

Research on Defect Recognition of Lithium Battery Pole Piece Based on Deep Learning

Li Jiwei ¹, Li Linsheng ^{2,*}, and Xu Changlu ³

¹Shanghai DianJi University, Shanghai, 201306, China

²Shanghai DianJi University, Shanghai, 201306, China

³Shanghai DianJi University, Shanghai, 201306, China

Abstract. In the field of defect recognition, deep learning technology has the advantages of strong generalization and high accuracy compared with mainstream machine learning technology. This paper proposes a deep learning network model, which first processes the self-made 3,600 data sets, and then sends them to the built convolutional neural network model for training. The final result can effectively identify the three defects of lithium battery pole pieces. The accuracy rate is 92%. Compared with the structure of the AlexNet model, the model proposed in this paper has higher accuracy.

1 Introduction

Driven by changes in energy technology and emerging technologies, lithium batteries are the first to be widely used in 3C digital fields such as mobile phones and laptops due to their high energy density, long cycle life, zero pollution, and small size. With the rapid development of lithium batteries, the market and users' requirements for the performance and safety of lithium batteries are constantly improving. In particular, the potential safety hazards of lithium batteries have always existed [1]. From mobile phones, laptops to electric vehicles, lithium batteries have been heated or even caught fire. A series of strict tests must be carried out during the production process of lithium batteries or before they leave the factory. As the key component of lithium battery, the quality of pole piece directly affects the electrochemical performance and service life of lithium battery.

In the production and inspection of lithium battery pole pieces, the current industrial inspection methods include manual inspection and the use of machine vision technology [2]. Due to the shortcomings of low efficiency and low accuracy [3], manual detection is no longer used in most factories. The mainstream machine vision technology must be hand-made to adapt to a specific field [4]. It has long model development cycles and poor adaptability, and cannot quickly adapt to new products. In the past, classic machine vision methods were very effective in solving these problems [5]; however, with the development of Industry 4.0, the production line is moving towards a general trend, requiring machine vision technology to quickly adapt to new products [6]. Traditional machine vision methods cannot guarantee such flexibility.

With the support of massive data and huge computing power, the emerging deep learning technology can automatically extract features and achieve accurate classification of pictures based on the extracted features. Deep learning technology has the advantages of high accuracy and strong adaptability, and will have broader application prospects in the field of defect recognition [7].

This paper proposes a method for identifying defects in lithium battery pole pieces. First make a self-made data set, combined with image processing technology to reduce the noise of the data set [8]. Then build a deep learning network model structure, and send the data set to the built convolutional neural network for iterative training. Build a classic AlexNet [9] structure neural network, send it to the same data set for training, and compare the accuracy of the results.

2 MATERIALS AND METHODS

2.1. Data set production and processing

2.1.1 Data set production

There is no public lithium battery pole piece defect data set available on the Internet, and a self-made data set is required. This article focuses on three common defects such as scratches, breakages and dark spots, as shown in Figure 1. Intercept the defect area of the picture and make a neural network data set.

On the factory production line, industrial cameras are used to take photos of defective lithium battery pole pieces. Then intercept the defect area of each picture, classify it according to the defect type, and make a neural network data set.

* Corresponding author: 1545928587@qq.com

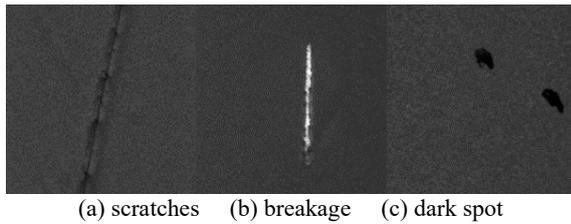


Fig.1 Lithium battery pole piece defect

According to the appearance of the defect, manually intercept and intercept the pictures of the data set. The picture size and resolution are different. Through image processing, the intercepted 200 scratched pictures, 200 damaged pictures, and 200 dark spot pictures are processed from pictures with different sizes of pixels into pictures with 128*128 pixels and a size of about 2KB.

There are 8 ways of data enhancement:

- a) Flip Transform; b) Random Crop;
- c) Color Jittering; d) Translation Transform;
- e) Scale Transform; f) Contrast Transform;
- g) Noise Disturbance; h) Reflection Transform.

Combining these 8 methods to enhance the data set, the enhanced results are shown in Table 1.

Table 1. The number of data sets

defect	scratches	breakages	dark spots	total
old quantity	200	200	200	600
new quantity	1200	1400	1000	3600

Number and name each picture in a uniform order. If necessary, the training set and the test set must be divided in a ratio of 3 to 1 and stored in two folders separately. It can also be set to divide automatically according to the proportion. During training, the program divides the training set and test set by itself.

