GARBAGE CLASSIFICATION BASED ON A CASCADE NEURAL NETWORK

X. Zhang[*], *N. Zhao*[†], *Q. Lv*[*], *Z. Ma*[‡], *Q. Qin*[*], *W. Gan*[§], *J. Bai*[¶], *L. Gan*[‖]

Abstract: Most existing methods of garbage classification utilize transfer learning to acquire acceptable performance. They focus on some smaller categories. For example, the number of the dataset is small or the number of categories is few. However, they are hardly implemented on small devices, such as a smart phone or a Raspberry Pi, because of the huge number of parameters. Moreover, those approaches have insufficient generalization capability. Based on the aforementioned reasons, a promising cascade approach is proposed. It has better performance than transfer learning in classifying garbage in a large scale. In addition, it requires less parameters and training time. So it is more suitable to a potential application, such as deployment on a small device. Several commonly used backbones of convolutional neural networks are investigated in this study. Two different tasks, that is, the target domain being the same as the source domain and the former being different from the latter, are conducted besides. Results indicate with ResNet101 as the backbone, our algorithm outperforms other existing approaches. The innovation is that, as far as we know, this study is the first work combining a pre-trained convolutional neural network as a feature extractor with extreme learning machine to classify garbage. Furthermore, the training time and the number of trainable parameters is significantly shorter and less, respectively.

Key words: *garbage classification, transfer learning, extreme learning machine, convolutional neural networks*

Received: February 5, 2022
Revised and accepted: April 30, 2023

DOI: 10.14311/NNW.2023.33.007

[*]Xiliang Zhang; Qinyuan Lv; Qin Qin – Corresponding author; School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University. 201209, Shanghai, China, E-mail: xlzhang@sspu.edu.cn, lvqinyuan0301@163.com, qinqin@sspu.edu.cn

[†]Na Zhao; School of Resources and Environmental Engineering. Shanghai Polytechnic University. E-mail: 20191510068@stu.sspu.edu.cn

[‡]Zhenyu Ma; Department of Electronic, Electrical and Systems Engineering, University of Birmingham, Birmingham, UK, E-mail: ZXM070@student.bham.ac.uk

[§]Weifei Gan – Corresponding author; School of Computer and Information Engineering, Shanghai Polytechnic University, 201209, Shanghai, China, E-mail: wfgan@sspu.edu.cn

[¶]Jianfeng Bai; Institute of Resource Recycling Science and Engineering, Shanghai Polytechnic University, 201209, Shanghai, China and Shanghai Collaborative Innovation Centre for WEEE Recycling, WEEE Research Centre of Shanghai Polytech-nic University Recycling Science and Engineering, Shanghai Polytechnic University, 201209, Shanghai, China, E-mail: jfbai@sspu.edu.cn

[‖]Ling Gan; Yu Chen Intelligent Technology (Shanghai) Co., Ltd., Shanghai, China, E-mail: 1141663856@qq.com

1. Introduction

Environmental problems, especially producing more garbage, are increasingly prominent in people's daily life, manufacturing industry, and so on. Traditional garbage classification costs a huge amount of labor and time. Accordingly, it has become a considerably severe requirement that how to effectively classify and further dispose of garbage.

Artificial intelligence, especially deep learning [2,9,20], has developed rapidly in recent years. In image recognition, classification, and other tasks, the accuracy of deep learning has far exceeded human beings. Garbage classification based on images can also be regarded as a combination of image classification and recognition. For example, there are many targets in a picture. Garbage need to be recognized in this picture before it is classified. Therefore, deep learning can be employed to guide garbage classification, which will save labor and time.

Recently, most studies on intelligent garbage classification are based on deep learning, and the accuracy is high. For example, Ozkaya and Seyfi focused on recyclable garbage classification. They utilized transfer learning with fine-tuning to classify 2,527 images into six classes and the accuracy reached 97.86% [11]. Feng and Tang collected 2,313 different office garbage pictures and classified them by transfer learning based on InceptionV3. The average accuracy of the experimental results reached 95.33% [1]. However, most studies focus on small categories, and some of them have even not provided datasets. Yuan et al. changed the original four types of garbage into five, without providing any specific classification details. Furthermore, the total amount of training data was only 2,000, and no dataset was attached [17]. Yang and Thung use a small dataset, and this dataset only has about 3,000 images which are divided into six categories [15]. Yang et al. proposed an approach based entirely on deep learning [23]. They also deployed the method in raspberry terminal. However, this method used a large number of parameters and spent a lot of time and data to train the network.

