Active Learning of Convolutional Neural Network for Cost-Effective Wafer Map Pattern Classification

Jaewoong Shim, Seokho Kang, and Sungzoon Cho

***Abstract*—Wafer maps provide important information for engineers for detecting root causes of failure in a semiconductor manufacturing process. Thus, there has been active research into the automation of wafer map pattern classification. With recent advances in deep learning, a convolutional neural network (CNN) has yielded state-of-the-art performance in wafer map pattern classification. Because a large amount of labeled training data is required, experienced engineers need to annotate large quantities of wafer maps manually which is costly. To construct a well-performing CNN model with a lower labeling cost, we propose a cost-effective wafer map pattern classification system based on the active learning of a CNN. In the system, a CNN model is constructed based on four main steps: uncertainty estimation, query wafer selection, query wafer labeling, and model update. By repetitively performing these steps, the performance of the CNN model is gradually and effectively increased. We compared several methods for uncertainty estimation and query wafer selection in our system. We demonstrated the effectiveness of the proposed system through experiments using real-world data from a semiconductor manufacturer.**

***Index Terms*—Wafer map pattern classification, active learning, convolutional neural network, uncertainty estimation.**

I. Introduction

IN THE semiconductor manufacturing process, wafers go through hundreds of fabrication steps which are at the core of the entire manufacturing process. Then, all wafers are subjected to a "wafer test" before packaging and final testing. In the wafer test step, all dies in each wafer undergo various electrical inspections to examine whether they can perform the required normal operations. As a result of testing, each die is classified as either pass or fail. The spatial information of whether each die is pass or fail constitutes a "wafer map". The wafer map shows how the fail dies are distributed on the wafer.

Manuscript received November 25, 2019; revised January 21, 2020 and February 11, 2020; accepted February 15, 2020. Date of publication February 19, 2020; date of current version May 5, 2020. This work was supported by the National Research Foundation of Korea (NRF) through the Korea Government (Ministry of Science and ICT) under Grant NRF-2017R1C1B5075685 and Grant NRF-2019R1A4A1024732. *(Corresponding author: Sungzoon Cho.)*

Jaewoong Shim and Sungzoon Cho are with the Department of Industrial Engineering and Institute for Industrial Systems Innovation, Seoul National University, Seoul 08826, South Korea (e-mail: shimjw@dm.snu.ac.kr; zoon@snu.ac.kr).

Seokho Kang is with the Department of Systems Management Engineering, Sungkyunkwan University, Suwon 16419, South Korea (e-mail: s.kang@skku.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSM.2020.2974867

The wafer map provides engineers with very important information for root cause identification. Wafers with similar spatial failure patterns are likely to have the same root cause in the fabrication process. For example, if wafers with the same wafer map pattern appear more frequently than a certain level in a short period of time, engineers can take action, e.g.,, finding a common fabrication facility or process for those wafers. Therefore, wafer map pattern classification is an important task for finding the cause of a failure, and provides clues to improve the manufacturing quality.

With this necessity and importance, there has been considerable research on wafer map pattern classification. Especially recently, deep learning-based research has been actively carried out [1], [2]. A convolutional neural network (CNN) is a typical deep learning-based methodology used in wafer map pattern classification. One requirement for the success of a deep learning methodology is a sufficient amount of labeled training data. However, it is too costly for experienced engineers to carry out manual labeling for the enormous number of wafers produced in the manufacturing process. It is even more difficult to obtain a sufficient number of wafer maps with spatial failure patterns, as most wafers do not have many failure dies, or correspond to non-pattern wafers. If we randomly sample wafers to be labeled, most wafers may correspond to a non-pattern wafer or may have spatial patterns similar to already-labeled ones. Thus, they would not be informative for improving the model. It is necessary to selectively annotate novel pattern wafers that have not yet been incorporated into the training set to improve the efficiency of the labeling.

In this study, we propose an cost-effective wafer map pattern classification system based on active learning of a CNN to address the above-mentioned problems. In the proposed system, there are four major steps: uncertainty estimation, query wafer selection, query wafer labeling, and model update. The CNN model for classifying wafer maps is trained by using an initial labeled set. With this model, the uncertainty of prediction in the unlabeled wafer maps is calculated in the uncertainty estimation step. Based on this uncertainty, the wafers to be labeled are selected in the query wafer selection step. In the query wafer labeling step, the selected wafers are inspected by the engineer, and are merged into the labeled set. The CNN model is updated with the labeled set in the model update step. Through the repetition of these four steps, the performance of the CNN model is gradually increased. This method is cost-effective, because it can achieve higher performance with a lower labeling cost. Several methods for uncertainty estimation and query wafer selection are compared

in this framework. The effectiveness of the proposed system is investigated through experiments using real-world data from a semiconductor manufacturer.

The main contributions of this work are summarized as follows. Firstly, we implement an efficient active learning system to build a CNN model with lower labeling cost for wafer map pattern classification. Secondly, diversified top-K selection method is proposed for the query wafer selection step that can alleviate class imbalance problem. Thirdly, we explore various uncertainty estimation methods to find the most suitable one for wafer map pattern classification.

The remainder of this paper is organized as follows. In Section II, we review the related work concerning active learning and wafer map pattern classification. In Section III, we introduce the proposed method. Section IV reports the experimental results. Finally, the conclusion and future work are provided in Section V.

