Integrating CNN and Attention Mechanism: A Breakthrough Approach to Wafer Map Pattern Classification

Ming-Yu Huang, Che-Yu Lin, Yu-Hsuan Ko, Chia-Hsin Lee

Dept. of Industrial Engineering and Engineering Management
National Tsing Hua University

Abstract

In the research of semiconductor wafer defect classification, the challenges lie in the detection and differentiation of various types of defects. The core issues involve using deep learning techniques to enhance classification accuracy, effectively detecting and identifying defects on wafer maps. Defining the potential defect types on wafer maps and establishing concrete classification criteria pose significant questions.

In addition, the developments of both CNNs and attention machanism on computer vision come up with numerous possible approaches. In this research, we dedicate to analyze the performances between CNNs, attention based models and the proposed model comprising both mechanisms. Additionally, establishing not only a model containing the benefits of both of the structures but constructing a more effective model are also our aspirations. We aim to develop a robust and accurate wafer defect classification model based on the Dual Attention Network (DANet) architecture. Make the model capable of detecting and distinguishing various potential defect types effectively.

The research framework encompasses a literature review, data collection, data preprocessing, model architecture design, and results analysis.

Keywords: Deep Learning, Smart Manufacturing, Wafer Recognition

Chapter 1 Introduction

1.1 Background and Motivation

Integrated circuits (IC) serve as the backbone of modern electronics, driving global technological advancements. As technology progresses and the demand for miniaturization grows, flawless wafer production becomes increasingly essential. Wafer map defect patterns contain crucial information about semiconductor manufacturing, and effective defect analysis technology is crucial for enhancing product yield. Instead of Manual defect marking, the semiconductor manufacturing industry urgently needs accurate, real-time quality monitoring and control.

Recent studies[2] emphasize the benefits of utilizing a stacking ensemble classifier, combining Manual Feature Extraction (MFE) and CNNs for wafer defection classification. This approach enhances the reliability and performance of automated wafer map defect classification. Additionally, another research suggests that Visual Transformer (ViT), hold promise in computer vision tasks. Tests comparing ViT and Residual Network (ResNet) in computer vision tasks[4] provide an opportunity to explore alternative approaches that may enhance defect classification in wafer maps.

1.2 Research Objectives

Motivated by the advancements of attention mechanisms on computer vision, our research aims to utilize deep learning techniques to improve defect pattern classification on wafer maps, enabling more effective detection and recognition. To achieve less parameters, faster execution time and classification effectiveness simultaneously, the study seeks to measure the efficacy of different models, especially the modified version of the DANet[8]. This version, enhanced with a pretrained ResNet backbone integrated with an attention mechanism, efficiently captures both local and global features. This combination improves the model's performance on specific patterns and significantly reduces training time, making it a promising tool for wafer defect classification.

To validate the proposed deep learning model, the research will deploy the WM-811k Wafer Map dataset for thorough testing. This exploration aims to advance semiconductor manufacturing processes, fostering heightened efficiency, reliability, and yield in wafer production.

Chapter 2 Related work

2.1 Computer Vision

The Transformer architecture, initially designed for Natural Language Processing (NLP), has been extended to computer vision tasks by Dosovitskiy[6]. Unlike CNN-based methods, this adaptation of the Transformer model directly employs sequences of image patches for image classification, which in turn causes the better performance on capturing semantic information of images. While ViT may show inferior performance to ResNet on moderately sized datasets, it excels with substantial pre-training data, such as ImageNet-21K and JFT-300M. In these cases, ViT outperforms CNNs, overcoming inductive bias limitations and demonstrating favorable transfer effects in downstream tasks[4].

2.2 Imbalanced Data

Imbalanced data is prevalent in wafer map defect classification, exacerbated by limited data. The problem is also found in the widely used WM-811K dataset, with the "None" class occupying 85.2% of the dataset. Common approaches to address imbalance, like random undersampling, risk losing valuable information from the majority class. To counter this, we adopt the F1 score as the primary evaluation measure, as it provides a balanced approach and mitigates the impact of class imbalance. This decision aligns with the practices of previous studies in wafer map pattern classification under class imbalance[11][2][12].