Extreme learning machine (ELM) [5] has some advantages, compared with those traditional machine learning techniques or other classifiers, such as semisupervised particle swarm optimization [22]. ELM provides better generalization performance. Moreover, ELM has lower computational complexity and higher learning efficiency. Therefore, it is a suitable choice to be a classifier.

In this study, a novel cascade neural network is proposed to reduce both the training time and the number of trainable parameters with better performance. Several commonly used convolutional neural networks (CNNs) are investigated and adopted as a feature extractor. ELM is also employed as a classifier. The advantages of our algorithm can be listed as follows: Firstly, the training time and the number of trainable parameters is obviously shorter and less than conventional neural networks, respectively. Secondly, our algorithm costs less because of no fine-tuning, whereas it has better performance.

The rest of this study is organized as follows: Related work is given in Section 2. Our algorithm and results are illustrated in Section 3 and Section 4, respectively. Finally, conclusions are drawn in Section 5.

2. Related Work

In this section several significant algorithms will be reviewed, such as CNNs, transfer learning, ELM, and most existing approaches of garbage classification.

AlexNet [8] was first proposed by Krizhevsky et al. in 2012, and won the championship of ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) [13]. As a result, it attracted more extensive attention on deep learning. It is mainly composed of five convolutional layers, some of which are followed by a maxpooling layer, three fully connected layers and the final classification layer [8]. It used ReLU [10] as the activation function instead of sigmoid to obtain better performance and utilized dropout [4] to alleviate overfitting. It employed overlapping max pooling to avoid the blurring effect of average pooling and adopted two GPUs to make training faster. However, both the receptive field and the number of trainable parameters is larger.

Simonyan and Zisserman proposed VGG at ImageNet Large Scale Visual Recognition Challenge 2014 [14]. With the increase in the depth of convolutional neural networks, the accuracy in large scale image recognition also improves. They all adopted the architecture of a 3×3 filter, and achieved good results in ILSVRC location and classification. Moreover, VGG also performs well on other datasets, indicating its strong generalization ability. Nevertheless, the depth of VGG network is up to 19 layers. If the depth is extended, the problem of vanishing/exploding gradients might occur.

He et al. proposed residual network (ResNet) in the ImageNet Large Scale Visual Recognition Competition 2015 [3], and won the first place in ImageNet detection, ImageNet positioning, COCO detection and COCO segmentation. ResNet has provided a wholly new direction in image recognition. The structure of ResNet is based on the classical convolutional backbone adding a shortcut connection, which solves the problem of vanishing/exploding gradients with the increase of the network depth. Most subsequent neural networks use ResNet as a backbone.

Transfer learning refers to the successful transfer of knowledge from the source domain to the target domain, so that the classification remains accurate and effective on the new target domain. The advantage of this approach is that only a small amount of labelled data in target domain is needed to train (fine-tune) parameters of the pre-trained network, rather than training a new network from scratch with a large amount of labelled data in target domain. However, feature spaces of the source domain and the target domain are possibly completely different, which will directly lead to the failure of transfer learning [12].

ELM is a fast learning method based on the construction of a single-hidden layer feedforward neural network (SLFN). It was proposed by Huang et al, in 2004 [5]. For the hidden layer, weights are randomly initialized by ELM, while for the output layer, weights are calculated. ELM has the advantages of greater convenience, faster learning speed, stronger generalization ability and so on. However, it suffers from unstable due to the random initialization.

Ozkaya and Seyfi conducted the recyclable garbage classification with transfer learning. 2,527 images were collected, and divided into six different classes. Specifically, several classical deep learning backbones, such as AlexNet, Googlenet, etc, and classifiers, for instance, support vector machines (SVMs [18]), etc, are com-

pared. Results showed that Googlenet+SVM outperformed others and the classification accuracy reached 97.86% [11]. However, the number of the dataset was small and they focused on small categories. In addition, the algorithm is not suitable for being arranged in hardware because of a huge amount of parameters and training time. Zhang et al. proposed a variation of ResNet18 to classify garbage with the same dataset [21]. Yang and Li proposed a new lightweight neural network garbage classification structure, WasNet. The network has 1.5 million parameters which is a half of mainstream neural networks, and the accuracy is 82.5% when tested on Huawei's dataset. In addition, they ported the model to a hardware platform and assembled a smart trash can [16]. However, it also suffers from small categories. More details can refer to [19].

