Hybrid Particle Swarm Optimization and Backpropagation Neural Network for Organic and Inorganic Waste Recognition

Christ R.A. Djaya, Novi Sucianti, Randy, and Lili A. Wulandhari(✉)

School of Computer Science, Bina Nusantara University,
Jl. K.H. Syahdan No. 9, 11480 Jakarta, Indonesia
christflvy@gmail.com, suciantinovi@gmail.com,
randygun24@gmail.com, lwulandhari@binus.edu

Abstract. Separation of organic and inorganic waste for daily need is one of efforts to yield sanitation. However, most people have difficulties to distinguish these kind of waste. Therefore this paper propose a system that can recognize organic and inorganic waste automatically. These system is developed using hybrid PSO-BPNN algorithm to recognize type of waste. Input data is organic and inorganic image which is captured around campus. This paper also presents comparison of BPNN, PSO and PSO-BPNN in recognizing type of waste. The results show that each algorithm achieves 77%, 69% and 95% for BPNN, PSO and hybrid PSO-BPNN respectively.

Keywords: Waste recognition · Organic · Inorganic · Hybrid PSO-BPNN · PSO · BPNN

1 Introduction

Waste is one of the common problem that government has been faced from time to time in every city in Indonesia, especially Jakarta. In 2003, Jakarta has produced 5000 tons of waste in a day from their citizen which is approximately 10 million. Even, Indonesia has reached 100.000 tons in a day and it increases every year by 4% [1], and it requires government concern to overcome this problem. It makes the government must work hard to overcome this problem.

Waste is divided into organic and inorganic based on content and degradation level. Organic waste contains organic compound and easy to be degraded like leaf litter and leftovers. While, inorganic waste is the reverse of organic waste, it does not contain organic compound and hard to be degraded like plastics and cans [2]. Organic and inorganic waste should be disposed and placed in different places, because some of the waste may contain any poisonous material which can harm the environment. However, in some place organic and inorganic waste separation has not been handled properly because it is often found waste are mixed in the same bin. It happens because people still have difficulty in determining whether a waste is organic or inorganic. Therefore, it is required a tool that can assist people to separate organic and inorganic waste automatically.

Previous researchers had been proposed some techniques in waste separation. Salmador et al. (2008) proposed K-Nearest Neighbor (KNN) to separate household and industrial waste which is applied in robot. Another researcher proposed mechanical approach to sort solid waste such as metal, plastic, glass, paper and old tiers to be recycled [3]. Intelligent system also be applied by previous researcher to separate inorganic such as aluminum cans, plastic bottles and plastic cutleries. They applied computer vision approach and KNN to identify type of wastes [4]. Waste separation especially in Indonesia, needs a system to recognize organic and inorganic automatically. Based on previous researches, it is found that researchers only proposed technique to recognized inorganic waste. Therefore, this paper presents a technique to identify organic and inorganic by using computational intelligence especially Backpropagation Neural Network (BPNN) and Particle Swarm Optimization (PSO).

BPNN is one of the most popular technique in recognition [5–8] which is implemented in machine, health, voice recognition and electrical field. BPNN is a suitable approach for modeling behavior of a system even if the data are affected by noise [9, 10]. However, BPNN encounters drawback due to random initial weight which affects to local minimum and bad convergent velocity [11]. Therefore, it is important to optimize initial weight of BPNN to overcome this weakness. PSO is one of the optimization technique that can be used in initial weight optimization.

PSO is a recognition algorithm that can find a set of unknown parameter better [12]. Compared to Genetic Algorithm (GA), PSO has a better performance since PSO has the ability to do global and local search simultaneously [13]. Moreover, PSO has some advantages like easy to be implemented and more flexible in maintaining the balance of global and local search from their search space [14].

This paper presents a technique to recognize organic and inorganic waste by using waste images as the input. Computer vision approach is implemented to process raw image data to obtain features which can be used as input in hybrid PSO-BPNN. This paper is arranged as follows; Section 1 explains problem background and current research in waste identification, Sect. 2 discusses about data collection and analysis, Sect. 3 describes proposed methodology which follows with result and discussion in Sect. 4. Finally, conclusion will be represented in Sect. 5.