II. Related Work

A. Wafer Map Pattern Classification

Numerous studies have been conducted on wafer map pattern classification. Most studies have focused on how to extract good features from wafer maps to classify their patterns. Wu *et al.* [3] performed wafer map classification through a support vector machine (SVM) by extracting geometry-based and radon-based features. They released the WM-811K dataset used in the experiment for our study. Jeong *et al.* [4] developed a methodology using a spatial correlogram to find spatial autocorrelation, and used it to classify wafer map patterns. Yu and Lu [5] proposed a method for extracting useful information from various features by using joint local and nonlocal linear discriminant analysis (JLNDA). Piao *et al.* [6] proposed a decision tree ensemble learning methodology using radon-based features. Fan *et al.* [7] performed multi-label classification of a wafer map using the "Ordering Point to Identify the Cluster Structure" (OPTICS) approach and a SVM. Saqlain *et al.* [8] proposed a voting ensemble classifier based on density, geometry, and radon-based features.

With advances in deep learning, CNNs have been actively applied to the semiconductor manufacturing process. Unlike traditional machine learning methods, a CNN enables automatic extraction of meaningful features from raw inputs without manual feature engineering. For the reason, many recent studies have demonstrated the effectiveness of CNN on learning from various types of data produced in the manufacturing process. Lee *et al.* [9] used a CNN to extract useful features for fault detection. Kim *et al.* [10] proposed a self-attentive CNN to detect faults from variable-length sensor data. As for the wafer map pattern classification problem, Nakazawa and Kulkarni [1] built a CNN model with synthetic wafer maps and applied it to the classification of actual wafer maps. Kyeong and Kim [2] proposed constructing an individual CNN model for each defect type to classify wafer maps with mixed-type defect patterns. Nakazawa and Kulkarni [11] employed convolutional autoencoder to detect unseen wafer map pattern. Wang *et al.* [12] adopted adversarial learning to improve a CNN to better address imbalanced distribution of wafer map patterns. The research mentioned so far aimed to improve the performance of wafer map pattern classification through CNN. However, there has been little effort to reduce labeling cost that necessarily comes with deep learning. This work addresses efficient labeling of wafers based on a CNN-based active learning system.

B. Active Learning

Active learning is a case of machine learning, in which a prediction model is built using active querying, by which instances are labeled for training. This method is used when it is difficult to label all instances because of a high labeling cost. In an active learning framework, informative instances are selected to be labeled by the labeler.

Uncertainty sampling is the most basic and traditional strategy, and queries the most uncertain instances from among the unlabeled set. An easy method for estimating uncertainty is to utilize the posterior probability of a predicted class [13]. Settles and Craven [14] proposed to query instances with the least confident instance. The margin between the highest posterior probability and the second highest probability was considered as the uncertainty in [15]. The entropy of the class posterior probabilities was exploited as an uncertainty measure in [16].

Some studies have also considered diversity of selected instances in uncertainty sampling methods because redundant instances add little information for updating the model. In Brinker [17], a diversity constraint was introduced for active learning with SVM. Yang *et al.* [18] proposed a method of applying diversity maximization constraints to a multi-class problem. Xu *et al.* [19] selected the cluster centers of the instances lying within the margin of a SVM. Hoi *et al.* [20] chose instances that reduce Fisher information. Azimi *et al.* [21] used a variant of a Gaussian Mixture Model to maintain diversity among selected instances.

Query-by-committee [22] is another well-known technique which employs multiple prediction models. The prediction is performed for all unlabeled instances and the instance that is the most inconsistent between prediction models is queried. Other methods include the method of expected error reduction [23], which estimates the expected error after a set of queries to find the optimal query instances, and the total expected variance minimization method [14], [24], which reduces the generalization error indirectly.

Recently, with the development of deep learning, active learning algorithms for deep neural networks have also been appearing. However, they are not prevalent compared to the algorithms for traditional machine learning [25]. Most of the methods are based on uncertainty sampling because they can compensate for the high computational cost of deep neural networks. Simple uncertainty sampling methods using posterior probability have been successfully applied to deep neural networks [26]. In applying Bayesian methods to deep learning, Gal and Ghahramani [27] showed an equivalence between a dropout and the approximate Bayesian inference. Thus, it was possible to estimate uncertainty through multiple forward passes with Monte Carlo (MC) dropout [25].

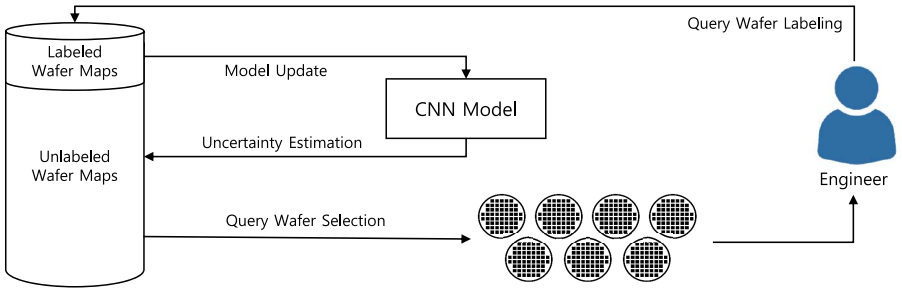

Fig. 1. Overview of the proposed wafer map pattern classification system.

In active learning for deep neural networks, there have been studies to take the diversity of the selected instances into account. Sener and Savarese [28] proposed a distribution-based method for choosing instances representing the distribution of the entire unlabeled pool in an intermediate feature space of a deep neural network. But it has a high computational cost because it requires solving mixed integer programming. Kirsch *et al.* [29] proposed batchBALD method that increase diversity of selected instances with approximation of mutual information. BatchBALD recently reported superior performance in batch mode active learning tasks.

