Final Report: Nexperia Kaggle in-class contest

Luyu Cen Jingyang Li Zhongyuan Lyu Shifan Zhao
Hong Kong University of Science and Technology

Abstract

We tested various models including VGG 16, and Resnet on the images of semi-conductors for image classfication. Also, since the good images take up most of the training set, we used styleGAN to generate some fake bad samples. We further utilized OCSVM and Isolation Forest to detect the anomalies in the training set and heatmap to localize the defect areas.

1 Introduction

Nexperia is one of the biggest Semi-conductor company in the world. They will produce billions of semi-conductors every year. Unfortunately, they are facing a hard problem now which is efficient anomaly detection of the semi-conductors. Nexperia accumulated lots of image data using online digital camera and hoped that human based anomaly detection could be greatly improved by the state-of-the-art deep learning techniques. The main task is to classify those images into two classes: good and bad. In this project, we are given 30k images for training and 3000 images are for testing.

Although there are a lot of successful models in image classification including Alexnet, VGG, Resnet, the difficulty in this semi-conductors image binary classification task lies in: (1) the imbalanced dataset; (2) abnormal images that may lead the model to learn something wrong. We mainly deal with these problems by generating fake bad images by transformation and styleGAN. Further, we did anomaly detection using one-class-SVM and isolation forest to exclude images with misleading information.

The rest of this paper is organized as follows. We first introduce how we preprocessed our data using transformation and styleGAN in Section 2. In Section 3, we give a simple introduction to the models we used. In Section 4, we demonstrate how we performed the anomaly detection and gave some visualisation using PCA and heatmap.

2 Data augmentation

In this project, we used two data augmentation methods to generate images for the minor class. One is by simple image transformations, the other one is by a generative adversarial network called StyleGAN.

2.1 Image transformation

Our first try of the models was based on the augmented training dataset using the image transformations such as rotation by a small degree, horizontal and vertical shifting, and adjusting the brightness, the contrast, and the saturation factor to compensate the minority of images labelled as "bad", since the transformations are easy and efficient. We generated enough "bad" images from the training images so that we have 40,000 images in total for the training dataset.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

2.2 Style GAN

StyleGAN is a novel generative adversarial network, invented by Nvidia research group. It has been well known for its impressive ability to generate high quality images. Some researches in medical imaging have indicated that styleGAN is very useful for limited dataset. Considering our case here, we have a very imbalanced dataset with much more good samples than the bad ones. Hence we want to exploit the powerful styleGAN to generate some bad samples and thus our dataset will be enriched. Someone may argue that if we use the original bad samples dataset as the input for GAN, the generator should not produce new information. This may be true for GAN, WGAN, but it's not true for styleGAN. Incorporating the embedding for latent space and disentanglement, the styleGAN could reveal more information than others. In Figure 1, we show some examples generated by styleGAN. We totally obtained around 3000 bad samples in this way to balance our dataset.

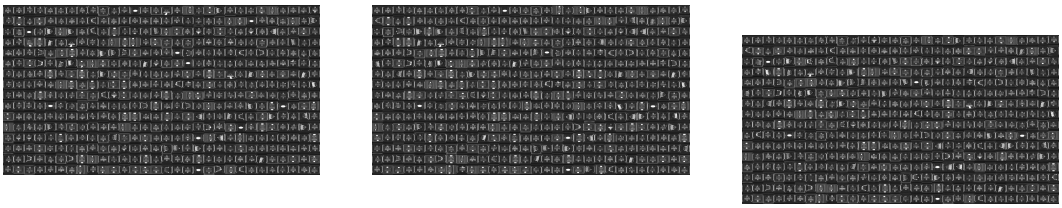


Figure 1: Generating bad samples using styleGAN

3 Models

3.1 VGG 16

The original VGG16 architecture consists of twelve convolutional layers, some of which are followed by maximum pooling layers, and four fully-connected layers and finally a 1000-way softmax classifier. However, we were dealing with a binary classification task, so we modified the last layer so that the output would be a two dimensional vector.

