International Journal of Innovative Computing, Information and Control
Volume **15**, Number **5**, October **2019**

 ISSN 1349-4198
pp. **1901–1912**

CLASSIFICATION OF METAL OBJECTS USING DEEP NEURAL NETWORKS IN WASTE PROCESSING LINE

Tuan Linh Dang[1], Thang Cao[2] and Yukinobu Hoshino[3]

[1]School of Information and Communication Technology
Hanoi University of Science and Technology
No. 1, Dai Co Viet, Hai Ba Trung, Hanoi 100000, Vietnam
linhdt@soict.hust.edu.vn

[2]Machine Imagination Technology Corporation (MITECH)
Fuchu, Koyanagi 3-7-87, Tokyo 183-0013, Japan
cao@mitech.jp

[3]School of Systems Engineering
Kochi University of Technology
185 Miyanokuchi, Tosayamada, Kami City, Kochi 782-8502, Japan
hoshino.yukinobu@kochi-tech.ac.jp

Received March 2019; revised July 2019

Abstract. *Each year, a factory releases a lot of metal debris which is normally used in a recycling phase. In order to be effectively recycled, it is necessary to classify the debris into different classes. The sorting by hand takes a lot of times and effort. Other classification approaches which use color, size, weight, electrostatic, or magnetic features may not obtain high accuracy. It has a lack of technique to classify the metal debris. Thus, this paper proposes a framework for classification of metal debris which is spread on a conveyor belt. The framework employs deep neural networks. Four different deep neural network models were investigated and compared in our framework called the AlexNet model, the GoogleNet model, the VGGNet model, and the ResNet model to choose a suitable model for the framework. In addition, the experiments can also investigate and compare the operation of different deep neural network models in a practical application instead of using conventional academic benchmarks. Experimental results demonstrated that the proposed framework could be one solution to separate the metal debris. Especially, the AlexNet model had the highest accuracy among the four models.*
Keywords: Deep neural network, Blob detection, Run-length code processing, Metal objects, Classification

1. **Introduction.** Nowadays, car assembly lines or electronics assembly lines usually generate a lot of debris such as screws, wires, small excess pieces. This debris is normally collected and gathered into one place as wastes. The processing of these wastes is a major environmental issue, especially, inside an industrial zone. More and more debris is released in daily activities. The influences of economic growth and the needs for environmental protection are generating some research question for garbage management. Normally the debris is processed through an electrostatic machine to separate plastic debris and metal debris. The collected metal objects and the collected plastic objects will be further processed for recycling.

However, there are varieties of metal pieces such as copper, brass, bronze, and aluminum. It requires much time and effort to manually identify and classify different classes of metal objects. It is necessary to have a method to classify metal objects. If the objects

DOI: 10.24507/ijicic.15.05.1901

could be well separated, the human time, effort, and manpower for its recycling will be reduced, especially in the metallurgy phase [1, 2].

Previous research had focused on debris classification but they tried to separate solid objects and liquid objects [3], separate air-bubbles and debris particles [4]. Other studies normally classified the garbage which has different materials such as metal, glass, plastic, and paper [5, 6, 7]. Several researchers also investigated the classification of different metal debris using color-based method [8, 9], density-based method [10, 11], and shape-based method [12]. However, these methods could be hard to separate the materials because the metal debris could have similar features concerning shape, color, density, etc. The magnetic-based approach was also used to separate the nonferrous metals and ferrous metals [13, 14, 15]. Another method used laser-induced breakdown but it requires a lot of hardware devices, for example, a device to create a Q-switched Nd: YAG laser, collection lens, a charge-coupled device (CCD) spectrometer, an optical fiber, a laser sensor [16]. It is needed to focus on the classification approach of metal debris which has similar materials and this approach does not require a lot of expensive devices.

The main contribution of this paper is to propose a framework to classify the metal objects in a factory. The metal debris is spread on a conveyor belt to become the input of our system. The debris is processed in this system. In the factory, classified objects and outputs materials are put one by one into the wastes boxes which have corresponding labels such as bronze or copper. The position of the debris classification in the wastes process line can be seen in Figure 1.

