Product Surface Defect Detection Based on CNN Ensemble with Rejection

Kai Su, Qiangfu Zhao, Lien Po Chun
School of Computer Science and Engineering
The University of Aizu
Aizuwakamatsu-city,Fukushima, Japan
{s1232002, qf-zhao, m5212107}@u-aizu.ac.jp

***Abstract*—Deep learning-based pattern recognition has achieved impressive accuracy recently. In particular, convolutional neural networks (CNNs) have been successfully applied to solving various practical problems. One of the problems is the image-based product defect inspection. Product quality inspection is an essential step in a production line. So far, this task has been conducted mainly by human inspectors, and the inspection results are often affected by various human factors like experiences, health conditions, and so on. CNN can be useful for reducing human factors and thus can be good for quality control. In practice, however, CNN-based defect detection is still a challenging task. One reason is that a single false negative (FN) error (i.e., classifying a defect as a normal case) may bring severe damage to the company, and this is the crucial difference between image recognition and defect detection. To reduce the FN error rate, in this study, we investigate the effect of "rejection" for defect detection. Experimental results demonstrate that a CNN ensemble with a proper rejection rate can have a very low FN error rate and can reduce human labor significantly.**

Keywords—Pattern recognition, image classification, product defect detection, convolutional neural network, transfer learning.

I. INTRODUCTION

With the continuous progress in industrial technology, companies can now provide a large number of products with high speed. To control the quality of the products, however, we still need a lot of human resources. Conventionally, product quality inspection has been conducted mainly by human operators, and the results often depend on various human factors (e.g., work experiences, health conditions, etc.). In addition, human-based inspection can be time-consuming and difficult to match the speed of modern industry. It is thus expected that an AI system can partially or entirely replace human operators for product quality inspection.

Basically, we can use an AI to perform product inspection based on methods proposed in the context of pattern recognition. In fact, pattern recognition is the starting point for an AI to emulate the cognitive ability of human. Here, patterns can be sounds, images, actions, etc. Generally speaking, even the reasoning or decision-making processes of human can be considered patterns. In this study, we consider only image-based pattern recognition because the methods proposed in this area can be useful for product defect detection.

For image recognition, convolutional neural network (CNN) is now considered the state-of-the-art method. Some CNN models (e.g., AlexNet [8], GoogLeNet [9], etc.) have been the winners of international image recognition contests in recent years. So far, CNN has been applied to a wide range of applications, including food image classification [2], railway defect inspection [3] [4], nuclear power plant crack detection [5], and so on.

The authors of [5] proposed to detect tiny cracks based on CNN for nuclear power plants. It is known that periodic inspection of nuclear power plant components is essential to guarantee safety operations. The CNN architecture given in [5] can detect tiny cracks with low contrast and variant brightness that are hardly visible even by human operators. The CNN model can achieve a higher hit rate than other methods. Since product defects are similar to cracks, CNN should also be useful for image-based defect detection.

Generally speaking, image recognition and product defect detection are two different problems, although they are similar. For the former, we usually measure the accuracy using the recognition rate. For the latter, however, it is necessary to reduce the false negative (FN) error rate as far as possible because a single FN error can cause significant damage to the brand of a company. If we set a very low FN error rate, the false positive (FP) error rate (also called false alarm rate) usually becomes exceptionally high, and as a result, human labor cannot be reduced significantly. Thus, it is almost useless if we adopt methods proposed for image recognition directly to product defect detection.

Instead of using CNN for two-class (i.e. positive or negative) classification, it is better to introduce the third class "rejection" to reduce the FN and FP error rates simultaneously. Generally speaking, patterns to be rejected are relatively "difficult" ones. That is, for these patterns, the CNN cannot make a "confident" decision. Since in most cases a skillful human inspector can distinguish NG (not good) products from good ones using his/her eyes, the defects must have some "special features". Theoretically, a properly trained CNN can extract these special features and can make confident decisions in most cases, and thus, the number of products rejected by CNN cannot be too many. Thus, we may expect that a CNN with rejection can reduce human labor significantly because the human inspector needs only to re-check the rejected products.

Fortunately, the outputs of CNN can be used to measure the confidence of a decision. For example, if the outputs of a CNN are defined by using a soft-max function, 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 to a certain class. If the confidence is not high enough, it is better to leave the conclusion to the human operator. In fact, other machine learning models (e.g., a support vector machine) also provide some kind of "confidence score" for making a

decision. The advantage of using CNN is that a CNN usually has higher potential, and does not need feature engineering for extracting and/or selecting useful features [6].

