Deep Learning for Image Pattern Recognition

Wei Chang, IT Division
Ramon Olavarria, IT Division
Joseph Gutierrez, IT Division

Abstract – Convolutional Neural Network (CNN), one type of emerging Deep Learning technique is very effective in detecting and classifying an image object. This approach unifies the steps of image feature extraction and object classification. The automatically learned features may provide valuable information that is not available through conventional image processing techniques. We demonstrated using EBEAM images that CNN with deep architecture can improved image classification performance from 68.5% to 81.1%.

I. Introduction

In the KT defect inspection products portfolio, algorithms working on the acquired images are used to search and characterize defects. When a defect image is available, the algorithm more often needs to seek a reference image to serve as a baseline to compare against. The reference image may come from adjacent die or simulated by mathematical modeling. However, the defect images themselves in many cases already contain rich information that has not been fully utilized by KT algorithms.

To the best of our knowledge, in the process of characterizing a defect based on acquired image, feature extraction in unavoidable. Yet a discriminating feature that separates one defect from another sometimes is hard to discover. It is well known that designing features for middle-level cues such as parallel lines, corners, line junctions and high level object representation is very difficult. Many KT engineers have spent significant amount of time to create complex algorithms to extract features. In addition, features that can characterize one set of defect images well may not work at all for another data set. During the engagement with customers, after customers provide data, KT algorithm engineers often need to develop customized features in a short period of time. Wouldn't it be great if an algorithm can generate useful features by itself, especially features that can represent mid-to-high level objects?

In this paper, we first introduce a technology, Convolutional Neural Network (CNN) that works very well on single image pattern recognition (w/o reference image) and can automatically learn features. We then use an EBEAM automatic defect classification problem as an example to demonstrate how CNN can significantly improves the performance over traditional feature based defect classification approach.

II. Convolutional Neural Network

A. *Deep Learning Architecture*

"Deep learning", rooted in neural network technology, is a probabilistic graph model with a lot of neuron layers, commonly known as a deep architecture. Yoshua [1] gave a good review on deep learning. "Deep learning" technology processes the information such as image, text, voice and so on in a hierarchical manner. As shown in figure 1, each node in one layer of the hierarchical probabilistic graph takes a linear combination of the inputs from nodes in the previous layer, then apply a nonlinearity to generate an output and pass it to nodes in the next layer. From bottom to top, the input pixel value was abstracted to local edge pattern to object part to final object concept.

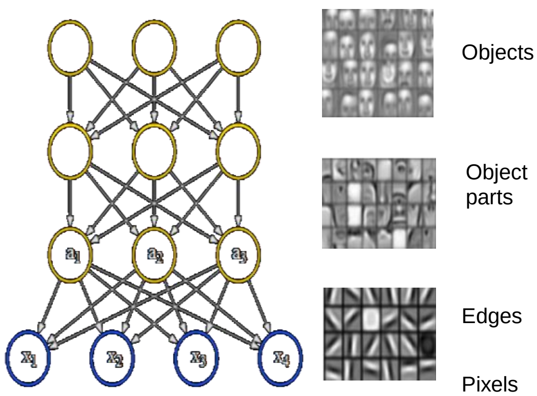


Fig1. Deep Learning Architecture

Depending on the probability specification and network architecture, a neural network with deep architecture has many variants, including Deep Belief Network (DBN), Restricted Boltzmann Machine (RBM), and Auto-Encoders. One particular type of deep neural network, convolutional neural network (CNN) works extremely well for image classification. We used a classic CNN architecture, LeNet5 [2] to explain the basic concepts of convolutional neural network. The actual implementation varies depending on the size of images, the number of images available and the nature of the problem.

B. Convolutional Neural Network

Convolutional Neural Network were inspired by visual systems structure. We know the visual cortex contains a complex arrangement of cells. These cells are sensitive to small sub-regions of the visual field, called a receptive field. As shown in figure 2, a small region in the input was processed by a neuron in the next layer. Those small regions are tiled up to cover the entire input images. To emulate the mechanism of the visual cortex, convolutional neural networks first convolve the input image with a small filter to generate feature maps (each pixel on the feature map is a neuron which corresponds to a receptive field). Then, a subsampling layer computes the max or average over each small window in the previous layer to reduce the size, and to obtain a small amount of shift invariance. The alternate between convolution and subsampling can be repeated multiple times. The final layer are fully connected traditional neural network.

