A Novel Defect Classification Scheme based on Convolutional Autoencoder Skip Connection in Semiconductor Manufacturing

Jaegyeong Cha
Department of Smart Factory Convergence, Sungkyunkwan University
South Korea
sean9887@naver.com

Jongpil Jeong
Department of Smart Factory Convergence, Sungkyunkwan University
South Korea
jpjeong@skku.edu

ABSTRACT: *The semiconductor process cannot avoid defects due to its complex and diverse processes. In particular, wafer can be said to be the core of semiconductor manufacturing because they are directly related to the productivity of semiconductors. Therefore, detecting and classifying defects on wafers can help engineers address the root cause of defects and improve yield. In this paper, we propose a convolutional autoencoder using skip connection for wafer map defect classification. First, the encoder and decoder are designed by constructing a convolutional block. And connect the symmetrical blocks with skip connection. Finally, the training data of the classifier is encoded using the weights of the learned encoder. The loss of the model was successfully reduced with skip connection, and improved performance was obtained by reusing the encoder.*

Keywords: Wafer, Convolutional Autoencoder, Skip Connection, Semiconductor

Received: 8 July 2021, Revised 19 September 2021, Accepted 8 October 2021

DOI: 10.6025/dspaial/2022/1/1/42-49

1. Introduction

The semiconductor industry has developed significantly through continuous growth in the past. With advances in science and technology, the semiconductor industry required a high degree of flexibility and innovation to continuously respond to the rapid pace of change. However, even in the complex and sophisticated semiconductor process due to improved technology, the problem of defects was unavoidable. In particular, the defect detection of wafers, which draws semiconductor integrated circuits, is one of the major challenges faced by semiconductor manufacturing companies. However, the existing inspection method is passive depending on the inspector, which has drawbacks such as subjective problems, labor costs, and the influence of external factors. As wafer defect detection has a great effect on yield, wafer defect inspection has received considerable attention from researchers to improve and manage yield. In particular, researchers are using artificial intelligence technology to automate inspections and improve performance.

Deep learning is one of the artificial intelligence technologies that sets basic parameters for data and trains it to learn by itself by recognizing patterns using multiple processing layers. Deep learning is mainly used for image classification, speech recog-

nition, object detection, etc., and can achieve fast learning and high performance. Many studies have been conducted on the classification and detection of wafer defects using deep learning. Takeshi Nakazawa and Deepak V. Kulkarni presented a method for wafer map defect pattern classification and image retrieval using Convolutional Neural Networks (CNN) [1]. Yang Yuan-Fu used CNN and Extreme Gradient Boosting (XGBoost) for wafer-map defect pattern classification [2]. Jianbo Yu proposed an Enhanced Stacked Denoising Autoencoder to learn effective features [3]. Jaewoong Shim, Seokho Kang and Sungzoon Cho proposed a cost-effective wafer map pattern classification system based on CNN's active learning [4].

Deep learning has shown excellent performance through various studies, but because it learns based on data, its performance varies greatly depending on the quality of the data. However, in most wafers, the proportion of defective dies is much lower than that of normal dies, and the proportion of defects that occur according to the defect pattern is also not constant, so a data imbalance problem occurs. When the data used for training of the deep learning model is unbalanced, classification and detection of a small number of classes is not performed accurately because it is concentrated on a high proportion of classes. Also, overfitting may occur, increasing the complexity of the model.

In this paper, we propose a Convolutional Autoencoder (CAE) using skip connection to solve the above-mentioned problem. To improve the CAE performance, the architecture is deeply designed with skip connection, and data augmentation is performed by controlling the CAE to learn how to express data more efficiently. Then, the learned CAE encoder is recycled to encode the training data input to the classifier, and then the classifier is trained. By reusing the trained model, improved performance can be achieved without new models or techniques. To verify the performance of the proposed system, the WM-811K data set was used, and a comparison of the CAE of different depths and the general method was performed.

The main contributions of this paper are summarized as follows. First, we built CAE to solve the disproportionate problem of wafer map data, and adjusted the depth of the layer and used skip connection to obtain improved performance. Second, the performance of the proposed system was improved without additional methods by reusing the learned model.