2 Image Collection and Preprocessing

This research uses organic and inorganic waste images as the data which is usually found around college. Total data collected are 40 images namely 20 organics and 20 inorganics image with dimension 960×700. Organic data consist of paper, tissue, leaf and box of milk (see Fig. 1) where inorganic data consist of plastic, can and bottle (see Fig. 2). Each image will pass preprocessing for instances; grayscaling, edge detection and feature extraction before used as input data.

First step of image pre-processing is Grayscaling, this process will change the image which contain 3 layers of color to grayscale image which contains only one layer of color (see Fig. 3). The formula to convert color image to grayscale image is given as follows [15]:
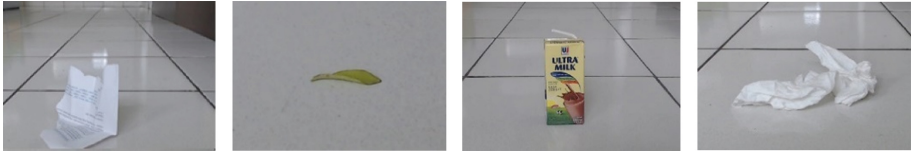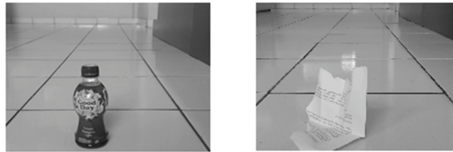
Fig. 1. Organic waste

Fig. 2. Inorganic waste

(a)

(b)

Fig. 3. Image before grayscaling (a), image after grayscaling (b)

$$I_g(i,j) = \frac{R(i,j) + G(i,j) + B(i,j)}{3} \tag{1}$$

Where

I_g = Intensity value of grayscale image at i,j pixel
R = Intensity value of red color on image at i,j pixel
G = Intensity value of green color on image at i,j pixel
B = Intensity value of blue color on image at i,j pixel

Grayscale image that we got from grayscaling process will through an edge detection process to detect each edge from the image object because training and
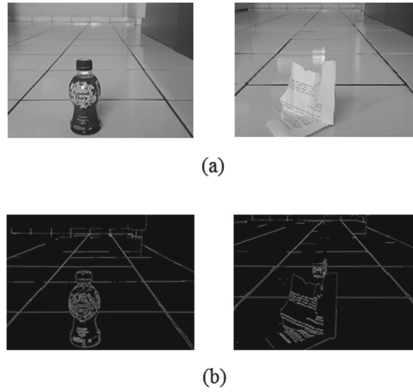
(a)

(b)

Fig. 4. Grayscale image (a) binary image after canny edge detection (b)

recognizing will be done based on object shape at the image. Canny Edge Detection algorithm is applied to obtain edge of image (see Fig. 4). This algorithm is well known for its performance in handling noise [16]. Process of canny edge detection is divided into five steps as follows:

Smoothing. Smoothing is a process of softening image to reduce the noise, usually using Gaussian Filter. Every image that taken from camera will have some noise and to avoid incorrect detection of noise into the edge, smoothing must be done.

Finding Gradient. Usually, Canny detect edge point on grayscale which had highest intensity value changes. Those areas discovered by determining gradient of an image which is calculated using Gradient Magnitude formula:

$$G = \sqrt{G_x^2 + G_y^2} \tag{2}$$

Where

G = Gradient magnitude
G_x = Gradient of horizontal direction (x)
G_y = Gradient of vertical direction (y)

As for calculating the angle direction will use the following formula:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{3}$$

Where

θ = Gradient of angle direction
G_x = Gradient of horizontal direction (x)
G_y = Gradient of vertical direction (y)

Angle produced from the calculation will be rounded to 0, 45, 90 and 135° based on closure.

Non Maximum Suppression The next step is get rid of non-maximum values because only local maximum that will be used as edge. Other pixel which have value that are not local maximum will be lowered to zero.

Double Thresholding. The remaining edge pixels from the previous step will be marked as strong pixels. Most of them is the real edge on image, but some may be caused by noise or color variations by rough surfaces. So, to distinguish them is by using a threshold value so that only the edge with a strong value will be kept. Canny algorithm uses double thresholding system (upper threshold and lower threshold) where the edges with a value higher than the upper threshold marked as strong point. Edges with a value less than lower threshold will be deleted and edges with value between upper threshold and lower threshold will be marked as weak point and will be deleted, except it is connected to edge which have strong point.

