applied sciences

MDPI

Article

Inspection and Classification of Semiconductor Wafer Surface Defects Using CNN Deep Learning Networks

Jong-Chih Chien [1], Ming-Tao Wu [2] and Jiann-Der Lee [2,3,4,*]

[1] Degree Program of Digital Space and Product Design, Kainan University, Taoyuan 33587, Taiwan; jcchien@mail.knu.edu.tw

[2] Department of Electrial Engineering, Chang Gung University, Taoyuan 33302, Taiwan; qwe85202001@gmail.com

[3] Department of Neurosurgery, Chang Gung Memorial Hospital at Linkou, Taoyuan 33305, Taiwan

[4] Department of Electrical Engineering, Ming Chi University of Technology, New Taipei City 24301, Taiwan

* Correspondence: jdlee@mail.cgu.edu.tw; Tel.: +886-3-211-8800 (ext. 5316)

Received: 16 July 2020; Accepted: 31 July 2020; Published: 2 August 2020

Featured Application: To detect and classify semiconductor wafer defects in order to help determine the cause(s) of the defects.

Abstract: Due to advances in semiconductor processing technologies, each slice of a semiconductor is becoming denser and more complex, which can increase the number of surface defects. These defects should be caught early and correctly classified in order help identify the causes of these defects in the process and eventually help to improve the yield. In today's semiconductor industry, visible surface defects are still being inspected manually, which may result in erroneous classification when the inspectors become tired or lose objectivity. This paper presents a vision-based machine-learning-based method to classify visible surface defects on semiconductor wafers. The proposed method uses deep learning convolutional neural networks to identify and classify four types of surface defects: center, local, random, and scrape. Experiments were performed to determine its accuracy. The experimental results showed that this method alone, without additional refinement, could reach a top accuracy in the range of 98% to 99%. Its performance in wafer-defect classification shows superior performance compared to other machine-learning methods investigated in the experiments.

Keywords: deep learning; semiconductor wafer defects; convolution neural network; automatic inspection

1. Introduction

In the progress of semiconductor design methodologies [1–5], more and more integrated circuit components can be patterned then etched onto semiconductor wafers. This is true especially in the DRAM (dynamic random access memory) industry, where, in addition to the demands of increasing the speeds of access and longer lifespans, there are other demands to be met: for example, each successive generation of the DRAM chips must become smaller and more compact, so that more memory can be fit into an even smaller space. But as the pressure for meeting these demands increases, the probabilities of manufacturing process-based defects appearing on the surface of the wafers also increases, and the yield becomes more likely to decrease. Since it appears that the defects are linked to fabrication steps in the process, the problem of identifying and classifying defect patterns on the wafers is inseparable from the problem of improving the manufacturing yields. The figure below, Figure 1, show the basic-block diagrams of a semiconductor manufacturing process. The purposes of some of the basic blocks are:

- Thin-film processing: the use of physical or chemical means to perform vapor deposition of crystals on thin film.
- Chemical-mechanical polishing: the principle of polishing to flatten the even contours on the wafers.
- Photolithography: using photoresist for exposure and development, so as to leave the photo-masked pattern on the wafer.
- Etching: to remove materials from the surface of the wafer by physical or chemical means wherever the surface is not protected by the photoresist.
- Diffusion and ion implantation: to use physical phenomena of heat diffusion to alter the semiconductor's electrical conductivity, then ionize the surface substance, then control the electrical current magnitude to control the concentrations of ions.
- Oxidation: to reduce the damage that can occur during the ion implantation stage.
- Metallization: mainly to perform the connections of metals.

The types of manufacturing processing problems that can occur could involve robot handoffs, contaminations, flow leakages, etc. Semiconductor engineers are able to use the defect patterns on the wafers to locate problems in the process, which would then become clues in helping improve the yield.

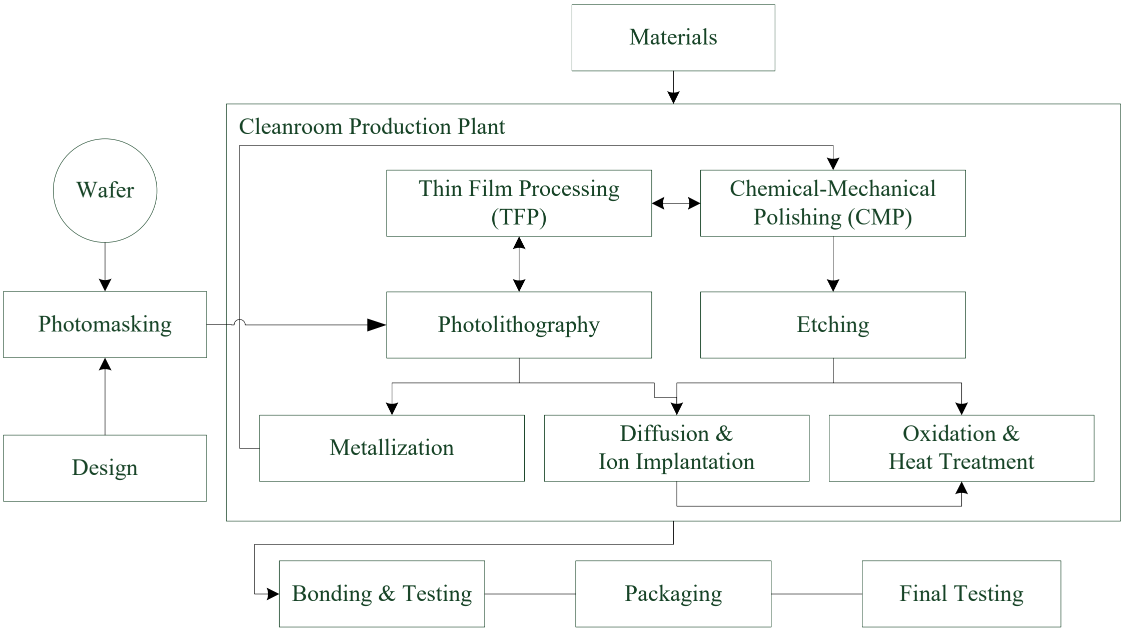
Materials
Cleanroom Production Plant
Wafer
Thin Film Processing (TFP)
Chemical-Mechanical Polishing (CMP)
Photomasking
Photolithography
Etching
Design
Metallization
Diffusion & Ion Implantation
Oxidation & Heat Treatment
Bonding & Testing
Packaging
Final Testing

