2018 IEEE 16th Int. Conf. on Dependable, Autonomic & Secure Comp., 16th Int. Conf. on Pervasive Intelligence & Comp., 4th Int. Conf. on Big Data Intelligence & Comp., and 3rd Cyber Sci. & Tech. Cong.

Product Surface Defect Detection Based On Deep Learning

Lien Po Chun, Qiangfu Zhao
Graduate School of Computer Science and Engineering
The University of Aizu
Aizuwakamatsu-city, Fukushima, Japan
{m5212107, qf-zhao}@u-aizu.ac.jp

***Abstract*—Image classification is a branch of computer vision that uses a computer to acquire image data and interpret them by mimicking human biological systems. This is a very important topic in today's situation because every second a large number of image data are acquired and used for various purposes around the world. One application of image classification is to detect defects on the surfaces of industrial products. Quality inspection is usually the final stage in a production line, and has been conducted mainly by human experts. This can be time consuming and mistake-prone. In this study, we investigate the possibility of replacing, fully or partially, human experts with a machine learner when the product defects are visible. In this study, we investigate several methods based on deep learning. The first one is to use a deep learner directly to detect the existence of defects in a given product surface image. The second one segments suspected parts first and then uses the deep learner to classify the segmented parts. The third method employs an ensemble of deep learners. Results show that the third method can provide the best results, and can be practically useful if we introduce a proper rejection mechanism.**

Keywords—Image classification, defect detection, convolutional neural network (CNN), support vector machine (SVM).

I. INTRODUCTION

Product inspection is important for quality control, and it is usually the final stage in a production line. Conventionally, product inspection is mainly conducted by human experts, and this can be time consuming and mistake-prone. For example, a human inspector may overlook certain defective products due to long hours of eye strain or other factors. To reduce the burden on human inspectors, we can use a computer to perform product inspection based on methods proposed in the context of pattern recognition. Patterns may include images, sounds, smells, etc. In this paper, we consider image-based product inspection.

Currently, it is still difficult to replace human experts completely. Our goal is to filter out as many products as possible to reduce human labor. That is, we can use a machine to classify products into easy and difficult ones. For the former, the machine can make very confident decisions. A human is needed to make the final decisions only for difficult products because the machine cannot make trustable decisions.

Product inspection is basically a pattern recognition problem, and many methods proposed in this area can be adopted. There are two "state-of-the-art" methods for pattern recognition. One is support vector machine (SVM) and the other is convolutional neural network (CNN). Theoretically, SVM is the best "shallow learner". It often outperforms other existing methods in the sense of generalization ability [9]. On the other hand, CNN is the best known "deep learner" for image classification [10]. A CNN can extract discriminative features for pattern recognition and can even outperform SVM for recognizing images, provided that a proper structure is pre-defined.

The authors of [2] have studied railway surface inspection based on CNN. Due to rapid development of railway industry, detection of track defects is becoming a critical issue. By inspecting the railways automatically and quickly using a machine, the task can be fulfilled more efficiently and safely. The authors of [2] put orbital images into a fine-tuned CNN and the CNN can extract partial features for classifying and positioning the defects. Since product inspection is similar to railway inspection, CNN is also useful for solving our problem.

CNN has proven to be outstanding on many computer vision and machine learning issues. Among many applications of CNN, image classification might be the most suitable one [10]. One example is individual recognition based on fingerprints. In recent years, fingerprint identification has become one of the most reliable technologies for security control. The authors of [3] used four CNNs for fingerprint liveness detection. The proposed system was evaluated on a data set containing 50,000 real and fake fingerprint images. The best CNN-based model is found to be much better than existing models in terms of recognition rate. In [4], CNN was also successfully employed for medical image classification.

In [5], generic object recognition is carried out in six categories. The categories include human figures, four-legged animals, airplanes, trucks, cars, and "none of the above". The research shows that even though CNNs are competent at learning invariant features, they do not always produce the best results for classification. On the other hand, SVM can produce good decision surfaces when feature vectors are given, but they cannot learn invariant features. Thus, it is necessary to combine CNN and SVM to solve complicate problems. In fact, in [1], CNN and SVM were combined for automatic target recognition at ground level, and the obtained results are better than those obtained by using CNN alone.

