Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-09571-5

Smartphone-based bulky waste classification using convolutional neural networks

Hanxiang Wang[1] · Yanfen Li[1] · L. Minh Dang[1] · Jaesung Ko[2] · Dongil Han[1] · Hyeonjoon Moon[1]

Received: 5 November 2019 / Revised: 3 August 2020 / Accepted: 6 August 2020
Published online: 11 August 2020

Abstract

The rapid urbanization process is escalating the urban waste problem, and ineffective management has worsened the issue, leading to severe consequences to the population health and economy. Although many countries have started to charge money for large household items, it is time-consuming and challenging for collectors to distinguish various types of bulky waste manually. As a result, this study introduces a mobile-based automatic bulky waste classification system. The original contributions include (1) a fine-tuned VGG-19 model is proposed to classify 95 types of bulky wastes; (2) three hybrid models are introduced to efficiently handle the imbalanced data problem, including class-weight VGG-19 (CW_VGG19), eXtreme Gradient Boosting VGG-19 (XGB_VGG19), and Light Gradient Boosting Machine VGG-19 (LGB_VGG19); (3) a large dataset that includes 95 classes, and each class contains over 500 images; and (4) the development of a mobile application that used the proposed model. Experiments show that the model obtained an accuracy of 86.19%, which outperforms existing models in classifying bulky waste. Moreover, the proposed hybrid models showed their robustness against imbalanced data under various scenarios.

Keywords Waste classification · CNN · Imbalanced data · XGB · LGB · Bulky waste

1 Introduction

Based on the United Nations statistics, about 60% of countries in the world emphasized their deep concern for managing waste and other environmental problems in the 2015 Earth Summit [23]. Therefore, it is crucial to reduce the environmental impact of waste through rational

✉ Hyeonjoon Moon
hmoon@sejong.ac.kr

1 Department of Computer Science and Engineering, Sejong University, Seoul, South Korea

2 Gatda corporation, 31 Baekbeom-ro 21-gil, Mapo-gu, Seoul, Republic of Korea

waste sorting and management. According to statistics in 2015, South Korea is one of the world's most densely populated nations with a density of 503 people per square kilometer [19, 26]. As a result, if ever-increasing garbage is not handled appropriately, severe environmental pollution is imminent, and the quality of life of the population will be significantly affected. For centuries, garbage disposal and management have been a thorny issue of great concern around the world. During the Middle Ages, in order to avoid the spread of diseases, governments took measures to remove garbage dumps as they were accelerating the increase of rat populations [1, 34]. The eighteenth century saw the start of recycling reusable materials, such as scrap metal, paper, and wood. Subsequently, modern industrial recycling began to appear in the nineteenth century [1, 24]. Through the end of the 1900s, with the improvement of environmental awareness, recycling has become an important part of waste disposal [1, 6]. Although many countries have promulgated relevant guidelines for waste classification, it is difficult for people to follow the guidance strictly, and human error is inevitable. Moreover, in the increasingly rapid pace of life, people tend to feel disdain when spending too much time sorting out and identifying all kinds of garbage. The traditional method is manual classification, which is inefficient and time-consuming because it depends mainly on human labor. Therefore, automatic recycled waste classification systems are necessary for waste companies to perform intelligent waste sorting.

Imbalanced data problem causes a high chance of overfitting during the deep learning training process and affects the overall performance of the model. As a result, many techniques have been proposed to tackle this problem, such as resampling techniques [37], cost-sensitive learning [7], and ensemble techniques [29]. The resampling data approach is implemented to balance the dataset by assigning similar samples for each class. This method is model agnostic, so many researchers applied this method. Another method called the cost-sensitive learning approach forces the classifier to focus on the minority classes by setting a weight for each class. The ensemble-based method combines the results from different models to obtain the final label [29]. Based on the collected dataset, cost-sensitive method, and ensemble-based method are used to handle the imbalanced data problem.

To put forward an efficient and accurate classification method, there is a lot of research that has been proposed to classify garbage. They achieved good performance both in computation time and recognition rate. The most common approach is to apply the transfer learning technique on previous state-of-the-art deep learning models using a new dataset [1, 25, 36]. However, most of the datasets were small and covered a limited number of categories. Thus, these datasets cannot be applied to perform garbage classification in real daily life.

In this research, two suitable ways were adopted to deal with the issue, and the specific details are given in Section 4.5.

With the proposed system, we answer the following questions, using the experimental result that we have obtained:

1. What is the most suitable model on the waste dataset among the three CNN models?
2. How do the two proposed solutions for the imbalanced data problem perform under the configuration of different ratios?
3. What is the performance (computational speed and classification accuracy) of the final model in the developed application?

By answering the questions, the fundamental contributions are listed below:

1. By observing the experiment results, the fine-tuned VGG-19 model worked well both in computation speed and classification results.
2. The two solutions proposed can perform well on the collected dataset, but the XGBoost method was particularly superior to the other method.
3. The final model obtained a high Top-1 accuracy of 86.19% and Top-4 accuracy of 97.01%, which outperforms existing models in classifying bulky waste.
4. A large bulky waste dataset that contains 95 classes was collected and validated manually.

The general structure of the paper is organized as follows: Section 2 analyzes some related work on waste classification. Some details of the collected dataset are introduced in the third part. In Section 4, the developed classification application is illustrated. After that, the proposed methodology is expounded in Section 5. Section 6 provides details of experimental results. In Section 7, the conclusion based on this study is given.

2 Related work

2.1 Waste classification

With the severity of environmental pollution, waste sorting has become a particularly popular research topic. Various types of datasets were created for the waste classification task. For example, a 6-classes Trashnet dataset was created for garbage classification, including glass, paper, metal, plastic, cardboard, and trash [36]. In the research, a traditional machine learning algorithm SVM was compared with a CNN model. The experiments showed that the SVM classifier achieved a 63% classification accuracy, and the CNN model obtained a 22% accuracy. Both performances of the two approaches are inferior. Besides, the VN-trash dataset was collected, including Organic, Inorganic, and Medical wastes [33]. The proposed model named DNN-TC performed well on two kinds of datasets. In addition, a dataset called Labeled Waste in the Wild was created for waste detection and classification [28]. Experimental results showed that compared with some deep learning methods, the proposed model obtained better performance on the 19-class dataset. Different from previous datasets, the proposed dataset is about bulky recycled waste. To the best of our knowledge, there are no publicly available datasets that are similar to it.

After the dataset collection process, various classification technologies have been used for waste classification, including the spectral imaging [14], thermal imaging [9], and deep learning-based method [5]. Compared with the deep learning method, the other technologies require expensive equipment and intricate process. Currently, some researchers have used deep learning architecture to intelligently classify garbage waste instead of employing humans as workers to identify waste. For example, Bircanoğlu, Atay et al. compared the performances of various models on the TrashNet dataset [1, 36]. The result showed the most suitable model was the fine-tuned DenseNet model, whose accuracy was up to 95%, but the prediction time was a little slow. The final optimized model reduced about 3 million parameters for real-time classification. Similarly, Ruiz, Sánchez et al. evaluated several CNN architectures for the waste classification [25]. In their work, the highest accuracy (88.66%) was obtained using a combined CNN model based on the TrashNet dataset. For previous works, the best model can achieve a good classification accuracy based on a 6-classes dataset through the comparison of various deep learning models. However, with the increase of garbage types, the classification performance of the model needs to be further verified.