Based on those drawbacks, a cascade neural network is proposed with less trainable parameters and training time achieving better performance.

3. Methods

In this section, the main idea of our algorithm is stated.

The proposed method consists of two blocks, that is, feature extraction and classification.

Features extracted from CNNs are highly believable and can be further analyzed, since conventional CNNs make a positive contribution to many tasks, such as image recognition, classification, etc. Therefore, a pre-trained CNN is employed as a backbone to extract features of garbage in the feature extraction block. Those features can be considered as extra information which will improve the accuracy as shown in Eq. (1) in the classification block.

$$\max \mathrm{P}(\mathbf{y}|\mathbf{x}) = \max \prod_{i=1}^{N} \mathrm{P}(y_i|x_i), \tag{1}$$

where $\mathrm{P}(\mathbf{y}|\mathbf{x})$ is the probability of $\mathbf{y}$ given $\mathbf{x}$; N is the total number of data; $\mathbf{y}$ and $\mathbf{x}$ are output and extra information extracted from the convolution block, respectively.

Remark: No matter what the classification block is, Eq. (1) maximizes the (posterior) likelihood. In theory, the expectation of the error is always equal to zero, providing $\mathbf{x}$ is given as extra information, such as features from the convolution block. While other commonly used algorithms in machine learning or system identification, etc, only maximize the (prior) likelihood without given extra information and the expectation of the error is obviously not equal to 0. More details can refer to Appendix.

With a fast classifier (ELM) instead of fully connected layers, both the training time and the number of trainable weights obviously reduce. For instance, taking AlexNet as an example, fully connected layers have a total of 58,631,144 $(4,096 \times 9,216 + 4,096 \times 1 + 4,096 \times 4,096 + 4,096 \times 1 + 1,000 \times 4,096 + 1,000)$ trainable parameters. While the proposed ELM has only 18,436,000 $(2,000 \times 9,216 + 2,000 \times 1 + 2,000 \times 1)$ parameters, where only 2,000 parameters need to be trained.

Specifically, ELM minimizes the mean squared error as shown in Eq. (2).

$$\min_{\beta} \|\mathbf{T} - \mathbf{H}\beta\|^2, \tag{2}$$

where β is the output weights; $\mathbf{T}$ and $\mathbf{H} = g(\mathbf{A}\mathbf{x} + \mathbf{b})$ are output and the hidden layer output matrix, respectively; g is the activation function; $\mathbf{A}$ and $\mathbf{b}$ are randomly generated weights, respectively, and do not need to be adjusted or updated by training; $\|\cdot\|^2$ is the Frobenius norm.

Note that the solution of Eq. (2) can be equivalent to Eq. (3) without any searching methods but simple calculations.

$$\beta^* = \mathbf{H}^{\dagger}\mathbf{T}, \tag{3}$$

where β^* is the solution; $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of $\mathbf{H}$.

Accordingly, the training process just involves calculation, which can be conducted in a CPU rather than a GPU. In addition, the number of trainable weights which are simply acquired by Eq. (3), is obviously less than transfer learning or multilayer perceptron.

The structure of a classical neural network (AlexNet) illustrated in Fig. 1 is compared with the structure of our algorithm shown in Fig. 2.

