Garbage Classification Using Deep Learning Techniques

1st Md. Mainul Ahsan
mainulahasanswaad@gmail.com

2nd Emdadul Haque
rafathaque1997@gmail.com

3rd Rezwan Ahmed
rezwanahmedhere@gmail.com

4th Md. Faraz Kabir Khan
farazkabir@gmail.com

***Abstract*—Garbage management is a major and unsolved issue across the country. Many countries have unique garbage management policies. There is no established method to manage the garbage appropriately. Proper waste recycling can give good profit to the government. That's why we propose an automation technique to detect and categorize waste properly. Garbage classification is an important stage in enabling cost-effective recycling among the tasks required for recycling. In this research, we try to recognize single waste objects in photographs and classify them into recycling categories.**

***Index Terms*—Garbage Classification, Recycling, Deep Learning, Convolutional Neural Networks, ResNet, MobileNetV2, DenseNet**

I. Introduction

Since the introduction of deep learning, image classification has become more efficient and accurate than ever. It has boosted the development of computer vision and helped people in various fields. Garbage classification can be done by image classification. In this work, we investigate modern deep learning or image classification techniques to classify garbage from an image.

Garbage is a by-product of urbanization. People's consumption levels have improved due to economic development, and residential waste has significantly increased. Currently, the world produces 2.01 billion tons of municipal solid trash every year, causing environmental damage. If the present situation continues, waste generation will rise by 70% in 2050 [9]. Recycling is gradually becoming an integral ingredient of a sustainable society. The acquisition, classification, and processing of recycled materials impose a substantial hidden cost on the entire recycling process. In garbage recycling, garbage classification is quite vital. If not correctly classified, it will not only squander reusable resources but may also pollute the environment and harm people's health. However, many residents are unaware of waste categorization or lack classification knowledge. Finding an automated approach to recycling is extremely valuable in an industrial and information-based world, as it has environmental and economic benefits.

Deep learning has gain traction in recent years, and its rapid improvement has greatly aided human convenience. Computer vision technology has had a significant impact on people's lives and has already established itself as an essential component of human life. Many researchers are currently conducting experiments using these innovations to apply to garbage classification tasks. In this work, we implement some of the image classifications techniques to categorize garbage. It is a challenging task as there can be several categories of waste. Also, deep learning models are said to be data-hungry while resources with garbage data are limited. Keeping the challenges in mind, we conduct experiments with different image classification models to classify garbage in respective categories. We then observe the performance and come to a conclusion.

II. Related Works

In the experiment [8] they showed that the SVM(Support Vector Machine) [7] performed better than the CNN. SVM [7] achieved a test accuracy of 63% using a 70/30 training/testing data split. In that experiment [8] the test accuracy that they achieved was only 22%. Which was way below acceptance level.They claim that the CNN's inability to learn is caused by poor hyper parameters organizations. As stated in this research paper [8], If the time frame had been longer, they could have collected much more data to improve their CNN results.In their experiment

In other experiment [5], they tried to identify a single garbage object in an image and place it in one of the recycling classes using Support Vector Machines (SVM) [7] with HOG features, simple Convolutional Neural Network (CNN), and CNN with residual blocks. In their experiment [5] they calculate the effectiveness of Simple CNN , and their test accuracy was 90%. In their research they stated that that very deep convolutional neural networks are difficult to train [3]. In their experiment [5] , after implementing ResNet they get 91.40% test accuracy.

The researchers of this paper [3] discovered, a multi-layer deep neural network can produce unexpected results. In this case, the training accuracy dropped as the layers increased, technically known as vanishing gradients. Therefore, the residual network was proposed to diminish this problem. A residual network, or ResNet [3] for short, is an artificial neural network that helps to build deeper neural network by utilizing **skip connections** or **shortcuts** to jump over some layers. There are two main types of blocks used in ResNet [3], depending mainly on whether the input and output dimensions are the same or different.

- **Identity Block:** When the input and output activation dimensions are the same.
- **Convolution Block:** When the input and output activation dimensions are different from each other.

DenseNet [4] (Dense Convolutional Network) is an architecture that focuses on making the deep learning networks go even deeper, but at the same time making them more efficient to train, by using shorter connections between the layers. DenseNet [4] is quite similar to ResNet [3] with some fundamental differences. ResNet [3] uses an additive method (+) that merges the previous layer (identity) with the future layer, whereas DenseNet [4] concatenates (.) the output of the previous layer with the future layer. The DenseNet has different versions, like DenseNet-121, DenseNet-160, DenseNet-201, etc.

