IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.Doi Number

A New Backbone Network for Instance Segmentation: Application on a Semiconductor Process Inspection

JUNGHEE HAN[1], SEONGSOO HONG[2]

[1]Graduate School of Convergence Science and Technology (GSCST), Seoul National University, Seoul, Korea
[2]Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea

Corresponding author: Junghee Han (e-mail: jhhanneo@snu.ac.kr)

ABSTRACT In this paper, we propose Instance Segmentation Detector (ISD) to extract the enhanced feature-maps under the situations where training dataset is limited in the specific industry domain such as semiconductor photo lithography inspection. ISD is used as a new backbone network of state-of-the-art Mask R-CNN framework for instance segmentation. ISD consists of four dense blocks and four transition layers. Each dense block in ISD has the shortcut connection and the concatenation of the feature-maps produced in layer with dynamic growth rate. ISD is trained from scratch without using recently approached transfer learning method. Additionally, ISD is trained with image dataset pre-processed by means of the specific designed image filter to extract the better enhanced feature map of Convolutional Neural Network (CNN). In ISD, one of the key principles is the compactness, plays a critical role for addressing real time problem and for application on resource bounded devices. To validate the model, this paper uses the real image collected from the computer vision system embedded in the currently operating semiconductor manufacturing equipment. ISD achieves consistently better results than state-of-the-art methods at the standard mean average precision. Specifically, our ISD outperforms baseline method DenseNet, while requiring only 1/4 parameters. We also observe that ISD can achieve comparable better results than ResNet, with only much smaller 1/268 parameters, using no extra data or pre-trained models.

INDEX TERMS Semiconductor process inspection, backbone network, instance segmentation, deep learning, convolutional neural networks, computer vision.

I. INTRODUCTION

The semiconductor photo lithography is a process of drawing semiconductor circuits on wafers, coating them thinly with photosensitive polymer materials that respond to light on wafers, then placing a mask on top of the desired pattern and pecking the light to form the desired pattern. In this process, the spin coating is used to spread the required thickness of the photoresist uniformly on the wafer. Therefore, the spin coating is an important process. If inspection faults occur in this process, a defective product is produced no matter how well the subsequent process is performed. It is greatly affecting the defect rate in wafer-based process. As illustrated in Fig. 1, the computer vision system is used to prevent defects in semiconductor products by monitoring these processes and predicting defects in the photo process in advance. Generally, the computer vision system uses the digital image processing [1]–[10] to try and perform emulation of vision at human scale. The computer vision system used in the process of spin coating also finds defects through digital image processing algorithm.

FIGURE 1. The computer vision system embedded in the currently operating semiconductor manufacturing equipment for photo lithography inspection.

However, many detection errors occur due to external environmental factors such as various types of wafers and photoresist, motor rotation speed, and diffuse reflection of light. Fig. 2 illustrates an example of image distorted by external environment factors. Digital image processing algorithm has high performance in case of images with little influence on the external environment. However, performance is extremely degraded when image distortion occurs due to the external environment. Therefore, in the computer vision system, if the characteristics of the image is changed or distorted, there is a disadvantage in that a new or modified technique of digital image processing algorithm and the specialized signal processing method should be applied to overcome it. To overcome the influence of various image distortion, we adopt deep learning that is robust even in the external environment.

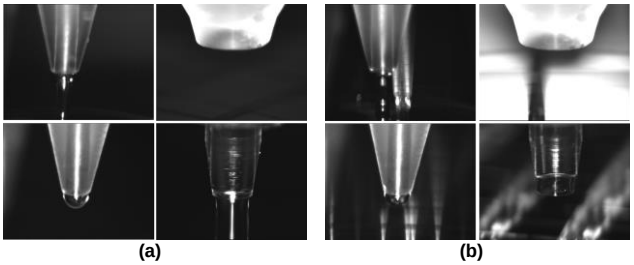

FIGURE 2. An example of image distorted by external environment factors: (a) Normal image; (b) Distorted image.

As illustrated in Fig. 3, there are three inspection type for detecting defects in the spin coating process of semiconductor photo lithography: first is the suck-back state of the nozzle that sprays the photoresist, second is the contamination state of the nozzle, and third is the time to spray the photoresist. In this paper, we propose a method for detecting defects by monitoring the first inspection type, the suck-back state of nozzle. Therefore, in order to this, it is necessary to find a specific area in an image and extract features within the area to determine whether the defect is defective. Deep learning techniques [11] that can detect specific areas in an image have object detection, semantic segmentation, and instance segmentation. Among them, the instance segmentation technique can be applied to inspect not only the suck-back state of nozzle but also the contamination of the nozzle.

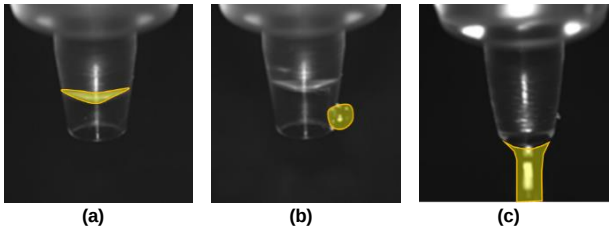

FIGURE 3. Three inspection type for detecting defects in the spin coating process of semiconductor photo lithography: (a) Suck-back state; (b) Contamination state; (c) Dispense state.

Image segmentation is a computer vision process designed to simplify image analysis by splitting input into segments that represent objects or parts of objects and form a collection of pixels. Instance segmentation is a subtype of image segmentation which identifies each instance of each object within the image at the pixel level. Instance segmentation can also be thought as object detection where the output is a mask instead of just a bounding box. Agarwal et al. [12] presented recent advances in object detection in the age of deep convolutional neural networks. The objective of instance segmentation is to detect specific objects in an image and create a mask around the object of interest.

