Wafer Map Defect Classification using Deep Learning Framework with Data Augmentation on Imbalance Datasets

Tsung-Han Tsai (han@ee.ncu.edu.tw)
Dept. of E. E., National Central Univ. https://orcid.org/0000-0001-7524-0621
Chieng-Yang Wang
National Central University

Research

Keywords: G2LGAN, Data augmentation, Generative adversarial network, Wafer Map Defect classification

Posted Date: September 29th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2078809/v1

Abstract

Wafer map defect classification is a key task for the semiconductor industry to improve the yield rate. Most wafer map defect classifications suffer from the problem of data imbalance and insufficient data. This paper proposes a global-to-local generative adversarial network (G2LGAN) method using the deep learning framework. It extracts global features and local features separately to generate effective data even in the imbalanced dataset. We use random under-sampling to suppress the majority class of data. We use MobilenetV2 as the classifier, and use two datasets for validation. One is open dataset 1WM-811K and the other is called 21-Defect built from the industry. Based on the serious dataset imbalance problem, this paper integrates data enhancement and random undersampling methods to optimize the dataset and uses the proposed classification network for classification tasks. The results of WM-811K dataset show that the proposed method has a classification accuracy of 98.39 and an F1-Score of 93.01. We also conduct cross-validation on the 21-Defect dataset and the results show that the proposed method has good robustness.

I. Introduction

Semiconductor manufacturing technology has greatly improved in recent years. The complexity of the process and the environment, human error and machine problems still make wafer defects unavoidable. The wafer map defect classification is essential to improving production yields. The defect pattern on the wafer map will show some important information about the production problem [1]-[2] and the root cause of failures.

Wafer defects can be classified into local defects and global defects [3]. Local defects are usually caused by a process problem that results in a regular distribution of defects at a certain location on the wafer. For example, uneven application of photoresist can lead to defects at the edges of the wafer. Global defects are caused by environmental problems such as dust, temperature, and humidity in space. As caused by these factors, they are evenly distributed throughout the wafer and can obscure local defects making defect classification more difficult. Therefore, some studies are attempting to eliminate overall defects. The engineers analyze the wafer map to find out the cause of the failure, and then take corresponding measures to solve the problem and improve the process yield. Nowadays, it is still relied on engineers to classify wafer map defects based on their experience. By manual classification, it requires additional labor costs and it mostly relies on the experience and subjective judgment of the engineer.

Nowadays, convolutional neural networks are a reliable method for image classification [4]-[5]. It uses convolution computation to extract the features of different dimensions from the image. In the task of image recognition, the convolutional neural networks even can operate more accurately than humans. However, neural networks have several problems to be overcome [6]. It always need a large amount of training data and a large number of parameters and calculations in neural networks. Also the problem of insufficient data is particularly serious in the task of wafer map defect classification. Generally the wafer

map implies commercially confidential information that companies will not easily disclose. Also the cost of data labeling is very high, so data with labels is very scarce.

In this paper, we apply the neural network to classify the defect types. To solve the problem of insufficient data, we propose an efficient method called global-to-local generative adversarial network (G2LGAN) to expand the limited data. We suppress the influence of the majority class on the model by undersampling. We also increase the data of the minority class by data augmentation to balance the data set. The difference from traditional generative adversarial network (GAN) [7] is that the proposed method can effectively generate data of conditional class on imbalanced datasets. We develop a neural network architecture using MobileNetV2 [8] as the backbone network with the less number of model parameters and calculations. The proposed method outperforms the high accuracy and low number of parameters and computation with other proposed approaches.

We organize the rest of the paper as follows. The related work for data augmentation and wafer map defect classification is provided in Section II. Section III introduces the details of Fig. 1. Data distribution of WM-811K. The dataset is composed of real-world wafer maps and labeled by experts in IC industry.

the training dataset, data pre-processing, GAN architecture, and classification network. In Section VI, the experimental results is shown and compared with the other methods. Finally, the conclusion is provided in Section V.

Ii. Related Work

A. *Imbalanced datasets*

When using neural networks for classification tasks, imbalanced datasets will cause the model classification results to lean toward the majority class. For imbalanced datasets, various resampling strategies have been used. One common method is random undersampling [9] which reduces the amount of data in the majority class by discarding data. However, overuse of undersampling causes overfitting and removal of potentially useful data. One of the common open datasets is the WM-811K [10] which is widely used in many papers. As shown in Fig. 1, it suffers from the imbalance classes. The majority class “None” and the minority class “Near-Full” in the WM-811K dataset account for 85.24% and 0.086% of the entire data set, respectively. Thus only using random undersampling has a limited effect on improving the performance of the model.

B. *Data augmentation*

Typical data augmentation is used to perform random rotations, flips, and translations as well as the addition of Gaussian noise to the original dataset. These transformations can produce more data from the existing data. Recently, generative models based on neural networks have been proposed. The most common methods are variational autoencoder (VAE) [11] and GAN. VAE used an encoder and decoder to down-sample and up-sample the data and added noise to the hidden layer to enable the network to

generate new data instead of simply reconstructing the data. Because VAE needs to be differentiated by hidden units, it cannot have discrete latent variables.