2.3 Wafer Map Pattern Classification

Wafer map pattern classification in semiconductor manufacturing aims at accurate defect identification. The stacking ensemble approach advocated by Kang and Kang[2] combines MFE and CNNs, where MFE extracts features for a classifier with traditional machine learning methods, and CNNs excels in automated feature learning. Previous research[14][15][13] highlights the effectiveness of deep CNNs as feature extractors, transforming images into high-level, abstract representations for image classification tasks.

Recent studies also demonstrate that models with attention mechanisms, like ViT, are adept at processing global information. Drawing inspiration from these methodologies, we adapt the DANet[8]. Our modified model incorporates a ResNet backbone with an attention mechanism, effectively capturing both local and global features. Complemented by a Fully Connected Network (FCN), this structural foundation enhances the model's overall performance in image analysis and classification tasks.

Chapter 3 Method

3.1 Overview of Dual Attention Network

Proposed by Jun Fu[8], DANet was designed to integrate local features with their global dependencies for scene segmentation. Two types of attention modules are introduced for building the association among features in order to explore the global contextual information. The two attention modules, Position Attention Module (PAM) and Channel Attention Module (CAM), were applied after the pretrained residual network with dilated strategy. That is to say, the local features processed by the ResNet would be fed into two parallel attention modules and eventually aggregated to obtain better features.

3.2 Position Attention Module and Channel Attention Module

The two attention modules are the primary algorithms of DANet, we adopt those two attention modules in our model without any adjustment. Hence, it is necessary to briefly introduce the design logics and mathematics in this part for better comprehension of DANet here created for wafer map classification.

3.2.1 Position Attention Module (PAM)

Given a local feature **A**$\in\mathbb{R}^{C\times H\times W}$, we can get two new feature maps and from the convolution layers, where {**B**, **C**}$\in\mathbb{R}^{C\times H\times W}$. Then reshape them into $\mathbb{R}^{C\times N}$, which $N = H \times W$, they perform matrix multiplication between transpose of **C** and **B**, and apply a softmax layer to obtain the spatial attention map **S**$\in\mathbb{R}^{N\times N}$:

$$s_{ij} = \frac{exp(B_i \cdot C_j)}{\sum_{i=1}^{N} exp(B_i \cdot C_j)} \tag{1}$$

Local feature **A** is also fed into a convolution layer to generate a new feature map **D** $\in\mathbb{R}^{C\times H\times W}$ and reshape it to $\in\mathbb{R}^{C\times N}$ as well. Then they perform a matrix multiplication between **D** and **S** and reshape it back to $\mathbb{R}^{C\times H\times W}$. Finally, they multiply the result by a scale parameter α and perform a element-wise sum operation with **A** to get the final output **E**$\in\mathbb{R}^{C\times H\times W}$:

$$E_j = \alpha \sum_{i=1}^{N} (s_{ji} \cdot D_i) + A_j \tag{2}$$

Where α learns weight from 0. The spatial attention map **S** provides the correlation between feature map **B** and **C**. Thus, in equation 2 the element-wise sum operation enables features **E** to have a global contextual view and the selectively aggregate contexts.

3.2.2 Channel Attention Module (CAM)

Given local feature **A** $\in\mathbb{R}^{C\times H\times W}$, they directly calculate the channel attention map **X** $\in\mathbb{R}^{C\times C}$ as:

$$x_{ij} = \frac{exp(A_i \cdot A_j)}{\sum_{i=1}^{C} exp(A_i \cdot A_j)} \tag{3}$$

Then conduct the matrix multiplication between transpose of **X** and **A** and reshape to $\mathbb{R}^{C\times H\times W}$. Similar to PAM, they multiply the result by a scale parameter β and perform the element-wise sum operation with original features **A** to get the final output **E**$\in\mathbb{R}^{C\times H\times W}$:

$$E_j = \beta \sum_{i=1}^{C} (x_{ij} A_i) + A_j \tag{4}$$

Where β learns weight from 0. According to equation 3, the feature map **X** constructs the similarity between channels of local feature maps, which in turn enable features **E** to own the better view of semantic dependencies in the aspect of channels.