2.2 Imbalanced data problem

During the last few years, several methods were introduced to efficiently deal with the imbalanced data problem. For example, in a cost-sensitive learning-based research, three classification costs were combined to decrease the classification cost for the imbalanced gene expression data. However, it is still hard to determine cost representation [17]. Similarly, an uncorrelated cost-sensitive multiset learning (UCML) approach was proposed to handle the imbalanced binary classification problem. Multiple balanced subsets were partitioned, and the discriminant features were learned from multiset by using the multiset feature learning [12]. The method was proposed to solve the binary classification, which is not suitable for the 95-class bulky waste dataset used in this research.

With a different approach from the cost-sensitive learning, Sun, Song et al. proposed an ensemble method to solve the imbalanced data issue [29]. In this method, the imbalanced dataset is converted into multiple balanced datasets and then constructs several classifiers on the converted data. Finally, a certain ensemble principle is used to integrate the classification results of various classifiers. Although the method based on ensemble learning performed well, there remained a problem with effectively applying a diversity of classification ensembles.

Based on previous research, a system was designed to classify waste by using a deep learning model. After that, the system can be installed on a mobile phone, which has a high classification accuracy with 95 classes.

In this research, a bulky waste classification system with 95 types of garbage is proposed to cope with some of the significant weaknesses of previous approaches. The imbalanced data problem inevitably appears in the collected dataset, which brings great challenges to the classification task. As a result, two different methods are implemented to solve the imbalanced data problem. The experiment shows that both of them greatly improve model performance in imbalanced settings. Besides, the proposed model is integrated into a mobile application to show that the model can be deployed to perform waste classification in real life. In this application, the client/server (C/S) structure was applied, in which each client can send a request to a server and obtain the response. The method used in this research should be a breakthrough for the classification performance with so many items.

3 Bulky waste classification

3.1 Dataset

As the neural network models automatically extract features from the data and learn the abstract representation of the problem from the extracted features, the performance of CNN models significantly depends on the dataset. The deeper the model becomes or the more complex the problems, the more data are required for training. Otherwise, the overfitting phenomenon likely occurs during the training process.

In this stage, the dataset was collected from different websites such as Yahoo, Bing, Google, Baidu, and Naver. And then, the data validation process was done manually. All images are stored as 224 × 224 colored images. Since there are 95 classes in total, and some of the classes are not common, the data acquisition process takes lots of time during the whole process.

Among 95 classes, there are some general classes such as dishwasher, armoire, and air conditioner, which can be found on any website. However, there are some special object categories that only appear in Korea as they belong to traditional Korea culture. Thus, it is challenging to search them on other search engines except the Korean search engine (Naver). Moreover, images belong to antique object categories is also hard to find from major websites or blogs. Therefore, the amount of data in each category is extremely different and even leads to the imbalanced data problem. There are a total of 69,737 images in the proposed dataset, and the number of images ranges between 500 to 3470 for each class. Figure 1 shows some of the sample images taken from the collected dataset. All classes are labelled with the corresponding name of each type of waste. The data was divided into two separate sets: the training set (62,763) and testing set (6974). The training set comprises 90% of the whole dataset and is used to get corresponding weights for the deep neural network. The testing set consists of 10% of images and is used on the model. Table 1 shows the database configuration details for each class, and 'No.' refers to the number of images for each class.

3.2 Waste classification system

In this section, the proposed waste classification system is described in detail. It is implemented by performing the transfer learning technique on previous state-of-the-art deep learning models. Figure 2 describes the overall architecture of the proposed framework. After conducting several experiments, VGG19 achieved the highest performance, so it is considered the most suitable model. And then, the structure of the VGG-19 model is optimized to achieve robust performance. After that, the imbalanced data problem is solved by using three different algorithms. For the testing of the waste classification system, there are two main parts, including client and server. Customers use the application to get important information (including waste categories and prices) through the smartphone by taking waste pictures and send to the server. After receiving images sent from the customers, the server feeds the images

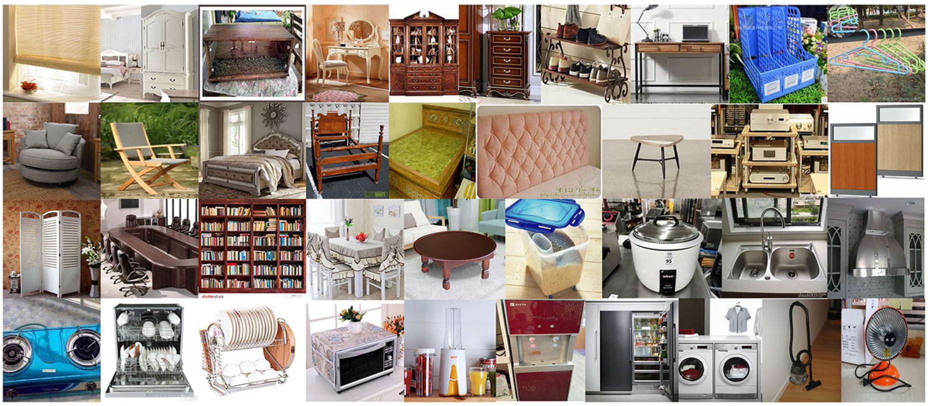

Fig. 1 Sample images from the collected dataset

Table 1 Description of 95 bulky waste classes with the total number of collected images for each class. 'No' means the total number of images for each class

Index	Name	No.	Index	Name	No.	Index	Name	No.	Index	Name	No.	Index	Name	No.	Index	Name	No.
1	Armoire	855	17	Bed frame	500	33	Ricer box	530	49	Air cleaner	828	65	Air conditioner	755	81	Clock	595
2	Mungab	506	18	Stone bed	660	34	Rice cooker	865	50	TV	639	66	Electric fan	677	82	Fish tank	1402
3	Dresser	640	19	Bed rest	519	35	Washing table	504	51	Audio set	500	67	Stove	504	83	Water tank	1182
4	Display stand	813	20	Latex	840	36	Kitchen ventilator	984	52	Tape	511	68	Boiler	792	84	Basin	703
5	Drawer	3225	21	Small desk	577	37	Dishwasher	855	53	Video player	604	69	Boiler oil tank	661	85	Bathtub	528
6	Small bookshelf	542	22	TV cabinets	962	38	Dish dryer	532	54	Computer	987	70	Heater	508	86	Toilet	777
7	Shoes-shelf	549	23	Audio set cabinet	699	39	Dish rack	539	55	Duplicator	597	71	Piano	528	87	Toilet seat	504
8	Desk	660	24	Low partition	511	40	Gas stove	726	56	Printer	500	72	Electric piano	532	88	Cloth closet	516
9	Book stand	538	25	Desk	714	41	Microwave	605	57	Fax machine	532	73	Bicycle	625	89	Wreath	524
10	Cabinet	697	26	High partition	502	42	Blender	509	58	Game machine	500	74	Fitness equipment	770	90	Pallet	500
11	File cabinet	600	27	Desk drawer	590	43	Water purifier	663	59	Large game machine	511	75	Baby stroller	644	91	Door	538
12	Hanger	582	28	Meeting desk	815	44	Ice box	500	60	Vendor	742	76	Mat	662	92	Store sign	521
13	Sofa	642	29	Bookshelf	531	45	Refrigerator	768	61	Humidifier	776	77	Pad	500	93	Waste wood	501
14	Chair	689	30	Dining table	518	46	Washing machine	707	62	Electric blanket	612	78	Carpet	819	94	wheelchair	2197
15	Bed	713	31	Marble table	933	47	Direct fired dryer	579	63	Karaoke machine	746	79	Luggage	603	95	Window blind	1938
16	Mattress	565	32	Gyoja table	539	48	Vacuum cleaner	502	64	Iron	582	80	Mirror	3470			

