Garbage Classification Based on ResNet50

Yanyu Chen 30920211154151,[1] Dinghao Chen 30920211154124, [1]
Shunwen Shen 30920211154165, [1] Zhenchao Chen 30920211154152, [1]
Weijun Xu 30920211154172 [1]
[1] School of Informatics Xiamen University
chenchenyanyu02@163.com, 171196162@qq.com, insist_edu98@163.com, 1243088632@qq.com, 2550784180@qq.com

Abstract

Under the goal of carbon peaking and neutrality, waste recycling has become a very important project of the whole world. For this reason, classifying garbage is am important goal for humanity and deep learning can be used for this purpose. In this paper, AlexNet, VGG16, GoogLeNet, DenseNet and ResNet50 are used for the dataset we collect from Internet. In addition, transfer learning is used to obtain shorter training as well as higher accuracy. As a result of the conducted experiments, the best results are found in the ResNet50, 214 different types of garbage images are correctly classified with an accuracy of not less than 95%.

Introduction

Garbage sorting is a word that appears frequently in our eyes these years. As the country attaches more importance to environmental protection, China began to implement garbage classification in 2019, and has carried out targeted experiments in big cities such as Shanghai and Beijing. By 2021, garbage classification, a new measure to improve the environment and save resources, has been promoted to the nationwide.

Careful students can find that there are volunteers downstairs in our dormitory building to help with garbage classification every day, because students cannot properly classify all kinds of garbage by themselves. Every worthless thing in our life can become garbage around us at any time, and it is difficult for us to remember each garbage category. In traditional way, in order to establish a model description or machine learning system with classical machine learning techniques, the feature vector must first be extracted. In order to extract the feature vector, specialists are needed. These transactions require a great deal of time and experts are often very busy. These techniques cannot be used to process raw data without pre-processing and without expert help.

After learning the course of deep learning in this semester, our team agreed that we could develop a garbage classification model based on computer vision to solve this hot problem in reality. This research can help us carry out efficient and accurate garbage classification in daily life, make people's life more convenient and make all kinds of garbage better distinguished.

CNN model is widely used in image recognition, so this time we decide to adopt the method based on CNN for image recognition. At present, there are many models based on image recognition in the market, such as R-CNN (Girshick et al. 2014), You Only Look Once (YOLO) (Redmon et al. 2016), GoogleNet (Szegedy et al. 2015), ResNet (He et al. 2016), etc. There is also MobileNet (Howard et al. 2017), a lightweight model deployed on mobile phones.

Through reading relevant papers, we have studied a lot of garbage classification models. Most of them are used after fine-tuning the original model, and many projects have already been implemented. Therefore, this project we choose is feasible. In order to verify the effectiveness of the methods used for our dataset for garbage classification, we carried out experimental tests respectively. From the experimental results, the proposed methods can complete the task well. Our main contributions can be summarized as following:

- Compare several common models on the market and check their effects after optimization.
- Choose the best model for tuning and get the final result. All the work we do is trying to make it practically used in photos of real scenes.

The rest of the paper is organized as follows, Section II is the related work and briefly clarify how the state-of-the-art solve the garbage classification problem. Section III introduces four classic deep neural networks for computer vision. Section IV introduces our experiments details and result. In sections V we make a conclusion.

Related Work

Umut Özkaya and Levent Seyfi fine-tune the common model on computer vision (Ozkaya and Seyfi 2019), such as vgg16 (Simonyan and Zisserman 2014), GoogleNet, ResNet, SquezeeNet (Iandola et al. 2016), DenseNet (Huang et al. 2017) and so on.after a series of experiments,they found the trash images were correctly classified the highest accuracy with GoogleNet + SVM (Hearst et al. 1998) ,the accuracy is about 97.86%.

RahMi Arda Aral et al. tested well-known Deep Learning models (Aral et al. 2018) such as Densenet121, Densenet169, InceptionResnetV2 (Szegedy et al. 2017), MobileNet, Xception (Chollet 2017) to provide the most efficient approach. As a result, the best results were found in the DenseNet121 using fine-tuning with a test accuracy rate of 95%.

The algorithm (Kang et al. 2020) applied to the system was based on ResNet-34 and three tailor-made modifications, including the multi feature fusion, the feature reuse of residual unit, and optimization of activation function. The ResNet-34 which combines all three modifications has the highest accuracy of 0.9796 and stable with the classification cycle as quick as 0.95 seconds on average.

SpotGarbage app (Mittal et al. 2016) utilizes the proposed deep architecture of fully convolutional networks for detecting garbage in images. The model has been trained on a newly introduced Garbage In Images (GINI) dataset, achieving a mean accuracy of 87.69%. The paper also proposes optimizations in the network architecture resulting in a reduction of 87.9% in memory usage and 96.8% in prediction time with no loss in accuracy, facilitating its usage in resource constrained smartphones.

