

# A Deep Convolutional Neural Network for Wafer Defect Identification on an Imbalanced Dataset in Semiconductor Manufacturing Processes

Muhammad Saqlain<sup>1</sup>, Qasim Abbas, and Jong Yun Lee

반도체 제조공정의 불균형 데이터셋에 대한 웨이퍼 불량 식별을 위한 심층 컨볼루션 신경망  
무하매드 사칼린, 카심 아바스, 이종연

**Abstract**—Wafer maps contain information about various defect patterns on the wafer surface and automatic classification of these defects plays a vital role to find their root causes. Semiconductor engineers apply various methods for wafer defect classification such as manual visual inspection or machine learning-based algorithms by manually extracting useful features. However, these methods are unreliable, and their classification performance is also poor. Therefore, this paper proposes a deep learning-based convolutional neural network for automatic wafer defect identification (CNN-WDI). We applied a data augmentation technique to overcome the class-imbalance issue. The proposed model uses convolution layers to extract valuable features instead of manual feature extraction. Moreover, state-of-the-art regularization methods such as batch normalization and spatial dropout are used to improve the classification performance of the CNN-WDI model. The experimental results comparison using a real wafer dataset shows that our model outperformed all previously proposed machine learning-based wafer defect classification models. The average classification accuracy of the CNN-WDI model with nine different wafer map defects is 96.2%, which is an increment of 6.4% from the last highest average accuracy using the same dataset.

**Index Terms**—Wafer maps, wafer defect identification, deep learning, convolutional neural network, data augmentation, batch normalization.

## I. INTRODUCTION

SEMICONDUCTOR manufacturing processes should produce high-quality products and improve the wafer yield to fulfill market demands. However, semiconductor fabrication is a very complex, costly, and time-consuming process that involves various chemical, mechanical, and electrical processes such as deposition, etching, photolithography, chemical planarization, ion implantation, and diffusion [1]. After applying all these processes, integrated circuits (ICs) are composed by making circuit structures on various layers of the same wafer and joining them with wires. A wafer surface should be very clean, and all layers of the circuit should be perfectly aligned to produce high-quality ICs. However, well-trained semiconductor engineers working with highly automated and precise equipment in a very clean environment, cannot produce error-free wafer dies [2]. At the end of the fabrication process, each wafer goes through a circuit prob test where defective and defect-free wafer dies are differentiated and the results of the test are represented in wafer map [WM], which is a 2-dimensional (2D) wafer image.

**요약** - 웨이퍼 맵에는 웨이퍼 표면의 다양한 불량 패턴에 대한 정보가 포함되어 있으며 이러한 **불량의 자동 분류**는 근본 원인을 찾는 데 중요한 역할을 합니다. 반도체 엔지니어는 웨이퍼 불량 분류를 위해 수동 육안 검사 또는 수동으로 유용한 특징을 추출하는 기계학습 기반 알고리즘과 같은 다양한 방법을 적용합니다. 그러나 이러한 방법은 신뢰할 수 없으며 분류 성능도 좋지 않습니다. 따라서 **본 논문에서는 자동 웨이퍼 불량 식별을 위한 딥러닝 기반 컨볼루션 신경망(CNN-WDI)을 제안합니다.** 본 논문에서는 클래스 불균형 문제를 극복하기 위해 **데이터 증량기법**을 적용했습니다. 제안된 모델은 수동으로 특징을 추출하는 대신 컨볼루션 계층을 사용하여 가치 있는 특징을 추출합니다. 또한 CNN-WDI 모델의 분류 성능을 향상시키기 위해 **배치 정규화(Batch Normalization) 및 공간 드롭아웃(Spatial Dropout)과 같은 최신 규제화(regularization) 방법**을 사용합니다. 실제 웨이퍼 데이터셋을 사용한 실험 결과 비교는 우리 모델이 이전에 제안된 모든 기계학습 기반 웨이퍼 불량 분류 모델보다 성능이 우수함을 보여줍니다. 9개의 서로 다른 웨이퍼 맵 불량이 있는 CNN-WDI 모델의 평균 분류 정확도는 96.2%로, 동일한 데이터셋을 사용한 **이전의 최고 평균 정확도에서 6.4% 증가**했습니다.

## 1. 서론

반도체 제조 공정은 고품질 제품을 생산하고 웨이퍼 수율을 개선하여 시장 요구를 충족시켜야 합니다. 그러나 반도체 제조는 증착, 식각, 포토리소그래피, 화학적 평탄화, 이온 주입 및 확산과 같은 다양한 화학적, 기계적 및 전기적 공정을 포함하는 매우 복잡하고 비용이 많이 들고 시간이 많이 소요되는 공정입니다[1]. 이러한 모든 공정을 적용한 집적회로(IC)는 동일한 웨이퍼의 다양한 레이어에 회로 구조를 만들고 와이어로 결합하여 구성됩니다. 웨이퍼 표면은 매우 깨끗해야 하고 회로의 모든 레이어는 고품질 IC를 생산하기 위해 완벽하게 정렬되어야 합니다. 그러나 매우 깨끗한 환경에서 고도로 자동화되고 정밀한 장비로 작업하는 잘 훈련된 반도체 엔지니어도 오류 없는 웨이퍼 다이를 생산할 수 없습니다[2]. **제조공정이 끝나면 각 웨이퍼는 회로 프로브 테스트를 거쳐 결함이 있는 웨이퍼와 무결함의 웨이퍼 다이를 구별하고 테스트 결과를 2차원(2D) 웨이퍼 이미지인 웨이퍼 맵[WM]으로 표시합니다.**

<p>Experienced process engineers are hired to define the wafer defect patterns and give them unique labels such as <i>Center</i>, <i>Donut</i>, <i>Local</i>, <i>Edge-Loc</i>, <i>Edge-Ring</i>, <i>Scratch</i>, <i>Random</i>, <i>Near-Full</i>, and <i>None</i> [3]. Additionally, these defects become more common due to increasing integration density of circuits and the wafer design complexity. Each wafer defect occurs due to specific abnormal behavior of some fabrication process. For instance, <i>Center</i> defects may occur because of uniformity issues in chemical and mechanical planarization, <i>Edge-Loc</i> defects may occur because of thin film deposition, and <i>Edge-Ring</i> defects occur due to etching problems [4]. So, WM defect analysis provides crucial information to discover the abnormal processes in semiconductor manufacturing and to take measures to resolve them.</p>	<p>숙련된 프로세스 엔지니어를 고용하여 웨이퍼 불량 패턴을 정의하고 <b><i>Center</i>, <i>Donut</i>, <i>Local</i>, <i>Edge-Loc</i>, <i>Edge-Ring</i>, <i>Scratch</i>, <i>Random</i>, <i>Near-Full</i> 및 <i>None</i></b>과 같은 고유한 레이블을 부여합니다[3]. 또한, 이러한 불량은 회로의 집적 밀도와 웨이퍼 설계 복잡성을 증가시켜 더 일반적이 됩니다. 각 웨이퍼 불량은 일부 제조 공정의 특정 비정상 동작으로 인해 발생합니다. 예를 들어, <i>Center</i> 불량은 화학적·기계적 평탄화에서 균일성에 문제가 있을 때 발생할 수 있으며, <i>Edge-Loc</i> 불량은 박막 증착으로 인해 생길 수 있으며, <i>Edge-Ring</i> 불량은 에칭 문제로 인해 발생할 수 있습니다[4]. 따라서 <b>WM 불량 분석은 반도체 제조의 비정상 프로세스를 발견하고 이를 해결하기 위한 조치를 취하는 데 중요한 정보를 제공합니다.</b></p>
<p>Accurate classification of WM patterns plays an important role in identification of wafer defects, which will enhance the semiconductor yield and quality by improving the wafer fabrication process. Previously, wafer defects were examined by experienced process engineers using high-resolution microscopes by measuring the physical parameters of the WMs like location, size, and color. Moreover, various machine learning (ML) based automated defect classification (ADC) systems were introduced to reduce labor and fabrication costs while improving quality and yield. Wu <i>et al.</i> [3] proposed a model called WMFPR, which extracts Radon-based and geometry-based features from WM and applies a support vector machine (SVM) classifier to classify wafer defect patterns. Piao <i>et al.</i> [5] extracted Radon transform features and applied an ensemble-based decision tree model to classify various WM defects. Recently, Saqlain <i>et al.</i> [6] also proposed an ensemble-based WM classification model called WMDPI, which combines state-of-the-art ML classifiers such as random forest (RF), logistic regression (LR), SVM, and artificial neural network (ANN). Three different types of features were extracted from the WMs like Radon-based, geometry-based, and density-based, but the classification accuracy of their model was very poor for some specific defect classes and a lot of manual inspections like features extractions and hyperparameter setting were also needed.</p>	<p>WM 패턴의 정확한 분류는 웨이퍼 불량 식별에 중요한 역할을 하며, 이는 웨이퍼 제조공정을 개선하여 반도체 수율과 품질을 향상시킬 것입니다. 이전에는 숙련된 공정 엔지니어가 위치, 크기 및 색상과 같은 WM의 물리적 매개변수를 측정하여 고해상도 현미경을 사용하여 웨이퍼 불량을 검사했습니다. 또한 다양한 머신러닝(ML) 기반 자동 불량 분류(ADC) 시스템을 도입하여 인건비와 제조비용을 절감하고 품질과 수율을 향상시켰습니다. Wu 등[3]은 WM에서 Radon 기반 및 기하학 기반 특징을 추출하고 웨이퍼 불량 패턴을 분류하기 위해 SVM(Support Vector Machine) 분류기를 적용하는 WMFPR이라는 모델을 제안했습니다. Piao 등[5]은 라돈 변환 특징을 추출하고 앙상블 기반 의사결정 트리 모델을 적용하여 다양한 WM 불량을 분류했습니다. 최근 Saqlain 등[6] 또한 랜덤 포레스트(RF), 로지스틱 회귀(LR), SVM 및 인공 신경망(ANN)과 같은 최첨단 ML 분류기를 결합한 WMDPI라는 앙상블 기반 WM 분류 모델을 제안했습니다. <b>WM에서 Radon 기반, 기하학 기반 및 밀도 기반과 같은 세 가지 다른 유형의 특징이 추출되었지만 그 모델들은 몇몇 특정 불량 클래스에서는 분류 정확도가 매우 낮고, 특징 추출 및 하이퍼파라미터 설정과 같은 많은 수동 검사가 필요했다.</b></p>
<p>Through previous studies, it can be found that most of the WM defect classification models require manually extracted useful features. To do so, experienced semiconductor engineers first analyze the wafer surface and understand the physical measurements, and then propose valuable features according to the corresponding diagnosis issue. Such models are relatively costly, time-consuming, and inefficient in the presence of big WM datasets. However, a deep learning (DL) based classifier such as convolutional neural networks (CNNs) does not require manually extracted features for classification [7]. The CNN consists of three types of layers such as convolution layers, pooling layers, and fully connected layers. Whereas the convolution layer is typically used to extract features, the pooling layer summarizes the extracted features by reducing the size, and fully connected layers finally classify the input image using the extracted features [8].</p>	<p>선행 연구를 통해 대부분의 WM 불량 분류 모델은 수동으로 추출된 유용한 특징을 필요로 함을 알 수 있습니다. 이를 위해 숙련된 반도체 엔지니어가 먼저 웨이퍼 표면을 분석하고 물리적 측정을 이해한 다음 해당 진단 문제에 따라 유용한 특징을 제안합니다. 이러한 모델은 큰 WM 데이터셋이 있는 경우 상대적으로 비용이 많이 들고 시간이 많이 걸리며 비효율적입니다. 그러나 <b>CNN(Convolutional Neural Networks)과 같은 딥러닝(DL) 기반 분류기는 분류를 위해 수동으로 추출된 특징이 필요하지 않습니다[7]. CNN은 컨볼루션층, 풀링층, 완전연결층의 세 가지 유형의 층으로 구성됩니다. 일반적으로 컨볼루션 계층은 특징을 추출하는 데 사용되지만 풀링 계층은 크기를 줄여 추출된 특징을 요약하고 완전연결 계층은 추출된 특징을 사용하여 최종적으로 입력 이미지를 분류합니다[8].</b></p>

