COMP9444 Group Project
The Use of Convolutional Nueral Network on Garbage Classification

*Team Rabbit

Tianwen Qin z5449068

Yuhao Ke z5365195

Qiuli Jin z5471857

Caiwei Liu z5391857

Weichong Huang z5437652

I. Introduction

With the rapid development of cities and the increase of popularity around the world, the disposal of garbage has led to serious problems. In Indonesia, 7,000 tons of trash arrives daily at Bantar Gebang, a giant mountain made of landfill that is more than 200 football fields wide and over 15 stories high, surrounded by villages with more than 20,000 residents. New York Times also documented the overwhelming trash problems with a piece of news that the increased population of rats attacked and sent four sanitation workers to the hospital over the course of five months [1] [2]. Garbage classification is an efficient way to sooth the pressure for the environment and human kind coming along with the production of garbage [3]. However, the traditional classification method by human hands is both expensive and inefficient [4]. Thus, An accurate and affordable classification method is needed.

Image classification is a machine learning field that focus on classifying labeled images. This technique has been used for the classification of garbage in the literature [5]. In this paper, we apply and compare the performance of two deep learning methods in the application of garbage classification. We also provide the discuss about the pre-pocess of the images, the limitation of each methods and potential future work.

The rest of the paper is structured as follows: In section 2, we will introduce some related work in the literature. We explain the methods we apply in section 3 and provide the experiment results in section 4. In section 5, we provide the discussion of both methods and in section 6, we provide the conclusion of the paper.

II. Literature Review

Image classification has been widely used in the field of garbage classification. In particular, deep learning shows promising results in such tasks. Rabano et al. utilise MobileNet to classify a data set of 2527 trash images and gain an accuracy of 87.2% [6]. They also manage to develop a smart phone application implemented with their model. Meng et al. [7] show that ResNet50 achieves better accuracy in garbage classification than Support Vector Machines (SVM), a simple machine learning method. They also find that simple convolutional neural network (CNN) is less efficient than ResNet50 but seems to get similarly good performance as the data size increases. Ozkaya et al. compare different models like Alexnet, VGG16, Googlenet and ResNet along with two different classifiers: SVM and SoftMax [8]. They find that GoogleNet combined with SVM achieves the accuracy of 97.86%.

Overall, there exist enough research to show that deep learning is an effective way to achieve the classification of garbage. Thus, deep learning is the main method under our investigation. However, we also notice that the discuss of the affects due to the image pre-process has been barely mentioned in existing researches. Meng et al. [7] mentioned the pre-process but did not investigate the affects thoroughly. In the rest of our paper, we will illustrate this affect in detail.

III. Methodology

A. ResNet Overview

ResNet has significantly advanced the capability of neural networks in terms of progression. Thanks to this method, the depth of neural networks has been enhanced. Proposed by Kaiming He and others in their groundbreaking paper "Deep Residual Learning for Image Recognition" (2015) [9], ResNet addresses the inherent degradation problem found in traditional deep neural networks.

ResNet introduces the concept of residual blocks, allowing layers to learn a residual function with reference to the layer inputs through shortcut connections. [9] This enables the training of much deeper neural networks (networks with over a hundred layers) to be possible.

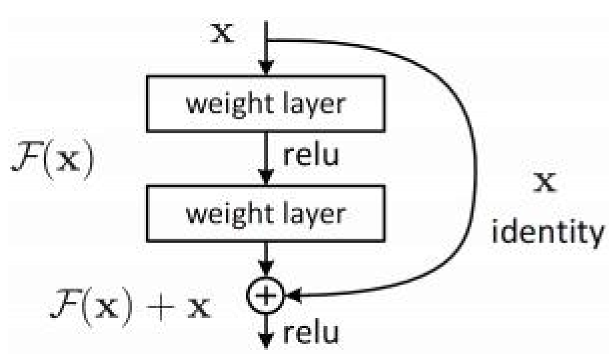

Fig. 1. The residual black of ResNet.

The figure1 shows the residual block. It permits the input to be directly connected to later layers through a structure colloquially referred to as a “shortcut connection”. Within this block, the input x first passes through two convolutional layers, each followed by a ReLU activation function. Then, the input x bypasses these two processing layers via the shortcut connection and is directly added to the processed result F(x). This design allows the network to learn the residual between the input and output, that is, F(x)+x. This allows the gradient to flow directly through the network, giving the network the ability to adaptively adjust the weights of each layer

B. MobileNetV2 Overview

Following MobileNetV1, the Google research team submitted the paper “MobileNetV2: Inverted Residuals and Linear Bottlenecks” (Sandler, Mark, et al. 2018) [10] in 2018. MobileNetV2 has fewer parameters and computational requirements compared to the V1 model, with higher accuracy.