Based on the above discussion, in this study, we also consider combining CNN and SVM, and expect to obtain better

978-1-5386-7518-2/18/$31.00 ©2018 IEEE
DOI 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00051

performance for product inspection. Generally speaking, a CNN can emulate the human eyes because it can extract features such as shapes, colors, and textures [7-8]. In addition, CNN-based feature extraction is relatively robust to translation, scaling, tilting, and other deformations. On the other hand, SVM can emulate the brain because it can make the best decision based on features extracted by CNN. Thus, combining CNN and SVM, we may expect to inspect the product more effectively [1] [12].

For the CNN, we adopt the AlexNet [6] model. Alexnet was a CNN model proposed in 2012 by Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever. This model was the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) contest. The rationale for us to select Alexnet is that this model can be re-used for solving different image recognition problems through transfer learning because key features for recognizing various images have already obtained in the lower layers.

The rest of this article is organized as follows. In Section 2, we briefly introduced SVM, CNN, and AlexNet. In Section 3, we introduce three different ways for product inspection, and verify their efficacies via experiment. Section 4 introduces a rejection mechanism to reduce the number of products for manual inspection, and confirms its efficacy via experiment. The last section draws some conclusions and introduces some topics for future study.

II. Priliminaries

To make this paper relatively self-contained, we provide a brief introduction to SVM, CNN, and AlexNet in this section. For more detail, readers may refer to the references.

A. Support Vector Machine

SVM was originally proposed for supervised learning, although it can also be used for semi-supervised learning, with some extensions. SVM is useful both for classification and regression. To obtain an SVM, we need a set of training instances. Each instance has a label belonging to one of the two categories. The SVM training algorithm creates a model that assigns new instances to one of the two categories.

In fact, an SVM model can also be considered a single hidden layer multilayer perceptron (MLP) and all support vectors together form the hidden layer. Using a proper kernel function, the hidden layer can map the patterns from the original feature space into another space, in which the problem becomes a linear one, and can be solved by using a linear classifier. This is not the only advantage of using SVM. In fact, the training algorithm can find a decision boundary that possesses the maximum margin. Intuitively speaking, the maximum margin means that the decision boundary has (roughly) equal distances to both positive data and negative ones, so that correct decisions can be made even if the observed new datum may contain some unpredicted noises. Maximum margin is the main reason why an SVM usually outperforms other methods for many problems.

To use an SMV, however, we must extract proper features first from the raw data. In our study, the raw data are surface images of the products. In product inspection, we can see or hear the defects, but in many cases, we do not know why. Thus, extraction of useful features for product inspection can be difficult. In this study, we use CNN for feature extraction.

B. Convolutional Neural Network

A CNN is a special MLP suitable for image classification. The network is invariant to translation, scaling, tilting, and other forms of deformation. A CNN usually consists of several convolutional layer and pooling layer pairs, followed by one or more full connected layers (Fig. 1). The main function of a convolutional layer is to detect special features via filtering. The main function of a pooling layer is to compress the results obtained by the previous convolutional layer via sub-sampling. In fact, in a CNN, convolution is used as another name of digital filters, although this may not be proper in a point of view of signal or image processing.

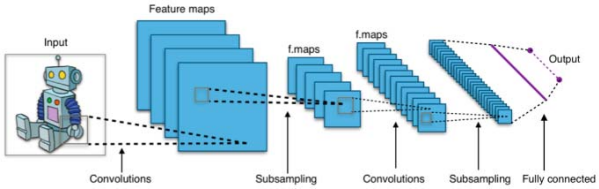

Fig. 1. Structure of a CNN (taken from *https://en.wikipedia.org/wiki /Convolutional_neural_network*)

The earliest CNN, called LeNet, was proposed by Yann LeCun in 1998 [11]. LeNet consists of three convolutional layers, two pooling layers, and a fully connected layer. Later CNN models are generalizations of the LeNet. In a CNN, a convolutional layer consists of many convolutional kernels. Each convolutional kernel translates a given 2-D plane to another 2-D plane using the weighted-sum operation. A convolutional kernel is actually a 2-D finite response (FIR) filter. If we use infinite impulse response (IIR) filter, or realize the FIR filter using fast Fourier transform, a CNN can be implemented more efficiently.