Our active learning system uses a CNN for a classification model. We focus on uncertainty sampling owing to its simplicity and computational efficiency. Furthermore, we consider the diversity of the selected instances in order to solve the class imbalance problem. We aim to investigate various possible uncertainty estimation methods and query selection methods that work with a CNN.

III. Proposed Wafer Map Pattern Classification System

A. System Overview

In this section, we introduce the entire structure of the proposed wafer map pattern classification system. The goals of this system are to reduce the amount of wafer maps inspected directly by engineers, and to build a classifier that achieves high classification performance with a small amount of labeled wafer maps. Fig. 1 illustrates an overview of the proposed system. The system consists of four main steps: uncertainty estimation, query wafer selection, query wafer labeling, and model update.

At the start of this system, and in view of the large number of wafer maps that are not yet classified, the engineer randomly selects and inspects wafer maps manually. These randomly selected and labeled wafer maps constitute the training set of the initial model. We construct the initial CNN model with this small number of wafer maps. This CNN model performs a prediction on the unlabeled wafer maps, and also estimates the uncertainty of the prediction (uncertainty estimation). Based on these uncertainty values, the system selects wafer maps to be labeled by the engineer (query wafer selection). The engineer inspects the selected wafer maps (query wafer labeling). Newly-labeled wafer maps are integrated with existing labeled wafer maps, and the CNN model is updated with these labeled set (model update). Through repetition of this process, the CNN model is continually updated using the informative wafer maps, and eventually a high-performance classification model can be obtained. Because the engineer only inspects and labels a small number of selected wafers in each repetition phase, this process is more cost-effective than inspecting all of the wafers.

TABLE I
Notations

Notation	Type	Description
$\mathbf{X}_i$	matrix	i-th wafer map in $\mathcal{D}$
y_i	scalar	label for $\mathbf{X}_i$
$\mathcal{D}$	set	whole wafer map dataset $\mathcal{D} = \{\mathbf{X}_i\}_{i=1}^N$
$\mathcal{D}^L$	set	labeled subset of $\mathcal{D}$.
$\mathcal{D}^U$	set	unlabeled subset of $\mathcal{D}$
K	scalar	query size for each repetition phase
T	scalar	number of iterations of forward passes for MC dropout
$\mathcal{W}$	set	CNN parameters
$\mathcal{W}_t$	set	randomly dropped CNN parameters in $\mathcal{W}$ for t-th iteration for MC dropout

Algorithm 1 Pseudocode for Proposed System

Input: Dataset $\mathcal{D}$, query size per repetition phase K
Output: CNN parameters $\mathcal{W}$

procedure
  $\mathcal{D}^L \leftarrow$ random sample from $\mathcal{D}$
  Label each wafer map in $\mathcal{D}^L$
  $\mathcal{D}^U \leftarrow \mathcal{D} \setminus \mathcal{D}^L$
  Initialize $\mathcal{W}$ with $\mathcal{D}^L$
  while not reach maximum phase **do**
    Estimate uncertainty for each wafer map in $\mathcal{D}^U$
    $Q \leftarrow K$ most uncertain wafer maps from $\mathcal{D}^U$
    Label each wafer map in Q
    Fine-tune $\mathcal{W}$ with $\mathcal{D}^L \cup Q$
    $\mathcal{D}^L \leftarrow \mathcal{D}^L \cup Q$, $\mathcal{D}^U \leftarrow \mathcal{D}^U \setminus Q$
  end while
  return $\mathcal{W}$
end procedure

The notations for explaining the proposed system are presented in Table I. Algorithm 1 presents the pseudocode for the entire process. The following subsections describe the four main steps.

B. Prediction Model

We adopt a CNN as a classification model for use in this system. The CNN is one of the representative methodologies of deep learning and has achieved state-of-the-art performance in various applications, especially in image classification tasks.

Many studies have attempted to classify wafer map patterns using CNNs, and have achieved remarkable performance.

A CNN consists of convolution layers, pooling layers, and fully-connected layers. Usually, the convolution layer is used to extract features from the input data. The pooling layer serves to reduce the dimensions of the input data. Classification is performed in the fully-connected layers, using the features extracted from the convolution and pooling layers. With this CNN architecture, there is no need to execute extra feature engineering.

It is possible to employ an existing CNN model whose architecture has been optimized for model performance. As will be discussed in Section III-C, if a Bayesian approximation is used, the model architecture should include a dropout.

C. Uncertainty Estimation

A representative method for estimating the uncertainty of prediction by the CNN model involves using the output value of the last softmax layer. As the softmax output values can be interpreted as probabilities of belonging to individual classes, there are several methods that use these values. Typically, the uncertainty is quantified by least confidence, margin, and entropy measures.