Edge Tracking by Hysteresis. To know the relation between each edge, edge tracking done using Binary Large Object. Binary Large Object applied by looking at weak point and the 8-Neighborhood. As long as there is a strong pixel on Binary Large Object, the weak pixel could be identified as something that can be included or stored.

2.1 Feature Extraction

Feature extraction is required to reduce number of neuron which be used in recognition process. Original binary image has 960×700 dimension which will represent as neuron in input layer. Feature extraction is carried out by segmented the original image into 48×35 dimension and calculate number of element 1 in each segmentation. This segmentation produces 20×20 features which be represented into 400 input neurons for every image (see Fig. 5).

(a)   (b)

Fig. 5. Binary image (a) extraction feature result (b)

3 Organic-Inorganic Recognition Technique

Organic-inorganic identification algorithm uses the combination of BPNN and PSO as optimization algorithm. PSO will take part in optimizing weight and bias value where every set of weight and bias will be used as particles in PSO. After PSO performed, the optimized set of weight and bias will be used for BPNN. The scheme of PSO-BPNN is given in a Fig. 6.
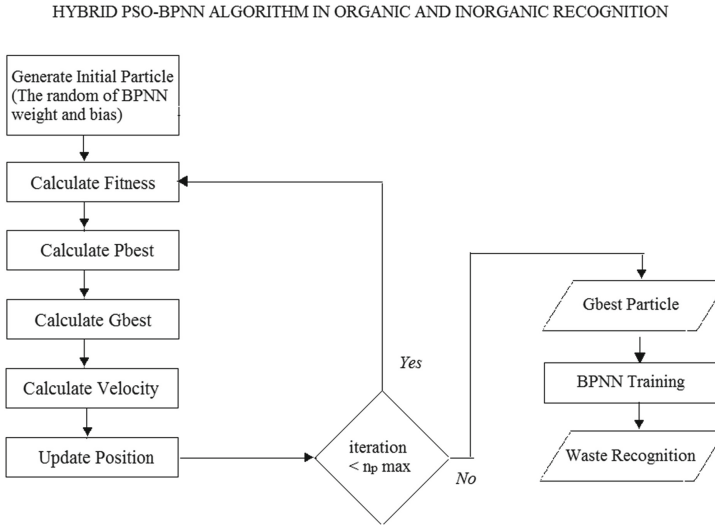
HYBRID PSO-BPNN ALGORITHM IN ORGANIC AND INORGANIC RECOGNITION

Fig. 6. Hybrid PSO-BPNN flowchart

Weight and bias values are taken a randomly in range $[-2, 2]$ that will be used as initial position of particle. Fitness function of PSO is obtained from Mean Square Error (MSE) with output O is obtained from forward process at BPNN which apply binary sigmoid as activation function.

$$O = \frac{1}{1 + e^{-(\sum w \cdot I + b)}} \tag{4}$$

Where w and b is set of weights and bias respectively. The algorithm of PSO-BPNN is described in the following steps:

1. Specify the topology of the BPNN to determine number of initial weights as the initial swarm of particle.
2. Let (I_b, T_b) is the bth input and target pair for BPNN, with $b = 1, 2, \ldots, N_{data}$ and N_{data} is the number of paired data.
3. Let N_i, N_x and N_{it} is the number of particle, number of dimension and number of iterations respectively, and $t_{it\,max}$ is maximum iteration.

4. Generate an initial particle. Each particle has dimension which correspond to BPNN random weights.
5. Set $t = 0$

   While $t \leq t_{it\,max}$ do

   Calculate fitness value $F(x_i)$ of the ith particle in swarm S_i.

$$F(x_i) = E(x_i, I_b, T_b) \qquad i = 1, 2, \ldots, N_i \tag{5}$$

Where $E(x_i, I_b, T_b)$ is the Mean Square Error (MSE) of the ith particle using (I_b, T_b) pair of data. It is calculated based on the selected BPNN architecture and activation function;

$$E(x_i, I_b, T_b) = \frac{1}{2} \sum_{b=1}^{N_{data}} (T_b - O_{b(x_i)})^2 \tag{6}$$

Where T_b is Target of the bth input and $O_{b(x_i)}$ is Output of the bth input which obtain using ith particle based on the selected BPNN architecture.