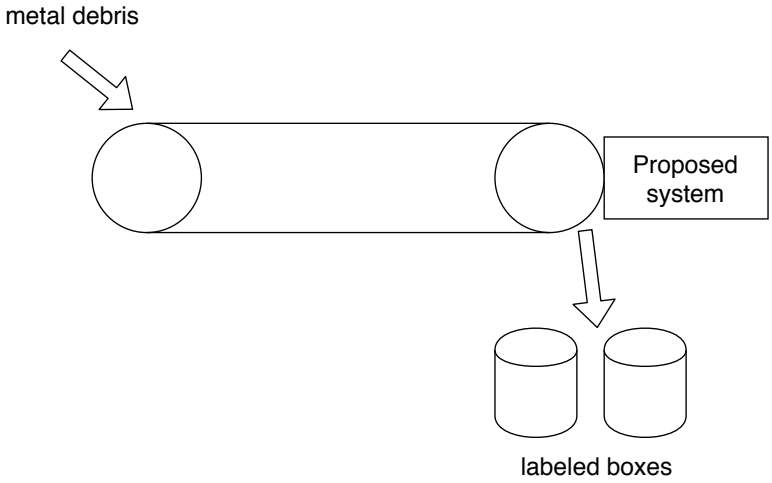

FIGURE 1. Debris classification in assembly line

Our proposed system consists of four main modules. In the first module, a smart camera captures the objects on the belt. The camera operates in run-length code (RLC) mode. From the captured image, the regions of interest (ROI) are extracted in the second module. The extraction phase employs the blob detection algorithm. These extracted boxes are pre-processed in the third module. In the fourth module, the processed boxes are sent to a deep neural network (DNN) to classify the metal debris.

The RLC algorithm is chosen because it is the lossless data compression. In addition, it is not difficult and does not require much central processing unit (CPU) to implement this algorithm. The collected debris images which consist of large dark and bright areas are also very suitable for this RLC algorithm [17]. The goal of the blob detection is to detect a group of connected pixels inside an image [18, 19, 20, 21]. The DNN is employed in our framework to classify the received ROI from blob detection [22]. Four different deep neural

networks models were investigated in our framework. They are four of the most famous neural models used in deep learning. The purpose of this paper is to have a framework to classify the metal debris with high accuracy, a practical application. Thus, the paper does not focus on building better models but rather understand often used model and investigate the operations of these models in our application after the data collection and data pre-processing phases. After that, the most suitable models will be found. In addition, the experiments can also investigate and compare the operation of different deep neural network models in a practical application instead of using conventional academic benchmarks.

The paper is presented as follows. Section 2 presents the proposed framework in detail. The experimental results are shown in Section 3. Section 4 concludes our paper.

2. **Proposed Framework.** The proposed framework has four main modules as can be seen in Figure 2. The details of these modules can be seen in Sections 2.1, 2.2, 2.3, and 2.4.

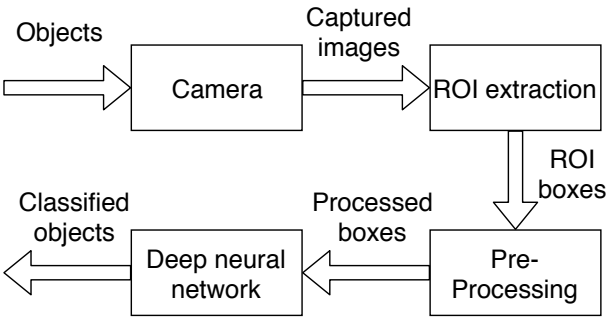

FIGURE 2. Operation of the proposed framework

2.1. **Camera.** In the first module, a camera will capture the object on the conveyor belt. Our system uses the VC-Nano Z camera provided by Vision Components [23]. This camera is selected because it can process high-resolution images in real time while requires low system costs. In addition, the VCLib library of Vision Components can be used to reduce the development time of our system [24].

To increase the operating speed, the captured images employ the run-length code (RLC) processing that is very fast. The run-length encoding (RLE) is the lossless data compression. In RLC, a run of the data is represented as a single data value and its count. For example, if the run of data is "LLLLLHHHLL", the RLC results are "5L3H2L". The RLC algorithm could be not difficult to implement and does not require much CPU power. The using of RLE will be suitable for the files which contain a lot of runs [17].

The RLC in image processing is based on the original RLC. The RLC can be obtained by evaluating the pixel side by side from the left to the right, from the top to the bottom. Each position of the pixel which has the value changed from dark to bright or bright to dark is stored. The first entry of each line has value -1 if the pixel is the bright pixel. On the other hand, the first entry of the line has value 0 if the first pixel in the line is the dark pixel. The white pixel is the pixel having value 1 while the dark pixel is the pixel having value 0. A special character "dx" is inserted at the end of each line.

If the image is represented as Figure 3, the RLC in line 0 is "0 1 3 dx", the RLC in line 1 is "0 2 dx". The RLC in line 2 is "-1 1 3 dx". The RLC in line 3 is "-1 3 dx". The final RLC data will be "0 1 3 dx 0 2 dx -1 1 3 dx -1 3 dx". This mechanism could increase the image processing speed.