All equations in this section represent the uncertainty of prediction for the i-th wafer. We will use the following equation to simplify the equations used for uncertainty estimation. t represents each forward pass for MC dropout and has an integer value from 1 to T, i.e.,, $t \in \{1, \ldots, T\}$.

$$\begin{aligned} p_j(\mathbf{X}_i, y_i) &= p(y_i = j|\mathbf{X}_i; \mathcal{W}) \\ p_j^t(\mathbf{X}_i, y_i) &= p(y_i = j|\mathbf{X}_i; \mathcal{W}_t). \end{aligned} \tag{1}$$

1) Least Confidence: The probability of the most probable class for an instance is called the confidence. If it is low, the uncertainty of the model for this instance is high. Thus, we can use the negative of the confidence as an uncertainty measure [14].

$$lc_i = -\max_j p_j(\mathbf{X}_i, y_i). \tag{2}$$

2) Least Margin: If the probability of the most probable class shows a large difference from the probability of the second-most probable class, the prediction can be regarded as certain. The margin is the difference between the values of the highest posterior probability and the second highest posterior probability [15]. Thus, the negative of the margin is considered as an uncertainty estimator. j_1 and j_2 in the following equation represent the first and second most probable class labels classified by the model.

$$lm_i = -\big(p_{j_1}(\mathbf{X}_i, y_i) - p_{j_2}(\mathbf{X}_i, y_i)\big). \tag{3}$$

3) Entropy: Entropy is considered as an uncertainty estimator which uses all class label probabilities [16].

$$en_i = -\sum_{j=1}^{C} p_j(\mathbf{X}_i, y_i) \log p_j(\mathbf{X}_i, y_i) \tag{4}$$

Another approach for estimating the uncertainty is to use a Bayesian model. A Bayesian model is capable of estimating uncertainty, because it produces a distribution of predictions. However, a Bayesian model is not suitable for practical usage, because it is computationally intensive in terms of both training and inference. Thus, we adopt MC dropout [27] technique for Bayesian approximation of a CNN model. Generally, dropout is used in the training process to solve the overfitting problem. If inferences are done with dropout activated, the model produces a different output for each forward pass. The uncertainty for an instance is estimated based on the statistics of the outputs from multiple forward passes of the instance. So, in our system, dropout is activated when the model is being updated or when estimating the uncertainty through MC dropout. Several uncertainty estimation methods based on MC dropout are introduced as below.

4) Predictive Entropy: The entropy is calculated using the average of the results of the multiple forward passes.

$$pe_i = -\sum_{j=1}^{C} \left(\frac{1}{T}\sum_{t=1}^{T} p_j^t(\mathbf{X}_i, y_i)\right) \log\left(\frac{1}{T}\sum_{t=1}^{T} p_j^t(\mathbf{X}_i, y_i)\right). \tag{5}$$

5) Bayesian Active Learning by Disagreement (BALD): This measure corresponds to the mutual information between predictions and model parameters $\mathcal{W}$ [30]. If a model has high uncertainty regarding an instance on average, but some forward passes make predictions that disagree with each other with high certainty, that instance will maximize this measure. This measure is calculated as follows.

$$bd_i = pe_i - \frac{1}{T}\sum_{t=1}^{T}\left(-\sum_{j=1}^{C} p_j^t(\mathbf{X}_i, y_i) \log p_j^t(\mathbf{X}_i, y_i)\right). \tag{6}$$

6) Variation Ratio: In the multiple forward passes, the relative ratio of the number of times classified into each class to the total can be regarded as a probability. Therefore, the largest value of this ratio represents the degree of certainty, and similar to confidence, the smaller the value, the higher the uncertainty [31]. This measure is calculated as follows.

$$vr_i = 1 - \max_i \left(\frac{1}{T}\sum_{t=i}^{T} \mathbb{1}\left(i = \arg\max_j p_j^t(\mathbf{X}_i, y_i)\right)\right). \tag{7}$$

7) Mean Standard Deviation (Mean-STD): The standard deviation of the probability is calculated for each class, and the average is used [32].

$$ms_i = \frac{1}{C}\sum_{j=1}^{C}\sqrt{\frac{1}{T}\sum_{t=1}^{T} p_j^t(\mathbf{X}_i, y_i)^2 - \left(\frac{1}{T}\sum_{t=1}^{T} p_j^t(\mathbf{X}_i, y_i)\right)^2}. \tag{8}$$

D. Query Wafer Selection

1) Simple Top-K Selection: We can calculate the uncertainty with the methods mentioned above for each unlabeled wafer. Assuming that K unlabeled wafers were selected and queried, it is common that K wafers with high uncertainty are selected. This selection is commonly used for uncertainty-based active learning of a CNN [25], [26].

2) Diversified Top-K Selection: This situation, which requires selecting K query instances at a time, corresponds to a batch mode active learning [28]. The K wafers selected through the simple top-K selection are likely to have similar spatial patterns, and thus can provide redundant information. Therefore, when sampling the K wafers, the diversity must be considered.

To prevent redundant selection of similar wafers, we propose to use a diversified top-K selection method. Instead of simply choosing the K wafers with the highest uncertainty among all wafers, we select the wafers with the highest uncertainty within each of the predicted classes.

E. Query Wafer Labeling

After the query wafers are chosen, the engineers proceed to label them manually, using visual recognition. These newly-labeled wafers are then merged into the existing labeled-wafers set.