Next generation of mobile vision applications is MobileNetV2 [1]. MobileNetV2 [1] is a substantial step forward from MobileNetV1 [2] and advances the state of the art in mobile visual identification, classifying object, object detection and semantic segmentation. MobileNetV2 [1] improves on the fundamentals principals of MobileNetV1 [2]. It used **depthwise separable convolution** as efficient building blocks. However, MobileNetV2 [1] was introduced two new features to the architecture. First one is **linear bottlenecks between the layers**, and second one is **shortcut connections between the bottlenecks**.

III. Project Objective(s)

To conduct all the experiments, we divide them into the following subtasks -

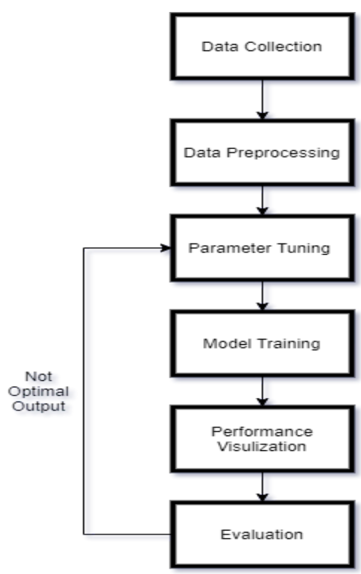


Fig. 1: Flowchart of subtasks.

Data Collection: We collect a suitable dataset from Kaggle which is good enough for garbage classification task.

Data Preprocessing: We preprocess our dataset before training the models. We resize the images and apply filters like - flipping, rotation in some of the experiments.

Tuning Hypermeters: We use four different models and tune the hyperparameters to get better outcomes and accuracy according to the models.

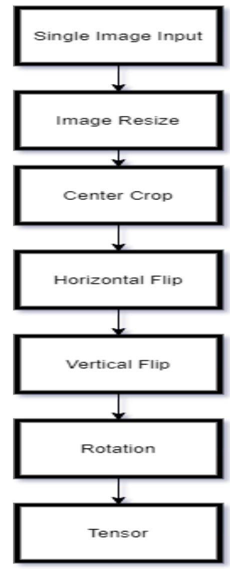


Fig. 2: Data Preprocessing.

Model Training: After tuning the hypermeters, we train four different models on this dataset separately. The models are : CNN, DenseNet121, ResNet50 and MobileNetV2.

Performance Visualization: We visualize the performance of this model while training. We observe training loss, validation accuracy while training and plot them in a graph.

Evaluation: After the training processes are completed, we test the performance of our models using test data. We evaluate the performance by calculating accuracy, precision, recall and f1 score and compare results. We also test the models with data outside of the dataset.

Below is an example of dummy input and output of our system -

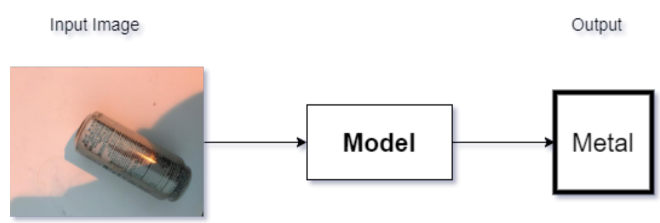


Fig. 3: Sample input and output.

IV. Methodologies

For our experiments, we use four different types of models. They are: **Simple CNN**, **DenseNet121**, **ResNet50** and **MobileNetV2**. We use pre-trained models for all the experiments except simple CNN.

Simple CNN : We design a simple CNN architecture to acquire a general inspection of the performance difference across models in order to analyze the performance of a basic CNN. To extract image features, this design employs 2D convolutional (conv.) layers.Filters of size 3X3 allow for more nonlinear activation function applications and have fewer parameters than larger filters [ref]. We add the max pooling layers between the 2D conv. layers to reduce the input dimensions and the amount of parameters to learn. This could help to preserve crucial features after converting layers and avoid overfitting. Here in Figure 4, we try to show a basic work flow of Simple CNN architecture.

