



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/174626/>

Version: Accepted Version

---

**Article:**

Li, J., Chen, J., Sheng, B. et al. (2022) Automatic detection and classification system of domestic waste via multi-model cascaded convolutional neural network. IEEE Transactions on Industrial Informatics, 18 (1). pp. 163-173. ISSN: 1551-3203

<https://doi.org/10.1109/TII.2021.3085669>

---

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Reproduced in accordance with the publisher's self-archiving policy.

**Reuse**

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.

# Automatic Detection and Classification System of Domestic Waste via Multi-model Cascaded Convolutional Neural Network

**Abstract**—Domestic waste classification was incorporated into legal provisions recently in China. However, relying on manpower to detect and classify domestic waste is highly inefficient. To that end, we propose a Multi-model Cascaded Convolutional Neural Network (MCCNN) for domestic waste image detection and classification. MCCNN combined three subnetworks (DSSD, YOLOv4, and Faster-RCNN) to obtain the detections. Moreover, to suppress the false-positive predicts, we utilized a classification model cascaded with the detection part to judge whether the detection results are correct. To train and evaluate MCCNN, we designed a large-scale waste image dataset (LSWID), containing 30,000 domestic waste multi-labeled images with 52 categories. To the best of our knowledge, the LSWID is the largest dataset on domestic waste images. Furthermore, a smart trash can (STC) is designed and applied to a Shanghai community, which helped to make waste recycling more efficient. Experimental results showed a state-of-the-art performance, with an average improvement of 10% in detection precision.

**Index Terms**—Domestic waste detection and classification, multi-model cascaded convolutional neural network, detection precision, smart trash can.

## I. INTRODUCTION

THE World Bank report shows nearly 4 billion tons of waste is produced in the world every year. It is estimated that by 2025, waste will increase by 70% [1]. In 2018, large and medium-sized cities generated 1.55 billion tons of general industrial solid waste, 46.43 million tons of industrial hazardous waste, and 817 thousand tons of medical waste in China. Apart from those, the amount of urban domestic waste generated was 211.47 million tons. Among the components of urban domestic waste, kitchen waste accounts for more than 50% [2]. According to the survey, the main treatment methods of kitchen waste are organic compost and landfill. Furthermore, once the kitchen waste that needs to be treated is mixed with a large amount of other waste, such as plastic bags, the environment will be heavily polluted. Therefore, domestic waste classification is critical, to ensure that a certain type of waste contains as little as possible other waste

The main obstacles to waste classification include (1) Government planning and budget: insufficient government regulations and budgets for waste management; (2) Family education: families do not understand the importance of self-classification of waste; (3) Technology: lack of adequate waste classification technology; and (4) Management cost: the high cost of manual waste classification[3].

With the development of waste classification technology and the government's strong support, many scholars have studied several smart devices and applications with specific sensors[4],

[5]. However, the domestic waste thrown by residents is complicated and challenging to identify with only specific sensors. Therefore, some methods based on machine vision are proposed to simplify the detection process [6], [7], [8], [9], [10], [11], [12], [13]. These methods generally utilize deep learning theory to classify waste through convolutional neural networks (CNN), of which Faster-RCNN[14], DSSD[15], YOLOv3[16], and YOLOv4[17] with different advantages in accuracy, speed, and sizes are the most common models. Unlike general object recognition, waste owns different shapes, sizes, and sometimes overlaps. So relying on a single model (Faster-RCNN, DSSD, or YOLOv3/v4) with limited feature extraction capabilities to remove false-positive predictions is not adequate to solve these related obstacles. So, we propose an automatic detection and classification system of domestic waste based on deep learning with high precision and practicality and combining the advantages of different algorithms. Our method uses three subnetworks (DSSD, YOLOv4, and Faster-RCNN) to obtain the detection results of waste images. Moreover, to suppress the false-positive predicts, we utilized a classification model cascaded with the detection part to judge whether the detection results are correct. And our goal is described in Fig. 1. The key contributions are listed as follows.

- We propose a domestic waste detection and classification system based on a convolutional neural network model, which has the advantage of covering more detection types (52 categories) and achieving higher precision (more than 90%) compared with other waste classification systems[6], [7], [8], [9], [10], [11], [12], [13], and it is the first work that can effectively distinguish between kitchen waste and non-kitchen waste.
- To decrease the false-positive predictions caused by waste shape, size, and even overlap, we propose a deep convolutional neural network based on the Multi-model Cascaded method named MCCNN for domestic waste image detection and classification. MCCNN fuses the advantages of various detection models with different sensitivities to target shape, size, and even overlap while improving the precision of detection by cascading a classification model behind the detection model.
- By verifying a large amount of data (LSWID), this method (MCCNN) achieves an average increase of 10% in detection precision. Moreover, the smart trash can (STC) with MCCNN model has been applied to communities and assists residents in waste classification, which achieved beneficial performance.

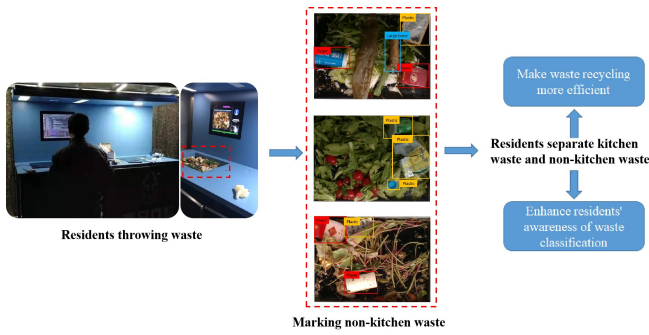


Fig. 1. Goal of our study. Starting from the distinction between kitchen waste and non-kitchen waste, we integrated automation and machine vision technologies to make waste classification more convenient. Not only can the residents' awareness of waste classification be improved, but also the recycling of waste can be made more efficient.

## II. RELATED WORK

Research on waste classification spans multiple fields, from traditional industries to the automation industry to deep learning. In this article, we focus on the detection and classification of domestic waste based on image detection methods.

In 2016, JayDonovan[8] created "AutoTrash" which is an automatically sorting trash can that can use Raspberry Pi-driven modules and cameras to identify compost and recycling. It should be pointed out that this project can only distinguish whether the target is recycling or compost, and the function is relatively simple. In the same year, TrashNet[7] was proposed by Yang and Thung, and they created a dataset containing around 2500 images and six classes, which was hand-collected. Their models used were support vector machines (SVM) with scale-invariant feature transform (SIFT) features and a convolutional neural network (CNN), and the two models achieved an accuracy of 63% and 87%, respectively. TrashNet became a public benchmark for waste classification; however, this dataset has not been public so far. Besides TrashNet, a few waste datasets, such as TACO[10], AquaTrash[9], and VN-trash, were established, and they have some shortcomings, such as the relatively small amount of waste in a specific environment. Also, the dataset was not open source. Adedeji, Olugboja, and Wang, Zenghui[11] continued Yand and Thung's work. To simplify the process, they proposed an intelligent waste material classification system, which is developed by using the 50-layer residual net pre-train (ResNet-50) Convolutional Neural Network model and Support Vector Machine (SVM), which is used to classify the waste into different groups/types such as glass, metal, paper, and plastic, etc. The proposed system is tested on the trash image dataset, which was developed by Gary Thung and Mindy Yang[7], and is able to achieve an accuracy of 87% on the dataset.