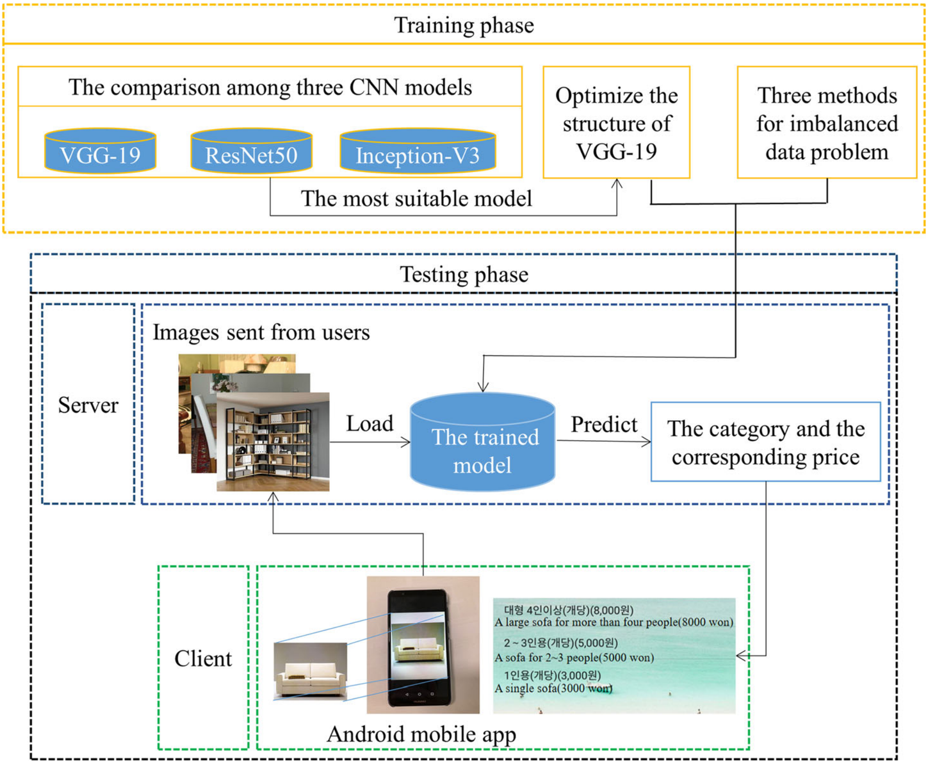

Fig. 2 Flowchart of the proposed framework. In the training phase, the structure of the VGG-19 model is selected and optimized, and the imbalanced data problem is solved. In the testing phase, by receiving the image from the client, the trained model is loaded, and the predicted result is returned to the client

into the trained model. Finally, the predicted results are returned to the user mobile application interface.

Figure 3 presents important functions of three main interfaces of the mobile application. The first interface has an image frame and two buttons, one is for taking photos, and the other allows users to select a photo from the local album. The photo function requests the phone camera to take waste pictures. After users take a real-time photo or select a suitable picture from the local album, the photo is resized automatically to fit the size of the image frame. At the same time, the application pops up the prompt whether to upload or not and then users can upload pictures to the server by the socket method. The socket has the advantages of fast data transmission, high performance, and strong security [22, 35]. The characteristics of this method determine that it is suitable for real-time information interaction between client and server. After users confirm and upload, a button that displays the results pops up below the interface. Click on this button to automatically jump to the second interface and show the classification results. The results shown here are the class names of the top four possibilities and their corresponding prices. In order to avoid the worst situation in object classification, the last interface was designed. If the correct item is not in the forecast list, click the corresponding button to enter the last interface. It shows all categories and the corresponding prices. The main interfaces of the classification application are shown as follows.

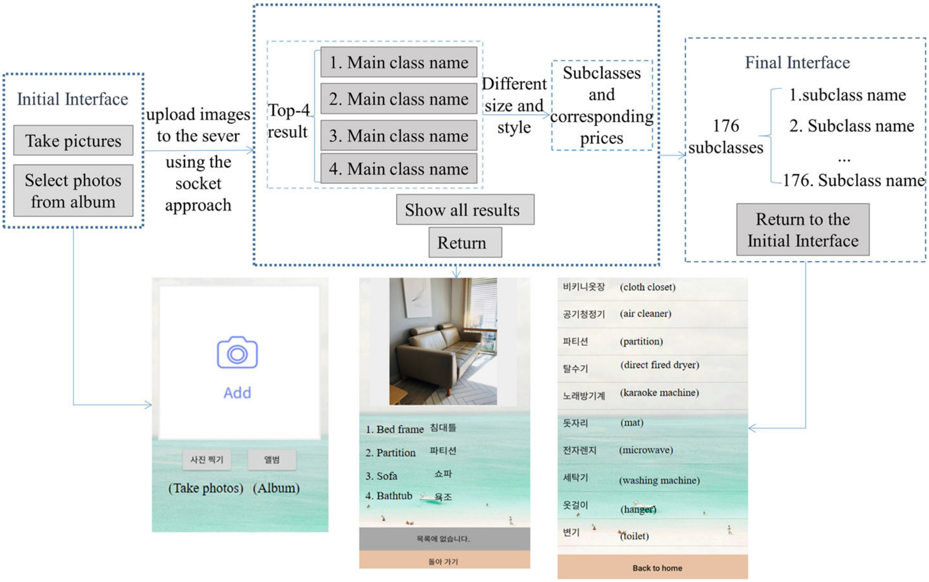

Fig. 3 Introduction for main interfaces (initial interface, second interface and final interface) of the mobile application

4 Methodology

4.1 CNN models

As a kind of deep neural network, CNN can obtain more accurate classification results and faster computational speed than some machine learning methods (such as KNN, SVM, and decision tree algorithm) in most classification tasks [20]. In the structure of CNN, the core idea of the convolutional layer is local receptive field and weight sharing, which can simplify the network parameters and give the network a certain degree of displacement, scale, scaling, and nonlinear deformation stability. The pooling layer is generally placed between successive convolutional layers, which can gradually reduce the spatial size of the data and decrease resource consumption. The fully connected layer brings together all the learned features and acts as a “classifier” throughout the network.

The softmax structure is a vital part of the multi-classification problem of CNNs [16]. It is generally applied at the end of the model to solve multi-classification problems. This algorithm can help us obtain the specific probability values of the input feature vectors belonging to a certain class. The output of softmax is a set of probability values with a sum of 1, which is more intuitive than a set of values belonging to a range of $(-\infty, +\infty)$. Softmax is defined as:

$$SoftMax(si) = \frac{e^{si}}{\sum_{j=1}^{N} e^{sj}} (i = 1, 2, \ldots, N) \tag{1}$$

The S_i in the formula (1) represents the score of the model on the i-th category of the input feature vector, and the value range of the score value changes from $(-\infty, +\infty)$ to $(0, +\infty)$ under the action of the e-index. The operation of the denominator summation is used to complete the normalization and make the sum have a value of 1, so that a set of probability values can be gotten, which is involved in calculating the loss function, and then use the gradient descent algorithm to adjust the weight parameters to carry out the next round of training.

In this paper, three prestigious CNN structures were used in the training process, including VGG-19, ResNet50, and Inception-V3. Compared with other models, these three models with a complicated architecture have achieved state-of-the-art performance based on the large-scale dataset in different applications recently [27, 31]. Moreover, as mentioned in Section 2, there are some recent works using these deep learning models on waste classification [1, 11, 25]. In their work, different models were compared and evaluated, and then good results were obtained based on the most suitable models.

