

Novel Smart Waste Sorting System based on Image Processing Algorithms: SURF-BoW and Multi-class SVM

Yijian Liu^{1,2}, King-Chi Fung², Wenqian Ding^{1,2}, Hongfei Guo^{1,3}, Ting Qu^{1,3} & Cong Xiao⁴

¹ Institute of Internet of Things and Logistics Engineering, Jinan University, Zhuhai 519070, China

² Department of Electronic Information Science and Technology, Electrical and Information College, Jinan University, Zhuhai 519070, China

³ School of Physical Internet, Jinan University, Zhuhai 519070, China

⁴ Department of Software Engineering, Electrical and Information College, Jinan University, Zhuhai 519070, China

Correspondence: Hongfei Guo, Institute of Internet of Things and Logistics Engineering, Jinan University, Zhuhai 519070, China. E-mail: ghf-2005@163.com; Ting Qu, School of Physical Internet, Jinan University, Zhuhai 519070, China. E-mail: quting@gdut.edu.cn

Received: May 14, 2018

Accepted: May 21, 2018

Online Published: June 2, 2018

doi:10.5539/cis.v11n3p35

URL: <https://doi.org/10.5539/cis.v11n3p35>

Abstract

Aiming at solving the waste sorting problems of smart environmental sanitation, this paper proposes a novel smart waste sorting system, which consists of two sub-systems including a hardware system and a software system. The hardware system is of a trash bin framework based on the core module Raspberry Pi and the software one is of an image classification algorithm platform based on SURF-BoW algorithm and multi-class SVM classifier. In our experiment, the images produced during training and testing are both obtained from webcam in our system and extra processing with affine transformation and noise-adding operation. The experimental results show that among the five categories of waste, the battery waste performs best with 100% classification accuracy. Besides, the average classification accuracy is up to 83.38%. Therefore, our system has reliable practicability and robustness, which is expected to be applied to deal with the waste sorting problems in our daily life.

Keywords: smart environmental sanitation, waste sorting, Raspberry Pi, image processing, SURF-BoW, multi-class SVM

1. Introduction

As an important part of a smart city, smart environmental sanitation relies on the Internet of Things (IoT) and mobile Internet of Things to realize real-time sanitation management, involving the aspects of human beings, cars and almost everything. Also, it provides rational plans for sanitation management modes and utilizes digital technology to improve sanitation operations (Wang & Cao, 2016). With the development of urbanization in China, the scale of cities and the number of residents are both soaring. At the same time, the amount of urban waste is also explosively raising, which becomes a complicated problem in city management (Gu, 2015). It is known that nowadays most of the urban waste around China is disposed by composting, landfilling or incineration. However, it causes air pollution and deteriorates the land, which will not ever be conducive to the recycling of resources. The above problems should be solved with an effective waste sorting method. However, the effect of current waste sorting solutions is never ideal. A research indicates that there are obvious differences between people's intention and behavior among waste sorting. The proportion of people willing to participate in waste sorting (82.5%) is significantly higher than that of people actually participating in it (13%) (Chen, Li & Ma, 2015).

Through literature research and data collection, we found that most of current automated waste sorting systems rely on hardware devices such as infrared sensor, metal sensor, etc. (Feng et al., 2014; Fan et al. 2017; Chen et al., 2014; Ye et al. 2017). Most of them identify metal materials or other special types of waste, and they become invalid once it comes to carton and plastic products. Moreover, there are few systems based on image processing methods. Meanwhile, there are little excellent smart waste sorting systems and the degree of intelligence and automation of waste sorting can be further improved. Therefore, the system combined with hardware platform and image processing software algorithm proposed in this paper has come up with better innovational solutions.

From the perspective of the algorithm, first of all, we need to extract feature points from waste images to form a feature descriptor and then build the Bag-of-Words model (BoW) (Wu et al., 2010). BoW is proposed by Sivic et al., which uses ideas and methods in the field of text analysis for reference (Sivic & Zisserman, 2003). There are various algorithms for generating visual dictionary descriptors, including the classic Scale-Invariant Feature Transform (SIFT) (Lowe, 2004), Gradient Location-Orientation Histogram (GLOH) (Huang et al., 2015), Principal Component Analysis Scale-Invariant Feature Transform (PCA-SIFT) (Tang, 2012) and Speeded Up Robust Features (SURF) (Bay et al., 2008), etc. Among them, SURF has an excellent effect on image recognition and classification with fast processing speed. SURF was initially proposed by Herbert Bay et al. It is an improvement of the SIFT algorithm, achieving faster operational speed and higher accuracy. Pan divides the image into blocks to improve the distribution of the extracted feature points, avoiding that the feature points are all located at the corner points and the crossing dots. The use of relative distance to eliminate false matching points improves the matching accuracy. However, there are problems. The lack of location information and the large dimensions extended the operation time (Pan, Hao & Zhao, 2017). Yan and his team creatively proposed a method to match SURF features with Delaunay triangular meshes. The experimental results demonstrate that there is good affine invariance and high accuracy (Yan, Jiang & Guo, 2014). Consequently, after studying the existing feature-points extraction algorithms and comparing different algorithms, we choose SURF to achieve this goal and combine BoW model to construct SURF-Bow algorithm. Secondly, we utilize specific multi-class classifier to train BoW and test samples while classification. Currently, the widely used classification algorithms include the K-Nearest Neighbour algorithm (KNN), Bayesian Classifier, and Support Vector Machine (SVM), etc. (MacQueen, 1967; Altman, 1992; Vapnik, 2013). In our system, we use SVM due to its better performance compared to other classifiers. Using SVM can help us find the global optimal solution while avoiding local convergence. It has better performance in small-sample, high-dimension and non-linear data sets, but it has a problem of demonstrating single feature. Fu et al. proposed an improved method for multi-featured SVM (Fu et al., 2011). They extract the integrated features from the target image, and then use Principal Component Analysis (PCA) to remove redundant information. Finally, they use Rbag to classify the SVM, which effectively improves the classification accuracy and speed. However, when the dimensionality of the data is high, the efficiency will be reduced. Wu et al. (Wu & Li, 2017) proposed a fast SVM classifier combining intrinsic decomposition and sampling learning, which reduces the consumed training time by increasing the operation rate of the nuclear matrix when the low-dimensional space is transformed into the high-dimensional space.

