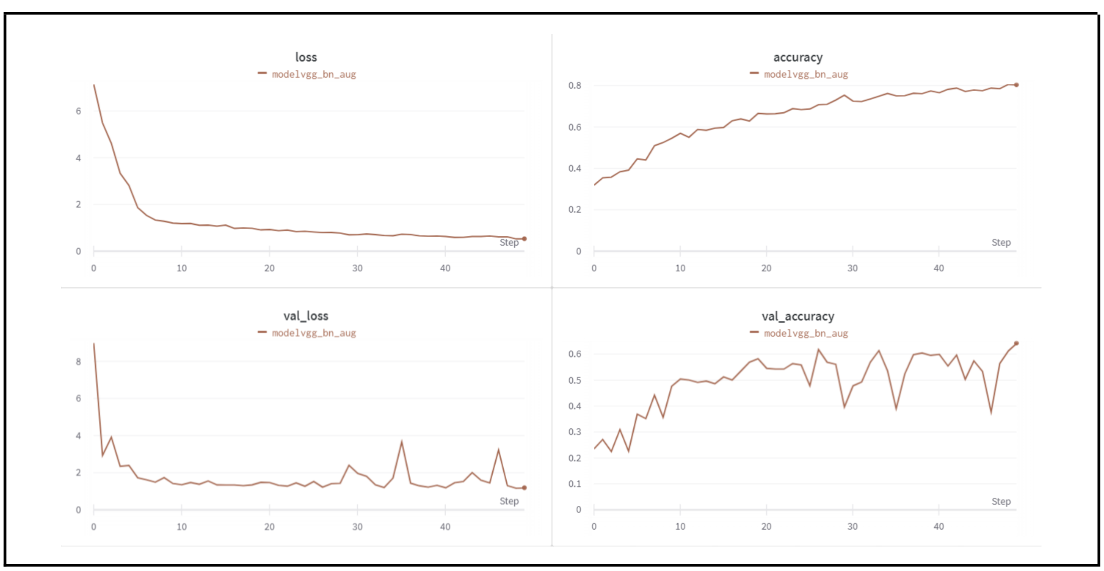
We ran numerous experiments on both our models with different regularization settings. All the iterations of our first model (Model 1) are described below along with their results.

The loss and accuracy graph of VGG-16, our baseline model, is shown below.



Model 1: VGG16 model(Fig-2)

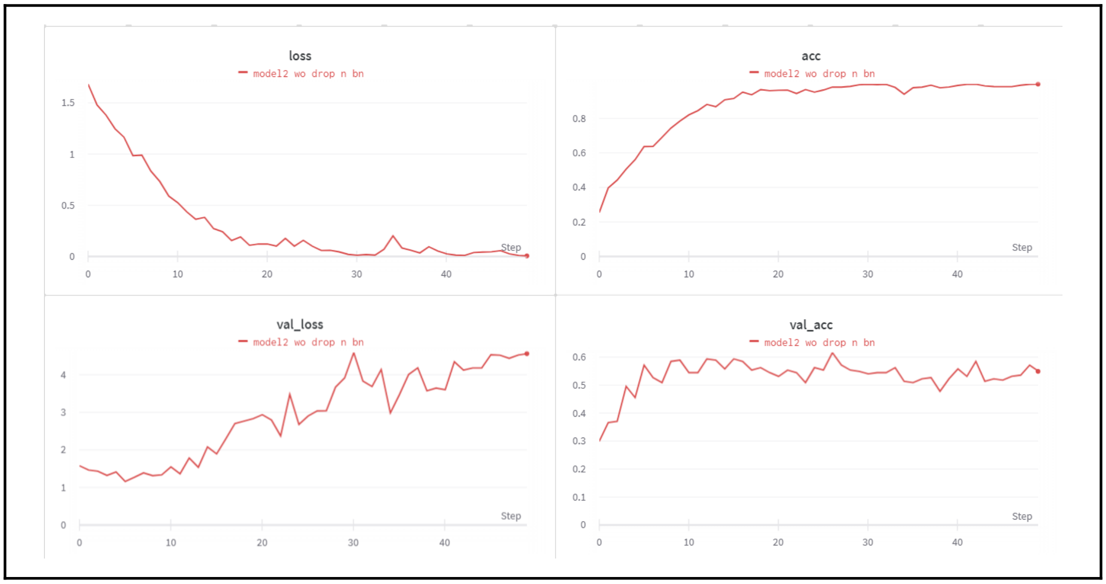
1. Our base model, inspired by the VGG16 model, consists of Convolution, Fully Connected and Max pooling layers without any regularisation or data augmentation techniques. We saw that the model massively overfit to our dataset and performed poorly.
2. In the second iteration of our model, we added the dropout layer with p = 0.6 to help with the overfitting seen in the first iteration above. However, we found that with dropout, our validation losses significantly increased, implying that the model was still overfitting.
3. In this iteration, we added 3 Batch normalization layers in our model. This helped with the performance further and we were able to achieve an accuracy of 68% on the last epoch.
4. In the final iteration, we used data augmentation on both the dropout model and the Batch Normalisation model where we changed the brightness_range (from 0.1 to 0.9), and zoom_range (from 0.5 to 1.5). This gave the best overall performance in terms of both loss and accuracy among all the iterations. The loss and accuracy graph of the model is shown below.