<p>Recently, many studies have been conducted that used CNN models to classify wafer defect patterns from the original WM images [9]. For example, Nakazawa and Kulkarni [10] applied CNN with a softmax activation function in the final layer to classify 22 WM defect patterns. The dataset they used was very small and highly imbalanced, so only simulated data was used to train and validate the model. Kyeong and Kim [11] also proposed a CNN model to classify single as well as mixed defect patterns on the same WM. They developed multiple CNN models and each model classifies a specific defect class which was practically very expensive in sense of time and computation. Cheon <i>et al.</i> [1] proposed a CNN model that can extract features from the real wafer image and accurately classified the input data into five different wafer defect classes. Their model can also classify the unknown defect classes after combining the CNN model with the k-nearest neighbors (<math>k - NN</math>) algorithm. The datasets used by all these studies were very small and highly imbalanced. Whereas, CNN models can get higher training accuracy in the presence of bigger datasets [12]. Additionally, imbalanced data distribution of various classes may cause the CNN to be biased for the majority data sample classes [13]. None of the previous studies has defined a suitable solution to overcome the data imbalanced issue for WM classification.</p>	<p>최근 CNN 모델을 사용하여 원본 WM 이미지에서 웨이퍼 불량 패턴을 분류하는 많은 연구가 수행되었습니다[9]. 예를 들어 Nakazawa와 Kulkarni[10]는 22개의 WM 불량 패턴을 분류하기 위해 최종 레이어에 소프트맥스 활성화 함수가 있는 CNN을 적용했습니다. 그들이 사용한 데이터셋은 매우 작고 불균형이 심했기 때문에 시뮬레이션된 데이터만 모델을 학습하고 검증하는 데 사용되었습니다. Kyeong and Kim[11]은 또한 동일한 WM에서 단일 및 혼합 불량 패턴을 분류하기 위해 CNN 모델을 제안했습니다. 그들은 여러 CNN 모델을 개발하고 각 모델이 하나의 특정 불량 클래스를 분류하도록 하였는데, 현실적으로 시간과 계산 측면에서 비용이 매우 큼니다. Cheon 등[1]은 실제 웨이퍼 이미지에서 특징을 추출하고 입력 데이터를 5가지 웨이퍼 불량 클래스로 정확하게 분류할 수 있는 CNN 모델을 제안했습니다. 그들의 모델은 또한 CNN 모델을 k-NN(k-nearest neighbor) 알고리즘과 결합한 후 알려지지 않은 불량 클래스를 분류할 수 있습니다. <u>이 모든 연구에서 사용된 데이터셋은 매우 작고 불균형이 매우 컸습니다. 반면 CNN 모델은 더 큰 데이터셋이 있는 경우 더 높은 학습 정확도를 얻을 수 있습니다[12]. 또한 다양한 클래스의 불균형 데이터 분포로 인해 CNN이 다수의 데이터 샘플 클래스에 대해 편향될 수 있습니다[13]. 이전 연구 중 어느 것도 WM 분류에 대한 데이터 불균형 문제를 극복하기 위한 적절한 솔루션을 정의하지 않았습니다.</u></p>
<p>In this paper, we propose a deep layered CNN-based wafer defect identification (CNN-WDI) model in a semiconductor manufacturing process. We use a real wafer dataset called WM-811K, which consists of nine different labeled classes of WM defect patterns. All defect classes are equally important because each of these occurs due to some specific abnormal behavior of the fabrication process. Semiconductor engineers classify the wafer defects to find the abnormal behavior behind these defects. But the available dataset is highly imbalanced that may ignore the classification of minority data sample classes. Thus, we implement a data augmentation method to increase the size of minority defect classes. Finally, we apply the proposed CNN-WDI model on the balanced dataset of 9 defect classes. For example, batch normalization and spatial dropout methods have been applied for regularization of the model. The experimental results comparison shows that CNN-WDI model has outperformed all previous models in terms of classification accuracy. The average classification accuracy of the proposed model with nine different WM defect patterns is 96.2%, which is an increase of 6.4% from the last highest average accuracy using the same dataset.</p>	<p>본 논문에서는 반도체 제조공정에서 심층 CNN 기반의 웨이퍼 불량 식별(CNN-WDI) 모델을 제안한다. <u>우리는 WM-811K라는 실제 웨이퍼 데이터셋을 사용합니다. 이 데이터셋은 WM 불량 패턴의 레이블이 지정된 9가지 클래스로 구성됩니다.</u> 모든 불량 클래스는 각각 제조공정의 특정 비정상 동작으로 인해 발생하기 때문에 똑같이 중요합니다. 반도체 엔지니어는 웨이퍼 불량을 분류하여 이러한 불량 뒤에 있는 비정상적인 동작을 찾습니다. 그러나 사용 가능한 데이터셋은 불균형이 매우 심해서 소수 데이터 샘플 클래스의 분류를 무시할 수도 있습니다. 따라서 <u>소수 불량 클래스의 크기를 늘리기 위해 데이터 증량 방법을 구현합니다.</u> 마지막으로 제안한 CNN-WDI 모델을 9개의 불량 클래스로 구성된 균형 잡힌 데이터셋에 적용합니다. 예를 들어, 모델의 정규화를 위해 배치 정규화 및 공간 드롭아웃 방법이 적용되었습니다. 실험 결과 비교는 CNN-WDI 모델이 분류 정확도 측면에서 모든 이전 모델을 능가함을 보여줍니다. <u>9개의 서로 다른 WM 불량 패턴을 가진 제안 모델의 평균 분류 정확도는 96.2%로, 동일한 데이터셋을 사용한 마지막 최고 평균 정확도보다 6.4% 증가했습니다.</u></p>
<p>The remainder of this research study is organized as follows. Section II introduces methods and material and Section III describes the proposed CNN-WDI model. Section IV presents experimental results and performance evaluation. Finally, Section V summarizes the whole study and gives a plan for our future research study.</p>	<p>본 연구의 나머지 부분은 다음과 같이 구성된다. II장에서는 방법과 재료를 소개하고 III장에서는 제안하는 CNN-WDI 모델을 설명한다. IV장에서는 실험결과와 성능평가를 제시한다. 마지막으로 V장에서는 전체 연구를 요약하고 향후 연구 계획을 제시한다.</p>

<p style="text-align: center;"><b>II. METHODOLOGY</b></p> <p>It is necessary to have a basic knowledge of CNN operations to understand the phenomenon of automatic feature extraction from WMs. This section briefly introduces the dataset, illustrates the data augmentation method, and describes the basic structure of CNN model.</p>	<p style="text-align: center;">2. 방법</p> <p>WM에서 자동 특징 추출 현상을 이해하려면 CNN 연산에 대한 기본 지식이 필요합니다. 이 섹션에서는 데이터셋을 간략하게 소개하고 데이터 증량 방법을 설명하고 CNN 모델의 기본 구조를 설명합니다.</p>
<p><b>A. Dataset</b></p> <p>The WM-811K dataset is a semiconductor dataset which consists of 811,457 real WM images [3]. The wafer images were collected from 46,293 lots in a circuit probe (CP) test of semiconductor fabrication process. A single lot contains 25 WMs, so there should be 1,157,325 WMs in total (i.e., 46,293 lots <math>\times</math> 25 wafer/lot). Since not all lots have exact 25 WMs due to some sensor faults or other unknown reasons, they were pruned from the dataset. The dataset also contains additional information about each WM such as lot name, die size, wafer index number, failure types, and training/test labels. This is the largest publicly available WM dataset that can be accessed at the MIR laboratory website [14]. There are different sizes of wafer images because of their two-dimensional nature and having different pixel values along the length and width of the image. We found total of 632 various sizes of wafer images ranging from (6<math>\times</math>21) to (300<math>\times</math>202).</p>	<p><b>A. 데이터셋</b></p> <p>WM-811K 데이터셋은 811,457개의 실제 WM 이미지로 구성된 반도체 데이터셋이다[3]. 웨이퍼 이미지는 반도체 제조공정의 회로 프로브(CP) 테스트에서 46,293 로트에서 수집되었습니다. 단일 로트에는 25개의 WM이 포함되므로 총 1,157,325개의 WM이 있어야 합니다 (즉, 46,293로트 <math>\times</math> 25 웨이퍼/로트). 일부 센서 오류 또는 기타 알 수 없는 이유로 인해 모든 로트에서 정확히 25개의 WM이 있는 것은 아니기 때문에 그 만큼 데이터셋에서 제거되었습니다. 데이터셋에는 로트 이름, 다이 크기, 웨이퍼 인덱스 번호, 실패 유형 및 학습/테스트 레이블과 같은 각 WM에 대한 추가 정보도 포함되어 있습니다. 이것은 MIR 연구소 웹사이트[14]에서 액세스할 수 있는 공개적으로 사용 가능한 가장 큰 WM 데이터 세트입니다. 웨이퍼 이미지는 2차원적 특성과 이미지의 길이와 너비에 따라 픽셀 값이 다르기 때문에 크기가 서로 다릅니다. 우리는 <b>(6<math>\times</math>21) ~ (300<math>\times</math>202) 범위의 총 632개의 다양한 크기의 웨이퍼 이미지를 찾았습니다.</b></p>
<p>Domain experts were responsible for defining nine different defect classes of WMs and assigning manual labels to 172,950 (21.3%) WMs in whole dataset. Unfortunately, the labeled dataset is highly imbalanced and only <i>None</i> defect class occupied 147,431 (85.2%) WMs of labeled dataset. The other eight defect classes that contains 25,519 (14.8%) WMs of labeled dataset as total are given as <i>Center</i>: 4294 (2.5%), <i>Donut</i>: 555(0.3%), <i>Edge-Loc</i>: 5189 (3.0%), <i>Edge-Ring</i>: 9680 (5.6%), <i>Local</i>: 3593 (2.1%), <i>Random</i>: 866 (0.5%), <i>Scratch</i>: 1193 (0.7%), and <i>Near-full</i>: 149 (0.1%). Fig. 1 shows the randomly selected wafer defect images from each class.</p>	<p>도메인 전문가는 WM의 9가지 서로 다른 불량 클래스를 정의하고 전체 데이터셋의 <b>172,950개(21.3%) WM에 수동 레이블을 할당</b>하는 일을 담당했습니다. 불행히도 레이블이 지정된 데이터셋은 불균형이 심해서, <b>None 불량 클래스만 레이블이 지정된 데이터셋의 147,431(85.2%) WM을 차지</b>했습니다. 레이블이 지정된 데이터셋의 나머지만 25,519개(14.8%) WM을 차지하는 다른 8개의 불량 클래스는 <b>Center: 4294(2.5%), Donut: 555(0.3%), Edge-Loc: 5189(3.0%), Edge-Ring: 9680(5.6%), Local: 3593(2.1%), Random: 866(0.5%), Scratch: 1193(0.7%) 및 Near-full: 149(0.1%)</b>와 같이 제공됩니다. 그림 1은 각 클래스에서 무작위로 선택된 웨이퍼 불량 이미지를 보여줍니다.</p>
<p><b>B. Data Preprocessing</b></p> <p>Wafer defect images were sampled from real-world dataset WM-811K which consists of 811,457 original WMs but only 21% of these wafers contain labeled classes. This dataset is highly imbalanced which leads to overtraining the majority data sample classes during the training of CNN-WDI model. Thus, we applied data augmentation using random rotation of 10-degree, horizontal flipping, and width shift, height shift, shearing range, channel shifting, and zooming by 20%, 20%, 15%,10%, and 10%, respectively. Thus, we increased the size of minority data sample classes up to 133%, 1702%, 93%, 178%, 1055%, 738%, and 6611% from the original data of <i>Center</i>, <i>Donut</i>, <i>Edge-Loc</i>, <i>Local</i>, <i>Random</i>, <i>Scratch</i>, and <i>Near-Full</i> classes, respectively. So, our new dataset consists of 90,000 wafer defect images of nine different balanced classes with 10,000 data samples of each class. The dataset was distributed into training, validation, and testing subsets at proportions of 65%, 20%, and 15%, respectively.</p>	<p><b>B. 데이터 전처리</b></p> <p>웨이퍼 불량 이미지는 811,457개의 원본 WM으로 구성된 실제 데이터셋 WM-811K에서 샘플링되었지만 이 웨이퍼 중 21%만이 레이블이 지정된 클래스를 포함합니다. 이 데이터셋은 불균형이 심하여 CNN-WDI 모델을 학습하는 동안 데이터 샘플이 많은 클래스를 과도하게 학습시킵니다. 따라서 <b>10°의 무작위 회전, 좌우 대칭 및 너비 이동, 높이 이동, 전단 범위, 채널 이동 및 확대/축소를 각각 20%, 20%, 15%, 10% 및 10%씩 사용하여 데이터 증량을 적용했습니다.</b> 따라서 소수 데이터 샘플 클래스의 크기를 Center, Donut, Edge-Loc, Local, Random, Scratch 및 Near-Full 클래스의 원본 데이터에서 각각 133%, 1702%, 93%, 178%, 1055%, 738%, 6611%까지 늘렸습니다. 따라서 우리의 새로운 데이터셋은 <b>각 클래스가 10,000개</b> 데이터 샘플과 갖는 9개의 서로 다른 균일한 클래스의 90,000개 웨이퍼 불량 이미지로 구성됩니다. 데이터셋은 각각 <b>65%, 20% 및 15%의 비율로 학습, 검증 및 테스트 데이터셋으로 나누었습니다.</b></p>