Training a good CNN model often requires a considerable amount of labeled data as well as powerful computing resources. Instead of training a CNN from scratch, researchers nowadays often re-train a well-designed CNN via transfer learning, with a relatively smaller dataset [18, 19, 20, 21]. AlexNet and GoogLeNet are two of the well-known CNN models trained using the dataset ImageNet [7]. ImageNet contains over 15 million high-resolution images with tags, and there are approximately 22,000 categories. Using transfer learning, we can re-use most parameters of a well-trained model, and modify the layers close to the output layer, to adapt the model to the domain under concern. Transfer learning is much more efficient and effective than training a CNN from scratch. In this study, we also use transfer learning to design CNNs for product defect detection.

As for the well-trained model, we adopt the AlexNet and GoogLeNet. AlexNet was proposed in 2012 by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton; and GoogLeNet was proposed in 2014 by Christian Szegedy, Wei Liu, Yangqing Jia, et al. These two models were the winners of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 and 2014, respectively, and have been widely used for solving various image recognition problems. In this paper, we report three sets of experimental results based on AlexNet and GoogLeNet models: (1) product defect detection by a single CNN model; (2) find a rejection threshold of a CNN model 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 models we used; Section 3 shows the experimental results; and in Section 4, we draw some conclusions and discuss some future work for product inspection.

II. Related Work

To make this paper relatively self-contained, we introduce the history and development of CNN, and two CNN models useful for product defect detection briefly. For more details, readers may refer to the references.

A. Convolutional Neural Network

Currently, CNN is known as the state-of-the-art technique for image recognition. A well-trained CNN even outperforms human for ImageNet dataset [7]. CNN, as a commonly used deep neural network; has been developed over a rather long period. In the 1980s, Kunihiko Fukushima put forwarded the Neocognitron, which was inspired by the visual perception mechanism [10]. Here, visual perception mechanism means that the information processing of the visual system is intrinsically hierarchical in a biological brain. CNN was inspired by the Neocognitron model. In the 1990s, Yann LeCun et al. studied CNN for handwritten character recognition, and the original model was called LeNet-5 [11]. LeNet-5 consists of 5 hidden layers, namely three convolutional layers and two pooling layers. Similar to multilayer perceptron (MLP), LeNet-5 has a fully connected output layer. LeNet-5 is considered the initial CNN model.

A CNN usually consists of various layers arranged in sequence. Each layer in the network transmits the feature data from one layer to another. The CNN is mainly composed of three types of layers, namely, convolutional layer, pooling layer, and fully connected layer. We can construct a CNN by stacking these layers together; the basic structure of a CNN is shown in Fig. 1.

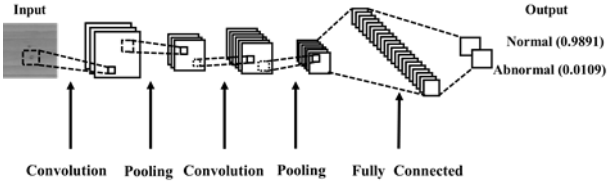

Fig. 1. Structure of a Convolution Neural Network

The primary function of the convolution layer is detecting image features by using convolution kernels. There are many convolution kernels in a convolutional layer; each kernel can detect a different image feature. Each convolution kernel translates a given image or matrix to a new matrix. As in a normal MLP, each element of the new matrix is the output of a non-linear activation function for the "effective input", which is a combination of information obtained from different channels. Activation functions commonly used for the convolutional layer are sigmoid, rectified linear unit (ReLU), leaky ReLU (LReLU), parametric ReLU (PReLU), and so on.

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 obtain the maximum value and average value, respectively, of the output matrix in a sliding window. Its function is to reduce the size of the image matrices and the number of parameters in the network so that the computational resources are less expensive, and over-fitting can be effectively controlled.

The fully connected layer is the "classifier" in the whole CNN that makes the final decision using results of the previous layers. There can be several fully connected layers in a CNN. In some sense, a CNN is actually a stacked structure of a feature extractor and a conventional MLP. The former usually contains several (convolution, pooling) layer pairs.

B. AlexNet

AlexNet is a classical CNN trained by using more than one million images from the ImageNet database, and AlexNet was the winner of ILSVRC in 2012. An AlexNet has eight layers with five convolutional layers and three fully connected layers. The structure of an AlexNet is shown in Fig. 2. The last three layers are fully-connected, and the last layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels.