2.1.2 Image preprocessing

In the image preprocessing stage, combined with image processing technology, the data set used for training is processed through gray-scale conversion and optimal threshold segmentation to remove irrelevant interference and show specific defects in the picture clearly, as shown in Figure 2.

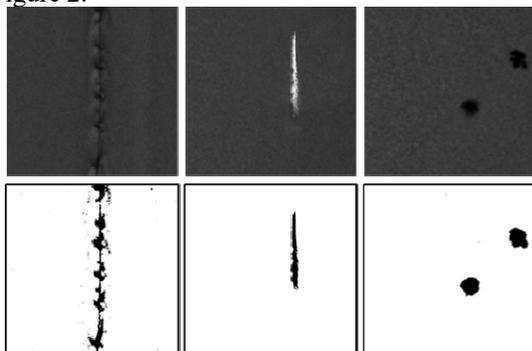


Fig. 2 Sample image after image preprocessing

The information contained in the images of the data set is cumbersome and disturbing, which will affect the calculation results and reduce the classification accuracy. As shown in Figure 3(a), the unprocessed original image has a lot of noise outside the dark spot feature area. If it is not processed, these noise points will also be sent to the neural network. This will increase the processing content of the neural network, bring greater load to the network, and the small neural network may even collapse. Preprocessing the data set not only reduces the calculation time of the network model, but also improves the accuracy of neural network recognition. As shown in Figure 3(b), the preprocessed data set image removes the irrelevant interference information around the two dark spots and retains the complete defect contour information; then the image is sent to the convolutional neural network for further processing.

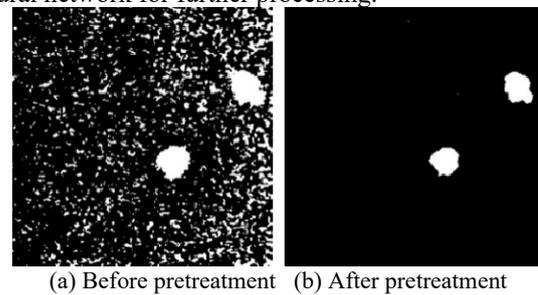


Fig. 3 Comparison of unprocessed and preprocessed data set pictures

The method of image preprocessing is threshold segmentation. The basic principle of the threshold segmentation method is: by setting different gray thresholds, the image pixels with similar gray thresholds are classified into one category, thereby dividing the image pixels into several categories. The main point of threshold segmentation is to find suitable threshold data as the segmentation point. In this paper, the dynamic threshold method is used to change the threshold in real time, and the dynamic selection is made according to the different segmentation effects of different thresholds on the image. After testing multiple sets of different thresholds, select the optimal threshold data according to the dynamically displayed processing effect.

Threshold segmentation processing can make the characteristics of the data set obvious, which is conducive to the subsequent convolutional neural network processing.

2.2. Deep learning network model construction

The deep learning network structure built in this article is based on the convolutional neural network. Its basic structure is: convolutional layer, batch normalization layer, activation layer, pooling layer, two fully connected layers, and the last softmax layer, such as Shown in Figure 4.

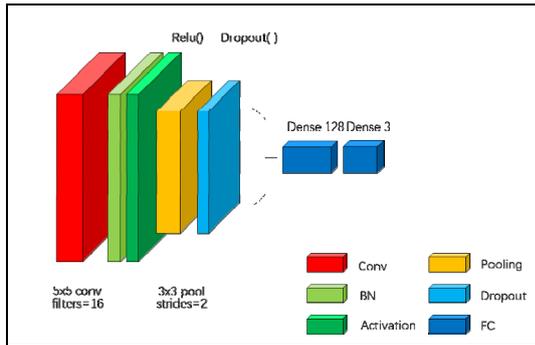


Fig. 4 Deep learning network structure model

The first convolution layer uses 16 5*5 convolution kernels (3 channels), and the convolution kernel weights category is set to Variable. The advantage of using variables is: generate a tensor of the Tensorflow framework, which can be modified as needed in the subsequent graph operations. Using all zero padding means that the edge of the image is filled with a circle of 0, so that the image size after convolution is consistent with the input size.

The weights value of the initial convolution kernel outputs random values from the truncated normal distribution. The dimension shape of the generated tensor is [3, 3, 3, 128], the standard deviation stddev is 1.0, and the tensor type is 32-bit floating point. The category of biases is set to Variable. The initial offset value uses constant constant, the constant value is 0.1, and the type is 32-bit floating point.