The rest of the thesis is structured as follows. A related study is proposed in section 2. Proposed ideas are introduced in section 3, and experimental results and discussions are reported in section 4. Finally, conclusions and future research are presented in Section 5.

2. Related Work

2.1. Semiconductor Manufacturing

Semiconductor manufacturing consists of eight processes and is largely divided into pre-process and post-process. The pre-process is to design a semiconductor chip and engrave it on the wafer, and the post-process is to cut the chip engraved on the wafer and wrap it with an insulator to lay wires to receive power stably. In particular, the pre-process is also called a wafer process, and it is a step in which a single semiconductor chip is made by repeatedly forming and cutting various types of films on the wafer surface while composing an electronic circuit. Photolithography, a process to transfer and form a semiconductor circuit pattern on a wafer; Etching, a process to cut out parts other than the circuit pattern; Deposition, a process to form an

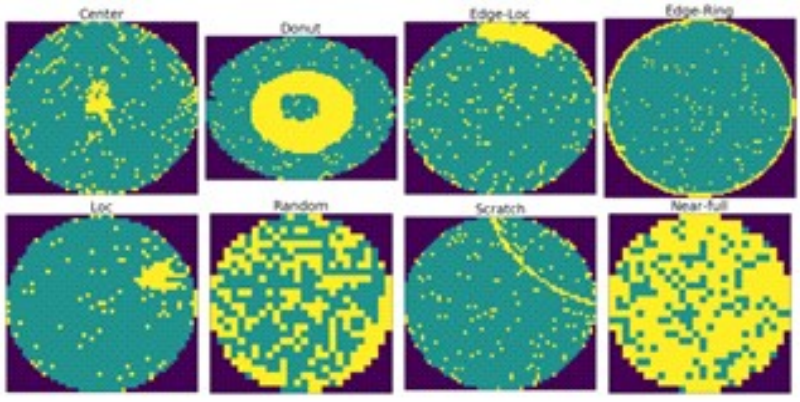

Figure 1. Types of defects

insulator thin film for separation and protection between metal and circuit; Metal to transmit electrical signals Various processes such as metallization, a process for forming wiring, are included in the previous process. Due to these various processes, the types of defects occurring on the wafer are also diverse. The data set used in this paper, WM-811K, is a large publicly available wafer map data set with 811,457 wafer maps collected from 46,393 lots. Wafer maps show the distribution of fail dies on the wafer, providing engineers with valuable information for root cause identification. The defect classes are composed of Center, Donut, Edge-Loc, Edge-Ring, Loc, Random, Scratch, and Near-full, and Figure 1 visually expresses the defects of WM-811K.

2.2. Convolutional Autoencoder

Autoencoder is a deep learning model proposed by Yoshua Bengio, Pascal Lamblin, Dan Popovici and Hugo Larochelle in 2007 [5]. Autoencoder consists of an encoder that converts input data into different values and a decoder that restores the original format. To put it simply, Autoencoder is to make the output output as equally as possible to the input through the structure of the encoder and decoder. In general, encoders reduce dimensionality, so they transform the input data into a latent vector to solve the problem. After all, the problem to be solved is to return the input value to the output value, so the latent vector contains information about the input value as if it were compressed as much as possible. The decoder has a symmetrical structure to the encoder and has the function of restoring the learned form from the latent vector. Autoencoder is a representative unsupervised learning network, and since it calculates the loss function through the difference between input and output, learning is possible without additional preprocessing of data.

Convolutional autoencoder is a model using CNN for the structure of autoencoder. CNN is a representative image classification network that extracts features and classifies images through convolution operation while maintaining spatial/regional information of input images. It is useful for recognizing images and finding patterns because it learns features by itself from data and uses patterns to classify images. Using the advantages of CNN as above, the convolutional autoencoder consists of an encoder as a convolutional layer and a decoder as a transposed convolutional layer. Information loss that occurs when using images as input data to the autoencoder can be prevented.