The captured images from the VC-Nano Z camera will be converted into the grayscale images. These grayscale images are also transferred to RLC image using $rlcmk()$ function

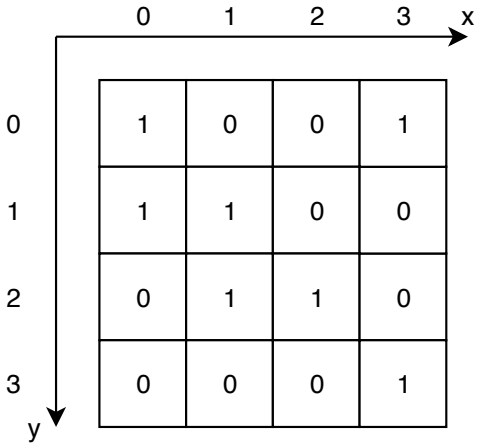

FIGURE 3. RLC operation in image processing

provided by Vision Components. In this function, the gray level g of a pixel is compared with a threshold value t ($0 \leq t \leq 256$). If ($g \geq t$) then the pixel is considered as a bright pixel. On the other hand, if ($g < t$), the pixel will be set as the dark pixel [24].

The obtained RLC images will be sent to the ROI extraction module.

2.2. **ROI extraction.** The goal of this phase is to locate the region of interests (ROI) boxes that contain parts of the metal objects. A famous method called blob detection is used. The name "blob" comes from "binary large object" which represents a set of connected pixels in a binary image [25, 26].

The blob detection algorithm is used to extract the blob in the image. The extracted blob is a searching object. Thus, the algorithm can separate the object from the background. The blob detection algorithm is also called connected component labeling. There are two main scan methods of the connected component labeling called recursive scan and sequential scan. The details of these methods can be seen in the previous studies [18, 19, 20, 21].

The idea of blob detection is to find connected binary pixels. In this situation, the neighbor pixels that have a similar gray level (bright or dark) will be selected into a cluster. Each cluster is assigned a unique number called object number. Finishing the processing, the foreground and the background of the image could be separated.

In our system, the blob detection is conducted on the obtained RLC images to locate the positions of the metal debris. The dark connected region inside the image is the ROI. The center of the dark area is considered as the center of the ROI. The center of each ROI box is considered as the center of one metal debris. The boxes which contain ROI of the images will be extracted from the image. The size of the boxes is 127×127 and 124×124 because our deep neural network uses models that have 127 inputs and 124 nodes. The blob detection of the RLC image in our program uses the API called $rlc\_label()$ provided by Vision Component [24].

2.3. **Pre-processing.** In the second modules, the ROI boxes are extracted. The mean gray level value of each detected ROI is also calculated.

All calculated mean gray level values are shifted to the value of 128 so that all the ROI boxes have a similar mean gray level value of 128. Because the ROI boxes have different intensity concerning the gray level, it is necessary to standardize the mean gray level value. The scale in each ROI box is calculated based on Equation (1).

$$scale\_value = \frac{original\_mean\_value}{128} \tag{1}$$

where $scale\_value$ is the scale value of one image, and $original\_mean\_value$ is the original mean gray level value of this image.

From the $scale\_value$, the new gray level values of all pixels are evaluated according to Equation (2).

$$new\_pixel\_value = \frac{original\_pixel\_value}{scale\_value} \tag{2}$$

As presented, the sizes of the box of ROI are 127×127 and 124×124, while the size of the metal debris could be smaller. In this situation, the background of the boxes must be removed. Thus, the pixels correspond to the detected metal debris are conserved. On the other hand, the background pixels are assigned to the value of zero in each box.

The processed boxes are sent to a deep neural network.

2.4. **Deep neural network.** The fourth module employs the deep neural network (DNN) that is the representation of a human brain [22]. The DNN reduced the required parameters from the normal fully-connected layer NN using local receptive fields, shared weights, and pooling layers. The local receptive field contains several nodes in the hidden layers. The inputs only connect to this field instead of whole hidden layers. The local receptive field is also called the filter. The size of the local receptive field is the filter size. The filter is shifted to connect to other inputs. The stride determines how the filter is moved. The pooling layer reduced the output size concerning weights and biases before this output becomes the input in the next layer. The details of the DNN can be read from the previous papers [22, 27].

In our research, the DNN is used to classify the obtained image-boxed from the prepossessing module. The metal objects in the boxes will be separated into different classes such as copper, aluminum, and brass. Four different architectures of the NNs are investigated in this paper which are the AlexNet, the GoogleNet, VGGNet, and ResNet.

2.4.1. *AlexNet.* The AlexNet model which is proposed by Krizhevsky et al. has eight layers. First five layers are convolutional layers, and three remaining layers are fully-connected layers. The AlexNet uses 11×11, 5×5, and 3×3 convolutions [28].