The full name of the BN layer is Batch Normalization, which aims to solve the problem of changes in the data distribution of the middle layer during the training process, to prevent the gradient from disappearing or exploding, and to speed up the training. The essential principle of BN is that when each layer of the network is input, another normalization layer is inserted, and then it enters the next layer of the network.

The role of the BN layer:

- 1) Speed up the training speed, so that we can use a larger learning rate to train the network.
- 2) Improve the generalization ability of the network.
- 3) The BN layer is essentially a normalized network layer, which can replace the local response normalization layer (LRN layer).
- 4) The sample training sequence can be disrupted, so that it is impossible for the same photo to be selected for training multiple times.

In the activation layer, the biases are added to the convolutional layer (conv), and each row of the tensor biases is added to the tensor conv, and the resultant dimension is the same as the conv tensor. The activation function uses the relu function, which is shown in Figure 5. Compared with the sigmod function and the tanh function, the Relu function has the following two advantages: 1) Overcoming the problem of vanishing gradient. 2) Speeding up training.

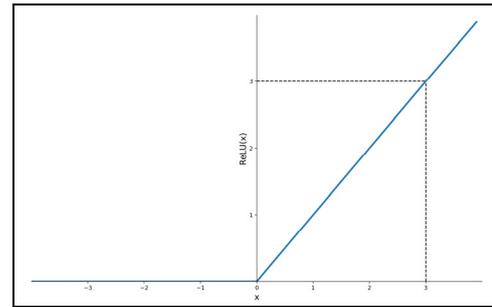


Fig. 5 Relu function curve

The pooling method of the pooling layer selects 3*3 maximum pooling, and the strides is set to 2. Then perform a partial response normalization operation (IRN). The pooling window ksize is [1, 3, 3, 1]; the strides are [1, 2, 2, 1], and the completion method uses all zero padding.

The maximum pooling operation is to calculate the maximum value of a specific feature in a region of the image. Pooling can reduce the dimensionality of the image (compared to using all the extracted features) and improve the result (less overfitting). Pooling can also help the input data to represent approximate invariance.

The discarding layer is to temporarily discard the neural network unit from the network according to a certain probability in the process of training the deep learning network. This method can reduce the interaction between neurons, which means that some neurons rely on other neurons to function, thereby reducing overfitting.

The Flatten layer is used to "flatten" the input, meaning that the multi-dimensional input is one-dimensional, and is used for the transition from the convolutional layer to the fully connected layer.

The first fully connected layer uses 128 neurons, and the activation function uses relu. Then add another layer of Dropout. The second fully connected layer uses 3 neurons, and the softmax activation function is used to classify the three results.

2.3. AlexNet model building

In order to test the effect of the neural network model proposed in this article, the existing public excellent network model structure AlexNet is used to compare with the model proposed in this article. The AlexNet network model is shown in Figure 6.

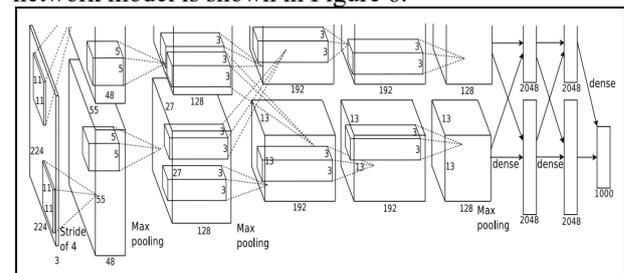


Fig. 6 Structure diagram of AlexNet model

There are 8 layers in AlexNet. The first layer uses 96 3*3 convolution kernels, with a step size of 1, and no

zero padding; using mainstream BN operations to achieve feature standardization; using relu activation function, using 3*3 pooling cores, with a step size of 2. Do max pooling without using Dropout.

The second layer uses 256 3*3 convolution kernels, with a step size of 1, and no zero padding; using BN operation; using relu activation function; using 3*3 pooling kernels with a step size of 2, making the maximum pooling Transformation; Dropout is not used.

The third layer uses 384 3*3 convolution kernels with a step size of 1, using all zero padding; no BN operation; relu activation function; no pooling and dropout.

The fourth layer is consistent with the third layer.

The fifth layer uses 256 3*3 convolution kernels, with a step size of 1, using all zero padding; no BN operation; using relu activation function; using 3*3 pooling kernels, with a step size of 2; doing maximum pooling; Do not use Dropout. Layers 6, 7, and 8 are fully connected layers.

The 6th and 7th layers use 2048 neurons, relu activation function, and 50% dropout. The eighth layer uses 3 neurons, and uses softmax to make the output conform to the probability distribution.