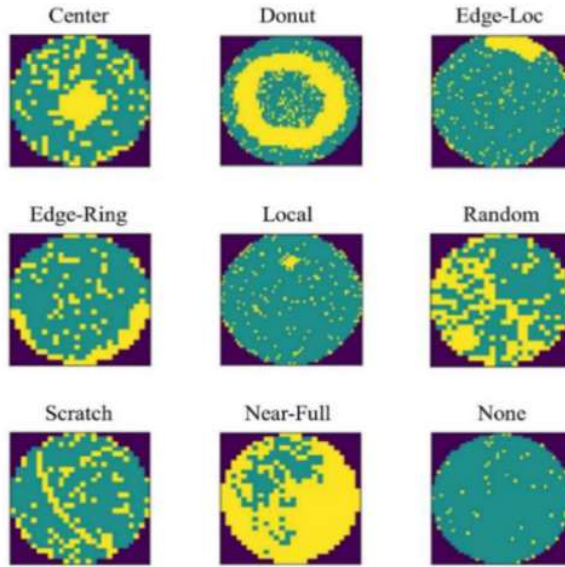


Fig. 1. Typical examples of nine wafer defect classes.

### C. Data Augmentation

Our dataset is not equally distributed and some of the wafer defect classes contain abundant data, while other classes have very little data. This issue is known as *class-imbalance problem*, and most image databases face this issue [15]. This unequal distribution of wafer defect classes can force the classification model to get higher accuracy for majority data sample classes during their training phase. Consequently, the training accuracy of the minority data sample class will be lower. The class-imbalance problem may also result in overfitting of training algorithm that means the algorithm will get higher training accuracy but have low testing performance. As all the wafer defects are equally important, it is necessary to correctly identify all wafer defect classes including minority data sample classes.

The easiest method to handle the class-imbalance and overfitting issues is to artificially increase the dataset of minority data sample classes. This method is called *data augmentation* and its commonly being used as model regularization technique in recent studies [8], [16], [17]. Deep neural networks (DNNs) can be trained well on massive amount of image datasets due to recent increasing computation capabilities. Thus, data augmentation becomes an effective method to improve both diversity and size of dataset by randomly augmenting the original data. Some common augmentation methods include flipping the image vertically or horizontally, shifting the image vertically or horizontally, and slightly rotating or zooming the image. This method helps the training model to be highly tolerant to changes in the size, position, and orientation of defects in the pictures.

### C. 데이터 증량

우리의 데이터셋은 균등하게 분포되어 있지 않으며 일부 웨이퍼 불량 클래스에는 지나치게 많은 데이터가 포함되어 있는 반면 다른 클래스에는 데이터가 거의 없습니다. 이런 문제를 **클래스 불균형 문제**라고 하며 대부분의 이미지 데이터베이스가 이 문제에 직면해 있습니다[15]. 웨이퍼 불량 클래스의 이러한 불균등한 분포로 인해 분류 모델은 학습 단계 동안 다수 데이터 샘플 클래스에 대해 더 높은 정확도를 얻을 수 있습니다. 결과적으로 소수 데이터 샘플 클래스의 학습 정확도가 낮아집니다. 클래스 불균형 문제는 또한 알고리즘이 더 높은 학습 정확도를 갖지만 낮은 테스트 성능을 갖는다는 것을 의미하는 학습 알고리즘의 과적합을 초래할 수 있습니다. 모든 웨이퍼 불량이 동등하게 중요하므로 소수 데이터 샘플 클래스를 포함한 모든 웨이퍼 불량 클래스를 정확하게 식별하는 것이 필요합니다.

클래스 불균형 및 과적합 문제를 처리하는 가장 쉬운 방법은 소수 데이터 샘플 클래스의 데이터셋을 인위적으로 늘리는 것입니다. 이 방법을 **데이터 증량(augmentation)**이라고 하며 최근 연구[8], [16], [17]에서 모델 규제화 기법을 일반적으로 사용하고 있다. 심층 신경망(DNN)은 최근 증가하는 계산능력으로 인해 방대한 양의 이미지 데이터셋을 잘 학습할 수 있습니다. 따라서 데이터 증량은 원본 데이터를 무작위로 증대하여 데이터셋의 다양성과 크기를 모두 향상시키는 효과적인 방법이 됩니다. 몇 가지 일반적인 증량 방법으로 이미지를 수직 또는 수평으로 뒤집기, 이미지를 수직 또는 수평으로 이동, 이미지를 약간 회전 또는 확대/축소하는 방법이 있습니다. 이 방법은 학습 모델이 사진에서 불량 크기의 크기, 위치 및 방향의 변화에 대해 높은 내성을 갖도록 돕습니다.

#### D. Convolutional Neural Network (CNN)

Recently neural networks, especially multilayer perceptron (MLP) and DNNs which have multilayered neural network structures got huge attention in the field of machine learning [18]. The DNN can achieve better performance than the traditional neural networks because it has more hidden layers which alternatively means more learning ability. One of the most popular DNN models is convolutional neural network (CNN). CNN is an advanced form of MLP and specially designed to classify the images [19]. It has many advantages over traditional ML classifiers because of its similarity to the human visual cortex and the ability to extract and learn 2-dimensional (2D) features. Moreover, the number of parameters of CNN are very few as compared to the same size of DNN architecture. Most CNN models are trained using a gradient-based learning approach to overcome the vanishing gradient problem (VGP). The VGP badly affects the lower layers of CNN and makes the training process very hard [20].

#### D. 컨볼루션 신경망(CNN)

최근 신경망, 특히 다층 신경망 구조를 갖는 MLP(Multilayer Perceptron)와 DNN이 머신러닝 분야에서 큰 주목을 받고 있습니다 [18]. DNN은 더 많은 학습 능력을 의미하는 더 많은 은닉층이 있기 때문에 기존 신경망보다 더 나은 성능을 달성할 수 있습니다. 가장 널리 사용되는 DNN 모델 중 하나는 CNN(Convolutional Neural Network)입니다. CNN은 MLP의 발전된 형태이며 특별히 이미지를 분류하도록 설계되었습니다[19]. 인간의 시각 피질과 유사하고 **2차원 (2D) 특징을 추출하고 학습**하는 기능 때문에 기존 ML 분류기에 비해 많은 이점이 있습니다. 또한 **CNN은 같은 크기의 DNN 아키텍처에 비해 파라미터 수가 매우 적다**. 대부분의 CNN 모델은 **기울기 소멸 문제(Vanishing Gradient Problem)**를 극복하기 위해 그래디언트 기반 학습 접근방식을 사용하여 학습됩니다. VGP는 CNN의 하위 계층에 나쁜 영향을 미치고 학습 과정을 매우 어렵게 만듭니다[20].

The basic architecture of CNN model consists of two major parts: a feature extraction network and a classification network as shown in Fig. 2. The original image enters the feature extraction network, where it passes through sequential pairs of convolutional (Conv) and pooling (Pool) layers for extraction of useful features. These features are used by the classification network to classify the input image. Each hidden layer of the CNN takes the output of the previous layer as its input and after processing, forwards its output to the next layer as the input. Both convolution and pooling layers create a group of 2D planes called feature maps. Higher-level features are extracted from features generated by lower layers. Various sizes of kernels are used in convolution layer to reduce the dimension of extracted features. The output of the last layer of the feature extraction network is used as the input of the classification network. The classification network contains a fully connected feed-forward neural network because of its better performance [21]. But these layers are comparatively expensive in terms of network parameters because each node of every layer is fully connected with each other. The final classification result is calculated by output layer or Softmax layer by giving the maximum values to the accurate classes. More details about each layer of CNN model can be found in [22].

