International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Impact Factor (2022): 7.741

Image Defect Detection of Solar Cell Based on Optimized GoogLeNet

Bowen Shao[1]

[1]North China Electric Power University, School of Control and Computer Engineering
No. 2 Beinong Road, Changping District, Beijing, China
18810116073[at]163.com

Abstract: ***In order to deal with the energy crisis, people focus on renewable energy. Solar energy is playing an increasingly important role. As the core component of photovoltaic power generation, the flaws of solar cells will lead to low utilization of solar energy. Therefore, it is very important to detect the defects of solar cells. In this paper, an optimized GoogLeNet model is used to detect the image defects of solar cells. By adjusting the network structure and optimizing the activation function, the accuracy of the model reached 95.92%.***

Keywords: solar cell, deep learning, defect detection, convolutional neural network

1. Introduction

After the 21st century, with the increasing development of human industrialization and civilization, the demand for energy continues to increase. It means that the use of fossil energy is increasing, and the reserve is gradually decreasing. The development and utilization of new energy has become an urgent problem facing mankind. To deal with the energy crisis, people turn their attention to clean energy such as solar energy, wind energy and nuclear energy. Among them, Solar energy is inexhaustible and don't pollute the environment. So it has received increasing attention. Solar cells have also been widely used as carriers of solar power generation.

Solar cells are prone to produce defects in the production process, such as hidden crack, scratch, grid break, black spots, etc. These defects can cause localized heating of the solar panel and impaired current transfer, affecting the lifespan and power generation efficiency of the solar panel. The detection methods of the solar panel defect are mainly divided into artificial visual detection, physical detection and machine vision detection.

Artificial visual detection is one of the most primitive detection methods, which determines whether there is a defect through the solar panel image taken by visible light camera or infrared camera. It is costly and inefficient.

The physical detection methods mainly include sound waves, visible light, lasers, etc. The representative works of physical methods mainly include: Tsuzuki et al. [1] used sound waves to detect cracks. This method made the solar cells generate appropriate resonance and then generates sound waves, and then analyzed the sound waves. The difference between the frequency and the standard frequency (normal solar cell) to judge whether there were cracks on its surface. Esquivel [2] used contrast-enhancing illumination to detect cracks. This method increased the light intensity on the surface of the solar cell so that cracks and flaws could be reflected to another plane. If image distortion occurs, the existence of flaws could be judged. Chen et al. [4] carried out detection based on the noise method. This method utilized the properties of low-frequency noise and reliability of crystalline silicon solar cells, and judged whether there was a defect by comparing the difference between the noise of defective solar cells and its non-defective noise.

The general process of image defect detection of solar panels through machine vision is as follows: The solar cell is imaged, and the pixel distribution information and features in the image are used for defect detection. According to the different mathematical modeling methods, it can be roughly divided into detection method based on gradient feature, detection method based on matrix decomposition and detection method based on machine learning. The representative work of detection methods based on machine vision is as follows: Anwar et al. [5], Tsai et al. [6] and Bakalexis et al. [7] used gradient features of images to sharpen and smooth different gradient regions in images. As the gradient value of defect edge area was high, the defect area was sharpened first, and the non-defect area with low gradient value was smoothed. Lu et al. [8] proposed a surface defect detection method based on matrix singular value decomposition. This method didn't use any feature extraction algorithm to extract image features. It directly took the original data of the image as a matrix, and then performed singular value decomposition on the matrix. Lu et al. [8] believed that different singular values represented different degrees of detail in images. By selecting singular values that can represent background texture to reconstruct the matrix, an image containing only background could be obtained, and the defect area in the image could be obtained by differentiating the reconstructed image from the original image Demant et al. [9] used support vector machine (SVM) to detect surface defects of solar cells. The method obtained a group of cracked and non-cracked samples, then extracts the features of these samples through a group of local descriptors. The obtained feature vectors represent samples sent to SVM for training. What's more, Crack samples have a varying crack width, which makes the set of crack samples more representative. In the detection of defects in solar cells, machine learning has become one of the main development directions due to its convenience, speed, and economics.

In this paper, we propose a method for detecting image defects in solar cells based on optimized GoogleNet. By optimizing the GoogLeNet network structure and adjusting the network activation function, the training speed and accuracy of the

model are improved, which provides a new approach to defect detection in solar cell images.