Aiming at solving the problem of waste sorting, we design a novel smart waste sorting system, which mainly includes two parts: a hardware platform and a software platform. The hardware platform is built with Raspberry Pi as the core controlling module, and use it to control two steppermotors in our framework to accomplish automatic garbage collection. While the software platform is based on image processing algorithms, combined with SURF-BoW algorithm and multi-class SVM classifiers. In the experiment, both waste training images and testing images are obtained based on our system. There are 3,000 training images with 600 in each category, and 1,000 testing images with 100 original images as well as 1200 processed images. The result shows that the classification accuracy of battery will perform the best with 100% accuracy rate, and the average classification accuracy rate is up to 83.38%. Therefore, the practicability of our system is reliable.

2. Design of Smart Waste Sorting System

In this section, this article will introduce the types of experimental waste, the design of the hardware platform and the software platform in details.

2.1 Categories of Experimental Waste

Before designing the system, it is important to determine which kinds of garbage need to be classified and what exactly they are. According to Sorting and Evaluation Standards of Urban Domestic Waste, urban waste is classified into six categories: recyclables, large-size refuse, compostable garbage, hazardous waste, combustible garbage, and others. Among them, important waste categories include recyclables, combustible waste and hazardous waste. Recyclables include paper, plastic, metal, glass and fabrics. Combustible waste refers to the waste that can be burned, such as abandoned plants and paper, as well as that is not suitable for recycling, such as abandoned plastic, rubber, old fabric supplies and useless wood. Hazardous waste refers to substances that are directly or potentially harmful to human health or the natural environment.

Based on the reference and reality, in order to maximize the efficiency of waste sorting, we finally determined the categories of sorting as follows: batteries, bottles, cans, paper-balls and paper-boxes.

2.2 Hardware System: Waste Sorting Hardware Framework based on Raspberry Pi

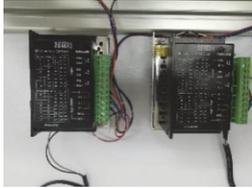
The design of the hardware platform mainly consists of two parts: system function design and hardware module design. The following section will minutely introduce our testing hardware framework and functional module. Table 1 shows our Hardware system function design:

Table 1. Hardware system design.

Platforms	Waste sorting system
Function describing	Combining image classification algorithms, sensors, motors, etc.; sorting 5 kinds of waste, including batteries, paper-balls, cans, bottles and paper-boxes.
Application	Mainly used as an environmental sanitation trash bin on the street; cannot being applied to sorting bagged waste (e.g. kitchen waste) and too much waste at once.
Modules	Core module, image acquisition module, stepping motor module, auxiliary operation module, and mechanical framework module.

More detailed information is shown in Table 2.

Table 2. Hardware modules.

Modules	Hardware	Demo picture	Description
Core module	Raspberry Pi 3 Model B		Controlling other modules and image processing platform.
Image acquisition module	720P webcam		Capturing video stream for motion detection and further image processing.
Stepping motor module	BYGH40 stepping motor, motor controller, 12V DC power supply	 	Mechanical device that makes the waste drop into the trash bin.

Auxiliary operation module	Wireless mouse, 10-inch LCD monitor with 1280*800 resolution		Auxiliary device used for software debugging.
Mechanical framework module	Aluminum frame with a size of 50cm*50cm*100cm and several stainless metal plates		Trash bin model and main testing platform.

Combining five modules shown above, we can establish the hardware system. The whole hardware system is shown in Figure 1.

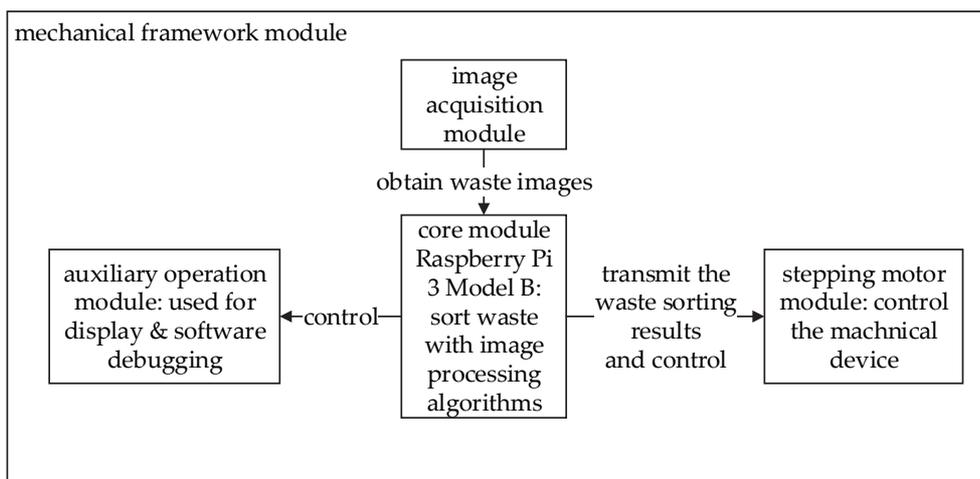


Figure 1. Hardware system

2.3 Software System: Waste Sorting Algorithm based on SURF-BoW and SVM

In the process of waste sorting algorithm, we combine SURF algorithm to extract feature points and the classic BoW model to design a novel image feature extraction algorithm SURF-BoW. Furthermore, we utilize multi-class SVM classifier to classify the waste images. The algorithm mainly consists of two parts, whose complete flowchart is shown in Figure 2.

