

MATH4995 Final Project: Nexperia Image Classification using CNN

WAHYU, Zoya Estella

Department of Mathematics

Introduction

Nexperia provided a dataset containing the images of semiconductors. The dataset contains 7039 images of defective semiconductors, including the part of defect, as well as 27420 images of good semiconductors. Our task is to classify whether each of the remaining 3830 images contains a defective semiconductor or not.

Methodology

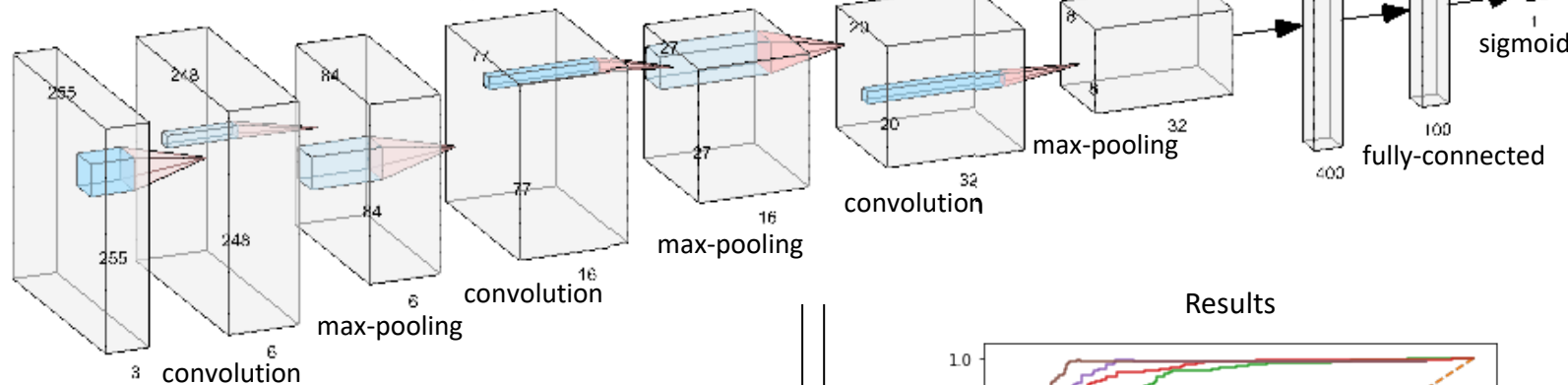
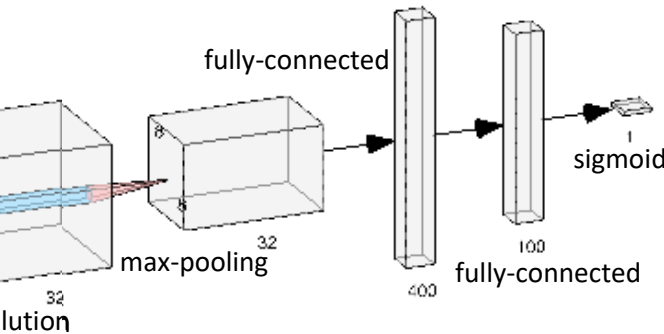


Fig.1 Model architecture

We approach the image classification problem by building a simple CNN model consisting 7 layers, where the input size is the cropped image of size 255×255×3, and the final output [0,1] denotes the probability that the corresponding image has a defective semiconductor. The initial model was built to be similar to LeNet, but the specific parameters were decided arbitrarily.

Hyperparameter Tuning

- Dropout rate:
 - 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45
- Number of nodes in the fully connected layer:
 - 1500/1000/600/400/200-1000/500/100/50/10



Results

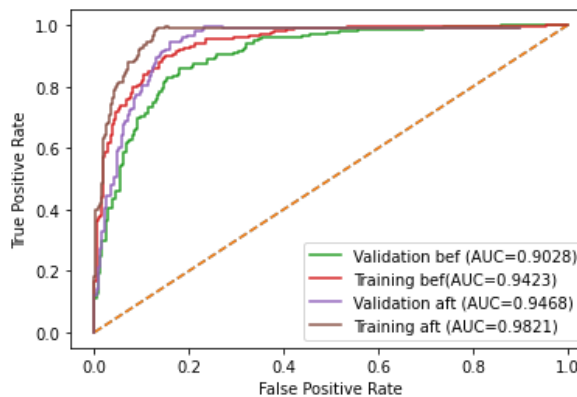


Fig. 2. ROC Curve before and after hyperparameter tuning

At first, we initialized the dropout rate to be 0.3. After hyperparameter tuning, we found the most optimum dropout rate to be 0.15

We also initialized the number of nodes in the dense layers to be 2048-1000-200-1. After the tuning, the most optimum number of nodes is 2048-400-100-1.

The AUC score before and after the tuning increased by four percent.

accuracy: 0.9831	label: defect	label: good
predict: defect	1373 (0.1993)	82 (0.0119)
predict: good	35 (0.0051)	5402 (0.7839)

Fig. 3. Confusion matrix of validation data

Precision: 0.9436, Recall: 0.9751, F1 Score: 0.9591

The Kaggle AUC score is 0.9782.

Conclusion

For further improvement, we can try hypertuning other parameters such as the total number of convolutional layers or the nodes sizes in other layers. Image pre-processing such as image segmentation or denoise may also be needed. We can also try to do transfer learning on preexisting trained models.

Repository:

<https://github.com/zoyaew/MATH4995/tree/master/project2>