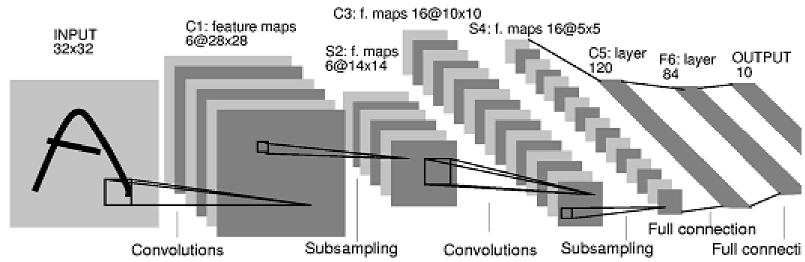


Fig 2. A Convolutional Neural Network (LeNET) [2]

C. Learning in CNN

Figure 3 is an over simplified explanation of how neural network with deep structure learned from data. Each learning epoch consists of forward inference and backward learning. Given the input data and weights that connect input layer and nodes on layer1, we can calculate the node values of layer1. We use the weights that link layer1 to layer2 to calculate node values of layer 2, then layer 3, and so on until we generate predicted outcome $h(g(f(X)))$. This completes the process of forward inference. During backward learning, the loss is first computed by comparing the predicted value and actual observation $L(h(g(f(x)), o)$. Then gradients are calculated to update all weight to reduce the learning loss. Thanks to the chain rule, it is sufficient to calculate the gradients of a node with respect to the previous level. Therefore, the entire gradient descent process can be conducted in a top down manner. Note, many state of art optimization techniques such as minibatch, early stop, regularization on extreme weights and adaptive learning rate are needed to help find a stable and quasi global optimal solution.

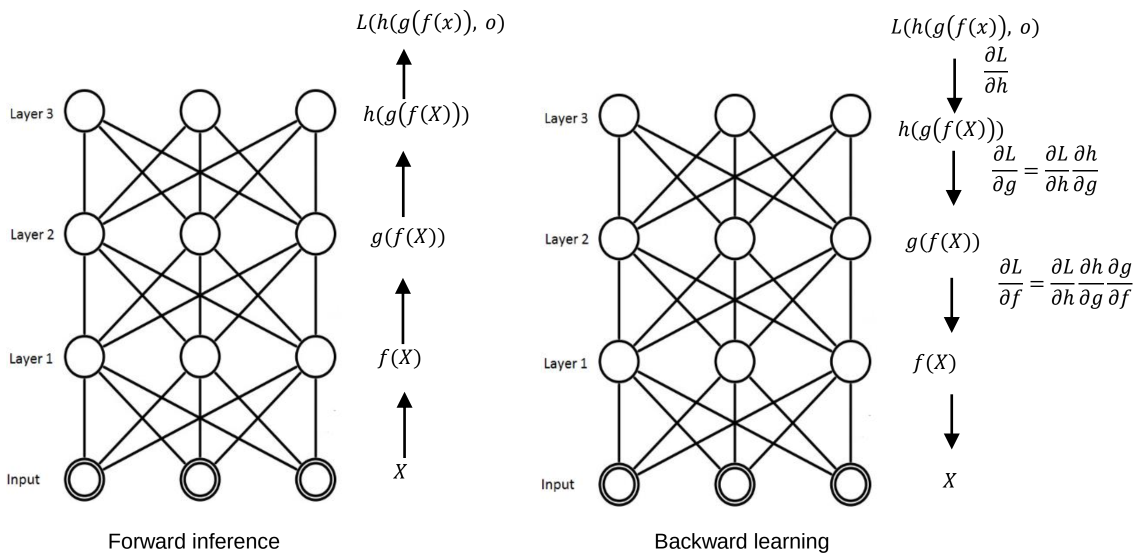


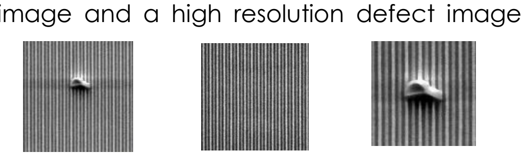
Fig 3. Learning in deep neural network

D. Choice of architecture

Different type of network architectures have been proposed. It is noteworthy that architecture complexity should be compatible with the size of the images. For 32*32 grayscale optical patch images, we recommend LeNet architecture that contains only 2 convolution layers. While for 480*480 high resolution defect review images, AlexNet [3] or GoogleNet [4] which consists of 7 and 11 convolution layers respectively are more suitable. Choosing the right convolution kernel size is tricky. The rule of thumb is to find the right level of "granularity" in order to create proper abstractions of object patterns, given a particular dataset.