CNN 모델의 기본 아키텍처는 그림 2와 같이 특징 추출 네트워크와 분류 네트워크의 두 가지 주요 부분으로 구성됩니다. 원본 이미지가 특징 추출 네트워크에 들어가면 일련의 컨볼루션(Conv) 및 풀링(Pool) 계층 쌍을 거치면서 유용한 기능을 추출합니다. 이러한 특징은 분류 네트워크에서 입력 이미지를 분류하는 데 사용됩니다. CNN의 각 히든 레이어는 이전 레이어의 출력을 입력으로 사용하고 처리 후 출력을 다음 레이어의 입력으로 전달합니다. 컨볼루션 계층과 풀링 계층 모두 피쳐맵(feature map)이라고 하는 2D 평면 그룹을 만듭니다. 상위 수준의 특징은 하위 계층에서 생성된 특징에서 추출됩니다. 컨볼루션 계층에서는 다양한 크기의 커널이 추출된 특징의 차원을 줄이기 위해 사용됩니다. 특징 추출 네트워크의 마지막 계층의 출력은 분류 네트워크의 입력으로 사용됩니다. 분류 네트워크는 더 나은 성능 때문에 완전히 연결된 피드포워드 신경망을 포함합니다 [21]. 그러나 이 계층은 모든 계층의 각 노드가 서로 완전히 연결되어 있기 때문에 네트워크 매개변수 측면에서 상대적으로 비용이 많이 듭니다. 최종 분류 결과는 정확한 클래스에 최대값을 부여하여 출력 계층 또는 Softmax 계층에서 계산됩니다. CNN 모델의 각 계층에 대한 자세한 내용은 [22]에서 찾을 수 있습니다.

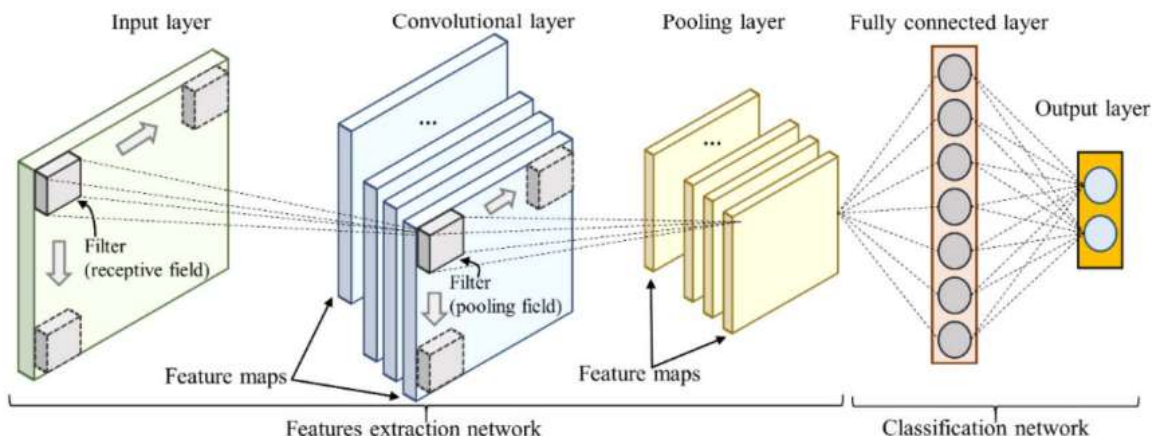


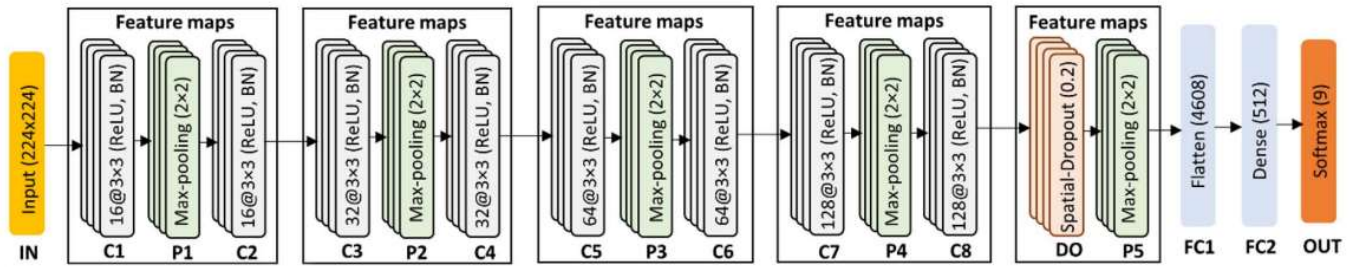
Fig. 2. The overall architecture of a convolutional neural network.

### III. PROPOSED DEEP CNN MODEL

A supervised learning approach is applied in the presence of labeled dataset. The WM-811K is a labeled dataset with 9 defect classes and each class contains complex features or patterns according to the defect nature. In this paper we constructed a 2D CNN model for WDI with one input layer, eight Conv layers each with Batch normalization (BN), padding, and Rectified Linear Unit (ReLU) activation, five Pool layers (four stacking pairs of Conv-Pool-Conv), one dropout layer, two fully connect (FC) layers, and one output layer. The layout of the proposed CNN model for WDI is shown in Fig. 3.

### 3. 제안 심층 CNN 모델

레이블이 지정된 데이터셋이 있는 경우 지도학습 접근 방식이 적용됩니다. WM-811K는 9개의 불량 클래스가 있는 레이블이 지정된 데이터셋이며 각 클래스에는 불량 특성에 따라 복잡한 특징이나 패턴이 포함되어 있습니다. 이 논문에서 우리는 하나의 입력층, 각각 배치 정규화(BN), 패딩 및 ReLU(Rectified Linear Unit) 활성화가 있는 8개의 Conv 계층, 5개의 풀 계층 (4개의 Conv-Pool-Conv 그룹), 1개의 드롭아웃 계층, 2개의 완전연결(FC) 계층 및 1개의 출력층을 갖는 2D CNN 모델을 만들었습니다. 제안하는 WDI용 CNN 모델의 레이아웃은 그림 3과 같다.



\*Note: IN denotes input layer; C convolutional layer; P pooling layer; DO dropout layer; FC fully connected layer; OUT output layer; and BN batch normalization

Fig. 3. Architecture of proposed deep CNN model for wafer defect identification.

The first Conv layer extracts the features from input training wafer images of size (224×224). Each Conv layer contains a set of learnable filters to extract unique feature maps. The number of filters increases with increasing depth of the Conv layer, so the number of feature maps also increases. However, feature maps become smaller and complex due to the Pool layer in deeper network. The proposed CNN-WDI model adopts 16, 32, 64, and 128 feature maps for the first, second, third, and fourth stacking pairs, respectively. As the number of stacking pairs increases, two results are induced: 1) the ability of CNN model to extract more distinct features, which is significant for WDI, is increased, and 2) the chance of loss of information due to small feature map size is reduced.

첫 번째 Conv층은 크기(224×224)의 입력 학습 웨이퍼 이미지에서 특징을 추출합니다. 각 Conv층에는 고유한 피쳐맵을 추출하기 위한 학습 가능한 필터 세트가 포함되어 있습니다. Conv층의 깊이가 증가함에 따라 필터의 수가 증가하므로 피쳐맵의 수도 증가합니다. 그러나 더 깊은 네트워크의 Pool 계층으로 인해 피쳐맵이 더 작고 복잡해집니다. 제안하는 CNN-WDI 모델은 첫 번째, 두 번째, 세 번째, 네 번째 Conv-Pool-Conv 그룹에 대해 각각 16, 32, 64, 128개의 피쳐맵을 사용합니다. Conv-Pool-Conv 그룹의 수가 증가할수록 1) WDI에 중요한 더 많은 특징을 추출하는 CNN 모델의 능력이 증가하고, 2) 작은 피쳐맵 크기로 인한 정보 손실 가능성이 줄어 들었습니다.

In this model, each Conv and Pool layer consists of sub-sampling filters of size 3×3 and 2×2, respectively. The small filter size assures the extraction of detailed features. The ReLU activation function was applied in all layers except Pool and output layers. The ReLU was proposed to solve the VGP during training process of deep learning models [8]. It simply keeps all the values above zero unchanged and replaces all negative values with zero. Although, the ReLU function significantly reduces the VGP at the start of training, but it doesn't guarantee that the same problem will not come back during further training. So, we used BN operation to address the VGP throughout the whole training process of the CNN model. The BN operation was designed to reduce the shift of internal covariance of DNNs. In other words, it lets the CNN model learn the optimal mean and scale of each input of the Conv layer by normalizing and zero-centering. The more detail of the BN method can be found in [23].

이 모델에서 각 Conv 및 Pool 계층은 각각 크기가 3x3 및 2x2인 서브샘플링 필터로 구성됩니다. 작은 필터 크기는 세부적인 특징의 추출을 보장합니다. **ReLU 활성화 함수**는 Pool층과 출력층을 제외한 모든 계층에 적용되었습니다. ReLU는 딥러닝 모델의 학습 과정에서 VGP를 해결하기 위해 제안되었습니다[8]. 단순히 0보다 큰 모든 값은 변경하지 않고 유지하고 모든 음수 값을 0으로 바꿉니다. ReLU 함수는 학습 시작 시 VGP를 크게 줄이지만 이후 학습 중에 같은 문제가 다시 발생하지 않는다는 보장은 없습니다. 그래서 CNN 모델의 전체 학습 과정에서 VGP를 해결하기 위해 BN 연산을 사용했습니다. **BN 연산**은 DNN의 내부 공분산의 변동을 줄이기 위해 설계되었습니다. 다시 말해, CNN 모델이 정규화 및 제로 센터링(평균을 0으로 만듦)을 통해 Conv층의 각 입력에 대한 최적의 평균과 크기를 학습할 수 있도록 합니다. BN 방법에 대한 자세한 내용은 [23]에서 찾을 수 있습니다.

For regularization of the CNN model, *Dropout* method is used which is a very simple but effective approach to prevent overfitting [24]. It improves the generalization performance of the network by avoiding activations from becoming highly correlated, which in turn leads to overfitting [25]. The main idea is to randomly remove or drop-out neurons and their connections by zeroing the activation of these neurons during training with some probability  $p_{drop}$ . We applied an advanced form of dropout called *SpatialDropout* (SD), proposed by Tompson *et al.* [26]. The standard dropout method fails to get required results when the network is fully convolutional, and activations of the feature maps are strongly correlated. The SD dropped-out entire feature maps of size  $n_f \times H \times W$  from the Conv layer which are then not used by pooling operation. Therefore, the pixels in the SD feature map are either all activated or all deactivated. Thus, SD helps to promote independence of feature maps of the Conv layer. We applied the SD function between the last Conv layer and last pooling layer with a rate of 0.2, that means 20% of the randomly selected nodes will be dropped-out for each weight update cycle.