The pooling layer in a CNN reduces the possibility of overfitting by reducing the number of parameters via sub-sampling. Two methods are often used for sub-sampling. The first one is max-pooling which uses the maximum value in a pooling window to replace the region covered by the window. The second one is mean-pooling which uses the average value in the pooling window to replace the region covered by the window.

The fully connected layer is the same as the output layer of a conventional MLP. Every output neuron connects all neurons of the previous layer (e.g. the last pooling layer).

C. The AlexNet

AlexNet is a special CNN which was the winner in the ILSVRC contest in 2012. The structure of an AlexNet is shown in Fig. 2. The net contains eight layers. The first five layers are convolutional and polling pairs. In each convolutional layer, the rectified linear unit (ReLU) activation function and the local response normalization (LRN) process are included. The remaining three layers are fully-connected and the output of the last layer is fed to a 1,000-way softmax activation function

which produces a distribution over the 1,000 class labels. The goal of the network is to maximize the average multinomial logistic regression, which is equivalent to maximizing the average across training cases of the log-probability of the correct label under the prediction distribution.

Using an AlexNet, an image is recognized as follows:

- Input the image (the size is re-sized to 227 × 227 × 3).
- Extract the features using 96 filters (or convolutional kernels) with the size of 11 × 11 × 3, and a stride of 4 pixels (a stride here is the distance between the receptive field centers of neighboring neurons in a kernel map).
- Use ReLU to ensure that the values of the feature are within a reasonable range.
- Compress the filter results using max-pooling.
- Normalize the local regions using LRN. The kernel size of the model is 3 × 3, which means the process is to deal with 3 × 3 regions.
- The process of the rest layers is similar to the first layer. The second layer takes as input the (normalized, pooled) output of the first layer and filters it with 256 kernels of size 5 × 5 × 48. The third, fourth, and fifth layers are connected to one another without any intervening pooling or normalization. The third layer has 384 kernels of size 3 × 3 × 256 connected to the normalized and pooled outputs of the second layer. The fourth layer has 384 kernels of size 3 × 3 × 192, and the fifth layer has 256 kernels of size 3 × 3 × 192. The fully-connected layers have 4,096 neurons each.

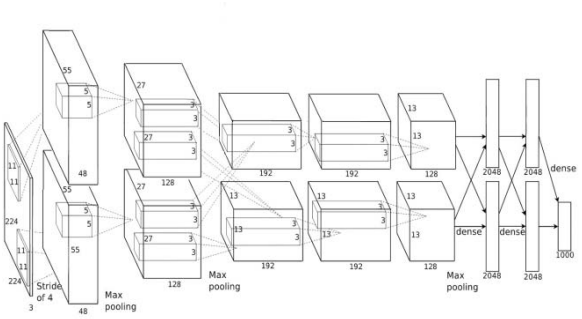

Fig. 2. Structure of an AlexNet

III. Methods for Product Inspection

Pattern recognition techniques can be categorized into two groups, namely unsupervised and supervised. In the case a product is produced massively every day, in other words, if there are a huge amount of un-labelled data, we may consider using some unsupervised method first to classify the product into several meaningful clusters, and then assign a label to each cluster manually. We can then design a model for on-line application using some supervised method.

As the first step, we just use labelled images for training. The main purpose is to see if a deep learner can achieve a performance similar to human experts. Fig. 3 and Fig. 4 show some product images. Products shown in Fig. 3 are defect-like, but normal. Products shown in Fig. 4 are true defects or abnormal ones. Our goal is to build a deep learner based on a training set containing both normal and abnormal images. In the following, we introduce three methods. In all methods we suppose that the deep learner is a combination of the AlexNet and an SVM.