3.3 Dual Attention Network Modification

DANet concatenates CNNs and attention modules together for enhancing the global semantic contextual from local features processed by CNN. Specifically, the 2 types of attention modules not only fortify the ability for catching global features but also increase the representation capability of the model in the aspects of both positions and channels. In addition, the attention modules extend only little numbers of parameters, which could attain better performance with a relatively smaller size of model. Considering the competence, we remain the PAM and CAM but change the backbone of it and add fully connected layers for the task of wafer map pattern classification. In conclusion, our model comprises three primary parts: feature extractor, 2 types of attention modules and fully connected layer. After this section, the DANet mentioned represents the one modified as shown in Fig. 1.

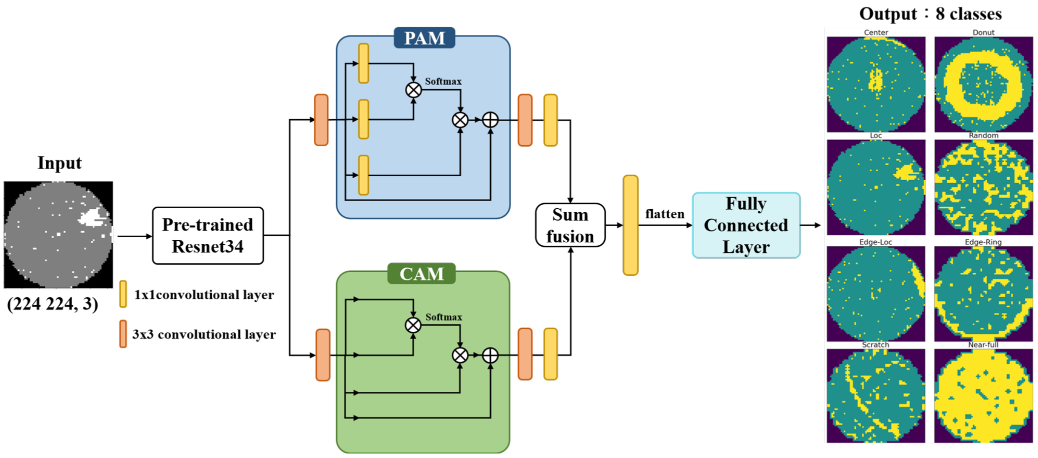


Fig. 1. Architecture of the Proposed Model

3.3.1 Feature Extractor

DANet is first designed for semantic segmentation, hence, the backbone of it was residual network with dilated strategy. Referring to the work by Liang-Chieh Chen[9], they introduced the atrous convolution to re-purpose the model from classification to image segmentation. Therefore, we apply the residual network with non-dilated strategy as the backbone of our model here to retain certain abilities to detect the refine feature. In order to use less parameters to get better performance in comparison with ViT, Resnetv2_50, Resnetv2_101, we choose Renset34 as the backbone. In addition, Resnet34 is pre-trained on ImageNet1K_V1 and also we remove the last adaptive average pooling layer and the last linear layer. Thus, we denote the pre-trained weights of ImageNet1K_V1 for Resnet34 as θ. The remaining part of Resnet34 with the weights θ would represent as the feature extract function f_{θ}^{ResNet} .

Take N wafer maps $X \in \mathbb{R}^{3\times H\times W}$ as input and each label $\boldsymbol{y} = \{\mathbf{0},\mathbf{1},\mathbf{2},\ldots\mathbf{7}\}$. The whole dataset $D = \{\boldsymbol{X_i}, \boldsymbol{y_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{N}}$, the relation between X and extracted features $A \in \mathbb{R}^{2048\times H\times W}$ would be:

$$A = f_{\theta}^{ResNet}(X) \tag{5}$$

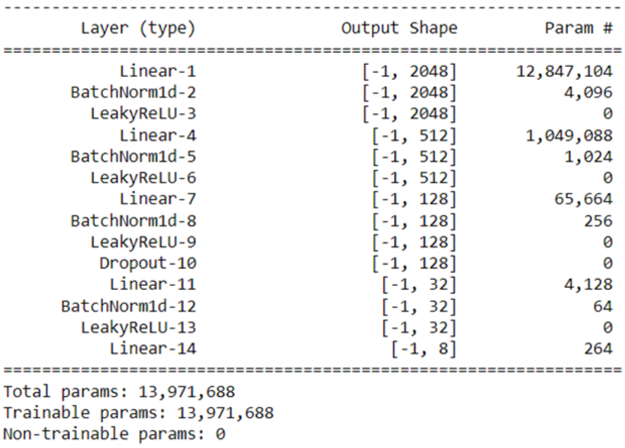
3.3.2 Attention Module and Convolutional Layer