CNN 모델의 과적합을 방지하기 위한 규제방법으로써 매우 간단하지만 효과적인 접근방식인 **드롭아웃(Dropout) 방법**이 사용됩니다 [24]. 이것은 활성화가 높은 상관관계가 되면서 결과적으로 과적합으로 이어지는 것을 방지함으로써 네트워크의 일반화 성능을 향상시킵니다[25]. 주요 개념은 일정 확률  $p_{drop}$ 으로 학습하는 동안 이러한 뉴런의 활성화를 0으로 하여 뉴런과 뉴런의 연결을 무작위로 제거하거나 탈락시키는(drop-out) 것입니다. 우리는 Tompson 등에서 제안한 **SpatialDropout(SD)**이라는 발전된 드롭아웃 형태를 적용했습니다. [26]. 표준 dropout 방법은 네트워크가 완전히 convolutional이고 피쳐맵의 활성화가 강한 상관관계가 있는 경우 필요한 결과를 얻지 못합니다. SD는 Conv층에서  $n_f \times H \times W$  크기의 전체 피쳐맵을 삭제한 다음 풀링 작업에서 사용하지 않습니다. 따라서 SD 피쳐맵의 픽셀은 모두 활성화되거나 모두 비활성화됩니다. 이처럼 SD는 Conv층의 피쳐맵의 독립성을 촉진하는 데 도움이 됩니다. 마지막 Conv층과 마지막 풀링층 사이에 **SD 함수를 0.2의 비율로 적용**했습니다. 즉, 각 가중치 업데이트 주기에 대해 무작위로 선택된 20%의 노드가 드롭아웃됩니다.

Zero padding was applied in all Conv layers to make sure that the dimensions of input and output feature maps are same. The Softmax activation function was applied to the output layer of our model. In addition, the Adam optimization method, which combines the concepts of Momentum optimization and root mean squared prop (RMSProp), was selected as the optimizer. This optimizer helps to achieve higher accuracy and improve training process of the CNN model [27]. Besides these, after many attempts, some other parameters like batch size and the number of epochs were assigned 100 and 20, respectively. The smaller batch size improves the generalization ability of the CNN model by computing an approximation of the gradient value and then updating the other parameters [28]. The detailed parameters of CNN-WDI model are given in Table I.

입력 및 출력 피쳐맵의 차원을 동일하게 유지하기 위해 모든 Conv 층에는 **제로 패딩(padding)**이 적용되었습니다. **Softmax 활성화 함수가 모델의 출력층에 적용**되었습니다. 또한 모멘텀 최적화와 RMSProp(Root Mean Squared Prop)의 개념을 결합한 **Adam 최적화** 방법을 옵티마이저로 선택했습니다. 이 옵티마이저는 더 높은 정확도를 달성하고 CNN 모델의 학습 과정을 개선하는 데 도움이 됩니다[27]. 이 외에도 많은 시도 끝에 배치 크기와 에포크(epoch) 수와 같은 몇몇 다른 매개변수에 각각 100 및 20이 할당되었습니다. 배치 크기가 클수록(←작을수록) 기울기 값의 근사값을 계산한 다음 다른 매개변수를 업데이트하여 CNN 모델의 일반화 능력이 향상됩니다 [28]. CNN-WDI 모델의 자세한 매개변수는 표 1에 나와 있습니다.

TABLE I  
PROPOSED DEEP CNN MODEL PARAMETERS

Layer	Type	Feature Maps	Output Size	Filter size	Padding	Activation
IN	Input	3 (RGB)	224×224	-	-	-
C1	Convolution1	16	222×222	3×3	No	ReLU
P1	Max Pooling1	16	111×111	2×2	No	-
C2	Convolution2	16	111×111	3×3	Yes	ReLU
C3	Convolution3	32	111×111	3×3	Yes	ReLU
P2	Max Pooling2	32	55×55	2×2	No	-
C4	Convolution4	32	55×55	3×3	Yes	ReLU
C5	Convolution5	64	55×55	3×3	Yes	ReLU
P3	Max Pooling3	64	27×27	2×2	No	-
C6	Convolution6	64	27×27	3×3	Yes	ReLU
C7	Convolution7	128	27×27	3×3	Yes	ReLU
P4	Max Pooling4	128	13×13	2×2	No	-
C8	Convolution8	128	13×13	3×3	Yes	ReLU
P5	Max Pooling5	128	6×6	2×2	No	-
FC1	Fully-Connected1	1	4608	-	-	ReLU
FC2	Fully Connected2	1	512	-	-	ReLU
OUT	Output	1	9	-	-	Softmax



<p style="text-align: center;">IV. EXPERIMENTS AND DISCUSSION</p> <p>In this section, we describe the training of the proposed CNN-WDI model, and its performance is compared with previously proposed models using various performance measures.</p>	<p style="text-align: center;">4. 실험 및 토의</p> <p>이 절에서는 제안한 CNN-WDI 모델의 학습에 대해 설명하고 다양한 성능지표를 사용하여 이전에 제안된 모델과 성능을 비교한다.</p>
<p><i>A. Training of CNN-WDI</i></p> <p>The training data was used to train the parameters of proposed CNN-WDI model which helps to reduce the loss function. We applied the categorical cross-entropy as our loss function to measure the loss function between estimated output probability distribution and actual class probability distribution. Batch normalization and spatial dropout with the probability of 0.2 were used for the regularization of the model. During the training, we applied backpropagation algorithm to calculate the gradient of loss function. Moreover, Adam Stochastic optimizer was applied to minimize the loss function with the learning rate and batch size of 0.001 and 100, respectively. The validation dataset is used to evaluate the model by fine-tuning its various hyperparameters. We evaluated the classification accuracy of our trained CNN-WDI model using this dataset. A well-trained deep learning model not only gets higher classification accuracy for training data but also for validation data. When we have all done for our training the model, next we have checked that how accurately it performs with testing dataset. The testing dataset is totally unseen for the model and it provides an unbiased evaluation of the final CNN-WDI model on the training dataset.</p>	<p>A. CNN-WDI의 학습</p> <p>학습 데이터는 손실함수를 줄이는 데 도움이 되는 제안된 CNN-WDI 모델의 매개변수를 학습하는 데 사용되었습니다. 추정된 출력 확률 분포와 실제 클래스 확률 분포 사이의 손실함수를 계산하기 위해 <b>범주형 교차 엔트로피를 손실 함수로 적용했습니다. 배치 정규화와 0.2의 확률로 공간 드롭아웃이 모델의 규제화에 사용되었습니다.</b> 학습 중에 손실 함수의 기울기를 계산하기 위해 역전파 알고리즘을 적용했습니다. 또한, <b>Adam Stochastic 옵티마이저를 적용하여 학습률과 배치 크기를 각각 0.001과 100으로 하여 손실 함수를 최소화했습니다.</b> 검증 데이터셋은 다양한 하이퍼파라미터를 미세 조정하여 모델을 평가하는 데 사용됩니다. 이 데이터셋을 사용하여 학습된 CNN-WDI 모델의 분류 정확도를 평가했습니다. 잘 학습된 딥러닝 모델은 학습 데이터뿐만 아니라 검증 데이터에 대해서도 더 높은 분류 정확도를 얻습니다. 모델 학습을 모두 마친 후 다음으로 테스트 데이터셋으로 모델이 얼마나 정확하게 수행되는지 확인했습니다. 테스트 데이터셋은 모델에 대해 완전히 보이지 않으며 학습 데이터셋에서 최종 CNN-WDI 모델에 대한 편견 없는 평가를 제공합니다.</p>
<p>The experiment was conducted on the personal computer with the following hardware specifications: Intel Xeon CPU E5-2696 v5 @ 4.40 GHz, 512 GB RAM, and NVIDIA GeForce GTX 1080 24 GB. The model was developed using <i>TensorFlow</i> [29] and <i>Keras</i> [30] libraries in <i>Jupyter Notebook</i> [31], which can handle different versions of Python language.</p>	<p>실험은 Intel Xeon CPU E5-2696 v5 @ 4.40GHz, 512GB RAM 및 NVIDIA GeForce GTX 1080 24GB의 하드웨어 사양을 가진 개인용 컴퓨터에서 수행되었습니다. 이 모델은 다양한 버전의 Python 언어를 처리할 수 있는 Jupyter Notebook[31]의 TensorFlow[29] 및 Keras[30] 라이브러리를 사용하여 개발되었습니다.</p>

### B. Performance Measures

The proposed CNN-WDI model is evaluated using different performance measures like accuracy, precision, recall, and F1-score. The accuracy of a classifier defines that how often it correctly predicts from whole data and defined as equation (1).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

where TP, FP, FN, and TN show the true positive, false positive, false negative, and true negative values, respectively. Both precision and recall are objection functions of any ML classifier which are defined as follows.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Equations (2) and (3) show that both performance measures are inversely proportional to each other and each of them has various classification measuring qualities. While, F1-score calculates the harmonic mean of precision and recall, and it is defined as follows.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

Equation (4) shows that *F1 - score* is the interpretation between actual and predicted probabilities. If these probabilities are close to each other than *F1 - score* will show the higher result and vice versa.

### C. CNN-WDI Results and Analysis

Table II presents the classification results of the CNN-WDI model using the original imbalanced dataset and the augmented balanced dataset. It can be seen in Table II that the proposed model got a very high value of precision, recall, and F1-score for all defect pattern classes for a balanced dataset. But the same model performed very poorly for most of the minority classes in the presence of an imbalanced dataset. For instance, the values of F1-score of minority classes like *Near-Full*, *Scratch*, and *Local* defect classes are only 74.2%, 76.6%, and 82.1%, respectively, which are comparatively very low. Moreover, this model got the value of the FI-score of *None* class up to 100% because it is a majority class of imbalanced dataset. The average values of the precision, recall, and F1-score of the CNN-WDI model for the balanced dataset are 96.24%, 96.24%, and 96.22%, respectively. Whereas, the same value for the imbalanced dataset are 90.32%, 86.39%, and 87.72, respectively. Thus, the data augmentation method improves the performance measure values of precision, recall, and F1-score up to 6.6%, 11.4%, and 9.7%, respectively. The reason is simple, CNN model gets better learning in the presence of large datasets. Therefore, these results show that the balancing dataset through data augmentation method can play a vital role in WM defect identification.

### B. 성능 평가

제안된 CNN-WDI 모델은 정확도, 정밀도, 재현율 및 F1 점수와 같은 다양한 성능지표를 사용하여 평가됩니다. 분류기의 정확도는 전체 데이터에서 얼마나 자주 정확하게 예측하는지로 정의하고 방정식(1)과 같이 정의됩니다.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

여기서 TP, FP, FN 및 TN은 각각 진양성, 가양성, 가음성 및 진음성 값을 나타냅니다. 정밀도와 재현율은 모두 다음과 같이 정의되는 모든 ML 분류기의 **목적(objective) 함수**입니다.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

식 (2)와 (3)은 두 성과지표가 서로 반비례하며 각각 다양한 분류 측정 품질을 가지고 있음을 보여준다. 반면 F1-score는 정밀도와 재현율의 조화평균을 계산하며 다음과 같이 정의된다.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

식 (4)는 F1-점수가 실제 확률과 예측 확률 사이의 해석임을 보여줍니다. 정확도와 정밀도가 서로 비슷하면(then) F1-점수는 더 높은 결과가 표시되며 그 반대의 경우도 마찬가지입니다.