The characteristics of the AlexNet are shown as follows:

- The input size is 227-by-227-by-3.
- AlexNet extracts the features using 1376 convolutional kernels (or filters) which include 96 11-by-11 filters with stride 4 and padding 0; 256 5-by-5 filters with stride 1 and padding 2; and 1024 3-by-3 filters with stride 1 and padding 1. Here, the

stride is the step size for scanning the input matrix using a filter, and padding m means add m "frames" with zeros around the border of the input matrix.

- AlexNet using Rectified Linear Unit (ReLU) as an activation function to replace traditional sigmoid activation function. Calculation efficiency and network training times are improved.
- AlexNet using dropout in two fully connected layers. That is, 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.

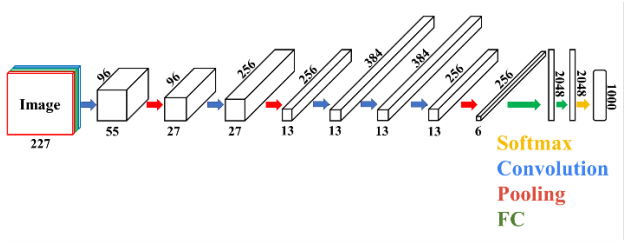

Fig. 2. Structure of an AlexNet

C. *GoogLeNet*

GoogLeNet (or Inception-v1) was the winner of the ILSVRC in 2014. GoogLeNet is smaller and more accurate than AlexNet on the original ILSVRC database, and can also classify the images into 1000 object categories. The simplest way to improve performance in deep learning is to use more layers and more data. GoogLeNet is deeper, and it has 22 layers. The structure of a GoogLeNet shown in Fig. 3.

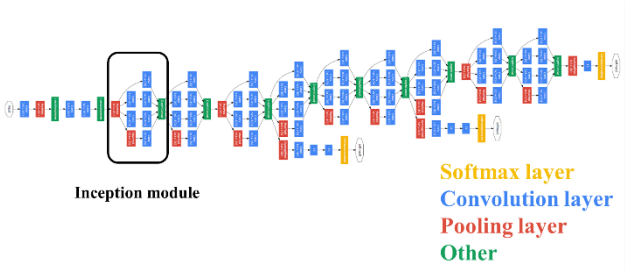

Fig. 3. Structure of a GoogLeNet (taken from [9])

The characteristics of the GoogLeNet are shown as follows:

- The input size is 224-by-224-by-3.
- GoogLeNet does not have fully connected layers, and there are only about 5 million parameters.
- Linear layer with softmax loss as the classifier.
- Using 1-by-1 convolutional kernels for dimension reduction and rectified linear activation.
- Put forward an efficient "Inception" module.

Especially, the GoogLeNet uses nine inception module. Each inception module is a "network within a network" structure, that can be regarded as a combination of multiple filters with different sizes. The inception modules can increase the representational power of the network.

III. Experiments for Product Inspection

This section describes three sets of product defect experiments based on transfer learning of CNN. Transfer learning is the process of taking a pre-trained model and usually use a new smaller data set to adjust the model. Based on this approach, we do not have to train the network for many epochs, and thus the training time can be reduced. In our study, we adopt transfer learning on AlexNet or GoogLeNet.