Figure 1. The semiconductor manufacturing diagram.

Kaempf [6] identified, in general, that manufacturing defects can be classified into three types: Type-A, Type-B, and Type-C.

1. Type-A defects are evenly random with a stable mean density. This type of defect is generated randomly, and no specific clustering phenomenon is visible, as shown in Figure 2a. The cause of this type of defect is complex and not fixed to particular patterns. It is difficult to find the cause of this type of defect. This type of yield abnormality can be reduced by improving the stability and accuracy of the process.
2. Type-B defects are systematic and repeatable from wafer to wafer. This type of defect has obvious clustering phenomenon, as shown in Figure 2b,c. The cause of this type of defect can usually be found by the distribution of defects on the wafer, which is used to find abnormalities in the process or machine, such as the misalignment of the mask position during photo development or excessive etching during the process, etc.
3. Type-C defects vary from wafer to wafer. This type of defect is the most common occurrence in semiconductor manufacturing. That is, it is a combination of Type-A defects and Type-B defects,

as shown in Figure 2d. In this type of defect, it is very important to eliminate the causes for random defects and keep systemic defects, so that engineers can find the cause of anomalies.

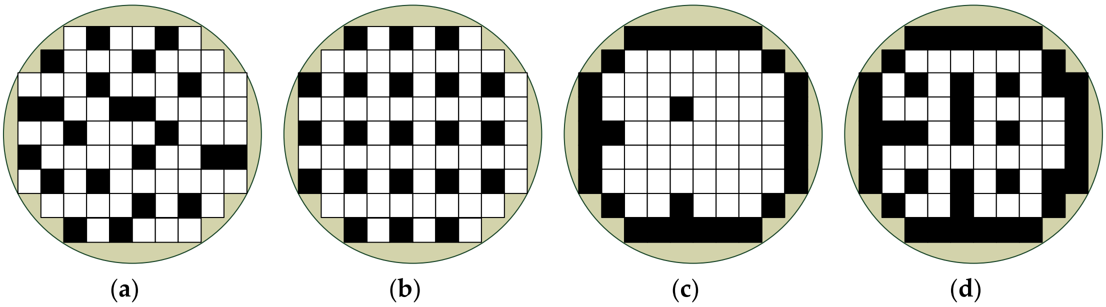

Figure 2. Wafer defects: (**a**) random defect, (**b**) repeatable defect, (**c**) systematic defect, and (**d**) combinational defect.

There are many different types of defects, both visible and invisible. However, based on Kaempf's classification system [6] and suggestions from engineers about different types of visible defects with known causes, the wafer-defect images will be grouped into four major classes: random, local, center, and scrape, for this study. Examples of these four major classes are shown in Figure 3. The defects in the random type are almost randomly distributed across the entire surface of the wafer. The defects in the local type are concentrated on the edge of the wafer but do not exhibit linear or curvy characteristics. The defects in the center type are concentrated around or near the center of the wafer in circular or ring-like patterns. The defects in the scrape type exhibit linear or curvy distribution from the edge moving toward the center of the wafer.

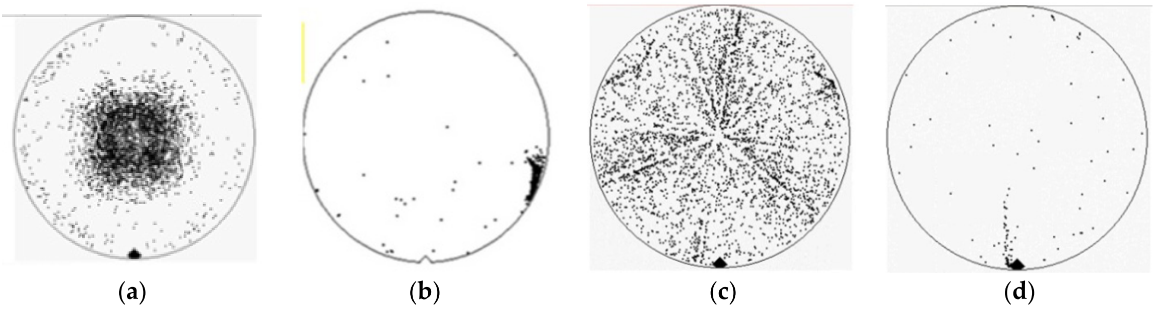

Figure 3. Defect types: (**a**) center, (**b**) local, (**c**) random, and (**d**) scrape.