F. Model Update

Once the labeled wafer set has been updated, a new CNN model must be built. However, training the model from scratch takes a long time thus inefficient. To make it efficient, we fine-tune the existing CNN model of the previous phase with the updated labeled set. This method is more time-efficient than training the model from scratch because the model from the previous phase already has information about existing labeled wafers. We set the loss for the newly-labeled wafer larger than the loss for the existing labeled wafer, so that the information in the newly-labeled wafer can be learned quickly while maintaining the information in the existing labeled wafer. The objective function used for fine-tuning can be described as follows. λ is a hyperparameter that represents the ratio of the loss for the newly-labeled wafer to the loss for the existing labeled wafer. It should be larger than 1.

$$\min_{\mathcal{W}} -\frac{\lambda}{K}\sum_{\{i|\mathbf{X}_i\in Q\}}\sum_{j=1}^{C}\mathbb{1}(y_i=j)\log p_j(\mathbf{X}_i,y_i) - \frac{1}{|\mathcal{D}^L|}\sum_{\{i|\mathbf{X}_i\in\mathcal{D}^L\}}\sum_{j=1}^{C}\mathbb{1}(y_i=j)\log p_j(\mathbf{X}_i,y_i). \quad (9)$$

IV. Experiments

A. Data Description

We conducted experiments to demonstrate the effectiveness of the proposed wafer map pattern classification system. The dataset used in the experiment is the WM-811K dataset, which is a real-world fabrication dataset. The dataset[1] was first released in [3] It consists of 811,457 wafer maps from a total of 46,294 lots. The defect type is labeled for approximately 20% of the maps, or 172,950 wafer maps. In this experiment, only the 172,950 labeled instances were utilized. There are nine classes in total: *Non-Pattern*, *Edge-Ring*, *Edge-Loc*, *Center*, *Loc*, *Scratch*, *Random*, *Donut*, and *Near-Full*. Fig. 2 shows examples of each class.

[1]http://mirlab.org/dataset/public/

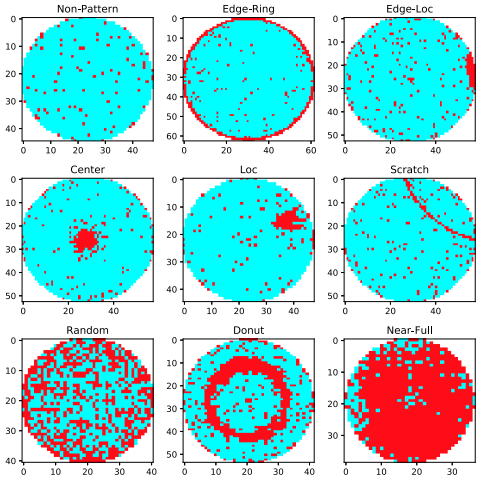

Fig. 2. Examples of wafer maps and their labels.

TABLE II
Dataset Description

Defect Type	No. Train	No. Test	No. Total
Non-Pattern	36,730	110,701	147,431
Edge-Ring	8,554	1,126	9,680
Edge-Loc	2,417	2,772	5,189
Center	3,462	832	4,294
Loc	1,620	1,973	3,593
Scratch	500	693	1,193
Random	609	257	866
Donut	409	146	555
Near-Full	54	95	149
Total	54,355	118,595	172,950

Table II shows the distribution of classes in the dataset, which are highly imbalanced. The *Non-Pattern* class occupies an overwhelming proportion, whereas the *Donut* and *Near-Full* classes occupy 0.3% and 0.1%, respectively. As in [3], we used 54,355 wafer maps of the dataset to build a wafer map pattern classification system. The remaining 118,595 wafer maps were used to evaluate the performance of the system. Because CNN requires fixed-size inputs, all wafer maps were resized to (64, 64).

B. Experimental Design

In the experiments, we simulated the proposed wafer map pattern classification system. Similar to the related literature [1], [2], we used a LeNet-5 [33]-like CNN architecture shown in Fig. 3. Our architecture is relatively smaller than modern CNN architectures such as AlexNet [34] and VGGnet [35]. This is adequate for our problem because wafer maps are much simpler than conventional image representations and the prediction model must be trained with a small amount of instances in the active learning setting. At the beginning, 400 randomly selected wafers were labeled. Setting aside 200 wafers as the validation set, the other 200 wafers were used as the initial training set. All the remaining wafers were used as the initial unlabeled set. This corresponds to real-world situations, where engineers inspect randomly selected unlabeled wafer maps. Extreme minority classes, such as *Donut*

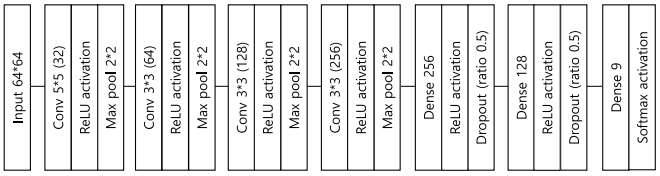

Fig. 3. CNN architecture used in this study.

and *Near-Full*, may not have been included in the initial training set. For the query wafer selection step, the query size K was set to the same as the number of classes. The query wafer labeling step was simulated by revealing the corresponding labels in the original dataset.

We used the Adam optimizer [36] with a learning rate of 0.001 and a batch size of 128. Each model training was terminated if the validation loss failed to decrease over 20 consecutive phases. For the model update step, the hyperparameter λ in Equation (9) was set to 10 by conducting a preliminary experiment to investigate its effect on the performance. For the uncertainty estimation step, the number of forward passes was set to 50 when MC dropout was used.

We conducted experiments on the seven uncertainty estimation methods described in Section III-C, and applied the two wafer selection methods described in Section III-D, for a total of 14 combinations. To succinctly represent each combination, suffixes '_d' or '_s' are used to indicate diversified top-K selection and simple top-K selection, respectively. As baselines, we used BatchBALD [29], random selection method, and full model. The random selection method randomly selects wafers to be labeled without uncertainty estimation. The full model is trained with the entire training set, assuming that the dataset is completely labeled.