### C. CNN-WDI 결과 및 분석

Table II는 원본 불균형 데이터셋과 증량 균형 데이터셋을 사용한 CNN-WDI 모델의 분류 결과를 보여줍니다. 표 2에서 제안된 모델은 균형 잡힌 데이터셋에 대한 모든 불량 패턴 클래스에 대해 매우 높은 정밀도, 재현율 및 F1-점수 값을 얻었음을 알 수 있습니다. 그러나 동일한 모델인데도 **불균형한 데이터셋이 있는 경우 대부분의 소수 클래스에 대해 매우 저조한 성능**을 보였습니다. 예를 들어 Near-Full, Scratch 및 Local 불량 클래스와 같은 소수 클래스의 F1-점수는 각각 74.2%, 76.6% 및 82.1%로 비교적 매우 낮습니다. 또한, 이 모델은 불균형 데이터셋의 다수 클래스이기 때문에 None 클래스의 F1-점수는 100%까지 얻었습니다. **균형잡힌 데이터셋에 대한 CNN-WDI 모델의 정밀도, 재현율 및 F1 점수의 평균은 각각 96.24%, 96.24% 및 96.22%입니다. 반면, 불균형 데이터셋에 대한 값은 각각 90.32%, 86.39% 및 87.72%입니다.** 따라서 데이터 증량 방법은 정밀도, 재현율 및 F1-점수의 성능지표를 각각 6.6%, 11.4% 및 9.7%까지 향상시킵니다. 그 이유는 간단합니다. CNN 모델은 대규모 데이터셋이 있을 때 더 나은 학습을 할 수 있기 때문입니다. 따라서 이러한 결과는 **데이터 증량 방법을 통한 균형잡힌 데이터셋이 WM 불량 식별에 중요한 역할**을 할 수 있음을 보여줍니다.

TABLE II  
PERFORMANCE EVALUATION OF CNN-WDI FOR BALANCED AND IMBALANCED DATASETS (%)

Defect Class	Balanced dataset			Imbalanced dataset		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Center	93.6	<b>98.0</b>	95.7	<b>94.7</b>	97.7	<b>96.2</b>
Donut	<b>96.0</b>	<b>98.3</b>	<b>97.2</b>	85.2	90.4	87.7
Edge-Loc	<b>97.4</b>	<b>93.1</b>	<b>95.2</b>	89.2	87.4	88.3
Edge-Ring	94.4	91.7	93.0	<b>97.7</b>	<b>98.3</b>	<b>98.0</b>
Local	<b>91.6</b>	<b>90.3</b>	<b>90.9</b>	82.8	81.4	82.1
Near-Full	98.9	<b>99.9</b>	<b>99.4</b>	<b>100.0</b>	59.1	74.2
Random	<b>96.7</b>	<b>96.5</b>	<b>96.6</b>	83.0	90.0	86.4
Scratch	<b>98.1</b>	<b>98.7</b>	<b>98.4</b>	80.4	73.2	76.6
None	99.5	99.7	99.6	<b>99.9</b>	<b>100.0</b>	<b>100.0</b>
Average	<b>96.24</b>	<b>96.24</b>	<b>96.22</b>	90.32	86.39	87.72

Note: Boldface numbers denote the highest values of different performance measures between balanced and imbalanced datasets

We draw a confusion matrix to represent the number of accurately classified and misclassified defect patterns of all classes as shown in Fig. 4, where *predicted label* and *true label* along the *x-axis* and *y-axis*, respectively, represents labels of various defect classes. The *predicted label* shows the number of predicted defect patterns of a specific class and the *true label* shows the numbers of actual defect patterns of the corresponding defect class. The main diagonal values show the ratio of correctly classified defect patterns. As the *Local* defect is very similar to *Donut*, *Edge-Loc*, and *Random* defect patterns, so CNN-WDI was confused during the feature extraction of these defect patterns. Thus, the number of misclassified of *Local* defect patterns with *Donut*, *Edge-Loc*, and *Random* defect patterns was up to 2.1%, 1.1%, and 2.1%, respectively. Due to various reasons, some WMs contain multiple defect patterns on their surfaces. Therefore, the CNN-WDI model misclassified some of the defect classes with other defect patters. The *Edge-Loc* class was misclassified with *Local* and *Center* classes with a ratio of 1.5% and 5.0%, respectively. The *Edge-Ring* class was misclassified with *Scratch* and *Local* classes with a ratio of 1.8% and 3.5%, respectively. The average classification result of the proposed model is up to 96.2%, which shows the importance of our model to classify WM defect patterns. The *None* defect class has no actual defect patterns in its wafer surface and this class contains the majority data like 85% of the original dataset of WM-811K. Thus, the higher classification result (i.e., 99.7%) of our model for the *None* defect class will boost the speed of testing process of WMs during semiconductor manufacturing.

우리는 그림 4와 같이 모든 클래스에 대해 정확하게 분류된 불량 패턴과 오분류된 불량 패턴의 수를 나타내는 혼동행렬을 그림합니다. 여기서 *x축*과 *y축*의 예측된 레이블과 실제 레이블은 각각 다양한 불량 클래스의 레이블을 나타냅니다. 예측 레이블은 특정 클래스의 예측된 불량 패턴 수를 나타내고 실제 레이블은 해당 불량 클래스의 실제 불량 패턴 수를 나타냅니다. 주대각선 값은 올바르게 분류된 불량 패턴의 비율을 나타냅니다. Local 불량은 Donut, Edge-Loc 및 Random 불량 패턴과 매우 유사하므로 이러한 불량 패턴의 특징 추출 중에 CNN-WDI가 혼동되었습니다. 따라서 Donut, Edge-Loc 및 Random 불량 패턴과 Local 불량 패턴의 오분류 수는 각각 2.1%, 1.1% 및 2.1%였습니다. 다양한 이유로 인해 일부 WM은 표면에 여러 불량 패턴을 포함합니다. 따라서 CNN-WDI 모델은 일부 불량 클래스를 다른 불량 패턴으로 잘못 분류했습니다. Edge-Loc 클래스는 각각 1.5% 및 5.0%의 비율로 Local 및 Center 클래스로 잘못 분류되었습니다. Edge-Ring 클래스는 Scratch 및 Local 클래스로 각각 1.8% 및 3.5%의 비율로 오분류되었습니다. 제안된 모델의 평균 분류 결과는 최대 96.2%로 WM 불량 패턴을 분류하는 모델의 중요성을 보여줍니다. None 불량 클래스는 웨이퍼 표면에 실제 불량 패턴이 없으며 이 클래스에는 WM-811K의 원래 데이터셋의 85%와 같은 대부분의 데이터가 포함됩니다. 따라서 None 불량 클래스에 대한 모델의 더 높은 분류 결과(즉, 99.7%)는 반도체 제조 중 WM의 테스트 프로세스 속도를 높일 것입니다.

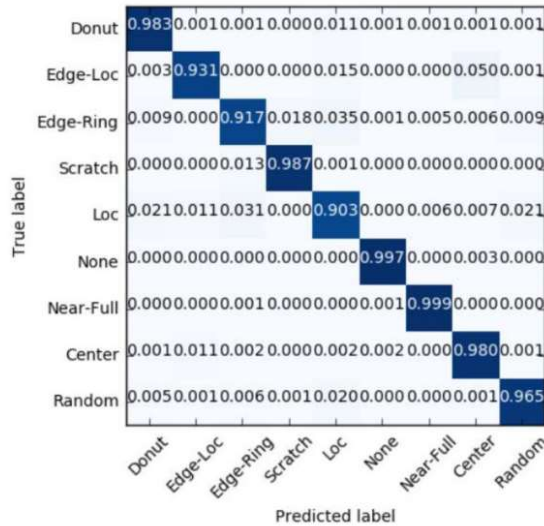


Fig. 4. A normalized confusion matrix of proposed CNN-WDI model.

Three more CNN classifiers (e.g., CNN-D, CNN-BN, and CNN-SD) with various hyperparameter settings were designed to evaluate the importance of different hyperparameters of the proposed model. The CNN-D model does not contain SD and BN methods, but only contains simple dropout method. The CNN-BN only contains BN but does not contain SD. The CNN-SD only contains spatial dropout but does not contain BN. The performances of all these CNN classifiers which contain various hyperparameter settings are compared with the proposed CNN-WDI classifier which contains both BN and SD methods.

다양한 하이퍼파라미터 설정을 가진 3개의 CNN 분류기(예: CNN-D, CNN-BN 및 CNN-SD)를 설계하여 **제안된 모델의 여러 하이퍼파라미터의 중요성을 평가**하였습니다. CNN-D 모델에는 SD 및 BN 방법이 포함되지 않고 단순 드롭아웃 방법만 포함됩니다. CNN-BN은 BN만 포함하고 SD는 포함하지 않습니다. CNN-SD는 공간적 드롭아웃만 포함하지만 BN은 포함하지 않습니다. 다양한 하이퍼파라미터 설정을 포함하는 이러한 모든 CNN 분류기의 성능은 **BN 및 SD 방법을 모두 포함하는 제안된 CNN-WDI 분류기와 비교**됩니다.

We also have compared the CNN-WDI classifier with state-of-the-art DL and ML classifiers like VGG-16, ANN, and SVM. The VGG-16 network is a well-known CNN model proposed to classify the image data with very high accuracy [32]. It composed of 16 layers including 13 Conv layer, 5 max-pooling layers, and 3 fully connected layers. Each Conv layer and first 2 fully connected layers have ReLU activation function, and the final layer has Softmax activation function for classification. The ANN model defines a relationship between input data and output value in the same way as derived from the animal brain [33]. We applied an ANN model with one input layer, one hidden layer with 100 neurons and ReLU activation function, and one output layer to classify the input data. The backpropagation algorithm and Adam optimizer were used to train and weight optimization of the ANN model. The SVM classifier is commonly used for image classification purpose especially for wafer defect classification [3]. The radial basis kernel function was implemented to handle the non-linear classification parameters due to complex image data. The penalty and stopping tolerance parameters were set to 1.0 and 0.001, respectively. The SVM is a binary classifier, so we applied one-vs-rest (OvR) method to classify multiple classes. We manually extracted Radon-based features by Radon transform from the raw wafer images to train ANN and SVM models. The same WM image dataset is used to train all the above classifiers as we used for CNN-WDI model.

또한 **CNN-WDI 분류기를 VGG-16, ANN 및 SVM과 같은 최첨단 DL 및 ML 분류기와 비교**했습니다. VGG-16 네트워크는 매우 높은 정확도로 이미지 데이터를 분류하기 위해 제안된 잘 알려진 CNN 모델입니다[32]. 13개의 Conv층, 5개의 max-pooling층, 3개의 완전연결층을 포함한 16개의 계층으로 구성됩니다. 각 Conv층과 처음 2개의 완전연결층에는 ReLU 활성화 함수가 있고 마지막 계층에는 분류를 위한 Softmax 활성화 함수가 있습니다. ANN 모델은 동물의 뇌에서 파생된 것과 같은 방식으로 입력 데이터와 출력값 간의 관계를 정의합니다[33]. 입력 데이터를 분류하기 위해 하나의 입력층, 100개의 뉴런과 ReLU 활성화 함수를 가진 하나의 은닉층, 하나의 출력층으로 구성된 ANN 모델을 적용했습니다. 역전파 알고리즘과 Adam 옵티마이저를 사용하여 ANN 모델의 학습 및 가중치 최적화를 수행했습니다. 특히 SVM 분류기는 웨이퍼 불량 분류를 위한 이미지 분류 목적으로 널리 사용됩니다[3]. 복잡한 이미지 데이터로 인한 비선형 분류 매개변수를 처리하기 위해 방사형 기저 커널 함수가 구현되었습니다. 패널티 및 정지 허용 오차 매개변수는 각각 1.0 및 0.001로 설정되었습니다. SVM은 이진 분류기이므로 다중 클래스를 분류하기 위해 **OvR(one-vs-rest) 방법을 적용**했습니다. ANN 및 SVM 모델을 학습하기 위해 원시 웨이퍼 이미지에서 **Radon 변환**을 통해 Radon 기반 특징을 수동으로 추출했습니다. CNN-WDI 모델에 사용한 것과 동일한 WM 이미지 데이터셋이 위의 모든 분류기를 학습하는 데 사용됩니다.