Jianfei Yang et al. present a novel incremental learning framework, GarbageNet (Yang et al. 2021), to address lack of sufficient data, high cost of category increment, and noisy data quality.It achieves performance of 96.96% with acceptable inference speed.

Deep Learning Models

Through a series of learning of image recognition, we have learned several models that can be applied to the garbage classification network. Now we give a brief introduction to them.

- **ResNet50** (He et al. 2016): To understand ResNet50, you must first understand the definition of ResNet. ResNet was proposed to solve the problem of network degradation in deep learning networks, traditional convolutional networks or fully connected networks In the process of information transmission, there are more or less problems such as information loss, loss, etc., and there are also problems that cause gradients to disappear or explode. ResNet solves this problem to a certain extent. By increasing the direct connection channel, the integrity of the information is protected. The entire network only needs to learn the difference between input and output, which simplifies the learning objectives and difficulty. the ResNet50 means that the number of layers with weights is 50 in ResNet, such as convolutional layers and fully connected layers. The idea is still used in the residual network, but the weight layer number of 50 layers is selected.
- **AlexNet** (Krizhevsky, Sutskever, and Hinton 2012): AlexNet deepens the network results on the basis of LeNet (LeCun et al. 1998), and can learn richer and higher-dimensional image features. AlexNet has a deeper network structure, including five-layer convolution and three-layer full connection.

  At the same time, three methods are used to improve the training process of the model. They are to use data augmentation to expand the data set to randomly change the training samples to avoid the model from overly relying on a certain These attributes suppress over-fitting to a certain extent, use Dropout to suppress over-fitting, and use ReLu activation function to reduce the phenomenon of gradient disappearance.
- **VGG16** (Simonyan and Zisserman 2014): VGG is the first network to open the era of small convolution. Compared with AlexNet, VGG uses several consecutive 3x3 convolution kernels to replace the larger convolution kernel in AlexNet. Simply put, three 3x3 convolution kernels can replace 7x7 convolution kernels, and two 3x3 convolution kernels can replace 5x5 convolution kernels. The purpose of this is to ensure that they have the same field of view. Enhancing the depth of the network improves the effect of the neural network to a certain extent. VGG16 is a VGG network that contains 16 hidden layers. Among the 16 hidden layers, there are 13 convolutional layers and 3 fully connected layers.
- **GoogLeNet** (Szegedy et al. 2015): GooLeNet introduces the Inception module, and wants to approximate the optimal sparse structure by building a dense block structure, so as to achieve the purpose of improving performance without increasing the amount of calculation. While controlling the amount of calculation and parameters, GoogLeNet achieved good classification performance, and removed the final fully connected layer in the design, used the global average pooling layer, and improved the utilization of parameters by introducing the Inception module. Like VGG, multiple small convolution kernels are used instead of large convolution kernels, and convolution kernels of different sizes are used to increase diversity.
- **DenseNet** (Huang et al. 2017): In deep learning networks, as the depth of the network deepens, the problem of vanishing gradients will become more obvious. At present, many papers have proposed solutions to this problem. For example, the ResNet introduced above, their core is to create short paths from early layers to later layers. DenseNet is to use this idea to the extreme. Through this idea, all layers are directly connected under the premise of ensuring the greatest degree of information transmission between layers in the network. DenseNet uses a dense connection mechanism, that is, all layers are connected to each other, and each layer is dimensionally concated with the channels of all previous layers to achieve feature reuse, and together as the input of the next layer.

Experiments

Dataset

The correct classification of garbage has a positive effect on our society. The model we built this time divides waste into four categories, namely hazardous waste, recyclable waste, kitchen waste, and other waste. Obtain a large number of training data sets from major websites by writing crawlers, and then manually label relevant labels. Here our data sets come from two parts, one is to crawl through Google Gallery, and the other part is to find relevant garbage classification data sets from various big data competition

Figure 1: The samples of waste categories

websites. Through these two jobs, we have collected more than 50,000 trash pictures with labels.

Among these more than 50,000 pictures, 53 types of food waste, 106 types of recyclable garbage, 36 types of other garbage, and 19 types of hazardous garbage are included. Figure 1 shows the samples of kitchen waste, recyclable waste, other waste, and hazardous waste in turn.

Data Preprocessing

In order to generate the corresponding label file based on the file name, we designed a python script. The label file is usually named dir_label.txt. In the label file, we have designed several columns to categorize the data, the front is the small classification label, the latter is the large classification label, and the last is the quantity and number identification.