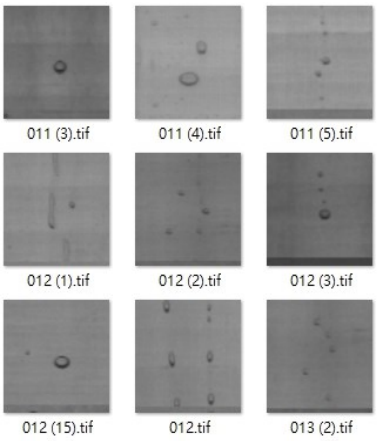

Fig. 3. Normal product images

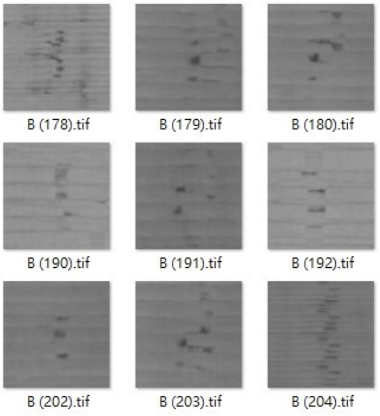

Fig. 4. Abnormal product images

A. Method I: Use the images directly

The first method is to use the given image data directly. That is, features for image recognition are extracted automatically by the AlexNet, and the SVM can be used to make the final decisions. In our model, we actually replaced the last layer of the AlexNet with a new layer, re-train the network, and then replace the last layer again with the SVM. In other word, we are using transfer learning to apply the AlexNet to solving a new problem.

In the experiments, we actually used the Statistics and Machine Learning Toolkit of Matlab to train the model. In Matlab, AlexNet is implemented by a 25-layer structure. The 22-th layer corresponds to the 7-th layer of the original AlexNet. We combine the first 22 layers with 3 new layers for transfer learning, and then use the output of the 20-th layer as feature to training an SVM. Fig. 5 shows an example of the training curve,