2. Method

GoogleNet is a brand new deep learning structure that was proposed by Christian Szegedy in 2014. Before this, AlexNet, VGG, and other structures increased the depth of the network (layers) to achieve better training effects. Nevertheless, the accumulation of layers will cause many detrimental effects, including over fitting, gradient disappearance, and gradient explosion. GoogLeNet introduces inception units into the network architecture to improve the overall performance of the model.

2.1 Inception Moudle

Inception is a Network in Network architecture. With Inception in the original node, the width and depth of the network structure can be expanded. This results in improved performance. The main idea of Inception is to use convolutional kernels of different sizes to realize perception at different scales. The network structure diagram is shown in figure 1.

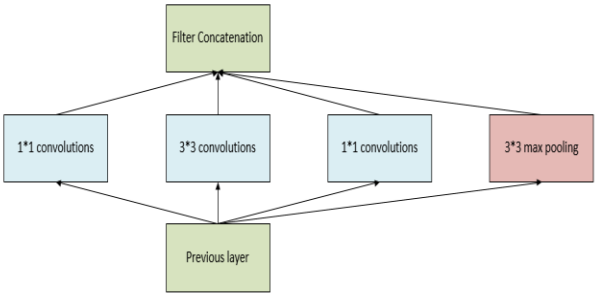

Figure 1: Inception v1

Inception module has four basic components: 1×1 convolutional layer, 3×3 convolutional layer, 5×5 convolutional layer, and 3×3 max pool. The results of the four components are combined on the channel. This is the core idea of Naive Inception: different convolutional kernels of different sizes are used to achieve different scales of perception, and the final fusion results in a better representation of the image.

However, Inception V1 has two very serious problems. Firstly, all convolutional layers are directly interfaced with the data input from the previous layer, so the computation in the convolutional layer will be heavy; Secondly, the maximum pooling layer used in this unit retains the depth of the feature graph of the input data, so the depth of the total output feature graph will only increase in the final merge, which increases the computation of the network structure after this unit. For example, for a 32×32×256 feature graph, 32 5×5×256 convolution kernels are used for convolution operation. To obtain the 32×32×32 feature graph with output 32×32×32, the total computation amount is 2.1×108 times, and the computation amount of convolution kernels is 2.78 times that of 3×3. In order to reduce the overall amount of computation, the Inception V1 structure is improved by catching asymmetric convolution to reduce the amount of computation, and at the same time, the data of 1×1 network dimension reduction convolution is used before the 3×3 and 5×5 convolution layers to reduce the amount of computation. The improved structure is shown in figure 2.

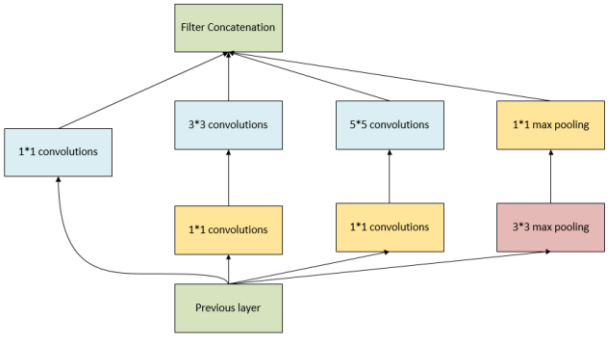

Figure 2: Inception optimized structure

2.2 GoogLeNet structure

GoogLeNet consists of an input layer, a five-layer convolution module and an output layer group layer, including 22 parameter layers and 5 pooling layers. The first and second groups of convolution blocks contain convolutional layers and max pooling layers. The third, fourth and fifth groups of convolutional modules are mainly composed of inception modules and maximum pooling layers. The output layer is composed of average pooling layers. It consists of a layer, a dropout, and a fully connected layer. The network structure diagram is shown in figure 3.