Next, we use a reasonable classification method to divide the data set into three categories. The three data sets classified by us are training set, validation set and test set. We use proper naming to identify these three types of files. The training set is named train.txt, the verification set is val.txt, and the test set is test.txt. The functions of these three types of files are different. The training set and the validation set are used to train the model, and the test set is used to check and accept the effect of the final model.

Before the experiment, certain pre-processing tasks need to be carried out. The first thing is to check the quality of the picture, use Image in a library called PIL in Python to read, and catch the Error and Warning exceptions. We will take measures to delete the pictures in question.We have obtained three text files using the above method, and our data sets are stored in the text files. Among them, the training set contains 48045(85%) pictures, the validation set contains 5652(10%) pictures, and the test set contains 2826(5%) pictures.

Next, we put the three data set files and label files in the same directory as the trash file library. The trash file library is the real picture of the pre-downloaded trash file. After that, we wrote a module to read the data and read the data into our program. After this, it is our operation to process the image data. We first need to load the data and preprocess the data. Our processing operation is to fill the image proportionally to a pure black image with a size of 280 times 280, and then transform it to a size of 224 times 224. This kind of data preprocessing operation is a very common processing method in image classification tasks.

In addition, in order to enhance our data to improve the effect of the model, we use the transforms module provided by pytorch to add two random operations such as horizontal flip and vertical flip to the image. Such data enhancement methods are also very common in image classification tasks.

Methods for Comparison

We choose GoogLeNet, VGG16, AlexNet and DenseNet121 as the objects for comparison, and compare these models with ResNet50 to study their performance in waste classification.

Implementation Details

In the Experiment, the batch size is set to 64, learning rate is set to 0.0001, learning rate step size is set to 20, and weight decay is set to 0.01. Moreover, the experiments were performed using GTX 2080ti GPU.

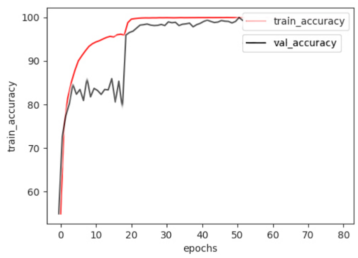
Training

After the data is read and preprocessed, we start to write the training code of the model. The first thing we do is to write the code of the ResNet50 model, which is not complicated. Our network structure directly uses the ResNet50 model provided by torchvision, and uses the ResNet50 pretraining model, in order to achieve better training results. The loss function used by our algorithm is the cross-entropy loss function, and the optimizer uses the conventional Adam, and uses StepLR to attenuate the learning rate in order to obtain a better model effect. Also, we et checkpoints during the training process. When the effect on the validation set is better updated, save the model at this time. Such a check will be checked once after each round of training to ensure that the best model effect can be captured.

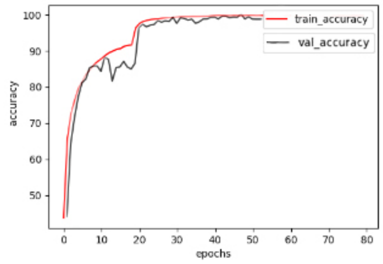
The Batch Size of our model training is set to 64, and the GPU memory is about 4G. During training, the effect of training is printed in real time and output to the log for our observation and analysis of the data after the experiment.

Similar to the above operation, we then trained four different models, these four models are AlexNet, VGG16, GoogLeNet and DenseNet121. It should be noted that the training set, validation set, and test set we use are the same in the above models, which is due to the consideration of control variables. Similarly, the rounds we train on these models are also consistent. Due to the limitations of the experimental conditions, we did not use the parallel training method, but to train the models one by one. After each model is trained, we will draw the performance of the above model on the training set and test set, and visually display it in the form of a graph. This can be easily done using the plt module in Python.

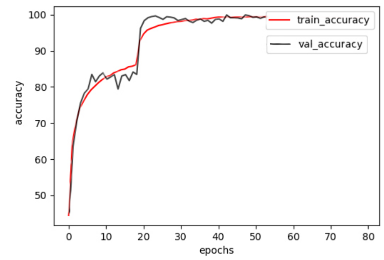
The log during model training will be saved. We use the excel tool to analyze the data for a second time, with which we gather the data together, use the table for horizontal comparison, and comprehensively compare the multiple models we trained. We get the results of experiment after discussion.