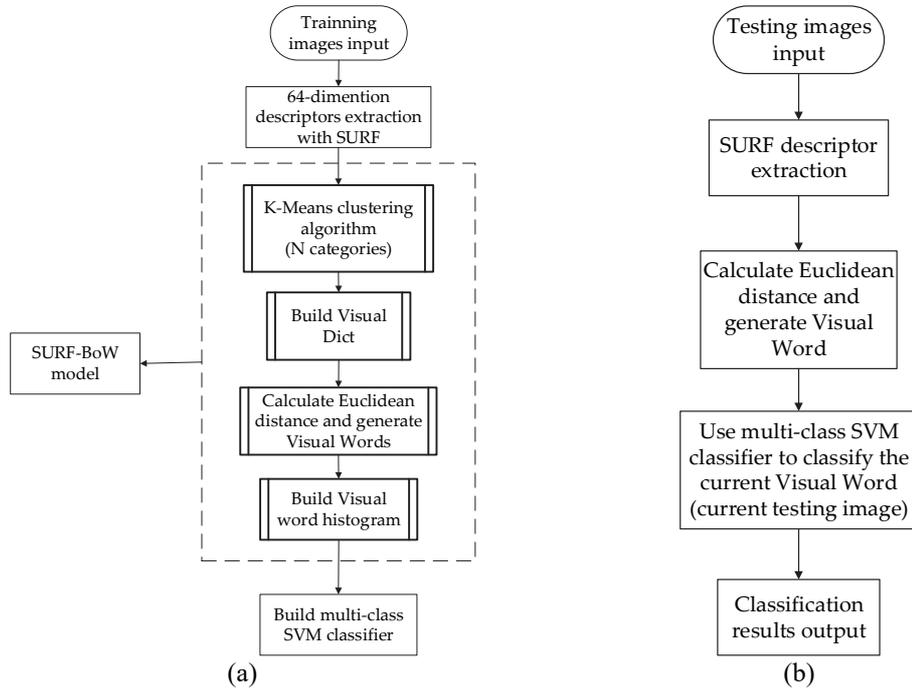


Figure 2. Flowchart of the software system. (a) Training images system; (b) Testing images system

2.3.1 SURF Feature points Extraction Algorithm

SURF algorithm (Speeded-Up Robust Features) is an accelerated version of SIFT algorithm (Scale-Invariant Feature Transform), which was proposed by Herbert Bay et al. The specific steps of the algorithm are as follows.

Step 1: Generate integral images

The concept of the integral image is proposed by Viola and Jones. Because the computation of the integral image in the different matrix regions is consistent, the integral image of the target image obtained before feature points' extraction can meet the fast operation of box filters with different sizes (Viola & Jones, 2001). It is calculated by summing up the pixel value of the diagonal region formed from each point in the original image to the top left corner of the image. The mathematical formula can be presented as:

$$I_{\Sigma(x)} = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x, y) \quad (1)$$

Step 2: Build Hessian matrix

Unlike the DoG images used in SIFT, SURF introduces the local maximum value of the determinant of the Hessian matrix to detect the feature points, that is, the detection of the stable edge points in the image, which can effectively improve the accuracy and the operating rate (Lowe, 1999). Given a certain point $X=(x, y)$ in the image, the corresponding Hessian matrix can be shown as:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2)$$

In the formula (2), $L_{xx}(x, \sigma)$ is the replacement formula of the convolution of Gaussian second-order differential

$\frac{\partial^2 g(\sigma)}{\partial x^2}$ on the point X . And the meaning of $L_{yy}(x, \sigma)$ and $L_{xy}(x, \sigma)$ is similar.

To speed up the extraction of feature points, Bay et al. proposed a box filter to replace Gaussian second-order differential (Bay et al., 2006). The Figure 3 shows the box filters.

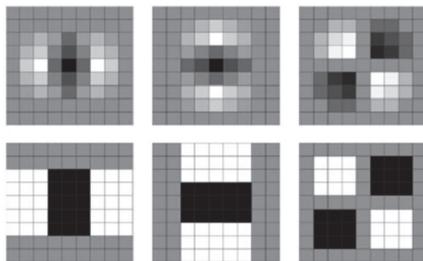


Figure 3. Box filters. Left to right: $L_{xx}(x, \sigma)$, $L_{yy}(x, \sigma)$ and $L_{xy}(x, \sigma)$

In Bay’s paper, they use a filter template with a size of 9*9 and introduce D_{xx} , D_{xy} and D_{yy} to replace derivative of Gaussian (DoG). In Figure 3, the white section of box filter $L_{xx}(x, \sigma)$ and $L_{yy}(x, \sigma)$ represents a weight of 1 and the dark section represents a weight of -2. The white section of box filter $L_{xy}(x, \sigma)$ represents a weight of 1 and the dark section represents a weight of -1. Also, Bay introduces a weight w to reduce the deviation caused by the DoG approximation. The discriminant of Hessian matrix is shown as below.