2.3. Skip Connection

After the advent of CNNs, studies on deep-structured CNNs have been started for better performance. It was expected that the performance would improve as the depth of the network increased, but rather the performance decreased. The reason for this is that the deeper the network, the harder it is to train, resulting in a vanishing gradient problem. To solve this problem, Kaiming he, Xiangyu Zhang, Shaoqing Ren and Jian Sun proposed a network using skip connection [6]. As shown in the figure below, skip connection performs the same operation differently from plain layer and then adds input x. By simply adding input x, the layer behind learns small information additionally instead of learning directly. That is, if the plain layer learns new information without preserving the previously learned information, skip connection connects the previously learned information so that the corresponding layer learns only the information to be additionally learned. Through this, the operation is simplified and learning becomes easier because only the difference value between the output of the previously learned layer and the output of the added layer needs to be learned.

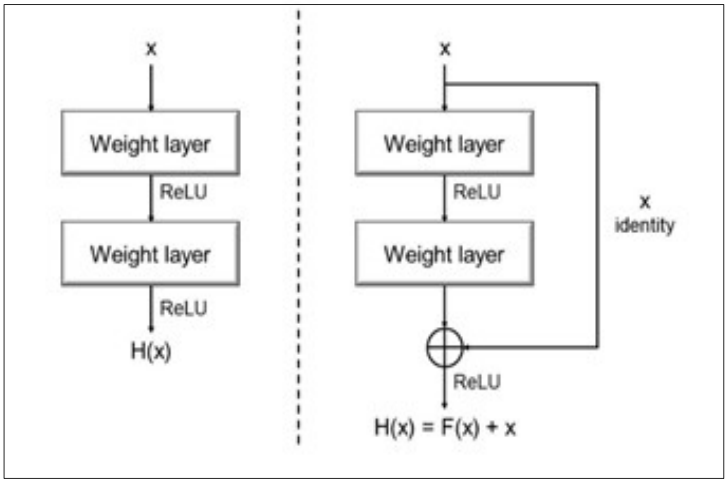

Figure 2. Plain layer and skip connection

3. Proposed Idea

3.1. System Model

In this paper, we propose CAE using skip connection (SCAE) to solve the problem of imbalance of semiconductor wafer data and improve defect classification performance. And we propose a method of encoding the input data of the classifier by reusing the learned encoder. Figure 3 shows the overall flow of the proposed model for better understanding.

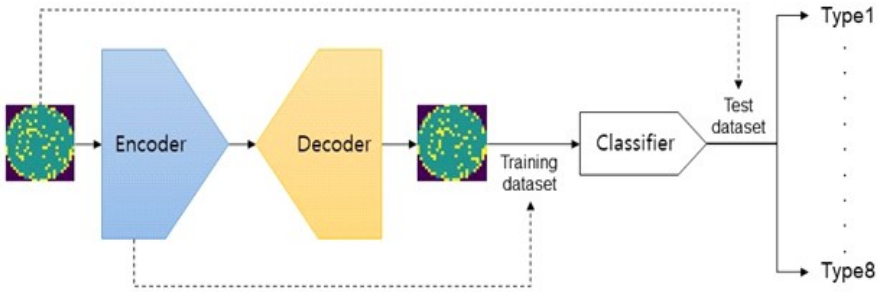

Figure 3. Overall structure of the SCAE

3.2. CAE with Skip Connection

Our proposed model consists of an encoder part and a decoder part. The encoder is composed of three convolutional blocks and plays a role in extracting the features of the input data. Here, the convolutional block has two 3x3 kernel size 2d convolutional layers, and batch normalization and ReLU follow each convolutional layer. A maxpooling layer is constructed between each convolutional block to prevent overfitting. The decoder has a symmetrical structure with the encoder, and the convolutional layer is replaced by the deconvolutional layer. The rest of the blocks are the same, and an upsampling layer is formed between each deconvolutional block to restore the image size reduced by the maxpooling layer. Figure 4 shows the structure of SCAE, and Figure 5 shows the structure of a convolutional block.