These four classes of defect have known possible cause. The likely causes of each of these types of defects are:

- Center type: due to abnormality of RF(Radio Frequency) power, abnormality in liquid flow, or abnormality in liquid pressure.
- Local type: due to silt valve leak, abnormality during robot handoffs or abnormality in the pump.
- Random type: due to contaminated pipes, abnormality in showerhead, or abnormality in control wafers.
- Scrape type: mainly due to abnormality during robot handoffs or wafer impacts.

If the defect type can be correctly identified, then by the use of the process of elimination, the engineers can localize the cause(s) of the defects in the process and thus should be able to correct them and increase the yield.

In previous studies on wafer defects, R. Baly et. al. [7] used an SVM(Support Vector Machine) classifier to classify 1150 wafer images into two classes, high and low yields and reported to have achieved an accuracy of 95.6%. H. Dong et al. [8] used the machine-learning algorithm of logistic regression to detect whether a wafer contains defects in order to predict yield, and the testing was done using synthetically generated images simulating six types of wafer defects; F-scores (F-measure) varying from 77.9% to 96.6% were reported. L. Puggini et al. [9] used random forest as a similarity measurement to separate faulty wafers from normal wafers using a total of 1600 wafers in the experiment; however, no accuracy was reported. M. Saqlain et al. [10] used a voting ensemble classifier (SVE) using density as a feature to detect wafer defect patterns using 25,519 unequally distributed wafer images containing eight classes of defects: center, donut, edge-local, edge-ring, local, random, scratch, and near-full, and the paper reported that it was able to achieve an average of 95.8% accuracy. X. Chen et al. [11] proposed a light-weight CNN model for training and classifying 13,514 28 × 28 wafer images into 4 classes: no defect, mechanical defects, crystal defects, and redundant defects; however, the paper did not mention how many images were used for training, but it did report that an average of 99.7% accuracy was achieved, which included a 100% detection rate on wafers with no defect.

In the current semiconductor manufacturing industries, the inspection process for visible defects still mostly relies on manual labor for errors. However, identification by manpower alone will eventually result in false identifications due to fatigue and lack of objectivity. Therefore, the purpose of this research is to find a reliable machine vision-based method to correctly identify and classify the type of wafer defects in the hope of replacing manual inspection. This paper investigates methods involved using variants of the convolution neural networks, which are deep-learning neural networks and have been shown to be effective in various complex vision-based tasks, because they are able to learn the nonlinear relationship(s) between the inputs and the expected outputs [12]. In the following section, we will discuss the parameters and design of the proposed convolution neural network, followed by experimental comparisons with the performances of other machine-learning methods presented in the literatures. In the conclusion section, we will discuss possible improvements and future works.

2. Methodologies

The system flowchart for the training portion of the primary method being investigated is shown below in Figure 4. In this initial study, 25,464 raw images with visible defects were collected online from the WM-811K [13] dataset, which contains 811,457 semiconductor wafer images from 46,393 lots with eight defect labels. Based on experiments performed in this paper, a few wafer images exhibit multiple types of defects and have only a single label: this is a limitation of this dataset. From this dataset, only images with visible defects were chosen for the purpose of this paper. However, these wafer images were acquired from different lots with different image acquisition methods, so the raw images do not have a uniform definition of defects. For the purpose of this investigation, a uniform definition would be better. So for uniformity, each raw image is preprocessed to extract only the area containing the wafer by blackening the image areas outside the wafer to remove them from consideration. Then, the contrast of the wafer area is enhanced before being binarized using threshold(s) obtained via the OTSU [14] algorithm. The purpose of binarization is to emphasize the defects. This operation is then followed by redefining the areas without defects as having the same grayscale value, while the defects are redefined using white pixels. The final step is to normalize the image to 256 × 256 pixels, while making sure that no white pixel is unintentionally removed by the normalization. About 75% of all the normalized images are used to train the convolution neural network, which include features extraction, features condensation, and classification. After training, about 25% of the normalized images are used for testing, and an additional 40 images are used for validation. For this paper, all 25,464 wafer images were labeled, 19,112 images were randomly chosen from each of the four types for training: center: 4464, local: 4680, random: 5928, scrape: 4040. These four classes were grouped from the eight defect labels in WM-811k due to similarities in the causes of the defects. The number of wafer images chosen for testing is 6312, and 40 images, 10 from each type of defect, were reserved for validation. The other method we investigated in this paper is the use of transfer learning on pretrained faster R-CNN models, which will be discussed later.

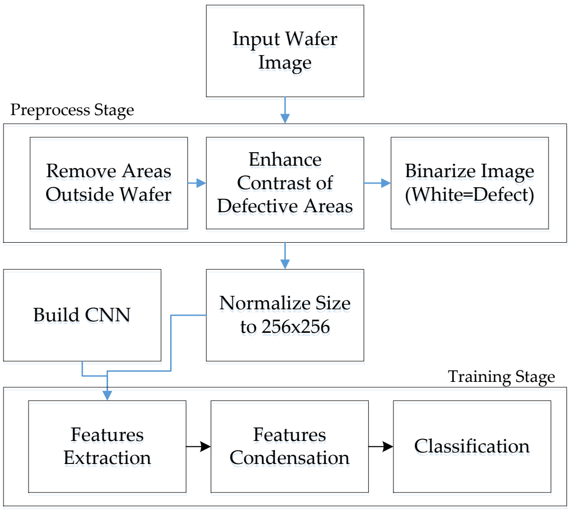
Input Wafer Image
Preprocess Stage
Remove Areas Outside Wafer
Enhance Contrast of Defective Areas
Binarize Image (White=Defect)
Build CNN
Normalize Size to 256x256
Training Stage
Features Extraction
Features Condensation
Classification

Figure 4. The system flowchart.