First, dimensionality is increased using 1x1 convolutions. The number of channels is increased through lightweight 1x1 convolutions, thus obtaining more feature representation space.

Then, features are extracted using 3x3 depthwise separable convolutions (depthwise separable convolutions) [11] . The expanded feature map is processed through depthwise separable convolutions. Spatial convolution (depth convolution) is first applied independently to each channel, followed by the use of 1x1 convolutions (pointwise convolutions) to combine these features.

Lastly, dimensionality is reduced using 1x1 convolutions, and no activation function is used during dimension reduction. This layer uses a linear activation (as opposed to a ReLU activation). This is the source of the linearity in the linear residual structure.

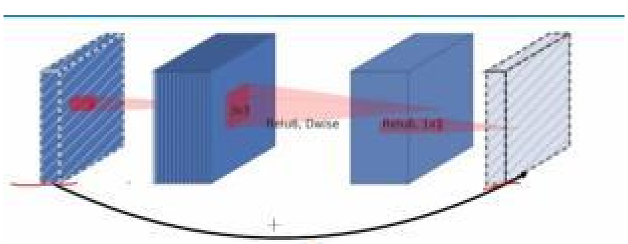

Fig. 2. Inverted linear residual structures and depthwise separable convolutions

Figure 2 shows the inverted linear residual block that features thin ends (fewer channels) and a thick middle (more channels), with bottlenecks at both ends of the residual block.

Input	Operator	Output
$h \times w \times k$	1 × 1 Conv2D, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3 × 3 Dwise Stride=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	Linear 1 × 1 Conv2D	$\frac{h}{s} \times \frac{w}{s} \times k'$

Fig. 3. The computational logic of MobileNetV2 residual block.

The computational logic of the MobileNetV2 residual block is illustrated in the figure3. The number of channels in the input layerisk, the number of channels in the output layer is k’, and the stride is s (s=1 or 2), with the parameter t being the channel expansion factor.

C. DenseNet Overview

Dense NetIII-C, short for Dense Convolutional Network, is a structure where, unlike traditional convolutional neural networks where each layer’s output is passed only to the next layer, in Dense Net, the output of each layer is used as an additional input for every subsequent layer. [12] This means that each layer is directly connected to all preceding layers. The input for each layer includes not only the output of the previous layer but also the outputs of all preceding layers.

Its connection method can alleviate the problem of gradient vanishing. Thanks to the network’s efficiency and connection style, it also has the advantages of encouraging feature reuse, reducing the number of parameters, and achieving the effects of a deeper network with fewer layers.

D. VGG Overview

The VGG model was originally developed by the Visual Geometry Group at the University of Oxford, hence the name “VGG”. Its main characteristics are its simplicity in structure and depth. The model utilizes repetitive structural units, each composed of convolutional layers, typically followed by a nonlinear activation function (such as ReLU). Following these repeated units, there are usually several fully connected layers and a softmax output layer for classification. [13]VGG demonstrates that increasing the depth of the network can significantly enhance performance. However, the VGG model is relatively large, with high computational costs, and is prone to overfitting.

IV. Experimental Setup

A. Data

The data set we used can be downloaded from this link: https://github.com/Jarvan39/9444Garbage-data-set. This data set contains 2467 labeled images in total. All the labels and the number of corresponding images are shown as follows:

1) Cardboard, 393
2) Glass, 491
3) Metal, 400
4) Paper, 584
5) Plastic, 472
6) Trash, 127

The garbage labeled as trash means that it can not be recycled. And we notice that the number of trash is well below the number of all other kinds.

B. Pipeline

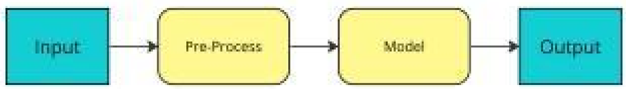

Fig. 4. The general pipeline for all the models.

Figure 4 shows the pipeline for the methods we investigate. It is worth to point out that resize is not included in the pre-process since models require different picture sizes. We need to resize the images accordingly before sending them to the models.

C. Metrics and Hyperparameters

In the experiment, we evaluate the performance of our model using accuracy rate and confusion matrix. In the MobileNetV2 model, we set the learning rate to be 0.001. In the ResNet model, we set the learning rate to be 0.000055, and the momentum to be 0.1. In the DenseNet model, we set the learning rate to be 0.001. In Vgg19, we apply the learning rate reduction approach with minimum to be 0.00001.

V. Experiment Results

TABLE I
Evaluation of different models and different pre-process.