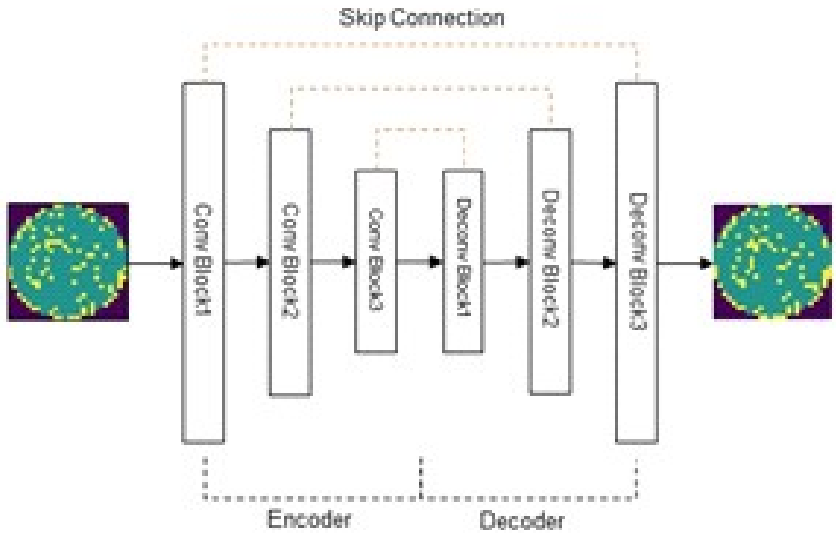

Figure 4. The structure of SCAE

As shown in Figure 3, when data is input to the convolutional layer, the feature f is extracted through the following.

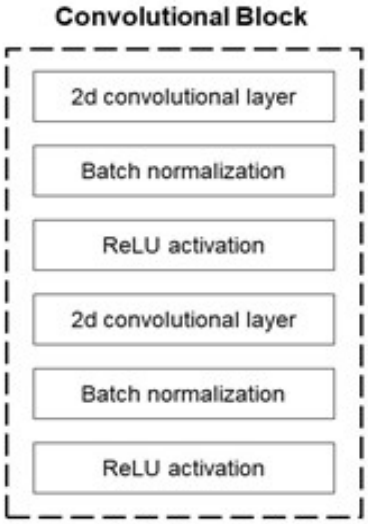

Figure 5. The structure of convolutional block

Here, X represents the input data, ωrepresents the weight of the convolution layer, and b represents the bias. The ReLU function [7] is a function that outputs 0 if the input value is less than 0, and outputs the input value as it is if it is greater than 0, and is expressed as follows.

$$f(X) = ReLU(\sum X * \omega + b) \tag{1}$$

Here, X represents the input data, ωrepresents the weight of the convolution layer, and b represents the bias. The ReLU function [7] is a function that outputs 0 if the input value is less than 0, and outputs the input value as it is if it is greater than 0, and is expressed as follows.

$$ReLU(x) = \begin{cases} (x < 0) & ReLU(x) = 0 \\ (x \geq 0) & ReLU(x) = x \end{cases} \tag{2}$$

$$= max\ (0, x) \tag{3}$$

To learn SCAE, batch size and epoch are set to 128 and 100, and the weight of SCAE is updated by adopting Adam optimizer. The loss function used Mean Square Error (MSE) [8] and is expressed as follows.

$$L_{MSE} = |X - X_d|^2$$

X_d means a decoded vector generated by mapping the features extracted through the encoder to the input space. It is calculated through the difference between the input and the reconstructed output and is also called Reconstruction Loss. Since the lower the loss means the smaller the difference between the output and the input, the model can be considered to have reconstructed the data close to the original.

3.3. Reusing Encoder Weights

In this paper, as shown in Figure 3, we want to reuse the encoder weights of SCAEs that have been trained. In autoencoders, encoders, also called recognition networks, are responsible for transforming inputs into internal representations. In the proposed method, SCAE, since the encoder is composed of a convolutional layer, it can be considered as an effective image feature extractor [9]. Therefore, we encode the data input to the classifier through encoder weights and then proceed with learning. Because a feature vector in which image information is preserved can be obtained through the encoder, defect classification is performed with a softmax classifier without using a classifier model with a complex structure [10]. Given the training data x, the feature vector h is extracted through the following formula.

$$h = F(x) \tag{5}$$

The extracted h is input to the softmax classifier, and the probability value is output through the following formula. Here, $S(x)$ means the softmax function.

$$S(h_i) = \frac{e^{h_i}}{\sum_j e^{h_j}} \tag{6}$$

4. Experiments and Result

All experiments in this paper were performed on a GTX 1080Ti GPU with Intel Core i7-8700K CPU, 12GB memory and 16GB RAM. The data set used in the study, WM-811K, has a ratio of labeled data and non-labeled data of 78:21, among which data with a defect pattern corresponds to 1/6.