GAN was widely applied in many areas. Due to the scarcity of wafer defect images, people expect that the excellent image generation capabilities of the GAN model can assist the classification model to achieve higher performance. [12] conditioned GANs on discrete labels. They changed the binary probability in the traditional GAN to conditional probability to generate conditional data. Mariani *et al.* [13] designed the balancing GAN (BAGAN) to resolve unbalanced datasets by applying class conditions in the potential space to push the generation process to the target class. Yu *et al.* [14] proposed the Multiple Granularities GAN (MGGAN) by feeding the multigranularity feature learned with an auxiliary feature extractor into the generator. Obviously, data augmentation has a significant impact on the convergence and accuracy of deep neural networks.

C. *Wafer map defect classification*

Some researchers have applied deep learning methods to wafer map classification. Wu *et al.* [10] extracted Radon-Based Features and Geometry-Based Features from wafer maps and used support vector machines (SVM) to classify wafer map defects. However, SVM is less effective on multiple classification tasks and difficult to train on large-scale data. Ishida *et al.* [15] proposed a data augmentation technique to reduce the noise of random defects by Hough transform so that the wafer map can retain the original features. Batool *et al.* [16] pointed out the high imbalance in the existing dataset, and the team proposed an undersampling approach. They selected 400 images from each category to participate in classifier training and validation and built a DCNN model for feature extraction. The results show that the used undersampling can effectively improve accuracy and maintain good results even with a small amount of data.

Alawieh *et al.* [17] used deep selective learning for wafer map defect pattern classification. Their method has an integrated rejection option where the model chooses to avoid predicting class labels when the risk of misclassification is high. To solve the class imbalance problem in wafer map classification, the team proposed a data enhancement framework around a convolutional autoencoder model. Han *et al.* [18] proposed an autoencoder to augment the data of the wafer map and use MobileNetV1 for classification. Meanwhile, autoencoder has the shortcomings of being difficult to train and generating images with poor resolution. Yu *et al.* [19] proposed the conditional two-dimensional principal component analysis algorithm to extract more effective features from the imbalanced wafer maps. It preserves more spatial information in comparison to the conventional 1DPCA.

Iii. Proposed Approach

The main problems with wafer defect classification are insufficient data and data imbalance. In this section, we introduce the current situation of wafer map datasets, the problems encountered, and the corresponding solutions. We apply some preprocessing methods to datasets. Then we introduce the

proposed GAN model. Finally, we use the classification network which maintains high accuracy with a low number of parameters and low computational effort.

A. *Preprocessing on Datasets*

In this paper, we use the WM-811K dataset to train our model. This dataset has 811457 wafer maps divided into 9 classes. Among them, only 3.1% (25519 wafers) of the entire dataset have actual defect patterns, 18.2% (147431 wafers) are marked as "None", and 78.7% (638570 wafers) are unmarked. It shows that WM-811K has a serious data imbalance problem. The learning goal of a general classifier is to optimize the accuracy by minimizing the loss function to optimize the model. When the dataset is unbalanced, the classifier will tend to judge the output as majority classes. This reduces the influence of a minority class and makes it difficult for the classifier to learn from the patterns of the minority class.

Algorithm 1 Random undersampling

Input:
Training data set: $D_{total} = \{D_0, D_1, \dots, D_c\}$
// $D_0, D_1, \dots, D_c$ are sorted in descending order by the number of samples in each classes
Parameter setting:
epoch: # of training for the entire training set
iteration: # of examples in the D_0 / batch size
n_c: # of classes
n_1: ⌈batch size / n_c⌉
n_2: ⌊batch size / n_c⌋
c_median: batch size % n_c
Our proposed random undersampling:
for i=0; i<epoch; i++ **do**
  D_{total} = Random shuffle (D_{total})
  for b=0; b<iteration; b++ **do**
    initialize list d_{total}
    for c=0; c< c_median; c++ **do**
      append $D_c[b \times n_1 : (b+1) \times n_1]$ to d_{total}
    for c_median; c< n_c; c++ **do**
      append $D_c[b \times n_2 : (b+1) \times n_2]$ to d_{total}

We use different methods for different categories to balance the dataset. For the majority class, we modify the traditional random undersampling. We adjust the number of all classes in each epoch instead of deleting the data of the majority class randomly. The proposed random under-sampling details are shown in Algorithm 1. We balance the number of each class in each iteration, where the number of iterations is determined by the minority class. In this way, all the data in the minority class can be trained. We shuffle the training data at the beginning of each epoch so that the remaining data in the majority class can be trained in the subsequent epochs. For the minority class, we use GAN to augment the data. Overall, we balance the dataset by suppressing the data of the majority class and generating new data of the minority class to increase the classification model accuracy.

As shown in Fig. 2, each point in the original wafer map is composed of three states: 0, 1, and 2, where 0 means None, 1 means Pass, and 2 means Fail. These three states are independent of each other, but the distance between them is not fixed. As a result, it causes uneven penalties when the neural network propagates backward. We used a hot encoding to fix the distance between them to 1 to make the network converge better. It maps one-dimensional data to three-dimensional orthogonal coordinates to increase the dimensionality of the data and balances the penalty of the loss function between the three types.