In computer vision, transfer learning is usually expressed through the use of pre-trained models. To achieve desired performance, the common practice in advanced instance segmentation systems is to fine-tune models pre-trained on ImageNet [13]. This fine-tuning process can be viewed as transfer learning [14]–[19]. Researchers usually train CNN models on large scale classification datasets like ImageNet [13] first, then fine-tune the models on target tasks, such as object detection [20]–[35], image segmentation [36]–[39], etc. However, we directly train model without involving any other additional data or extra fine-tuning process. There are numerous state-of-the-art pre-trained CNN models available. Fine-tuning on pre-trained models can quickly convergence to a final state and requires less instance-level annotated training data than basic classification task. As is well-known, fine-tuning can mitigate the gap between different target category distributions. However, it is still a severe problem when the source domain (e.g., ImageNet) has a huge mismatch to the target domain such as industrial images, medical images, etc. As illustrated in Fig. 3, the image used for inspection is completely different from the image on source domain (e.g., ImageNet). Without having enough number of dataset, deep artificial neural networks cannot be trained well and it is difficult to collect enough data size in the specific industry domain.

In this work, we investigate three questions. First, is it possible to train instance segmentation networks from scratch directly with only smaller dataset without the pre-trained models? Second, are there any principles to design a resource efficient network structure for instance segmentation, meanwhile keeping high detection accuracy? Third, is there any methodology to improve inspection performance other than network design? To meet this goal, we propose instance segmentation detector (ISD) and pre-processing that is performed by using image filter before training.

II. RELATED WORK

A. INSPECTION METHOD

Computer vision systems [40]–[46] are widely used for on-line inspection and quality control to improve the finished product quality and lower the costs in various industries. The computer vision system used in current semiconductor

industries performs the specialized digital image processing and signal processing to extract features necessary for defect detection, and determines the defect by means of a neural network as a classifier. The specialized digital image processing removes noise from the input image of specific domain, improves brightness or contrast, emphasizes edges, and makes the image more clearly to extract features. Feature extraction is obtained by the signal processing method that calculates the sum of the vertical component pixels and the horizontal components of the pre-processed image by means of digital image processing, and applies an adaptive threshold. Recognizing the extracted features and determining whether there are defects is composed of a neural network. Fig. 4 (c) illustrates an example of automatically detecting the contamination state of nozzle by means of digital image processing. Fig. 5 also illustrates an example of automatically detecting the suck-back state of nozzle by means of signal processing during the spin coating process of semiconductor photo lithography.

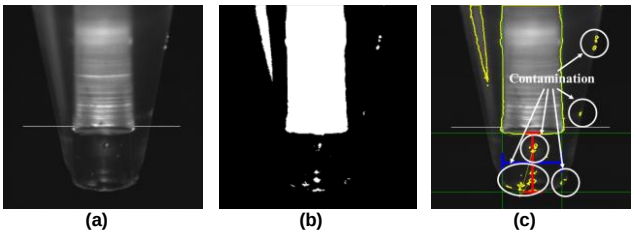

FIGURE 4. An example of detecting the contamination state of nozzle by means of the specialized digital image processing: (a) Original image; (b) Pre-processed Image; (c) Image where contamination is detected.

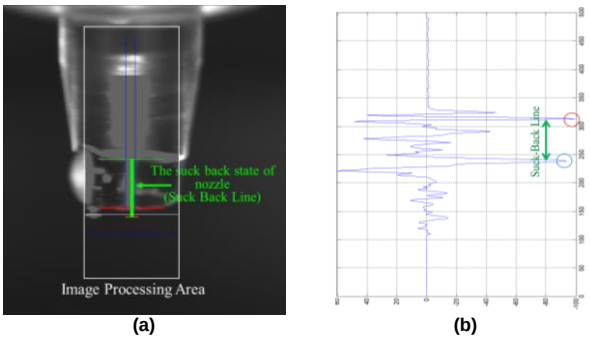

FIGURE 5. An example of detecting the suck-back state of nozzle by means of the specialized signal processing: (a) The suck-back line is detected by means of filtering image within processing area; (b) The suck-back line is detected by means of signal processing which is adopting adaptive threshold and sum of pixels in x direction.

In the spin coating process of semiconductor photo lithography, various types of nozzle for spraying photoresist are used depending on the kind of photoresist and the characteristic of wafer. Fig. 6 illustrates an example of various types of nozzle. Therefore, digital image processing and signal processing method used in the computer vision system should be applied to the specialized technique depending on external environment such as various types of nozzle, wafer characteristics and diffuse reflection of light etc. If a new nozzle or a new wafer is used, the defect detection accuracy of the computer vision system is inevitably reduced.

Considering these problems, we propose instance segmentation method based on generalized deep learning in order to be more robust to the external environment and further improve performance instead of the specialized digital image processing and signal processing method used for semiconductor photo lithography inspection.

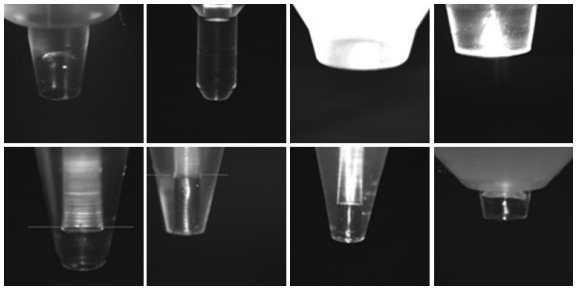

FIGURE 6. An example of various types of nozzle for spraying photoresist.

B. ENHANCED FEATURE MAP

The discriminative feature is very important factor in image classification problem, and the smaller the variance within the same class and the larger the variance between different classes, the easier it is to solve the classification problem in general. The feature-map of CNN to detect nozzle type is clearly distinguished between the nozzle types. However, since the inspection in semiconductor photo lithography is performed in the same nozzle type, it is difficult to extract the discriminative CNN feature-map. It is hard to extract the discriminative feature from the proposed regions of the Region Proposal Network (RPN) using CNN feature-map of the same nozzle type. As illustrates in Fig. 7, the mask area cannot be achieved without the discriminative CNN feature-map in the proposed regions.

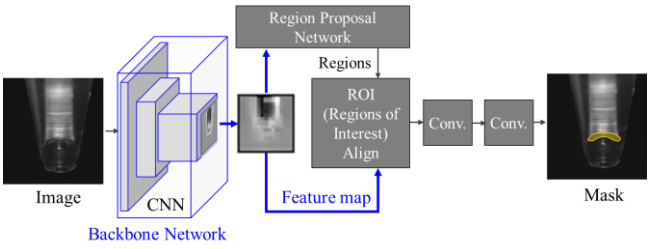

FIGURE 7. An example of instance segmentation process.