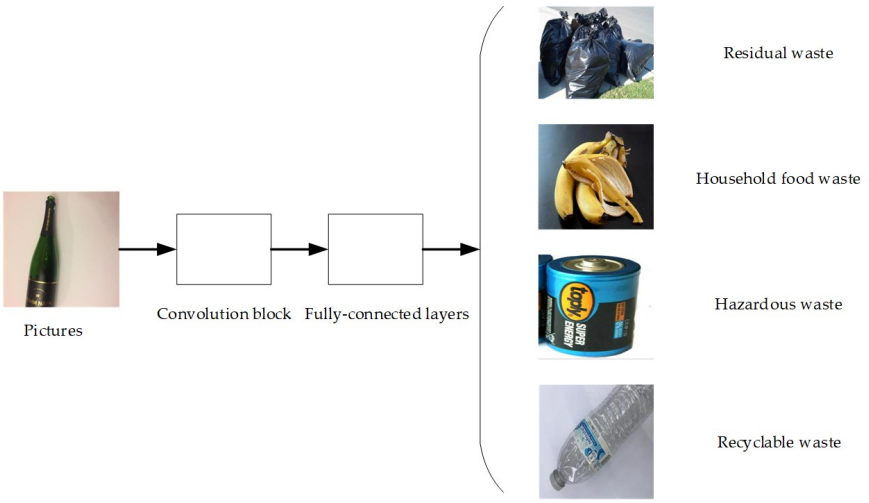

Fig. 1 *Classical networks with transfer learning.*

Fig. 1 is the classical structure of a CNN (AlexNet). There are two parts: One is the convolution block which consists of all convolution and pooling layers; the other is the classification block, or more specifically the fully connected block which comprises all fully connected layers and the classification layer. Transfer learning is the process that parameters of the CNN are retrained with only a few pieces of data after changing the number of classes. Garbage can be recognized by this CNN with transfer learning. The training process is as follows: First, AlexNet is loaded. Then the number of classes is changed. Providing the training data, the AlexNet is retrained to fine-tune weights. The whole process is done with a GPU.

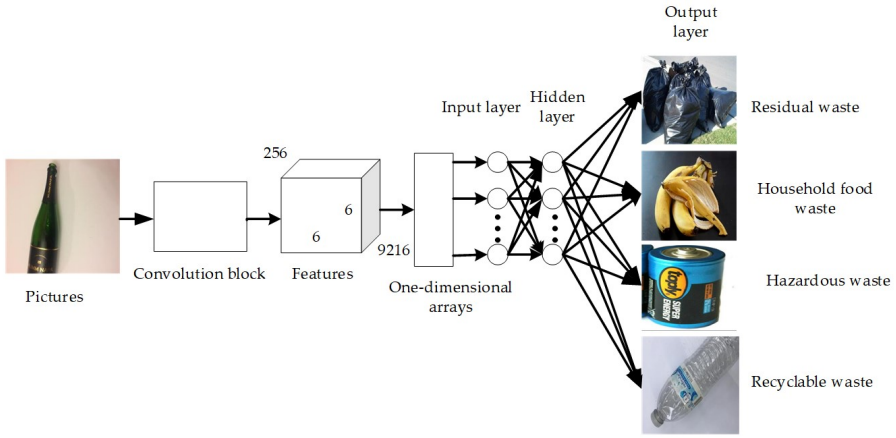

Fig. 2 *The structure of our algorithm.*

Fig. 2 is the structure of this cascade network with the same convolution block but an ELM classifier. The advantage is that our algorithm does not need fine-tuning. The training process of ours is described as follows: AlexNet is used to extract features of the garbage dataset. Then features are reshaped into one-dimensional arrays, and inputted to ELM. After that, weights of output are computed. Accordingly, the training time and the number of trainable weights is distinctly less than transfer learning. In addition, our algorithm is a lightweight neural network, which makes it possible for us to run our algorithm on small devices. However, due to the random input matrix of ELM, the result is not stable (sometimes better but sometimes worse), but still acceptable.

By comparison, there are a number of important differences between transfer learning and our algorithm. First, the former needs CNNs to be retrained with new data, whereas the latter only load the pre-trained backbone without fine-tuning. Second, the former needs at least a GPU to accelerate the training process, while the latter only requires a CPU. Finally, both the training time and the number of weights of the former are longer and greater than that of the latter.

Generally, the construction, training, and testing processes of our cascade network are presented in Algorithm 1, Algorithm 2, and Algorithm 3, respectively.

Algorithm 1 The construction process of the proposed cascade neural network.

Load a pre-trained neural network.
Extract the feature extraction block of the pre-trained network as the feature extraction block of our cascade network.
ELM is constructed as the classification block according to the number of hidden nodes (the only hyperparameter of ELM).
The feature extraction block and the classification block are cascaded together to construct the proposed cascade neural network.