The performance of the CNN model for wafer map pattern classification was evaluated on the test set in terms of the area under the receiver operating characteristic curve (AUROC). Dropout in the model is deactivated when making predictions for performance evaluation. Because the original AUROC is for binary classification, we used a multi-class modification of the AUROC [37]. The multi-class AUROC is calculated by taking the average of multiple AUROCs, each of which having been obtained separately by binary classification of discriminating a single class from all of the other classes.

All experiments were performed with 10 independent repetitions with different random seeds. We report an average over the 10 repetitions.

C. Results and Discussion

Fig. 4 shows the comparison results of the seven uncertainty estimation methods with the diversified top-K selection. The results of the three baseline methods are also shown. The X-axis represents the repetition phase of the proposed active learning system. The performance increased as the phase progressed. As shown in the figure, all seven of the uncertainty estimation methods were superior to the random selection method. While the random selection method showed a slower performance increase per phase, proposed method had a faster performance increase. Among the seven uncertainty

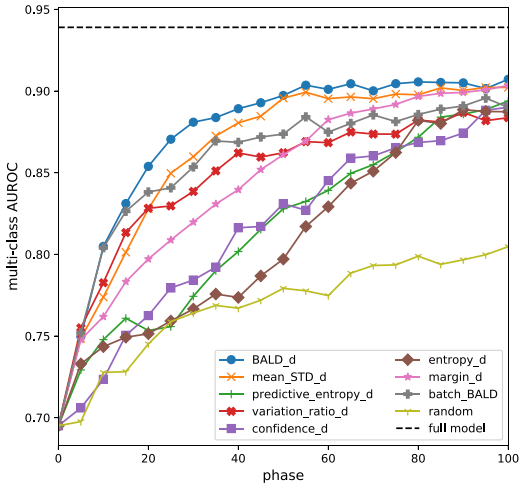

Fig. 4. Overall comparison of uncertainty estimation methods with diversified top-K selection.

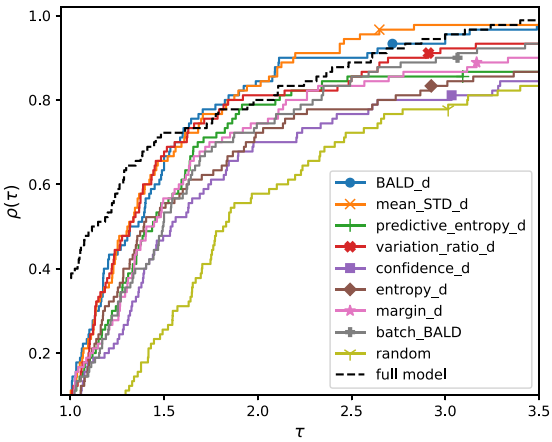

Fig. 5. Dolan-More curves of the CNN models at the 60th phase.

estimation methods, BALD and mean-STD showed the best performances, in order. The performance of these two methods at the 60th phase was quite close to that of the full model. The BatchBALD was not superior to the other methods, as it has been known to perform worse under class imbalance [29].

We assessed the overall performance across 9 defect type classes and 10 different initial training sets using Dolan-More curve [38]. For the 90 problems, $\rho(\tau)$ indicates the fraction of the problems that the method's error rate is not greater than τ times the best error rate. The Dolan-More curves of CNN models at the 60th phase for the compared methods are presented in Fig. 5. The full model yielded the highest $\rho(\tau)$ at $\tau = 1$, as the fraction of problems with the lowest error rate was the highest. With increased τ, $\rho(\tau)$ of BALD and mean-STD became greater than that of the other methods including the full model, meaning that BALD and mean-STD evenly performed well across the problems.

The comparison results for each of the defect types are shown in Fig. 6. In active learning for a class imbalance problem, it is important to select instances of the minority class. In the case of *Near-Full* which has the least instances, the performance difference between the uncertainty estimation