The reason for not being able to extract the discriminative feature in the proposed regions is that it is not enough to extract the discriminative feature by means of only original pixel information in the corresponding area as a gray scale image. In order to enhance a feature-map of CNN with only the original pixel information of the image, it may be possible to extract the discriminative feature by performing a lot of deep learning by increasing the network layer of CNN with a

large number of various training images. Deep convolutional neural networks require a large corpus of training data in order to avoid over-fitting. Over-fitting refers to the phenomenon when a network learns a function with very high variance such as to perfectly model the training data. Unfortunately, many application domains do not have access to big data, such as industrial image analysis and medical image analysis, and collection of such training data is often expensive and laborious.

Data augmentation overcomes this issue by artificially inflating the training set with label preserving transformations. Recently there has been extensive use of generic data augmentation to improve CNN task performance. Data augmentation encompasses a suite of techniques that enhance the size and quality of training datasets such that better deep learning models can be built using them. Data augmentations based on basic image manipulations are geometric transformation, flipping, color space, cropping, rotation, translation, noise injection, color space transformations, geometric versus photometric transformations, kernel filters, mixing images, random erasing, feature space augmentation, adversarial training, generative adversarial networks, neural style transfer, and meta-learning [47]–[52].

However, we propose the pre-processing method that reduces the amount of training images and decreases the number of network layer in CNN rather than data augmentation. The specialized image filter for the semiconductor photo lithography inspection is applied to the pre-processing method in order to enhance the feature-map of CNN.

C. BACKBONE NETWORK FOR INSTANCE SEGMENTATION

A lot of deep convolutional neural networks (CNN) [53] originally designed for classification tasks have been adopted for the detection task as well. And a lot of modifications have been done on them to adapt for the additional difficulties encountered. Object detection is a natural extension of the classification problem. The constant challenge is to correctly detect the presence and accurately locate the object instance in the image. It is a supervised learning problem in which, given a set of training images, one has to design an algorithm which can accurately locate and correctly classify as many object instances as possible in a rectangle box while avoiding false detections of background or multiple detections of the same instance. The process of detecting instance segmentation can be spilt into three parts: extracting feature-maps, proposing regions, classifying and regressing binary mask. Among them, the backbone network that extracts feature-maps play a major role in instance segmentation detection models. Huang et al. [54] partially confirmed the common observation that, as the classification performance of the backbone increases on ImageNet [13] classification task, so does the performance of object detectors based on those backbones. It is the case at least for popular object detectors like Fast R-CNN [21], Faster R-CNN [22], Mask R-CNN [55] and R-FCN [23] although for SSD [24] the object detection performance remains around the same. Since there are significant efforts that have been devoted to design network architectures for image classification, many diverse and powerful networks are emerged, such as VGGNet [56], GoogLeNet [57] , ResNet [58], DenseNet [59], DPN [60] etc. In practice, most of the detection methods [20], [21], [22], [24], [55] directly utilize these structures pre-trained on ImageNet [13] as the backbone network for detection task. Some other works try to design specific backbone network structures for object detection, but still require to pre-train on ImageNet [13] classification dataset in advance. Kim et al. [61] proposes PVANet for fast object detection, which consists of the simplified "Inception" block from GoogLeNet [57]. Huang et al. [54] investigated various combination of network structures and detection frameworks, and found that Faster R-CNN [22] with Inception-ResNet-v2 [62] achieved very promising accurate performance. Nakazawa et al. [63] proposed the CNN architecture for wafer map pattern generation in the semiconductor manufacturing.

Therefore, we propose a suitable backbone structure for extracting the enhanced feature-map to detect instance segmentation in industrial domain, which is the proposed ISD instead of ResNet [58] that is the backbone network of state-of-the-art Mask R-CNN framework.

D. LEARNING NETWORK MODEL FROM SCRATCH

There are no previous works that train deep CNN-based instance segmentation in industrial domain from scratch. In generic object detection, Shen et al. [64] proposed Deeply Supervised Object Detectors (DSOD), an object detection framework that can be trained from scratch. In semantic segmentation, J´egou et al. [65] demonstrated that a well-designed network structure can outperform state-of-the-art solutions without using the pre-trained models. It extends DenseNet [59] to fully convolutional networks by adding an up sampling path to recover the full input resolution.

Thus, our proposed approach has very appealing advantage in that it is learning network model from scratch without using the pre-trained model on ImageNet [13] for instance segmentation.

III. OUR APPROACH

We first introduce the whole framework of our ISD architecture, following by pre-processing for extracting the enhanced feature-map. Then we describe the training process and objective in detail.

A. ISD ARCHITECTURE

The whole framework for semiconductor photo lithography inspection is based on Mask R-CNN framework. There are two stages of Mask R-CNN framework. First, it generates proposals about the regions where there might be an object based on the input image. Second, it predicts the class of the object, refines the bonding box and generates a mask in pixel

level of the object based on the first stage proposal. Both stages are connected to the backbone network structure.

Many approaches to instance segmentation are based on segment proposals. However, our approach is focus on the backbone network which extracts the enhanced feature-maps for the object mask. The state-of-the-art Mask R-CNN framework uses ResNet [58] and ResNetXt [66] as backbone network. However, as illustrates in Fig. 8, our approach uses the compact ISD instead of ResNet [58] for addressing real time problem and learning from scratch.