4.1.1 VGG-19 model

VGGNet won second place in the classification task in the ILSVRC-2014 competition [27]. The main thought is to use a small convolution to increase the depth of the network and effectively improve the model's effect, and VGGNet has a good generalization ability for other data sets besides ImageNet. In the training period, the input image size of the first convolutional layer is 224×224. Each convolutional layer uses a small receptive field (3×3) with a convolution step of 1 pixel and a convolutional layer filled with 1 pixel to ensure that each active map retains the same space size as the previous layer. Each time the convolutional layer is passed, a Rectified Linear unit (Relu) activation is performed, which makes the output of the convolution layer nonlinear to increase the training difficulty of the neural network. The maximum pooling layer has a kernel size of 2×2 and a step size of 1 pixel to reduce the spatial dimension of the output. The three fully connected layers of the original pre-trained model are responsible for the final classification based on the ImageNet dataset, and their channel numbers are 4096, 4096, and 1000, respectively.

4.1.2 ResNet50 model

In 2015, ResNet was first proposed by Kaiming He and obtained great achievement in the ImageNet competition. The residual block can form a very deep CNN and solve the gradient degradation problem. In addition to simply stacking the convolutional layers, Resnet uses the Bottleneck network structure to speed up the training. This structure includes two 1×1 convolutional layers, and the first one is used to reduce the computational complexity by decreasing the dimension. The second is used to increase the dimension to make it the same as the input dimension for identity mapping [10]. In the paper, Resnet50 is selected as a classic model of ResNet and compared with other models. It is also one of the residual networks frequently used by many researchers. This model has five sets of convolutional structures. The first set of convolutions has an input image size of 224×224, and the last set has an output of 7×7. A softmax layer is used for the final classification after the convolutional layer.

4.1.3 Inception-V3 model

GoogLeNet achieved outstanding performance in the ILSVRC 2014 competition, which was the first time that the Inception model appeared. This model can decrease the computational complexity and the amount of parameters while ensuring good classification accuracy. This is mainly on account of the use of a special Inception Module based on the Hebbian principle. In the Inception Module of Inception-V1, small convolutions such as 1×1, 3×3, and 5×5 are used to connect highly relevant nodes to complete the construction of the Hebbian principle structure [30]. Inception-V3 is used as one of the frequently occurring CNNs in this paper. Its significant improvement is to decompose the convolution layer, that is, to decompose the size of N x N convolution kernel into the size of 1 x N and N × 1 convolution kernels. This can not only speed up the calculation of the model but also deepen the depth of the network, thus enhancing the nonlinear fitting ability of the neural network. In addition, the input image size of the first layer of the convolutional layer of Inception-V3 is different from other models, and its network input has changed from 224×224 to 299×299 [31].

4.2 Fine-tuning process

Since the size and shape of the collected datasets are somewhat different from the ImageNet dataset, and the first few convolutional layers are used to learn some more general features. All the convolutional layers of the pre-training model were initialized, and the parameters of the first few convolutional layers were frozen.

The red box in Fig. 4 shows the adjustment for the model in the fine-tuning process. Two convolution layers with the convolution kernel size of 3×3 were added between the pooling layer and the convolution layer of the last convolution block. After that, the number of convolutional layer is changed from 4 to 6 in convolutional block 5. Since little detail of image features was obtained, the recognition accuracy of the original training model is not good before the modification. In the neural network, the later convolution layers are responsible for obtaining more details. Therefore, the idea was verified by increasing the number of layers of the network appropriately. Batch normalization is a method to optimize neural networks, and the main purpose is to unify the scattered data, avoid the problem of gradient disappearance or explosion, and increase the robustness of the network. This technology has been proven to significantly improve training performance and is now a standard component of many of the most advanced networks [18]. Thus, in this phase, the batch normalization layer was added between two full connection layers, as shown in Fig. 4. After the normalization operation, the correct rate was higher than the original model.

4.3 Imbalanced data problem

In most multi-class image classification tasks, there are some classes that are abundant and easy to collect, whereas there remain many classes that are rare and difficult to collect, which leads to the problem of data imbalance. In this study, the number of images for each category is different among 95 main target categories. Table 2 shows the imbalanced data problem in some classes of the collected dataset. Through the table, the most severe imbalanced ratio is 1:7 (game machine: mirror).

In practical, if the number of some classes in the training set is much lower than that of other classes, it will force the classification results to be biased towards the number of classes [21]. In this paper, two different methods based on a cost-sensitive learning approach and ensemble-based approach were adopted to solve the imbalanced data problem, as shown in Fig. 5. One is to set different training weights for the different number of categories in the training model [4], and the other is to combine a CNN model with an Extreme Gradient Boosting algorithm [2] or Light Gradient Boosting Machine (LightGBM) algorithm [13] to

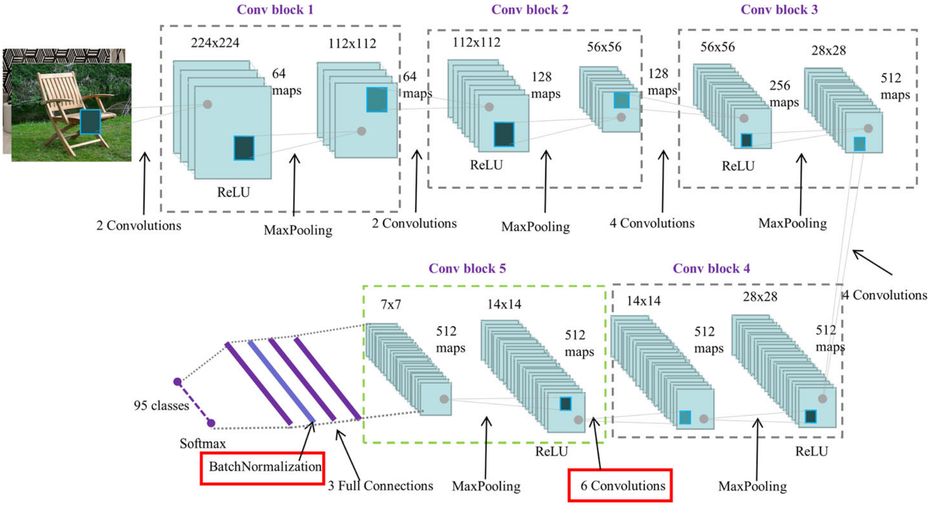

Fig. 4 Structure of fine-tuned VGG-19 model, the structure within the gray dashed box is the frozen layer structure, and the layers within the green dashed box are trainable. Two convolutional layers are added in the fifth convolutional block, as well as, the batch normalization layer is added between the second and third full connection layers

address the problem of imbalanced data. In most classification tasks, classification accuracy is not the only criterion for evaluating models. For instance, when categories are unbalanced, accuracy is confusing and meaningless. In this study, a confusion matrix, ROC curve, and AUC value were used to estimate the classifier performance.

4.3.1 Cost-sensitive learning approach

As for a cost-sensitive learning method, the classification performance can be improved by setting different weights for each category according to the number of sample images in the training process [7]. Weight can also be a called penalty item, which can correspondingly magnify or reduce a certain class of loss in the training process. Weight is obtained by calculating the occurrence frequency of a certain class in the whole dataset. If the weight of a certain class is large, then the loss of this class is enlarged. The model will pay more attention to this class in the process of feature extraction. Correspondingly, for some classes with a large number of images, a smaller weight was set to reduce the model’s learning for this class. The training weight will only magnify or reduce the loss in reverse learning to indirectly affect the direction of learning.