Identical to the original DANet, we feed the extracted features A into a convolutional layer to generate local features B and C where their spatial dimensions decrease to 1/8 of A. We represent PAM and CAM as function f^{PAM}, f^{CAM} respectively. Since both attention modules have not been pre-trained on any dataset, we do not add the weight terms in those functions. Subsequently, we create 2 CNN blocks, CNN^{1} and CNN^{2}, with only 2 convolutional layers to make our model more robust to variations before entering the fully connected layer. Eventually, operate the sum between two output features to generate the features E. The mathematical representation would be:

$$E = CNN^{1}(f^{PAM}(B)) + CNN^{2}(f^{CAM}(C)) \tag{6}$$

3.3.3 Fully Connected Layer

After passing through a single convolutional layer and flattening the results G, we feed the resulting features into fully connected layers particularly designed for wafer map pattern classification. The number of neurons in the input layer is equal to the size of the input feature, and then the next layer would be designed as. The number of neurons in the rest layers would be divided by 4 until 32 consecutively. Note that we settle batch normalization and Leaky ReLU between the layers. More specific structure is shown in Fig. 2.



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                 [-1, 2048]      12,847,104
       BatchNorm1d-2                 [-1, 2048]           4,096
         LeakyReLU-3                 [-1, 2048]               0
            Linear-4                  [-1, 512]       1,049,088
       BatchNorm1d-5                  [-1, 512]           1,024
         LeakyReLU-6                  [-1, 512]               0
            Linear-7                  [-1, 128]          65,664
       BatchNorm1d-8                  [-1, 128]             256
         LeakyReLU-9                  [-1, 128]               0
          Dropout-10                  [-1, 128]               0
           Linear-11                   [-1, 32]           4,128
      BatchNorm1d-12                   [-1, 32]              64
        LeakyReLU-13                   [-1, 32]               0
           Linear-14                    [-1, 8]             264
================================================================
Total params: 13,971,688
Trainable params: 13,971,688
Non-trainable params: 0
```

Fig. 2. Fully Connected Layers for Wafer Map Pattern Classification

Chapter 4 Experiments

4.1 Data description

We performed research using the WM-811K dataset, acquired from a semiconductor producer (MIR Lab, 2018), containing 811,457 wafer map samples. Out of these, 25,519 maps were annotated with defect types by specialists in that field. These annotated maps were utilized for our experimental analyses. Every wafer map was categorized into one of nine established defect categories: Center, Donut, Edge-Ring, Edge-Local, Local, Random, Near-Full, Scratch, and None. Table 1. shows the number of wafers in each pattern. Fig. 3. provides illustrations of wafer maps for each defect category. Each category corresponds to distinct issues within the production process with imbalanced distribution.

Pattern	Numbers of wafers
Center	4294
Donut	555
Edge-Loc	5189
Edge-ring	9680
Loc	3593
Random	866
Scratch	1193