TABLE I: Simple CNN Architecture

Layer Name	Output Shape	Param
Conv2D(3x3) - relu	300, 300, 32	896
MaxPooling2(2)	150, 150, 32	0
Conv2D(3x3) - relu	150, 150, 64	18496
MaxPooling2(2)	75, 75, 32	0
Conv2D(3x3) - relu	75, 75, 32	18464
MaxPooling2(2)	37, 37, 32	0
Flatten	43808	0
Dense	64	2803776
Dense(2)	6	390

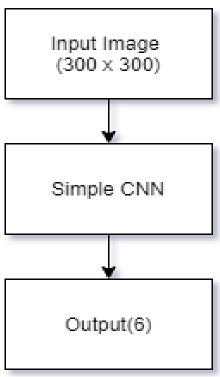


Fig. 4: Simple CNN workflow.

ResNet50 : It is a variation of the CNN architecture. Residual blocks are used in this architecture which tries to learn the residual part of the true output. In the last layer, after ResNet50, sigmoid is used as the activation function in our experiment. Here in Figure 5, a basic workflow of ResNet50 architecture is shown.

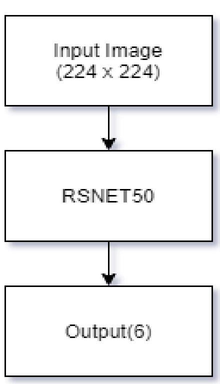


Fig. 5: ResNet50 workflow.

DenseNet121 : To implement DenseNet121 with this data set, firstly, we need to preprocess our dataset. For that we resize our images to 100 X 100 matrices. Then flipped the images horizontally and vertically. We keep the rotation range to 16. If we dive deep into DenseNet-121 architecture then we can see that in DenseNet-121 architecture there are **Convolution and Pooling layer**, **Transition layer**, **Classification layer**, **Dense layer**. Here in Figure 6, we show a simple work flow of DenseNet-121 architecture.

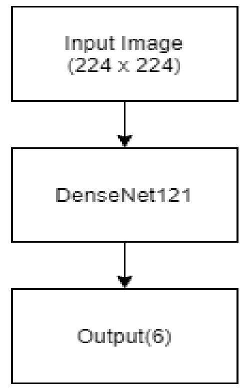


Fig. 6: DenseNet121 workflow.

MobileNetV2 : The architecture of this can be called as a modification of CNN architecture. Uses of inverted residuals and linear bottlenecks can be found in this architecture. A simple work flow is shown in Figure 7.

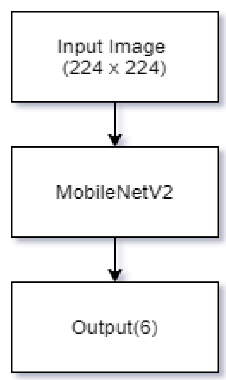


Fig. 7: MobileNetV2 workflow.

V. Experiments

In this section, we briefly discuss our experiments and results. We divide section this into 3 different subsections. They are **Dataset**, **Evaluation Matrices** and our **Experimental Result**.

A. Dataset

For our work, we use a popular garbage classification [6] dataset from Kaggle. It has total 2532 files. There are 2527 images in this dataset. These images are 384*512 pixels in size and fall into one of six recycling categories: cardboard, glass, metal, paper, plastic, and trash. In Table II we see an overview of this dataset -

TABLE II: Data Statistics

Class Name	Number of data
Cardboard	393
Glass	491
Metal	400
Paper	584
Plastic	472
Trash	127

Here are some of the sample images from the dataset -

Fig. 8: Data samples

B. Evaluation Metric

To evaluate our models, we calculate **accuracy**, **precision recall**, **f1-score** using our test data.

*a) **Accuracy**:* The accuracy is the ratio of properly predicted observations to all observations.

*b) **Precision**:* The ratio of accurately predicted positive observations to total expected positive observations is known as precision.

$$\frac{TruePositive}{TruePositive + FalsePositive}$$

*c) **Recall**:* Recall is the ratio of properly predicted positive observations to the total number of observations in the class is known as recall.

$$\frac{TruePositive}{TruePositive + FalseNegative}$$

*d) **F1-Score**:* F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives values.

$$\frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

C. Result

In this subsection, we see the training loss of different models on this dataset. We also see the change in validation accuracy per epoch. We compare the evaluation matrices here. We also show the confusion matrices. The results show that MobileNetV2 achieved the best performance among all the models used in all experiments while accomplishing 98% accuracy.