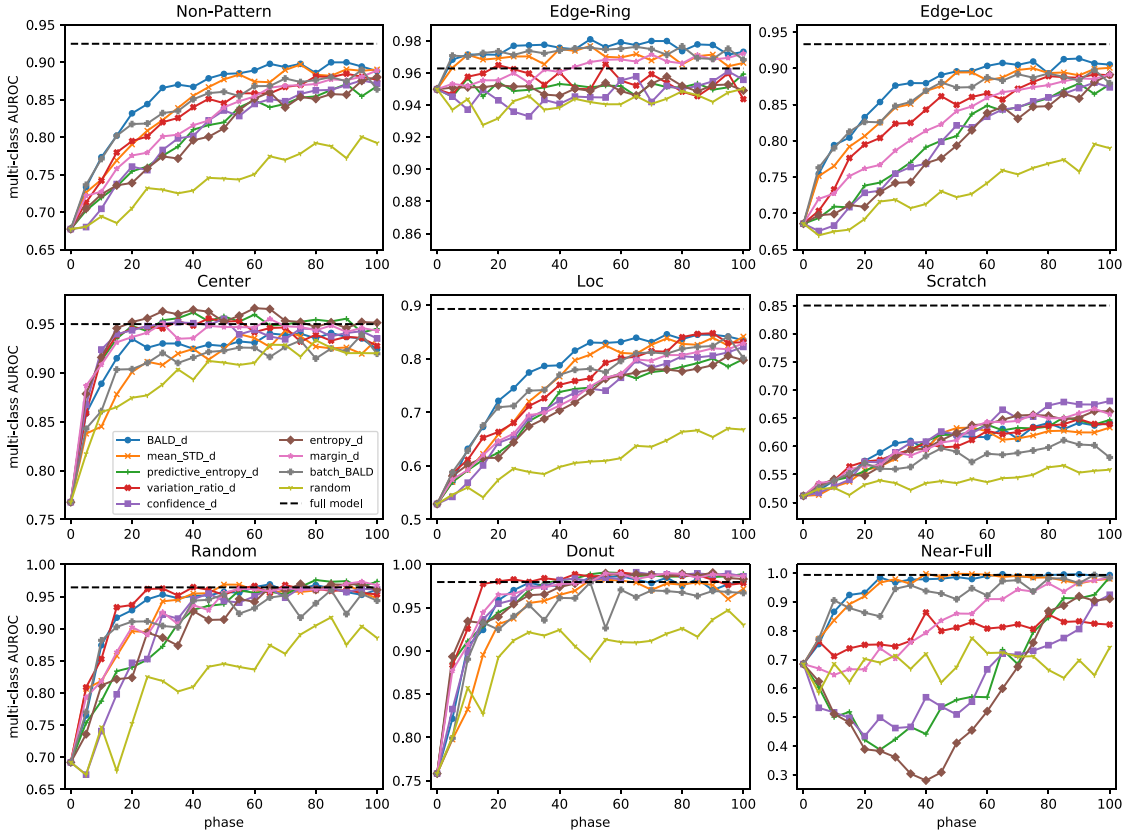

Fig. 6. Comparison of uncertainty estimation methods with diversified top-K selection for each defect type. Each graph has a different y-axis range to clarify the performance difference between methods.

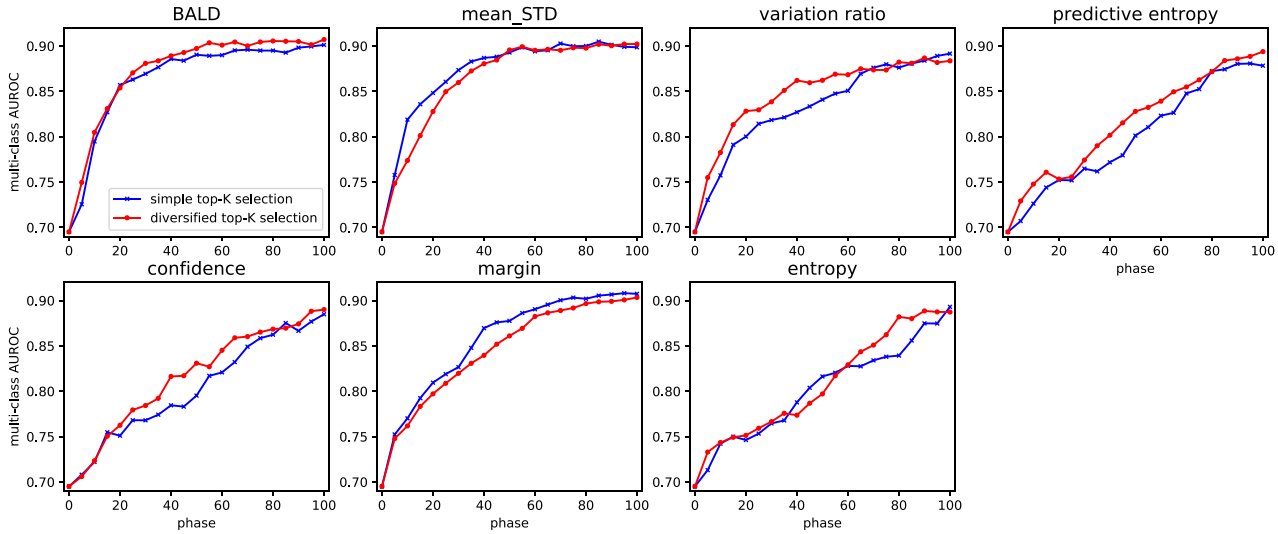

Fig. 7. Comparison between the diversified top-K selection and simple top-K selection.

methods was particularly severe. The performances differed greatly depending on whether or not the instances corresponding to the defect types were included enough in the training set. BALD and mean-STD outperformed other methods. For other minority classes, *Random* and *Donut*, the proposed system showed high performance as well.

For the *Non-Pattern*, *Edge-Loc*, *Loc*, and *Scratch*, the performance of the proposed method was far worse than the full model and increased slowly by active learning. These classes have relatively large variations in their spatial defect patterns. So, they require a large amount of various training instances. Performance differences between the uncertainty estimation methods were also significant for these classes. This suggests that it is more important to select informative training wafer maps. Furthermore, the proposed method even outperformed the full model for some particular classes, which

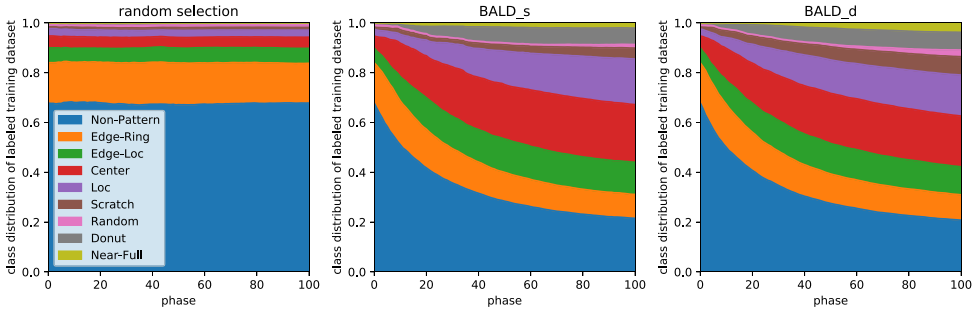

Fig. 8. Class distributions of the labeled training set as the phase progressed.

demonstrates that using a small informative training set can be better than using the whole dataset.