Some of the tasks mentioned above are designed from the perspective of intelligent hardware and have been applied to actual engineering. However, the structure of these algorithms is based on the existing models, and there is no design change for the characteristics of waste images. In the waste image, the size and shape of waste are different, and there is overlap. Therefore, it is challenging to solve all the problems with a single model. We are thinking about whether we

can obtain the advantages of multiple models by cascading while improving the recognition frame's accuracy. In 2017, Zhaowei Cai and Nuno Vasconcelos[18] proposed Cascade R-CNN, which consists of a sequence of detectors trained with increasing IoU thresholds, to be sequentially more selective against close false positives. Inspired by the multi-stage object detection framework, we proposed a deep convolutional neural network based on the Multi-model Cascaded method named MCCNN.

In this task, CNN plays an important role, but deep neural networks are like a black box. Although they can provide excellent performance, they lack decomposability and cannot be intuitively understood, making it difficult to explain. To make the network more visible, Bolei Zhou et al [19]. described the procedure for generating class activation maps (CAM) using global average pooling (GAP) in CNNs. Through CAM, we can clearly understand which part of the image has a more significant impact on the result. This technique is beneficial but has some flaws. First of all, we must change the network structure, such as changing the fully connected layer to a global average pooling layer, which is not conducive to training. The second is that this is a visualization technique based on classification problems, which may not have such a good effect on regression problems. To solve the first problem, an improved technology called Grad-CAM [20] appeared in 2017. Grad-CAM can visualize without changing the network structure and is applied in multiple scenarios. To get better results (especially when there is more than one object in a specific category in the image), Chattopadhyay et al [21]. further proposed Grad-CAM++. The main change is in the weight of the feature map corresponding to a specific category. ReLU and weight gradient are added to the representation, and the gradient can be calculated with only one backpropagation.

## III. METHODOLOGY

### A. Datasets and Augmentation

This article uses deep learning to identify the waste thrown by residents, but there is currently no unified dataset for domestic waste. The datasets about waste images that we can find are as follows: AutoTrash, TrashNet, TACO and AquaTrash. Among them, AutoTrash has 50 categories, and each category has 100 photos, which is only using for image classification. TrashNet was proposed by Yang and Thung, who created a dataset containing around 2527 images and six classes, which was hand-collected. TACO includes 28 categories, 1500 pictures, and 4784 annotations, which is applied to waste image segmentation. AquaTrash consists of 369 images from 4 different categories related to various litter items, including glass, metal, paper and plastic. However, our large-scale waste image dataset (LSWID) contains 30,000 domestic waste multi-labeled images with 52 categories, which surpasses other datasets far in scale and quality. And our dataset is collected in the scene of actual waste disposal by residents, which has very important practical significance. We collected 30,000 waste images and marked each image to form a large-scale waste image dataset (LSWID). And in our dataset the images cover one annotation target to multiple annotation targets. In our

dataset, each image corresponds to a file in txt. The format of each line in tx files is “classid, xcenter, ycenter, width, height”, which corresponds to the categories and locations data of the bounding boxes.

It is observed that the non-kitchen waste contained in the kitchen waste mainly includes paper, plastic, shell, large bone, and cigarette(more than 80%), so we will train and test these five types of waste during the experiment.

In order to improve the generalization performance of the actual project, we consider preprocessing the training for data augmentation[22], subjecting the training source image to random brightness conversion, stretch conversion, and mirror conversion before inputting it into the networks.

## B. Architecture

In the entire waste classification system, target detection plays a critical role. The system uses target detection to determine whether the domestic waste released by residents contains other types of waste to ensure the purity of various waste types to reduce the pressure of later steps. Object detection is a fundamental visual detection problem in computer vision and has been widely studied in the past decades. Object detection techniques using deep learning have been actively studied in recent years. All the methods here can be divided into two categories: two-stage detectors and one-stage detectors. Two-stage detectors split the detection task into two stages: (i) proposal generation; and (ii) making predictions for these proposals, like RCNN[23], SPP-Net[24], Fast RCNN[25], Faster-RCNN[14] and R-FCN[26], while one-stage detectors do not have a separate stage for proposal generation, like OverFeat[27], YOLO[28], SSD[29], YOLOv2[30], RetinaNet[31], DSSD[15], YOLOv3[16], and YOLOv4[17]. In the past ten years, the algorithms for image classification mainly include VGG16[32], Xception[33], MobileNet[34], ResNet50 and ResNet101[35].

There are two significant challenges, first includes different shapes, sizes, and sometimes overlaps of waste. So, relying on a single model (Faster-RCNN, DSSD, or YOLOv3/v4) with limited feature extraction capabilities to solve these related obstacles is not adequate. Secondly, in waste image detection task, we noticed that compared to the target recall rate, the precision of target detection is more crucial, which means that we should remove as many false-positive predictions as possible. To overcome the above-mentioned challenges, we proposed the MCCNN method, which uses three subnetworks (DSSD, YOLOv4, and Faster-RCNN) to obtain the detection results of waste images. Moreover, to suppress the false-positive predicts, we exploited a classification model cascaded with the detection part to determine whether the detection results are accurate.

The combination of these three sub-networks has their own advantages which helped to solve mentioned challenges. The feature extraction layers of YOLOv4 use a feature pyramid down-sampling structure and a Mosaic data enhancement method during training, so it has a good effect on small target detection. Faster-RCNN is a two-stage detection algorithm. The first stage is to generate candidate regions, and the second

stage is to adjust and classify the position of the candidate regions. And the recognition error rate is low. DSSD adds contextual information in the model. It can more capture the information of deep and shallow feature maps, which is conducive to solving the overlap problem.

The target detection model’s final prediction results are the position and category of the target in the image, and the image classification is to predict the category of a single target. In the actual application process, we know that the criteria for measuring the target detection model are recall rate and precision[36]. The methods for calculating the recall rate and precision are shown in Eq. (1) and Eq. (2), respectively.

$$recall = \frac{tp}{tp + fn} \quad (1)$$

$$precision = \frac{tp}{tp + fp} \quad (2)$$

where  $tp$ : the number of correct detection frames.  $fp$ : the number of frames that detect background as targets.  $fn$ : the number of frames that detect targets as background. In layman’s terms, the recall rate indicates the degree of missed detection for the target. The higher recall rate performs the less missed detection, and the precision presents the correct rate of the detected bounding boxes. In the scenario of waste detection and classification, the precision should be as high as possible, allowing a certain degree of missed detection.