The details of the AlexNet model are shown in Figure 4. The first five from $conv1$ to $conv5$ are convolutional layers. The first layer uses a filter size of 11×11, the second layer uses a 5×5 filter, while other layers have 3×3 filters. The second parameters after the filter size in each convolutional layer are the number of outputs. For example, the $conv1$ has 96 outputs. The final parameter is the value of stride which defines the movement of the filter.

Some of the convolutional layers are followed by a max pooling layer. Three last layers named from "FC6" to "FC8" are the fully-connected layers. The number after the name of each fully-connected layer is the number of outputs. For example, the FC6 and FC7 have 4096 outputs while FC8 has the outputs depending on the problems. The output of FC8 is also the output of the DNN which are the number of labeled classes.

2.4.2. *GoogleNet.* In the GoogleNet model, a new module called inception is presented. This module employs different filter sizes from 1×1 to 5. These filters are convolved in the GoogleNet model as can be seen in Figure 5. The GoogleNet has 9 inception modules [29].

The advantage of GoogleNet is that this model can cover not only the big area of images by big filter size but also can process the small area using smaller filter size.

Another advantage of GoogleNet is the using of a small filter that can reduce the number of parameters in the NN concerning the weights and the parameters.

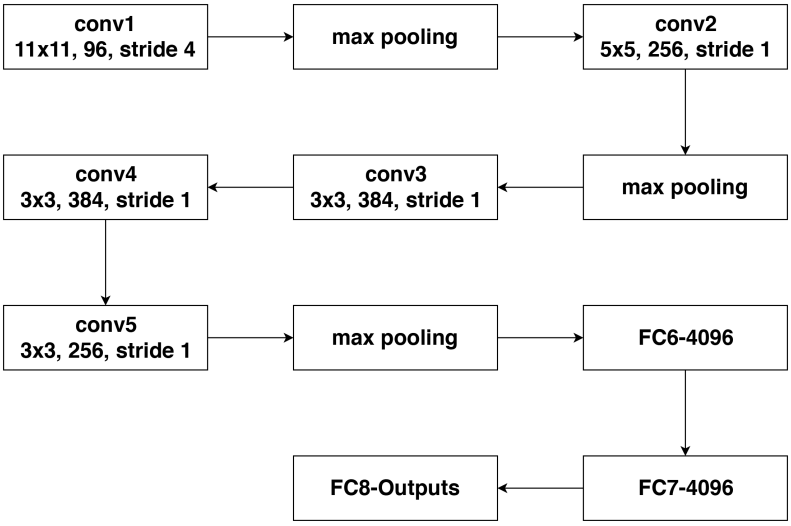

FIGURE 4. AlexNet model

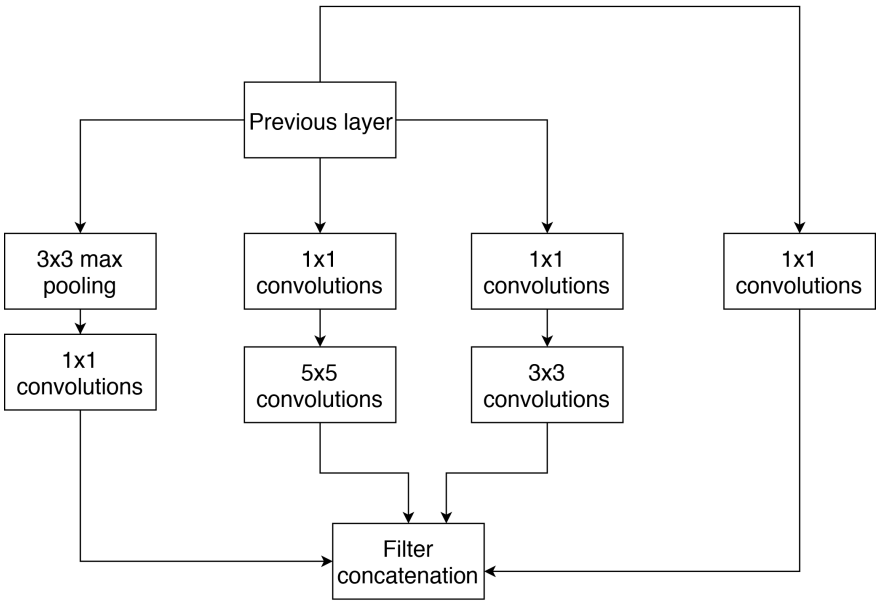

FIGURE 5. GoogleNet model

2.4.3. *VGGNet model.* The VGGNet is proposed by the VGG group, Oxford. This architecture uses only 3×3 convolutions as the filters. However, a lot of 3×3 kernel-sized filters are used. These 3×3 kernel-sized filters are put one after another. The details and configurations of VGGNet can be seen in the previous study [30].