Model	Pre-Process	Number of Epochs	Loss	Accuracy
MobileNetV2	None	10	0.6153	81.4%
MobileNetV2	Normalisation and Augmentation	10	0.3924	89.8%
DenseNet	None	10	0.7528	80.63%
DenseNet	Augmentation	10	0.6417	82.61%
Vgg19	None	12	1.5835	33.6%
Vgg19	Normalisation	12	0.7831	**76.5%**
ResNet	None	30	1.1029	**93.25%**
ResNet	Normalisation	30	1.1115	**93.25%**

A. Accuracy

Table I shows the evaluation for all the methods we investigate. Number of epochs stands for the minimum number of epochs required for the model to yield the approximately highest accuracy. In the pre-process column, we did not mentioned the resize, as some models like ResNet require specific image sizes. We change the image size for these models accordingly. The main pre-process approaches we use are normalisation and augmentation. We have done all the possible combinations out of these 2 methods and only kept the combination with the best performance. The column loss and accuracy show the loss value and accuracy rate after the last epoch.

The highest accuracy is 93.25% among all the models and pre-process. ResNet model achieves this accuracy no matter any pre-process applied or not. Model Vgg19 shows the highest difference after the application of pre-process: from 33.6% to 76.5%. This shows that the pre-process is essential to the training of Vgg19 model. MobileNetV2 and DenseNet shows similar performance when no pre-process is applied. However, MobileNetV2 shows more improvement in accuracy after the pre-process in applied. In general, pre-process of the images can improve the performance of deep learning models.

B. Efficiency

As table I shows, MobileNetV2 and DenseNet can yield to its best performance in the least number of epochs. Vgg19 needs slightly more epochs and ResNet requires the most. However, the trend of yielding is mostly detected by human-eyes. In this case, we can only make the most general argument that ResNet requires significantly more time to train than the other models.

C. Confusion Matrix

While the rate of accuracy is able to show the overall performance of a model, the confusion matrix could illustrate more detailed information. We provide the confusion matrix for all the 4 models with pre-process methods as pre-process methods can generally improve the accuracy. All the rest matrices can be viewed in this link: https://drive.google.com/file/d/1jfkfzWfPT3zNUuNel4TyCSKOwYtAUtc/view?usp=drive_link.

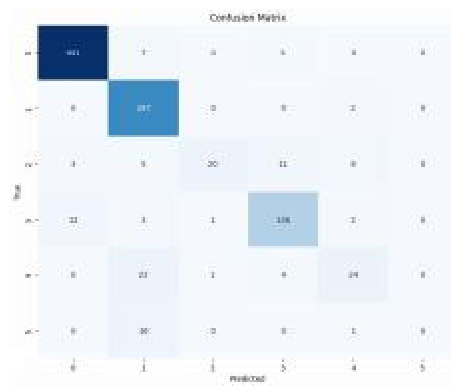

Fig. 5. The confusion matrix of MobileNetV2.

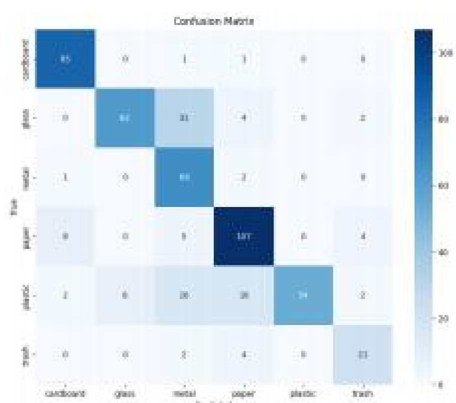

Fig. 6. The confusion matrix of DenseNet.

Figure 5,6,7, and8 show the confusion matrices for all the models we investigate. Despite the variation of the accuracy among the models. All of the 4 models classify trash much worse than other classes. The cause of this imbalance has

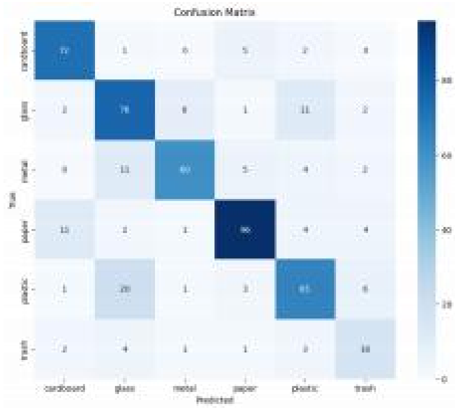

Fig. 7. The confusion matrix of Vgg19.

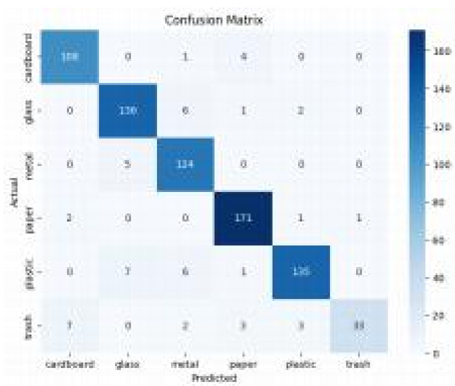

Fig. 8. The confusion matrix of ResNet.

two possibilities. The first is the imbalance of the data. Even though we have augmented the data in some models, the augmentation applied for all the classes. Thus, the number of images with label “trash” sent to the model is still much lower than the others. The second possible cause is that all the images with label “trash” stand for garbage not recyclable. That means the variance of the items represented by the images in this label is much higher than other thoes with other labels. Surprisingly, Vgg19, the model with the lowest accuracy rate of all the models handle this imbalance better than MobileNetV2 and DenseNet.