To balance recall rate and precision, we need to change the ratio of difficult cases in RPN[14], or the parameters of the anchors, and also need to adjust the threshold of the NMS[37], which may lose the number of output target frames of the original model. So we thought that we could cascade a classification model behind the detection model. At the same time, the training dataset of the classification model consists of the result of cutting out the multi-label dataset of the detection model. This means that the classification model can perform more data augmentation and has more features in a single classification task. Reliability, the secondary verification can greatly improve the accuracy of target frame detection.

According to Eq. (2), reducing the  $fp$  is necessary to improve precision. Based on the detection model mentioned above, the position and category information of each bounding box in the image can be obtained. If the detection result could be judged again, the precision of detection can be increased. The final output model can significantly improve the detection precision based on ensuring the recall rate as much as possible. Therefore, we proposed an algorithm via Multi-model Cascaded Convolutional Neural Network for asynchronously checking the detection model results and the classification model, as shown in Fig. 2. Due to the different detection accuracy, detection speed, and sensitivity to large and small objects, we select three detection networks (DSSD, YOLOv4, and Faster-RCNN) with distinctive characteristics as sub-networks of the detection model. Moreover, ResNet101 is to be used as a classification network to verify the detection results.

The calculation process related to MCCNN is shown in Algorithm 1. In Algorithm 1,  $I$  represents the input picture,  $Rec\_model_i$  represents the  $i^{th}$  detection network,  $Cla\_model$

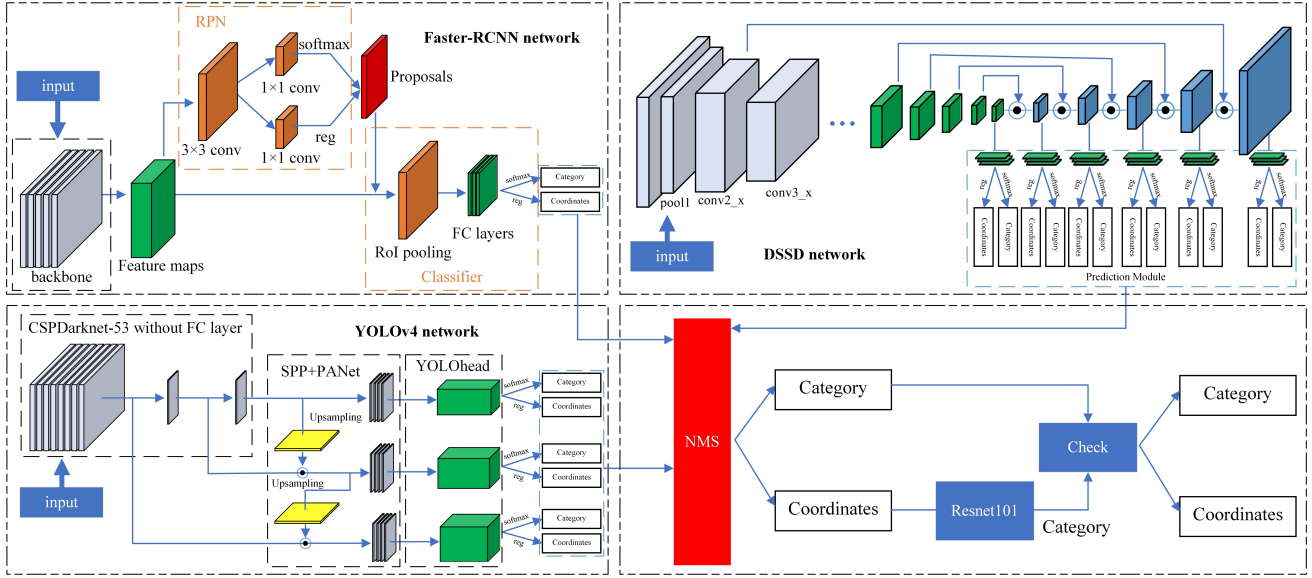


Fig. 2. The architecture of MCCNN. It consists of two parts, the detection model and the classification model. The detection model includes three sub-networks (DSSD, YOLOv4, and Faster-RCNN), and the results of each sub-network are combined and then passed through the NMS algorithm, finally obtaining the results of the detection model, including categories and bounding boxes. We crop the waste image into patches using the bounding boxes. The classification model (ResNet101) takes the predicted image patch as input and predicts the category of the patch. We keep the bounding box if the predicted category and input category are consistent.

represents the classification model,  $BBoxes \langle i, j \rangle$  represents the  $j^{th}$  detection frame of the  $i^{th}$  detection network. There are six parameters  $[x1, y1, x2, y2, Conf, Class]$ , which represent the upper left corner coordinates, lower right corner coordinates, confidence level, and category information of the detection frame in  $I$ .  $Res\_Cla \langle i, j \rangle$  represents the output category after the  $j^{th}$  detection frame of the  $i^{th}$  detection network that passes the classification model. Firstly, resize the waste image to a fixed size, and send it to the detection model. The three sub-networks of the detection model separately forward the waste image, and the results of each sub-network are combined and then passed through the non-maximum suppression algorithm, finally obtaining the results of the detection model, which includes the position of the bounding boxes and the category information of the corresponding bounding boxes. Then, images cropped according to the bounding box change to a fixed size, and next, they are loaded into the classification model, which can output the category value of each of them, and finally verify the output category result of the detection model. If they are consistent, then retain the corresponding bounding box; if not, delete the corresponding bounding box.

### C. Loss Function

According to the architecture of MCCNN, it can be seen that its loss is composed of four parts, three losses of detection sub-networks and the loss of a classification model. There are two options for the loss function. The simplest method is to splice these networks and train them together with weighted loss functions. As shown in Eq. (3),  $L_{Faster-RCNN}$ : the loss of Faster-RCNN sub-network.  $L_{DSSD}$ : the loss of DSSD sub-network.  $L_{YOLOv4}$ : the loss of YOLOv4 sub-network.  $L_{ResNet101}$ : the loss of the classification model.  $\alpha, \beta, \lambda, \mu$  are

the weight coefficients of these four losses, which ensure that each loss value is within an order of magnitude.

$$Loss = \alpha L_{Faster-RCNN} + \beta L_{DSSD} + \lambda L_{YOLOv4} + \mu L_{ResNet101} \quad (3)$$

However, in MCCNN, the three detection sub-networks' loss functions are well constructed, mainly including regression loss and classification loss, but it is complicated to define the loss function of the MCCNN classification model.