This research chooses randomly one configuration of the VGGNet model which is called VGG16 architecture. The details of the VGG16 architecture are shown in Figure 6.

In Figure 6, *conv*3-64 means it is the convolution layer which has 3 kernel size of 3×3 and 64 outputs. The notation "FC6-4096" means this layer is considered as the fully-connected layer and the output of this fully-connected layer is 4096. In this architecture, the outputs of FC8 layers are our desired outputs which correspond to the number of labeled classes.

2.4.4. *ResNet model.* The previous paper has proposed the residual neural network (ResNet) to solve the issue related to the accuracy degradation of the deep neural network.

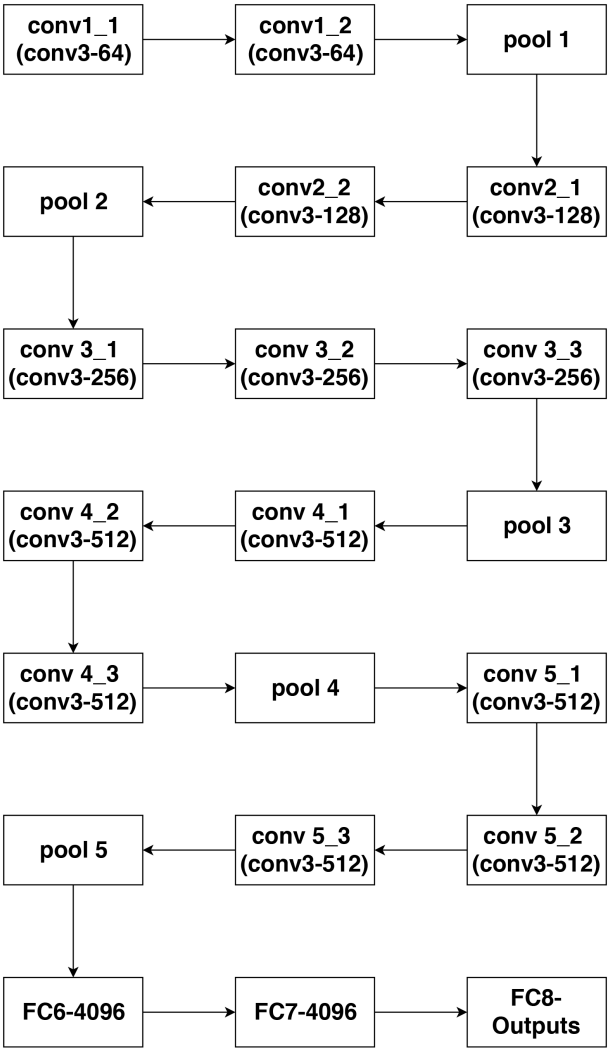

FIGURE 6. VGG16 architecture

The accuracy could be degraded when the neural network has a big depth. The ResNet provides a skipping mechanism to jump the connections over some layer. Examples of skipping can be seen in Figure 7. Previous experimental results showed that the "skip connection" obtained higher accuracy when compared with the conventional deep convolutional neural network [31].

The ResNet has different models. In this research, the 50-layer ResNet (called ResNet50) is chosen randomly as our model. The details of the 50-layer ResNet can be seen in Figure 8.

3. **Experiments.** Our research conducted experiments to classify the metal debris. The data is provided by a company. The confidentiality of the data must be protected using an agreement.

Four different deep neural network models (AlexNet, GoogleNet, VGGNet, ResNet) were investigated in our experiments. The computer used in the experiments was HP Z800 Workstation. This computer was equipped with dual Intel Xenon X5675 processor, and Geforce GTX 1080 Ti graphics card.

We conducted two different experiments. The first experiment investigated the operations of the proposed framework with three different classes called brass, copper, and other metals. The second experiment tested with two metal classes which were aluminum

