Digital Image Final Project Report - Automated Optical Inspection

107061522 沈煒翔
107061629 蔡喻翔
107061539 施文弘

I. Introduction

Automated Optical Inspection (AOI) is the process to inspect optical defects in industrial production. Traditional inspection method is time-costly and inaccurate. By combining image processing techniques and computer vision techniques, we are able to build to system that automatically inspects optical defects.

In this project, we classify the optical defects into five different categories, void, horizontal defect, vertical defect, edge defect, particle. After classification, we need to mark the defect region, so that maybe the defect can be fixed.

For classification, we build a Convolutional Neural Network (CNN) model. For segmentation, we apply various image processing to mark the defect regions according to the predicted classification result of the CNN.

II. Optical Defect Classification

A. Convolutional Neural Network Model

Convolutional Neural Network (CNN) has shown great results in the field of image classification. The convolutional layers is naturally suitable for image data, since it acts like feature extraction filters. We use CNN to classify optical defects into different categories.

We construct and train a new CNN model (fig. 1). The CNN consists of 3 convolution layers and 2 fully-connected layers. The 3x3 kernel size of the convolutional layers are inspired by VGG-Net [1], which showed that 3x3 convolution kernel is standard in image classification CNNs. Also, we use random search to optimize the hyperparameters, including layers count, node count and learning rate. Drop out layers are added the make the training stage more stable.

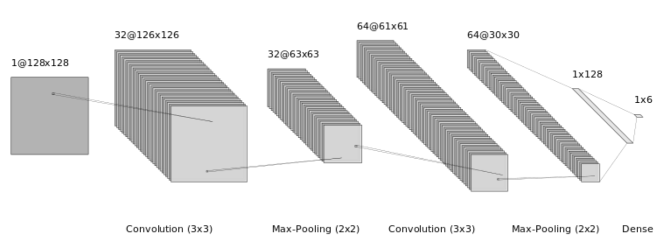

Fig. 1. Neural Network Architecture. Notice that drop out layers are not drawn in this figure.

B. Image Augmentation

In the optical defect dataset, the categories are imbalanced. The CNN would refuse to predict the smaller categories. Therefore, we try to resample the smaller categories by naively oversampling them, so that the CNN have more training samples in those smaller categories. However, naively oversampling cause the categories to converge in different speed; one category may begin to over-fit while the others may still be under-fitting. Therefore, we conclude that naively oversampling would not work, and turn to image augmentation.

Image Augmentation were used on both training and validation datasets. The augmentation includes vertical and horizontal flip, and padding. After the augmentation, the dataset is approximately 15 times larger than the original dataset. By enlarging the dataset, we can prevent overfitting, and make the training stage more stable.

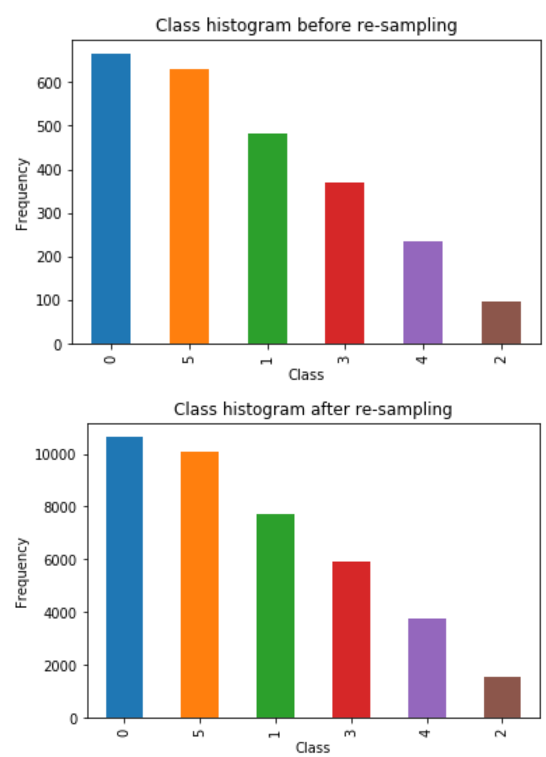

Fig. 2. Comparison of doing and not doing image augmentation. Notice the change of data size.

III. Optical Defect Segmentation

Different image processing procedures are applied to different defect categories. We assume the prediction of CNN is correct, and apply the following image processing methods.

A. Void Defect

First, The noise is removed by using Gaussian filter (fig. 3b) and median filter (fig. 3c). Gaussian filter is used to reduce image noise. Median filter is used remove extreme value. After using the filters, the image is smoothed.

We want to find the threshold between normal region and defect region. To determine the threshold, we observe the histogram distribution of pixel intensity (fig. 4). There are two obvious peaks that contribute to normal region and void region.