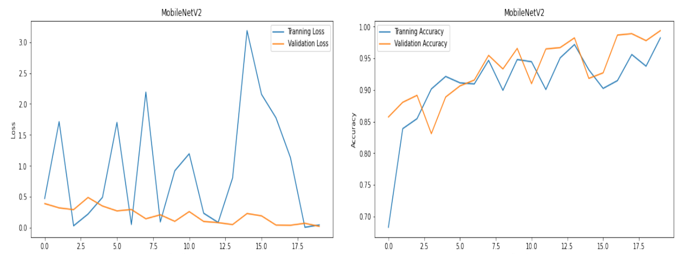


Fig. 9: MobileNetV2 Loss and Accuracy plot

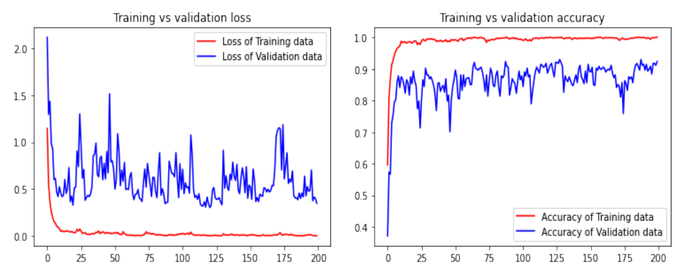


Fig. 10: DenseNet121 Loss and Accuracy plot

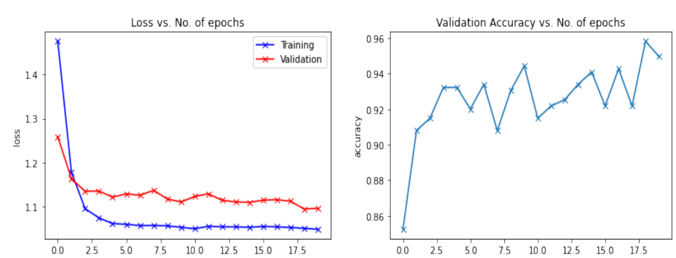
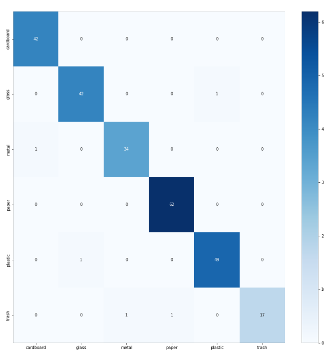


Fig. 11: ResNet50 Loss and Accuracy plot

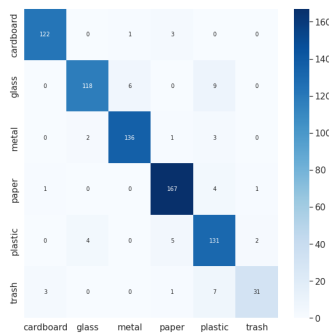
Fig. 12: Simple CNN Loss and Accuracy plot

TABLE III: Experimental Result

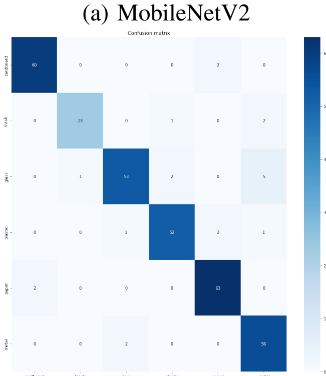
Model Name	Accuracy	Precision	Recall	F1-Score	Epoch
Simple CNN	0.76	0.76	0.76	0.75	100
ResNet50	0.93	0.93	0.91	0.92	20
DenseNet121	0.94	0.93	0.93	0.93	200
MobileNetV2	0.98	0.98	0.97	0.98	20



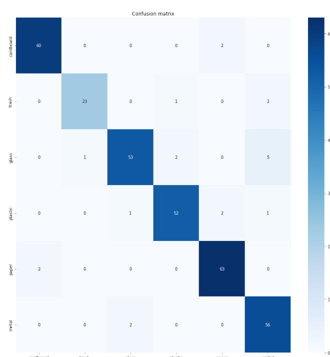
(a) MobileNetV2



(b) ResNet50



(c) DensNet121



(d) Simple CNN

Fig. 13: Confusion Matrix

VI. Conclusion And Future Works

Garbage classification is an important step in the waste management process. Our research shows that a variety of deep learning approaches may be utilized to categorize waste with great accuracy. We use a garbage image dataset available on Kaggle to train various CNN-based models in this work. Even though these models performed well, it is not certain that they will perform well in real-life circumstances because most of the photographs in the dataset had clean backgrounds. In future, we plan to train these models with a larger dataset with additional categories. We also wish to incorporate real-time garbage detection to make the garbage collection process more efficient.