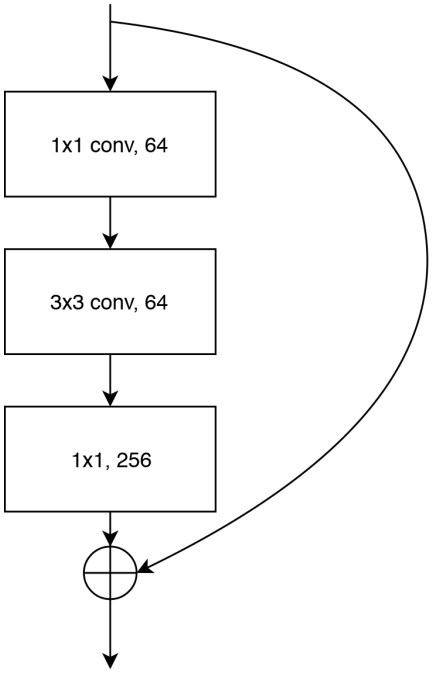

FIGURE 7. Operation of one layer in ResNet

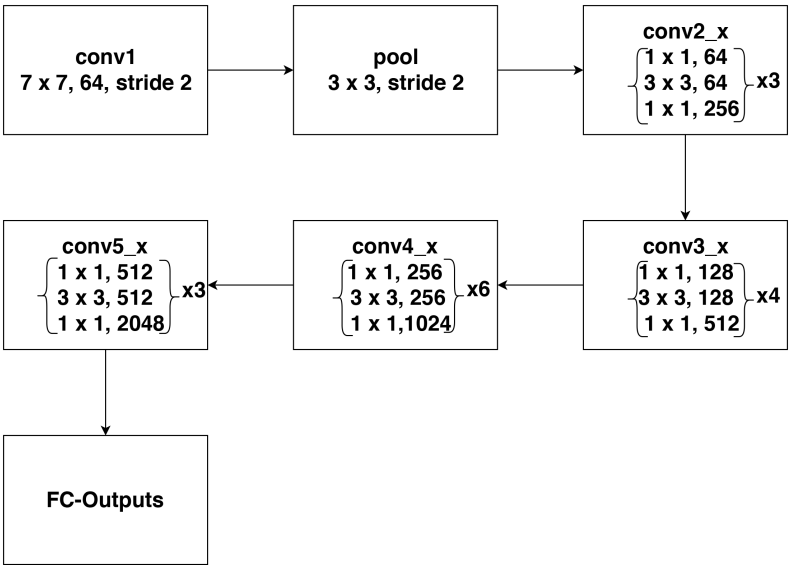

FIGURE 8. 50-layer ResNet

and copper. Hence, in the first experiment, the neural network had three outputs while the neural network had two outputs in the second experiment. Both experiments were investigated with four DNN models.

3.1. **First experiment.** The first experiment was used to classify three different classes which are brass, copper, and other metals. Thus, the DNN had three outputs. Our data had 5327 training images and 2281 testing images.

The experiments were investigated with all four DNN models (AlexNet, GoogleNet, VGGNet, and ResNet). In each model, the number of training iterations was increased from 1 to 50000. The iteration that had the highest number of recognition rate of each

model was stored. To use with the models, the images before the AlexNet were cropped to the size of 127×127. On the other hand, the images before GoogleNet, VGG, and ResNet were cropped to the size of 124×124 based on previous studies. Other configurations of the DNN were also similar to previous papers [28, 29, 30, 31].

Table 1 presents the experimental results with three classes called brass, copper, and other metals, respectively. The highest recognition rates for each class and all classes are shown in this table. Among the three classes, the copper was the most difficult to recognize for the system. The other metals class had the highest recognition rates in all models when all models had higher than 98% of accuracy. Experimental results demonstrated that our framework can separate the metal debris. All four models had higher than 85% of recognition rate. Especially, the AlexNet model, the GoogleNet model, and the ResNet had more than 90% of the recognition rate. Among these three models, the AlexNet obtained the highest accuracy with 97.15% of correct classification. Even with the copper samples while VGGNet and ResNet had lower than 80% of the recognition rate (64.32%, and 79.16% respectively), the AlexNet could achieve 96.04% of correct recognition. The VGGNet had a low recognition rate concerning the copper class at 64.32% while it had a high recognition ate regarding the other metal class at 99.50%.

TABLE 1. Results with three classes

	AlexNet	GoogleNet	VGGNet	ResNet
Iteration	25000	20000	42000	30000
Brass correct	668/691 (96.67%)	652/691 (94.36%)	644/691 (93.20%)	661/691 (95.66%)
Copper correct	751/782 (96.04%)	741/782 (94.76%)	503/782 (64.32%)	619/782 (79.16%)
Other correct	796/808 (98.64%)	804/808 (99.50%)	796/808 (98.51%)	803/808 (99.38%)
All correct	2216/2281 (97.15%)	2197/2281 (96.17%)	1943/2281 (85.18%)	2083/2281 (91.32%)

3.2. **Second experiment.** The second experiment was conducted to investigate different configurations of the DNN. In this case, the DNN was tested with data that had two classes called aluminum and copper, respectively. Therefore, the DNN had two outputs. Other configurations of the DNN were similar to the first experiment.

The data of this experiment had 3338 images for training and 1431 images for testing. These images were also investigated with four DNN models called AlexNet, GoogleNet, VGGNet, and ResNet, respectively.