B. Data Augmentation

The data augmentation network is to solve the problem of insufficient training data and uneven distribution among the dataset classes to improve the performance of the classification task. Here we present our modifications to the network architecture of the deep convolutional generative adversarial network (DCGAN) [20] and the new training strategy proposed in this paper. DCGAN is an extension of the original GAN. As shown in Fig. 3, it consists of two sub-models: the generator and the discriminator. The purpose of the discriminator is to determine whether the input images are the real images from the dataset or the fake images generated by the generator. Therefore, the generator needs to generate images that be similar to the real image to confuse the discriminator and then train each other through a zero-sum game. The generator of DCGAN has a major disadvantage. Since deconvolution with stride 2×2 is used to do up-sampling, it results in checkerboard artifacts [21]. This occurs when the kernel size of the deconvolution cannot be divided by stride.

We choose to resize the image to high resolution first and extract features by convolution to avoid the checkerboard artifacts. The loss function of GAN is shown as follows:

$$min_G max_D \left\{E_{x\sim p_{data}(x)}[\log(D(x))] + E_{z\sim p_z(z)}\left[\log\left(1 - D(G(z))\right)\right]\right\} \quad (1)$$

Where $p_z(z)$ is a prior on input noise variables; $G(z)$ is the generator that maps a sample *z* drawn from a distribution $p_z(z)$ to the data space; $p_{data}(x)$ is the real data distribution; $D(x)$ represents the probability that *x* came from the data rather than the generator. According to [23], under the optimal discriminator, minimizing the generator's loss is equivalent to minimizing the Jensen-Shannon (JS) divergence between p_{data} and p_G. Since p_{data} and p_G are almost impossible to have an overlap, the JS divergence is always approached constantly $\log 2$. Finally, it leads to the disappearance of the generator gradient. We replace the original loss function with Hinge Loss of (2).

$$\begin{aligned} min_G max_D \{ & E_{x\sim P_{data}(x)}[max(0, 1 - D(x))] \\ & + E_{z\sim P_z(z)}[max(0, 1 + \mathrm{D}(G(Z)))] \\ & - E_{z\sim P_z(z)}[\mathrm{D}(G(Z))]\} \end{aligned} \quad (2)$$

Hinge Loss was widely used in SVM for binary classification. The goal of the discriminator in GAN is also to perform binary classification of the real image and fake image so that it also has a good performance

on GAN tasks.

In the strategy of training GAN, the traditional GAN uses a discriminator to classify whether the input image is true or false. This makes the data generated by GAN without labeled data. In the past, there were three main methods to obtain labeled data. The first is the additional manpower to classify the generated data. The second is to train its generative model for each category. In the case of insufficient data, the model will not converge or the generated data lacks diversity. The third is to send category information into the model for training through architectures like CGAN and ACGAN. However, it will make the minority class less effective than in an unbalanced dataset.

Since different classes of data usually have certain correlations, we call them global features. The unique features of individual classes are called local features. The way we generate conditional classes is to train a GAN for each class. However, when the data is insufficient, GAN will be more difficult to converge. Even if GAN successfully converges, it is easy to produce model collapse.

Due to the general lack of data and imbalance in the wafer map dataset, we propose a new training strategy called G2LGAN, which can generate effective{ images even when the data is imbalanced. G2LGAN divides the training of the GAN into two stages as shown in Fig. 4. The first stage uses the data of each class as training data to train the generative model. We expect that the generative model can learn the global features of the dataset in the first stage. For example, the global features of WM-811K are the outline of the wafer map and some random defects on the wafer. The second stage is to fine-tune the model through various classes. In this stage, the model learns the local features of the class. G2LGAN enables generative models for a minority class to be trained on a better basis rather than using initialized models, and thus can effectively address the lack of generative diversity caused by minority classes in an imbalanced dataset.

Assuming that the training time of the model is proportional to the training data. The number of the training data is D, there are nine classes in D, the amount of data for each class is $1/9*D$, the epoch is e, and the other training parameters are the same. There are two ways to generate labeled data using GAN, the first is to train a GAN model independently for each class. The time cost could be roughly estimated as $T=(1/9*D*e)*9 = D*e$, and the second is a conditional GAN like BAGAN, ACGAN, CGAN. The estimated time cost is $T = D*e$. Our proposed G2LGAN has a time cost $T = D*e/3+(1/9*D*e*2/3)*9 = D*e$. It shows that the time cost of the proposed method is the same as that of the existing method.

Wafer Map Defect Classification Network

More complex models will cause long computation time and low efficiency. However, the simpler models will result in low accuracy and failure to maintain product yield. To balance the computation and accuracy, we use MobileNet V2 as the backbone network. MobileNet V2 has the advantage of a low number of parameters and low computation, but still maintains high accuracy.

The main concept of MobileNet V2 is the inverted residual block, which consists of 1×1 convolution, and depthwise separable convolution. The purpose of the 1×1 convolution is to let the dimension of the feature map raise so that it can reduce information loss caused by non-linear functions. The other is depthwise separable convolution, which can be divided into depthwise convolution and pointwise convolution.

According to [20], the shallow features can only capture local details and shapes, while the deep features have a wider field of view and therefore can capture more global information. In the wafer map defect classification task, we believe that the shape of the defect is more important than the location of the defect. So our architecture stacks the bottleneck layer when the feature map depth is 8 and 16 to extract more shallow features. We take a 64×64 wafer map as input, eliminate the full convolution of the first layer of MoiblenetV2 and directly use the inverted residual block for downsampling. We stack the layers in the shallow network to obtain more information about the wafer defect shapes. The remaining structure is shown in Table I. The loss functions of the network are as follow:

$$Loss_{cross} = \sum_{c=1}^{C}\sum_{i=1}^{n} -y_{c,i}\log_2\left(p_{c,i}\right) \quad (3)$$

where *c* is the number of classes and *n* is the number of samples for a single iteration. $y_{c,i}$ is the ground truth of the *i*-th data. $p_{c,i}$ is the probability that the *i*-th data predicted for class *c*.