Table 2 Several samples for different ratios between imbalanced classes

Class1	Number of images	Class2	Number of images	Ratio
Game machine	500	Mirror	3470	1:6.94
Mungab	500	Drawer	3225	1:6.45
Bed frame	500	Wheelchair	2197	1:4.30
Ice box	500	Window blind	1938	1:3.90
Pad	500	Water tank	1182	1:2.36

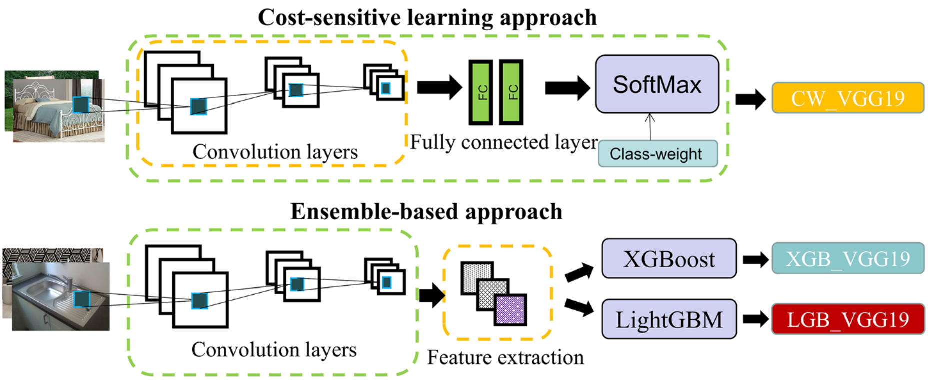

Fig. 5 Two methods for imbalanced data problem. The first method reduces the impact of imbalanced data by setting weight parameters (CW_VGG19), whereas features extracted from the VGG-19 model were fed to the XGBoost classifier (XGB_VGG19) and LightGBM classifier (LGB_VGG19), respectively to deal with the imbalanced data

4.3.2 Ensemble-based approach: XGBoost classifier

The basic rule of the Boosting algorithm is to adjust the sample distribution according to the learning performance so that the samples with poor performance get more attention during the training of the next base learner with the adjusted samples iteratively until the number of base learners reaches the specified number. The Gradient Boosting algorithm (GB algorithm) [8] uses gradient calculation to regress fitting residuals based on the traditional Boosting algorithm. It then generates a new base learner based on the residuals and calculates the optimal overlapping weighted weight.

The XGBoost algorithm [2] was first proposed by Chen T in 2014 as one of the efficient implementations of the GB algorithm, and the regularization term was introduced into the loss function of the base learner to control overfitting in the training process. Secondly, the XGBoost algorithm has done a lot of optimization processing in computing mode. It supports parallel computing and sparse training data processing, making it more efficient than other gradient boosting approaches.

$$Obj = \sum_{i=1}^{n} L\left(yi, \hat{yi}\right) + \sum_{k=1}^{K} \Omega(fk) \tag{2}$$

The objective function has two parts. The first part is a loss function L, which is used to calculate the gap between the predicted result $\hat{yi}$ and the target result yi. The other part is the regularization component Ω, which is used to manage the overfitting problem of the model by adjusting the number of leaf nodes, the scores of each leaf node, and the prediction threshold. XGBoost uses the same idea of the greedy algorithm as the CART regression tree [8] to traverse the feature partition points of all features, but XGBoost uses different objective functions as evaluation functions.

4.3.3 Ensemble-based approach: LightGBM classifier

In this study, LightGBM is used as a neural network classifier to deal with unbalanced data. LightGBM pays more attention to the training speed of the model based on the integrated learning method of XGBoost. The main idea of the algorithm is to apply Gradient-based One-

Side Sampling (GOSS), and Exclusive Feature Bundling (EFB), two technologies to solve the problem of XGBoost repeated scanning data [13]. GOSS reduces the computational complexity by removing some data with smaller gradients that are not conducive to calculating information gain. The EFB algorithm allows features that are not independent of each other to be fused and bound to reduce the number of features. Moreover, the LightGBM algorithm accelerates its convergence rate by applying the histogram-based algorithm, weighted quantile sketch algorithm, and leaf-wise strategy with max depth limitation to handle large datasets better.

5 Experimental result

In this section, firstly, important system specifications information is described in Table 3. The C/S structure was used in this research, which means two parts (server and client) were included in the whole process. Secondly, the optimal value for each parameter was obtained in each CNN model, and the classification accuracy and those running speeds are compared to get the best model. Thirdly, the structure of the selected model is modified to improve the final classification accuracy. After that, the imbalanced data problem was solved by using two different methods. Finally, the result of the developed application is shown and explained.

5.1 Comparison of different CNN models

During the estimation process, different parameters were changed to get better accuracy, and after that, the optimal value for each parameter was obtained. As shown in Table 4, several important parameters were selected, such as learning rate, optimizer, momentum, and iterations, and the responding accuracy for each model was also shown in the table. If accuracy is regarded as a vital factor in this research, then the fine-tuned VGG-19 is the optimal model for the collected dataset, and it achieved the highest accuracy at 86.19%.

In the task of object recognition, besides recognition accuracy, the recognition speed is also a standard of measurement that cannot be ignored. The speed of recognition directly affects whether the user has a good application experience. For the computation complexity, the total parameters for each model and running time for each iteration are shown in Fig. 6. It can be seen that the fine-tuned VGG-19 model has the least parameters, and the running speed is not slow among the three models. In addition, the total parameters of the Inception-V3 model are less than the amount parameters of the ResNet50 model, and the running speed of the

Table 3 Specification information of the system including server and client

	Name	Specification
Sever	GPU	NVIDIA PACAL TITAN Xp GDDR5 12GB PCI-express
	Processor	Intel® Core™ i7-5820 K (6Core / 3GHz / 15 MB)
	RAM	DDR4 16GB PC4–17000
	HDD	SATA (serial-ATA) 4 TB / 7200 RPM Enterprise
	Library	Tensorflow/Keras Library
	OS	Ubuntu 14.04
Client	Devices	HUAWEI nova lite 2
	CPU	Hisilicon Kirin 659
	Version	Android 8.0.0

Table 4 Optimal parameter setting and the responding accuracy for three different CNN models

Model	Learning rate	Optimizer	Momentum	Iterations	Accuracy (%)
Fine-tuned VGG-19	0.001	SGD (Stochastic gradient descent)	0.9	1764	86.19
Inception-V3	0.0001	Adagrad	–	1000	81.15
ResNet50	0.0001	Adam	–	441	79.63

Inception-V3 model is the slowest on the collected dataset. The reason why Inception-V3 is slower than ResNet50 may be that the number of freezing layers is the same for both models. However, Inception V3 has a deeper layer structure than ResNet50, so most of the layers need more time to participate in testing, and the ResNet50 model contains a lot of short-cut structure, which reduces the amount of computation.

According to the classification accuracy and computation complexity of each model, the most suitable model for the collected dataset can be selected. From the analysis of experimental results, the fine-tuned VGG-19 model trained the dataset and achieved robust performance both on accuracy and speed.

5.2 Performance of the fine-tuned VGG-19 model

After the estimation process, the fine-tuned VGG-19 was selected as the target model to train the collected dataset, and the average accuracy for 95 classes was 86.19%. Due to the huge number of classes, the class name was replaced with the class index to represent each class, and the full class name for each class can be checked in Table 1.