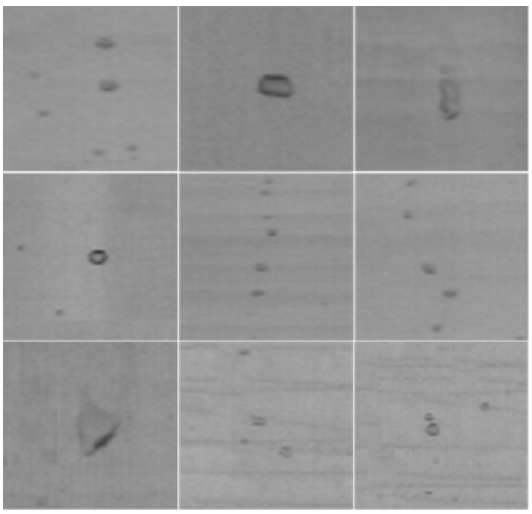

Fig. 4. Examples of normal products

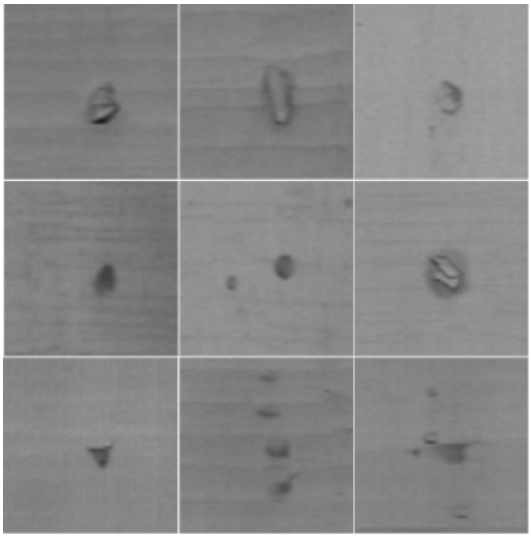

Fig. 5. Examples of abnormal products

As the first step of the experiment, we need to collect product image data and labels of the data. In this study, our partner company provided us with two sets of image data, and inspection experts labeled all image data. The first one is used for training, and validation, and to determine hyper-parameters related to learning (e.g., number of epochs, etc.). The second dataset is reserved only for evaluating the trained model. The first dataset contains 3,034 normal product images and 3,020 abnormal product images. The second dataset includes 20,481 product images. Fig. 4 and Fig. 5 show some examples of normal product images and abnormal ones. Due to the confidentiality of these image data, we are unable to describe the details of these product images.

In the following, we introduce three sets of experiments. In these experiments, we used the deep learning toolbox of MATLAB as an experiment platform, and the experimental computer is equipped with an Intel Core i7-7800X CPU, 8-GB memories, and an NVIDIA GeForce GTX 1080 GPU.

A. Experiment I: Classification using a single CNN

The first experiment is to use the first dataset for transfer learning using one of the two CNN models. Here, we consider defect detection as a classification problem and use CNN directly for defect detection. For transfer learning using a pre-trained network, we need to pay attention that both AlexNet and GoogLeNet have 1000 outputs. This parts must be replaced by two outputs (normal or abnormal) for our purpose.

In the experiment, we randomly divided the first dataset into two subsets, 70% for training, and 30% for testing. For the training set, 70% were actually used for training, and 30% were used for validation. The second dataset was used for the final evaluation. Table I shows the accuracy of the two CNNs averaged over ten runs. For each model, we have two columns of results. The first column is the test accuracy (recognition rate) for the test data of the first dataset, and the second one is the accuracy for the second dataset. From these results, we can see that GoogLeNet is a little bit better than AlexNet.

In addition, we can see that the recognition rates in both AlexNet and GoogLeNet were not stable in these results. For example in the first column, recognition rates fluctuated between 97.25 and 99.06, the range was up to 2%. It shows that single networks are very occasional when identifying some product images; the single network might have better or worse accuracy.

TABLE I. RESULTS OF THE EXPERIMENT I

Run	Recognition rates for different runs			
	AlexNet #1	*AlexNet #2*	*GoogLeNet #1*	*GoogLeNet #2*
1	98.98	95.87	98.90	95.47
2	98.51	96.18	99.06	96.81
3	97.25	95.77	99.13	96.31
4	98.82	96.03	98.74	96.66
5	98.98	95.64	98.51	96.07
6	98.51	95.85	99.06	96.56
7	98.66	96.22	98.98	96.02
8	98.58	96.08	98.90	96.17
9	99.06	96.01	99.21	96.00
10	98.35	96.13	98.90	96.46
Avg.	98.57	95.978	98.939	96.253
Std.	0.4945	0.1795	0.1921	0.3726

B. Experiment II: Reject Data by Classification Threshold

In Experiment I, we achieved around 96% accuracy for the evaluation dataset (i.e., the second dataset). This result looks good, but after analyzing the results in more detail, we found that the false negative error rate (or rate of classifying an abnormal product to a normal one) was still not good enough for real defect detection because we cannot tolerate too many mistakes in the real situation.