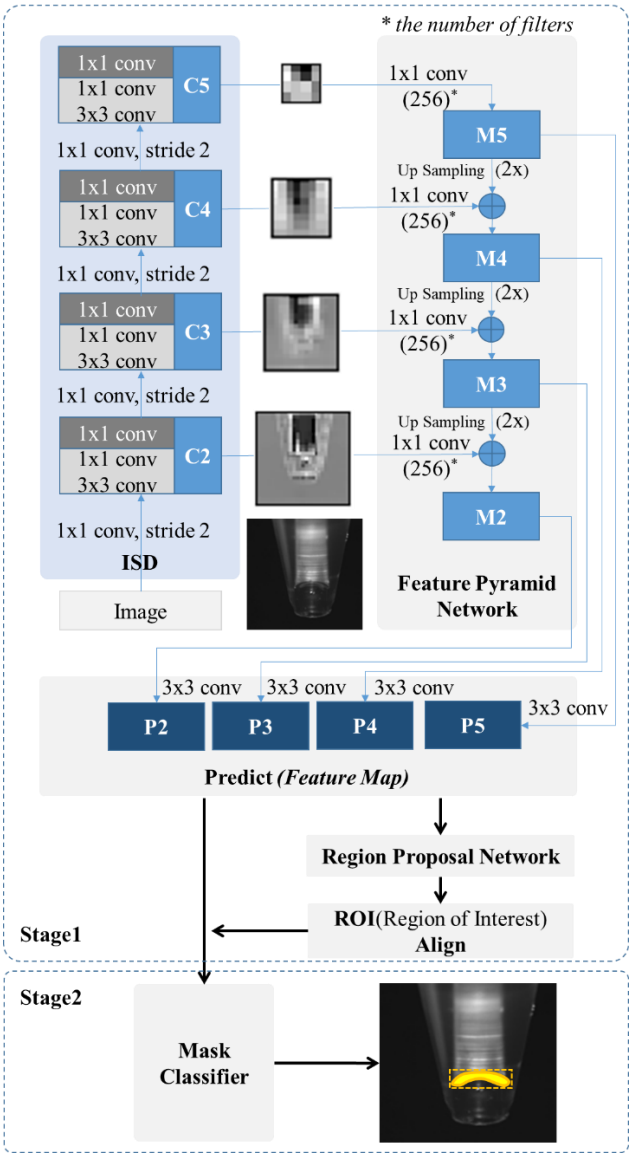

FIGURE 8. The network structure by using ISD for instance segmentation on Mask R-CNN framework.

ISD based on the state-of-the-art DenseNet [59] is motivated by combining the advantage of shortcut connection and concatenation of the feature-maps produced in layers with dynamic growth rate. In order to improve the performance of instance segmentation with better parameter efficiency, we investigated all the state-of-the-art CNN based instance segmentation. The design principle of ISD is compact model, which is suitable for real time embedded system such as computer vision system and make them easy to train under reducing over fitting on tasks with smaller training set sizes.

ISD comprises layers, each of which implements a composite function of operations such as Batch Normalization (BN) [67], rectified linear units (ReLU) [68], Pooling [69], or Convolution (Conv). ISD has the concatenation of the feature-maps produced in layers in order to encourage strengthen feature propagation and feature reuse. Further, ISD has the shortcut connection for addressing vanishing and exploding gradients. ISD is composed of four dense blocks and four transition layers similar to DenseNet [59]; see Table 1.

TABLE 1. ISD architecture.

Layers	Output Size (input 3×120×120)	ISD-38
Convolution	12×60×60	3×3 conv, stride 2
Dense Block (1)	24 ×30×30 24×30×30	1×1 conv, stride 2 $\begin{bmatrix}1\times1\text{ conv}\\3\times3\text{ conv}\end{bmatrix}\times 2$
Transition Layer (1)	24×30×30	1×1 conv, stride 1
Dense Block (2)	48×15×15 48×15×15	1×1 conv, stride 2 $\begin{bmatrix}1\times1\text{ conv}\\3\times3\text{ conv}\end{bmatrix}\times 4$
Transition Layer (2)	48×15×15	1×1 conv, stride 1
Dense Block (3)	72×8×8 72×8×8	1×1 conv, stride 2 $\begin{bmatrix}1\times1\text{ conv}\\3\times3\text{ conv}\end{bmatrix}\times 4$
Transition Layer (3)	72×8×8	1×1 conv, stride 1
Dense Block (4)	96×4×4 96×4×4	1×1 conv, stride 2 $\begin{bmatrix}1\times1\text{ conv}\\3\times3\text{ conv}\end{bmatrix}\times 4$
Transition Layer (4)	96×4×4	1×1 conv, stride 1
Prediction	-	Pooling/Dense

However, crucially in contrast to DenseNet [59], ISD combine features through summation before they are passed into a dense block combined features by concatenating them with post-activation. Fig. 9 illustrates this layout schematically. Santhanam et al. [70] presented the result that pre-activation ResNets consistently outperforms the original post-activation only at very high-network depths (≥ 152 depths). ISD has 38 depths at low-network depths and post-activation ISD outperformed pre-activation on the results of experiment. Thus, in our approach, ISD has a structure with post-activation as shown in Fig. 9. Moreover, as illustrated in Fig. 10, there is dynamic growth rate unlike DenseNet [59], which applies different growth rate in each layer in order to optimize the model. The growth rate that regulates the amount of information on each layer determine the number of feature-

map. The dynamic growth rate substantially reduces the number of parameters, optimizing the model more compact and improving the performace.

ISD has mainly three hyper-parameters: First, we refer to *n* as number of layers in each dense bock. Second, we refer to *k* as growth rate of the network. Third, we refer to *bw* as bottleneck width. We optimized the hyper-parameters through experimental results.