As shown in Table 5, most of the classes (89) achieved high accuracy of over 90%. However, there are 6 classes with extremely low accuracy of below 60% (highlighted in Table 5). By analyzing the classes with low accuracy, it is discovered that these object categories are challenging because they are similar to each other, for example, Mungab (a traditional Korean cabinet), Drawer, Cabinet, and Desk drawer.

Sample images of those classes are shown in Fig. 7. Among these four classes, it is extremely similar both on the structure and style.

To handle the problem, the Top-4 result was applied to get higher accuracy [15]. More choices can be generated by using this way so that some classes with high similarity can be easily classified, and users can choose whatever they want. The Top-4 result greatly improves the overall average recognition accuracy, from 86.19% to 97.01%. Table 6 shows the Top-4 results for each category (the CNN model produces four most likely results for each input).

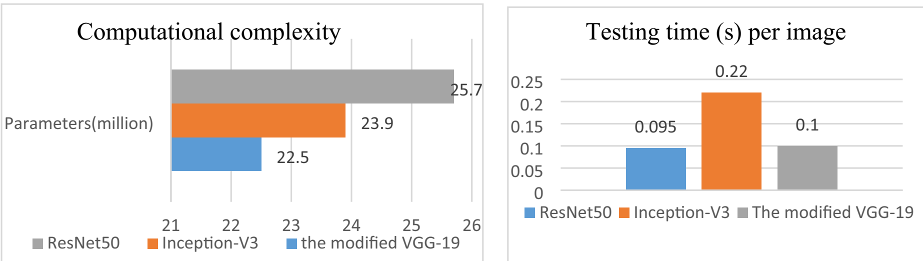

Fig. 6 Computational complexity of three CNN models (fine-tuned VGG-19, Inception-V3 and ResNet50)

Table 5 Top-1 accuracy for 95 classes (89 classes have an accuracy rate of over 90%, and 6 classes have an accuracy rate of below 60%)

Class	Accuracy (%)	Class	Accuracy (%)	Class	Accuracy (%)	Class	Accuracy (%)	Class	Accuracy (%)
1	96.2	20	79.8	39	96	58	91.7	77	92
2	**44**	21	80.7	40	98.6	59	96.1	78	88.9
3	90.2	22	86	41	93.3	60	97.3	79	86.4
4	85.5	23	81.2	42	98	61	85.7	80	97.3
5	**35.6**	24	90.2	43	92.3	62	85.1	81	98.3
6	89.8	25	70.4	44	88	63	79.5	82	92.6
7	73.9	26	87.5	45	89	64	94.8	83	87.6
8	80.3	27	**59.3**	46	92.9	65	93.3	84	92.8
9	78.3	28	75.3	47	84.2	66	94	85	84.3
10	**40.6**	29	89.1	48	86	67	92.5	86	96.1
11	74	30	80.4	49	92.7	68	94.7	87	74
12	90.9	31	79.6	50	85.7	69	90.9	88	94.1
13	96.8	32	83	51	73.7	70	62.2	89	96.2
14	90.9	33	84.4	52	95	71	92.2	90	98.2
15	81.3	34	100	53	93.3	72	98	91	84.6
16	80.4	35	96	54	92.9	73	98.4	92	94.2
17	73.2	36	100	55	100	74	94.8	93	100
18	98.4	37	**54.1**	56	84.7	75	96.9	94	98.6
19	98	38	**5.7**	57	100	76	78.1	95	98.3

5.3 Imbalanced data problem (IDP)

In this study, two methods were used to solve the IDP. As mentioned in Section 4.5, one is to set different weights for each class with different numbers. The other is to use the XGBoost

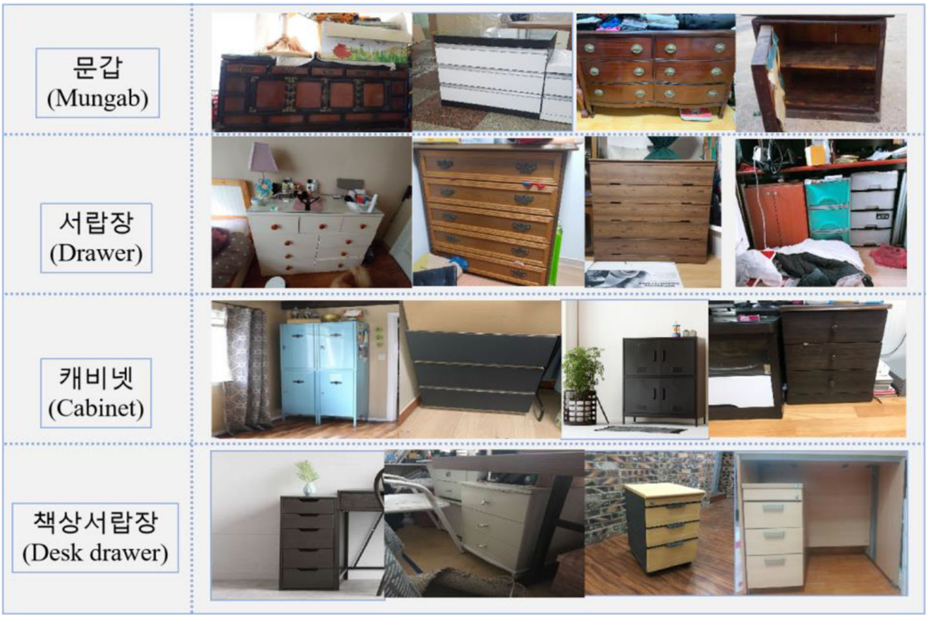

Fig. 7 Sample images of similar classes (mungab, drawer, cabinet and desk drawer)

Table 6 Top-4 accuracy for 95 classes (All classes have an accuracy rate of over 87%)

Class	Accuracy (%)	Class	Accuracy (%)	Class	Accuracy (%)	Class	Accuracy (%)	Class	Accuracy (%)
1	98.7	20	97.6	39	96	58	98	77	98
2	88	21	96.5	40	100	59	100	78	100
3	98.4	22	97.7	41	100	60	100	79	98
4	94.7	23	94.2	42	98	61	94.8	80	100
5	89	24	96.1	43	98	62	96	81	100
6	98	25	90.1	44	96	63	95	82	100
7	89.1	26	100	45	98.6	64	100	83	94
8	93.9	27	94.9	46	97	65	97.3	84	100
9	87	28	90.1	47	98	66	97	85	100
10	88.4	29	98.2	48	90	67	98	86	100
11	98	30	98	49	98.8	68	99	87	100
12	96.4	31	96.8	50	93.7	69	97	88	98
13	95.4	32	92.5	51	89	70	84	89	100
14	97	33	95.6	52	100	71	100	90	98
15	98.4	34	100	53	98	72	98	91	96.2
16	100	35	98	54	99	73	98.4	92	100
17	94.6	36	100	55	100	74	99	93	100
18	100	37	100	56	98	75	100	94	100
19	100	38	98	57	100	76	95	95	100

algorithm or LightGBM algorithm for classification training. Various experiments are implemented to prove the improvements made using these two methods and which method has a greater impact on the collected dataset. The fine-tuned VGG-19 model was proved to be the most suitable architecture in both accuracy and computational time through the previous comparison for different CNN models. Thus, two different methods for solving IDP are used in the final model. As for imbalanced data, two specific classes (mungab and drawer) were selected, and there exists an obvious IDP on these two classes. Based on the collected images, different ratios were set for two classes, as shown in Table 7.