Experimental Results

This section will explain the datasets and evaluation metrics. Based on it we can show the implementation results and make some comparisons with other works. Quantitative assessment is also evaluated in detail.

A. Datasets and Evaluation Metrics

This paper uses the WM-811K dataset and 21-Defect dataset to evaluate the effectiveness of the proposed architecture. The 21-Defect dataset, as shown in Fig. 5, is extracted from real wafer maps in the industry to provide more classes by [15]. It includes a total of 16388 wafer maps in 21 categories. Since the number of categories is more than that of WM-811K, and the probability of occurrence of some categories is low, the problem of data imbalance is more serious than that of WM-811K. We expect the proposed method applies to more severe tasks as well. For both datasets, we first use 70% of the dataset as a training set and 30% as a test set. We provide the data from the training set to the GAN for training and merge the generated data with the training set, and then provide the merged data to the classification network for training. The test set is independent and not trained by the GAN or added to the data generated by the GAN to ensure the fairness of the test results. Since WM-811K is collected from different lots, the wafer map dimension will be different according to the different die sizes of different lots. So we resize all data to 64×64 to preserve as much as possible the feature of each size.

Most of the existing GAN methods use human evaluation. However, human evaluation is often biased towards the quality of the generated samples and ignores the diversity of the samples. We use the method proposed by [22] for G2LGAN evaluation. It includes Inception Score (IS), Mode score (MS), Fréchet inception distance (FID), Kernel maximum mean discrepancy (Kernel MMD), Wasserstein distance (WD), 1- nearest neighbors algorithm. If the model is good, the IS and MS should be as high as possible and the FID, Kernel MMD, and WD should be as low as possible. And 1-nearest neighbor accuracy is close to 0.5, and the better the result.

On the classification network, most of the existing methods use accuracy to judge model performance. When the data set is not balanced, the accuracy of the model is overestimated and fairness is lost. Therefore, we use both precision, recall, and F1-score as the evaluation metrics of our model. Because we consider each category of metrics to be equally important, we use Macro-average rather than Weighted-average when balancing metrics multi-class.

Implementation Results

We evaluate the effectiveness of the data augmentation network and classification network separately. In the data augmentation network evaluation, the proposed method is compared with those of CGAN, ACGAN, and BAGAN. Since the official source code is not available, we replicated them and achieved similar results. In the wafer classification network evaluation, we train the model using augmented data and compare the classification network to state-of-the-art works.

In the training setup, our method is implemented in Tensorflow. All the models are trained by adaptive moment estimation (Adam). The optimizer of G2LGAN have $\beta1 = 0$ and $\beta2 = 0.9$. The learning rate for the discriminator is 0.0004 and the learning rate for the generator is 0.0001. G2LGAN first trains 3 epochs with all the data and then trains 10 epochs with each class of data. Each epoch contains 10000 iterations, and each batch size is set to 64. The optimizer of classification network with $\beta1 = 0.9$, $\beta2 = 0.999$ and the initial learning rate is 0.1. We use WM-811K and virtual data generated by G2LGAN and run for 1000 epochs using a step decay of learning rate at the factor of 10 at epochs 200, 500, and 800.

C. *Quantitative Assessment of G2LGAN*

We show the comparison with the conditional image GAN that is currently able to generate the specified classes, and the generated image results are shown in Fig. 4. Let's take Donut as an example. Donut only accounts for 2.17% (555 images) of the classes with patterns, which is one of the minority classes. From Fig. 6, we can see that G2LGAN generates better results than other methods. It not only preserves the global features of the wafer map but also generates the class features of Donut.

However, evaluating the model only by generating images tends to focus on the generation effect and ignore the importance of diversity. So we conduct a quantitative analysis of each method as the results are shown in Table II. When trained on WM-811k at 64×64 resolution, G2LGAN achieves an IS of 8.763, FID of 15.241, and 1-NN accuracy of 0.531. It shows that all the methods have low scores in the IS and

MS projects. We classify the WM-811K dataset using InceptionNet pre-trained on ImageNet as shown in Table III. More than half of the data in WM-811K were classified into Petri dishes, and more than 90% of the data were classified into 4 of the 1000 classes in ImageNet. This is because IS and MS are calculated using the Inception network pre-trained on ImageNet. Since the image of the wafer map is very different from the spatial distribution of the ImageNet data, no matter how well it is generated, the WM-811K data will not have enough diversity in ImageNet and thus the IS and MS scores are low.

The proposed method scores 15.241, 0.478, and 7.402 in the comparison of FID, MMD, and WD. These three distance-related metrics are extracted by using InceptionNet pre-trained on ImageNet to extract features and then compare the distance between real data and generated data by different methods. Since the feature maps are compared directly instead of the classification results, the impact of the dataset on the metrics is smaller. The accuracy of G2LGAN in 1-NN is 0.531. The ideal value of GAN's 1-NN accuracy is 0.5, which means that the G2LGAN generation results are very realistic so that the 1-NN classifier cannot classify between the real and the generated data. From the experimental results, the proposed method is the best in all the metrics, especially in the 1-NN accuracy we are very close to the ideal state. In the next subsection, we will combine the generated data with the training data to further verify whether our G2LGAN can work well in real applications.