For the following reasons: 1. The detection model and classification model in our task have different dataset structures. In this task, the classification model's final output needs to be selected according to the results of the detection model, so its loss function cannot be defined directly. 2. After the four networks are spliced, the memory occupied during training is enormous, and the convergence speed is plodding. 3. Besides, for the detection sub-network, the batch is easy to define, but according to the architecture of MCCNN, the input of the classification model needs to intercept the target area of each picture in the current batch, so the batch of the classification model cannot be determined. 4. The architecture we proposed focuses on each network's optimal results, rather than balancing the weight ratio of each network, and separate training increases the data enhancement, making the model more generalized. Finally, we decided to train these four networks separately to achieve the best results and then spliced all the networks for detection and classification.

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases} \quad (4)$$

For the loss functions of Faster-RCNN and DSSD, we did not make more changes. We made some improvements to YOLOv4. This article adjusts the loss function, mainly to change the  $L_2$  loss function in the positioning to  $Smooth L_1$

---

**Algorithm 1** Waste Image Detection and Classification via Multi-model Cascaded Convolutional Neural Network
 

---

**Input:** input single image:  $I \in R^{C \times W \times H}$ , input detection models:  $Rec\_model \in \{DSSD, YOLOv4, Faster - RCNN\}$ , input classification model:  $Cls\_model \in \{ResNet101\}$ , and  $Res\_Cla \langle i, j \rangle$ ,  $Class \in \{paper, plastic, shell, bargebone, cigarette\}$ , input result list:  $Results = []$

**Output:** Return new detection  $Results$

```

1: Initialize the system and configuration;
2: Input image  $I$ ;
3: According to the longest side, pad  $I$  and then resize  $I$  to
   a fixed size ( $512 \times 512$ );
4: for each  $i$  in  $Rec\_model$  do
5:    $BBoxes = Rec\_model_i(I)$ ;
6:   Add  $BBoxes$  to  $Results$ ;
7: end for
8:  $Results = NMS(Results)$ ;
9: for each  $i$  in  $Results$  do
10:  for each  $j$  in  $i$  do
11:     $Res\_Cla \langle i, j \rangle = Cls\_model(I[Results \langle i, j \rangle [0]$ 
12:    to  $Results \langle i, j \rangle [2], Results \langle i, j \rangle [1]$ 
13:    to  $Results \langle i, j \rangle [3]])$ ;
14:    if  $Res\_Cla \langle i, j \rangle$  is the same as  $Results \langle i, j \rangle [-1]$ 
        then
15:      Keep  $Results \langle i, j \rangle$ ;
16:    end if
17:  end for
18: end for

```

---

loss function, such as Eq. (4). *Smooth L1* loss is less sensitive to outliers than *L2* loss and is more robust because when  $|x|$  is greater than 1, the form of *L1* loss (linear loss) is used to avoid the problem of gradient explosion. It can be seen from the form of the derivative of the loss function in backpropagation that when  $|x|$  is greater than 1, the derivative of *L2* loss increases linearly, while the derivative of *Smooth L1* loss is constant. It can be said that *Smooth L1* loss combines the excellent points of *L1* loss and *L2* loss. When the error between the prediction box and the actual box is too large, the gradient value is not too large, and when the error between the prediction box and the actual box is small, the gradient value is also small enough. This makes positioning regression converge fast. It is also more stable and helps to increase the training speed of the network. So the definition of its loss function is shown in Eq. (5).

Where,  $I_{ij}^{obj}$ : the  $j^{th}$  prediction box of the  $i^{th}$  grid unit, which is responsible for predicting the non-kitchen waste.  $I_{ij}^{noobj}$ : the  $j^{th}$  prediction box of the  $i^{th}$  grid unit, which is not responsible for predicting the non-kitchen waste.  $\hat{C}_i^j$ : the same as  $I_{ij}^{obj}$ .  $C_i^j$ : the probability that the algorithm estimates there is non-kitchen waste in the prediction frame.  $\hat{x}_i^j$ : the lateral offset of the GT (Ground Truth) center relative to the upper-left corner of the grid unit.  $x_i^j$ : the lateral offset of the prediction box center relative to the upper-left corner of the grid unit.  $\hat{y}_i^j$ : the longitudinal offset of the GT center relative

to the upper-left corner of the grid unit.  $y_i^j$ : the longitudinal offset of the prediction box center relative to the grid unit's upper-left corner.  $\hat{w}_i^j$ : the width ratio of GT to the image.  $w_i^j$ : the width ratio of the prediction box to the image.  $\hat{h}_i^j$ : the height ratio of GT to the image.  $h_i^j$ : the height ratio of the prediction box to the image.  $\hat{P}_i^j$ : the probability that GT belongs to category  $c$ , which is 0 or 1.  $P_i^j$ : the probability that the prediction box belongs to category  $c$ .

It should be pointed out that because the position error is not equally important as the classification error [38], it is necessary to add a weight  $\lambda_{coord}$  to the position error, the value is 5. Moreover, the number of backgrounds is much larger than the number of detected objects, so the weight  $\lambda_{noobj}$  is added, and the value is 0.5. In order to make it easier to detect the small volumes of non-kitchen waste, adding  $(2 - \hat{w}_i^j * \hat{h}_i^j)$  can make the weight of small volumes of non-kitchen waste in the loss function larger. This is a value that can be fine-tuned according to the actual dataset characteristics. If a small volume of non-kitchen waste in a data set accounts for a large proportion, it can be appropriately increased  $\lambda_{noobj}$  is added, and the value is 0.5. In order to make easier detect the small volumes of non-kitchen waste, adding  $(2 - \hat{w}_i^j * \hat{h}_i^j)$ , and vice versa.

$$\begin{aligned}
Loss = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} (2 - \hat{w}_i^j \hat{h}_i^j) [smooth_{L1}(x_i^j - \hat{x}_i^j) \\
& + smooth_{L1}(y_i^j - \hat{y}_i^j)] \\
& + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} (2 - \hat{w}_i^j \hat{h}_i^j) [smooth_{L1}(w_i^j - \hat{w}_i^j) \\
& + smooth_{L1}(h_i^j - \hat{h}_i^j)] \\
& - \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] \\
& - \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{noobj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] \\
& - \sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{c \in classes} [\hat{P}_i^j \log(P_i^j) + (1 - \hat{P}_i^j) \log(1 - P_i^j)]
\end{aligned} \tag{5}$$

The other two parameters for adjusting the specific gravity can also be adjusted according to the actual situation. The ultimate goal is to make the model's actual detection more effective in identifying non-kitchen waste and the loss function definition of the Faster-RCNN and classification model, referring to [14], [35] will not be repeated here.