2.1. Convolution Neural Network

The architectures for the convolution neural network come in many different flavors [15]. However, the fundamental architecture must contain convolution layers with activation functions, pooling layers, and fully connected layer(s) for combining features before classification, as shown in Figure 5. The convolution layers extracted the feature maps, then the pooling layers concentrate the feature maps using various functions, such as maximizing or averaging values within a given window in order to reduce the processing complexity for the following layers. These layers are followed by a fully connected layer(s) before the classification. The proof of the effectiveness of CNN in feature extraction has been discussed in the literatures [16], so it will not be repeated in this paper. During the training phase, each of the labeled images is provided as input, and the CNN is expected to output the correct label. The entire training phase can take a long time, depending on the value set for number of epochs or the convergence threshold.

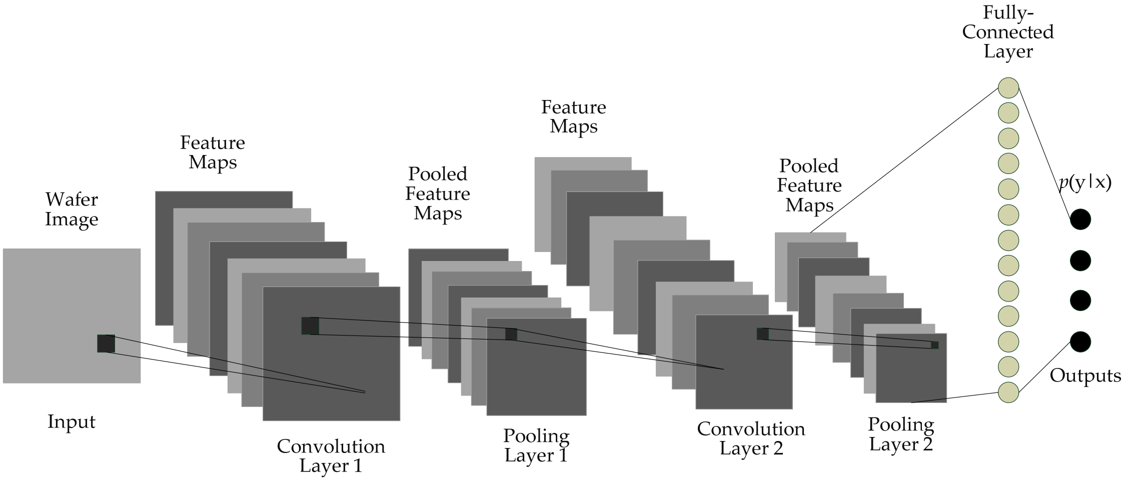

Figure 5. The layers of a convolution neural network.

Each convolution layer consists of a weight matrix *w* and *a* bias *b*: the values of *w* and *b* would be initialized and updated during the training of the layer. The output of each convolution layer would be [17]:

$$x_i^l = \sigma(\sum_{i \in FM_j} w_i x_j^{l-1} \times k_{ij}^l + b_j^l) \tag{1}$$

where l is the layer, FM is the feature map, k is the convolution kernel, b is the bias, and $\sigma(.)$ is the activation function for each connection from i to j. The activation function is, generally, the ReLU function [17], where $ReLU(x)$ = max (0,x), and

$$\begin{aligned} ReLU(x) &= \max\,(0,\ x),\ \text{and} \\ \tfrac{d}{dx}ReLU(x) &= \begin{cases} 1,\ if\ x > 0 \\ 0,\ otherwise \end{cases} \end{aligned} \quad (2)$$

After experimentations probing for a good design, the following CNN network design is proposed for this investigation. In this design, the input layer accepts 256 × 256 images, and the middle layers contains five convolution layers with ReLU activation functions. Four of the convolution layers are followed by max pooling layers with 2 × 2 windows, and the last convolution layer with ReLU activation function is followed by one average pooling layer with a 2 × 2 window. The averaging pooling layer, which would then be the last pooling layer before the fully connected layer, is followed by a softmax function [17] used to map the output for classification. This proposed design for the CNN network is chosen based on a good tradeoff between the amount of training time and accuracy for the 256 × 256 input images.

2.2. *Transfer Learning on Faster R-CNN*

The training of a CNN network takes a long time, especially with a large dataset. In this paper, we also investigated whether the training time can be reduced by using the transfer learning technology on pretrained deep learning neural networks. For this purpose, a special type of CNN, the faster R-CNN [18], is chosen. Faster R-CNN is a variant of the CNN deep-learning neural network first proposed to address the object detection problem. It differs from the standard CNN network shown in Figure 5 by the addition of a detection network for a set of possible regions containing objects by forming bounding boxes at different scales, which is termed the region proposal generation network. The block diagram of a faster-R-CNN network is illustrated in Figure 6.