By setting weights for each class and using the XGBoost / LightGBM classifier, the problem of imbalanced data was solved. For the first way, the prior probability of each type of data is used to calculate their corresponding weights. And then, the weight parameter was used to bring the training weight into the final model. For the second way, the whole model is divided into two parts, including the feature extraction part and the resampling part. All convolution layers and pooling layers are retained as the first part. Through CNN, various kinds of features can be extracted from each layer. The features in the layer structure closest to the output layer contain more details. After that, all extracted features are trained in the XGBoost / LightGBM classifier of the second part [3]. Figure 8 shows the AUC values of four models (three novel models and the fine-tuned VGG-19 model) on the imbalanced dataset. In this paper, three novel models are called class-weight VGG-19 (CW_VGG19),

Table 7 Number of mungab and drawer images on various balancing ratios

	1:1	1:2	1:3	1:4	1:5	1:6	1:7	1:8	1:9	1:10	1:100
Mungab	200	200	200	200	200	200	200	200	200	200	30
Drawer	200	400	600	800	1000	1200	1400	1600	1800	2000	3000

eXtreme Gradient Boosting VGG-19 (XGB_VGG19), and Light Gradient Boosting Machine VGG-19 (LGB_VGG19), respectively.

As shown in Fig. 8, when the balancing factors are equal to 1:1 and 1:2, the AUC values of the three models are extremely similar. However, only the AUC value of the VGG-19 model declines dramatically to 0.63 when the balancing factor is 1:100. It also can be seen that the proposed three models were not significantly affected by the IDP when the balancing factor changed significantly. Compared with other models, the LGB_VGG19 model achieved better results in terms of AUC when the imbalance ratio is 1:100 and 1:10. In addition, when the balancing factor increases to 1:8 or more, XGB_VGG19 performs better than CW_VGG19. In particular, when the balancing factor is equal to 1:100, the XGB_VGG19 model can get 0.91 AUC, whereas the CW_VGG19 model can obtain 0.88 AUC. On the whole, the LGB_VGG19 model and XGB_VGG19 model are relatively stable than the CW_VGG19 model, and the AUC value of the LGB_VGG19 model is higher than the XGB_VGG19 model with the increase of imbalanced ratio. As for the running time, the LGB_VGG19 model is much faster than the other models.

5.4 Comparison with other work

In order to verify the generalization ability of the proposed models, they were compared with the state-of-the-art deep learning model [33]. In their work, a model named DNN-TC was modified based on the ResNet-101 model, and it performed well on the Trashnet dataset [36]. In this part, the performances of original VGG-19, fine-tuned VGG-19, and LGB-VGG19 were tested and compared with the DNN-TC model on the Trashnet dataset.

As shown in Table 8, the optimizer, learning rate and momentum used in DNN-TC model were different in the different training periods, while these parameters were set up the same value in the whole training process for the other models. The experiments show that the recognition accuracy of the fine-tuned VGG-19 is 2.53% higher than that of the original VGG-19, which proves the optimized VGG-19 model learned the abstract features of each class well.

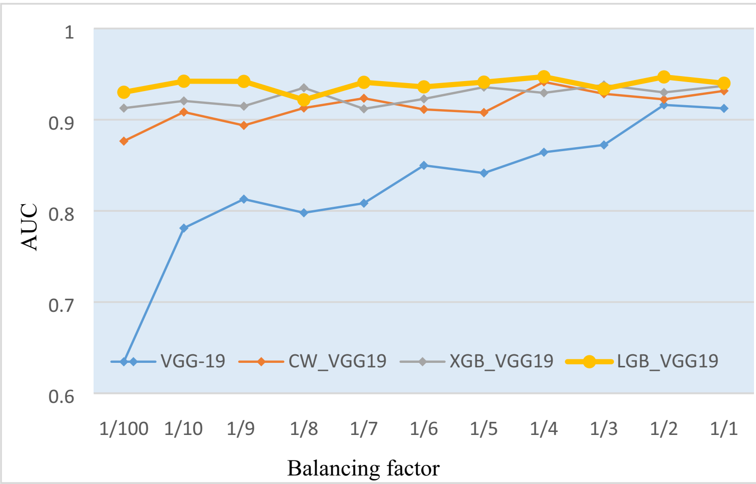

Fig. 8 AUC values of three models on imbalanced dataset, where the number of mungab images is minor compared to the number of drawer images (No.mungab/ No.drawer = 1:100, 1:10, 1:9, 1:8, 1:7, 1:6, 1:5, 1:4, 1.3, 1:2, 1:1)

Table 8 Parameter setting and the accuracy for the experimental methods based on Trashnet dataset

Model	Learning rate	Optimizer	Momentum	Accuracy (%)	Recall	Precision
DNN-TC [33]	0.001 & 0.0001	Adam & SGD	0.9 & 0.999	94	NA	NA
VGG-19	0.001	SGD	0.9	92.64	92.22	93.61
Fine-tuned VGG-19	0.001	SGD	0.9	95.17	94.38	96.50
LGB_VGG19	0.001	SGD	0.9	96.23	95.42	96.97

Moreover, the classification result of LGB-VGG19 achieved the highest accuracy of 96.23% among the four models. The main reason is that multiple weak classifiers of the ensemble learning algorithm enhance the learning ability of the model by increasing the difference between different categories [13]. In addition to accuracy, precision and recall are also used as evaluation metrics to comprehensively reflect the performance of the experimental networks. The results of precision and recall show good robustness of the experimental models used in this study.

In addition, the recognition performances of different CNN architectures (VGG-16, Res-Net, Mobile-Net, Inception-Net, and Dense-Net) were compared with the designed network in other recent work [32] based on the waste classification data v2 (https://www.kaggle.com/sapal6/waste-classification-data-v2). The dataset consists of three classes, including organic waste, non-recyclable waste, and recyclable waste. The proportion of the number of three types of images in the dataset is about 54:13:33, which has the imbalanced data problem. Based on their experiments, the highest accuracy of 92.5% was obtained in the transfer learning process, and the proposed model achieved 81.25% accuracy. As shown in Fig. 9, the best recognition performances for each model with some training parameters are given. Only the designed network [32] cannot be trained to 30 epochs because of large computational complexity. The other CNN architectures are trained up to 30 epochs with a batch size of 50. In their work, the imbalanced data problem was not addressed. Thus, compared with the transfer learning models, the hybrid LGB_VGG19 model obtained a better test result of 93.10%.

6 Conclusion

In this research, several models were used to recognize types of waste under a complex background, and a practical application was developed based on the Android platform. 69,737

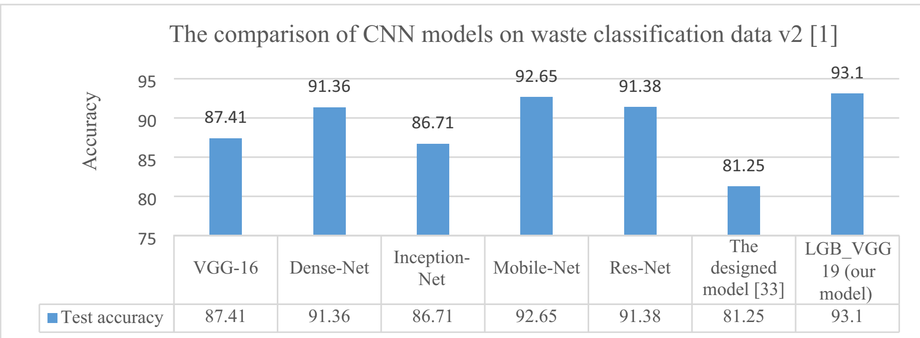

Fig. 9 The test accuracy for different CNN models based on waste classification data v2

images were collected from various websites, and the whole set of images were validated manually. In the model estimation process, computation speed and accuracy were the main factors to assess each model. The fine-tuned VGG-19 model achieved the second-fastest running speed and the highest accuracy. The average accuracy of the final model achieved 86.19% for 95 classes of waste, much higher than previous research in waste classification. Two main problems exist in this research process. One of them is the high similarity between some classes, and the other is the IDP. In order to solve the issue of similar classes, Top-4 accuracy was used in the result, and it gets a high average accuracy of 97.01%. As for the IDP, two methods were applied to solve it, and then they were compared with the original result. Through the experiment, the proposed method can solve the problem very well. In addition, an application was developed to classify waste on android phones based on the fine-tuned VGG-19 model. Users need to take a picture or select an image from the album and then upload it to the server, and finally, the class name and responding price are sent to mobile phones.

