2016 Nicograph International

Improvements of classification accuracy of film defects by using GPU-accelerated image processing and machine learning frameworks

Hidetoshi Ando, Yuki Niitsu
Department of Computer Science and Engineering
University of Yamanashi
4-3-11 Takeda, Kofu, Yamanashi, JAPAN
ando@yamanashi.ac.jp
g14mk017@yamanashi.ac.jp

Masaki Hirasawa, Hiroaki Teduka, Masao Yajima
Image Processing & Measurement Division
Takano Co., Ltd.
137 Miyada-mura, Kamiina-gun, Nagano, JAPAN
m-hrsw@takano-net.co.jp
h-teduka@takano-net.co.jp
yajima@takano-net.co.jp

***Abstract*—Research on image classification for natural images are quite actively worked on and recent achievements using deep learning techniques are tremendous. On the other hand, image classification techniques of defects in industrial products are mostly kept secret, partly because defective images contain very sensitive information about the products and the confidential manufacturing technologies. With the help of a leading company in a visual inspection of film defects, we investigated the effectivity of using machine learning techniques for classification of defect images. We also made use of GPU to accelerate both image processing to assist detection of defects and machine learning. We propose the combination of deep neural networks with random forest classifier for image classification of film defects, which performed better than using either of the two techniques alone.**

Keywords- Image Classification, Film Defects, GPU, Random Forest, Deep Learning

I. Introduction

Detection and classification of defects of industrial products are essential for manufacturers to avoid shipping defective products to market or to check the mechanical condition of manufacturing machines. Visual inspection is one of the key technologies for non-invasive inspection of products. Takano Co., Ltd. is a leading company in the field of visual inspection systems for film-shaped products. Fig. 1 shows an optical inspection system using line sensors. The system with high-performance cameras provides quick inspection, so that it is ideal for quality control that is under increasing severity. This system has been delivered to wide industries including electronics, packaging and energy industries, and is especially contributing to the production of cell phone, TV set and solar cell. One of the advantages in terms of hardware is the originally developed line sensor camera with the best speed in the industry and processing board to balance quick inspection and fine resolution.

After line sensor camera takes 8000x8000 resolution image, GPU-accelerated image processing libraries including adaptive contrast enhancement clearly discriminate between good and defective areas. Fig. 2 shows image processing examples where the original images captured by the camera are on the left side of the arrows, and the processed images are on the right side. Defects are located and outputted as 128x128 images. Using these images, film defects are easily be found even with human eyes, but the hard part is checking the kind of defect from those images. Currently, there are two types of software to help users classify the images, one to extract image features from the images, and the other to help users create a decision tree by specifying which feature values to use and what threshold values to compare against. The feature extracting software turns the input image into 28 floating point values representing 28 features such as mean brightness, roundness, contour length, and so on. By seeing the statistical properties such as the average and distribution of each feature value, users determine which feature to use to classify the image into a certain defect type. The latter process is quite tedious even for expert visual inspectors or software engineers, and typical classification accuracy is between 50% and 75% even after very careful tuning of threshold values.

To improve the classification accuracy and free human from spending hopeless time to make hand-crafted classification algorithms, we introduce some machine learning algorithms and see how effectively the accuracy rises and what kind of techniques can be added to further improve the results of machine learning, and how machine learning algorithms can be accelerated by using GPU (Graphics Processing Unit).