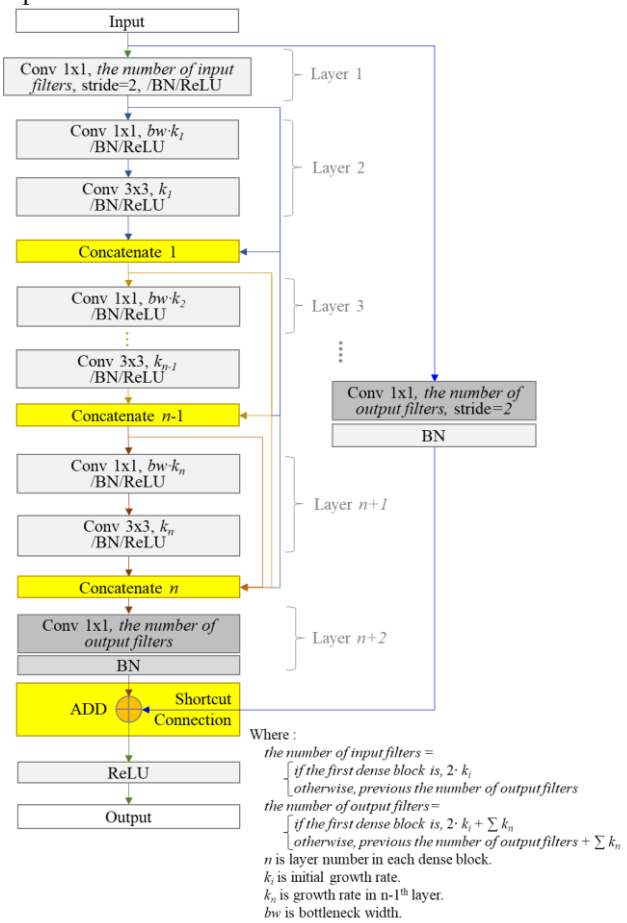

FIGURE 9. Dense block network model with post-activation in ISD.

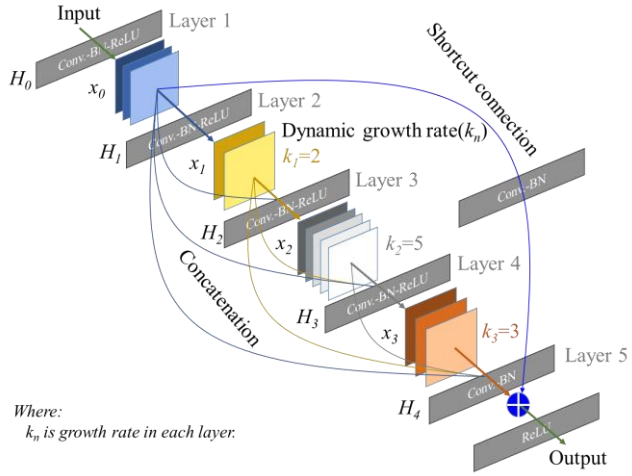

FIGURE 10. A dense block with dynamic growth rate of k = 2, 5, 3 in each layer on ISD.

B. PRE-PROCESSING FOR ENHANCED FEATURE MAP

Edge detection is one of the significant section of the image processing algorithms which have many applications like image morphing, pattern recognition, image segmentation and image extraction etc. As the edge is one of the major information contributors to any image, hence the edge detection is a very important step in many of the image processing algorithms. It represents the contour of the image which could be helpful to recognize the image as an object with its detected edges. Kabade et al. [71] proposed block level canny edge detection algorithm which is the special algorithm to carry out the edge detection of an image in order to reduce the time and memory consumption. In case of the suck-back state among the inspection types shown in Fig. 3, it is hard to extract the feature from an image overlapped by nozzle image and photoresist image. In addition, the image of photoresist is varied by depending on the type of nozzle, and the image of nozzle is varied by depending on the kind of photoresist. The specific image filter modified by the sobel edge detector [72], which is composed of a pair of 3×3 convolution masks, one estimating gradient in the horizontal x-direction and the other estimating gradient in vertical y-direction, is adopt to identify points in an image at which the image brightness changes sharply or, more formally, has discontinuities. Pre-processing is performed by using convolution on the image by means of the specific image filter.

The edge occurs where there is a discontinuity in the intensity function or a very steep intensity gradient in the image. Thus, the edge could be located at which the derivative is maximum. The gradient is a vector, whose components measure how rapid pixel value are changing with distance in the x and y direction. Thus, the components of the gradient may be found using the following approximation:

$$\frac{\partial f(x,y)}{\partial x} = \Delta x = \frac{f(x+dx,y) - f(x,y)}{dx} \tag{1}$$

$$\frac{\partial f(x,y)}{\partial y} = \Delta y = \frac{f(x,y+dy) - f(x,y)}{dy} \tag{2}$$

Where dx and dy measure distance along the x and y directions respectively. In discrete images, one can consider dx and dy in terms of numbers of pixel between two points, dx = dy = 1

$$\Delta x = f(x+1,y) - f(x,y) \tag{3}$$

$$\Delta y = f(x,y+1) - f(x,y) \tag{4}$$

The different operation in "(3)" and "(4)" correspond to convolving the image with the following image filter mask.

$$\Delta x = \begin{bmatrix} -1 & 0 & 1 \\ g & 0 & g \\ -1 & 0 & 1 \end{bmatrix} \tag{5}$$

$$\Delta y = \begin{bmatrix} -1 & g & 1 \\ 0 & 0 & 0 \\ -1 & g & 1 \end{bmatrix} \tag{6}$$

In "(5)" and "(6)", g is adaptively applied according to the image intensity. The image pre-processed by means of the specific image filter is shown in Fig. 11. The pre-processed image that is used as the input of ISD has significance in extracting the enhanced feature-map for inspection

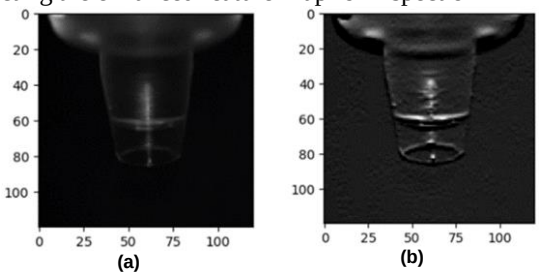

FIGURE 11. The image pre-processed by means of the specific digital image filter: (a) Original image; (b) The image processed by means of the filter of equation "(6)" (g = 4).

C. MODEL TRAINING

In our approach, we focus on the instance segmentation task without using the pre-trained models. We train models on target dataset directly without using IamgeNet dataset as shown in Fig. 12. ISD is trained with various nozzle image as shown in Fig. 12, to classify the nozzle type.