The accuracy of the defect classification results of the wafer data was evaluated. Using True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN), the model defines the relationship between the expected answer and the actual answer, expressed as:

$$Accuracy = \frac{|TP|+|TN|}{|TP|+|FP|+|FN|+|TN|}$$

Figure 6 shows the loss graph according to the epoch of the proposed model, SCAE. To verify the performance of the proposed model, we experiment with CAE that does not use skip connection and change the depth of the model. SCAE-6 consists of 6 convolutional blocks and SCAE-8 consists of 8 convolutional blocks. Here, the convolutional block means that it includes both encoder and decoder blocks.

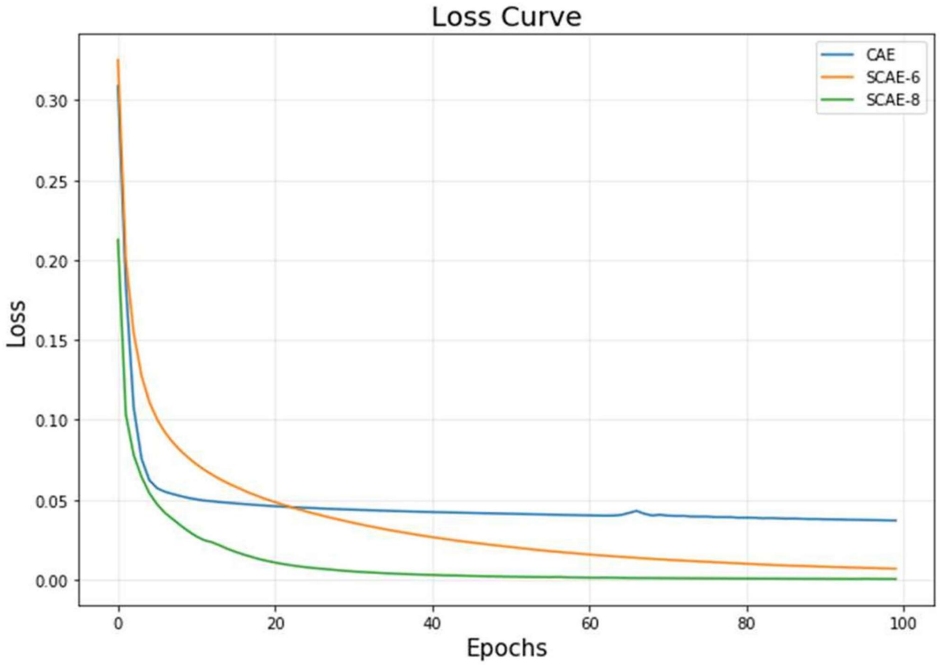

Figure 6. The loss graph

Looking at Figure 6, it can be seen that the loss value is lower when skip connection is used. And as a result of comparing the design depth of the model differently, better performance can be confirmed when 8 convolutional blocks are used. A low loss value in an autoencoder-based model means that the encoder extracts important features from the input data. Therefore, the output value generated by the decoder is as close to the input value as possible, which means that high-quality data can be

secured in terms of data augmentation. On the other hand, since the Loss value converges to almost 0, the model was not designed deeper, but the proposed model can be designed deeper when using field data with noise.

Table 1 shows the result of data augmentation through SCAE. All fault classes consisted of about 3000, which was able to solve the data imbalance problem.

Defect Class	**Number**
Center	3140
Donut	3002
Edge-Loc	3440
Edge-Ring	3061
Loc	3455
Scratch	3032
Random	3144
Near-full	3104

Table 1. Result of Data Augmentation

The newly generated data is used as training data and validation data of the classifier in a ratio of 8:2, and the defect classification result is obtained using the raw data as test data. In this study, to verify the proposed methodology, an experiment was conducted by applying the above three models (CAE, SCAE-6, SCAE-8) whether or not the learned encoder is reused. The accuracy of the test set was high in the order of SCAE-8, SCAE-6, and CAE, and when the learned encoder was reused, the accuracy was slightly increased. Figure 7 shows the defect classification results as an Accuracy graph.