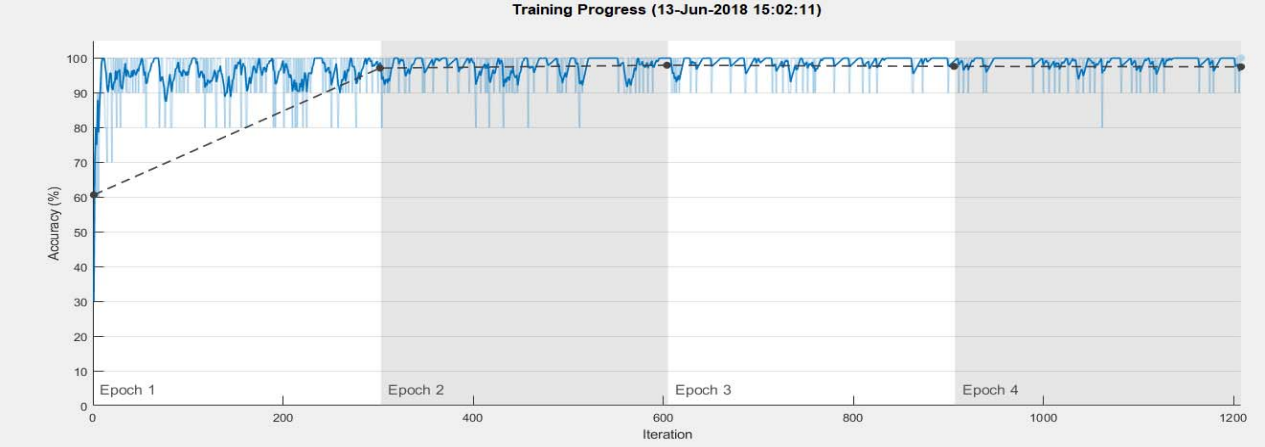

Fig. 5. An example of training curve for transfer learning

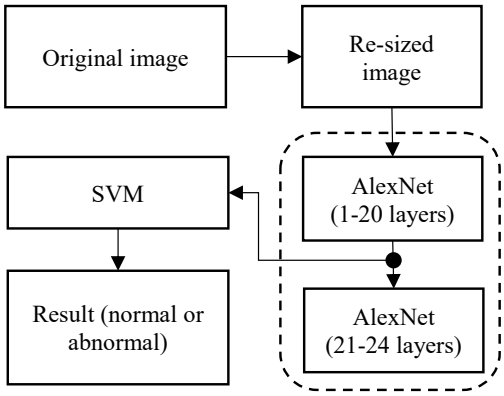

Fig. 6. Block diagram of Method I

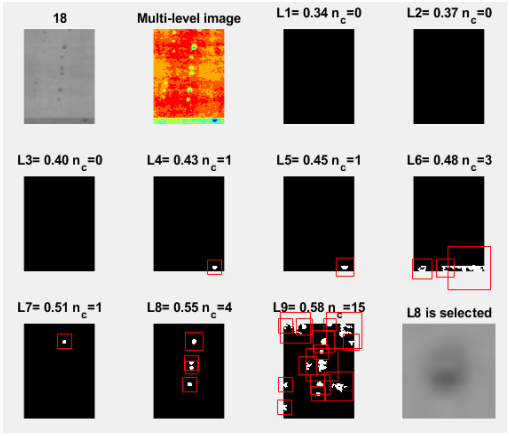

Fig. 7. An example of image segmentation

and Fig. 6 is the block diagram of decision making based on Method I.

B. Method II: Classification Based on Segmented Images

In Method II, we first convert the original gray image into a multi-level one using the method proposed by Otsu [13], and then find a proper threshold T_0 to binary the original image. There are several factors for determining T_0. For example,

1) If the number of connected regions is larger or less than a given threshold T_1.
2) If the number of connected regions changes significantly from level i to level $i+1$.

The thresholds are specified by us based on our observations (concrete parameter values cannot be provided here because of request from our partner company). An example of image segmentation is shown in Fig. 7.

With the binarized image, we can segment the connected regions as sub-images, and use them for training the deep learner. The deep learner model itself is the same as that used for Method I. To see if a given image is normal or abnormal, we input all sub-images segmented from this image to the deep learner, one-by-one. If the sum of all results is above a given threshold, the whole image is considered abnormal (note that the output of the SVM is 1 for abnormal and 0 for normal). The block diagram of decision making based on Method II is given in Fig. 8.

C. Method III: Classification Based on Neural Network Ensemble

In Method III, we use the original images as we do in Method I. The difference is that we use several models to form a deep learner ensemble, with the expectation to improve the accuracy. To design a good ensemble, it is necessary to make all members as different as possible. That is, we need to train deep learners with different behaviors, so that to get better performance when they are put together. Although many methods have been proposed in the literature, in this study we adopt the bagging method [14], which is simple and easy to use.

The basic idea of bagging is to design several weak learners based on training sets randomly selected from the original

training set. In this study, we obtain 7 (an odd number) training sets by extracting $0.7 \times N_t$ data randomly from the original training set, where N_t is the total number of training data, and train 7 deep learners. For a given image, we first find the outputs of all learners, and then make the final decision based on majority voting. Fig. 9 illustrates the basic flow of Method III for decision making.