We assume the local minimal point between two peaks is the threshold. To prevent finding points at the very start or very end of the histogram, the 20% beginning and end value were deleted. By doing so, we can find the adaptive threshold to distinguish void region and normal region.

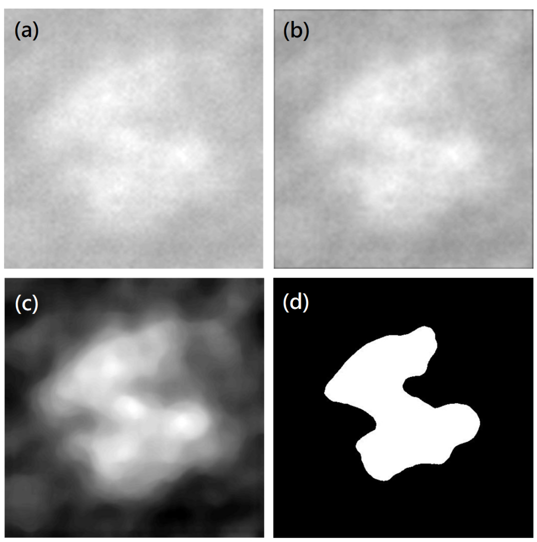

Fig. 3. Void Defect Segmentation. (a)Original Image, (b)After Gaussian filter, (c)After Median filter, (d)After adaptive thresholding.

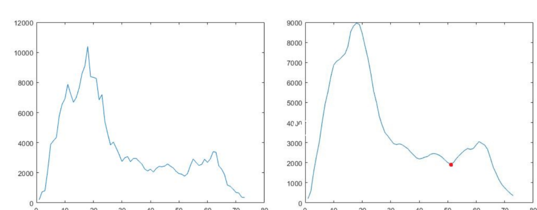

Fig. 4. Histogram distribution of pixel intensity. The red point in the right point is the adaptive threshold.

B. Horizontal Defect

In horizontal defect, there is an obvious horizontal line/lines in the image. If we average all pixel intensity in each row, we can see an drop-off in the intensity in each row (fig. 5b).

We use derivative method to find the drop-off intensity area. We use 5x5 median filter followed by 5x5 averaging filter. Before differentiating the image, it should be averaged every 100 columns. Then we can differentiate the image by convolution with a gradient operator [-1;1] (fig. 5c). At last, we smooth the image with 1x9 averaging mask to get a difference image.

For both ramp edge and roof edge, its position should be near the peak value of the first derivative. So we set a threshold with magnitude half of the peak value. If the peak value is positive, all pixels higher than threshold are set to 1, others are set to zero. If the peak value is negative, all pixels lower than threshold are set to 1, others are set to zero (fig. 5d). Finally, we resize the image back to 512x512.

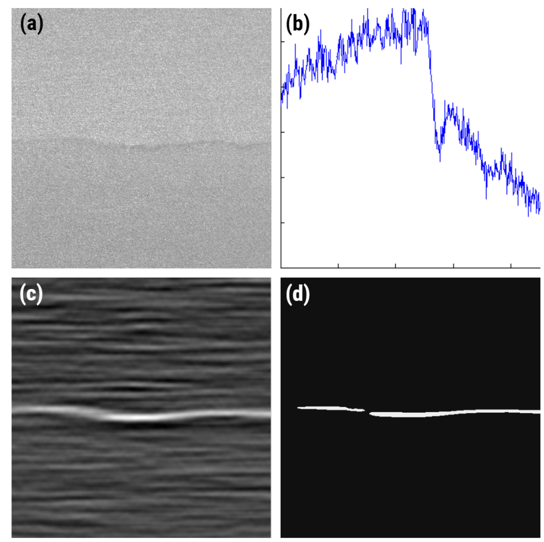

Fig. 5. Horizontal Defect Segmentation. (a)Original Image, (b)Average intensity of each row, (c)Difference image after gradient operator, (d)Image after thresholding.

C. Vertical Defect

We rotate the image 90 degree so that it becomes a horizontal defect image, and pass it into the same algorithm used in horizontal defect. At last, transpose it again and we can get the final result.

Note that because vertical edges are more visible than horizontal edges, we average by every 50 columns instead of 100 columns, and the threshold is set to 0.3 instead of 0.5 in horizontal defects (fig. 6).

D. Edge Defect

We can see obvious feature on the images, so we find the edge boundary. To remove the noise and extreme value, we use median filter (fig. 7b). we use Canny edge detector [2] and larger value of standard deviation of the Gaussian filter (fig. 7c). Then we dilation the boundary with a 3x3 matrix of ones to make the boundary more visually visible (fig. 7d).

