Nexperia Image classification

Junming CHEN(20750649),
Zifan SHI(20619455),
Rongrong GAO (20619663)

Overview

- Data preprocessing
- Traditional machine learning method
- End-to-end deep learning method
- Results
- Conclusion

Nexperia Dataset

- Dataset Overview
  - It aims to classify images of semiconductor devices into two main classes, good and defect.
  - It contains 34457 train images (27420 good and 7039 bad) and 3830 test images with similar good-to-bad ratio.
- Task
  - To use **deep learning** methods to help pick out as many defect devices as possible while preserving the good ones
  - To automatically examine all of the products and **select the good** ones from the unqualified devices, thus improving their yield rate.

Nexperia Dataset

- Problems & Solutions
  - Noisy labels
    - Problem: Non-devices and those without ground truth
    - Solution: Checking annotation area and remove those without annotations.
  - Class imbalance
    - Problem: Good : Defect ≈ 4:1
    - Solution: Focal loss.
  - Duplicates
    - Problem: Duplicates may exist in both training and validation sets
    - Solution: Duplicates detection by hash scan.

Traditional method

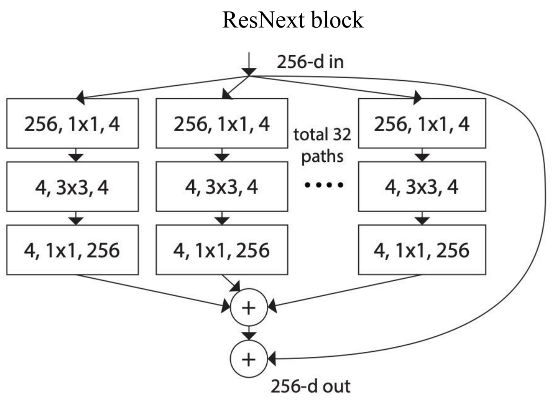
- Feature extraction + classifier
  - Feature extraction
    - ResNeXt-101 pretrained on ImageNet dataset.
    - ResNeXt-101 fine-tuned on Nexperia dataset.
  - Random forest
    - A set of decision trees are constructed to fit on the various sub-samples of the dataset
    - Averaging is adopted among all the decision trees to get the final prediction.
- Our contribution
  - In this study, we explore the influence of pretrained and fine-tuned features for image classification with random forest.

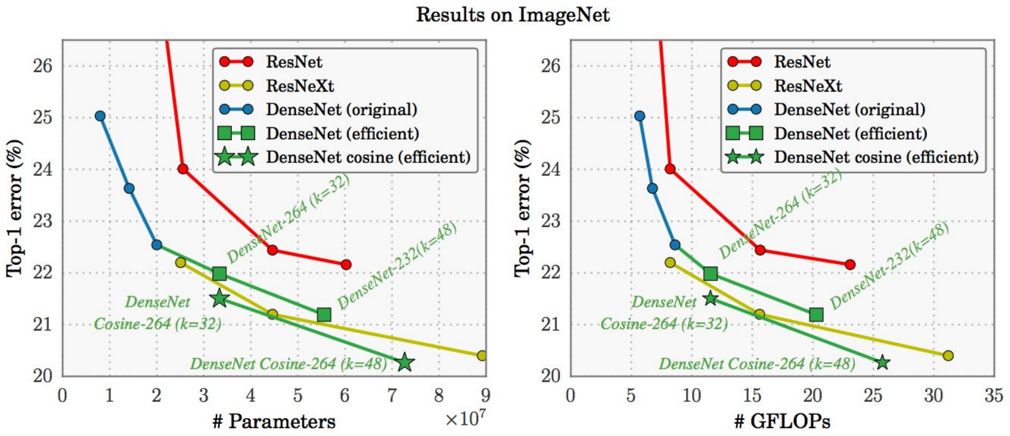
End-to-End learning

- Traditional method
  - Disadvantages
    - Difficult to extract feature with effective representation on our task.
    - Human intervention in the classification step, e.g., set the hyperparameters
- End-to-end method
  - Advantages
    - Being able to learn a strong feature for a specified task.
    - Even a simple and automatic classifier can work well on that feature.

Our Method

- ResNext
  - ResNeXt block
    - Combining the advantages of ResNet and Inception.
  - Performance compared with other methods.