Let x_i denote position of particle ith and y_i denote the best position of particle ith

For each particle $i = 1, 2, \ldots, N_i$ do

   If $F(x_i) < F(y_i)$ then

    $y_i = x_i$

   End

Let $\hat{y}$ denote the global best position in a swarm

   If $F(y_i) < F(\hat{y})$ then

    $\hat{y} = y_i$

   End

For each particle $i = 1, 2, \ldots, N_i$ do

   Update the velocity using equation:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[\hat{y}_j(t) - x_{ij}(t)\right] \tag{7}$$

Where $v_{ij}(t)$ is the velocity of ith particle in dimension $j = 1, 2, \ldots, N_x$ at iteration t. $x_{ij}(t)$ is the position of ith particle in dimension $j = 1, 2, \ldots, N_x$ at iteration t. w is inertia weight, where this research uses linear decreasing inertia weight [17] c_1 and c_2 are positive constant acceleration, $r_{1j}(t)$ and $r_{2j}(t)$ are random values in range $[0, 1]$.

Update the position using equation:

$$x_{ij}(t+1) = v_{ij}(t) + v_{ij}(t+1) \tag{8}$$

With $x_{ij}(0)$ in range (x_{min}, x_{max}).

$$t = t + 1$$

   End
6. Apply the best solution in current swarm as the initial weights for BPNN learning.

The BPNN learning from the BPNN part produces recognition of the organic and inorganic waste where its performance is validated by calculating MSE and percentage of accuracy. The performance results of PSO-BPNN algorithms are presented in Sect. 4.

4   Result and Analysis

This section presents performance evaluation of PSO-BPNN and its comparison with individual BPNN and individual PSO in MSE and percentage of accuracy. Percentage of accuracy is defined as follows [18]:

$$Accuracy(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \qquad (9)$$

Assuming percentage of accuracy above is corresponds to the P class, where TP is true positives, which means that the prediction gives correct classification of P class according to the actual; TN is true negatives, which means the prediction gives correct label classification for all non-P class;FN is false negatives, which means the prediction gives incorrect classification label of the actual P class and FP is false positives, which means the prediction gives incorrect classification labels another class as P class.

Performance evaluation of PSO-BPNN is conducted by using 400-400-1 BPNN architecture, namely 400 neurons in input layer, 400 neurons in hidden layer and 1 neuron in output layer. The data is divided into 80 data for training and 20 data for testing. Training is conducted 5 times for 50 and 100 iterations for each algorithm with learning rate 0.5 for BPNN, 0.3 and 0.5 for c_1 and c_2 respectively for PSO. The comparison of BPNN, PSP and PSO-BPNN performance is given in Table 1 as follows.

From training result with 50 iterations, it is known that BPNN has MSE 0.051201 and 11.56 s for processing time. While PSO has 0.120857of average error and 49.31 s of average processing time which is 0.069 higher compare with BPNN. Hybrid PSO-BPNN has the lowest MSE with difference 0.04 with BPNN. While for training with 100 iterations, it is obtain that hybrid PSO-BPNN also has the lowest MSE compared with BPNN and PSO. Testing results for 50 and 100 iterations, also shows that Hybrid PSO-BPNN has the highest accuracy and achieve around 22% and 17% higher compare with BPNN for 50 and 100 iteration respectively.

Table 1. Training result

Iteration	Algorithm	Average error	Processing time (s)	Testing accuracy
50	BPNN	0.051201	11.56	70%
	PSO	0.120857	49.31	62%
	Hybrid PSO-BPNN	**0.003760**	**13.87**	**92%**
100	BPNN	0.053751	22.68	77%
	PSO	0.120293	110.77	69%
	Hybrid PSO-BPNN	**0.003751**	**25.26**	**95%**

5   Conclusion

This paper presents hybrid PSO-BPNN for organic and inorganic recognition. The experiments result show that PSO is capable to optimize weights of BPNN, such that BPNN learning can achieve good result. From the evaluation PSO-BPNN has higher accuracy compare to individual BPNN with only 2.58 s for additional processing time. This results shows that PSO-BPNN has good performance and can be implemented to develop organic-inorganic separation system.