Although the system used in this research can achieve better accuracy and faster running speed than other deep CNN models, there are limitations in this research. As mentioned in the experimental result, the final VGG-19 model has a higher demand for computer memory because of the large number of parameters. Moreover, before training the collected dataset, because the collected dataset is large, there is no image pre-processing operation on the image with a complex background, which has a significant effect on the recognition result.

In the future, some important issues will be explored to improve this research. First of all, batch background removal technology should be applied to the application. Except for the target object, there are many other objects in one image sent from the user, and the classification accuracy was influenced by other objects. As a result, it is an effective approach for a good classification accuracy to do complex background removal. However, the dataset used in this research is too large to finish it efficiently. A methodology that combines batch background removal technology will probably improve the classification accuracy of the model. Secondly, the current graphical user interface design of the developed application is not pleasing to the eye, so it should be improved. Besides, the portability of the other IOS platform needs to be taken into account. Thus, the application can benefit more users, and a better experience will be provided to users through the improvement of the interface.

Acknowledgements This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03038540) and by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2019-0-00136, Development of AI-Convergence Technologies for Smart City Industry Productivity Innovation).

References

1. Bircanoğlu, C, et al. (2018). 1RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks. In 2018 Innovations in Intelligent Systems and Applications (INISTA). IEEE
2. Chen, T and C Guestrin (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM
3. Dang LM, Hassan SI, Im S, Moon H (2019) Face image manipulation detection based on a convolutional neural network. Expert Syst Appl 129:156–168
4. Dang LM, Min K, Lee S, Han D, Moon H (2020) Tampered and computer-generated face images identification based on deep learning. Appl Sci 10(2):505

5. Dang LM et al (2020) *Sensor-based and vision-based human activity recognition: A comprehensive survey*. Pattern Recogn 108:107561
6. Davidson Cummings L (1977) Voluntary strategies in the environmental movement: recycling as cooptation. Journal of Voluntary Action Research 6(3–4):153–160
7. Fernández, A, et al. (2018). Cost-sensitive learning, in Learning from Imbalanced Data Sets. Springer. p. 63–78
8. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29:1189–1232
9. Gundupalli SP, Hait S, Thakur A (2018) Classification of metallic and non-metallic fractions of e-waste using thermal imaging-based technique. Process Saf Environ Prot 118:32–39
10. He, K, et al. (2016). Deep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition
11. Huang, GL, He J, Xu Z, Huang G (2020). A combination model based on transfer learning for waste classification. Concurrency and Computation: Practice and Experience: p. e5751
12. Jing, XY, Zhang X, Zhu X, Wu F, You X, Gao Y, Shan S, Yang JY (2019). Multiset feature learning for highly imbalanced data classification. IEEE Trans Pattern Anal Mach Intell, 1
13. Ke, G, et al (2017). Lightgbm*: A* highly efficient gradient boosting decision tree. in Advances in neural information processing systems
14. Li J, Li C, Liao Q, Xu Z (2019) Environmentally-friendly technology for rapid on-line recycling of acrylonitrile-butadiene-styrene, polystyrene and polypropylene using near-infrared spectroscopy. J Clean Prod 213:838–844
15. Liao, H (2016). A deep learning approach to universal skin disease classification. University of Rochester Department of Computer Science, CSC
16. Liu, W, et al. (2016). Large-margin softmax loss for convolutional neural networks. in ICML
17. Lu H, Yang L, Yan K, Xue Y, Gao Z (2017) A cost-sensitive rotation forest algorithm for gene expression data classification. Neurocomputing 228:270–276
18. Masters, D. and C. Luschi (2018). Revisiting small batch training for deep neural networks. arXiv preprint arXiv:1804.07612
19. National/Politics. Available from: http://english.chosun.com/site/data/html_dir/2014/02/14/2014021400795.html
20. Nguyen G, Dlugolinsky S, Bobák M, Tran V, López García Á, Heredia I, Malík P, Hluchý L (2019) Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artif Intell Rev 52(1):77–124
21. Nguyen TN, Lee S, Nguyen-Xuan H, Lee J (2019) A novel analysis-prediction approach for geometrically nonlinear problems using group method of data handling. Comput Methods Appl Mech Eng 354:506–526
22. Nguyen TN, Nguyen-Xuan H, Lee J (2020) A novel data-driven nonlinear solver for solid mechanics using time series forecasting. Finite Elem Anal Des 171:103377
23. Noorani, S and M Fernandes (2017). Evaluation of Convolutional neural networks for Waste Identification. In 2017 International Conference on Computing Methodologies and Communication (ICCMC). IEEE
24. Palmer JA (1995) Environmental thinking in the early years: understanding and misunderstanding of concepts related to waste management. Environ Educ Res 1(1):35–45
25. Ruiz, V, et al. (2019). Automatic Image-Based Waste Classification. In International Work-Conference on the Interplay Between Natural and Artificial Computation. Springer
26. Seoul Statistics. Available from: http://stat.seoul.go.kr/Seoul_System5.jsp?stc_cd=412
27. Simonyan, K. and A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
28. Sousa, J, A Rebelo, and JS Cardoso (2019). Automation of Waste Sorting with Deep Learning. In 2019 XV Workshop de Visão Computacional (WVC). IEEE
29. Sun Z, Song Q, Zhu X, Sun H, Xu B, Zhou Y (2015) A novel ensemble method for classifying imbalanced data. Pattern Recogn 48(5):1623–1637
30. Szegedy, C, et al. (2015). Going deeper with convolutions. in Proceedings of the IEEE conference on computer vision and pattern recognition
31. Szegedy, C, et al. (2016). Rethinking the inception architecture for computer vision. in Proceedings of the IEEE conference on computer vision and pattern recognition
32. Thanawala, D, A Sarin, and P Verma (2020). An Approach to Waste Segregation and Management Using Convolutional Neural Networks. In International Conference on Advances in Computing and Data Sciences. Springer
33. Vo AH, Vo MT, Le T (2019) A novel framework for trash classification using deep transfer learning. IEEE Access 7:178631–178639
34. Williams, PT (2005). Waste treatment and disposal.: Wiley

35. Xue, M and C Zhu (2009). The socket programming and software design for communication based on client/server. In 2009 Pacific-Asia Conference on Circuits, Communications and Systems. IEEE
36. Yang, M and G Thung (2016). Classification of trash for recyclability status. CS229 Project Report, 2016
37. Yu L, Zhou R, Tang L, Chen R (2018) A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data. Appl Soft Comput 69:192–202