Near-full	149
Total	25519

Table 1. Number of wafers in each pattern

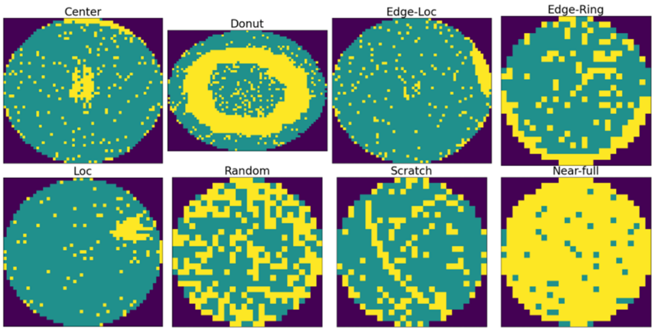


Fig. 3. Wafer maps with different defect classes

4.2 Experimental settings

Each wafer map is represented by a two-dimensional grid, in which each cell is marked as 1 for a defective die and 0 for a non-defective one. Since the actual wafer maps are circular, all the cells that fall outside of this circular area are assigned a value of 0. Furthermore, due to the different sizes of wafers in the dataset, we standardized each wafer map by resizing them to the shape of (224,224) before using them as inputs for models. To evaluate the performance of our experiments, we conducted ten independent replicates, each with a particular random seed. From the 25,519 labeled wafer maps with defect types, we selected 80% to form the training dataset and 20% to comprise the testing dataset for performance evaluation by applying the random split under given random seed. In our experiments, we selected ResNet (ResNet50 and ResNet101), ViT and DANet , which consist of ResNet and attention mechanisms, for comparison with our proposed model. Regarding the hyperparameters of the models, we employed an Adam optimizer with a learning rate of 1e−4 and a batch size of 32. Concerning the number of epochs, we set DANet to train for 50 epochs, while the other pre-trained models were trained for 15 epochs. Our experiments are conducted using the Tesla V100-SXM2-16GB GPU.

To evaluate the classification performance for each category, we computed the F1-score for the respective classes. The F1-score represents the harmonic mean of precision and recall. This measure is a beneficial way to achieve a balance between precision and recall. For assessing the overall classification performance, as discussed in Section 2.2, we acknowledged the issue of class imbalance present in our dataset, which could impact the accuracy of our evaluation. To address this, we employed macro-average F1 (F1-macro) on the test dataset. The way is widely recognized for evaluating wafer map pattern classification in scenarios with class imbalance [17][11], which is described in Eqs. (7)(8)(9). Let C be the set of all classes. For each class i belonging to C, that is $\forall i \in C$:

$$P_i = \frac{True\ positive_i}{True\ positive_i + False\ positive_i} \tag{7}$$

$$R_i = \frac{True\ positive_i}{True\ positive_i + False\ negative_i} \tag{8}$$

$$F1_{macro} = \frac{1}{C}\sum_{i=1}^{C}\frac{2 \times P_i \times R_i}{P_i + R_i} \tag{9}$$

F1-macro is calculated for each class independently, and then takes the average of these scores. It means giving equal weight to each class regardless of its frequency. It can help us ensure that the performance on less frequent classes has a proportionate impact on the overall metric.

4.3 Results

The experimental results presented below are derived from ten replications, with the mean and standard deviation of each metric reported for comparative clarity. In these tables, we compared the best model and the runner-up model based on the outcomes.

Table 2. Evaluation of model performance by F1 Scores across categories

Class	Resnet50[1]	Resnet101[1]	ViT[6]	DANet
Center	0.9656±0.0074	0.9655±0.0070	**0.9663±0.0039**	0.9657±0.0034
Donut	0.8478±0.0466	0.8476±0.0574	0.8624±0.0361	**0.8761±0.0305**