Model 1: Data Augmentation with Batch Normalisation and Dropout(Fig-3)

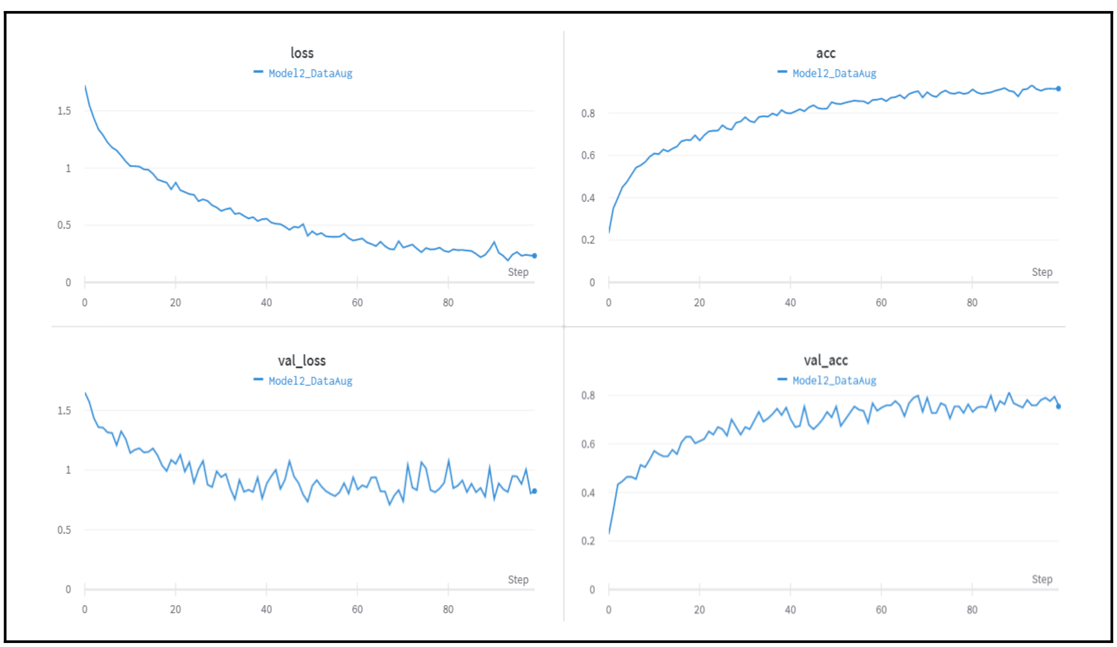
All the iterations of our second model (Model 2) are described below.

1. Our base model consisted of convolution, fully connected and max pooling layers without any data augmentation and regularisation techniques. The results are shown below.