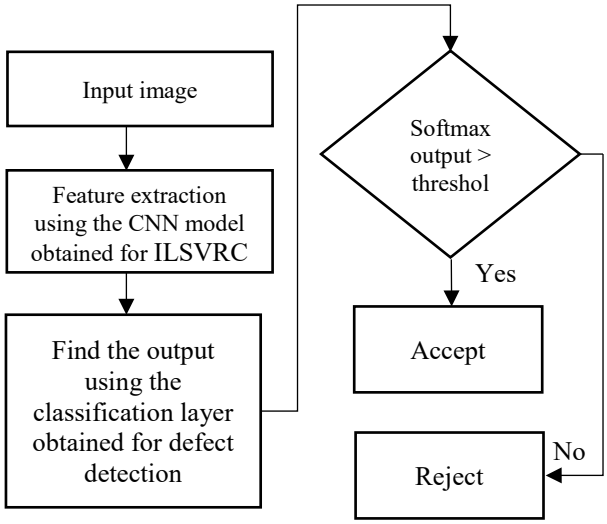

Fig. 6. Flowchart of the proposed method

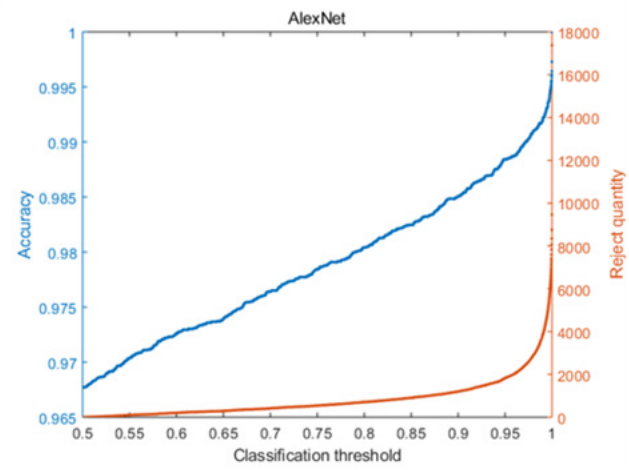

Fig. 7. Variation of accuracy and number of rejections for AlexNet

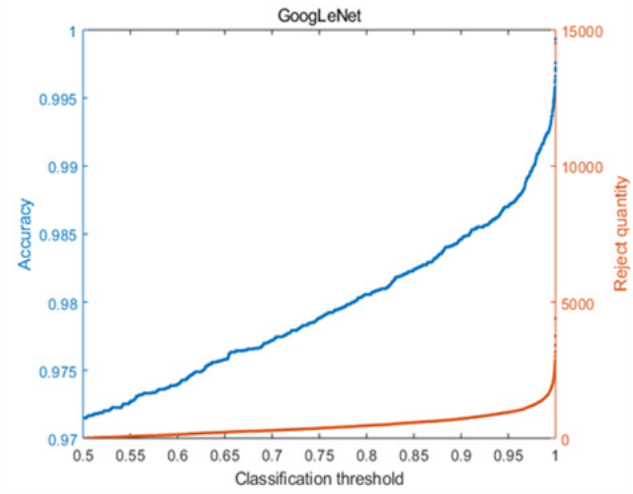

Fig. 8. Variation of accuracy and number of rejections for GoogLeNet

To reduce the number of FN errors, we may decrease the threshold for making a defect decision. If the defect rate is low, this method is quite useful. However, if the number of abnormal images is large, decreasing the threshold may make "automatic defect detection" meaningless because

too many normal products will be classified to abnormal ones. In this paper, we propose to solve the problem using "rejection". The basic idea is that, if the decision made by a CNN is of high confidence, we can accept the decision; otherwise, we can reject the product, and leave it to the human expert for making the final decision.

To incorporate the rejection mechanism into the defect detection process, we modify the discriminant strategy in the classification layer of CNN model. The classification layer of the CNN uses softmax activation function, and each output value can be considered the posterior probability of a certain class given an observation. If the classification probability of a given image is lower than a threshold, we can reject that image (or the corresponding product). Fig. 6 shows the flowchart of the proposed method.

Figs. 7-8 show the relation between the threshold value and the accuracy for AlexNet and GoogLeNet, respectively. In the figures, the number of rejections is also included. Note that the accuracy is nothing but the probability of correct decision when the output more than or equal to threshold. Clearly, the higher the threshold is, the more difficult to accept the decision made by the network. From these results, we can see that as the threshold increases, both the number of rejections and the classification accuracy rate increase. This is true for both AlexNet and GoogLeNet. However, it is interesting to see that the accuracy increases with a higher speed. That is, by adding a few more rejections, we can obtain a much better accuracy. For example, if we reject and leave about 20% of the data to human detection, we can have an accuracy higher than 99% both for AlexNet (Tables II) and for GoogLeNet (Table III). Fig. 9 shows the receiver operating characteristic curve (ROC curve) of the GoogLeNet model with and without rejection. Clearly, the area under the curve (AUC) becomes larger if we use rejection.