Algorithm 2 The training process of the proposed cascade neural network.

Take all the images of the training dataset as input and put them into the feature extraction block of the proposed network.
Obtain the corresponding features and reshape them into one-dimensional arrays $\mathbf{x}$.
Construct the label matrix $\mathbf{T}$.
Randomly generate weights $\mathbf{A}$ and $\mathbf{b}$, and choose an activation function g.
Calculate the hidden layer output matrix $\mathbf{H}$ and the trainable parameters β^* according to Eq. (3).

Algorithm 3 The testing process of the proposed cascade neural network.

Take all the images of the testing dataset as input and put them into the feature extraction block of the proposed network.
Obtain the corresponding features and reshape them into one-dimensional arrays.
Construct the label matrix.
Calculate the hidden layer output matrix and the output matrix.
Compare and print the accuracy.

4. Results

All experiments are conducted in MATLAB 2020b environment on a PC running on Windows 10 operating system with 64 bits, 1.70 GHz Intel®Xeon®CPU E5−2603 and 32 GB of RAM. All experiments of transfer learning are trained on NVIDIA GeForce GTX 1080 Ti GPU.

Most garbage images in our dataset are from Huawei's open source garbage classification dataset [6]. Others are collected by ourselves. There are 16,440 pictures in total.

The dataset is divided into two different training and testing pairs (with ratio 7:3) aiming to two tasks. One is for the task that the source domain (training) is different from the target domain (testing) and the other is for the task that two domains are the same. For instance, in the former task, medicine is in the source domain, while batteries are in the target domain. Contrastively, both appear in the source and target domain in the latter task. Both tasks have four categories, i.e., residual waste, household food waste, recyclable waste, and hazardous waste.

The hyperparameters of transfer learning are given as follows. The learning rate is 0.0001, the maximum of training epochs is 100, and the number of minibatch is 32. The ADAM [7] is adopted, GradientDecayFactor and SquaredGradientDecayFactor are set to 0.5 and 0.99, respectively.

The only hyperparameter of ELM is the number of hidden nodes. An experiment is conducted to acquire the suitable one as shown in Tab. I. The first column lists backbones of classical deep neural networks which are used as a feature extractor to obtain feature maps as input for ELM. The top row is the number of hidden nodes. Results in Tab. I are the testing accuracy which is an average of three times. A conclusion can be drawn that when the number of hidden nodes is near 2,000, ELM has the "best" performance for all feature extractors, i.e., the classical CNN backbones. Therefore, in the following experiments, the number of hidden nodes is set to 2,000 as default.

	10,000 [%]	9,216 [%]	7,000 [%]	5,000 [%]	3,000 [%]	2,000 [%]	1,000 [%]
AlexNet	59.98	59.64	66.33	71.86	76.66	**76.81**	75.99
VGG19	64.34	64.95	72.73	78.37	80.73	**81.26**	79.56
ResNet18	80.75	81.13	82.05	84.79	86.44	**86.65**	85.91
ResNet101	80.16	80.40	83.62	87.23	88.43	**88.51**	87.46

Tab. I *The comparison between different hyperparameters. The top row is the number of hidden nodes.*

Notation in both Tabs. II and III: ♣: TL-test accuracy refers to the test accuracy with transfer learning. ♢: ELM-test accuracy is the test accuracy using the ELM with a feature extractor of a pre-trained deep neural network. ♡: TL-time is the training time of transfer learning in minutes. ♠: ELM-time is the training time of ELM in minutes.

4.1 Task 1: The source domain is the same as the target domain

Results are shown in Tab. II.

As shown in Tab. II, most comparison results between transfer learning and our cascade neural network are similar, except for VGG19, where the latter obviously outperforms the former. The best performance model is ResNet101 as a backbone with ELM as a classifier. The classification accuracy is 88.51% (better than WasNet). The best training time model is ResNet18 with ELM, whose training time is 0.27 minutes which is slightly shorter than that of the best performance model (ResNet101 + ELM) (0.30 minutes). Considering the difference in computational complexity and the depth of two models, such a tiny difference in training time can be ignored. Consequently, the best model is ResNet101 + ELM, whose classification accuracy is higher than transfer learning. Besides, it is noticeable that the training time of ours visibly decreases. Moreover, calculations are computed on the CPU only. While transfer learning is usually computed on a single GPU. Actually, transfer learning can also be computed on a CPU, the cost of time is several times longer nevertheless. In addition, deep learning usually converts the computation into a graph that only contains addition and multiplication, which is more suitable for a GPU instead of a CPU that usually aims at more complex tasks.