#### D. STC System Based on MCCNN

We have designed the entire waste disposal process, shown in Fig. 3. First of all, in urban communities, public places, there are areas dedicated to waste disposal and this is where most domestic waste is concentrated. As the most an essential part of the entire system, STC system integrates various types of sensors and components that identify the types, weight, and impurity ratios of waste discarded by residents. Then, it

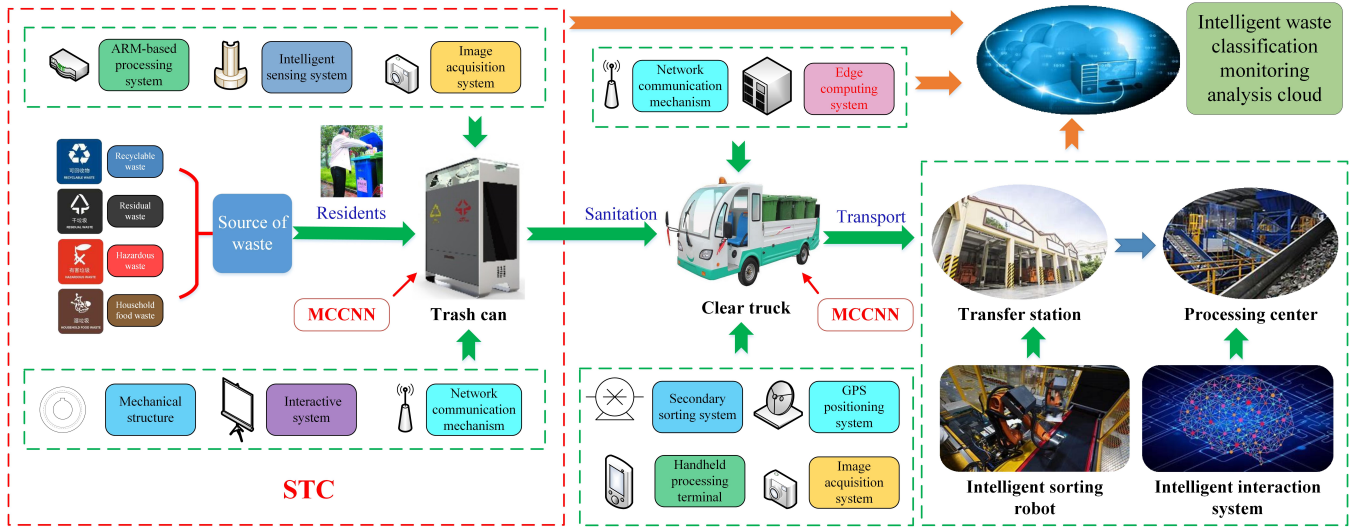


Fig. 3. Pipeline of our proposed waste disposal process. The essential steps, such as clearing trucks, transfer stations, and processing centers, are integrated with artificial intelligence-related technologies and equipment based on the existing decentralized processes. The full link from generation to disposal of domestic waste can be observed and traced.

is the waste transferring process, when the collection volume of the waste reaches a certain level, the waste truck with intelligent equipment collects the waste and transports it to large waste transfer stations. In the end, the domestic waste will be transported to large transfer centers and processing centers, and sorted again to ensure that the purity of each type of waste is maximized before final processing.

In this article, we mainly introduce one of the subsystems, the STC system. To verify the MCCNN model in waste detection and classification, we deployed an STC system with MCCNN in a Shanghai community. The STC system serves as a front-end part of the entire waste disposal process. After the identity recognition is successful, residents can put waste on the tray of STC. The system takes an image of waste through the camera and uses the MCCNN model to recognize the image. The recognition results will be displayed on the screen, and the category of non-kitchen waste will be marked. The residents need to put the marked waste into the corresponding tray. After that, the system automatically opens the trays, and the waste falls into the trash can.

In the whole process, the system records the weight information of the waste, the residents' putting information, identification records, the effect of waste classification. All the statistical data will be transmitted to the community management department by the network.

## IV. EXPERIMENTS AND DISCUSSIONS

### A. Model Training

MCCNN is composed of a detection model and a classification model. The simplest training method is to splice the two networks and train them together by weighting the loss function. However, the training of such a huge compound model is extremely cumbersome, and because the detection results of inconsistent prediction categories are eventually discarded, the definition of the loss function is also not easy. Therefore, we use the strategy that two models are trained

separately, and the trained networks are stitched together for deployment. We selected the DSSD, YOLOv4, and Faster-RCNN networks for the sub-network of the detection model and trained them separately. For the classification model, we chose the ResNet101 network with better performance.

1) *Detection Model:* We chose three sub-networks as the detection model of MCCNN, YOLOv4, and DSSD are one-stage detection networks, Faster-RCNN is a two-stage detection network. Simultaneously, these three networks belong to the anchor-based algorithm, the calculation method of anchor boxes, as described in[14].

We selected 5804 waste images in the dataset, which included five categories: paper, plastic, shell, barge bone, and cigarette. The dataset is divided into the training set and the testing set according to the ratio of 9:1. For the classification model, the corresponding part of the bounding box is cropped through the label file and saved into the corresponding category folder, which constitutes the dataset for classification, and is also divided into a training set and testing set according to 9:1.

All of our detection sub-networks are based on the anchor mechanism, which is the most important RPN structure. However, the strategy for generating anchors for each sub-network is different. In YOLOv4, we clustered the bounding boxes of labels using K-means in the dataset to generate nine anchors of different sizes as a priori boxes. The anchors in our dataset are listed as (43,36), (48,54), (71,71), (100,85), (121,122), (142,174), (182,119), (206,204), (342,387). In Faster-RCNN and DSSD, we utilize different scales and aspect ratios to generate anchors. The scale is  $\{32^2, 64^2, 128^2, 320^2\}$ , the aspect ratio is  $\{1 : 1, 1 : 2, 2 : 1, 1 : 3\}$ . They group each other to produce 16 anchors.

We used pre-trained models for each sub-network. For the Faster-RCNN and DSSD networks, we used a pre-trained model on Pascal VOC dataset with ResNet-101 backbone. YOLOv4 is pre-trained on MC COCO dataset using CSP-

Darknet53 backbone. We choose the Resnet101 as classification model pre-trained on ImageNet dataset. The hardware environment includes 64G memory, 20 cores, and a 2080Ti GPU. Before the training of the detection model, the relevant configuration is shown in Table I. After starting the training, the system will save the loss value of a model once each step. And every 2000 steps, we save the model weight value once. Simultaneously we test the model on the testing set and calculate the AP [36] value of various categories and final mAP value [36]. The loss change of each detection model

TABLE I  
RELATED CONFIGURATION OF EACH DETECTION NETWORK

|                  | DSSD[15]      | YOLOv4[17]    | Faster-RCNN[14] |
|------------------|---------------|---------------|-----------------|
| Backbone Network | ResNet101     | CSPDarknet53  | ResNet101       |
| Input Size       | 512x512       | 608x608       | 600x800         |
| Has RPN          | False         | False         | True(align)     |
| Optimizer        | SGD           | SGD           | SGD             |
| Learning Rate    | 0.001         | 0.001         | 0.001           |
| Max Step         | 40000         | 40000         | 40000           |
| Gamma            | 0.1           | 0.1           | 0.1             |
| Batch Size       | 8             | 8             | 8               |
| Lr Steps         | [15000,30000] | [15000,30000] | [15000,30000]   |
| Decay            | 0.0005        | 0.0005        | 0.0001          |
| Momentum         | 0.9           | 0.9           | 0.9             |
| Num Workers      | 10            | 10            | 10              |

during the training process is shown in Fig. 4 below. The three sub-figures show the convergence process of the three networks. Due to each network's different loss functions and structure, the difference of the loss is obvious, but the three networks have gradually converged after 20,000 steps. When training to 40,000 steps, the DSSD network's loss value is between 1 and 2, the YOLOv4 network's loss value drops faster, it is close to 0 finally, and the Faster-RCNN network's loss value declines fastest, and the final loss value is also close to 0.