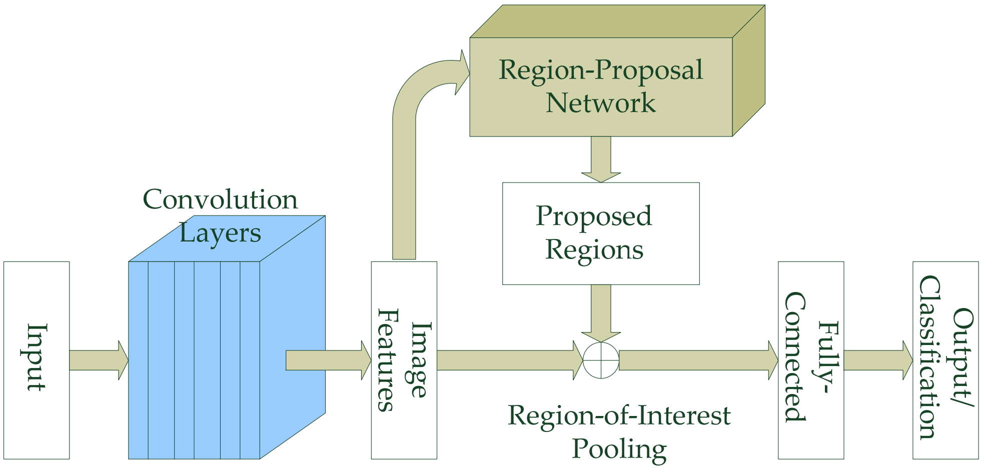

Figure 6. The block diagram of a faster-R-CNN network.

The technology of transfer learning is to extract the parameters of a trained deep-learning neural network, such as weights and bias, and apply them to the same type of network but for different purposes and domains, plus some additional training for the images with new classes, so that the new classes can be generated using smaller training sets. For example, take a network trained to detect different objects and use it instead to locate handguns [19]. The use of transfer learning in deep-learning neural networks has an advantage that the network does not need to be trained from zero, which may take a long time, and has shown to be effective in certain applications. The proof of effectiveness of transfer learning has been discussed in the literature [20]. In this paper, we take two pretrained faster-R-CNN models on different datasets for object detection, COCO [21] and KITTI [22]—which are available in the Tensorflow package [23]—and use them to detect wafer defects on a subset of training images. Due to the reduction in the training set, additional images can be added to the testing set.

2.3. Comparisons and Validation

The testing and validation results of the proposed CNN wafer defect classifier will be presented in the next section. Comparisons will be made with other machine-learning-based classifiers presented in the literatures: SVM [7], logistic regression [8], random forest [9], and weighted average (or soft voting ensemble) [10]. The characteristics of these classifiers are:

- SVM is a well-known supervised classifier that performs by separating classes using hyper-planes, which are called support vectors.
- Logic regression is a variant extended from linear regression. The main algorithm of logistic regression is used in a binary classification algorithm to solve linearly separable problems.
- Random forest is an algorithm that integrates multiple decision trees, which is applied to the combination of various decision trees of different subsamples of the original data set.
- Weighted average or soft voting ensemble (SVE) is a voting classifier integration method, the decision value of the classifier is assigned a higher weight to improve the overall performance of the overall classifier. Mainly, the results from the conventional classifier are input into the SVE method to summarize and obtain the final wafer defect classification results.

Unless otherwise specified, the default settings will be used for the above classifiers in the experiments. The validation will be done using 40 randomly chosen images that were not in the training or testing dataset. The result will show whether further training is required for the proposed CNN classifier.

2.4. Evaluation Measurements

In the first two experiments, the confusion matrix and the accuracy measure are used to present the classification results for ease of visualization. This measurement works the best when the number of classes is few. It presents the correct classified results along the diagonal entries in a tabular format. The incorrectly identified results are off the diagonal, and the class to which they were wrongly classified can easily be identified.

The definition of accuracy measure is as follows: [10]

$$Accuracy = \frac{All\ Wafer\ Images\ in\ the\ Same\ Class\ Correctly\ Identified}{All\ Wafer\ Images\ Labeled\ In\ the\ Same\ Class} \tag{3}$$

This measure defines how well a classifier performs for a certain class. The overall accuracy is just the average of all the accuracy measures for all the classes.

In the last experiment, we selected additional measures, include precision, recall, and the F-measure. Precision is the measure of how close the predicted results are to the actual value. Recall is the measure of the classifier's ability to predict all the data of interest. The F-measure is a trade-off between these two, and its value is high when the predicted results and the actual results are close to each other. The definition of precision, recall and F-measure is [10]:

$$Precision = \frac{Wafer\ Images\ in\ the\ This\ Class\ Truly\ Identified}{All\ Wafer\ Images\ Classified\ in\ This\ Class}, \tag{4}$$

$$Recall = \frac{Wafer\ Images\ in\ the\ This\ Class\ Truly\ Identified}{All\ Wafer\ Images\ Labeled\ in\ this\ Class}, \tag{5}$$

$$F-Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

The AUC (area under the roc curve) measure is the area under the ROC curve [10]. The ROC curve is a plot of Recall vs. the False Positive Rate, which is:

$$False\ Positive\ Rate = \frac{Wafer\ Images\ in\ the\ This\ Class\ Falsely\ Identified}{All\ Wafer\ Images\ Not\ In\ This\ Class}, \tag{7}$$

That shows the degree of reparability within the same class. The higher the AUC value implies that the classifier is better at predicting each label class or distinguishing between different defect classes. These are all measurements used in previous papers [7–10] as measurements for the effectiveness of their classifier.

3. Results

Three experiments were performed to evaluate the performance of the proposed CNN, as well as the use of transfer learning on pretrained faster-R-CNN models for wafer image inspection. The first experiment compared the testing results of the trained CNN classifier vs. a well-known machine classifier, the SVM, using 6312 testing images. The results will be visually displayed using the confusion matrices. The second experiment evaluates two transferred learned faster-R-CNN models retrained using fewer, 16,000, training images and 9411 test images. The last experiment will compare the results of the trained CNN classifier vs. logistic regression (LR), random forest (RF), and weighted average (soft voting ensemble, SVE) in terms of precision, recall, F-measure, and AUC for evaluations and comparisons. These are the measures used to help identify the effectiveness of a classifier. Finally, the validation results of the proposed CNN model will be shown.