Table III presents the performance comparison of all classifiers with overall performance measures. It shows that the CNN-WDI model outperformed all the other classifiers to classify the wafer defects and acquire the highest value of overall validation accuracy, testing accuracy, precision, recall, and F1-score of 96.4%, 96.2%, 96.4%, 96.2%, and 96.2%, respectively. Testing accuracy is the most important component of performance measures because it uses unseen data to evaluate a model. The proposed model improved the testing accuracy up to 20.1%, 33.6%, and 195.1% of VGG-16, ANN, and SVM, respectively. The CNN-WDI model also got the better results as compare to the CNN-D, CNN-BN, and CNN-SD, which proves the importance of hyperparameters used in proposed model. Although, ANN model showed very good performance measures and got better results of training accuracy, validation accuracy, and F1-score. However, we need manually extracted features to train the ANN and SVM model. Moreover, these classifiers performed very poorly for minority data sample defect classes such as *Local*, *Random*, *Scratch*, and *Near-Full* defect classes. The VGG-16 has a very high number of trainable parameters (i.e., 134.2 million of parameters) as compared to the proposed model which has only 2.7 million trainable parameters, still our model improved the training accuracy up to 18.6%. The training time of VGG-16 was up to  $4.5 \times 10^5$ , whereas CNN-WDI model took only  $1.6 \times 10^5$  for training.

표 III은 모든 성능지표에 대해 모든 분류기의 성능 비교를 보여줍니다. CNN-WDI 모델이 웨이퍼 불량을 분류하는 데 있어 다른 모든 분류기보다 성능이 우수하여 전체 검증 정확도, 테스트 정확도, 정밀도, 재현율 및 F1-점수가 96.4%, 96.2%, 96.4%, 96.2%, 96.2%로써 가장 높은 값을 얻었음을 보여줍니다. 테스트 정확도는 모델을 평가하기 위해 보이지 않는 데이터를 사용하기 때문에 성능지표 중에서 가장 중요한 요소입니다. 제안된 모델은 VGG-16, ANN, SVM의 테스트 정확도를 각각 20.1%, 33.6%, 195.1% 향상시켰다. CNN-WDI 모델도 CNN-D, CNN-BN, CNN-SD에 비해 더 나은 결과를 보여 제안 모델에 사용된 하이퍼파라미터의 중요성을 입증했습니다. ANN 모델은 매우 우수한 성능 측정치를 보여주었고 학습 정확도, 검증 정확도 및 F1-점수에서 더 나은 결과를 얻었습니다. 그러나 ANN 및 SVM 모델을 학습하려면 수동으로 추출한 기능이 필요합니다. 더욱이 이러한 분류기는 Local, Random, Scratch 및 Near-Full 불량 클래스와 같은 소수 데이터 샘플 불량 클래스에 대해 매우 저조한 성능을 보였습니다. 학습 매개변수가 270만 개에 불과한 제안된 모델에 비해 VGG-16은 학습 매개변수의 수(즉, 1억 3,420만 개)가 매우 많지만, 여전히 우리 모델은 최대 18.6%의 학습 정확도를 개선했습니다. VGG-16의 학습 시간은 최대  $4.5 \times 10^5$ 인 반면 CNN-WDI 모델은 학습에  $1.6 \times 10^5$ 만 소요되었습니다. (단위가 초라면, 각각 5.2일, 1.9일로 약 1/3 소요)

TABLE III  
OVERALL PERFORMANCE COMPARISON OF VARIOUS CLASSIFIERS (%)

Classifier	Training Acc	Validation Acc	Testing Acc	Precision	Recall	F1-Score
CNN-WDI	98.9	<b>96.4</b>	<b>96.2</b>	<b>96.2</b>	<b>96.2</b>	<b>96.2</b>
CNN-D	97.6	95.5	95.2	95.2	95.2	95.2
CNN-BN	<b>99.4</b>	95.6	95.6	95.6	95.6	95.6
CNN-SD	98.6	94.7	94.8	94.8	94.8	94.8
VGG-16	82.3	80.0	80.1	80.3	80.1	79.9
ANN	95.9	95.9	72.0	95.2	95.9	95.4
SVM	91.3	91.0	32.6	87.5	91.0	88.0

Note: Boldface numbers denote the highest values of different performance measures and Acc accuracy

Table IV shows the testing accuracy performance comparison of CNN-WDI model with recently proposed wafer defect classification models such as WMFPR [3], DTE-WMFPR [5], and WMDPI [6], using the same dataset of WM-811K. As all defect patterns are equally important because each of these has a specific reason to occur. So, we cannot ignore any defect class during classification. It is clear from the table that deep learning-based CNN-WDI model outperformed all the previous models to classify the *Center*, *Donut*, *Edge-Loc*, *Local*, *Random*, *Scratch*, and *Near-Full* defect classes with the accuracy of 98.0%, 98.3%, 93.1%, 90.3%, 96.5%, 98.7%, and 99.9%, respectively. The classification accuracy of remaining defect classes is also very high such as above 92%. Whereas, all other classification models have poor accuracy to classify some of the defect classes. For example, WMFPR has 68.5% accuracy for *Local* class, WMDPI has 60.0% and 34.0% for *Local* and *Scratch* classes, respectively, and DTE-WMFPR has 83.5% for both *Edge-Loc* and *Local* classes. The average wafer defect classification accuracy of proposed model is 96.2%, which is improved up to 6.4% of the last highest average accuracy. Thus, these results show the importance of CNN-WDI model to classify wafer defect patterns.

Table IV는 CNN-WDI 모델과 동일한 WM-811K 데이터셋을 사용한 WMFPR[3], DTE-WMFPR[5], WMDPI[6]와 같은 최근 제안된 웨이퍼 불량 분류 모델과 테스트 정확도 성능 비교를 보여줍니다. 모든 불량 패턴은 각각 발생하는 특정 이유가 있기 때문에 똑같이 중요합니다. 따라서 분류 중에 불량 클래스를 무시할 수 없습니다. 딥러닝 기반 CNN-WDI 모델이 이전 모든 모델을 능가하여 Center, Donut, Edge-Loc, Local, Random, Scratch 및 Near-Full 불량 클래스를 98.0%, 98.3%, 93.1%, 90.3%, 96.5%, 98.7%, 99.9%의 정확도로 각각 분류하는 것을 분명히 알 수 있습니다. 나머지 불량 클래스의 분류 정확도도 92% 이상으로 매우 높습니다. 반면에 다른 모든 분류 모델은 일부 불량 클래스를 분류하는 정확도가 낮습니다. 예를 들어 WMFPR은 Local 클래스에 대해 68.5%의 정확도를, WMDPI는 Local 및 Scratch 클래스에 대해 각각 60.0% 및 34.0%, DTE-WMFPR은 Edge-Loc 및 Local 클래스에 대해 모두 83.5%의 정확도를 갖습니다. 제안된 모델의 평균 웨이퍼 불량 분류 정확도는 96.2%로 이전의 제일 높은 평균 정확도를 6.4%까지 향상시켰습니다. 따라서 이러한 결과는 웨이퍼 불량 패턴을 분류하는 CNN-WDI 모델의 중요성을 보여줍니다.

TABLE IV  
COMPARISON OF DEFECT CLASSIFICATION RESULTS FOR DIFFERENT CLASSIFICATION MODELS (%)

Classification model	Center	Donut	Edge-Loc	Edge-Ring	Local	Random	Scratch	Near-Full	None	Average
CNN-WDI	<b>98.0</b>	<b>98.3</b>	<b>93.1</b>	91.7	<b>90.3</b>	<b>96.5</b>	<b>98.7</b>	<b>99.9</b>	99.7	<b>96.2</b>
WMFPR [3]	84.9	74.0	85.1	79.7	68.5	79.8	82.4	97.9	95.7	83.1
WMDPI [6]	86.0	77.0	77.0	<b>95.0</b>	60.0	94.0	34.0	97.0	<b>100.0</b>	80.0
DTE-WMFPR [5]	95.8	92.2	83.5	86.8	83.5	95.8	86.0	N/A	100.0	90.4

Note: Boldface numbers denote the highest values of testing accuracies for various classification models

Fig. 5 shows the comparison of training accuracy and loss with validation accuracy and loss of CNN-WDI model and VGG network. In Fig. 5(a), the training accuracy of CNN-WDI model increases suddenly in initial epoch and then gradually increases with increasing number of epochs. The behavior of validation accuracy is opposite because it decreased initially and then increases gradually and reaches up to 96.4%. But the training process of VGG-16 is very slow and its validation accuracy stops increasing after achieving 80.0% as shown in Fig. 5(b). The same behavior can be observed for the training and validation losses of these models. The training and validation losses of proposed model decreased suddenly at initial stage and gradually decreased in next epochs and reached the value of 0.03 and 0.15, respectively, as shown in Fig. 5(c). Whereas, for the same number of epochs training and validation losses of VGG-16 network got the value of 0.48 and 0.80, respectively, as shown in Fig. 5(d).

그림 5는 CNN-WDI 모델 및 VGG 네트워크의 검증 정확도 및 손실 (loss)과 학습 정확도 및 손실을 비교한 것입니다. Fig. 5(a)에서 epoch 수에 따른 CNN-WDI 모델의 학습 정확도를 보면, 초기 epoch에서 급격히 증가하다가 그 이후 점진적으로 증가합니다. 검증 정확도의 추세는 반대로 처음에는 감소하다가 점차 증가하여 최대 96.4%에 도달합니다. 그러나 VGG-16의 학습 과정은 매우 느리고 그림 5(b)에서 보는 바와 같이 검증 정확도가 80.0%를 달성한 후 더 이상 증가하지 않습니다. 다른 모델들의 학습 및 검증 손실에 대해서도 동일한 추세를 관찰할 수 있습니다. 그림 5(c)와 같이 제안된 모델의 학습 및 검증 손실은 초기 단계에서 급격히 감소하고 다음 epoch으로 갈수록 점차 감소하여 각각 0.03 및 0.15 값에 도달했습니다. 반면, 동일한 Epoch 수에 대해 VGG-16 네트워크의 학습 및 검증 손실은 그림 5(d)와 같이 각각 0.48 및 0.80의 값을 얻었습니다.