References

1. Hartono, R.: Penanganan dan Pengolahan Sampah. Penebar Swadaya, Jakarta (2008)
2. Sejati, K.: Pengolahan Sampah Terpadu dengan Sistem Node, Sub Point dan Center Point. Kanisius, Yogyakarta (2009)
3. Huang, J., Pretz, T.. Bian, Z.: Intelligent solid waste processing using optical sensor based sorting technology. In: International Congress on Image and Signal Processing (CISP 2010), pp. 1657–1661 (2010)
4. Torres-García, A., Rodea-Aragón, O., Longoria-Gandara, O., Sánchez-García, F., González-Jiménez, L.E.: Intelligent waste separator. Computación y Sistemas **19**(3), 487–500 (2015)
5. Payganeh, G., Khajavi, M.N., Ebrahimpour, R., Babaei, E.: Machine fault diagnosis using MLPs and RBF neural networks. In: Applied Mechanics and Materials, pp. 5021–5028 (2012)
6. Zhao, K., Wang, C., Hu, J., Yang, X., Wang, H., Li, F., Zhang, X., Zhang, J., Wang, X.: Prostate cancer identification: quantitative analysis of T2-weighted MR images based on a back propagation artificial neural network model. Sci. China Life Sci. **58**, 666–673 (2015)
7. Hosom, J.-P., Vermeulen, P.J., Shaw, J.: Speaker verification and identification using artificial neural network-based sub-phonetic unit discrimination. United States Patent US 9230550 B2 (2016)
8. Roman, A.J., Kreitzer, P.J., Ervin, J.S., Hanchak, M.S., Byyd, L.W.: Flow pattern identification of horizontal two-phase refrigerant flow using neural networks. Int. Commun. Heat Mass Transf. **71**, 254–264 (2016)
9. Puscasu, G., Palade, V., Stancu, A., Buduleanu, S., Nastase, G.: Sisteme de Conducere Clasice si Inteligente a Proceselor. MATRIX ROM, Bucharest (2000)
10. Bocaniala, C.D., Palade, V.: Computational intelligence methodologies in fault diagnosis: review and state of the art. In: Palade, V., Jain, L., Bocaniala, C.D. (eds.) Advanced Information and Knowledge Processing, pp. 1–36. Springer, London (2006)
11. Nawi, N.M., Khan, Abdullah, Rehman, M.Z.: A new back-propagation neural network optimized with cuckoo search algorithm. In: Murgante, B., et al. (eds.) ICCSA 2013. LNCS, vol. 7971, pp. 413–426. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39637-3_33
12. Zhao, H.-B., Yin, S.: Geomechanical parameters identification by particle swarm optimization and support vector machine. Appl. Math. Modell. **33**(10), 3997–4012 (2009)

13. Rajendra, R., Pratihar, D.K.: Particle swarm optimization algorithm vs genetic algorithm to develop integrated scheme for obtaining optimal mechanic structure and adaptive controller of a robot. Intell. Control Autom. **2**(4), 430–449 (2011)
14. Sathya, P.D., Kayalvizhi, R.: PSO-based Tsallis thresholding selection procedure for image segmentation. Int. J. Comput. Appl. **5**(4), 39–46 (2010)
15. Kanan, C., Cottrell, G.W.: Color-to-grayscale: does the method matter in image recognition? Plos One **7**(1), e29740 (2012)
16. Shrivakshan, G.T., Chandrasekar, C.: A comparison of various edge detection techniques used in image processing. IJCSI Int. J. Comput. Sci. Issues **9**(5), 272–276 (2012)
17. Bansal, J.C., Singh, P.K., Saraswat, M., Verma, A., Jadon, S.S., Abraham, A.: Inertia weight strategies in particle swarm optimization. dalam: 2011 Third World Congress on Nature and Biologically Inspired Computing (2011)
18. Kohavi, R., Provost, F.: Glossary of terms: special issue on applications of machine learning and the knowledge discovery proces. Mach. Learn. **30**, 271–274 (1998)
19. Salmador, A., Cid, J.P., Novelle, I.R.: Intelligent garbage classifier. Int. J. Interact. Multimed. Artif. Intell. **1**(1), 31–36 (2008)