Fig. 5 shows the precision, recall rate, and mAP changes in the testing set during training. As the training deepens, they all gradually increase. After about 20,000 steps, it tends to stabilize. Faster-RCNN has the highest mAP, above 60%, and the mAPs of other networks are around 50%. For comparison, we chose the weight at step 40,000 for testing; each category's AP and mAP values are shown in Table II. Table III and Table IV show the recall rate and the precision of various networks on the testing set, respectively. After the training, the size of the three detection networks is 931MB, 235MB, and 360MB.

2) *Classification Model*: In MCCNN, the classification model is a highly important part. As mentioned above, although the detection model can obtain the location and category information of the detection target, due to the complex characteristics of the waste, the wrong target for category detection often appears (non-kitchen waste is detected as kitchen waste). In the actual process, we pay more attention to the model's precision than the recall rate. Therefore, the system should try to avoid false detection. In order to au-

TABLE II  
THE AP AND MAP OF VARIOUS METHODS ON THE TESTING SET(%)

| Network         | AP on the waste dataset |         |       |            |           | mAP  |
|-----------------|-------------------------|---------|-------|------------|-----------|------|
|                 | paper                   | plastic | shell | large bone | cigarette |      |
| DSSD[15]        | 64.2                    | 71.6    | 40    | 53.9       | 20        | 49.9 |
| YOLOv4[17]      | 56.7                    | 68.3    | 15.2  | 58.1       | 41.6      | 48   |
| Faster-RCNN[14] | 75.4                    | 77.6    | 50.1  | 84.6       | 36.3      | 64.9 |

TABLE III  
THE RECALL RATE OF VARIOUS NETWORKS ON THE TESTING SET(%)

| Network         | Recall Rate on the waste dataset |         |       |            |           |
|-----------------|----------------------------------|---------|-------|------------|-----------|
|                 | paper                            | plastic | shell | large bone | cigarette |
| DSSD[15]        | 60                               | 58.5    | 40    | 61.9       | 20        |
| YOLOv4[17]      | 64.4                             | 71.9    | 24    | 61.5       | 50        |
| Faster-RCNN[14] | 81.6                             | 84.6    | 58    | 92.8       | 40        |

TABLE IV  
THE PRECISION OF VARIOUS NETWORKS ON THE TESTING SET(%)

| Network         | Precision on the waste dataset |         |       |            |           |
|-----------------|--------------------------------|---------|-------|------------|-----------|
|                 | paper                          | plastic | shell | large bone | cigarette |
| DSSD[15]        | 81.6                           | 58.5    | 95.2  | 70.2       | 100       |
| YOLOv4[17]      | 67.8                           | 79.3    | 44.4  | 92.3       | 50        |
| Faster-RCNN[14] | 56.9                           | 57.2    | 34.1  | 59.1       | 22.2      |

tomatically discard the wrong target of category detection, another classification network is needed. We use a mature ResNet101 network, and the flow can be described as the following: identifying the location and type of non-kitchen waste by detection model, cutting out the detected non-kitchen waste from the picture and inputting it into the ResNet101 classification network; the output of the detection model is compared with the category output by ResNet101. If the category is different, the detection box is discarded. When the training reaches 40,000 steps, the model has converged, and the accuracy on the testing set has reached 85%. After training, the size of the classification network is 162MB.

## B. System Evaluation

After the detection model and classification model training is completed, the feedforward detection model is constructed according to the algorithm of MCCNN. DSSD, YOLOv4 and Faster-RCNN are used as the detection sub-networks of MCCNN. While ResNet101, is used as the classification network. In order to design comparative experiments, we separately tested the MCCNN model composed of a detection subnetwork (DSSD, YOLOv4 or Faster-RCNN) and a classification model (ResNet101), and MCCNN with three detection sub-networks were also tested. For all detection sub-networks, the IOU threshold for non-maximum suppression is 0.5, and the object confidence threshold uses 0.5, and the IOU threshold

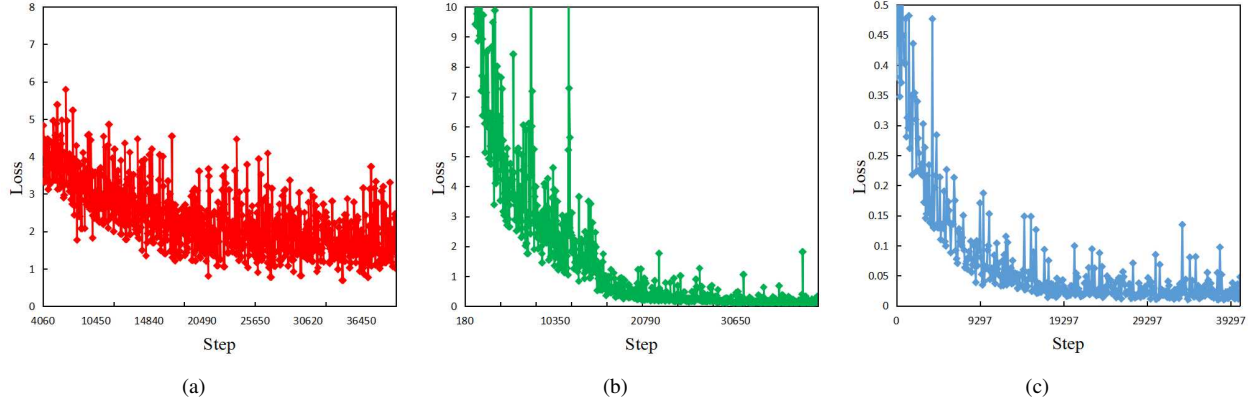


Fig. 4. The loss of each detection network. (a) DSSD loss(Detection network). (b) YOLOv4 loss(Detection network). (c) Faster-RCNN loss(Detection network).