	♣ [%]	♢ [%]	♡ [min]	♠ [min]
AlexNet	79.22	76.81	333.46	0.43
VGG19	72.10	81.26	1130.8	0.71
ResNet18	87.81	86.65	363.23	**0.27**
ResNet101	88.19	**88.51**	1787.53	0.30

Tab. II *The comparison of the accuracy and running time (training).*

4.2 Task 2: The source domain is different from the target domain

Results are shown in Tab. III.

As shown in Tab. III, the best performance model is still ResNet101 + ELM. The classification accuracy is 69.97%. The best training time model is ResNet18 with ELM, whose training time is 0.29 minutes which is slightly shorter than that of the best performance model (ResNet101 + ELM) (0.31 minutes). Accordingly, the same conclusion can be drawn that the cascade neural network (ResNet101 + ELM) surpasses transfer learning. Compared with results from Tab. II, all performance is worse, possibly because of the difference between the source domain and the target domain. However, it also indicates that the robustness of our cascade neural network is comparable to that of transfer learning.

Results in both Tabs. II and III also show that VGG19 as a backbone with ELM is significantly better than transfer learning. It indicates that the structure of VGG is more suitable to extract features. In addition, it also implies that better performance might be acquired providing change all filters into 3×3 ones for the backbone of ResNet.

	♣ [%]	♢ [%]	♡ [min]	♠ [min]
AlexNet	69.14	64.61	251.68	0.43
VGG19	61.50	67.36	1278.90	0.69
ResNet18	65.04	67.84	244.06	**0.29**
ResNet101	68.74	**69.97**	1687.13	0.31

Tab. III *The comparison of the accuracy and running time (training).*

5. Conclusion

In this paper, we propose a cascade neural network, which works on a CPU only. In addition, the training time and the number of trainable parameters is obviously shorter and less, which indicates that our algorithm is more likely to be implemented in small devices. So, a conclusion that whether the source domain is different from the target domain or not, ResNet101 backbone combined with ELM has the best performance and efficiency can be drawn. Moreover, providing change filters into 3×3 ones, the performance may be improved. The limitation is that the proposed algorithm cannot classify the mixed garbage. This is also our future work. For instance, we will segment the mixed garbage figure with some other networks such as the semantic segmentation algorithms. Other feature extractors, such as autoencoders, will be also considered in future work. Also recurrent neural networks (RNNs) might be used in future video garbage detection.

Acknowledgement

This research was funded by National Key R&D Program of China (2017YFE0118700, 2019YFC1906100), EU H2020 FIRST project (Grand No.734599, FIR ST:vF In-

teroperation supporting business innovation), and Shanghai Intelligent Manufacturing Collaborative Logistics Equipment Engineering Technology Research Center (A10GY21H004-18).

We thank Wenjun Gu for assisting us with designing an APP to collect data and building a repository to store and label data, which are beyond the scope of this paper. We also thank Yuxin Zhong for testing adversarial attacks for our algorithm which is also beyond the scope of this paper.

References

[1] FENG J., TANG X. Office garbage intelligent classification based on inception-v3 transfer learning model. *Journal of Physics: Conference Series*, 2020, 1487(1), 012008, doi: 10.1088/1742-6596/1487/1/012008.

[2] GOODFELLOW I., BENGIO Y., COURVILLEE A. *Deep learning*. MIT press. 2016, Available from: https://books.google.com.hk/books?hl=zh-CN&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&dq=deep+learning&ots=MNQ1isoAPR&sig=hCiIRx14q4dBNr2LPSMqT79hKyI&redir_esc=y&hl=zh-CN&sourceid=cndr#v=onepage&q=deep%20learning&f=false

[3] HE K., ZHANG X., REN S., SUN J. Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778, doi: 10.1109/cvpr.2016.90.