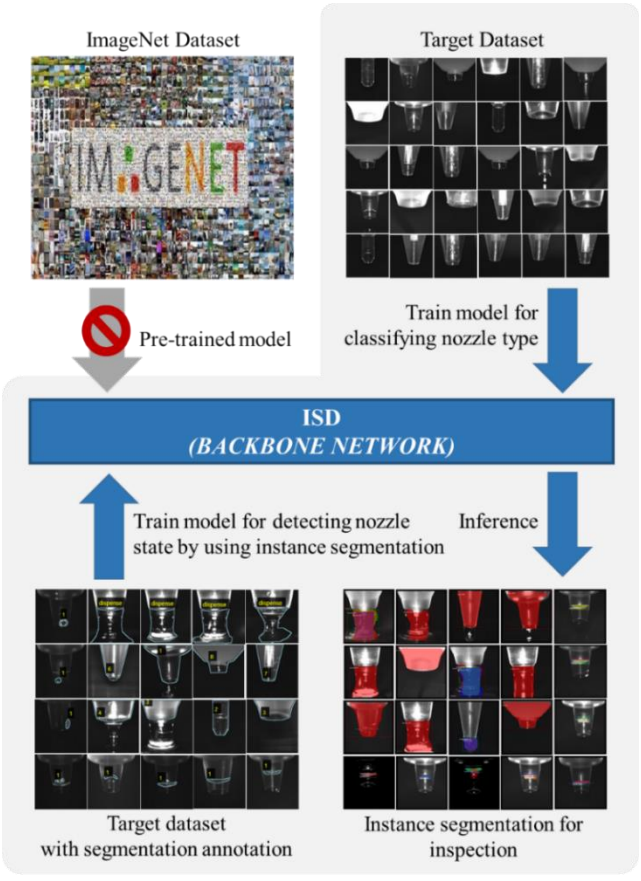

FIGURE 12. Illustration of training model on target dataset directly.

The image dataset used to train Mask R-CNN is prepared by using image annotation tool (i.e. VGG image annotator) which manipulates the labeled segmentation of image. In addition, filtering the input dataset is performed for pre-processing of training model. Fig. 13 illustrates the training process.

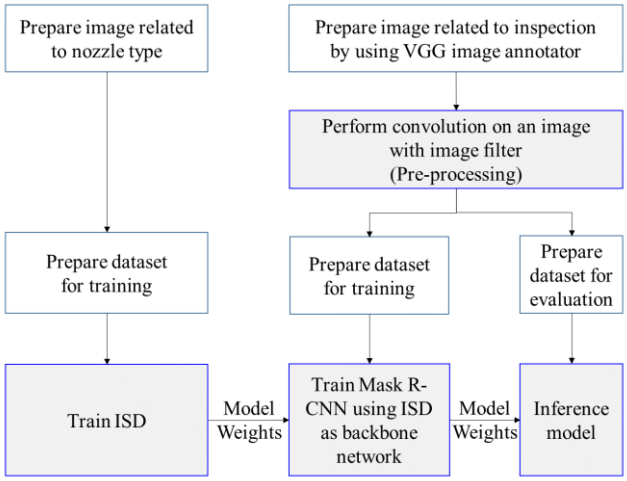

FIGURE 13. Training process of instance segmentation using ISD as the backbone network of Mask R-CNN framework.

D. TRAINING OBJECTIVE

The training objective is the losses being used to converge the huge number of weights and the hyper-parameters that must be conducive to this convergence.

In training model for classifying nozzle type, categorical cross entropy loss generally used to classify image is adopt to the loss of ISD (i.e. L_{ISD}). It is a softmax activation plus a cross entropy loss.

$$L_{ISD} = -\log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$$

Where:

s_p= the CNN score for the positive class

C = the number of classes

s_j = the score inferred by the network for each class in C

In training model for detecting suck-back state of nozzle, the training loss is adopt from Faster R-CNN and Mask R-CNN, which is a weighted sum of the classification loss(cls), the localization loss(box) and segmentation mask loss(mask). Where L_{total_cls} and L_{total_box} are same as in Faster R-CNN [22] and L_{total_mask} is same as in Mask R-CNN [55]

$$L_{total} = L_{total\_cls} + L_{total\_box} + L_{total\_mask}$$

$$L_{total\_cls} = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*)$$

$$L_{cls}(p_i, p_i^*) = -p_i^* \log p_i - (1-p_i^*) \log (1-p_i)$$

Where:

p_i = Predicted probability of anchor i being an object

p_i^* = Ground truth label of whether anchor i is an object

N_{cls} = Normalization term, set to be batch size

$$L_{total\_box}=\frac{\alpha}{N_{box}}\sum_i p_i^* \cdot L_1^{smooth}(t_i-t_i^*)$$

Where:

t_i = Predicted four parameterized coordinates

t_i^* = Ground truth coordinates

N_{box} = Normalization term, set to the number of anchor locations

α = Balancing parameter

$$L_{total\_mask}=-\frac{1}{m^2}\sum_{1\le i,j\le m}\left[y_{ij}\log\hat{y}_{ij}^k+(1-y_{ij})\log(1-\hat{y}_{ij}^k)\right]$$

Where:

y_{ij} = Label of cell(i,j) in the true mask for the region of size m×m

$\hat{y}_{ij}^k$ = Predicted value of the same cell for the ground truth class k

IV. EXPERIMENT

We implement ISD based on the tensorflow platform [73]. The hardware platform is notebook with two GPUs as illustrated in Table 2. Since image related to semiconductor process is not available in open datasets for deep learning such as ImageNet, MS COCO, pascal VOC etc., the experimental dataset is acquired from computer vision system embedded in the currently operating semiconductor manufacturing equipment for photo lithography inspection. The size of image is 640×495 pixels and gray color. Intuitively, larger input images will bring better performance for instance segmentation. However, an additional difficulty is that real world applications like computer vision system demand inspection to be solved in real time. Fastest detectors are usually better than the best performing ones. Thus, we reduced the size of image used as the input of ISD to 120×120 pixels. We evaluate ISD with different depth and growth rates for compactness. We verify the effectiveness of the method through the comparison experiment. A consistent setting is imposed on all the experiments, unless when some components or structures are examined. We adopt the standard mean Average Precision (mAP) to measure the instance segmentation performance.

TABLE 2. Hardware specification.