3.2 Resnet | Wide Resnet | Resnext

It is believed that deeper networks perform better. But after inserting more identical layers into CNNs, people find they usually become less efficient. That's because of the well-known gradient explosion and vanishing. This problem was beautifully solved by Resnet proposed by Kaiming He and his collaborators. In their paper, they introduced residual learning blocks. By using feedforward neural networks and "shortcut connection", their network actually is learning residual mapping rather than the original mapping. In our context, we used the pretrained Resnet-18 provided by Pytorch models. To make it fit for the binary classification, we just changed dimension of the fully connected layer to be $512 * 2$ and trained the model again on our dataset. We tried to freeze some top layers but we found it was better to train all the parameters again. We also tested Wide Resnet and Resnext. These two models made some minor changes to Resnet. To be more specific, Wide Resnet made the channels of Resnet wider and Resnext added a new dimension to Resnet-cardinality. But in fact, they just added more residual blocks in the bottleneck.

3.3 Comparsion of models

Comparsion List			
Models	Training Accuracy	Testing Accuracy	AUC
Resnet18	0.9938	0.986152	
Resnet50	0.9882	0.987141	0.98866
Wide Resnet50-2	0.9935	0.984174	
Resnext50-32x4d	0.9936	0.985163	
VGG 16	0.9777	0.977700	

4 Beyond Classification

4.1 Anomaly Detection

Outliers can significantly affect the performance of deep learning models, and thus unsupervised abnormal outlier detection is necessary. However, we cannot directly apply conventional outlier detection techniques on image data due to the spatial structure of them. To address this issue, we conducted a method utilizing conventional outlier detection methods on features extracted by a pretrained Resnet-50 on the original image data, in the light of [1]. The pretrained ResNet-50 model was trained on ImageNet [2] and we used the next to the last layer as the features for further outlier detection. Note that in our work the anomaly detection is another task to gain some insights about the characteristics of the data, independent from the prediction task.

4.1.1 OCSVM

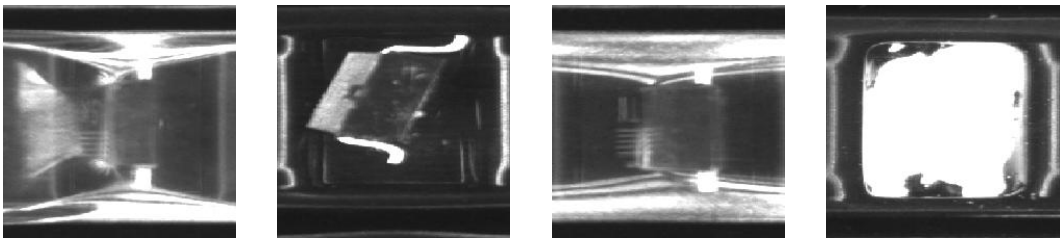
The basic idea of OCSVM is to find a hyperplane having the maximum margin differentiating the regions with high density of points (normal points) and with low densities(outliers). We used OCSVM with Gaussian rbf(radial basis function) kernel. The hyper-parameters we need to specify were γ and ν. γ is the kernel coefficient in the conventional SVM, i.e., the parameter that is propotional to the reciprocal of the variance of Gaussian distribution in our case. We left it as default setting in *sklearn* which uses $1/(p \cdot var(X))$ as value of γ, where p is the number of features we extracted. ν is the fraction of the data that is contaminated, i.e., identified as outliers. We set $\nu = 0.01$ in our experiments.

4.1.2 Isolation Forest

Isolation Forest is another outlier detection technique based on binary decision trees [3]. The basic assumption of it is that anomalies are few and far from the rest of the observations. The algorithm randomly picks a feature from the feature space and a random split value ranging between the maximums and minimums for all the observations to build a tree in the training set. A standard tree ensemble is made which averages all the trees in the forest. The number of splittings required to isolate a sample is equivalent to the path length from the root node to the terminating node, which becomes a measure of normality and the decision function. The only hyper-parameter we need to specify in *sklearn* is $contamination$, which is similar to the ν in OCSVM. We set $contamination = 0.01$ for comparison with OCSVM.