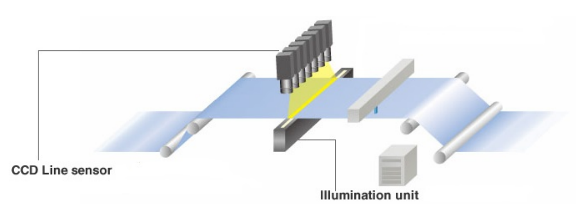


Figure 1. An optical inspection system of film defects

978-1-5090-2305-9/16 $31.00 © 2016 IEEE
DOI 10.1109/NicoInt.2016.15

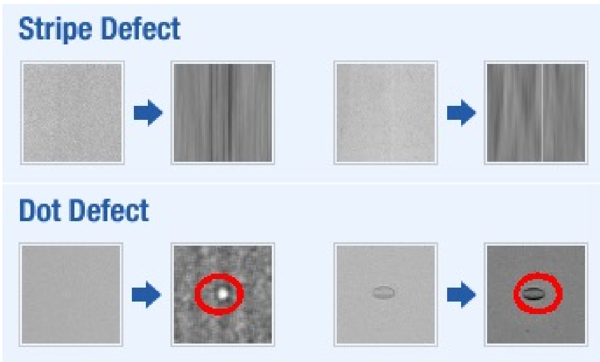


Figure 2. Visual enhancements of candidate defect images with 128x128 resolution

II. RELATED WORKS

There has been and are numerous research works in the field of visual inspection and image classification using machine learning techniques. One of the most popular competitions of image classification is the ImageNet LSVRC (Large Scale Visual Recognition Challenge) [1] using 150,000 photographs to classify into 1000 categories. In 2012, a deep learning technique was introduced [2] and won the competition leaving other competitors in the dust. Since then every year new type of deep neural networks such as in [3] take the 1st place, and the networks become deeper and deeper. The latest champion of ILSVRC2015 [4] has more than 150 layers.

Training such deep neural networks require very heavy computation, and most of them utilize GPUs for acceleration, but often takes several days to train. GPU is not only good for deep learning frameworks, but it also can be used to accelerate shallow machine learning methods like random forest [5]. We will show below how we accelerate our implementation of random forest to deal with big amount of data.

Image classification techniques of defects in industrial products are mostly kept secret today, partly because defective images contain very sensitive information about the products and the confidential manufacturing processes and technologies.

III. DEFECT IMAGE CLASSIFICATION USING A SHALLOW MACHINE LEARNING TECHNIQUE

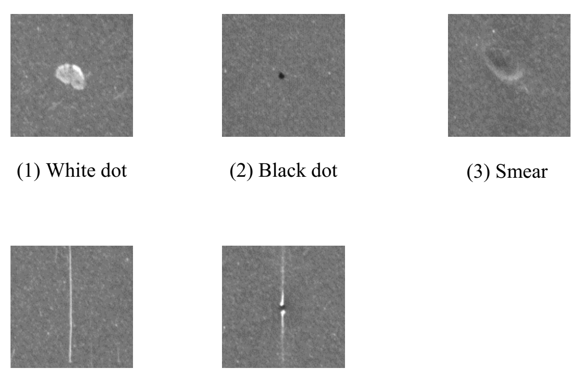
As we already have a feature extracting software from defect images, we can apply a shallow machine learning techniques where users specify the set of features and use the feature vectors as input to the machine learning algorithm. Our first experiment is to use random forest method [5] for the creating classifier. Random forest was introduced by L. Breiman and has been used quite widely for a variety of machine learning tasks including classification, regression, clustering, and so on. It is based on ensemble learning using multiple decision trees and is sometimes called randomized trees or randomized decision trees. Each decision tree corresponds to a weak classifier, and a set of trees (forest) form a highly accurate classifier.

We have accelerated random forest software implementation using GPU to handle huge amount of data. There are already a few implementations of random forest using GPU, [6][7][8] and among them, CURFIL[8] looks to be the best to deal with big data where the size of data excels the size of GPU memory. CURFIL is also the fastest of them as it is reported to be 28 times faster than CPU implementation. Our GPU-accelerated random forest is about 67% faster than CURFIL, by using the latest GPU techniques such as warp shuffle, the use of Read-Only cache memory, and so on.

The set of image data we used for our experiments is described in Table 1 and example images are presented in Fig. 3. There are five different classes of defects as shown in Fig. 3, white dot as class 1, black dot as class 2, smear as class3, white stripe as class4, and fish eye as class5. The total number of defect images is 4511, with different number of images for five classes. There are also some tunable parameters for executing random forest such as the number of trees to use, maximum depth of trees, how many features to check for each decision, and so on, and these values affect the accuracy of the final classifier. We have run through numerous tests by changing these parameters and the best result was obtained by the values shown in Table 2.