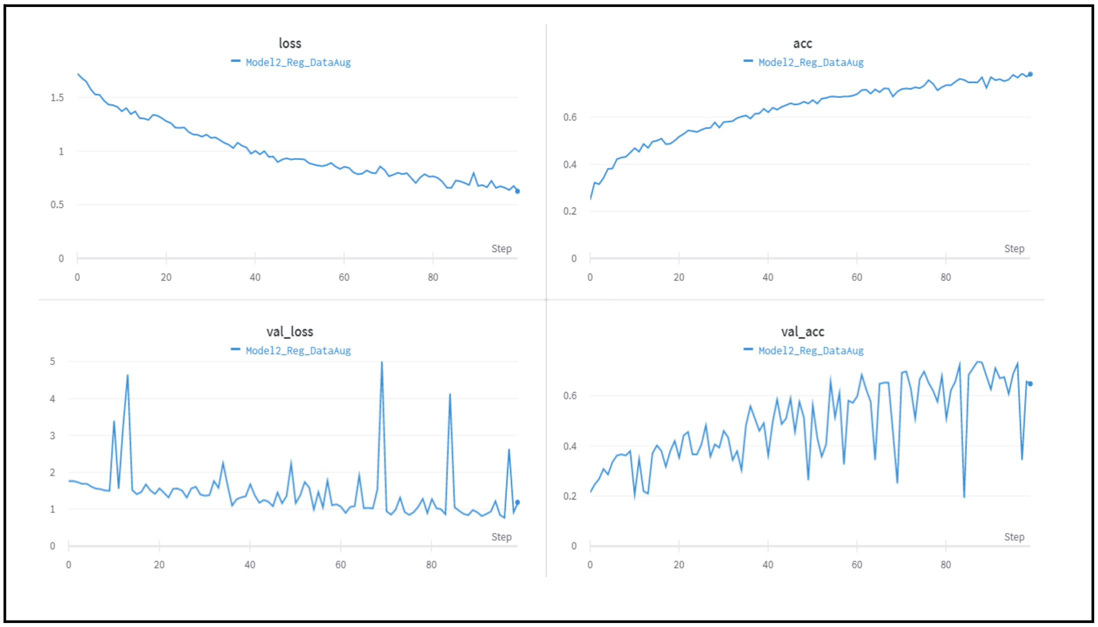
Model 2: Base Model(Fig-4)

2. The second iteration included 1 batch normalization and 2 dropout layers (p = 0.4). The model did not perform any better than the base model as the dataset was not big enough for the model to learn relevant patterns in the garbage images. Hence, we decided to include data augmentation in the next iteration.
3. In the third iteration data augmentation techniques like random brightness change, rotation, scaling, translation and shearing of the images were included. This model performed the best and achieved the training accuracy of 90% with validation accuracy of 79.8%.



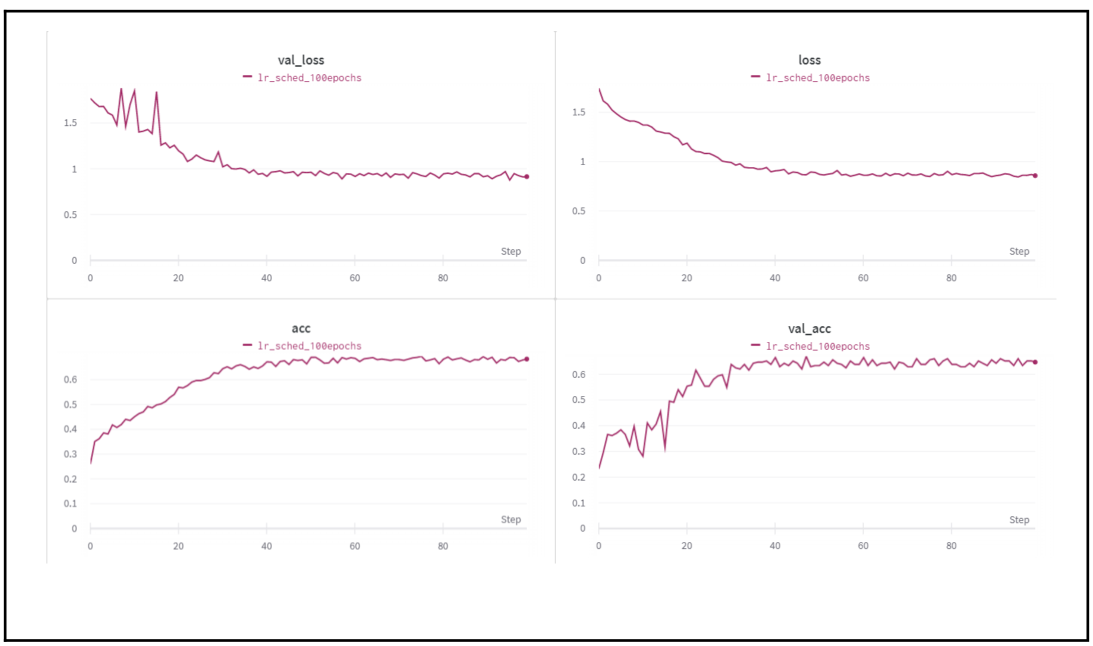
Model 2 with Data Augmentation(Fig-5)

4. In this iteration we included both the data augmentation and regularisation techniques. Although this model did fairly well, adding regularisation slightly worsened the results achieved in the former architecture.



Model 2 with Data Augmentation and Regularisation(Fig-6)

5. In the last iteration, we applied learning rate scheduling, a predefined framework which adjusts the learning rate as the training progresses. During the initial training, the learning rate is set to a large number to reach a good enough set of weights, and gradually decreases to fine tune our model into giving a good accuracy.