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2, \tag{3}$$

where $w = 0.9$.

Step 3: Establish scale pyramid

The scale pyramid is constructed by constructing different sizes of box filter templates. The octave number of the pyramid is 4 or 5 and this is because the number of feature points detected decreases rapidly as the scale increases. In SURF, the size of the images among different octaves is consistent, but the size of box filter templates gradually increases. Besides, in the same octave, the size of the box filters is the same, but the fuzzy coefficient gradually increases as the layer in the octave increases.

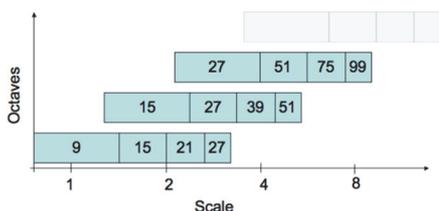


Figure 4. Box Filters Size in Different Octaves.

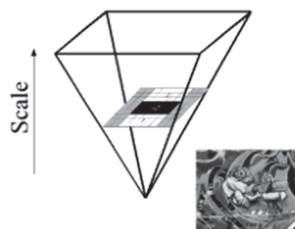


Figure 5. SURF Scale Pyramid.

After using the Hessian matrix to generate the extreme value, the 3-dimension linear interpolation is introduced to obtain the feature points of the pixels. At the same time, points which are below the threshold, low-contrast or on the edges are removed to ensure the stability of the algorithm.

Step 4: Extract feature descriptors

The SURF algorithm preserves the rotation invariance of SIFT, so a main direction to each feature point is needed. The main direction matching method is as follows:

- (1) Take the feature points as the centre and compute the Harr-wavelet response in the X and Y directions of the neighbourhood whose radius is 6S (S is the scale value of the feature point);
- (2) Introduce Gaussian weighting to get the direction vectors, which can describe the regional directions where the pixel values change sharply;
- (3) Sum up the response in the 60° range area to form a new vector, and traverse the entire circular region to select the longest vector as the main direction of the feature point;
- (4) Generate feature descriptor. Calculate the Harr-wavelet response of the image and choose a rectangular block with 4*4 size around the feature points. The direction is the main direction of the feature points.

In every sub-area, calculate the horizontal and vertical Harr response of 25-pixel values, which are d_x and d_y . Feature vectors are $\sum d_x$, $\sum |d_x|$, $\sum d_y$ and $\sum |d_y|$. They are shown as Figure 6.

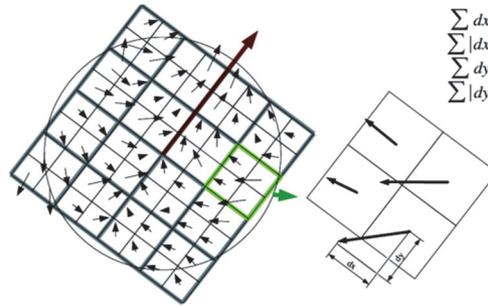


Figure 6. Representation of Feature Description Operator

2.3.2 SURF-BoW Model

BoW (Bag-of-Words) model is a widely used document representation method in the field of information retrieval (Zheng, Zhang & Yan, 2014; Yang & Peng, 2014). In this field, the BoW model treats a document as a collection of words, ignoring its word order, syntax and other elements. The occurrence of every word in the document is independent. That is to say, any word that appears in any location in the document is not related to syntax (Yang & Peng, 2014). In BoW model, each element in the vector represents the number of its related elements of the dictionary in the document, regardless of the word order. Therefore, we suppose that there is a document collection T, which stores a total volume of M documents. Then extract all the words in T and build the dictionary by these N words. Finally, each document can be represented by an N-dimension vector (Li, Xie & Wu, 2008).

Similar to text analysis, we can apply the BoW model to image classification. In this paper, in order to represent images of different waste categories, we combine the SURF feature points' extraction algorithm and the BoW model to construct a novel model SURF-BoW. The following section are the steps of constructing SURF-BoW models for training images.

Step 1: Build $database_{SURF}$ based on SURF algorithm

Introduce the SURF algorithm to extract the feature points of the training image $Image_i$, and obtain the descriptor (vector), which is a 64-dimension vector and will be stored in $database_{SURF}$. Judge whether all the training images has completed feature points detection. If not, then repeat Step 1, otherwise skip to Step 2.

Step 2: Build $Dict_{K \times 64}$ based on K-Means clustering

K-Means algorithm is an indirect clustering method based on similarity measurement among samples. This algorithm takes K as a parameter and divides N objects into K clusters. At the same time, the data in the same cluster has high similarity, while the similarity in different clusters is low.

Algorithm 1. K-Means

Input:	Cluster number K; input sample $D = \{x_1, x_2, \dots, x_M\}$
Output:	Output category labels $C = \{C_1, C_2, \dots, C_M\}$, where $C_i \in \{1, 2, \dots, K\} (i = 1, 2, \dots, M)$ is the corresponding category of data $x_i (i = 1, 2, \dots, M)$.
Algorithm:	<ol style="list-style-type: none"> 1: Obtain the dimension of D: Dim 2: Generate randomly K Dim-dimension points: $Center_i (i = 1, 2, \dots, K)$ 3: while (algorithm is not convergent) 4: for m = 1 to M 5: calculate the category C_m of all data 6: for k = 1 to K 7: Find all the data points of category k 8: Change the value of $Center_k$ to the average value of these data points

```

9:   end for
10:  end for
11: end while
    
```

Utilize K-Means algorithm to divide the $database_{SURF}$ into K categories, and finally calculate the K clustering centres to get the matrix form of the dictionary $Dict_{K \times 64}$.