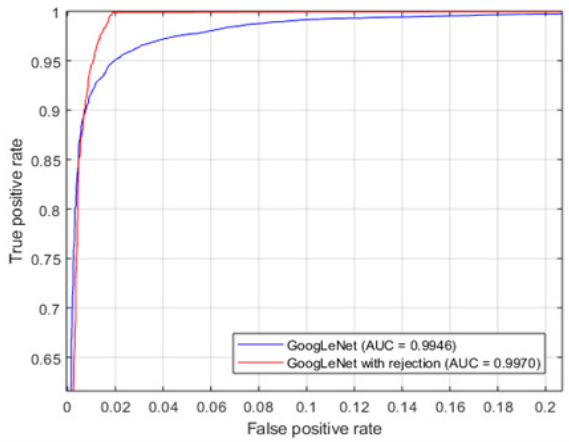

Fig. 9. ROC comparison with conventional GoogLeNet model

TABLE II. CONFUSION MATRIX WITH REJECTION FOR ALEXNET

	Predicted normal	Predicted abnormal
True normal	6066	81
True abnormal	12	9553
Accuracy	99.41 %	
#Rejection (rate)	4769 (23.28%)	

TABLE III. CONFUSION MATRIX WITH REJECTION FOR GOOGLENET

	Predicted normal	Predicted abnormal
True normal	5601	129
True abnormal	**8**	10750
Accuracy	99.17 %	
#Rejection (rate)	**3993** (19.50%)	

TABLE IV. CONFUSION MATRIX WITH CNN ENSEMBLE FOR ALEXNET

	Predicted normal	Predicted abnormal
True normal	5584	51
True abnormal	**7**	10139
Accuracy	99.63 %	
#Rejection (rate)	4700 (22.95%)	

TABLE V. CONFUSION MATRIX WITH CNN ENSEMBLE FOR GOOGLENET

	Predicted normal	Predicted abnormal
True normal	6316	78
True abnormal	**6**	10686
Accuracy	99.51 %	
#Rejection (rate)	**3395** (16.58%)	

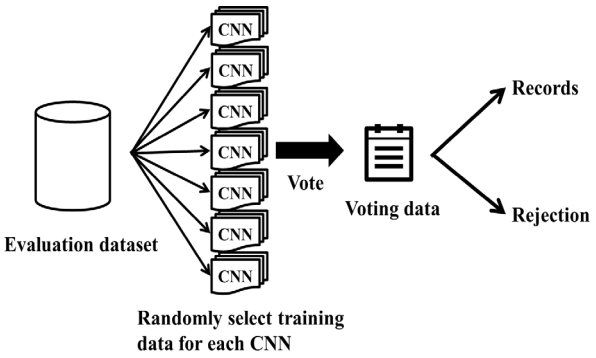

Fig. 10. Decision making by CNN ensemble voting system

C. CNN Ensemble with Rejection

From the above experiments, we can find that the classification results of a single CNN are not stable due to the randomness in selecting the training data. Therefore, in this experiment, we use multiple CNNs (ensemble) to obtain better predictive performance. In this study, we used seven (an odd number) CNN models which were trained via transfer learning with the first dataset and use the applicable threshold obtained in Experiment II. For each CNN model, we used 70% of the data randomly selected from the first database (and 30% of the data were used for validation). In the CNN ensemble, each CNN model can classify or reject a given image. The final decision is then made based on majority voting. Fig. 10 illustrates the flow of an ensemble voting system for decision making.

Using a CNN ensemble system to detect product defect, we can obtain better recognition rates than a single network.

Tables IV-V show the confusion matrices of Experiment III. The percentages of rejected data are 22.95% and 16.58%, respectively for AlexNet and GoogLeNet. It is interesting to note that although there are still 6 FN errors (see Fig. 11), 4 of them are actually "true negative" data. It seems that only the two images shown in the right column correspond to true positive data (these two images are not clear enough to make definite decisions even by human experts). These data have been wrongly classified" (or labeled) by human experts. In this sense, the CNN ensemble can help us to correct human mistakes.