III. Experiment Results

In the task of classifying defects, three types of images for each defect are available from KT EBEAM review tool: a defect image, a reference image and a high resolution defect image. An



Defect Image   Reference Image   High Resolution Defect Image

automatic defect classification algorithm is used to determine the type of defect as shown on the image. The current approach aligns the defect image and reference image, subtracts them, calculates features from the difference images and then output a suggested class code using a classifier such as Random Forest. In contrast, CNN takes only the high resolution defect image as the only input and automatically figures out feature extraction and transformation function from features to defect types all through the learning process.

One customer provided us with a data set that consists of 19 types of defects with total 7762 defects. The EBEAM division has already run the feature extraction with Random Forest classifier on the test data set with total 2976 defects. We use two different criteria to compare the CNN results with results provided by EBEAM.

In the first comparison, both approaches classify all defects in the test data set. We adopt two commonly used metrics purity and accuracy to benchmark the performance. Purity measures the confidence of the prediction and accuracy measures the percentage of defects in manually classified category included in the prediction. Table 1 summarizes the results and it clearly shows that CNN in most of defect categories outperforms existing approach on both purity and accuracy.

$$Purity = \frac{\#correctly\ classified}{\#\ total\ defects\ in\ prediction}$$

$$Accuracy = \frac{\#correctly\ classified}{\#\ total\ defects\ in\ manual\ classcode}$$

Class Code	Purity: Feature extraction +Random Forest	Purity: CNN	Accuracy: Feature extraction +Random Forest	Accuracy: CNN
13	0.57	0.69	0.69	0.79
16	0.27	0.44	0.31	0.55
26	0.82	0.9	0.57	0.95
30	0.93	0.83	0.6	0.85
37	0.87	0.94	0.79	0.96
42	0.9	0.95	0.95	0.99
49	0.95	0.94	0.76	0.84
51	0.82	0.87	0.77	0.88
75	0.71	0.79	0.5	0.67
76	0	0.75	0	0.75
77	0.27	0.68	0.5	0.72
85	0.77	0.92	0.88	0.96
86	0.73	0.96	0.64	0.78
99	0.78	0.85	0.93	0.95
Total Hit Rate	**77.60%**	**86.70%**		

Table 1: compare CNN with Random Forest

In the second comparision, the algorithm only classifies the defects with high confidence and leaves the remaining for operators to review. We calculate the total number of defects from categories with purity above 0.9. These defects will not be reviewed by operators. Tables 2 shows that CNN improves the total number of defects that do not need operators to review from 68.5% to 81.1%, which significantly save operators' time.

ClassCode	# defect that operators do not review: Feature extraction + Random Forest	# defect that operators do not review: CNN
13	21	87
16	1	4
26	24	41
30	42	53
37	272	325
42	96	96
49	290	329
51	156	188
75	63	122
76	0	3
77	1	21
85	338	493
86	33	81
99	700	572
Total	**2037 (68.5%)**	**2415 (81.1%)**

Table 2: comparison on total number of defects that do not need operators to review

IV. Conclusion and Discussions

This paper has demonstrated in a defect classification example that, working only on defect images Convolutional Neural Network outperforms existing approach that is based on feature extraction. The performance superiority may attribute to the features automatically learned by CNN. CNN unifies the steps of feature extraction and classification. Thus, the features extracted and selected in CNN may be optimized to differentiate different types of defects.

The application of CNN should not be limited to defect classification. Any problem that relates image patterns to certain characteristic can be tackled by CNN as long as there are enough training samples. In particular, when human can easily associate an image with certain characteristic using the naked eye and yet the existing coded algorithm fails, it is the time to give CNN a shot.

V. References

[1] Y. Bengio, Learning Deep Architectures for AI, 2009.

[2] Y. LeCun, L. Bottou, Y. Bengio and P. HaffnerR, "Gradient-Based Learning Applied to Document Recognition," in *Proceedings of the IEEE*, 1998.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," in *arXiv:1409.4842*, 2014.

Acknowledgement

1. Thank EBEAM division for providing the data set
2. Thank He LI (EBEAM), Hawren Fang & Danping Peng (RAPID), Scott Young & Kris Bhaskar (WIG) for the feedback during the discussion of CNN and defect classification.