D. Quantitative Assessment of Wafer Map Classification

We combine the images generated by G2LGAN with the training set and undersampling to balance the dataset. Our undersampling is different from the traditional random undersampling, which randomly deletes excess data. The advantage of this approach is that no valid data is deleted and most of the untrained data may be trained in the next epoch. We generate the data to 5000 if the number of original data is less than 5000. The balanced data distribution is shown in Table IV.

Table V shows the results of the classification network by different works. In this table, the rank of number one is marked in red color. In [10], VGG16 is used for the classification network, and therefore the maximum number of parameters is used. When the training data is insufficient, using a large network structure will lead to overfitting. [15] and [16] used the standard CNN component classification networks. The standard CNN uses a larger number of parameters compared to the depthwise separable convolution. [17] only used three layers of convolutional layers and one layer of fully connected layers to achieve low parameter values. However, too few parameters make the classification model unable to effectively infer the correct defect category. In [18], the depth-separable convolution is also used as the backbone of the classification network.

[14] and [19] are the newest works. In [14], a multigranularity GAN was used to generate synthetic wafer maps for WMDR which is similar to our method. However, our G2LGAN achieves better classification accuracy since it extracts global features in the first stage, and then fine-tunes the model by each class in the second stage. In [19], an additional 2DPCA framework is used here to extract more features from wafer maps. However, it also increases the overall complexity of the classification model. In contrast, we

focus more on extracting low-dimensional features and reducing the number of convolutions for high-dimensional features to reduce the number of parameters and maintain high accuracy.

Our proposed method is the best in almost all the metrics compared with the current methods. Since most of the methods do not use undersampling to suppress the amount of data for most classes, it makes accuracy overestimated. Since our classification network uses depthwise separable convolution and focuses on low-dimensional features, we can maintain accuracy with a low number of parameters.

Commonly, a network architecture with a lower number of parameters often requires sacrificing model accuracy. Thus we emphasize the data augmentation to balance the data set and improve model performance. In Table V, we also show our method without G2LGAN as a reference result. The results show that the dataset enhanced with G2LGAN increases accuracy by 9.16%, F1-Score by 8.76%, precision by 6.31%, and recall by 11.39%. This shows that the data generated by G2LGAN can be effectively applied with obvious improvement.

Table VI shows the confusion matrix. It demonstrates the prediction accuracy of the proposed method for each class on the dataset. The numbers in the matrix indicate the distribution of the predicted labels and the actual labels and the subscripts of the diagonal data indicate the recall rate of that class. It can be easily seen that None-class is more misclassified than other classes. Since we want to keep the fairness of the testing set, we do not balance the testing set, which further explains the reason that the proposed method has only 90.9% precision. The performance of each class is shown in Table VII. Note that Donut has the worst performance in the WM-811 category because it has fewer test data and it is easier to pull down the score due to a small amount of misclassification.

We also conducted experiments on 21-Defect. Since the data set is too small, and even some categories have only single-digit samples, we first use rotation and flipping for preliminary data enhancement, and then use G2LGAN to generate data for each category. Since 21-Defect is a non-public dataset, hereby we firstly show the difference between the works with G2LGAN and without G2LGAN respectively. As shown in Table VIII, the scores of the data after using G2LGAN can be improved by 15% in each indicator. Compared to our previous work [18], although our F1-Score is only 0.1% higher, our model is 70% smaller than [18].

Since WM-811K and 21-Defect have overlapping categories, including Center, Mount, Edge-Arc (Edge-Loc), Edge-Ring, Random, Scratch-Acr&Scratch-Line, Near-full. We feed the data of overlapping categories in 21-Defect into the classification model trained on WM-811K for classification, to test whether the data in the non-training data set can be correctly classified by the classification network. The confusion matrix of the classification results is shown in Table IX, the horizontal axis is the data label of 21-Defect, and the vertical axis is the label predicted by the Our + model trained with WM-811K. The experimental results show that most of the overlapping categories can be accurately classified, representing no over-fitting and good robustness of our model.

V. Conclusions

In this paper, we propose G2LGAN to generate images where it extracts global features and local features separately to generate effective data. Especially in the imbalanced dataset, we consider suppressing the influence of the majority class on the model by undersampling and also increasing the data of the minority class by data augmentation to balance the data set. Compared with the existing conditional GANs, our G2LGAN can generate high-quality images even on data imbalanced datasets. In our G2LGAN, it achieves 0.531% 1-NN accuracy on WM-811K. We further use G2LGAN to construct the wafer classification model. We use MobileNet V2 as the backbone network and use random undersampling and G2LGAN to balance the WM-811K dataset. The proposed method maintains high accuracy and is superior to the existing methods. In the number of parameters and computation, our proposed model has significant performance compared with other proposed approaches.It can effectively classify the defect patterns on WM-811K and achieve an F1-Score of 93.01%.

Declarations

Abbreviations

Not applicable

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIAL

Please contact the author for data requests.

COMPETING INTERESTS

The authors declare that they have no competing interests.

FUNDING

This research was supported and funded by the Ministry of Science and Technology,

Taiwan, under Grant MOST 111-2221-E-008 -089 -MY3.

Authors' Contributions

Chieng-Yang Wang performed the experiments. Tsung-Han

Tsai reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

References

1. A. Carlson, T. Le “Correlation of Wafer Backside Defects to Photolithography Hot Spots Using Advanced Macro Inspection” 2006 31st International Symposium, Microlithography - An SPIE Event.
2. M. H. Hansen, V. N. Nair, and D. J. Friedman, “Monitoring wafer map data from integrated circuit fabrication processes for spatially clustered defects,” Technometrics, vol. 39, no. 3, 1997, in press.
3. K. Kyeong and H. Kim, "Classification of Mixed-Type Defect Patterns in Wafer Bin Maps Using Convolutional Neural Networks," in IEEE Transactions on Semiconductor Manufacturing, vol. 31, no. 3, pp. 395-402, Aug. 2018, doi: 10.1109/TSM.2018.2841416.
4. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). Cnn-rnn: A unified framework for multi-label image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2285-2294).
5. M. Zhang, W. Li, and Q. Du, “Diverse region-based CNN for hyperspectral image classification,” IEEE Trans. Image Process., vol. 27, no. 6, pp. 2623–2634, Jun. 2018.
6. J. Li, J.-H. Cheng, J.-Y. Shi, and F. Huang, “Brief introduction of back propagation (BP) neural network algorithm and its improvement,” in Advances in Computer Science and Information Engineering, Heidelberg, Germany: Springer, 2012, pp. 553–558
7. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al. Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680, 2014.
8. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).
9. M. Bach, A. Werner and M. Palt, "The Proposal of Under-sampling Method for Learning from Imbalanced Datasets", Procedia Computer Science, vol. 159, pp. 125-134, 2019. Available: 10.1016/j.procs.2019.09.167.
10. M. Wu, J. R. Jang and J. Chen, "Wafer Map Failure Pattern Recognition and Similarity Ranking for Large-Scale Data Sets," in IEEE Transactions on Semiconductor Manufacturing, vol. 28, no. 1, pp. 1-12, Feb. 2015, doi: 10.1109/TSM.2014.2364237.
11. D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in Proc. Int. Conf. Learning Representations, 2014.
12. A. Odena, C. Olah, J. Shlens, “Conditional image synthesis with auxiliary classifier GANs,” in International conference on machine learning. PMLR, 2017. P. 2642-2651.
13. G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi. “BAGAN: Data augmentation with balancing GAN,” arXiv preprint arXiv:1803.09655, 2018.
14. J. Yu and J. Liu, "Multiple Granularities Generative Adversarial Network for Recognition of Wafer Map Defects," in IEEE Transactions on Industrial Informatics, vol. 18, no. 3, pp. 1674-1683, March 2022, doi: 10.1109/TII.2021.3092372.

15. T. Ishida, I. Nitta, D. Fukuda, and Y. Kanazawa, “Deep learning based wafer-map failure pattern recognition framework,” in Proc. 20th Int. Symp. Qual. Electron. Design (ISQED), Santa Clara, CA, USA, Mar. 2019, pp. 291–297
16. U. Batool, M. I. Shapiai et al., “Convolutional Neural Network for Imbalanced Data Classification of Silicon Wafer Defects,” in Proceedings of 2020 16th IEEE International Colloquium on Signal Processing Its Applications (CSPA), February 2020, pp. 230–235.
17. M. B. Alawieh, D. Boning, and D. Z. Pan, “Wafer map defect patterns classification using deep selective learning,” in Proc. 57th ACM/IEEE Design Autom. Conf. (DAC), Jul. 2020, pp. 1–6.
18. T. -H. Tsai and Y. -C. Lee, "A Light-Weight Neural Network for Wafer Map Classification Based on Data Augmentation," in IEEE Transactions on Semiconductor Manufacturing, vol. 33, no. 4, pp. 663-672, Nov. 2020, doi: 10.1109/TSM.2020.3013004.
19. J. Yu and J. Liu, "Two-Dimensional Principal Component Analysis-Based Convolutional Autoencoder for Wafer Map Defect Detection," in IEEE Transactions on Industrial Electronics, vol. 68, no. 9, pp. 8789-8797, Sept. 2021, doi: 10.1109/TIE.2020.3013492.
20. A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.
21. Odena, et al., "Deconvolution and Checkerboard Artifacts", Distill, 2016. http://doi.org/10.23915/distill.00003.
22. M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” in Proc. 34th Int. Conf. Machine Learning, 2017, pp. 214–223.
23. L. Sun, J. Chen and K. Xie, ”Deep and shallow features fusion based on deep convolutional neural network for speech emotion recognition,” International Journal of Speech Technology, vol. 21, no. 4, pp. 931-940, December. 2018.

Tables

Table I to VIII is available in the Supplementary Files section.