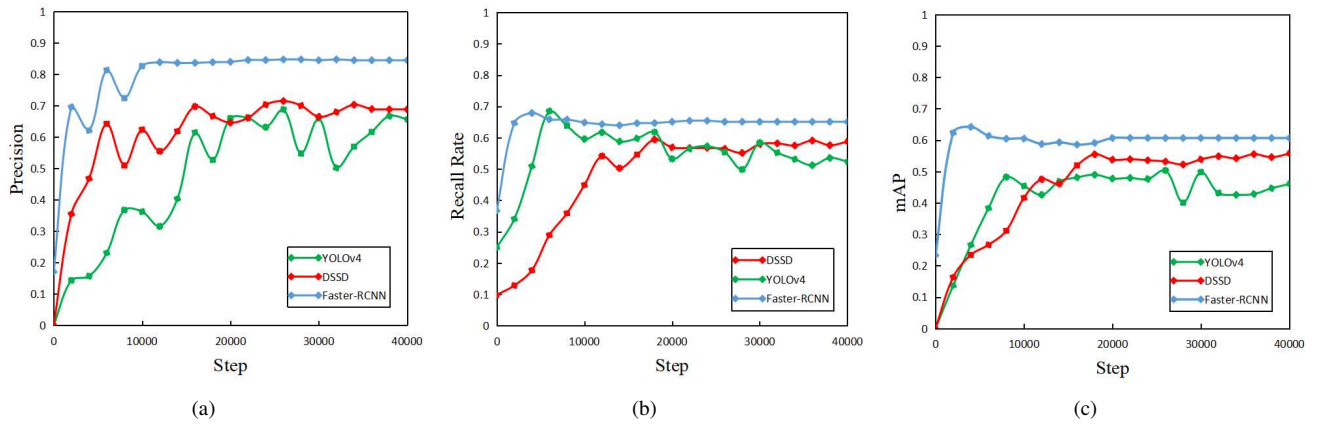


Fig. 5. (a) Precision in various networks of training. (b) Recall Rate in various networks of training. (c) mAP in various networks of training.

required to qualify as detected defines 0.5. Based on the above settings, the testing results are shown in Table V, Table VI, and Table VII. In these tables, MCCNN-DSSD means that MCCNN’s detection model only includes a detection sub-network (DSSD), and MCCNN’s classification model is ResNet101. MCCNN-YOLOv4 and MCCNN-Faster-RCNN have similar explanations. “Lift” indicates that compared with only one detection model in the entire process, MCCNN has a lift percentage. “MCCNN-Fully” represents the complete model of MCCNN, as shown in Fig. 2.

The core idea of MCCNN is to cascade a classification model based on the detection model, which combines three have higher detection ability. MCCNN aims to increase the value of precision based on less impact on the recall rate. It can be seen from Table V that the AP values of the paper, plastic, shell, and large bone have all been reduced based on MCCNN model with the single detection sub-network. For the MCCNN-YOLOv4 model, the AP value drops the least. However, MCCNN-fully has a better performance on the AP and mAP. Table VI shows the relevant changes in the recall rate. The MCCNN-YOLOv4 and MCCNN-DSSD models have a small decrease. MCCNN-Fully still has the highest recall rate. ‘d’ Although MCCNN-Fully has the highest accuracy, its size is more than 1.5G, and it is difficult to deploy on small embedded devices. Considering the model’s

size and detection effect, we finally chose the MCCNN-YOLOv4 model for testing in the STC system. The model size is 397MB, including 235MB of detection sub-network and 162MB of the classification network. We deployed the STC system model with an i5 processor, 8G memory, and no GPU. The loading of the model can be completed within 10s, and the detection time of each picture is completed within 500ms, which fully meets the daily needs of residents. The detection result of the waste picture is shown in Fig. 6.

TABLE V  
THE AP AND MAP OF MCCNN ON THE TESTING SET(%)

| Method            | AP on the waste dataset |         |       |            |           | mAP   |
|-------------------|-------------------------|---------|-------|------------|-----------|-------|
|                   | paper                   | plastic | shell | large bone | cigarette |       |
| MCCNN-DSSD        | 58                      | 58.1    | 26    | 44.2       | 20        | 41.3  |
| Lift              | ↓6.2                    | ↓13.5   | ↓14   | ↓9.7       | ↑0        | ↓8.6  |
| MCCNN-YOLOv4      | 55.3                    | 61.2    | 15.2  | 44.5       | 50        | 45.2  |
| Lift              | ↓1.4                    | ↓7.1    | ↓0    | ↓13.6      | ↑8.4      | ↓2.8  |
| MCCNN-Faster-RCNN | 67.2                    | 62.2    | 37    | 52.5       | 39.4      | 51.7  |
| Lift              | ↓8.2                    | ↓15.4   | ↓13.1 | ↓32.1      | ↑3.1      | ↓13.2 |
| MCCNN-Fully       | 77.1                    | 80.4    | 53    | 88.5       | 44.1      | 68.6  |

TABLE VI  
THE RECALL RATE OF MCCNN ON THE TESTING SET(%)

| Method            | Recall rate on the waste dataset |                           |                         |                           |                      | Ave                       |
|-------------------|----------------------------------|---------------------------|-------------------------|---------------------------|----------------------|---------------------------|
|                   | paper                            | plastic                   | shell                   | large bone                | cigarette            |                           |
| MCCNN-DSSD        | 60                               | 58.4                      | 26                      | 45.2                      | 20                   |                           |
| Lift              | $\uparrow\mathbf{0}$             | $\uparrow\mathbf{0.1}$    | $\downarrow\mathbf{14}$ | $\downarrow\mathbf{16.7}$ | $\uparrow\mathbf{0}$ | $\downarrow\mathbf{6.16}$ |
| MCCNN-YOLOv4      | 57.6                             | 62.5                      | 20                      | 46.1                      | 50                   |                           |
| Lift              | $\downarrow\mathbf{6.8}$         | $\downarrow\mathbf{9.4}$  | $\downarrow\mathbf{4}$  | $\downarrow\mathbf{15.4}$ | $\uparrow\mathbf{0}$ | $\downarrow\mathbf{7.12}$ |
| MCCNN-Faster-RCNN | 71.6                             | 67.7                      | 40                      | 54.7                      | 40                   |                           |
| Lift              | $\downarrow\mathbf{10}$          | $\downarrow\mathbf{16.9}$ | $\downarrow\mathbf{18}$ | $\downarrow\mathbf{38.1}$ | $\uparrow\mathbf{0}$ | $\downarrow\mathbf{16.6}$ |
| MCCNN-Fully       | 76.3                             | 73.8                      | 51.6                    | 58.6                      | 50                   |                           |