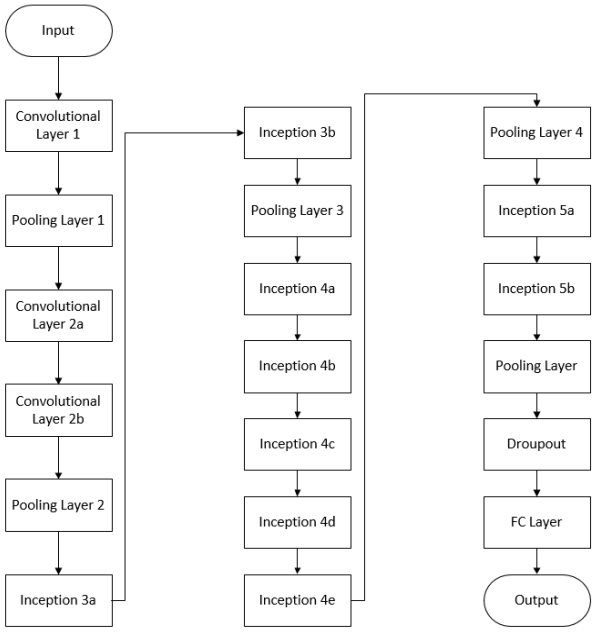

Figure 3: GoogLeNet structure

2.3 Optimized GoogLeNet

Although GoogLeNet has excellent performance and strong learning ability, it has a complex network structure and fewer parameters than VGG and AlexNet, but it also has 2.4×107 parameters. According to the characteristics of solar panel experimental image, we improve the GoogLeNet network. We reduce the original first 3 convolutional layers with different sizes to a 3×3 convolutional layer, and reduce the original 9

inception moudle to 6 Inception moudle. The dimensions of inception moudle remain unchanged. The number of convolutional cores is increased from 64 to 512, which makes inception structure more regular, conducive to hardware acceleration and program writing, and reduces code redundancy. The improved inception structure will reduce the number of features extracted from images, but greatly reduce the number of model parameters. Each pixel of the input feature image is affected by 1×1 convolution kernel, and the depth of the output feature image is reduced by setting the number of 1×1 covolution kernel less than the depth of the input feature image.The number of parameters and the amount of calculation are reduced, the convergence of the model is accelerated, and overfitting is reduced.

The GoogLeNet network removes the last full connection layer and replaces it with a classification layer that includes droupout layer and equal-pool layer. Droupout layer randomly inactivates some neurons and reduces the number of parameters to prevent mode overfitting. Global average pooling is to take the average value of the feature graph, use the value to represent the feature graph, and input the value into the softmax layer to get the corresponding probability distribution, so as to achieve the classification effect. Compared with the full connection layer, the number of parameters and calculation of global average pooling are less.

We further improve the accuracy of the model by adjusting the activation function. When selecting a function, it is necessary to ensure that its gradient is not equal to 0 at infinity, so as to achieve faster convergence of the activation function and alleviate the disappearance of gradient.

ReLu function is as follows:

$$f(x) = \max(0, x) \tag{1}$$

Equation (2) is the derivative of ReLu. When x≤0, the value of ReLu remains 0, and the gradient disappearance of ReLu occurs. Several experiments in [10] pointed out that with the increase of neural network layers; the gradient disappearance of ReLu model became more obvious.

$$f(x) = \begin{cases} 1, x>0 \\ 0, x \le 0 \end{cases} \tag{2}$$

We adopt the Swish function, which still has very good convergence performance as the network depth increases. The Swish function is shown in equation (3).

$$g(x) = x * sigmoid(\beta x), x \in (-\infty, +\infty) \tag{3}$$

When β=0, g(x)=x/2 . its output range is (−∞, +∞). When β→∞, g(x) becomes a function similar to ReLU, and its output is max(0, x) . When β=1, g(x)=x*sigmoid(x), which outputs in the range (−0.5, +∞). The default here is = 1. If u(x)=sigmoid(x), then g(x)=x*u(x) . Its derivative is:

When β=0, Swish becomes a linear function g(x)=x/2. When β=1, Swish function is approximately linear when x>0, approximately saturated when x<0, and has a certain non-monotonic property. When β→+∞, Swish function approximates ReLu function. The derivative of Swish is as follows:

$$g'(x) = x*u'(x) + u(x) \tag{4}$$

$$g'(x) = g(x) + u(x)(1\text{-}g(x)) \tag{5}$$

From equation (5), we can find that the derivative of Swish's function at infinity is not zero no matter whether x is positive or negative. Therefore, Swish function has a better convergence rate than ReLu.

3. Experiment

3.1 Experimental data and environment

The experiment used the ELPV-data-master data set provided by Deitsch, S[11]. The dataset contains 26,24 samples of 300x300 pixels 8-bit grayscale images of functional and defective solar cells of varying degrees of degradation from 44 different solar modules. We consider images with 0% defect probability as non-defective samples, and images with probability 33%, 67%, and 100% as defective samples.

All images are normalized with respect to size and perspective. Additionally, any distortion induced by the camera lens used to capture the EL images was eliminated prior to solar cell extraction. As shown in Figure 4, the processed image is arranged in order of defect probability from small to large.

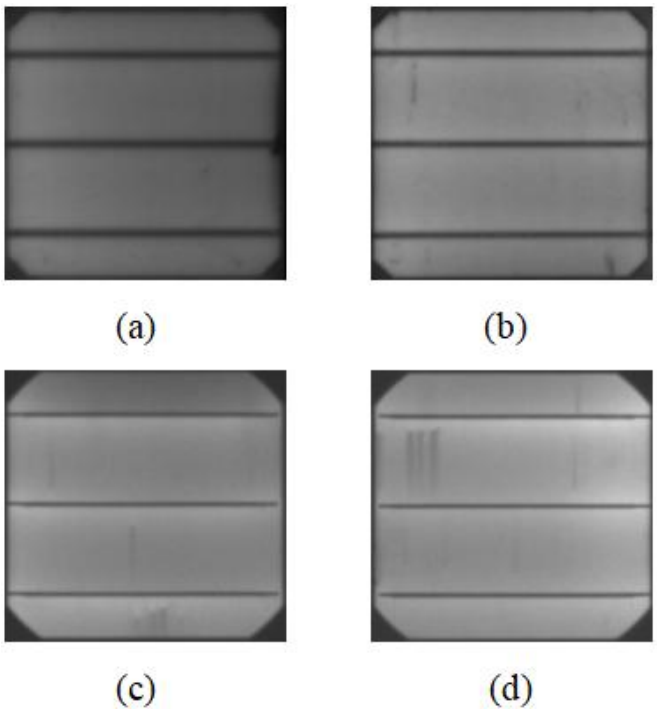

Figure 4: solar cell imagine

The experiment was carried out on the TensorFlow platform of the Windows 10 operating system, and GPU acceleration training was implemented by NVIDIA GTX1060.

3.2 Analysis of experimental results

80% of the elpv-data-master data set was used as the training set, and 20% as the test set. We completed training and testing by feeding data into optimized GoogLeNet, VGG16, CNN and LeNet-5. Accuracy rate (ACC) and F1-Score were used to evaluate the GoogLeNet model. ACC is the overall judgment accuracy of the model, and F1-Score is a judgment indicator that integrates accuracy and recall rate. It ranges from 0 to 1, with 1 being the highest score and 0 the lowest score.The calculations are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \tag{7}$$

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP} \tag{8}$$

$$F1 - Score = \frac{2 * \mathrm{Re}call * \mathrm{Pr}ecision}{\mathrm{Re}call + \mathrm{Pr}ecision} \tag{9}$$

Where TP, TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative, respectively.

After 170 generations of training, the loss function changed stably and the model converged. Table 1 shows the confusion matrix of classification results. The accuracy of the model was 95.92% and f1-score was 0.953. The results of each model are shown in Table 2. According to this index, our method has a better effect.

Table 1: Confusion matrix

		Predict	
		Positive	Negative
Real	Positive	1071	45
	Negative	62	1446

Table 2: Comparison of accuracy of different models

Model	ACC%
Optimized GoogLeNet	95.92
VGG16	94.34
CNN	89.64
LeNet-15	92.33

4. Conclusion

In this paper, the surface defects of solar cells are detected with optimized GoogLeNet. Experiments show that the accuracy of our model can reach 96%. Our network model has fewer parameters, which reduces overfitting to a certain extent. The training time of the model is also shorter. The subsequent work studied how to efficiently segment the defect image and further explored the problem of solar cell defect detection.

References

[1] Tsuzuki K, Murakami T, Yoshino T, et al. Inspection method and production method of solar cell module: US, US6271462 B1[P]. 2001. ESQUIVEL O . Contrast imaging method for inspecting specular surface devices: US6433867B1 [P] . 2002-08- 13.

[2] Sawyer D E, Kessler H K. Laser Scanning of Solar Cells for the Display of Cell Operating Characteristics and Detection of Cell Defects [J]. IEEE Transactions on Electron Devices, 1980, 27(4):864-872.

[3] X. Y. CHEN, A. PEDERSEN, O. G. HELLESO, et al. Electrical noise of laser diodes measured over a wide range of bias currents [J]. Microelectronics and reliability, 2000, 40 (11):1925-1928.