Edge-Loc	**0.9256±0.0072**	0.9203±0.0072	0.9202±0.0077	0.9114±0.0110
Edge-Ring	**0.9867±0.0024**	0.9862±0.0019	0.9859±0.0036	0.9832±0.0035
Loc	**0.8696±0.0094**	0.8673±0.0168	0.8547±0.0148	0.8655±0.0090
Near-full	0.9304±0.0394	0.9123±0.0275	**0.9505±0.0347**	0.9076±0.0364
Random	0.9186±0.0189	0.8882±0.0149	**0.9265±0.0164**	0.9159±0.0127
Scratch	0.8783±0.0112	**0.9254±0.0623**	0.8740±0.0258	0.8684±0.0200
Overall performance matric				
F1-macro	0.9153±0.0106	0.9141±0.0171	**0.9176±0.0084**	0.9117±0.0091

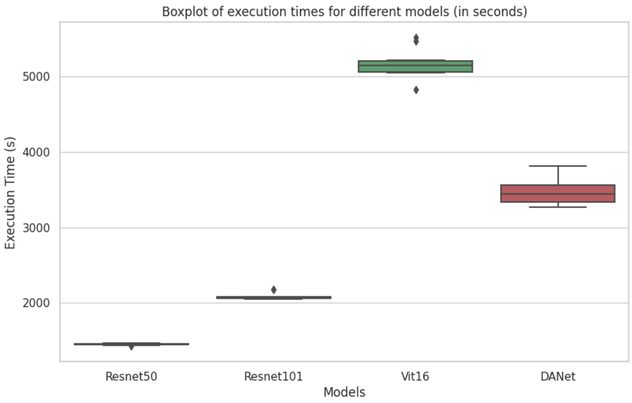
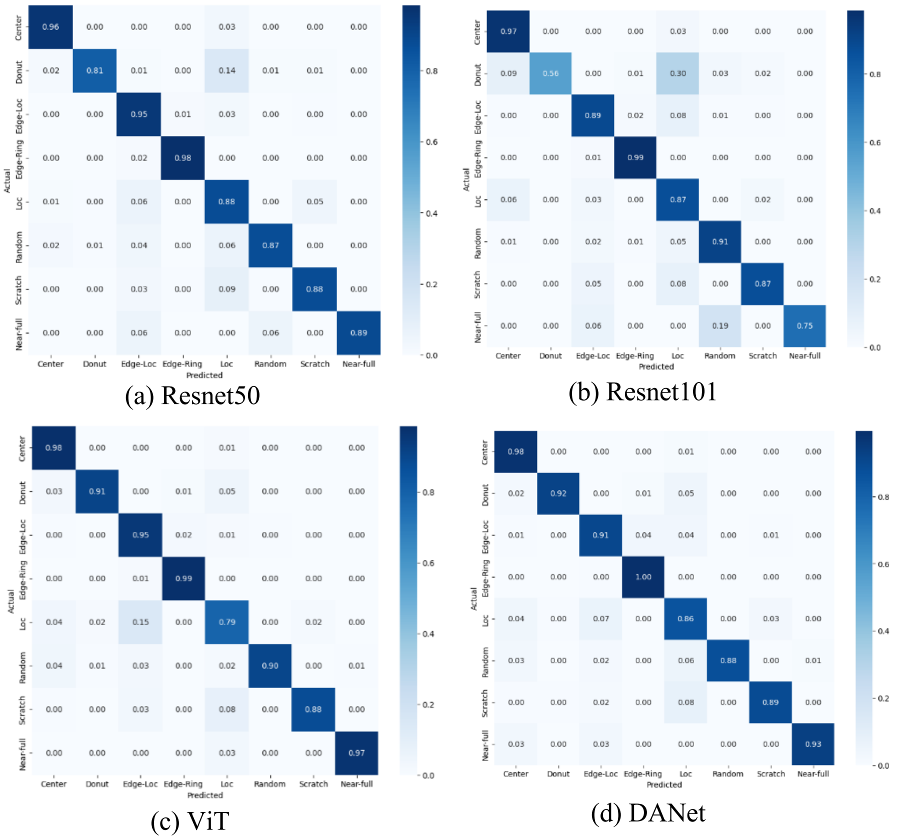


Fig. 5. Execution time of models



(a) Resnet50

(b) Resnet101

(c) ViT

(d) DANet

Fig. 6. Confusion Matrix of models under random seed 46

Fig. 5. illustrates the execution time of our four models, providing insights into the model's computational efficiency. Execution time of ViT is significantly higher, as shown by its boxplot's position on the graph, with a much wider spread of data points, including some outliers. It indicates that ViT has an inconsistency in the performance of execution time. DANet exhibits a median execution time that is higher than both ResNet models but lower than ViT. Its interquartile range suggests good consistency in execution time. Moreover, the absence of outliers for DANet suggests that our model provides a stable performance without the extremes of variability seen in ViT. The findings suggest that DANet is a promising model for the wafer map pattern classification task, offering a trade-off between accuracy and computational time.

Table 1 shows that performance of DANet is consistent across different classes, often closely approaching the best scores. Specifically, for the "Donut" class, DANet indeed achieves the highest F1-score, and for the "Edge-Ring" and "Center" classes, it performs slightly below the best but is still competitive. As far as "Loc" class is concerned, notwithstanding Renset50 dominates, performance of DANet is better than the one of ViT. Furthermore, we also focus on the overall metrics, F1-macro scores, for model evaluation. Overall F1-macro of DANet has good performance with 0.9117, yet they do not surpass the highest scores achieved by the F1-macro of ViT with 0.9176 but close to it. Fig. 6. shows one of the experiments conducted by DANet, the balanced classification performance suggests that the model is robust and not biased towards any particular class in this experiment. Due to the heatmaps and f1-scores, we discern there is a tendency of ViT to predict "Loc" as "Edge-Loc" easily, while ResNets would tend to predict "Donut" as "Loc". The results ensure the drawback of ViT to discriminate refine local features and the downside of ResNet to capture global semantic features. In the heatmap of DANet, those two tendencies do not appear, which in turn interprets DANet conclude both of the benefits of CNNs and attention mechanism on classify given classes.

Chapter 5 Conclusion