Step 3: Build BoW for training images

Traverse all training images and extract descriptors (64 dimensional vectors) for each image. Then traverse all descriptors and calculate the Euclidean distance between it and every visual word in the dictionary.

$$distance = Calculate_Euclidean-distance(descriptor_{64 \times 1}, word_{1 \times 64}) \tag{4}$$

According to the nearest Euclidean distance, classify all descriptors and count frequency of different categories. Therefore, a K-dimension vector can be obtained from a training image, that is, the visual word histogram. Finally, store all the visual word histograms data and obtain a data matrix $BoW_{Q \times K}$ (bag of words, BoW), where Q is the quantity of training images, and K is the cluster number. Figure 7 shows the detailed steps of constructing BoW.

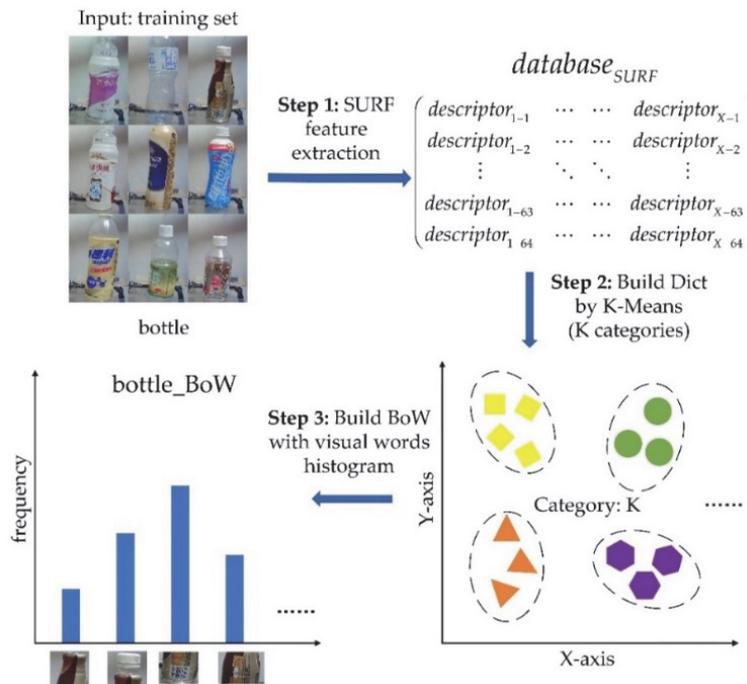


Figure 7. BoW constructing steps

2.3.3 Multi-class SVM Classifier

Support Vector Machine (SVM) is a supervised machine learning algorithm based on statistical learning theory. The purpose of the SVM classifier can be summarized as follows: finding a classification hyperplane, which can classify two categories of sample points, while the sample points are as far away from the hyperplane as possible (Zhou, 2016). Figure 8 is a visual representation of the SVM algorithm:

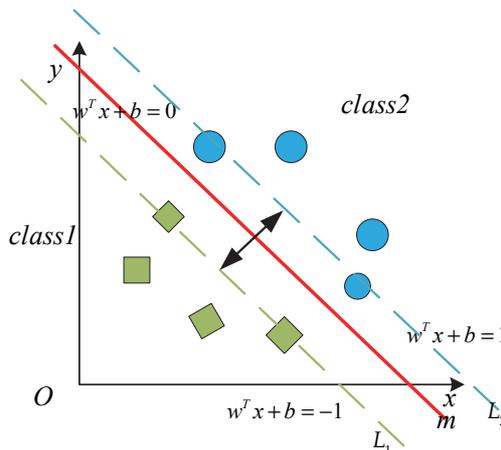


Figure 8. Best Classification Hyperplane in SVM

Figure 8 is an example of the application of SVM on linearly separable problem, in which the linear m is the best classification hyperplane in the two-dimensional space, the circular and square marks are all training data, and the lines L_1 and L_2 are support vectors. In the SVM algorithm, if all the data can be correctly separated by a hyperplane, and the distance between the two different categories of vectors is nearest, the hyperplane is the largest (i.e. the edge maximization), then the hyperplane is the optimal one, then the two vectors are called the support vectors.

The following section shows the classic SVM algorithm steps (Wu, 2009; Xue, 2011).

Step 1: Input binary classification training data

Given the training data $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, where $y \in \{-1, 1\}, x_i \in \mathbb{R}^d$, assume that training data can be linearly separated into two classes by a hyperplane. The mathematical expression of hyperplane is as follows.

$$(w \cdot X) + b = 0, \tag{5}$$

where, $w = (w_1, w_2, \dots, w_d)$ is the normal vector of the hyperplane, b is the displacement term of the hyperplane, which determines the distance between the hyperplane and the origin. Therefore, a hyperplane can be represented by a unique vector (w, b) .

Step 2: Calculate support vectors

SVM algorithm converts binary problems into optimization models, which calculates a vector $\alpha_{m \times 1}$ and the sample number m . Each element α_i of $\alpha_{m \times 1}$ is corresponding to a specific sample, and the sample point corresponding to a non-zero item α_i is the supportive vector point.