3.1. CNN vs. SVM

In this experiment, the default values for the parameters are used for the SVM classifier to classify the input features into four classes. The proposed CNN uses the parameters mentioned above and is set to train for 100 epochs. Figure 7 shows examples of correctly classified images by the proposed CNN classifier but missed by SVM. The results of the performance comparison between the proposed CNN classifier and the SVM classifier using the confusion matrices and accuracy measure are shown below in Table 1. The numbers of correctly identified wafer defect images are displayed in bold numbers. Based on the numbers in the diagonal entries of the confusion matrices and the accuracy measurements, it is clear that the proposed CNN classifier outperforms the SVM classifier for every type of defects investigated.

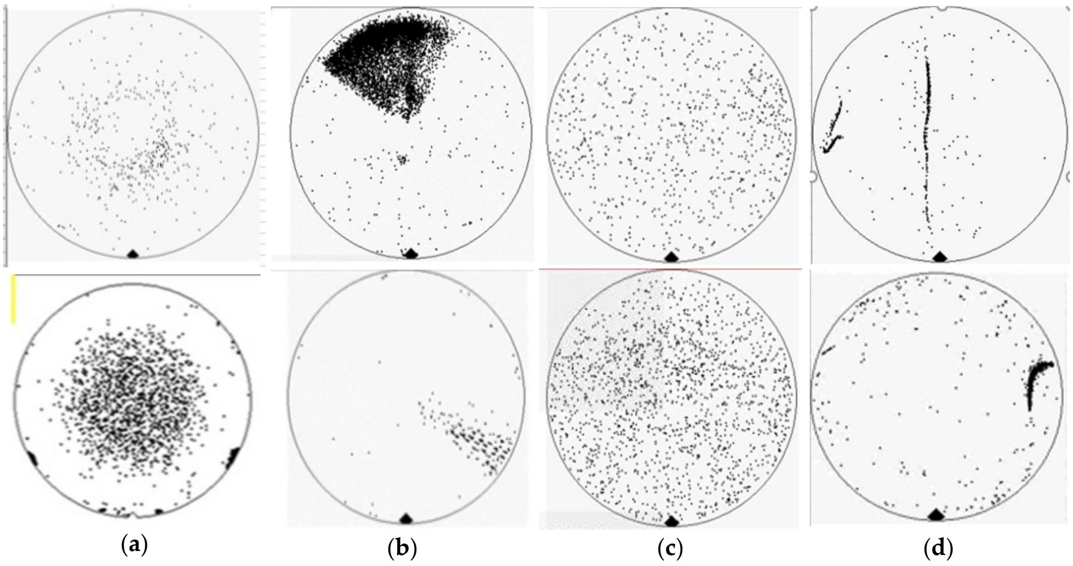

Figure 7. Examples of correctly classified images: (**a**) center, (**b**) local, (**c**) random, and (**d**) scrape.

Table 1. Confusion matrices of SVM and CNN classifications.

		Proposed CNN Model				**Accuracy**
True Class	Center	**1250**	4	9	1	98.89%
	Local	5	**1430**	21	24	97.00%
	Random	10	19	**2696**	3	98.83%
	Scrap	5	11	16	**808**	96.19%
		Predicted Class				97.73%
		SVM				**Accuracy**
True Class	Center	**1212**	0	10	42	95.89%
	Local	22	**1065**	185	208	72%
	Random	41	26	**2358**	303	86.44%
	Scrap	22	68	134	**616**	73.33%
		Predicted Class				83%

3.2. Transfer Learning Using Faster R-CNN (COCO vs. KITTI)

The pretrained faster-R-CNN models used in this experiments were trained on two different datasets: COCO and KITTI. Their performance comparisons on wafer defect classification using confusion matrices are shown below in Table 2. The parameter settings are: input size of 256 × 256, four additional classes, 4000 samples from each defect class for retraining, and 50 training epochs. The technology of transfer learning was utilized, and a smaller training set, compared to the proposed CNN model, was used to train these models. Because of this, a larger test set, 9411 test images, can be used to test these faster-R-CNN models. A quick glance at the results will show that the KITTI-pretrained faster R-CNN performs slightly better than the COCO-pretrained model in most cases. Their accuracies are comparable to the proposed CNN model even with a smaller training set and shorter training time.

Table 2. Confusion matrices of faster R-CNN trained on COCO and KITTI datasets.

		COCO				**Accuracy**
True Class	Center	2183	11	30	8	97.81%
	Local	9	2305	20	6	98.50%
	Random	10	15	2936	3	99.06%
	Scrap	11	55	12	1942	96.14%
		Predicted Class				97.88%
		KITTI				**Accuracy**
True Class	Center	2208	11	8	5	98.92%
	Local	10	2276	15	39	97.27%
	Random	7	14	2942	1	99.26%
	Scrap	2	30	2	1986	98.32%
		Predicted Class				98.44%

3.3. CNN vs. LR, RF, and SVE

In the third experiment, the parameters for the classifiers to be tested all use default values. The results of the performance comparison between the proposed CNN classifier with logic regression (LR), random forest (RF), and SVE classifiers using precision, recall, F-measure, and AUC are shown below in Table 3. The best performances are displayed in bold characters. Again, a glance at these values shows that the performance of the proposed CNN classifier is better than the other classifiers being investigated.

Table 3. Classification results of various methods.