CNNs excel at capturing local information but struggle with global information. This limitation is evident in categories like "Donut" class, which require a global perspective for effective classification. In our research, we observed significant improvements in this aspect with ViT. Our proposed model modified from DANet further enhances this capability, outperforming both CNNs and ViT in capturing global information. However, ViT does not perform as well in categories heavily reliant on local features, such as "Loc" class. Here, our model shows slightly better performance, likely due to its ResNet with non-dilated strategy, which combines the strengths of CNNs and attention mechanisms.

Remarkably, DANet requires only about 67% of the training time compared to ViT models, yet achieving similar or even better performance. This efficiency is crucial in practical applications like manufacturing and technology industries, where it can reduce computational costs and improve efficiency. We believe future research could explore architectures combining CNNs with attention mechanisms for image classification tasks. Such models could offer enhanced accuracy, scalability, and robustness when dealing with large training datasets, opening up new avenues for development in the field.

Reference

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

Deep Residual Learning for Image Recognition

[2] Hyungu Kang, Seokho Kang
A stacking ensemble classifier with handcrafted and convolutional features for wafer map pattern classification

[3] Tongwha Kim, Kamran Behdinan
Advances in machine learning and deep learning applications towards wafer map defect recognition and classification: a review

[4] Srinadh Bhojanapalli, Ayan Chakrabarti , Daniel Glasner, Daliang Li, Thomas Unterthiner , Andreas Veit, Google Research
Understanding Robustness of Transformers for Image Classification

[5] Alexander Kolesnikov ,Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, Neil Houlsby
Big Transfer (BiT): General Visual Representation Learning

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby
An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

[7] Tsung-Han Tsai, Chieng-Yang Wang
Wafer Map Defect Classification using Deep Learning Framework with Data Augmentation on Imbalance Datasets

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, Hanqing Lu
Dual Attention Network for Scene Segmentation

[9] Liang-Chieh Chen, George Papandreou, Senior Member, IEEE, Iasonas Kokkinos, Member, IEEE, Kevin Murphy, and Alan L. Yuille, Fellow, IEEE
DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

[10] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, Alexey Dosovitskiy
Do Vision Transformers See Like Convolutional Neural Networks?

[11] Hyungu Kahng and Seoung Bum Kim , Member, IEEE
Self-Supervised Representation Learning for Wafer Bin Map Defect Pattern Classification

[12] Tsutomu Ishida1 , Izumi Nitta1 , Daisuke Fukuda1 , Yuzi Kanazawa1
Deep Learning-Based Wafer-Map Failure Pattern Recognition Framework

[13] Jyostna Devi Bodapati, N. Veeranjaneyulu
Feature Extraction and Classification Using Deep Convolutional Neural Networks

[14] Matthew D. Zeiler, Rob Fergus
Visualizing and Understanding Convolutional Networks

[15] Suresh Dara1, Priyanka Tumma
Feature Extraction By Using Deep Learning: A Survey

[16] Jianchuan Huang, Kuo-Yi Lin, Jia Xu, Lili
Imbalanced Wafer Map Dataset Classification with Semi-Supervised Learning Method and Optimized Loss Function

[17] Saqlain, M., Abbas, Q., Lee, J.Y.
A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes.