Machine Learning: Use ML Models in Helping Garbage Classification in Virginia

CS4991 Capstone Report, 2023

Hanzhang Zhao
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
hz9xs@virginia.edu

ABSTRACT

In our daily life, it is a burden to think about which trash bin we should put our garbage in; so it is important to develop a tool that can help people to classify garbage easily. To solve this problem, two friends of mine and me developed our own machine-learning models that could be used to identify the type of garbage by looking at garbage images. To implement the model, we first decided to use Convolutional Neural Network (CNN) as our model. Then, we found a dataset containing twelve types of garbage images on Kaggle and used these images to train our CNN model. Finally, we checked the validation score to make sure our model was trained. After training the model, we tested the model using real-life garbage images taken manually and our model resulted in a 79% overall accuracy for classifying real-life garbage images. With further training on the model and more and better input of garbage images, our model will achieve higher accuracy in the future.

1. INTRODUCTION

Do you care about the issue of garbage classification and throw garbage into the right trash bin in your daily life? Many people will answer “no” to this question. People in a hurry may not take the time and trouble to put their garbage into the proper bin.

For decades, the governments has raised many solutions to solve this problem, including classifying trash bins, advertising for garbage classification, and even fining those who are not disposing of garbage properly. However, these methods do not work well. Therefore, our project was developed to make garbage classification more convenient and accurate, requiring fewer human resources to make garbage classification easier.

On Kaggle, we found a dataset containing twelve types of garbage images, which we used to develop our machine-learning model. By doing so, we could contribute to the recycling of materials in our local community and increased the efficiency of disposal.

2. RELATED WORKS

DinoKing (Axford, 2022) implemented a machine-learning model as a solution to the problem of garbage classification. He used the previous matured model as the base for his own model and ultimately reached an accuracy rate of about 92%. He first created a mobilenetv2 model without the last layer; then added pre-processing and pooling layers, followed by a softmax layer. The advantage of his model was its high accuracy; the drawback was the amount of time required for model training. Inspired by this model, we used it as our base and transferred it to our updated model, which was simpler but achieved a lower accuracy rate despite its greater training efficiency.

Daniel (Daniel, 2022) implemented a machine-learning model as a solution to the problem of garbage classification. He used mobilenetv3 to develop a model based on a pre-trained model; then used a dataset from Kaggle composed of nine types of garbage to train his model so it could differentiate nine types of garbage. The advantage of his model was that training was more efficient and had relatively high accuracy; the limitation was that his model could only differentiate between nine types of garbage. We learned from Daniel's project that Kaggle provides many great garbage classification datasets, so we found a dataset composed of twelve types of garbage (Mohamed, 2021), enabling our model to differentiate between twelve instead of nine types of garbage.

3. PROCESS DESIGN

The design process contains three sections. The first section was the process of choosing and constructing the model that we used for coding, including how and why we chose our model. The second section was the coding part, which showed the detail process of constructing our machine learning model program. The last section was validation and testing, which tested and proved the accuracy of our model, giving the overall results.

3.1 Choose and Constructing Model

We chose CNN as our main model. There were several reasons to use CNN. First, our team was familiar with CNN and had coding experience on it (Nguyen, 2022). Second, CNN supports multi-class classification, which was suitable for our project since our dataset had twelve classes. Finally, CNN can detect features without human supervision, so that we did not need to manually digitize images before using them, which was convenient.

After choosing our model architecture, we had to construct our model. We decided to use pre-trained models with transfer learning—building our model based on existing models—because that was more accurate and reliable than what we constructed ourselves. Among many pre-trained models, we tried the Xception network, the mobilenetv2 network, the Densenet network, and others. The Densenet Network turned out to be the best.

3.2 Coding

After we had decided our main algorithm and the model, we did the coding using Jupyter Notebook. To begin with, we pre-processed the dataset to make it easy to use. We first cleaned the dataset by using the one-hot-decoder to transform outlier data into the desired format and we used pipeline to normalize the entire dataset. After the data was cleaned, we separated the dataset into the training set and the validation set. The training set was used to train the model and the validation set was used to determine if our model was fit enough to be used. To make sure we were getting the right traning data, we visualized some of the train samples from the training set and checked to ensure that the images corresponded to the label.

After pre-processing our data, we constructed our model using Densenet Network with transfer learning. We took the entire Densenet Network model except for its first layer (the input layer) and its last layer (the output layer), and added our own input layer and output layer based on the input image size and the ourcome function, which combined into our model. We trained the model by freezing the pre-train weights and only trained the top layer of the model (Nguyen, 2022). It turned out that training the top layer had a better result than training the entire model, which was less efficient.

3.3 Validation and Testing

After training the model, we evaluated the model using the validation set and drew a confusion matrix to analyze the result to make sure our model was well fit. After our model was trained well, we tested our model by doing experiments that evaluatd the model using real-life images. We took a set of real-life garbages, inputed them into our model, and got the result of their classifications. We compared the results from the model with the true value. Then we drew a confusion matrix to analyze the accuracy of predicting each class and to see whether there were relations between different classes when predicting. At last, we compared the confusion matrix from the validation set to that of the testing set, and drew some conclusions from the comparison.