Figures

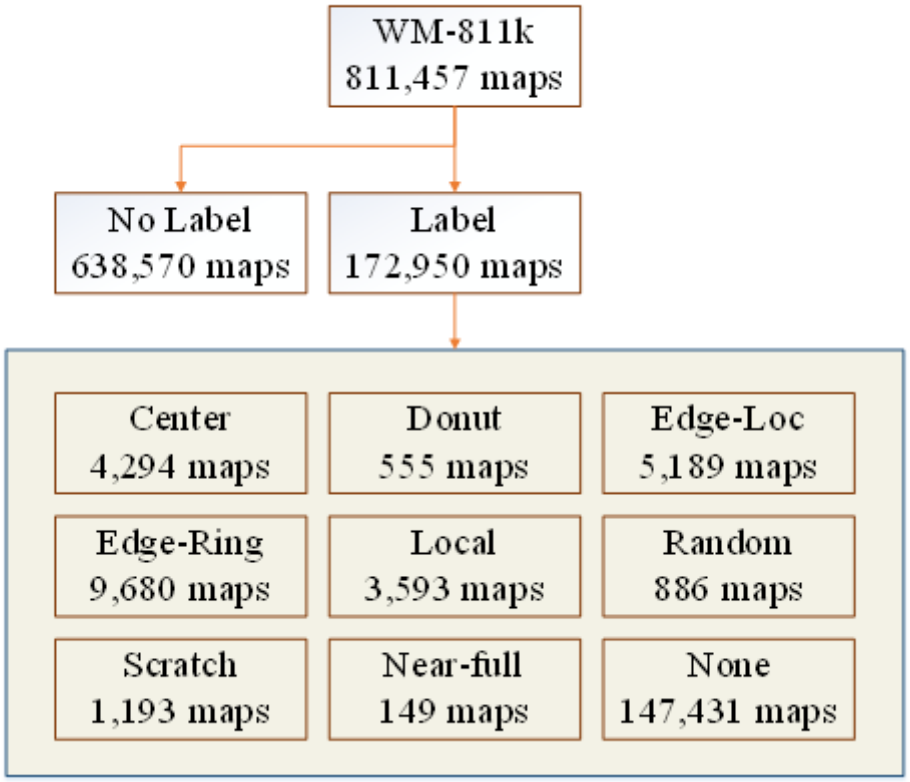

Figure 1

Data distribution of WM-811K. The dataset is composed of real-world wafer maps and labeled by experts in IC industry.

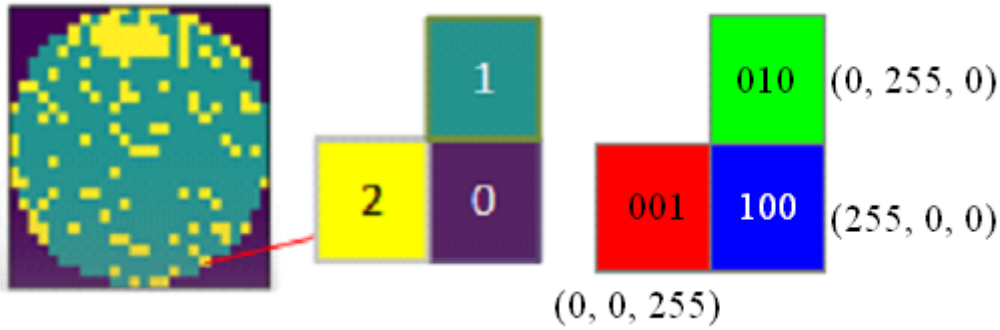

Figure 2

Data pre-processing using one hot encoding.

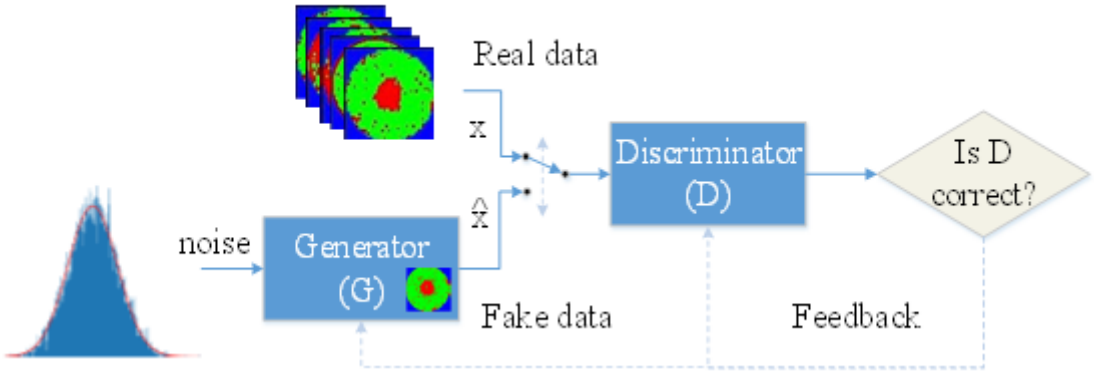

Figure 3

GAN architecture and training flow. The generator maps the Gaussian noise to the real data space, and the discriminator randomly selects input from real data and fake data and judges whether it is real data.

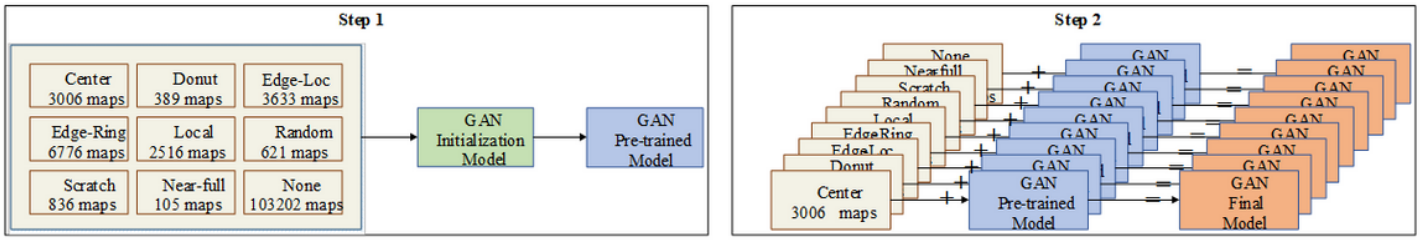

Figure 4

G2LGAN flowchart. G2LGAN is a two-step network. The first step allows the model to learn the global features of the entire dataset. The second step trains by the pre-trained model on the entire dataset with classes of conditions to learn the local features of the conditional classes.