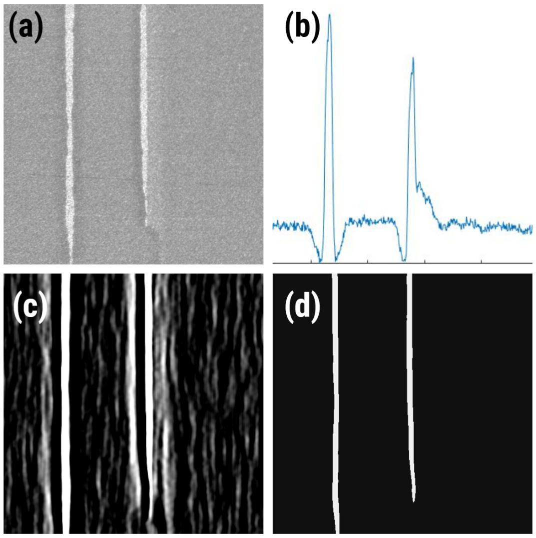

Fig. 6. Horizontal Defect Segmentation. (a)Original Image, (b)Average intensity of each row, (c)Difference image after gradient operator, (d)Image after thresholding.

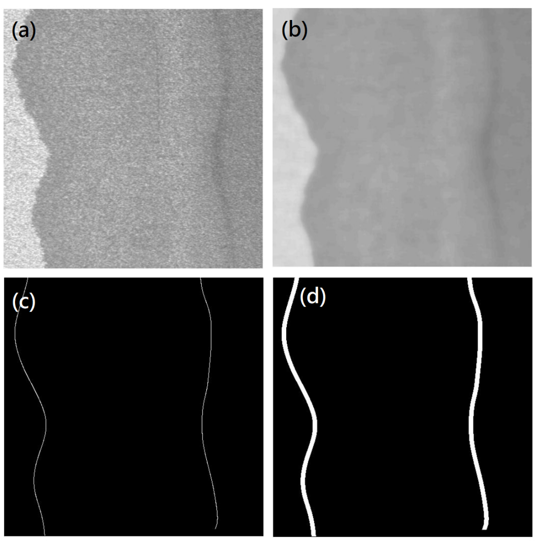

Fig. 7. Edge Defect Segmentation. (a)Original Image, (b)Image after median filter, (c)Canny edge detection using large deviation of Gaussian filter, (d)Dilation to make the boundary more visually visible.

E. Particle Defect

We can see that there is a dark region in the middle of image, and many speckles around the image. In particle defect category, we would mark both defects.

First, we remove the noise by using median filter. (fig. 8b) To mark the speckles in the image, we define the mean of the image as the threshold (fig. 8c).

For the dark region in the middle of the image, we find the local minimal point between two peaks in histogram distribution of pixel intensity. The method is similar as those used in void defect. It can clearly mark dark region in the middle (fig. 8d).

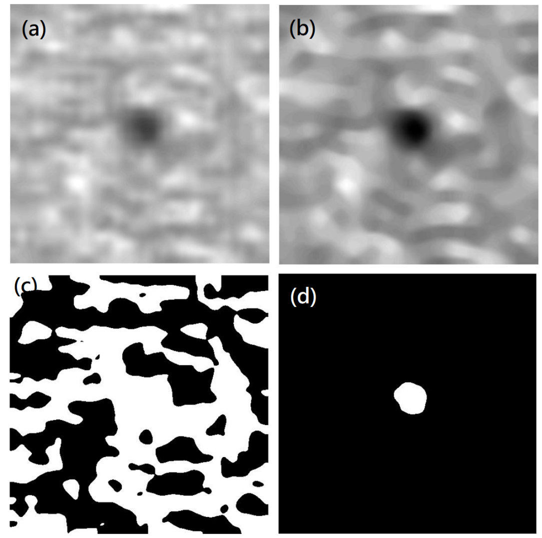

Fig. 8. Edge Defect Segmentation. (a)Original Image, (b)Image after median filter, (c)Speckle defect by adaptive thresholding, (d)Dark region defect by adpative thresholding.

IV. Results and Discussion

A. CNN Result

The CNN model is trained 25 epochs by Adam optimizer using learning rate of 0.01. Detail training settings can be viewed in the provided code.

We separate the training dataset into training and validation set in the ratio of 0.8 and 0.2. 5-fold cross validation is used for estimating the performance of the model. The proposed CNN model achieve 98.21% during cross validation and achieve 98.99% accuracy in the private leader board of AIdea website (fig. 9).

B. Segmentation Result

Segmentation of void defect sometimes cannot mark defect region. We conclude several circumstances that the algorithm could fail. One, some images classified in void looks normal. Second, the adaptive threshold sometimes fail to find an optimal point to distinguish to normal and defect region. In some case (fig. 10), the void defect region is not defined very well.

For edge defect, There are some images' edges are visually unclear (fig. 11), which have many wrong edges in it. However, changing the threshold can remove wrong edges. That is because the threshold is same to all images, but it should be adjust between different images. Maybe we can decide threshold by the images' standard deviation or other statistical method.

V. Conclusion

In this project, we classify the optical defect images into six different categories. We build and train a new CNN for image classification, and achieve 98.99% accuracy on the private leader board of AIdea, which is the 3rd team in class.

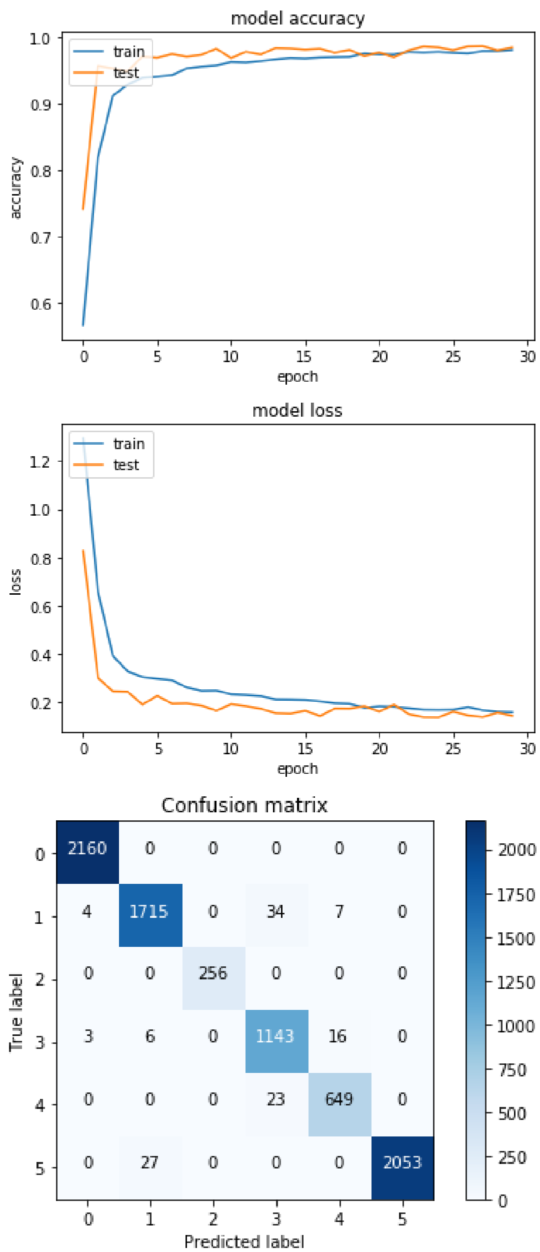

Fig. 9. Accuracy and loss with each epoch. Note that the orange line is actually validation accuracy and loss. Confusion matrix is plotted using validation dataset.

Image segmentation is used to mark the defect region. The segmentation rely on the prediction of CNN, since different algorithms are used in different defect categories. For most images, we are able to mark the defect region clearly.

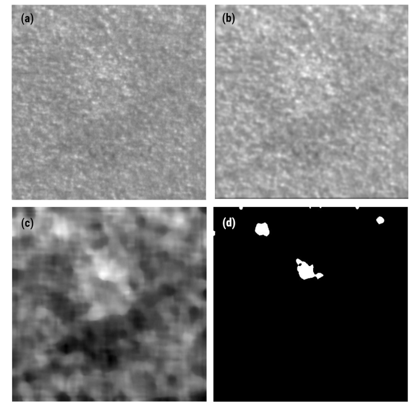

Fig. 10. Fail segmentation of void category due to unclear definition of defect region.

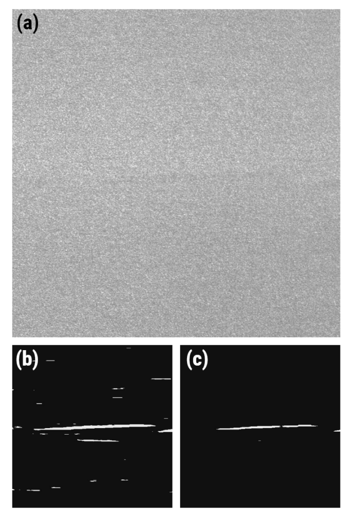

Fig. 11. Fail segmentation of edge category due to unclear definition of defect region and fixed threshold.

References

[1] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. CoRR, 1993.

[2] John Canny. *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Nov. 1986.