Model 2 with Learning Rate Scheduler(Fig-7)

All the models described above have been summarized in the table below.

Table 1: Results of Model 1

Model Description	Train Accuracy (%)	Validation Accuracy (%)	Epochs
VGG16	99.83	63.11	50
Model 1	81.33	61.5	50
Model 1 + dropout	98.5	65.14	50
Model 1 + dropout + BN	97.85	68.38	50
Model 1 + dropout + data aug	**90.55**	**70.5**	**50**
Model 1 + dropout + BN + data augmentation	80.37	64.3	50

Table 2: Results of Model 2

Model Description	Train Accuracy (%)	Validation Accuracy (%)	Epochs
Model 2	99.81	55.49	50
Model 2 + dropout + BN	95.52	49.55	50
Model2 + data augmentation	**90**	**79.8**	**100**
Model 2 + dropout + BN + data augmentation	78	70	100
Model 2 + LR scheduling	68.3	66.73	100

Conclusion and Future Ideas:

In this paper, we explored many techniques and models and came up with a VGG-16 inspired, and a novel architecture to build a multilabel image classifier for waste classification into 6 categories. With our best CNN model, we managed to reach a stable accuracy of 70% to 80% with data augmentation, adam optimizer and learning rate of 0.0001. After reviewing a lot of papers, we realised that any model with the accuracy above 90% utilized transfer learning techniques.

In future, we hope to explore and experiment with other models, such as ResNet and Inception, and utilise their techniques to build an architecture that achieves better results. Along with that, we also wish to upgrade the dataset by including more classes and images per class which would help improve the overall performance and generalization capabilities of the model.

References:

[1] Simonyan Karen and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." 2014, arXiv preprint arXiv:1409.1556.

[2] Sergey Ioffe and Christian Szegedy. “Batch normalization: accelerating deep network training by reducing internal covariate shift.” 2015, In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'2015). JMLR.org, 448–456.

[3] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: a simple way to prevent neural networks from overfitting”. J. Mach. Learn. Res. 15, *1* (January 2014), 1929–1958.

[4] Nanni, Loris, Michelangelo Paci, Sheryl Brahnam, and Alessandra Lumini. "Comparison of Different Image Data Augmentation Approaches". 2021, Journal of Imaging 7, no. 12: 254, https://doi.org/10.3390/jimaging7120254

[5] Padilla Rafael, Netto Sergio, da Silva Eduardo. “A Survey on Performance Metrics for Object-Detection Algorithms”. 2020, International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 237-242, doi: 10.1109/IWSSIP48289.2020.9145130.

[6] C. Ferri, J. Hernández-Orallo, R. Modroiu. “An experimental comparison of performance measures for classification”. 2009, Pattern Recognition Letters, Volume 30, Issue 1, Pages 27-38, ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2008.08.010.

[7] Yanminsun, Wong Andrew, Kamel Mohamed S. “Classification of imbalanced data: a review”. 2011, International Journal of Pattern Recognition and Artificial Intelligence. 23. 10.1142/S0218001409007326.

[8] Branco Paula, Torgo Luís, Ribeiro Rita. “A Survey of Predictive Modelling under Imbalanced Distributions”. 2015.

[9] G. Thung and M. Yang. "Dataset of images of trash; Torch-based CNN for garbage image classification". GitHub Repository, 2016, https://github.com/garythung/trashnet.

[10] Y. He, Q. Gu and M. Shi. “Trash Classification Using Convolutional Neural Networks Project Category: Computer Vision”. CS230 Project Report, Spring 2020, https://cs230.stanford.edu/projects_spring_2020/reports/38847029.pdf.

[11] Yang, Mindy, and Gary Thung. "Classification of trash for recyclability status." CS229 Project Report 2016 (2016).

[12] S. Ciresi and A. Natu. “Sustainable Waste Classification using Deep Convolutional Networks”. CS230 Project Report, Winter 2018, https://cs230.stanford.edu/projects_spring_2019/reports/18678247.pdf.

[13] Chu, Ying-Chien, Chen Huang, Xiaodan Xie, Bohai Tan, Shyam Kamal and Xiaogang Xiong. “Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling”. 2018, Computational Intelligence and Neuroscience.

[14] C. Bircanoğlu, M. Atay, F. Beşer, Ö. Genç and M. A. Kızrak, "RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks". 2018, Innovations in Intelligent Systems and Applications (INISTA), 2018, pp. 1-7, doi: 10.1109/INISTA.2018.8466276.

[15] Kingma, Diederik & Ba, Jimmy. “Adam: A Method for Stochastic Optimization. International Conference on Learning Representations”. 2014.