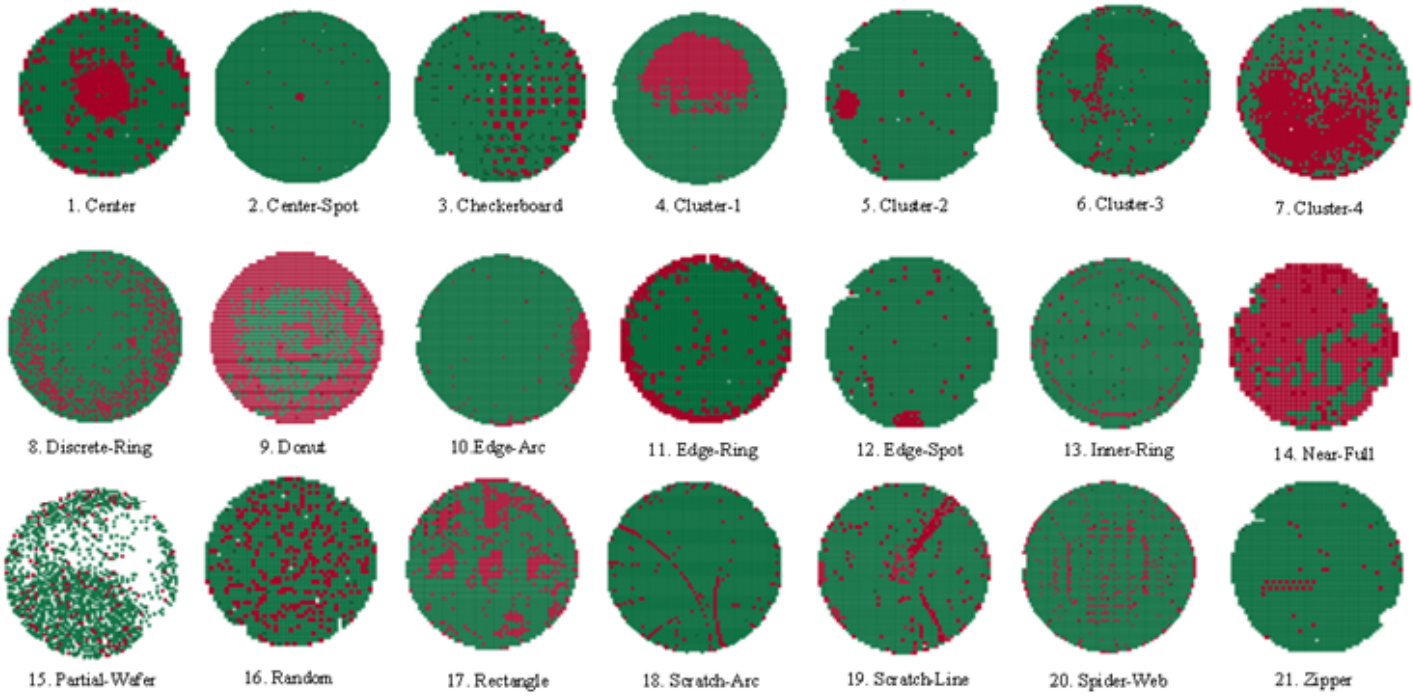

Figure 5

Visualization of 21-defect dataset defect pattern.

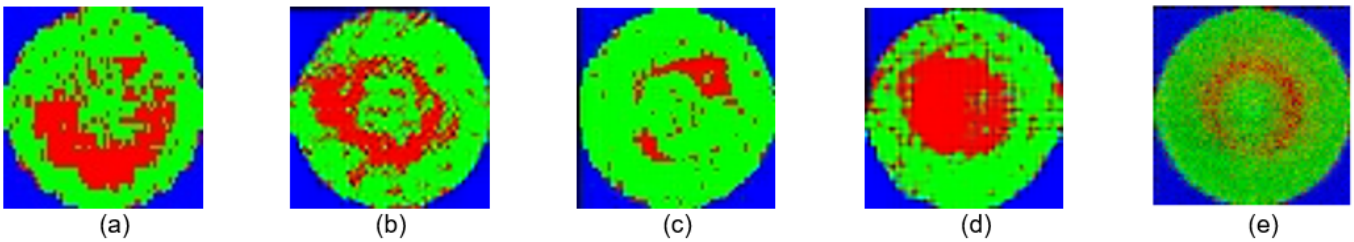

Figure 6

From left to right: (a). Real data of WM811K, (b). the proposed G2LGAN, (c). BAGAN optimized for data imbalance, (d). ACGAN using auxiliary classifier, and (e). CGAN to supply class labels for generator and discriminator.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Table1to8.docx