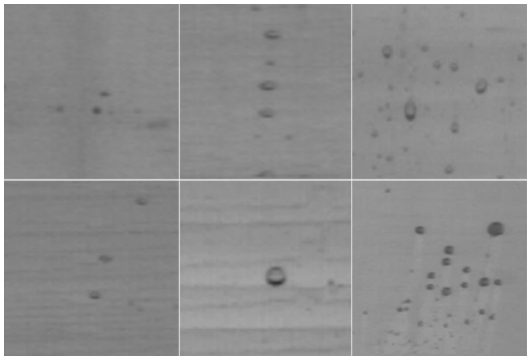

Fig. 11. The six FN errors made by the GoogLeNet ensemble.

IV. CONCLUSION

In this paper, we have applied AlexNet and GoogLeNet to product defect detection. Experimental results have shown that these two CNN models can detect product defects very accurately, especial when they are used as a member of an ensemble. Among them, GoogLeNet can make more accurate judgments and reduce the number of rejections effectively.

In the next step, we will try to classify the defect types as well, so that human experts can use the results to improve the production line and reduce the defects as much as possible. We also need to carry out experiments with more effective algorithms to obtain higher recognition results. To reduce the FN errors, we will also try to put more weights on abnormal data for training, so that characters of abnormal data can be learned more exactly.

Besides, since ensemble training is usually time-consuming, it is necessary to adopt some highly efficient models in the ensemble method. We will try a new pre-trained model called MobileNet [12][13], which is known to be faster than other CNN models. In fact, MobileNet was proposed to build a smaller model with fewer parameters and faster classification time for mobile devices. Using MobileNet, it is possible to design a portable classifier without using GPU for detecting product defect in the future.

ACKNOWLEDGMENT

We would like to thank our partner company for providing the labeled product data and equipment.

REFERENCES

[1] Lien, Po Chun, and Qiangfu Zhao. "Product Surface Defect Detection Based on Deep Learning." 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). IEEE, 2018.

[2] Singla, Ashutosh, Lin Yuan, and Touradj Ebrahimi. "Food/non-food image classification and food categorization using pre-trained googlenet model." Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management. ACM, 2016.

[3] Shang, Lidan, et al. "Detection of rail surface defects based on CNN image recognition and classification." Advanced Communication Technology (ICACT), 2018 20th International Conference on. IEEE, 2018.

[4] Chen, Junwen, et al. "Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network." IEEE Transactions on Instrumentation and Measurement 67.2 (2018): 257-269.

[5] Chen, Fu-Chen, and Mohammad R. Jahanshahi. "NB-CNN: deep learning-based crack detection using convolutional neural network and naive Bayes data fusion." IEEE Transactions on Industrial Electronics 65.5 (2018): 4392-4400.

[6] Wen, Si, et al. "Comparison of Different Classifiers with Active Learning to Support Quality Control in Nucleus Segmentation in Pathology Images." AMIA Summits on Translational Science Proceedings 2017 (2018): 227.

[7] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. Ieee, 2009.

[8] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[9] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[10] Fukushima, Kunihiko, and Sei Miyake. "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition." Competition and cooperation in neural nets. Springer, Berlin, Heidelberg, 1982. 267-285.

[11] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

[12] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

[13] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[14] Gu, Jiuxiang, et al. "Recent advances in convolutional neural networks." Pattern Recognition 77 (2018): 354-377.

[15] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400 (2013).

[16] Zhong, Zhuoyao, Lianwen Jin, and Zecheng Xie. "High performance offline handwritten chinese character recognition using googlenet and directional feature maps." Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE, 2015.

[17] Kaneda, Yuya, et al. "Improving the performance of the decision boundary making algorithm via outlier detection." Journal of information processing 23.4 (2015): 497-504.

[18] Ribeiro, Eduardo, et al. "Transfer learning for colonic polyp classification using off-the-shelf CNN features." International Workshop on Computer-Assisted and Robotic Endoscopy. Springer, Cham, 2016.

[19] Prajapati, Shreyansh A., R. Nagaraj, and Suman Mitra. "Classification of dental diseases using CNN and transfer learning." 2017 5th International Symposium on Computational and Business Intelligence (ISCBI). IEEE, 2017.

[20] Kieffer, Brady, et al. "Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks." 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA). IEEE, 2017.

[21] Sert, Mustafa, and Emel Boyacı. "Sketch recognition using transfer learning." Multimedia Tools and Applications (2019): 1-18.