[4] Anwar, Said Amirul, and Mohd Zaid Abdullah. "Micro-crack detection of multicrystalline solar cells featuring an improved anisotropic diffusion filter and image segmentation technique." EURASIP Journal on Image and Video Processing 2014.1 (2014): 1-17.

[5] Tsai, Du-Ming, Chih-Chieh Chang, and Shin-Min Chao. "Micro-crack inspection in heterogeneously textured solar wafers using anisotropic diffusion." Image and vision computing 28.3 (2010): 491-501.

[6] S. A. Bakalexis, Y. S. Boutalis and B. G. Mertzios, "Edge detection and image segmentation based on nonlinear anisotropic diffusion," 2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628), 2002, pp. 1203-1206 vol.2, doi: 10.1109/ICDSP.2002.1028309.

[7] Lu, Chi-Jie, and Du-Ming Tsai. "Automatic defect inspection for LCDs using singular value decomposition." The International Journal of Advanced Manufacturing Technology 25.1 (2005): 53-61.

[8] Demant, M., Welschehold, T., Oswald, M., Bartsch, S., Brox, T., Schoenfelder, S., & Rein, S. (2015). Microcracks in silicon wafers I: Inline detection and implications of crack morphology on wafer strength. IEEE Journal of Photovoltaics, 6(1), 126-135.

[9] Ramachandran, Prajit, Barret Zoph, and Quoc V. Le. "Searching for activation functions." arXiv preprint arXiv:1710.05941 (2017).

[10] Deitsch, Sergiu, et al. "Automatic classification of defective photovoltaic module cells in electroluminescence images." Solar Energy 185 (2019): 455-468.

[11] Deitsch, Sergiu, et al. "Automatic classification of defective photovoltaic module cells in electroluminescence images." Solar Energy 185 (2019): 455-468.

[12] H. Yanagisawa, T. Yamashita and H. Watanabe, "A study on object detection method from manga images using CNN," 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 2018, pp. 1-4.

[13] L. Lu, Y. Yi, F. Huang, K. Wang and Q. Wang, "Integrating Local CNN and Global CNN for Script Identification in Natural Scene Images," in IEEE Access, vol. 7, pp. 52669-52679, 2019.

[14] H. Tahir, M. Shahbaz Khan and M. Owais Tariq, "Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles," 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2021, pp. 587-594.

[15] J. Tao, Y. Gu, J. Sun, Y. Bie and H. Wang, "Research on vgg16 convolutional neural network feature classification algorithm based on Transfer Learning," 2021 2nd China International SAR Symposium (CISS), Shanghai, China, 2021, pp. 1-3.
[16] A. Panthakkan, S. M. Anzar, S. A. Mansoori and H. A. Ahmad, "Accurate Prediction of COVID-19 (+) Using AI Deep VGG16 Model," 2020 3rd International Conference on Signal Processing and Information Security (ICSPIS), DUBAI, United Arab Emirates, 2020, pp. 1-4.
[17] A. R, C. C K and S. Chaudhari, "Comparative study of CNN, VGG16 with LSTM and VGG16 with Bidirectional LSTM using kitchen activity dataset," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 836-843.
[18] H. Aung, A. V. Bobkov and N. L. Tun, "Face Detection in Real Time Live Video Using Yolo Algorithm Based on Vgg16 Convolutional Neural Network," 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Sochi, Russia, 2021, pp. 697-702.
[19] H. Wang, "Garbage Recognition and Classification System Based on Convolutional Neural Network VGG16," 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Shenzhen, China, 2020, pp. 252-255.
[20] A. Huang, Q. Wang, L. Jiang and J. Zhang, "Automatic Segmentation of Median Nerve in Ultrasound Image by a Combined Use of U-Net and VGG16," 2021 IEEE International Ultrasonics Symposium (IUS), Xi'an, China, 2021, pp. 1-4.
[21] S. Tian, W. Li, S. Li, G. Tian, L. Sun and X. Ning, "Image Defect Detection and Segmentation Algorithm of Solar Cell Based on Convolutional Neural Network," 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP), 2021, pp. 154-157, doi: 10.1109/ICSP51882.2021.9408827.
[22] F. Yuesheng et al., "Circular Fruit and Vegetable Classification Based on Optimized GoogLeNet," in IEEE Access, vol. 9, pp. 113599-113611, 2021, doi: 10.1109/ACCESS.2021.3105112