TABLE VII  
THE PRECISION OF MCCNN ON THE TESTING SET(%)

| Method            | Precision on the waste dataset |                         |                          |                         |                         | Ave                      |
|-------------------|--------------------------------|-------------------------|--------------------------|-------------------------|-------------------------|--------------------------|
|                   | paper                          | plastic                 | shell                    | large bone              | cigarette               |                          |
| MCCNN-DSSD        | 83.7                           | 88.4                    | 92                       | 90.5                    | 100                     |                          |
| Lift              | $\uparrow\mathbf{2.1}$         | $\uparrow\mathbf{29.9}$ | $\downarrow\mathbf{3.2}$ | $\uparrow\mathbf{20.3}$ | $\uparrow\mathbf{0}$    | $\uparrow\mathbf{9.82}$  |
| MCCNN-YOLOv4      | 79                             | 90.9                    | 47.6                     | 94.7                    | 100                     |                          |
| Lift              | $\uparrow\mathbf{11.2}$        | $\uparrow\mathbf{11.6}$ | $\uparrow\mathbf{3.2}$   | $\uparrow\mathbf{2.4}$  | $\uparrow\mathbf{50}$   | $\uparrow\mathbf{15.68}$ |
| MCCNN-Faster-RCNN | 62.3                           | 66.7                    | 37                       | 82.1                    | 33.3                    |                          |
| Lift              | $\uparrow\mathbf{5.4}$         | $\uparrow\mathbf{9.5}$  | $\uparrow\mathbf{2.9}$   | $\uparrow\mathbf{23}$   | $\uparrow\mathbf{11.1}$ | $\uparrow\mathbf{10.38}$ |
| MCCNN-Fully       | 85.7                           | 91.2                    | 93                       | 96.5                    | 100                     |                          |

## V. CONCLUSION

In this article, a novel deep convolutional neural network based on the Multi-model Cascaded method is proposed to detect and classify domestic waste, capable of increasing detection precision as much as possible while ensuring the high detection recall. We developed a smart trash bin(STC) system as the front-end carrier of domestic waste disposal, directly interacting with residents, providing data support for the entire platform. Additionally, we collected 30,000 waste images posted by residents as a data set for model training and labeled 52 types of waste. Simultaneously, MCCNN, a multi-target detection model for waste images, is introduced to maximize the detection precision. To the best of our knowledge, it is the first work to apply deep learning and other technologies to domestic waste treatment so systematically based on the mandatory waste classification. Our research aims to improve the data connection in the waste disposal link, and at the same time, improve the consciousness of residents' waste classification. The experiments show that our proposed method can improve detection precision performance with an average increase of more than 10%, and it also has good performance in model size and detection time. Recently our system has been applied to a community in Shanghai in China, which helped save time and make the waste recycling more efficient. With the gradual accumulation of garbage images, the model accuracy will be further improved in the next stage. We will also pay attention to waste detection and automatic sorting.



Fig. 6. Results of Testing. These are the waste images collected from the residents through the STC system. The STC system uses the MCCNN model to identify waste images. The non-kitchen waste in the image is marked by some rectangular frames, and there are five kinds, including shell, large bone, cigarette, paper, and plastic. The remaining waste unmarked is kitchen waste.

## REFERENCES

- [1] D. H. Perinaz. Bhada-Tata, "A Global Review of Solid Waste Management," *Urban Development and Local Government Unit, World Bank, Washington, DC 20433 USA*, pp. 1–116, 2012.
- [2] M. of Ecology and E. of the People's Republic of China, "2018 national annual report on the prevention and control of environmental pollution by solid waste in large and medium-sized cities," *Bulletin of China's Ecological Environment*, pp. 1–44, 2018.
- [3] N. Kollikkathara, H. Feng, and E. Stern, "A purview of waste management evolution: special emphasis on USA." *Waste Management*, vol. 29, no. 2, pp. 974–985, 2009.
- [4] A. Rovetta, F. Xiumin, F. Vicentini, Z. Minghua, A. Giusti, and H. Qichang, "Early detection and evaluation of waste through sensorized containers for a collection monitoring application," *Waste Management*, vol. 29, no. 12, pp. 2939–2949, 2009.
- [5] G. Mittal, K. B. Yagnik, M. Garg, and N. C. Krishnan, "SpotGarbage: smartphone app to detect garbage using deep learning," *ubiquitous computing*, pp. 940–945, 2016.
- [6] M. AdhithyaPrasanna, S. Kaushal, and P. Mahalakshmi, "Survey on identification and classification of waste for efficient disposal and recycling," *International journal of engineering and technology*, vol. 7, p. 520, 2018.
- [7] G. T. Mindy Yang, "Classification of Trash for Recyclability Status," *CS229 Project Rep*, 2016.
- [8] J. Donovan, "Auto-trash sorts garbage automatically at the techcrunch disrupt hackathon," 2016.
- [9] H. Panwar, P. Gupta, M. K. Siddiqui, R. Morales-Menendez, and I. H. Sarker, "AquaVision: Automating the detection of waste in water bodies using deep transfer learning," *Case Studies in Chemical and Environmental Engineering*, p. 100026, 2020.
- [10] P. F. Proenca and P. Simoes, "TACO: Trash Annotations in Context for Litter Detection," *arXiv: Computer Vision and Pattern Recognition*, 2020.

- [11] O. Adedeji and Z. Wang, "Intelligent Waste Classification System Using Deep Learning Convolutional Neural Network," *Procedia Manufacturing*, vol. 35, pp. 607–612, 2019.
- [12] D. Rutqvist, D. Kleyko, and F. Blomstedt, "An Automated Machine Learning Approach for Smart Waste Management Systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 384–392, 2020.
- [13] T. Liu, Y. F. Li, H. Liu, Z. Zhang, and S. Liu, "RISIR: Rapid Infrared Spectral Imaging Restoration Model for Industrial Material Detection in Intelligent Video Systems," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.
- [14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [15] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. Berg, "DSSD : Deconvolutional Single Shot Detector," *ArXiv*, vol. abs/1701.06659, 2017.
- [16] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv: Computer Vision and Pattern Recognition*, 2018.
- [17] A. Bochkovskiy, C.-Y. Wang, and H. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *ArXiv*, vol. abs/2004.10934, 2020.
- [18] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6154–6162.
- [19] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.
- [20] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, pp. 336–359, 2019.
- [21] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847, 2018.
- [22] I. Sato, H. Nishimura, and K. Yokoi, "APAC: Augmented PAttern Classification with Neural Networks," *Computer ence*, vol. abs/1505.03229, 2015.
- [23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.
- [24] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [25] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.
- [26] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *ArXiv*, vol. abs/1605.06409, 2016.
- [27] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," *CoRR*, vol. abs/1312.6229, 2014.
- [28] J. Redmon, S. K. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.
- [29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *European conference on computer vision*, pp. 21–37, 2016.
- [30] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6517–6525, 2017.
- [31] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 318–327, 2020.
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *CACM*, 2017.
- [33] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1800–1807.
- [34] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv: Computer Vision and Pattern Recognition*, 2017.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [36] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [37] N. Bodla, B. Singh, R. Chellappa, and L. Davis, "Soft-nms — improving object detection with one line of code," *2017 IEEE International Conference on Computer Vision*, pp. 5562–5570, 2017.
- [38] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2446–2455, 2019.