Our Method

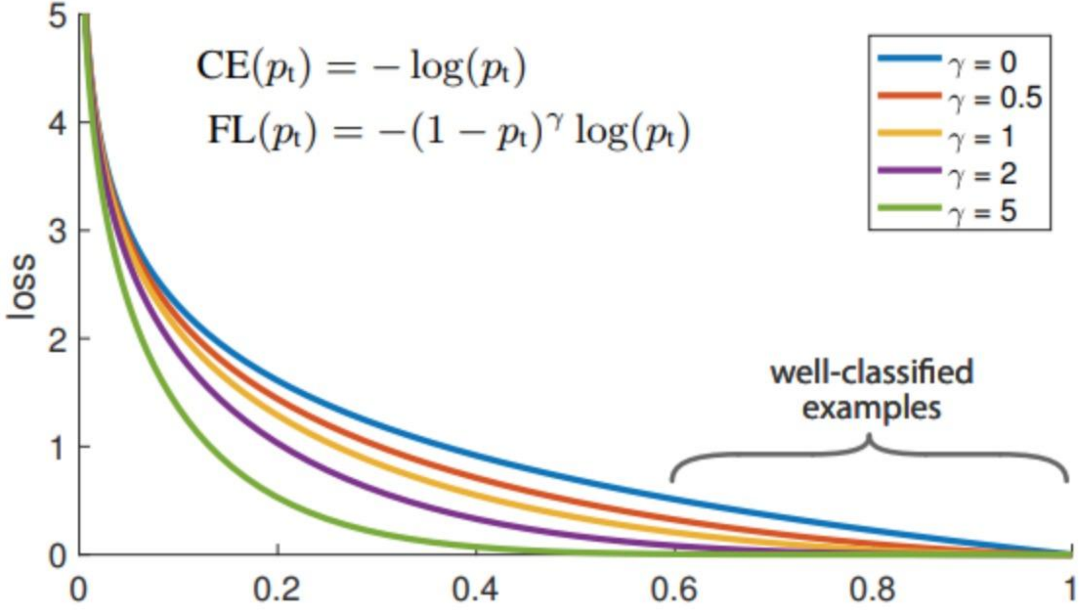
- Focal loss
  - Cross entropy loss
    - Easy cases contribute to the majority of the loss and the class imbalance make it worse.
  - Focal loss
    - Enhance the ability of distinguishing difficult cases.

$$L_f(p_t) = -(1 - p_t)^{\gamma} log(p_t)$$

Where p_t is the estimated probability, ϒ is a tunable parameter.

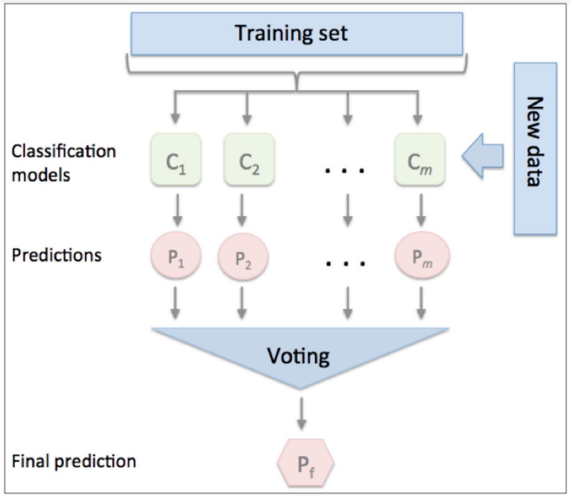
Our Method

- Focal loss vs Cross entropy loss



Our Method

- Ensemble learning
  - Observation：Average models to reduce model variance
  - Solution：Combine many simple “weak” classifiers into a single “strong” classifier.



- One possible way:
  - Get scores from multiple models.
  - Voting/average to get the final score.

Results

- Different feature extraction + random forest classifier.

Table 1: Different feature extractions for random forest

	Pre-trained	Fine-tuned	Fine-tuned with focal loss
AUC	0.94065	0.95207	**0.95327**

- Conclusion:
  - Fine tune is useful when considering a new dataset.
  - Focal loss improves the performance.

Results

- Ablation study on focal loss.

Table 2: Ablation study on focal loss

	Random Forest	wide-ResNet	wide-ResNet+Focal Loss	ResNeXt	ResNeXt+Focal Loss
AUC	0.95327	0.99612	0.99585	0.99616	**0.99831**

- Ablation study on ensemble methods:

Table 3: Ablation study on focal loss

	two ResNeXt	two ResNeXt+wide-ResNet	four models
AUC	**0.99844**	0.99820	0.99794

Conclusions

- Take-home information
  - Preprocess is important for a stable model.
  - End-to-end learning performs better.
  - Focal loss helps to solve the class imbalance.
  - Ensemble method is helpful to obtain a strong classifier from some weak classifiers.
- Challenge and future work
  - The simple voting method does not improve the performance much.
  - Possible solution: Better ensembling methods, such as bagging, adaboosting and so on.

Thanks