Item	Specification
CPU	Intel Core i7-8750H 2.2GHz
Memory	16GB, 3200MHz DDR4
GPU0	Intel UHD Graphics 630
GPU1	NVIDIA GeForce GTX 1050 Ti

A. CLASSIFICATION RESULTS ON ISD

In order to classify nozzle type, 18,304 images that have already been correctly classified into 8 types of nozzle, were collected from real operating semiconductor manufacturing equipment. Then, we split these images randomly into 13,728 training datasets and 4,576 validation datasets. The classification training accuracy after only 10 epoch is 99.8% and the validation accuracy is 99.9% for classifying nozzle type. The classification training and validation accuracy in each epoch is illustrated in Fig. 14. The average processing time for each epoch is 37 seconds. In addition to classification of nozzle type, we also test to detect instance segmentation of nozzle type. We used 385 training dataset and 138 validation dataset for instance segmentation. Fig. 15 illustrates the result on detecting instance segmentation in each nozzle type.

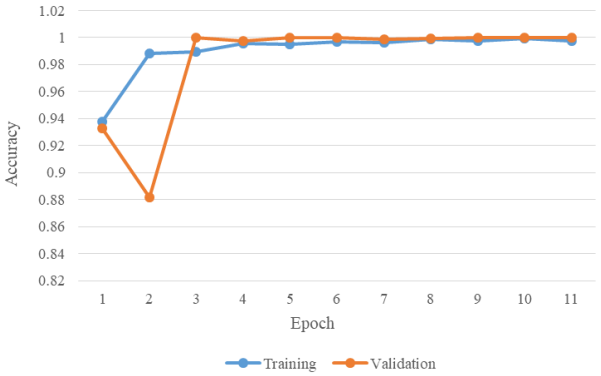

FIGURE 14. The classification training and validation accuracy in each epoch. ISD has 38 depths with shortcut connection. The uniform growth rate (k) is 6.

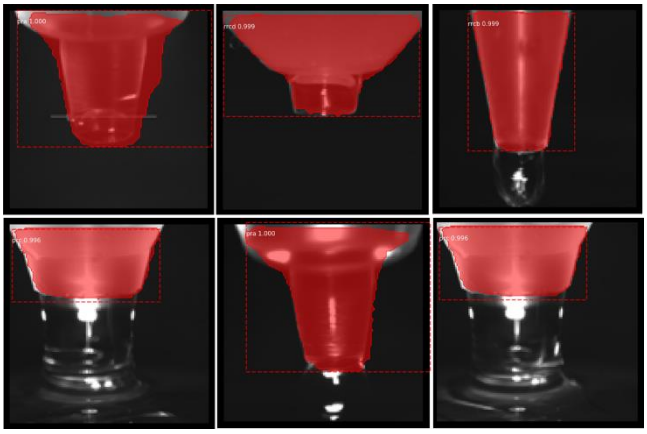

FIGURE 15. The experimental result of detecting instance segmentation of nozzle type.

B. COMPARISON WITH PRE-PROCESSING

In order to detect suck-back state of nozzle, we used 266 training datasets and 144 validation datasets for instance segmentation in each nozzle type. The average processing time for each epoch is 208 seconds. We evaluate the performance of pre-processing on instance segmentation task in the standard mean average precision.

In aspect of the mask, the mask of nozzle type was detected well even without pre-processing using image filter. However,

the mask of suck-back state for inspection was not detected or incorrectly recognized when the pre-processing is not performed. Fig. 16 illustrates comparison with pre-processing in aspect of mask. We can observe that the pre-processing using image filter can achieve higher accuracy, which is consistent to our conjecture that the enhanced feature-map is extracted by pre-processing.

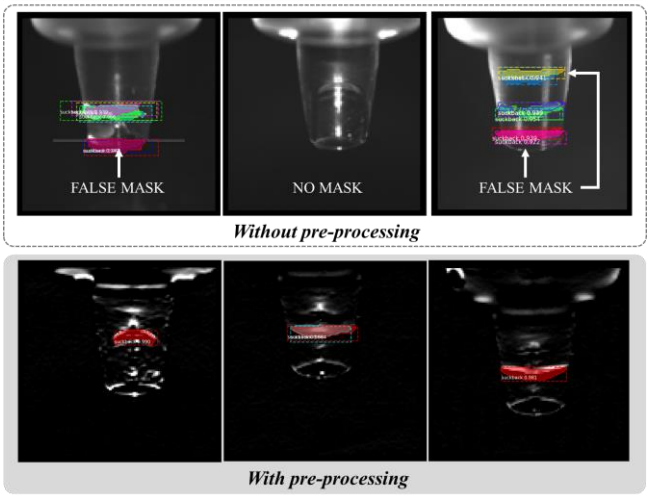

FIGURE 16. Instance segmentation of suck-back state for inspection. In case of training ISD with pre-processing, the mask performance is better than without pre-processing.

In aspect of the standard mean average precision, comparison of pre-processing is illustrated in Table 3. mAP@0.50 in validation is improved by 4.27% when the pre-processing is performed. Interestingly, mAP@0.75 in validation is improved with a large margin (18.19%) when the pre-processing is performed. We can observe that the greatest task performance improvement was yielded by pre-processing.

TABLE 3. Comparison of performing pre-processing. ISD has 38 depths with shortcut connection and the uniform growth rate (k) is 6.

ISD	Test (mAP, %)		Train (mAP, %)	
	IoU[1]:0.50	IoU[1]:0.75	IoU[1]:0.50	IoU[1]:0.75
w/o pre-processing	90.97	38.95	87.97	43.87
w/ pre-processing	95.24	57.14	95.42	68.67

[1] Intersection over Union.

C. INSTANCE SEGMENTATION RESULTS ON ISD

Model optimization and performance are an important trade-off for the applications of deep neural networks in actual instance segmentation tasks for real time application. In order to optimize ISD, we conduct experiments with three cases which are the number of depth, shortcut connection and dynamic growth rate.