For linearly separable problems, the mathematical model can be shown as:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\ & \alpha_i \geq 0, i = 1, 2, \dots, m \end{aligned} \tag{6}$$

For those nonlinearly problems, utilize kernel function to solve them.

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\ & \alpha_i \geq 0, i = 1, 2, \dots, m \end{aligned} \tag{7}$$

where, $K(x_i, x_j)$ represents kernel function. In SVM, kernel function is a kind of function that calculates the

inner product of x_i and x_j in the feature space. It is used to solve the problem where the inner product is difficult to calculate in the high-dimension space.

Classic SVM algorithm provides following frequently-used kernel functions (Zhou, 2016).

Table 3. Frequently-used kernel functions.

Kernel Functions	Expression	Parameters
Linear kernel	$K(x_i, x_j) = x_i^T x_j$	N/A
Polynomial kernel	$K(x_i, x_j) = (x_i^T x_j)^d$	d is the polynomial order ($d \geq 1$)
Gaussian kernel	$K(x_i, x_j) = e^{-\frac{\ x_i - x_j\ ^2}{2\sigma^2}}$	σ is the width of Gaussian kernel ($\sigma \geq 0$)
Laplacian kernel	$K(x_i, x_j) = e^{-\frac{\ x_i - x_j\ }{\sigma}}$	σ is the width of Laplacian kernel
Sigmoid kernel	$K(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta)$	Hyperbolic tangent function ($\beta > 0, \theta < 0$)

In our experiment, we compare different kernels, choose linear kernel and finally we can obtain vector $\alpha_{m \times 1}$.

Step 3: Obtain the parameters of the hyperplane

After obtaining vector α , we can calculate the parameter (w, b) of the hyperplane.

$$w = \sum_{a_i \neq 0} a_i y_i x_i, b = y_i - w \cdot x_i \quad (8)$$

Step 4: Input testing samples and output classification results

Based on the previous work, we can input testing samples to test the training model.

For linearly separable problems, the mathematical model is as follows.

$$f(x) = \sum_{i=1}^m \alpha_i y_i x_i^T x + b \quad (9)$$

Similar to linearly separable problems, the mathematical model of nonlinearly one is:

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x, x_i) + b \quad (10)$$

If $f(x) > 0$, the testing classification result belongs to a positive category, otherwise it belongs to a negative one.

SVM classifier is mostly used in the binary classification problems. In our smart waste sorting system, we should classify 5 kinds of waste, so the binary-class classifier cannot meet the requirements. Actually, how to apply SVM effectively to multi-class problems has always been a problem for many scholars. In order to achieve the goal, we use multi-class SVM to classify waste images. The widely used multi-class SVM consists of “one-against-all” SVM, “one-against-one” SVM, DDAGSVM and binary tree SVM (Xue, 2011; Li et al., 2012; Shan et al., 2012).

In this paper, we choose the “one-against-all” SVM to build a multi-class SVM for its simplicity and efficiency. The following section is the multi-class SVM algorithm steps.

Algorithm 2. “one-against-all” SVM

Input: Category N, input $D = \{D_1, D_2, \dots, D_N\}$ for training samples; testing sample T.

Output: Categories of T.

Algorithm: 1: // training section

- 2: for n = 1 to N
- 3: Positive Sample ← D_n , Negative Sample ← other samples except D_n
- 4: Store the data of SVM_n classifier
- 5: end for
- 6: // testing section
- 7: for n = 1 to N
- 8: Use classifier SVM_n to calculate the value of $f(x)_n$
- 9: end for
- 10: Compare all $f(x)_n$, output the n corresponding to the maximum of $f(x)$

3. Experimental Test and Result Analysis

3.1 Experiment Preparation: Training and Testing Images

In our experiments, training and test images are obtained by camera in our waste sorting framework. The quantity of training images is 3000. In order to increase the robustness of the system, feature points are required to be multi-scale, so we introduce affine transformation (rotation, resizing etc.) to increase the quantity of training images. The quantity of training images is 1300, which consists of 100 original images and 1200 processed ones (all generated by 2 kinds of resizing transformation and 4 kinds of noise addition process including Gaussian noise, Gamma noise, Rayleigh noise and salt and pepper noise). The Table 4 is an example to show the differences between four different kinds of noise.

Table 4. Comparison between Four Different Noises.

	Original image	Salt and pepper noise	Rayleigh noise	Gamma noise	Gaussian noise
Samples					
SNR(dB)	N/A	25.297	7.686	2.828	0.487

Analysing the Table 4, we can find that SNR varies from the noise types. Images with Gaussian noise has the lowest SNR and that with salt and pepper constitutes the highest SNR. Here, the purpose of introducing different types of noise is to test the robustness of our system.

3.2 Experimental Results Analysis

In our software experiment, we use Code::Blocks 16.01 IDE on Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz Laptop with 16.00 GB RAM. Furthermore, we write C/C++ code using the OpenCV library to finish our experiment. In this paper, we identify a variable $\eta^{correct}$ to measure the classification accuracy of 5 categories (battery, bottle, can, paper-ball and paper-box):

$$\eta^{correct}_i = \frac{R_i}{T_i} (i = 1, 2, 3, 4, 5) \tag{11}$$

In the formula, $\eta^{correct}_i$ is the classification accuracy of the ith category waste; R_i is the quantity of correct

classification results of the i th category waste; T_i is the total quantity of the i th category waste image test set. Besides, the relationship between the variable i and the waste category is shown in Table 5.