		LR	RF	SVE	CNN
Precision	Center	89.88	94.22	92.54	**98.89**
	Local	70.49	85.75	83.91	**96.62**
	Random	86.25	95.18	95.78	**98.83**
	Scratch	75	89.66	81.36	**96.19**
Recall	Center	77.85	82.70	87.31	**98.43**
	Local	31.0	57.37	55.78	**97.68**
	Random	77.53	88.76	89.33	**98.32**
	Scratch	16.12	32.23	39.67	**96.65**
F-Measure	Center	83.44	88.08	89.85	**98.66**
	Local	43.13	68.74	67.01	**97.15**
	Random	81.66	91.86	92.44	**98.57**
	Scratch	26.53	47.41	53.33	**96.42**
AUC	Center	99.45	99.57	99.80	**99.85**
	Local	93.08	98.36	98.75	**99.68**
	Random	99.95	99.97	**99.98**	99.79
	Scratch	91.21	96.90	97.90	**99.75**

3.4. CNN Validation

The validation of the proposed CNN model was performed using 40 reserved wafer images, 10 from each defect class, and the result is shown in Table 4. It shows that 38 out of the 40 images were classified correctly. Two images, one in each of the center and local defects, appears to have been misclassified. A closer examination of these two misclassified images shows that they both exhibit multiple types of defect, yet only one type was labeled, as the example in Figure 8 shows. The wafer image in Figure 8 was labeled as a local defect, but both local and scrape-defect types can be observed. Since this study only included the four main types of defects and did not include mixed types, the two images were misclassified.

Table 4. CNN validation results.

Defect Type	Sample Size	Correct
Center	10	90%
Local	10	90%
Random	10	100%
Scrape	10	100%
Total	40	95%

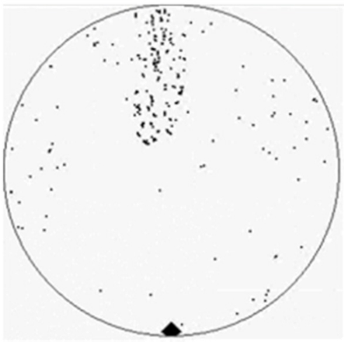

Figure 8. Wafer image exhibiting both scrape and local defect types.

3.5. Discussion of Results

In previous studies, R. Baly et al. [14] achieved an accuracy of 95.6% in classifying 1150 wafers into two classes, normal and faulty, using an SVM classifier. L. Puggini et al. [16] did a similar experiment using random forest as the similarity measurement using 1600 wafer images. However, both the number of wafer images and the number of classes were too few and would not be able to help semiconductor engineers to identify the sources of the defects. The experiment of H. Dong et al. [15] using logistic regression on synthetic images lacks useful performance evaluation on real wafer images. The experiment of M. Saqlain et al. [17] using a voting ensemble classifier on eight classes of defects achieved an accuracy of 95.8% on 25,519 unequally distributed wafer images. X. Chen, et al. [19] trained and tested a light-weight CNN with 13,514 28 × 28 wafer images with four classes, including one for no defect and report an average accuracy 99.7%. Not only were the few defect classes not very helpful in helping to identify the causes of the defects, the image size could be too small to catch smaller, but still visible, defects. For example, assuming a wafer size of one inch in diameter, 28 × 28 can classify a defective pixel no smaller than 0.1 cm^2, while 256 × 256 can be useful in classifying a defective pixel as small as 0.01 cm^2. Compared to these previous studies, the proposed CNN classifier, shows better performance with real, larger wafer images, as shown using the proposed measurements. However, to ascertain this, experiments were designed to compare the proposed CNN classifier against the classifiers mentioned using the same test dataset.

In the experiments performed in this paper, 256 × 256 wafer images with defects were trained and classified into four classes, each with known causes, and compared with methods presented in previous studies: SVM, logistic regression, random forest and soft voting ensemble. Because the testing test sets are the same, the performance results can then be compared. According to the experiment results shown above, the proposed CNN architecture with 19,112 training samples performed better than the other methods in terms of accuracy, precision, recall, F-Measure, and AUC. With this we can conclude that the proposed trained CNN model can more accurately classify defective wafer images, according to the four classes with known causes of defects, than the other methods. With an input size of 256 × 256, it should be possible to capture smaller defects than the light-weight CNN model proposed by X. Chen et al. [19]. In terms of accuracy, the proposed CNN classifier achieved an average accuracy of 98.88%, while the SVM achieved only an average of 83% across the four tested defect types. In the comparison experiment against the other classifiers, SVE and RF methods achieved nearly 90% in average precision; however, lower F-measures in the local and random classes indicate that, as classifiers for the given test dataset, their performances lag behind our proposed CNN classifier. However, the AUC measurements show that the SVE may be slightly better than the proposed CNN classifier in the random class.

In the transfer learning experiment using two pretrained faster-R-CNN models, by re-using the pretrained models from Tensorflow [23], it was possible to retrain them using a smaller training set for a few new classes that were not in the original labels. By running a larger testing set of 9411 wafer images through these retrained models, the results show that the KITTI model was a slightly better performer than the COCO model, and the overall performance in terms of accuracy was no worse than the proposed CNN model with a larger training set. Based on the result of this experiment, the use of the technology of transfer learning appears to be a feasible avenue of research in the future for wafer defect classification.

4. Conclusions

In this paper, two ways to use the deep learning convolution neural networks to classify semiconductor defect images were presented. The first way is to train a carefully designed CNN with five convolution layers using 19,112 images of semiconductor wafer with defects, and the second way is to use a pretrained faster R-CNN and apply transfer learning using just 16,000 images. Both ways achieved similar performances, but discovering which way is better requires additional investigations. Regardless, both methods seek to present reliable alternatives to manual inspection of semiconductor wafers. Because the raw materials used in the semiconductor industry are expensive, and defects caused by the manufacturing process can result in great losses, thus careful inspection of different types of defects would help the engineers to locate and fix the problem at its source(s) and improve the yield.

The results of the experiments presented in this paper show that the use of convolution neural networks in classifying wafer images can be a feasible alternative to manual inspection and have been shown to perform well above other known machine-learning methods, such as SVM, logistic regression, random forest, and soft voting ensemble. However, the misclassification that occurred during the validation phase shows that the proposed design can be further improved. In future research, other defect types, including mixed types, should be added into the types of defects to be classified and, if possible, there should be an increase in the number of training samples used in future investigations. There are plans to investigate the effective methods for classifying defect images using other flavors of the convolution neural network architectures in the future, including the Mask R-CNN [24].

Author Contributions: Conceptualization, analysis, validation, writing, J.-C.C. and J.-D.L.; software, M.-T.W.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by Ministry of Science and Technology (MOST) and Chang Gung Memorial Hospital, Taiwan, Republic of China, under Grants MOST107-2221-E-182-026-MY2 and CMRPD2G0121, respectively.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chandrasekar, K.; Goossens, S.; Weis, C.; Koedam, M.; Akesson, B.; Wehn, N.; Goossens, K. Exploiting expendable process-margins in DRAMs for run-time performance optimization. In Proceedings of the Desig, Automation & Test in Europe Conference & Exhibition, Dresden, Germany, 24–28 March 2014; pp. 1–6.
2. Huang, P.S.; Tsai, M.Y.; Huang, C.Y.; Lin, P.C.; Huang, L.; Chang, M.; Shih, S.; Lin, J. Warpage, stresses and KOZ of 3D TSV DRAM package during manufacturing processes. In Proceedings of the 14th International Conference on Electronic Materials and Packaging (EMAP), Lantau Island, Hong Kong, China, 13–16 December 2012; pp. 1–5.
3. Hamdioui, S.; Taouil, M.; Haron, N.Z. Testing Open Defects in Memristor-Based Memories. *IEEE Trans. Comput.* **2013**, *64*, 247–259. [CrossRef]
4. Guldi, R.; Watts, J.; Paparao, S.; Catlett, D.; Montgomery, J.; Saeki, T. Analysis and modeling of systematic and defect related yield issues during early development of a new technology. In Proceedings of the Advanced Semiconductor Manufacturing Conference and Workshop, Boston, MA, USA, 23–25 September 1998; Volume 4, pp. 7–12.
5. Shen, L.; Cockburn, B. An optimal march test for locating faults in DRAMs. In Proceedings of the 1993 IEEE International Workshop on Memory Testing, San Jose, CA, USA, 9–10 August 1993; pp. 61–66.
6. Kaempf, U.; Ulrich, K. The binomial test: A simple tool to identify process problems. *IEEE Trans. Semicond. Manuf.* **1995**, *8*, 160–166. [CrossRef]
7. Baly, R.; Hajj, H. Wafer Classification Using Support Vector Machines. *IEEE Trans. Semicond. Manuf.* **2012**, *25*, 373–383. [CrossRef]
8. Dong, H.; Chen, N.; Wang, K. Wafer yield prediction using derived spatial variables. *Qual. Reliab. Eng. Int.* **2017**, *33*, 2327–2342. [CrossRef]
9. Puggini, L.; Doyle, J.; McLoone, S. Fault Detection using Random Forest Similarity Distance. *IFAC PapersOnLine* **2015**, *48*, 583–588. [CrossRef]
10. Saqlain, M.; Jargalsaikhan, B.; Lee, J.Y. A Voting Ensemble Classifier for Wafer Map Defect Patterns Identification in Semiconductor Manufacturing. *IEEE Trans. Semicond. Manuf.* **2019**, *32*, 171–182. [CrossRef]
11. Chen, X.; Chen, J.; Han, X.; Zhao, C.; Zhang, D.; Zhu, K.; Su, Y. A Light-Weighted CNN Model for Wafer Structural Defect Detection. *IEEE Access* **2020**, *8*, 24006–24018. [CrossRef]
12. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object Detection with Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]

13. Kaggle SM-811K Wafer Map. WM-811K Wafer Map. Available online: https://www.kaggle.com/qingyi/wm811k-wafer-map (accessed on 14 February 2019).
14. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
15. Al-Saffar, A.A.M.; Tao, H.; Talab, M.A. Review of deep convolution neural network in image classification. In Proceedings of the 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Jakarta, Indian, 23–24 October 2017; pp. 26–31.
16. Wiatowski, T.; Boelcskei, H. A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction. *IEEE Trans. Inf. Theory* **2018**, *64*, 1845–1866. [CrossRef]
17. Indolia, S.; Goswami, A.K.; Mishra, S.; Asopa, P. Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Comput. Sci.* **2018**, *132*, 679–688. [CrossRef]
18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal network. In Proceedings of the Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015.
19. Gyanendra, V.; Anamika, D. A Handheld Gun Detection using Faster R-CNN Deep Learnin. In Proceedings of the 7th International Conference on Computer and Communication Technology, Allahabad, India, 24–26 November 2017; pp. 84–88.
20. Wang, Z. Theoretical Guarantees of Transfer Learning. *arXiv* **2018**, arXiv:1810.05986.
21. Coco dataset. COCO-Common Objects in Context. Available online: https://cocodataset.org/ (accessed on 14 February 2020).
22. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
23. Tensorflow. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 12 October 2019).
24. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.