Fig. 7 compares the performance between the diversified top-K selection and simple top-K selection. In most uncertainty estimation methods, the diversified top-K selection yielded performance similar to or better than the simple top-K selection, which indicates that the diversified top-K selection helps to choose more informative wafer maps to improve the classification performance. Above all, the performance of the diversified top-K selection was superior when used with BALD, which showed the best performance.

To investigate how the proposed method works, we visualized in Fig. 8 the change in the class distribution of the labeled training set with the phase progress for the random selection method, BALD_s, and BALD_d. The random selection method showed no change in class distribution, whereas BALD selected minority class instances more so that the proportion of minority classes increased with the progress. This demonstrate that the proposed method selects informative instances without duplication. Additionally, BALD_d exhibited more balanced class distribution than BALD_s. This suggests that diversified top-K selection effectively alleviate the class imbalance problem.

V. Conclusion

Wafer map pattern classification has played an important role in production quality management of a semiconductor manufacturing process. In this paper, we proposed a cost-effective wafer map pattern classification system through the active learning of a CNN. In the system, a CNN model is constructed based on four main steps: uncertainty estimation, query wafer selection, query wafer labeling, and model update. By repetitively performing these steps, a CNN model can be constructed with a higher classification performance and a lower labeling cost. We carried out experimental comparisons between various uncertainty estimation methods for the CNN. The proposed method improved the classification performance compared to the baselines. We found that BALD and mean-STD with diversified top-K selection perform better for the proposed system.

As future work, we would like to develop more intelligent querying strategies and efficient model updating strategies that suit the wafer map pattern classification task to achieve a better trade-off between classification accuracy and labeling costs. In addition, combining the proposed system with a semi-supervised learning scheme would improve the effectiveness through exploiting abundant unlabeled wafer maps during the construction of the system. Finally, we plan to develop a partial automated system based on a reject option to allow ambiguous wafer maps to be manually labeled by human engineers.

References

[1] 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.

[2] 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.

[3] M.-J. 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.

[4] Y.-S. Jeong, S.-J. Kim, and M. K. Jeong, "Automatic identification of defect patterns in semiconductor wafer maps using spatial correlogram and dynamic time warping," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 4, pp. 625–637, Nov. 2008.

[5] J. Yu and X. Lu, "Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 1, pp. 33–43, Feb. 2016.

[6] 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.

[7] M. Fan, Q. Wang, and B. van der Waal, "Wafer defect patterns recognition based on OPTICS and multi-label classification," in *Proc. IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf.*, 2016, pp. 912–915.

[8] 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.

[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.

[10] E. Kim, S. Cho, B. Lee, and M. Cho, "Fault detection and diagnosis using self-attentive convolutional neural networks for variable-length sensor data in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 3, pp. 302–309, Aug. 2019.

[11] T. Nakazawa and D. V. Kulkarni, "Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 250–256, May 2019.

[12] J. Wang, Z. Yang, J. Zhang, Q. Zhang, and W.-T. K. Chien, "AdaBalGAN: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 3, pp. 310–319, Aug. 2019.

[13] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proc. Int. Conf. Mach. Learn.*, 1994, pp. 148–156.

[14] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, 2008, pp. 1070–1079.
[15] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden Markov models for information extraction," in *Proc. Int. Symp. Intell. Data Anal.*, 2001, pp. 309–318.
[16] R. Hwa, "Sample selection for statistical parsing," *Comput. Linguist.*, vol. 30, no. 3, pp. 253–276, 2004.
[17] K. Brinker, "Incorporating diversity in active learning with support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 59–66.
[18] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *Int. J. Comput. Vis.*, vol. 113, no. 2, pp. 113–127, 2015.
[19] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang, "Representative sampling for text classification using support vector machines," in *Proc. Eur. Conf. Inf. Retrieval*, 2003, pp. 393–407.
[20] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 417–424.
[21] J. Azimi, A. Fern, X. Z. Fern, G. Borradaile, and B. Heeringa, "Batch active learning via coordinated matching," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 307–314.
[22] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proc. Annu. Workshop Comput. Learn. Theory*, 1992, pp. 287–294.
[23] N. Roy and A. McCallum, "Toward optimal active learning through Monte Carlo estimation of error reduction," in *Proc. Int. Conf. Mach. Learn.*, 2001, pp. 441–448.
[24] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin–Madison, Madison, WI, USA, Rep. 1648, 2009.
[25] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian active learning with image data," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1183–1192.
[26] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2591–2600, Dec. 2017.
[27] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
[28] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *Proc. Int. Conf. Learn. Represent.*, 2018.
[29] A. Kirsch, J. van Amersfoort, and Y. Gal, "BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning," 2019. [Online]. Available: arXiv:1906.08158.
[30] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian active learning for classification and preference learning," 2011. [Online]. Available: arXiv:1112.5745.
[31] L. C. Freeman, *Elementary Applied Statistics: For Students in Behavioral Science*. Hoboken, NJ, USA: Wiley, 1965.
[32] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 1–9.
[33] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[37] F. Provost and P. Domingos, "Tree induction for probability-based ranking," *Mach. Learn.*, vol. 52, no. 3, pp. 199–215, 2003.
[38] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Math. Program.*, vol. 91, no. 2, pp. 201–213, 2002.