Table 5. Relationship of the Variable i and the Waste Category

Variable i	1	2	3	4	5
Waste category	batteries	bottles	jars	paper-balls	paper-boxes

Based on the design of the smart waste sorting system, we obtain the results of classification accuracy $\eta^{correct}_i$ of each category and they are shown in Table 6.

Table 6. Classification accuracy $\eta^{correct}_i$

	Batteries	Bottles	Jars	Paper-balls	Paper-boxes	Total test set
Total quantity	260	260	260	260	260	1300
Successful quantity	260	167	212	256	189	1084
Failed quantity	0	93	48	4	71	216
$\eta^{correct}_i$	100%	64.62%	84.23%	98.08%	75.77%	83.28%

As shown in Table 6, the quantity of our test set is 1300 and the average classification accuracy is 83.28%. Among 5 categories we have tested, we can find that the classification accuracy of batteries and paper-balls are the best up to 100% and 98.08%, which means that we can apply our system in our daily life to recognize batteries and paper-balls. Besides, the classification effects of cans and paper-boxes are respectively 84.23% and 75.77%, which prove our system’s availability once again. And the lowest recognition rate is of bottles with classification accuracy at 64.62%. Although the result of bottles is not good, our smart waste sorting system is still effective and the reason will be mentioned in the following section.

Besides the result in Table 6, we also obtain the results of testing four different kinds of noise which are shown in Figure 9.

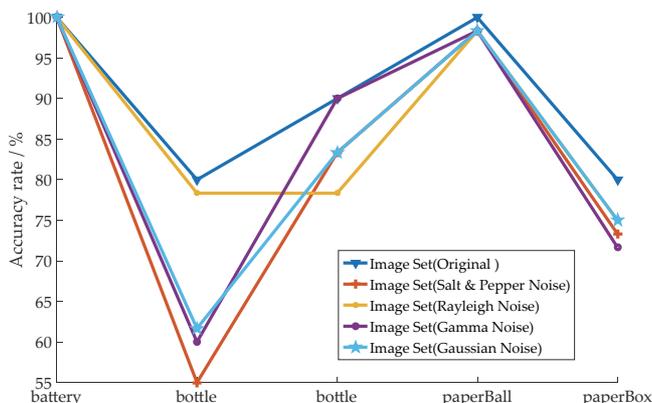


Figure 9. Classification Accuracy of Four Different Kinds of Noise.

As shown in Figure 8, our test set consists of five sections, including original images (without noise), and processed images with four different kinds of noises which are salt-and-pepper noise, Rayleigh noise, Gamma noise and

Gaussian noise respectively. For batteries, we can see that the classification accuracy is always up to 100%. Besides, the second-best result is from paper-balls and the classification accuracy is always over 90%. As for cans and paper-balls, no matter which types of noise, the former accuracy will all vary from 80% to 90% and the latter will vary from 70% to 80%. Especially for bottles, the worst one, the accuracy distribution has two extreme peak values: 60% and 80%.

Based on the analysis of the experimental results above, we can conclude that our smart waste sorting system can be applied in daily life to sort out the waste rather precisely among the waste of batteries, cans, paper-balls and paper-boxes. As for the waste bottles, the testing result is not satisfying and it is caused by low Signal to Noise Ratio (SNR). In the test of the original images and the high-SNR images with salt-and-pepper noise, the recognition rates of bottles and paper boxes are distributed around 80%; but in lower SNR test set, the accuracy declines. However, in daily life, the SNR of cameras in the system are much higher, which means that this result will not appear in the practical application process of our system.

Consequently, based on SURF-BoW algorithm and SVM multi-classifier, our smart waste sorting system can be applied in the field of smart environmental sanitation and smart cities to solve the waste sorting problems in our daily life, which means high practicability is guaranteed.

5. Conclusion

In this paper, aiming at solving the waste sorting problems in smart environmental sanitation, we propose a novel smart waste sorting system. The framework consists of two parts, one of which is the hardware platform with Raspberry Pi as the core module and the other is the software platform based on SURF-BoW and multi-class SVM algorithm. In the experiment, all training and testing images are obtained by the camera in our framework rather than the Internet. For training images, we introduce affine the transformation method to increase the quantity; for testing images, both affine transformation and different kinds of noise processing method are introduced for testing the robustness of our system.

Experimental results demonstrate that our smart waste sorting system has 83.38% average accuracy rate. Among five kinds of waste, batteries enjoy the best accuracy with 100% and the following two categories of paper-balls and cans are at 98.08% and 84.23 accuracy rate respectively, which all possess a satisfying practicability. As for paper-boxes and bottles, the accuracy rate of them are nearly 70%. Considering the SNR of testing images are so low that there are no images with such low SNR in the processing of application. Hence, our system will perform better in application.

In the long term, we will evaluate various feature extraction algorithms and classifiers to realize higher image classification accuracy. Meanwhile, we expect our waste sorting system could be utilized in the field of smart environmental sanitation to improve waste management in daily life.

Acknowledgments

This research was funded by the National Innovative Experiment Program for Undergraduate Students of Jinan University with the grant number No. 201710559020 and Challenge Cup Program with the grant number No. 17113032. The authors are thankful to the Institute of Internet of Things and Logistics Engineering, where the experiment is accomplished, and the significant guidance from our mentors Ting Qu and Hongfei Guo.