4. RESULTS

For the validation part of the process, our results showed that the overall accuracy that our prediction matched the true value was about 95%. We drew a confusion matrix that clearly displayed the accuracy of each class. (Figure 1)

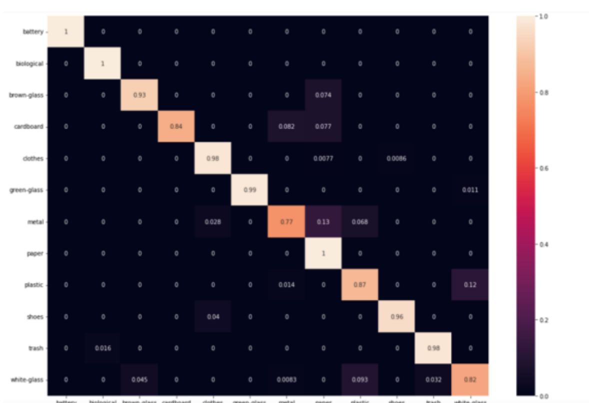

Figure 1: The confusion matrix for validation

For the confusion matrix, the grid at row x and column y represents for the percentage that class x has been classified as class y. And the whiter the grid, the higher the percentage (at most 1). If x equals y, the grid represents the accuracy of predicting this class by our model.

From the confusion matrix we can see that for most classes our predictions matched the true value. However, for the metal class, our model classified about 13% of metal images as the paper class. For plastic class, our model classified about 13% of the plastic image into as white-glass class, probably due to the similar color. Similar results happened for the white-glass class, in which about 9.3% of white-glass images were classified as the plastic class. In conclusion, although there were minor mistakes, our model worked well on the validation set.

As for the testing dataset, we had an overall prediction accuracy of about 79% for 253 garbage images. This result was worse than that of the validation set but within our expectation. We also drew a confusion matrix in order to deeply analyze our results. (Figure 2)

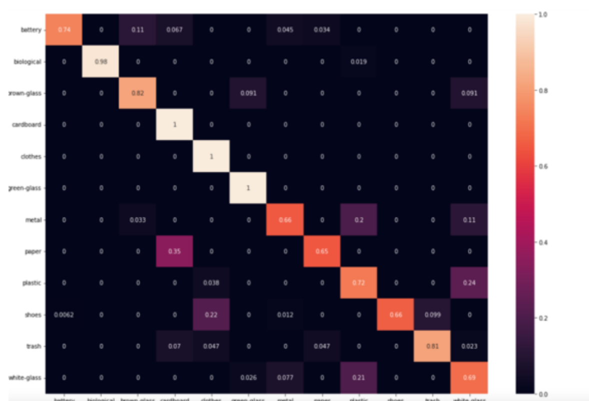

Figure 2: The confusion matrix for testing

For the battery class, we can see that 74% of battery images were classified correctly while about 11% were classified as brown-glass. This result was likely because some of the battery images we took had a brown color. For the brown-glass class, about 9.1% of brown-glass images were classified as green-glass while about 9.1% were classified as white-glass. This made sense since they all belonged to glass with similar shapes, and their colors might be mixed up due to the photoing angle.

For the metal class, about 20% of metal images were classified as plastic. This relation was also reflected when testing our validation set, in which 6.8% of metal images were classified as plastic. This might be due to the fact that metal bottles looked similar to plastic bottles.

For the paper class, 65% of paper images were classified correctly while 35% were classified as cardboard. This might be the result of their similar shapes.

For the plastic class, 24% of plastic images were classified as white-glass, while in the white-glass class, 21% of white-glass images were classified as plastic. Considering the same situation happened in our validation set, it was very likely that our model had difficulties differentiating between plastic and white-glass.

For the shoes class, 22% were classified as clothes. This confused us a lot since the shoes and the clothes had entirely different outlooks. Overall, our model had a good performance, but it had difficulties differentiating certain classes for various reasons.

5. CONCLUSION

Our model worked best on the validation sets. When testing our model with real life images, the accuracy is 79%, which was acceptable since there were many reasons affecting the image quality in real life. Although there were certain situations where our model faced difficulties, we believe with further training on the model and more and better input images of all kinds of garbage, our model could overcome those difficulties and come out with a higher accuracy. Our model had the potential of being used at both trash throwing end or trash disposal end. By implementing our model at the trash throwing end, people could be reminded quickly where their trash should go. For the garbage disposal case, our model could be combined with robotics to quickly sort out various types of garbage such as landfill or recyclable. Both fields are promising. For my own perspective, I learned how to select certain machine learning models and construct them to help me complete certain missions, which will be useful for my future career.

6. FUTURE WORK

First, the flaw of our model is that it is too inaccurate to be used a in real-ife scenario. The accuracy rate is 79%, which means more than 20% garbages will be classified incorrectly. To improve the accuracy, we need to train the model with more and better images. Also, we might need more sophisticated model layers to advance the model.

Second, our model needs to be combined with image scanners to be used. For example, we need cameras to take the picture or scan the image of the trash when people are trying to throw off the garbage. The image taken should be sent to our system as the input and our model would give back the output to direct people for garbage throwing.

REFERENCES

Axford, A. (2022, March 11). Garbage Classification: MobileNetV2 92% accuracy. Kaggle. https://www.kaggle.com/code/alexfordna/garbage-classification-mobilenetv2-92-accuracy/notebook

Daniel, S. (2022, January 9). Advanced Waste Classification with Machine Learning. Towards Data Science. https://towardsdatascience.com/advanced-waste-classification-with-machine-learning-6445bff1304f

Mohamed, M. (2021, January 24). Garbage classification (12 classes). Kaggle. Retrieved December 7, 2022, from https://www.kaggle.com/datasets/mostafaabla/garbage-classification

Nguyen, R. (n.d.). CS4774 Codeathon 2. Google Colab. Retrieved December 7, 2022, from https://colab.research.google.com/drive/1HyPl8ewBjHMw45shkcNZWmVhDyHMsVDm