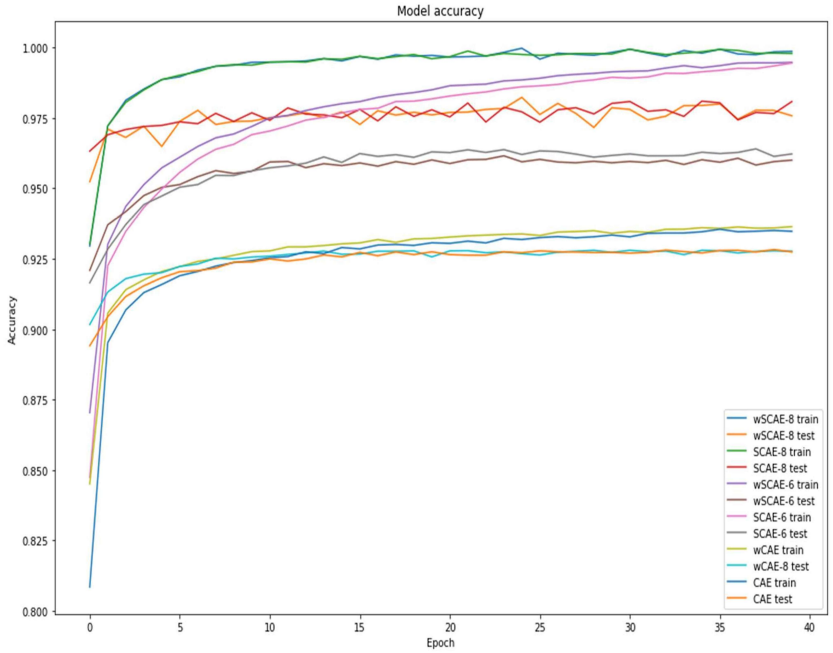

Figure 7. The Accuracy graph

5. Conclusion

In this paper, CAE using skip connection is proposed for semiconductor wafer map defect classification. The proposed model was able to secure high-quality data with a small loss value by adjusting the depth of the model due to skip connection. In addition, the accuracy of wafer defect classification could be improved by reusing the learned encoder to encode the input data of the classifier. The proposed model was divided into SCAE-6 and SCAE-8 according to the number of convolutional blocks, and the performance was verified by comparing the results when using CAE alone. This study used the open data set WM-811K, but if real field data with noise are available, models of various depths can be compared.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-2018-0- 01417) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

References

[1] Nakazawa., Takeshi., Kulkarni, Deepak V. (2018). Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing,* 31 (2) 309-314.

[2] Yuan-Fu., Yang. (2019). A deep learning model for identification of defect patterns in semiconductor wafer map. In 2019 30th *Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 1-6.

[3] Yu, Jianbo. (2019). Enhanced stacked denoising autoencoder-based feature learning for recognition of wafer map defects. *IEEE Transactions on Semiconductor Manufacturing* 32 (4) 613-624.

[4] Shim., Jaewoong., Kang, Seokho., Cho, Sungzoon. (2020). Active learning of convolutional neural network for cost-effective wafer map pattern classification. *IEEE Transactions on Semiconductor Manufacturing,* 33 (2) 258-266.

[5] Bengio., Yoshua., Lamblin, Pascal., Popovici, Dan., Larochelle, Hugo. (2007). Greedy layer-wise training of deep networks. *In*: *Advances in neural information processing systems*, 153-160.

[6] He, Kaiming., Zhang, Xiangyu., Ren, Shaoqing., Sun, Jian. (2016). Deep residual learning for image recognition. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.

[7] Agarap, Abien Fred. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

[8] Allen, David M. (1971). Mean square error of prediction as a criterion for selecting variables." Technometrics 13, no. 3, 469-475.

[9] Amaral., Telmo., Kandaswamy, Chetak., Silva, Luís M., Alexandre, Luís A., Marques De Sa, Joaquim., Santos, Jorge M. (2014). Improving performance on problems with few labelled data by reusing stacked auto-encoders. *In*: 2014 13th International Conference on Machine Learning and Applications, 367-372.

[10] Liao., Bin., Xu, Jungang., Lv, Jintao., Zhou, Shilong. (2015). An image retrieval method for binary images based on DBN and softmax classifier. *IETE Technical Review,* 32 (4) 294-303.