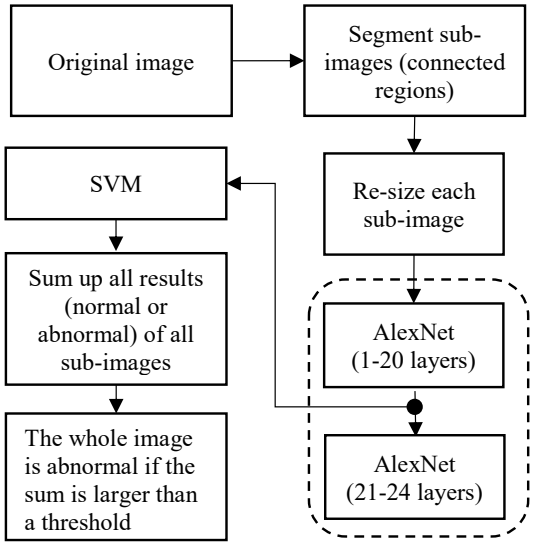

Fig. 8. Block diagram of Method II

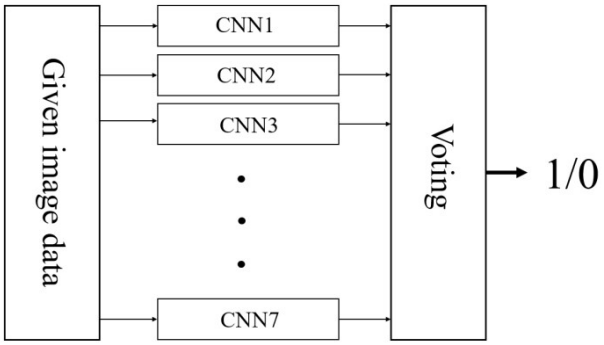

Fig. 9. Flow of decision making using Method III

D. Experimental results

To validate the three methods discussed above, we conducted experiments with the given data set. There are altogether 6,283 data. We ran the programs 10 times. Each time we selected 70% of the data randomly for training and the other 30% for testing. In the experiments, we used the Statistics and Machine Learning Toolkit of Matlab, and all parameters took default values. Table 1 shows the results. From these results we can see that Method III is a little bit better compared with the other two methods.

Next, we used another completely new dataset provided separately by the company to test the performance of the three methods. This set contains 20,481 labelled images and was reserved for testing only. Through training based on the previous 6,283 data, we obtained a model using one of the methods, and then tested the model using the 20,481 new images. Tables 2-4 show the confusion matrices obtained by the three methods. The recognition rates of the three methods are 93.65%, 94.29%, and 95.26%, respectively. From these results, again, we can see that Method III is the best one.

Table 1: The recognition rates for test data (in percentage)

Run	Method I	Method II	Method III
1	97.51	97.79	98.68
2	98.02	97.97	98.24
3	98.29	97.73	98.25
4	98.29	98.91	98.07
5	98.62	98.63	98.51
6	98.18	98.15	97.84
7	98.79	98.30	98.01
8	98.13	98.00	98.29
9	98.46	98.42	98.28
10	97.25	98.39	98.31
Avg.	98.154	98.229	**98.248**
Std.	0.47181	0.37409	**0.23962**

Table 2: Confusion matrix of Method I

		Predicted	
		Normal	Abnormal
True	Normal	7875	972
	Abnormal	**329**	11305

Table 3: Confusion matrix of Method II

		Predicted	
		Normal	Abnormal
True	Normal	7885	962
	Abnormal	**207**	11427

Table 4: Confusion matrix of Method III

		Predicted	
		Normal	Abnormal
True	Normal	8012	835
	Abnormal	**135**	11499

Table 5: Confusion matrix of Method III with rejection

		Predicted	
		Normal	Abnormal
True	Normal	6727	83
	Abnormal	52	10053

IV. An Improved Method for Real Application

In practice, however, the methods discussed in the previous section are not enough for product inspection. In fact, from Table 4 we can see that even if the recognition rate is high, there are many false positive and false negative errors. For any given

image, we actually do not know whether the decision provided by the deep learner is really correct or not. That is, a human expert must re-examine all images to make the final decisions.

To reduce the labor of human inspectors, it is necessary to make decisions automatically for most products but leave a small number of "difficult ones" to human inspectors for further confirmation. For this purpose, we need to define a "confidence degree" for each result. If the confidence degree is not high enough, we can reject it, and ask the human expert to confirm; otherwise, we can trust the decision made by the machine.

Using an SVM, the confidence degree can be measured intuitively by the distance between the given image (which is a point in the feature space) and the decision boundary. The closer the image to the decision boundary, the lower the confidence degree is. Fortunately, the SVM obtained using Matlab toolbox also provides parameters for determining the confidence degree. Along with the final decision (-1 or 1), we also obtain a loss matrix for each input data. If the loss for outputting a 1 (or -1) is zero of smaller than a certain threshold, we can trust the result provided by the SVM. However, if the loss for outputting a 1 is close to that for outputting a -1, we can reject the input image, and leave it for manual re-check.