Figure 9 shows the confusion martix of Vgg19 without any pre-process. In contrast to figure 7, when the data is not normalised, the model simply guesses all the images other than cardboard as paper. This shows the importance of data normalisation for the model Vgg19.

VI. Conclusion

In this paper, we use four deep learning algorithm models, MobileNetV2, DenseNet, Vgg19, and ResNet, to solve the problem of garbage image classification. They are commonly used convolutional neural network architectures in deep learning. In our experiment, we found differences in performance among these algorithms.

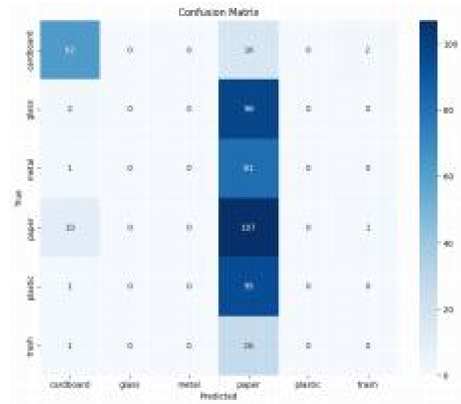

Fig. 9. The confusion matrix of Vgg19 without pre-process.

MobileNetV2 has lower latency and faster speed in the inference stage due to its fewer parameters and computational complexity. This makes it very suitable for real-time applications.

When testing the DenseNet model, data pre-process showed a slight improvement in the accuracy of the model. Due to the need to save intermediate feature maps, DenseNet has high memory requirements, which may limit its application in some resource constrained environments.

We found that data preprocessing has a significant improvement on Vgg19. However, Vgg19 contains a large number of parameters, which results in a large model file and requires more computational resources for training and inference.

The depth of ResNet enables it to capture more functionality and complexity, thus performing well in such complex tasks. We can use ResNet models pre trained on large-scale image classification tasks, and then fine tune them to adapt to specific tasks, thereby accelerating training and improving performance. And because ResNet has good robustness, the model is not sensitive to data. However, due to the depth of ResNet, ResNet may require more computational time and resources during the inference phase, making it unsuitable for applications with high latency requirements.

In conclusion, for garbage classification tasks that prioritize accuracy, we first excluded DenseNet and Vgg19, which had lower accuracy in this experiment. We recommend using the ResNet algorithm model when promoting this solution. Although compared to MobileNetV2 with similar accuracy, we have made slight compromises in terms of portability and efficiency. However, considering the large scale of equipment in waste treatment plants, we do not need to overly consider portability. At the same time, compared to manual classification, ResNet’s efficiency is still far ahead.

References

[1] K. Nakamura, “7 genuinely shocking facts about our global waste problem,” Dec 2021. [Online]. Available: https://www.globalcitizen.org/en/content/global-waste-facts-plastic-fashion-food/

[2] J. Gross, “Outside climate summit, trash in glasgow piles high,” Nov 2021. [Online]. Available: https://www.nytimes.com/2021/11/11/world/europe/glasgow-garbage-strike.html

[3] Y. Kuang and B. Lin, "Public participation and city sustainability: Evidence from urban garbage classification in china," *Sustainable Cities and Society*, vol. 67, p. 102741, 2021.

[4] Y. Guo, Q. Gu, and W. Pan, "Comparative study on garbage classification influence based on analysis of yangpu district of shanghai," in *IOP Conference Series: Earth and Environmental Science*, vol. 300, no. 3. IOP Publishing, 2019, p. 032066.

[5] Z. Kang, J. Yang, G. Li, and Z. Zhang, "An automatic garbage classification system based on deep learning," *IEEE Access*, vol. 8, pp. 140 019–140 029, 2020.

[6] S. L. Rabano, M. K. Cabatuan, E. Sybingco, E. P. Dadios, and E. J. Calilung, "Common garbage classification using mobilenet," in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. IEEE, 2018, pp. 1–4.

[7] S. Meng and W.-T. Chu, "A study of garbage classification with convolutional neural networks," in *2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN)*. IEEE, 2020, pp. 152–157.

[8] U. Ozkaya and L. Seyfi, "Fine-tuning models comparisons on garbage classification for recyclability," *arXiv preprint arXiv:1908.04393*, 2019.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[11] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conferenceon computer vision and pattern recognition*, 2017, pp. 4700–4708.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.