[4] HINTON G., SRIVASTAVA N., SUTSKEVER I., SALAKHUTDINOV R. Improving neural networks by preventing co-adaptation of feature detectors. *Computer Science*. 2012, 3(4), pp. 212–223, doi: 10.9774/GLEAF.978-1-909493-38-4_2.

[5] HUANG G.B., WANG D.H., LAN Y. Extreme learning machines: a survey. *International journal of machine learning and cybernetics*. 2011, 2(2), pp. 107–122, doi: 10.1007/s13042-011-0019-y.

[6] KAZA S., YAO L., BHADA-TATA P., WOERDEN F. What a Waste 2.0. *World Bank Publications*. 2018. Available from: https://ideas.repec.org/b/wbk/wbpubs/30317.html

[7] KINGMA D., BA J.A. A Method for Stochastic Optimization. In: *International Conference on Learning Representation*, 2015. Available from: https://arxiv.org/pdf/1412.6980.pdf.

[8] KRIZHEVSKY A., SUTSKEVER I., HINTON G. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*. 2017, 60(6), pp. 84–90, doi: 10.1145/3065386.

[9] LECUN Y., BENGIO Y., HINTON G. Deep learning. *Nature*. 2015, 521(7553), pp. 436–444, doi: 10.1038/nature14539.

[10] NAIR V., HINTON G. Rectified Linear Units Improve Restricted Boltzmann Machines. In: *International Conference on Machine Learning*, 2010, pp. 807–814. Available from: https://icml.cc/Conferences/2010/papers/432.pdf.

[11] OZKAYA U.,SEYFI L. Fine-Tuning Models Comparisons on Garbage Classification for Recyclability. In: *2nd International Symposium on Innovative Approaches in Scientific Studies*, 2018. Available from: https://arxiv.org/ftp/arxiv/papers/1908/1908.04393.pdf.

[12] PAN S., YANG Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*. 2010, 22(10), pp. 1345–1359, doi: 10.1109/TKDE.2009.191.

[13] RUSSAKOVSKY O., DENG J., SU H., KRAUSE J., SATHEESH S., MA S., HUANG Z., KARPATHY A., KHOSLA A., BERNSTEIN M., BERG A., LI F. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*. 2014, pp. 1–42, doi: 10.1007/s11263-015-0816-y.

[14] SIMONYAN K., ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *International Conference on Learning Representation*, 2015. Available from: https://arxiv.org/pdf/1409.1556.pdf.

[15] YANG M., THUNG G. Classification of trash for recyclability status. *Project Report*. 2016. Research report. Available from: http://cs229.stanford.edu/proj2016/report/ThungYang-ClassificationOfTrashForRecyclabilityStatus-report.pdf

[16] YANG Z., LI D. WasNet: A Neural Network-based Garbage Collection Management System. *IEEE Access*. 2020, 8, pp. 103984–103993, doi: 10.1109/ACCESS.2020.2999678.

[17] YUAN J., NAN X., LI C., SUN L. Research on Real-Time Multiple Single Garbage Classification Based on Convolutional Neural Network. *Mathematical Problems in Engineering*, 2020, doi: 10.1155/2020/5795976.

[18] VENKATESAN C., KARTHIGAIKUMAR P., PAUL A., SATHEESKUMARAN S., KUMAR R. ECG signal preprocessing and SVM classifier-based abnormality detection in remote healthcare applications. *IEEE Access*. 2018, 6, pp. 9767–9773, doi: 10.1109/ACCESS.2018.2794346.

[19] Abdu H., Noor M. A Survey on Waste Detection and Classification Using Deep Learning. *IEEE Access*. 2022, 10, pp. 128151–128165, doi: 10.1109/ACCESS.2022.3226682.

[20] NAQVI R., ARSALAN M., REHMAN A., REHMAN A., LOH W., PAUL A. Deep learning-based drivers emotion classification system in time series data for remote applications. *Remote Sensing*. 2020, 12(3), pp. 587, doi: 10.3390/rs12030587.

[21] ZHANG Q., ZHANG X., MU X., WANG Z., TIAN R., WANG X., LIU X. Recyclable waste image recognition based on deep learning. *Resources, Conservation and Recycling*. 2021, 171, pp. 105636, doi: 10.1016/j.resconrec.2021.105636.