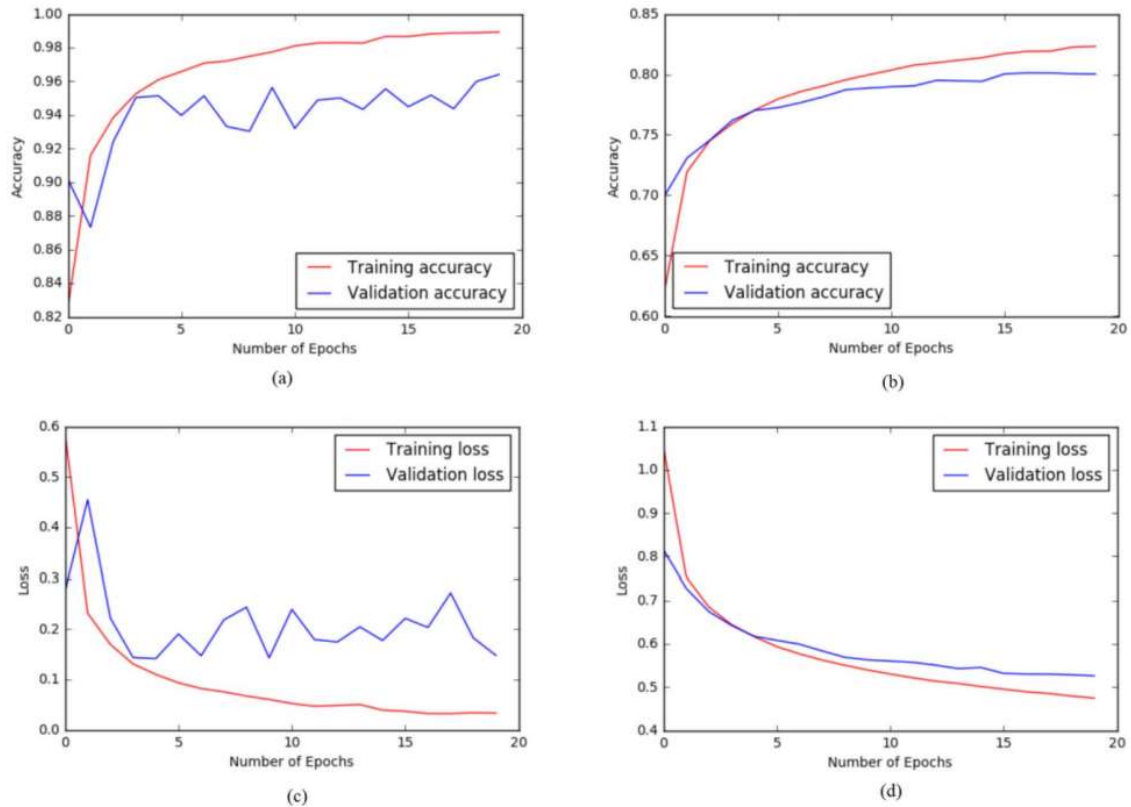


Fig. 5. The performance results of CNN-WDI model and VGG-16 model: (a) The accuracies of CNN-WDI, (b) The accuracies of VGG-16, (c) The losses of CNN-WDI, and (d) The losses of VGG-16.

## V. CONCLUSION

In this paper, a deep learning-based CNN-WDI model was proposed to classify wafer map defects in the semiconductor fabrication process. Semiconductor engineers apply automatic wafer classification for early diagnosis of wafer defects without requiring specialized or empirical knowledge. Most of the previous wafer defect analysis had applied machine learning-based classification models, which required manual feature extraction and a lot of hyperparameter settings, whereas CNN model can automatically extract effective features of various defect classes. We have used a real wafer map dataset of WM-811K, that contains nine different defect patterns. This dataset is highly imbalanced, so we have applied data augmentation technique to solve this issue. The sequence of convolutional and pooling layers has applied to extract valuable features from raw wafer images. Batch normalization and spatial dropout methods have been used for regularization of the proposed model which improves the training speed and classification accuracy of the model. Our model has got very high performance for all defect classes as compared to previously proposed models such as WMFPR, DTE-WMFPR, and WMDPI using the same dataset, and achieved the classification accuracy of 96.2% on average. The CNN-WDI outperformed VGG-16, SVM, and ANN classifiers. For future work, we will extract multiple defects on the same wafer image to improve classification accuracy.

## 5. 결론

본 논문에서는 반도체 제조공정에서 웨이퍼 맵 불량을 분류하기 위해 딥러닝 기반 CNN-WDI 모델을 제안하였다. 반도체 엔지니어는 전문 지식이나 경험적 지식 없이 웨이퍼 불량의 조기 진단을 위해 자동 웨이퍼 분류 기술을 사용합니다. 기존의 웨이퍼 불량 분석은 대부분 수동 특징 추출과 많은 하이퍼파라미터 설정이 필요한 머신러닝 기반 분류 모델을 적용한 반면 CNN 모델은 다양한 불량 클래스의 유효 특징을 자동으로 추출할 수 있습니다. 우리는 9개의 다른 불량 패턴을 포함하는 WM-811K의 실제 웨이퍼 맵 데이터셋을 사용했습니다. 이 데이터셋은 불균형이 심하므로 이 문제를 해결하기 위해 데이터 증량 기법을 적용했습니다. 일련의 컨볼루션 및 풀링 계층을 적용하여 원시 웨이퍼 이미지에서 중요한 특징을 추출했습니다. 제안한 모델의 규제화를 위해 배치 정규화와 공간적 드롭아웃 방법을 사용하여 모델의 학습 속도와 분류 정확도를 향상시켰습니다. 우리 모델은 동일한 데이터셋을 사용하여 기존에 제안된 WMFPR, DTE-WMFPR, WMDPI 등의 모델에 비해 모든 불량 클래스에 대해 매우 높은 성능을 보였고 평균 96.2%의 분류 정확도를 달성했습니다. CNN-WDI는 VGG-16, SVM 및 ANN 분류기를 능가했습니다. 향후 연구를 위해 동일한 웨이퍼 이미지에서 다중 불량을 추출하여 분류 정확도를 개선할 것입니다.

## REFERENCES

- [1] S. Cheon, H. Lee, C. O. Kim, and S. H. Lee, "Convolutional neural network for wafer surface defect classification and the detection of unknown defect class," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 163–170, May 2019, doi: [10.1109/TSM.2019.2902657](https://doi.org/10.1109/TSM.2019.2902657).
- [2] C. Wang, W. A. Y. Kuo, and H. Bensmail, "Detection and classification of defect patterns on semiconductor wafers," *IIE Trans.*, vol. 38, no. 12, pp. 1059–1068, Jan. 2006, doi: [10.1080/07408170600733236](https://doi.org/10.1080/07408170600733236).
- [3] M. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 1–12, Feb. 2015, doi: [10.1109/TSM.2014.2364237](https://doi.org/10.1109/TSM.2014.2364237).
- [4] C. H. Jin, H. J. Na, M. Piao, G. Pok, and K. R. Ryu, "A novel DBSCAN-based defect pattern detection and classification framework for wafer bin map," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 3, pp. 286–292, Aug. 2019, doi: [10.1109/TSM.2019.2916835](https://doi.org/10.1109/TSM.2019.2916835).
- [5] M. Piao, C. H. Jin, J. Y. Lee, and J. Y. Byun, "Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 250–257, May 2018, doi: [10.1109/TSM.2018.2806931](https://doi.org/10.1109/TSM.2018.2806931).
- [6] M. Saqlain, B. Jargalsaikhan, and J. Y. Lee, "A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 171–182, May 2019, doi: [10.1109/TSM.2019.2904306](https://doi.org/10.1109/TSM.2019.2904306).
- [7] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015, doi: [10.1016/j.neunet.2014.09.003](https://doi.org/10.1016/j.neunet.2014.09.003).
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: [10.1145/3065386](https://doi.org/10.1145/3065386).
- [9] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 2, pp. 135–142, May 2017, doi: [10.1109/TSM.2017.2676245](https://doi.org/10.1109/TSM.2017.2676245).
- [10] T. Nakazawa and D. V. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 309–314, May 2018, doi: [10.1109/TSM.2018.2795466](https://doi.org/10.1109/TSM.2018.2795466).
- [11] K. Kyeong and H. Kim, "Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 3, pp. 395–402, Aug. 2018, doi: [10.1109/TSM.2018.2841416](https://doi.org/10.1109/TSM.2018.2841416).
- [12] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, pp. 1–21, Feb. 2015, doi: [10.1186/s40537-014-0007-7](https://doi.org/10.1186/s40537-014-0007-7).
- [13] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 1, pp. 281–288, Feb. 2009, doi: [10.1109/TSMCB.2008.2002909](https://doi.org/10.1109/TSMCB.2008.2002909).
- [14] Mirlab.org. (2018). *MIR Corpora*. Accessed: Jan. 4, 2019. [Online]. Available: <http://mirlab.org/dataSet/public/>
- [15] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018, doi: [10.1109/TNNLS.2017.2732482](https://doi.org/10.1109/TNNLS.2017.2732482).
- [16] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "BAGAN: Data augmentation with balancing GAN," Jun. 2018. [Online]. Available: <https://arxiv.org/abs/1803.09655>.

## 참고문헌

- [17] E. D. Cubuk, B. Zoph, D. Man, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," Apr. 2019. [Online]. Available: <https://arxiv.org/abs/1805.09501>.
- [18] P. Kim, *MATLAB Deep Learning With Machine Learning, Neural Networks and Artificial Intelligence*. Heidelberg, Germany: Springer, 2017, doi: [10.1007/978-1-4842-2845-6](https://doi.org/10.1007/978-1-4842-2845-6).
- [19] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *Gen. Program. Evolvable Mach.*, vol. 19, no. 1, pp. 305–307, Jun. 2018, doi: [10.1007/s10710-017-9314-z](https://doi.org/10.1007/s10710-017-9314-z).
- [20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Int. Conf. Mach. Learn. (ICLM)*, Jun. 2010, pp. 807–814.
- [22] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *J. Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Jun. 2017, doi: [10.1162/NECO\\_a\\_00990](https://doi.org/10.1162/NECO_a_00990).
- [23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Mar. 2015. [Online]. Available: <https://arxiv.org/abs/1502.03167>.
- [24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jun. 2014. [Online]. Available: <http://jmlr.org/papers/v15/srivastava14a.html>
- [25] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," Jul. 2012. [Online]. Available: <https://arxiv.org/abs/1207.0580>.
- [26] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," Jun. 2015. [Online]. Available: <https://arxiv.org/abs/1411.4280>.
- [27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Jan. 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>.
- [28] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," Feb. 2017. [Online]. Available: <https://arxiv.org/abs/1609.04836>.
- [29] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," Mar. 2015. [Online]. Available: <https://arxiv.org/abs/1603.04467>.
- [30] Keras.io. (2018). *Keras Documentation*. Accessed: May 21, 2019. [Online]. Available: <https://keras.io/>
- [31] Jupyter.org. (2019). *Project Jupyter*. Accessed: May 15, 2019. [Online]. Available: <http://jupyter.org/>
- [32] K. Simonyan and Z. Andrew, "Very deep convolutional networks for large-scale image recognition," Apr. 2015. [Online]. Available: <https://arxiv.org/abs/1409.1556>.
- [33] R.-S. Guh, "On-line identification and quantification of mean shifts in bivariate processes using a neural network-based approach," *Qual. Rel. Eng.*, vol. 23, no. 3, pp. 367–385, Apr. 2007, doi: [10.1002/qre.796](https://doi.org/10.1002/qre.796).