4.1.3 Some insights

To some extent, we combined OCSVM and Isolation Forest to do the anomaly detection. To be specific, we picked images identified as anomalies by both OCSVM and Isolation Forest as our final outliers. It is evident that some of these images are indeed abnormal and make the discriminator hard to identify. For instance, Figure 2a and 2b are good semi-conductors but even for human it is hard to say whether it is good or bad. Figure 2c and 2d are indeed "bad", but due to the quality of the image, they are anomalies in the training process since it does not reflect the characteristic of what a truly defective semi-conductor looks like.



(a) outlier in "good" (b) outlier in "good" (c) outlier in "bad" (d) outlier in "bad"

Figure 2: Outliers

4.2 Data Visualization

4.2.1 PCA

Besides deep neural networks, we tried conventional machine learning methods, for example, SVM, randomforest, LDA and etc., using the features extracted by ResNet-50. However, the performance did not improve compared to the last fully connected layer of original networks. To see why this happens, it's worthwhile to do PCA on the these features. The first two principle components shows that they can be separated relatively well by a linear subspace. In the light of this, it is not necessary to conduct conventional classification methods on these features.

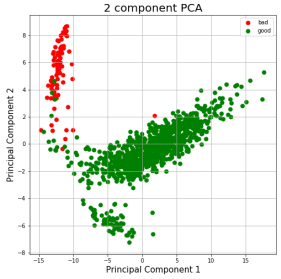


Figure 3: Visualization for first two PCs

4.2.2 Heatmap

Because of the small size of filters, an activation on a feature map towards the end of convolutional networks still retains some local information of the corresponding region in the original image, despite that the receptive field gets larger as the network goes deeper[4]. This makes ConvNets useful not only for classification, but also for object localization. Here we used two heatmap methods, class activation mapping and Grad-CAM, to localize the defect, and meanwhile, to see which part of the original image arouses most attention of different convolutional neural layers.

Class Activation Mapping(CAM) A $class\ activation\ mapping$(CAM) is a function of the pixel space of the image that quantifies how much a specific pixel contributes towards letting the image belong to a given class. Zhou et al.[5] proposed a way to calculate the CAM by taking advantage of $global\ average\ pooling$(GAP) layer. At the end of convolutional layers, networks, such as ResNet and Inception use GAP. A GAP averages activations on each feature map and feeds them to the linear classifier, which further calculates the weighted average of these averages, to obtain the class score for a given class. For each class, each feature map thus associates to a learned weight. The weights measure the importance of the feature map towards classifying the image to a given class. Instead of the weighted average of the averages of feature maps, the class activation map is simply this weighted average of the feature maps, subject to upsampling to the original pixel space by interpolation since the feature map space is smaller than the original one.
After we obtained the CAM, we found the largest connected region with class activation above 0.4 times the maximum class activation. As shown by Figure **??**, the method successfully localizes the defect area of this image.

Grad-CAM To see the attention shifting in the training process and gain some understanding about the perspective of neural networks, we applied the Grad-CAM to visualize the image. This method was proposed in [6], inspired by the previous work[5]. The authors invented a more general method. Since CAM actually just maps the weights to the image after global average pooling(GAP), it cannot be used for networks without GAP. In order to free this method of this restriction, Grad-CAM was invented, which just need the gradient information of the network's parameters. Hence we can apply it to each layer and trace the attention shifting. What's more, Grad-CAM can localize the important regions more accurately than CAM. In Fig.5, we can see very clearly the focus of resnet18 changes very suddenly. From Fig.5b, Fig.5c, Fig.5d, we can infer first three layers can not localize