[22] ZHANG X., JIAO L., PAUL A., YUAN Y., WEI Z., SONG Q. Semisupervised particle swarm optimization for classification. *Mathematical Problems in Engineering*. 2014, doi: 10.1155/2014/832135.

[23] YANG Z., BAO Y., LIU Y., ZHAO Q., ZHENG H., BAO Y. Research on deep learning garbage classification system based on fusion of image classification and object detection classification. *Mathematical Biosciences and Engineering*. 2023, 20(3), pp. 4741–4759, doi: 10.3934/mbe.2023219.

Appendix

This section will show why adding extra information will improve the accuracy.

First, notation is given as follows: $\mathbf{x} \sim \mathcal{N}(\mathbf{m_x}, \mathbf{P}_{22})$ is the set of features (extra information) from the convolution block, where $\mathcal{N}(\mathbf{m_x}, \mathbf{P}_{22})$ is the normal distribution with mean $\mathbf{m_x}$ and covariance $\mathbf{P}_{22}$ and the symbol $\sim$ means "satisfies"; $\mathbf{y} \sim \mathcal{N}(\mathbf{m_y}, \mathbf{P}_{11})$ is the set of output; $\mathbf{e} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is the Gaussian white noise and $\mathbf{I}$ is identity matrix; $\mathbf{m_y} = \mathrm{E}[\mathbf{y}] = \hat{\mathbf{x}}_k$ is the mean of $\mathbf{y}$ and the symbol E is the meaning of expectation.

Therefore, without extra information, the expectation of output is $\hat{\mathbf{x}}_k$.

Then, assume that $\mathbf{P}_{22} > 0$ ($\mathbf{P}_{22}$ is positive definite) and $\mathbf{x}$ can be expressed by a complex nonlinear function f, that is,

$$\mathbf{x} = f(\mathbf{y}) + \mathbf{e}.$$

Therefore, the joint distribution, provided $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{e}$ obey the Gaussian distribution, respectively, is given by:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m_y} \\ \mathbf{m_x} \end{pmatrix}, \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix} \right).$$

A new random variable $(\mathbf{y}|\mathbf{x})$ is considered as a new measurement given extra information $\mathbf{x}$. According to the Bayes formula, the marginal distribution, and

properties of normal distribution, it is also Gaussian. The formula is given as follows:

$$\mathrm{P}(\mathbf{y}|\mathbf{x}) = \frac{\mathrm{P}\begin{pmatrix}\mathbf{y}\\ \mathbf{x}\end{pmatrix}}{\mathrm{P}(\mathbf{x})} \sim \mathcal{N}(\mathbf{m}_{\mathbf{y}|\mathbf{x}}, \mathbf{P}),$$

where $\mathbf{m}_{\mathbf{y}|\mathbf{x}} = \mathrm{E}[\mathbf{y}|\mathbf{x}] = \hat{\mathbf{x}}_{k+1}$ is defined as the new expectation of output given extra information.

According to the least squares estimate, it can be further simplified as follows:

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + \mathbf{P}_{12}\mathbf{P}_{22}^{-1}(\mathbf{x} - \mathbf{m}_{\mathbf{x}}).$$

The error is defined as $\tilde{\mathbf{y}} = (\mathbf{y}|\mathbf{x}) - \hat{\mathbf{x}}_{k+1}$. The expectation of the error is given as follows:

$$\mathrm{E}[\tilde{\mathbf{y}}] = \mathrm{E}[\mathbf{y}|\mathbf{x}] - \mathrm{E}[\hat{\mathbf{x}}_{k+1}] = \hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_{k+1} = \mathbf{0}.$$

Therefore, our estimate is unbiased.

Moreover, the error between the new measurement given extra information and the original expectation of output is given as follows:

$$\mathrm{E}[\mathbf{y}|\mathbf{x}] - \mathrm{E}[\hat{\mathbf{x}}_k] = \hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k = \mathbf{P}_{12}\mathbf{P}_{22}^{-1}(\mathbf{x} - \mathbf{m}_{\mathbf{x}}) \neq \mathbf{0}.$$

It is obvious that the original expectation of output is bias given extra information. CNNs can also be interpreted in this cascade framework, i.e., a convolution block as extra information and a classification block as the new measurement given extra information.