In Method III, we can define two thresholds for rejection based on the test sets. Remember that we design 7 models using 70% of the data randomly selected from the given data set. Each time, we have 30% of the data left, and they can be used to determine the thresholds. Roughly speaking, if the loss for outputting a 1 (or -1) is larger than a threshold, we reject that input image. This threshold is so defined to make the false negative error (an error that classifies an abnormal pattern into normal pattern) zero for the test sets.

Table 5 is the confusion matrix of the Method III after introducing the rejection mechanism, for the reserved testing set. Among 20,481 data, we rejected 3,566, and the accuracy for other data is 99.201%. The percentage of rejected data is 17.41%. These data must be re-examined by human experts. For not rejected data, however, there are still 52 false negative errors. These errors are produced mainly because the original training data are not pure. That is, some training images may not have correct labels. As for the 52 misclassified images, some of them cannot be classified correctly even by human.

V. Conclusion

In this article, we have used the AlexNet to extract useful features and the support vector machine (SVM) to make the final decision. Experimental results show that a deep learner ensemble can classify the images very accurately. By introducing a rejection mechanism, it is also possible to reduce human labor.

In the next step, we need to ask the human experts to re-examine the image data, so that the deep learner can be trained using correctly labelled data. Another method to improve the performance is to introduce an "outlier" detector. We can remove the outliers for training, and reject the outliers directly for testing. In fact, SVM can be a good model for detecting outliers [15].

In addition, we will try other deep learning models (e.g. GoogleNet) and find the best classification method by adjusting the parameters. In the future experiments, we will continue to increase our training data and filter out incorrect training data.

Acknowledgement

The authors would like to thank our partner company for providing the labelled data, and the anonymous reviewers for their invaluable comments for improving this article.

References

[1] S. Wagner, Combination of convolutional feature extraction and support vector machines for radar ATR, pp. 1-6, 2014.

[2] Lidan Shang, Qiushi Yang, Jianing Wang, Shubin Li, Weimin Lei, " Detection of rail surface defects based on CNN image recognition and classification", pp. 45-51, 2018.

[3] R. F. Nogueira, R. de Alencar Lotufo, R. C. Machado, "Fingerprint Liveness Detection using Convolutional Networks", IEEE Trans. Inf. Forensics Secur., vol. 11, no. 6, pp. 1206-1213, 2016.

[4] L. Lu, H. Shin, H. R. Roth, M. Gao, L. Lu, S. Member, Z. Xu, I. Nogues, J. Yao, D. Mollura, R. M. Summers, "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures Dataset Characteristics and Transfer Learning Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures Dataset Characteristics and Transfer", IEEE Trans. Med. Imaging, vol. 35, no. 5, pp. 1285-1298, 2016.

[5] F. J. Huang, Y. LeCun, "Large-scale learning with SVM and convolutional nets for generic object categorization", Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 1, pp. 284-291, 2006.

[6] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in Neural Information Processing Systems, vol. 2, pp. 1097-1105, 2012.

[7] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets", Proceedings of the British Machine Vision Conference, 2014.

[8] M. D. Zeiler, R. Fergus, "Visualizing and understanding convolutional neural networks", Proceedings of the European Conference on Computer Vision, 2014.

[9] J. Chorowski，J. Wang，JM Zurada，Neurocomputing Review and performance comparison of SVM-and ELM-based classi fiers，vol。128，pp。507-516,2014。

[10] K. He, J. Sun, "Convolutional neural networks at constrained time cost", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5353-5360, 2015.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, 1998.

[12] J. Neumann, C. Schnörr, G. Steidl, "Combined SVM-based feature selection and classification", Mach. Learn., vol. 61, no. 1–3, pp. 129-150, 2005.

[13] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, pp. 62-66, 1979.

[14] L. Breiman, "Bagging predictors," Machine Learning, Vol. 24, No. 2, pp. 123–140, 1996.

[15] Y. Kaneda, Y. Pei, Q. F. Zhao, and Y. Liu, "Improving the Performance of the Decision Boundary Making Algorithm via Outlier Detection," Journal of Information Processing, Vol. 23, No. 4, pp. 497–504, 2015.