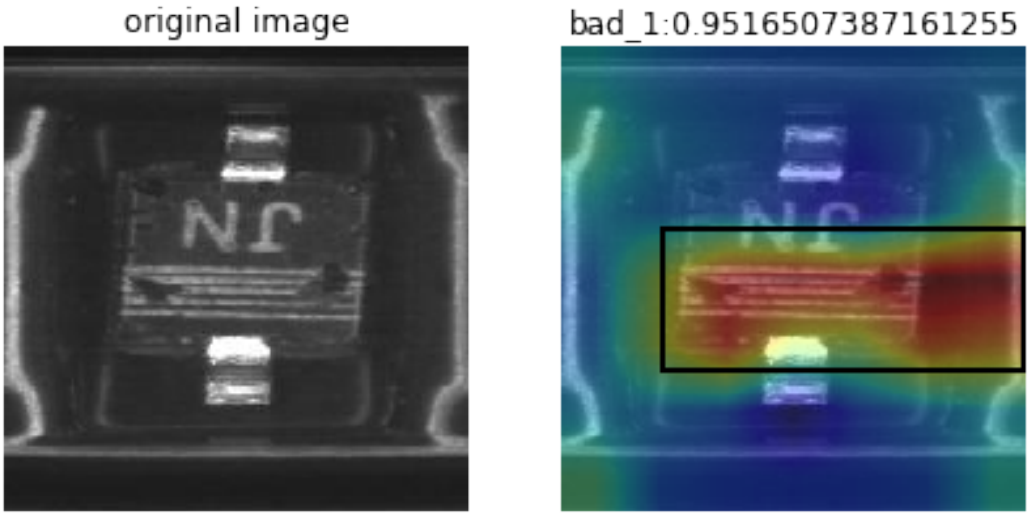
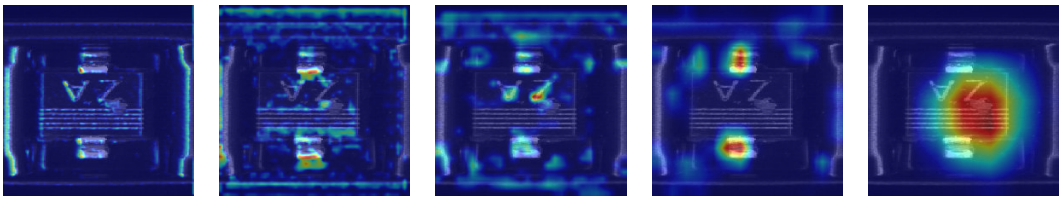


Figure 4: Defect localization by CAM

the important regions accurately. Worse than that, the handles of the semi-conductor was regarded as the most important regions which can be seen from Fig.5d where the red regions mean the most important regions for the Resnet18, which might mislead it far from the real case. But surprisingly, in the last layer, Fig.5e shows a coarse localization map highlighting true important regions in the image for predicting the defect in this semiconductor. This may provide us with some meaningful lessons that we could find some way to reduce the influence of the handles of the semi-conductor. For example, we could merge the handles with its surrounding parts so that the resnet will mainly focus on the defect ares in previous layers. Maybe this will improve the accuracy and could be considered for doing classification.



(a) relu (b) layer1 (c) layer2 (d) layer3 (e) layer4

Figure 5: Grad-CAM using Resnet18

5 Contributions

- Luyu Cen did the initial data augmentation using simple image trainsformations, tested Resnet50, extracted features from the pretrained and also the freshly trained Resnet50, and applied CAM for defect localization.
- Shifan Zhao used styleGAN to generated bad samples, tested the model on Resnet18, Wide Resnet and Resnext, and applied Grad-CAM to visual the training process and images.
- Zhongyuan Lyu conducted PCA and anomaly detection (OCSVM and Isolation Forest) using pretrained features and tried conventional machine learning methods using pretrained features.
- Jingyang Li used VGG 16 to test on original data and test on the data without abnormal images, and on the data with images generated by StyleGAN.

References

[1] Y. Wang, R. Yoshihashi, R. Kawakami, S. You, T. Harano, M. Ito, K. Komagome, M. Iida, and T. Naemura, "Unsupervised anomaly detection with compact deep features for wind turbine blade images taken by a drone," *IPSJ Transactions on Computer Vision and Applications*, vol. 11, no. 1, p. 3, 2019.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[3] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, IEEE, 2008.

[4] J. L. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?," in *Advances in Neural Information Processing Systems*, pp. 1601–1609, 2014.

[5] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.