3 EXPERIMENTAL RESULT

Put the prepared data set into the convolutional neural network training, set the number of feeds (batch) to the network at a time to 16, the number of iterations (epoch) to 15, and add restrictions: the training is stopped when the loss value is less than 0.2.

The data set is sent to the AlexNet model for training, the batch is set to 16, the epoch is set to 15, and the restriction is added: the training is stopped when the loss value is less than 0.2. Compare the training results of the two models, the final training result is shown in Figure 7.

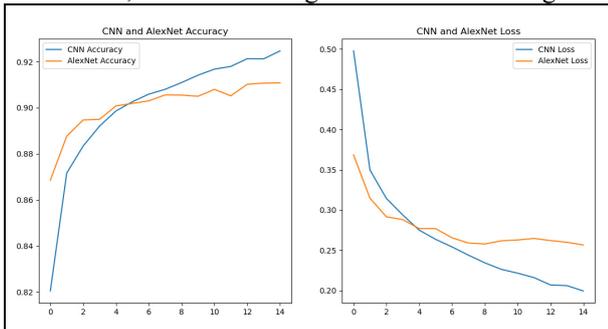


Fig. 7 Comparison of training results

In Figure 7, CNN is a deep learning network structure model based on a convolutional neural network structure, and AlexNet is a comparison model. According to the results, before the fourth iteration, the accuracy of the AlexNet model is higher than that of the CNN model, but the accuracy of both is lower than 90%. As the number of iterations increases, the accuracy of the CNN model gradually rises, surpassing the AlexNet model. In the end, the accuracy of the AlexNet model is 91%, and the accuracy of the CNN model is 92%.

The loss value loss results show that, before the fourth iteration, the loss value of the AlexNet model is lower than that of the CNN model. After the fourth iteration, the loss value of the CNN model is lower than that of the AlexNet model. The final AlexNet model loss value is about 0.25, and the CNN model loss value is 0.20. Combining the two results of accuracy and loss value, it shows that the deep learning network framework proposed in this paper is better than the classic AlexNet network model.

4 CONCLUSION

This paper proposes a deep learning network model for defect recognition of lithium battery pole pieces. First, the lithium battery pole piece defect data set is produced, and then the data set is enhanced to expand the number of data sets. The data set is processed for image noise reduction to reduce the computational load of the subsequent network model. A deep learning network model based on convolutional neural network is built. In order to test the effect of the model, a classic AlexNet network model was built. The data set is sent to two models for training, and the experimental results of the two models are compared. The experimental results show that the deep learning network model proposed in this paper has higher accuracy (92%) and lower loss value (0.20), which is better than the AlexNet network model.

The deep learning network model proposed in this paper can effectively identify three kinds of lithium battery pole piece defects, and has flexible scalability, and can be applied to the field of industrial defect recognition. In the subsequent work, the deep learning network model will be further improved, the recognition accuracy will be improved, and more model application scenarios will be expanded.

References

1. Liang, Y.H., Fan, L.Z. (2020) Mechanical failures in solid-state lithium batteries and their solutions. *Acta Physica Sinica*, 69(22): 17-32.
2. Han, X.X., Geng, S.Y., Li, H.Y. (2019) Summary of Defect Detection Application Based on Machine Vision. *Electric Engineering*, 14: 117-118+132.
3. Xiao, Y.J., Qi., H., Zhou, W. (2019) Detection and Identification of Roll Surface Defects in Lithium Battery Pole Piece Rolling Mill. *Journal of Electronic Measurement and Instrumentation*, 10: 148-156.
4. Oztemel, E., Gursev, S. (2020) Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, 31: 127-182.
5. Bulnes, F.G., Usamentiaga, R., Garcia D.F., et al. (2016) An efficient method for defect detection during the manufacturing of web materials. *Journal of Intelligent Manufacturing*, 27(2): 431-445.

6. Masci, J., Meier, U., Cireşan, D. (2012) Steel defect classification with Max-Pooling Convolutional Neural Networks. In: The 2012 International Joint Conference on Neural Networks (IJCNN). IEEE. Brisbane.
7. Sarigül, M., Ozyildirim B.M., Avci, M. (2019) Differential convolutional neural network. *Neural Networks*, 116:279-287.
8. Wu, Y.Q., Meng, T.L., Wu, S.H. (2015) Research Progress of Image Threshold Segmentation Method for 20 Years (1994—2014). *Journal of Data Acquisition and Processing*, 30(01): 1-23.
9. Krizhevsky, A., Sutskever, I., Hinton, G. (2012) ImageNet Classification with Deep Convolutional Neural Networks. In: NIPS. Curran Associates Inc.