TABLE I. DEFECT IMAGE DATA

Classes	# of Total data	Training data	Test data
Class1	17	10	7
Class2	183	60	123
Class3	1940	1920	20
Class4	123	60	63
Class5	2288	1920	368
Total	4511	3970	581



(1) White dot (2) Black dot (3) Smear (4) White Stripe (5) Fish Eye

Figure 3. Five types of film defects in our experiments

TABLE II. KEY PARAMETERS FOR RANDOM FOREST

Max tree depth	Min # of samples	Feature selection	# of trees	Overall precision (%)
10	10	1	100	94.7

As the algorithm uses randomness in many ways, we have run the test 10 times and calculated the average value. As seen from the table, the overall precision of 94.7% was achieved using Random Forest, and this is by far the better than the hand-crafted classifier described before. More detailed precision, recall, and F-measure for each of the 5 classes are shown in Table 3 and the graphical representation in Fig. 4. The numbers look good except for class3. As shown in Fig. 3, compared to other images, the class3 image is blurry and faint, and this can lead the feature extractor to have trouble calculating proper feature values to distinguish the defect from other types of defects. This is one of the limitations of determining the best image features by hand, and we will discuss this issue later by using deep learning techniques.

In the above experiment, we used all of the 28 feature values calculated by the extractor, which resulted in 94.7% overall precision of the classifier. The question now is if all these values are really necessary to accomplish the result, leading to the idea some features may be omitted. Reducing the number of feature values lead to less computation time and storage consumption, which is ideal for computers with limited amount of memory just like those used in embedded systems. We evaluated the importance of features and see how the precision change by removing those features with less importance.

TABLE III. RANDOM FOREST RESULTS

Classes	Precision	Recall	F-measure
Class1	0.98	1.00	0.99
Class2	0.89	1.00	0.90
Class3	0.37	0.32	0.34
Class4	1.00	0.91	0.95
Class5	1.00	0.86	0.92
Overall	0.947	0.947	0.947

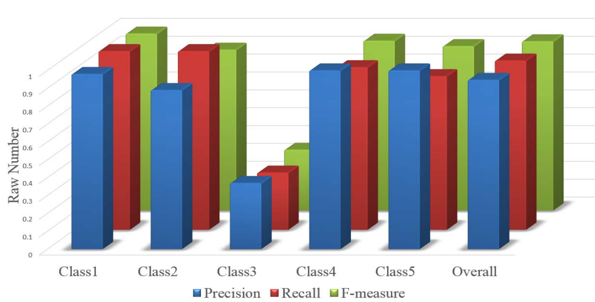


Figure 4. Random forest results in 3D graph

One way to check how important a feature is to the overall precision is to remove the chosen feature from the dataset, but as the number of features is changed, we have to reconstruct the random decision trees. Instead of changing the number of features, we chose one feature and replace the values of the chosen feature in all data with totally random numbers. This results in messing up the chosen feature and effectively make the feature useless, without reconstructing the classifier. By doing this on all features, we can see for example the most important feature or the least significant one. Fig. 5 shows how reducing the number of features affects the overall precision. At first, we pick the least important feature from the original 28 features by applying the above method and calculate the precision, and then we pick the least important feature from the rest of the 27 features until there is no more feature to remove.

From the graph, we can see that removing the first 13 features has almost no side effect to the overall precision, but removing more features results in gradually decreasing the precision value until the 24th feature. Removing the 25th feature and so on drastically reduce the precision which is almost unacceptable. But we showed that we can effectively reduce the number of features by almost 50% while keeping the overall precision by carefully examining the importance of the features.

IV. DEFECT IMAGE CLASSIFICATION USING A DEEP LEARNING TECHNIQUE

As Takano Co., Ltd. has a long history in visual inspection of film defects, the engineers have much knowledge about what kind of feature set is effective from experience, but this is not the case with most other companies. Determining the proper set of features is very hard, and the extraction of those values from defect images is another very hard task. This leads to the idea if the classification of film defects can be done without manually determine the features, and the deep learning technique comes into play.

Deep learning is a machine learning framework where the algorithm automatically captures features from training data. But for this to work, we need a much bigger amount of data compared to shallow machine learning algorithms. In most cases enough number of defect images can be obtained from daily manufacturing processes. We use deep convolutional neural networks and see how well film defects can be classified.

Figure 5. The effect of feature reduction to overall precision

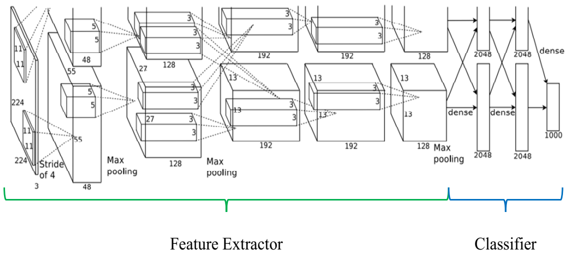


Figure 6. Typical deep convolutional network (AlexNet)[2]

Fig. 6 shows a typical structure of deep convolutional network [2], which consists of multiple layers of combinations of convolution layer and pooling layer and these play the role of image feature extraction and output 4096 dimension feature values. These values are then handled by the densely connected neural network which behaves classifier and this network handles 1000 classes as output. We modified the above network to set the input image size to 128x128 and changed the classifier part to handle our 5 categories of film defects. We then used the same dataset used for random forest classifier as described in Table 1 to train the network and achieved classification precision of 95.8% which is even better than using random forest.