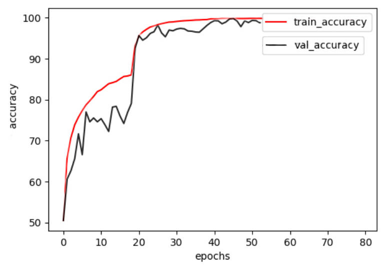
(a) The accuracy of ResNet50



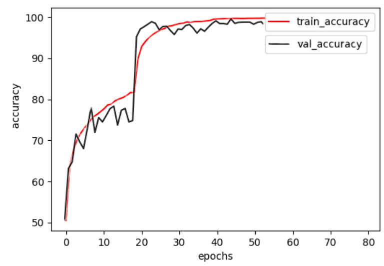
(b) The accuracy of GoogLeNet



(c) The accuracy of VGG16



(d) The accuracy of AlexNet



(e) The accuracy of DenseNet121

Figure 2: The accuracy of the models in training set and verification set

Experiment Results

In this study, the experiments were performed on known CNN models. As a result of these experiments, the applied deep learning models achieved more than 92% test accuracy. The GoogLeNet model with 80 epochs achieved a 94.2% of test accuracy. VGG16 model with 80 epoch, we achieved a 93.7% of test accuracy. And AlexNet achieved 92.6%, DenseNet121 achieved 92%. The highest test accuracy with ResNet50 model is 95%, with 80 epochs. The results were shown in Table 1.

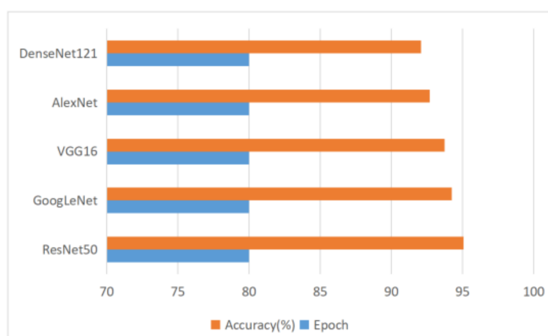
Table 1: The test accuracies of models

Model	Epoch	Accuracy(%)
ResNet50	80	95.046
GoogLeNet	80	94.233
VGG16	80	93.719
AlexNet	80	92.675
DenseNet121	80	92.056

We also made histograms comparing the accuracy of the five models on the test set, as shown in Figure 3 . It can be seen from the experimental results that From the comparison chart of the experimental results, it can be seen that among these several neural network models, ResNet50 has the fastest convergence speed and the best effect.

ResNet50 model was applied to 18 epoch in pre-training phase. At the next training phase, 80 epochs were applied and 95% test accuracy was achieved. When we analyzed the validation results during the training of the ResNet50 model, we noticed that the validation test results pumped after the 18 epoch in the pre-training phase. After this evaluation, in the next experiment, we decided to fifinish the pre-training phase in 18 epochs and increase the subsequent training phase. Then, we set the training phase to 80 epoch and with this trial, we achieved 95.0% test accuracy. The results can be seen in Figure 2(a).

With these results, we applied the same experiment on GoogLeNet, VGG16, AlexNet and DenseNet121 Model. In this experimental result, we achieved a 94.2%, 93.7%, 92.6%, 92.0% of test accuracy respectively. The result were shown in Figure 2(b)-(e).



	ResNet50	GoogLeNet	VGG16	AlexNet	DenseNet121
Accuracy (%)	95.046	94.233	93.719	92.675	92.056
Epoch	80	80	80	80	80

Figure 3: The accuracy comparison chart of different models

Conclusion

The main task of this paper is to classify waste by recognizing all kinds of waste pictures. We collected more than 50,000 waste images from the Internet and used these images to train and compare five classic image recognition models, namely ResNet50, GoogLeNet, VGG16, AlexNet and DenseNet121.

As a result, the most successful test accuracy rate is ResNet50, with an accuracy rate of 95.046%. In the light of the conducted experiments, it can be said that deep learning algorithms can be used to classify waste. In the next study, size of the dataset will be increased to obtain more successful results. It will also aim to realize various improvements on the models.

References

Aral, R. A.; Keskin, Ş. R.; Kaya, M.; and Hacıömeroğlu, M. 2018. Classification of trashnet dataset based on deep learning models. In *2018 IEEE International Conference on Big Data (Big Data)*, 2058–2062. IEEE.

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

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hearst, M. A.; Dumais, S. T.; Osuna, E.; Platt, J.; and Scholkopf, B. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4): 18–28.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; and Keutzer, K. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

Kang, Z.; Yang, J.; Li, G.; and Zhang, Z. 2020. An automatic garbage classification system based on deep learning. *IEEE Access*, 8: 140019–140029.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Mittal, G.; Yagnik, K. B.; Garg, M.; and Krishnan, N. C. 2016. Spotgarbage: smartphone app to detect garbage using deep learning. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 940–945.

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

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

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

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Yang, J.; Zeng, Z.; Wang, K.; Zou, H.; and Xie, L. 2021. GarbageNet: A Unified Learning Framework for Robust Garbage Classification. *IEEE Transactions on Artificial Intelligence*.