References

- Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175-185. <https://doi.org/10.2307/2685209>
- Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3), 346-359.
- Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.
- Chen, S. J., Li, R. C., & Ma, Y. B. (2015). Dissociation of willingness and behavior: A study on the classification mechanism of urban residents' domestic waste. *China Population Resources and Environment*, 25(9), 168-176.
- Chen, X., Fang, X. M., Huang, X. M., & Li, T. T. (2014). Design and Research of Automatic Recyclable Waste Bins. *Jiangxi Building Materials*, (5), 2-2. <https://doi.org/10.3969/j.issn.1006-2890.2014.05.002>
- Fan, X., Guo, F. Y., Gao, R., & Zhang, H. (2017). Design of Automatic Sorting Waste Bin Based on STM32 Control. *Technology Innovation and Application*, (25), 20-21.

- Feng, Y., Yao, J. H., Wang, J., & Fu, X. J. (2014). Discussion on Design of Automatic Garbage Sorting and Recycling Robots. *China Computer & Communication*, (11), 60-62.
- Fu, Y., & Ren, M. M. (2011). Image classification based on multiple features and improved SVM integration. *Computer Engineering*, 37(21), 196-198. <https://doi.org/10.3969/j.issn.1000-3428.2011.21.067>
- Gu, W. (2015). *Study on the classification and treatment of urban garbage in China* (Master's thesis, Nanjing University of Science and Technology). <https://doi.org/10.16116/j.cnki.jskj.2016.21.006>
<https://doi.org/10.1109/iccv.1999.790410>
- Huang, L., Chen, C., Shen, H., & He, B. (2015). Adaptive registration algorithm of color images based on SURF. *Measurement*, 66, 118-124. <https://doi.org/10.1016/j.measurement.2015.01.011>
- Introduction of Urban Domestic Waste Classification and Evaluation Standards. *Environmental Sanitation Engineering*, (2005), 63-64.
- Li, L., Fang, X. P., & Zhang, N. (2012). A new multi-classification method for SVM based on geometric distribution. *Computer Technology and Development*, (2012-11), 172-175.
- Li, X., Xie, Y. X., & Wu, L. D. (2008). Image classification system design based on SVM. *Journal of System Simulation*, (z1), 385-387. <https://doi.org/10.3969/j.issn.1004-731X.2006.z1.119>
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). iee.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110. <https://doi.org/10.1023/b:visi.0000029664.99615.94>
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(14), 281-297.
- Pan, J. P., Hao, J. M., & Zhao, J. P. (2017). Image Registration Improvement Algorithm Based on SURF. *Remote Sensing for Land & Resources*, 29(1), 110-115. [https://doi.org/10.13873/J.1000-9787\(2017\)11-0151-03](https://doi.org/10.13873/J.1000-9787(2017)11-0151-03)
- Shan, Y. G., Wang, H., & Dong, S. (2012). Improved one-to-one support vector machine multi-classification algorithm. *Computer Engineering and Design*, 33(5), 1837-1841.
- Sivic, J., & Zisserman, A. (2003, October). Video Google: A text retrieval approach to object matching in videos. In *null* (p. 1470). IEEE. <https://doi.org/10.1109/iccv.2003.1238663>
- Tang, J. X. (2012). *Image classification and retrieval based on Bag of Features [D]* (Doctoral dissertation, Xidian University).
- Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.
- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (Vol. 1, pp. I-I). IEEE. <https://doi.org/10.1109/cvpr.2001.990517>
- Wang, S. B., & Cao, M. (2016). Status and trends of development of smarter sanitation in China. *Science and technology for construction*, (21), 26-28.
- Wu, D. H. (2009). Research on Intelligent Aided Quality Diagnosis Based on Multi-Class Support Vector Machine. *Journal of System Simulation*, (6), 1689-1692.
- Wu, H. Y., & Li, W. P. (2017). Fast SVM classifier combining intrinsic decomposition and sampling learning. *Electronic Technology Applications*, (9), 141-145. <https://doi.org/10.16157/j.issn.0258-7998.170174>
- Wu, L., Hoi, S. C., & Yu, N. (2010). Semantics-preserving bag-of-words models and applications. *IEEE Transactions on Image Processing*, 19(7), 1908-1920. <https://doi.org/10.1109/tip.2010.2045169>
- Xue, J. J. (2011). Comparative study of multi-class support vector machine classifiers. *Computer Engineering and Design*, 32(5), 1792-1795.
- Yan, Z. G., Jiang, J. G., & Guo, D. (2014). Image matching based on SURF features and Delaunay triangular meshes. *Acta Automatica Sinica*, 40(6), 1216-1222. <https://doi.org/10.3724/SP.J.1004.2014.01216>
- Yang, Q., & Peng, J. Y. (2014). Chinese Sign Language Recognition Algorithm Based on Depth Information and SURF-BoW. *Pattern Recognition and Artificial Intelligence*, 27(8), 741-749.
- Yang, Q., & Peng, J. Y. (2014). Chinese Sign Language Recognition Research Using SIFT-BoW and Depth Image

Information. *Computer Science*, 41(2), 302-307. <http://10.3969/j.issn.1002-137X.2014.02.066>

Ye, G. Y., Xiong, Z. Y., Li, M. S., & Chen, X. (2017). Design of smart bins for the monitoring of smart bins. *Digital Technology and Application*, 1, 87. <https://doi.org/10.3969/j.issn.1008-4800.2017.13.066>

Zheng, Z., Zhang, Y., & Yan, L. (2014). Global and local exploitation for saliency using bag-of-words. *IET Computer Vision*, 8(4), 299-304. <https://doi.org/10.1049/iet-cvi.2013.0132>

Zhou, Z. H. (2016). *Machine learning*. Tsinghua University Press.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).