We further sought to improve the overall accuracy by modifying the network. As the densely connected part works as a classifier, this part can be substituted by other classifiers. We experimented this modification by using SVM (Support Vector Machine) and random forest, and the best result was obtained when using random forest as the classifier. Table 4 shows detailed results for each class, and the overall precision was 97.1%. And its graphical representation is in Fig. 7.

As compared to Table 3, all values for class3 are improved implying that some kind of features were obtained by the convolutional neural network, and values for cass5 decreased slightly.

TABLE IV. DEEP CONVOLUTIONAL NETWORK RESULTS

Classes	Precision	Recall	F-measure
Class1	0.995	0.999	0.997
Class2	0.958	0.937	0.947
Class3	0.648	0.745	0.692
Class4	0.995	0.937	0.965
Class5	0.817	0.986	0.891
Overall	0.971	0.971	0.971

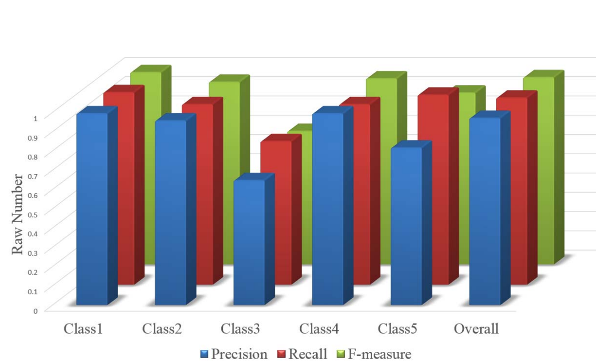


Figure 7. Deep convolutional network results in 3D graph

We tried to combine other techniques such as dimension reduction, singular value decomposition, principal component analysis, to seek to improve the classification accuracy, but these didn't work better than using random forest as the classifier.

V. CONCLUSION AND FUTURE WORKS

We applied both shallow machine learning and deep learning to classify film defects. Using random forest for classification resulted in overall precision of 94.7% while using deep convolutional network resulted in 95.8%, and both of them are far better than hand-crafted classifier. By using random forest as the classifier of the deep convolutional network, we achieved overall precision of 97.1% which is better than using neural networks for the classifier. This combination technique can also be used for other types of defect images.

Due to image characteristics, some type of defect images were hard to classify precisely, and we will continue to improve the classification precision using some other ideas which were not tested, including image data augmentation, modification of neural network structure and layer parameters and so on. It would be very interesting to see how newer types of deep neural networks as proposed in [3] and [4].

ACKNOWLEDGMENT

The authors thank Mr. Aoki, Mr. Usuda, Mr. Harada and other members of Takano Co., Ltd. for providing us various kinds of defect images, putting them classification labels and providing information how traditional classifier determine defects.

The authors also thank Akihiro Inoue, Shiho Akaike and Hiroto Iwata of the University of Yamanashi, for software development, deep neural network training, and for providing several ideas for extending the frameworks.

This work was partly supported by JSPS KAKENHI Grant Number 15K00167.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS, 2012.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, “Going Deeper with Convolutions”, CVPR 2015, http://arxiv.org/abs/1409.4842, 2015.

[4] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", http://arxiv.org/abs/1512.03385, 2015.

[5] L. Breiman, “Random Forests”, Machine Learning, 45, pp.5-32, 2001.

[6] T. Sharp, “Implementing Decision Trees and Forests on a GPU”, in ECCV, 4, Springer, 2008.

[7] D. Slat, “Random Forests for CUDA GPUs”, Master Thesis of Blekinge Institute of Technology, MSE-2010, 18, 2010.

[8] B. Waldvogel, “Accelerating Random Forests on CPUs and GPUs for Object-Class Image Segmentation”, Master's Thesis of Rheinische Friedrich Wilhelms Universitt Bonn, 2013.