Product Defect Detection Based on Transfer Learning of CNN

Kai Su, Qiangfu Zhao
School of Computer Science and Engineering
University of Aizu, Aizuwakamatsu, Japan
{s1232002, qf-zhao}@u-aizu.ac.jp

Abstract: In this paper, we focus on product surface defect detection. Defect detection is an essential step in a production line. So far, this task has been conducted mainly by human inspectors. The inspection results are often affected by various human factors like inspector's experiences, health conditions, and so on. To improve the accuracy, in this study we apply the convolution neural network (CNN) to support the human inspector. In recent years, CNN has been applied successfully for image recognition in various fields. In this article, we investigate several methods based on CNN, and report results obtained through experiments on image datasets provided by our partner company. Results show that both AlexNet and GoogLeNet can recognize surface defect very well with the recognition rates of 99.63% and 99.51%, respectively. The proposed system can "reject" a certain percentage of the data and leave them for human-based inspection. Also, the system can also detect wrongly labeled data or outliers, and thus can help human inspectors to purify the training data.

Keywords: Product defect detection, convolution neural network (CNN), image classification, transfer learning.

I. INTRODUCTION

With the improvement of production technology, the product yield and production efficiency are substantially increased, but there is still no guarantee that all products comply with production standards. To control the product quality, quality inspection is usually an essential step in a production line. So far, product defect detection has been conducted mainly by human experts. However, human operation is usually slow and inefficient, and the inspection results are often influenced by various human factors (e.g. experiences and health condition). Therefore, it is expected that an intelligent machine can replace the human inspector partially or completely.

Various methods proposed in the context of machine learning can be used to detect product defect, and convolution neural network (CNN) is one of the most successful ones. So far, CNN has been applied to a wide range of image recognition applications, including handwritten digits recognition, food image classification, railway surface inspection, and so on. We believe that CNN is also useful for product defects detection.

However, compared to other classification problems, it is challenging to replace human inspectors by using CNN because a single false negative error rate can discredite a company. Thus, instead of using CNN for perfect defect detection, it is better to use CNN for "machine detectable" defect detection. That is, the CNN may detect "easy" defects, but leave "difficult" ones to the human experts. If most defects are easy, the labor of the human experts can be reduced greatly.

Fortunately, the outputs of a CNN can be used to classify the "easiness" of a defect. In most case, the outputs of a CNN are defined by using a soft-max function, and the value can be considered the posterior probability of a certain class given an observation. This probability in turn can be considered the confidence of assigning an input pattern into a certain class. If the confidence is low, it is better to leave the conclusion to the human expert.

In fact, other machine learning models also provide some kinds of "scores" for making a certain conclusion. The advantage of using CNN we believe, is that CNN can reduce the number of difficult patterns. That is, using CNN, we can reduce the labor of the inspectors as much as possible. Our belief is based mainly on the fact that CNN is usually more powerful than other models.

In this paper, we report three sets of experimental results: (1) product defect detection by different CNN models; 2) find a rejection threshold to filter out a certain percentage of the data; and (3) study the possibility of improving the performance using a CNN ensemble.

The rest of this article is structured as follows. In Section 2, we briefly introduce the CNN and the CNN models we use; Section 3 shows the experimental results; and in Section 4, we draw some conclusions and discuss the future work.

II. PRELIMINARIES

In this section, we briefly introduce the history and development of CNN, and introduce two CNN models used in our experiments. For more details, readers may refer to the references.

A. Convolution Neural Network

Currently, CNN is known as the state-of-the-art technique for image recognition. CNN as a commonly used deep neural network, it has been developed over a relatively long period. In 1980, Kunihiko Fukushima put forwarded the Neocognitron inspired by the natural visual perception mechanism. Since the 1990s, Yann LeCun et al. designed a multi-layered artificial neural network, called LeNet-5, to classify handwritten digits. LetNet-5 consists of three convolutional layers, two pooling layers, and a fully connected layer. LeNet-5 has established the basic framework of CNN.

The basic structure of a CNN is shown in Figure 1. It contains one or more convolution layers, pooling layers, and fully connected layers.

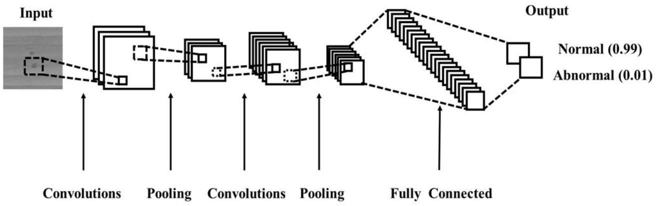

Figure 1. Structure of a Convolution Neural Network

The primary function of the convolutional layer is detecting image features by using convolution kernels. There are many convolution kernels in a convolutional layer; each convolution kernel can detect a different image feature. Each convolution kernel translates the original image matrix to a new image matrix. The primary function of the pooling layer is compressing matrix obtained by convolutional layer. In the process of pooling, the most common pooling operations are mean-pooling and max-pooling They respectively are to obtain the maximum value and average value of the output of each convolution kernels. The fully connected layer is the "classifier" in the whole CNN, that makes the final classification using results of the previous layers.