For all models, the number of training iterations was modified from a very small number of 1 iteration to a high number of 50000. The iterations that had the highest recognition rate of each DNN model were kept.

The results of this experiment are presented in Table 2. Experimental results once again confirmed that the proposed framework can be used to classify the metal debris while all four models had more than 98% of recognition rate in all classes. Similar to the previous experiment, the AlexNet also obtained the highest accuracy with 99.93%. And the GoogleNet had a better recognition rate than the VGGNet and the ResNet. On the other hand, the ResNet had the lowest recognition rate (only 98.46%) in this experiment. This experiment had two scenarios of 100% accuracy when the AlexNet and the ResNet were used to classify the aluminum metal. The AlexNet had also best accuracy in each class with 99.89% and 100% of the copper class and the aluminum class, respectively.

TABLE 2. Results with two classes

	AlexNet	GoogleNet	VGGNet	ResNet
Iteration	45000	50000	22000	45000
Aluminum correct	528/528 (100%)	514/528 (94.36%)	513/528 (97.16%)	528/528 (100%)
Copper correct	902/903 (99.89%)	902/903 (97.35%)	900/903 (99.67%)	881/903 (97.56%)
All correct	1430/1431 (99.93%)	1416/1431 (98.96%)	1413/1431 (98.74%)	1409/1431 (98.46%)

Compared to the first experiment, the second experiment with two classes obtained a higher accuracy. It can be explained that the first experiment had "other metals class" which may be mistakenly classified as "brass" class or "copper" class. In both experiments, the AlexNet model obtained the highest percentage number of correct recognition at 97.15% and 99.93%, respectively. Thus, it can be considered that the AlexNet model is the best suitable model in our classification framework. The results can be explained that in our framework, the AlexNet had only 8 deep layers while the VGGNet, the GoogleNet, the ResNet had 16 deep layers, 22 deep layers, 50 deep layers, respectively. On the other hand, the deeper the neural network is, the easier it is to be overfitting. The deeper neural network is not usually better. With several data, a shallow neural network may obtain a high recognition rate while a deeper neural network can get a low recognition rate. This reason could be used to demonstrate the performance of the AlexNet model with our data. With other data, the deeper neural network may be better in classification but it has to tweak the neural network parameters quite a lot. In addition, it is also necessary to redesign the neural network to be suitable with the data to have a high number of correct classification rate. It will require much time and effort to redesign the neural network to increase the recognition rate. Therefore, the AlexNet model becomes a suitable model for the metal debris classification framework. Another advantage of the shallower neural network, such as AlexNet model, is that it is also more feasible than a deeper neural network model, such as ResNet, VGGNet or GoogleNet, to implement on field-programmable gate array (FPGA) devices because FPGA device has limited number of resourcing concerning the logic elements and memory bits.

4. **Conclusions.** This paper proposed a framework to classify the metal debris in order to reduce the time and effort in the recycling phase. Our framework contains four different modules. The first module had the camera with run-length code processing called VC-Nano Z camera. The second module is used for the ROI extraction of the obtained image from the camera. The obtained extraction is processed in the pre-processing module before sending to the DNN module.

Experimental results demonstrated that the proposed framework can classify the metal debris. Four different models of the DNN were also investigated in the DNN module called the AlexNet, the GoogleNet, the VGGNet, and the ResNet. The results also showed that the AlexNet model was more suitable among the four models because it obtained the highest recognition rate in both experiments.

In our future research, we will optimize the proposed framework to increase the recognition rate. More DNN models will also be tested to investigate the most suitable models for our framework.

REFERENCES

[1] K. Barbakadze, W. Brostow, G. Granowski, N. Hnatchuk, S. Lohse and A. T. Osmanson, Separation of metal and plastic wastes from wire and cable manufacturing for effective recycling, *Resources, Conservation and Recycling*, vol.139, pp.251-258, 2018.

[2] B.-G. Cho, Y.-J. Cho, J.-chun Lee and K. Yoo, Korea's metal resources recycling research project – Valuable recycling, *Geosystem Engineering*, vol.22, no.1, pp.48-58, 2019.

[3] B. Ramalingam, A. K. Lakshmanan, M. Ilyas, A. V. Le and M. R. Elara, Cascaded machine-learning technique for debris classification in floor-cleaning robot application, *Applied Sciences*, vol.8, no.12, 2018.

[4] I. Szatmari, A. Schultz, C. Rekeczky, T. Kozek, T. Roska and L. O. Chua, Morphology and autowave metric on CNN applied to bubble-debris classification, *IEEE Trans. Neural Networks*, vol.11, pp.1385-1393, 2000.

[5] G. Saravana Kannan, S. Sasi Kumar, R. Ragavan and M. Balakrishnan, Automatic garbage separation robot using image processing technique, *International Journal of Scientific and Research Publications*, vol.6, no.4, pp.326-328, 2016.

