MATH 6380o Project 2: Anomaly Detection in Semiconductors

Ng Yui Hong, yhngap @connect.ust.hk
Department of Mathematics, HKUST

1. Introduction

In this project, we try to detect “bad” semiconductors from a dataset of 30,000 labeled images of semiconductors provided by Nexperia. 27,000 images in the dataset are “good” and only the remaining 3,000 are labeled as “bad”, thus, this problem dictates using techniques to tackle the inherent class imbalance. To this end, we use image transforms to augment our dataset, oversample from the “bad” class and perform test time augmentation to produce predictions. I was able to obtain an AUC score of 0.943 on the Kaggle public leaderboard based on this implementation.

2. Nexperia Dataset (In-class Kaggle Competition)

Methodology
I used a pre-trained ResNet18 as the backbone model for this project.
Why?
Taking inspiration from Project 1, I initially wanted to use a combination of Scattering Net and XGBoost, but ultimately my choice was dictated by the ability to do minibatch learning which is not possible with XGBoost. Thus, pre-trained deep CNNs were a natural choice.
Class Imbalance
As stated before, the “bad” examples constitute only 10% of the training dataset. The figure below shows the confusion matrix when a pre-trained ResNet18 is finetuned on the dataset. A very high false positive rate is apparent.

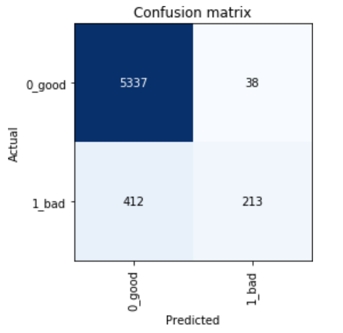

Fig 1. Confusion matrix after training with imbalanced dataset

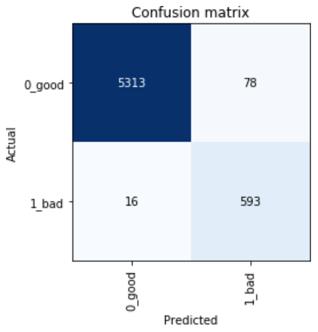

Fig 2. Confusion matrix after training with oversampling and transforms

3. Oversampling and Data Augmentation

I oversample from the underrepresented “bad” class to make the class distribution approximately uniform. I also noticed that not all of the images in the dataset were taken from the same angle. That necessitated in teaching the model the various orientations in which the images could occur. For this, I applied random flips, rotates, zooms, contrast and brightness changes etc. Fig 3 shows an example of the transforms applied. Fig 4 attempts to show a visualization of the top images the network wrongly classified as good, along with a heatmap. Some of the images are indeed very hard to classify as good or bad just by looking at the image itself and therein lies the biggest challenge.

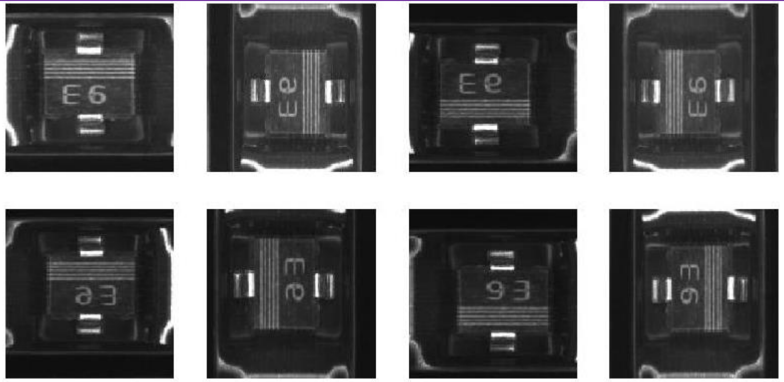

Fig 3. the transformations applied to training data

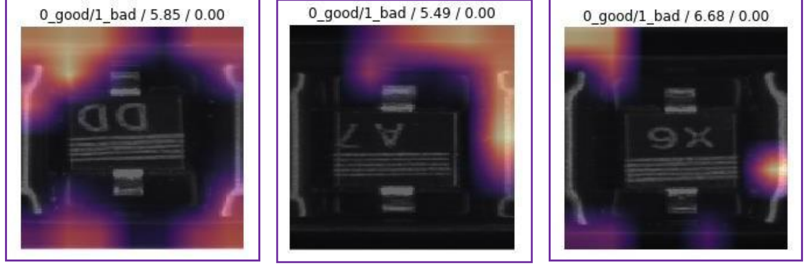

Fig 4. Images misclassified as good. The heatmap attempts to indicate what areas of the image the network found useful to make its decision

4. Detailed Methodology, Analysis and Conclusion

The success of this classification has relied heavily on characteristics of transfer learning. As mentioned in Section 2, I used a pre-trained ResNet18 as the backbone model – first I trained the newly added deeper layers keeping shallow layers frozen. In the second pass, I unfroze the entire model and fine tuned all the 11 million+ parameters. Differential training was employed – different layer weights were updated using different learning rates, chosen from an interval.

Predictions were done using Test Time Augmentation – 8 transforms of each test image were generated and a prediction was generated on each transformed image. The final prediction was then computed as the majority vote of the 8 predictions. With some modifications on TTA transformations to match the transformations done on the training dataset, I was able to improve my AUC score on Kaggle public leaderboard from 0.91+ to 0.943+. I was consistently able to get an accuracy of 98.5%+ validation sets in my experiments.

It is easily observable that even though without domain expertise, one might find it hard to differentiate between good and bad semiconductors just from images alone, deep CNNs do a commendable job at it. There are definitely patterns that hidden from the human eye which are exploited by the model. In this sense, this problem is very different from MNIST, CIFAR etc where humans are a formidable baseline.

5. Future Work

It would be fascinating to compare the performance of the current model with:

1. A Generative Adversarial Network
2. Variational AutoEncoder
3. A combination of Scattering transform for feature extraction followed by gradient boosting (XGBoost) for classification

#3 in particular would be incredibly interesting, seeing as we established in Project 1, that scattering transforms are very good at extracting discriminative features.

6. References

- Simonyan et. al, *Very Deep Convolutional Networks for large-scale image recognition*, ICLR 2015