B. AlexNet

AlexNet was the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) championship in 2012. An AlexNet has eight layers with five convolutional layers and three fully connected layers. The structure of an AlexNet is shown in Figure 2. The last layer is fully-connected and is fed to a 1000-way softmax which produces a distribution over the 1000 class labels.

The characteristics of the AlexNet shown as follows:

- It requires the input size of images: 227-by-227-by-3.
- Extract the features using 96 convolutional kernels of size 11-by-11-by-3 with a stride of 4 pixels (the stride meaning is the length required for each scanning movement of the convolution kernel).
- Using Rectified Linear Unit (ReLU) as an activation function to replace traditional activation function such as sigmoid and tanh, the calculate efficiency and network training times are improved.
- Using dropout in two fully connected layers. So, in the fully connected layers, some neurons do not contribute to the forward pass and do not participate in backpropagation. This method can effectively reduce overfitting.

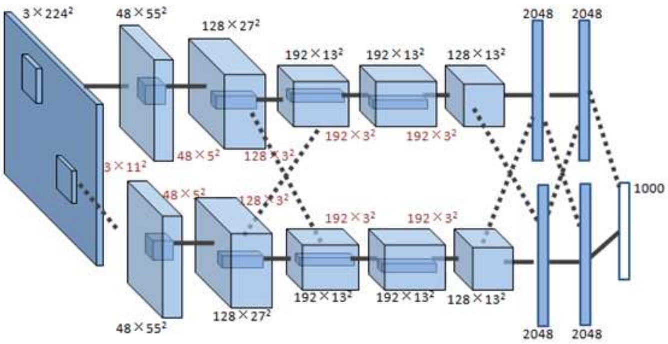

Figure 2. Structure of an AlexNet

C. GoogLeNet

GoogLeNet was the winner of the ILSVRC championship in 2014. GoogLeNet is smaller and more accurate than AlexNet on the original ILSVRC dataset, and GoogLeNet also can classify the images into 1000 object categories. The simplest way to improve performance in deep learning is to use more layers and more data. Therefore, GoogLeNet is more in-depth; it is 22 layers deep. The structure of a GoogLeNet shown in Figure 3.

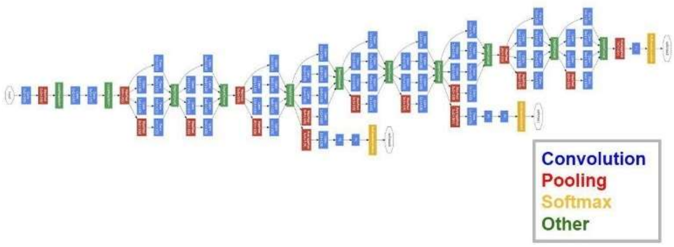

Figure 3. Structure of a GoogLeNet

The characteristics of the GoogLeNet shown as follows:

- It requires the input size of images: 224-by-224-by-3.
- Extract the image features using different size of convolutional kernels.
- Using 1×1 size convolution with 128 convolution kernels for dimension reduction and rectified linear activation.
- Dropout layer with 70% ratio of dropped outputs.
- Linear layer with softmax loss as the classifier.

Especially, the GoogLeNet use nine inception layers; inception layer is a "Network-in-Network" module, that can regard as a combination of multiple network layers, to increase the representational power of neural networks.

Table 1: Recognition rates averaged over 10 runs

	AlexNet (Test Data)	GoogLeNet (Test Data)
For testing dataset	0.9857	**0.9894**
For evaluation dataset	0.95978	**0.96253**

III. EXPERIMENTS FOR PRODUCT INSPECTION

This section describes the three sets of product defect experiments based on transfer learning of CNN. Transfer learning is a deep learning approach that has been trained for the original task is used as a starting point to train our model. Through this approach, our network does not need to be trained for as many epochs, and thus reduce training time and compute resources. Therefore, we use a transfer learning approach by using AlexNet and GoogLeNet model as the starting point for our experiment.

As the first step of the experiment, we need to collect product image data and labels of all the image data. In this study, all images were provided by our partner company. We have two datasets. The first one is used for training, validation, and testing. The second dataset is reserved only for evaluating the trained network. The first dataset contains 3,034 normal product images and 3,020 abnormal product images. The second dataset contains 20,481 product images. In the following, we introduce three experiments. In these experiments, we used neural network toolbox of MATLAB as an experiment platform.