[6] I. R. Novelle, J. P. Cid and A. Salmador, Intelligent garbage classifier, *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.1, no.1, pp.31-36, 2008.

[7] C. R. Djaya, N. Sucianti, L. A. Wulandhari et al., Hybrid particle swarm optimization and back-propagation neural network for organic and inorganic waste recognition, *Computer Science On-line Conference*, pp.168-177, 2017.

[8] N. K. Myshkin, H. Kong, A. Ya. Grigoriev and E.-S. Yoon, The use of color in wear debris analysis, *Wear*, vol.251, no.1, pp.1218-1226, 2001.

[9] M. Kutila, J. Viitanen and A. Vattulainen, Scrap metal sorting with colour vision and inductive sensor array, *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)*, pp.725-729, 2005.

[10] E. J. Sommer Jr, C. E. Roos, D. B. Spencer and R. L. Conley, *Method and Apparatus for Sorting Materials according to Relative Composition*, US Patent 7,564,943, 2009.

[11] D. A. Wahab, A. Hussain, E. Scavino, M. Mustafa and H. Basri, Development of a prototype automated sorting system for plastic recycling, *American Journal of Applied Sciences*, vol.3, no.7, pp.1924-1928, 2006.

[12] M. Wang, Z. Peng, K. Vasilev and N. Ketheesan, Investigation of wear particles generated in human knee joints using atomic force microscopy, *Tribology Letters*, vol.51, pp.161-170, 2013.

[13] W. Hong, W. Cai, S. Wang and M. M. Tomovic, Mechanical wear debris feature, detection, and diagnosis: A review, *Chinese Journal of Aeronautics*, vol.31, no.5, pp.867-882, 2018.

[14] M. D. O'Toole, N. Karimian and A. J. Peyton, Classification of nonferrous metals using magnetic induction spectroscopy, *IEEE Trans. Industrial Informatics*, vol.14, pp.3477-3485, 2018.

[15] A. Iuga, R. Morar, A. Samuila, I. Cuglesan, M. Mihailescu and L. Dascalescu, Electrostatic separation of brass from industrial wastes, *IEEE Trans. Industry Applications*, vol.35, pp.537-542, 1999.

[16] S. Shin, Y. Moon, J. Lee, H. Jang, E. Hwang and S. Jeong, Signal processing for real-time identification of similar metals by laser-induced breakdown spectroscopy, *Plasma Science and Technology*, vol.21, 2018.

[17] S. Golomb, Run-length encodings (corresp.), *IEEE Trans. Information Theory*, vol.12, no.3, pp.399-401, 1966.

[18] R. Acevedo-Avila, M. Gonzalez-Mendoza and A. Garcia-Garcia, A linked list-based algorithm for blob detection on embedded vision-based sensors, *Sensors*, vol.16, no.6, p.782, 2016.

[19] B. R. Kiran, K. R. Ramakrishnan, Y. S. Kumar and K. P. Anoop, An improved connected component labeling by recursive label propagation, *National Conference on Communications (NCC)*, pp.1-5, 2011.

[20] M. B. Dillencourt, H. Samet and M. Tamminen, A general approach to connected-component labeling for arbitrary image representations, *Journal of the ACM (JACM)*, vol.39, no.2, pp.253-280, 1992.

[21] L. G. Shapiro, Connected component labeling and adjacency graph construction, in *Topological Algorithms for Digital Image Processing*, T. Y. Kong and A. Rosenfeld (eds.), vol.19 of *Machine Intelligence and Pattern Recognition*, North-Holland, 1996.

[22] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks*, vol.61, pp.85-117, 2015.
[23] V. Components, *VC nano Z Series Operating Manual*, https://www.vision-components.com/fileadmin/external/documentation/hardware/VC_nano_Z/index.html, 2018.
[24] V. Components, *VCLib Documentation 6.5.0.*, https://www.vision-components.com/fileadmin/external/documentation/software/lib/libvc-base/latest/html/index.html, 2017.
[25] R. Burns and I. Narang, Version management and recoverability for large object data, *Proc. of International Workshop on Multi-Media Database Management Systems*, pp.12-19, 1998.
[26] A. G. Ghuneim, *Contour Tracking*, http://www.imageprocessingplace.com, accessed on 30-April-2019.
[27] Y. Bengio, Learning deep architectures for AI, *Foundations and Trends® in Machine Learning*, vol.2, no.1, pp.1-127, 2009.
[28] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger (eds.), Curran Associates, Inc., 2012.
[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, Going deeper with convolutions, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
[30] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *CoRR*, arXiv:1409.1556, 2014.
[31] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.