References

[1] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017; arXiv:1704.04861.

[2] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks, 2018, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520; arXiv:1801.04381.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition, 2015; arXiv:1512.03385.

[4] Gao Huang, Zhuang Liu, Laurens van der Maaten and Kilian Q. Weinberger. Densely Connected Convolutional Networks, 2016; arXiv:1608.06993.

[5] S. Meng and W. -T. Chu, "A Study of Garbage Classification with Convolutional Neural Networks," 2020 Indo – Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), 2020, pp. 152-157, 10.1109/Indo-TaiwanICAN48429.2020.9181321.

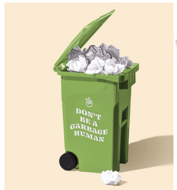
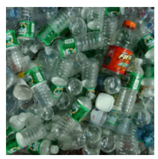
[6] The Garbage Classification Dataset contains 6 classifications. kaggle link https://www.kaggle.com/asdasdasasdas/garbage-classification

[7] Noble, William S, "What is a support vector machine?".journal - Nature biotechnology,publisher - Nature Publishing Group noble2006support

[8] Yang, Mindy and Thung, Gary "Classification of trash for recyclability status". CS229 Project Report noble2006support

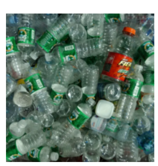
[9] S. Kaza, L. Yao, P. Bhada-Tata, and F. Van Woerden, What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. The World Bank, 2018.

TABLE IV: Some output of MobileNetV2

Input Image						
Predicted	**Trash**	**Trash**	**Metal**	**Metal**	**Cardboard**	**Cardboard**
Ground Truth	**Trash**	**Trash**	**Metal**	**Metal**	**Cardboard**	**Cardboard**
Input Image						
Predicted	**Plastic**	**Plastic**	**Glass**	**Glass**	**Paper**	**Paper**
Ground Truth	**Plastic**	**Plastic**	**Glass**	**Glass**	**Paper**	**Paper**

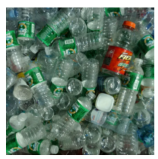
.

TABLE V: Some output of DenseNet121

Input Image						
Predicted	**Trash**	**Trash**	Paper	**Metal**	**Cardboard**	**Cardboard**
Ground Truth	**Trash**	**Trash**	**Metal**	**Metal**	**Cardboard**	**Cardboard**
Input Image						
Predicted	**Plastic**	**Plastic**	**Glass**	Plastic	**Paper**	**Paper**
Ground Truth	**Plastic**	**Plastic**	**Glass**	**Glass**	**Paper**	**Paper**

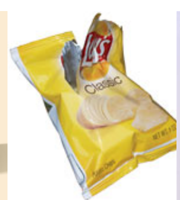
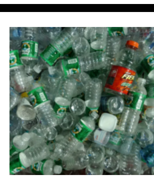
.

TABLE VI: Some output of ResNet50

Input Image						
Predicted	**Trash**	**Trash**	Metal	Glass	**Cardboard**	Metal
Ground Truth	**Trash**	**Trash**	**Metal**	**Metal**	**Cardboard**	**Cardboard**
Input Image						
Predicted	**Plastic**	**Plastic**	**Glass**	**Glass**	**Paper**	**Paper**
Ground Truth	**Plastic**	**Plastic**	**Glass**	**Glass**	**Paper**	**Paper**

.

TABLE VII: Some output of Sample CNN

Input Image						
Predicted	Trash	Paper	**Metal**	**Metal**	Glass	**Cardboard**
Ground Truth	**Trash**	**Trash**	**Metal**	**Metal**	**Cardboard**	**Cardboard**
Input Image						
Predicted	**Plastic**	**Plastic**	Glass	Trash	**Paper**	Metal
Ground Truth	**Plastic**	**Plastic**	**Glass**	**Glass**	**Paper**	**Paper**

.