A. *Experiment I: Classification via CNN models*

The first experiment is to use the first dataset for transfer learning using the two CNN models. For transfer learning, we replaced the last three layers of the CNN and set the final classification layer to have only two output neurons. We slit the dataset into three subsets, 70% for training (among them 30% were used for validation), and 30% for testing. The second dataset is used only for final evaluation. Table 1 shows the accuracy of the two CNNs averaged over 10 runs. From these results, we can see that GoogLeNet is a little bit better than AlexNet.

B. *Experiment II: Reject data by classification threshold*

In the last experiment, we achieved more than 95% accuracy in our two datasets, these results look good, but this was still not enough for real detection applications. After we analysed these results, we found many misclassified images; we cannot tolerate too many mistakes in the real factory. So, we hope our network models can leave some images that do not have sufficient confidence to make a decision.

To achieve this goal, we modified the discriminant strategy in the classification layer of our network models; if the classification probability of given image is lower than the threshold, we reject that image. After this, we evaluated the network using the reserved dataset. Figs. 4-5 show the variation in the reject quantity and accuracy of the two network models at different thresholds.

From these results we can see that as the threshold increases, the reject quantity and classification accuracy rate are increasing, either in the AlexNet model or GoogLeNet model. In this study, we expect the network to leave about 20% of the data to human detection. Therefore, we can obtain the best threshold with an acceptable quantity based on the above results

C. *Experiment III: CNN ensemble*

In this experiment, we use seven (an odd number) CNN models which were trained by transfer learning with the categorized dataset and used the excellent threshold which obtained by experiment II, for each CNN model using $0.7 \times N_t$ data randomly to transfer learning of CNN model, where N_t is the total number of the categorized dataset. In this CNN ensemble system, each CNN model has classified or rejected the given image and then make the final decision based on majority voting. Tables 3-4 show the confusion matrices of this experiment in two CNN models.

In these results above, we can obtain the very good recognition accuracies from this two CNN ensemble system, the percentage of rejected data are 22.94% and 16.58% respectively, these part data are ambiguous for our CNN

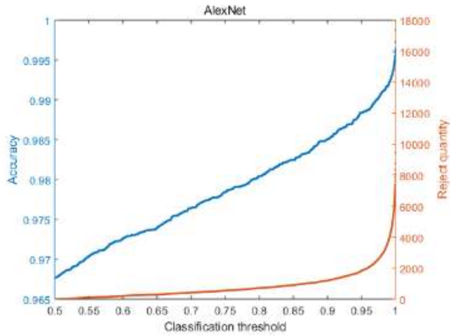

Figure 4: Variation of accuracy and reject quantity on AlexNet model

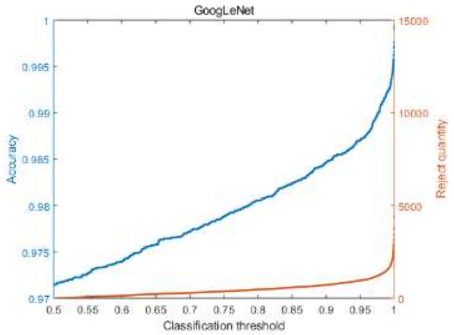

Figure 5: Variation of accuracy and reject quantity on GoogLeNet model

ensemble system, so human experts must re-examine these data. For other not rejected data, this system has the absolute confidence to decide. However, there are still 58 and 84 errors in our results, in these images, there are some apparent misclassifications. In other word, some training data may not have correct labels from the beginning. In this sense, the CNN can help human experts to find labelling mistakes in the training dataset. After reducing mis-labeled data, we may expect to obtain better results in the next step.

VI. CONCLUSION

In this paper, we have applied the AlexNet and GoogLeNet to the task of product defect detection; experimental results show that two CNNs can detect the product defects very accurately. In the case of similar accuracy, the GoogLeNet can make more accurate judgments; it is to reduce the number of rejections effectively.

In the next step, we will try to classify the types of the defects in more detail, so that human experts can use the results to improve the production line and reduce the defects as much as possible.

Table 2: Confusion matrix of AlexNet

	Predicted normal	Predicted abnormal
True normal	5584	51
True abnormal	7	10139
Accuracy	0.9963	
Rejection	4700	

Table 3: Confusion matrix of GoogLeNet

	Predicted normal	Predicted abnormal
True normal	6316	78
True abnormal	6	10686
Accuracy	0.9951	
Rejection	**3395**	

REFERENCES

[1] Lien Po Chun, Qiangfu Zhao, Product surface defect detection based on deep learning, 2018.

[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] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern Recognition, 2018, 77: 354-377.

[4] 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.

[5] Fukushima K, Miyake S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition, pp. 267-285, 1982.

[6] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[7] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in neural information processing systems. 2012: 1097-1105.

[8] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[9] Zhong Z, Jin L, Xie Z. High performance offline handwritten chinese character recognition using googlenet and directional feature maps[C]. Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE, 2015: 846-850.

[10] Singla A, Yuan L, Ebrahimi T. Food/non-food image classification and food categorization using pre-trained googlenet model[C]. Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management. ACM, 2016: 3-