The number of depth. We have experimented with various depths on ISD. As illustrated in Table 4, we empirically demonstrate that the deeper layer is the better performance, as is well known. However, using 42 depths is sufficient to deliver good performance and it is better in aspect of resource effectiveness. We can observe that our compactness model with only 85K parameters achieves performance to 95.49% at mAP@0.50 in validation, which shows great potential for applications on computer vision system in real time.

Shortcut connection. We have experimented with and without shortcut connection. We observe that ISD with 62 depths using shortcut connection significantly improves the performance from 42.55% to 57.87% at mAP@0.75 in validation. We experimentally found that shortcut connection improves the performance by means of alleviating vanishing and exploding gradients, encouraging feature reuse.

TABLE 4. Comparison with different depths and shortcut connection. We experiment with model weights having the lowest validation loss obtained during the training up to 100 epochs. The uniform growth rate (k) is 6.

ISD			Test (mAP, %)		Train (mAP, %)	
D[1]	SC[2]	Param[3]	IoU[4]:0.50	IoU[4]:0.75	IoU[4]:0.50	IoU[4]:0.75
24		15,038	84.03	44.10	86.84	49.72
	√	26,666	83.31	48.97	87.48	57.60
38		38,288	94.79	55.00	96.30	59.90
	√	69,776	95.24	57.14	95.42	68.67
42		46,700	88.19	50.64	91.35	59.25
	√	85,580	95.49	59.42	97.74	65.21
54		80,000	89.58	52.78	85.90	53.73
	√	148,832	93.40	50.12	94.17	58.55
62		106,472	83.99	42.55	84.80	47.82
	√	199,376	94.33	57.87	97.37	61.01

[1] Depth, [2] Shortcut, [3] Parameters (bytes), [4] Intersection over Union.

Growth rate. As aforementioned, ISD use dynamic growth rate that applies different growth rates in each layer. In Table 5, we compare three options: (A) uniform growth rates (k,k,⋯,k) are used; (B) increasing growth rates (1,2,3,⋯,k) are used; (C) decreasing growth rates (k,k-1,k-2,⋯,2,1) are used; As illustrated in Table 5, we observe that ISD with 54 depths using increasing growth rates improves the performance from 91.32% to 95.83% at mAP@0.50 in validation, while requiring only 1/2 parameters. We experimentally found that dynamic growth rate improves the performance better than uniform growth rate. It substantially reduces the number of parameters.

TABLE 5. Comparison of dynamic growth rate with shortcut connection. We experiment with model weights having the lowest validation loss obtained during the training up to 100 epochs.

ISD			Test (mAP, %)		Train (mAP, %)	
D[1]	G[2]	Param[3]	IoU[4]:0.50	IoU[4]:0.75	IoU[4]:0.50	IoU[4]:0.75
38	12A	266,336	92.71	63.16	95.30	70.90
	12B	98,142	92.79	63.71	94.59	64.74
	12C	159,953	93.40	57.18	96.05	66.28
42	12A	281,672	95.14	53.10	95.52	61.52
	12B	104,553	91.20	44.12	87.97	45.03
	12C	179,968	95.66	55.85	96.81	65.30
54	12A	514,592	91.32	50.89	90.35	51.45
	12B	251,904	95.83	52.54	96..43	61.63
	12C	386,482	93.06	52.43	94.93	58.87

[1] Depth, [2] Growth rate (k), [3] Parameters (bytes), [4] Intersection over Union.

D. COMPARISON WITH STATE-OF-THE-ART METHODS

We compare our results with state-of-the-art backbone networks of Mask R-CNN framework. Results are summarized in Table 6. ISD achieves consistently better results than stat-of-the-art methods with much more compactness structure. Specifically, our ISD-38 achieves 95.24% at mAP@0.50 in validation, which outperforms the baseline DenseNet-38 with a large margin (16.97%, mAP@50), while requiring only 1/4 parameters. We also observe that ISD-38 can achieve comparable better results at mAP@0.75 than ResNet-38 requiring a huge memory space to store the massive parameters, with only much smaller 1/268 parameters, which shows great potential for application on resource bounded devices.

As the size of the network increases, the inference and the training become slower and require more data. There is generally a trade-off between performance and speed. When one needs real time detectors, like for computer vision, one loses some precision. In Table 6, the highest result of 96.59% at mAP@50 in validation are obtained with ResNet-38. Our ISD-42 achieves 95.49% at mAP@50 in validation, 1.1% lower. However, the speed has improved significantly by 217 times. Interestingly, our ISD-42 is 3.45% higher than ResNet-38 at mAP@75 in validation.

TABLE 6. Comparison with state-of-the-art backbone networks. We experiment with model weights having the lowest validation loss obtained during the training up to 100 epochs. The growth rate of DenseNet is 12. The uniform growth rate of ISD is 6.

BN[1]	Param[2]	Test (mAP, %)		Train (mAP, %)	
		IoU[3]:0.50	IoU[3]:0.75	IoU[3]:0.50	IoU[3]:0.75
ResNet-26	14,008K	94.82	61.37	98.76	75.47
ResNet-38	18,496K	96.59	55.97	96.93	78.57
ResNet-50	23,604K	93.36	50.58	95.49	67.45
ResNet-101	42,674K	94.85	51.58	97.60	65.14
DenseNet-24	98K	78.77	36.04	84.33	48.81
DenseNet-38	253K	78.27	35.20	82.01	42.98
DenseNet-42	308K	85.76	40.74	86.28	42.70
DenseNet-54	516K	84.72	35.71	82.90	40.53
DenseNet-62	681K	85.59	37.57	85.43	46.35
ISD-24	26K	83.31	48.97	87.48	57.60
ISD-38	69K	95.24	57.14	95.42	68.67
ISD-42	85K	95.49	59.42	97.74	65.21
ISD-54	148K	93.40	50.12	94.17	58.55
ISD-62	199K	94.33	57.87	97.37	61.01

[1] Backbone Network, [2] Parameters (kilobyte), [3] Intersection over Union.

V. CONCLUSION

This paper presents a novel backbone network, the ISD, to solve the problem that training dataset limited in specific industry domain might cause overfitting at training and quality mismatch at inference, for addressing real time problem and for application on resource bounded devices. Our model is simple to construct and can be trained directly on full images. According to our method including pre-processing, enhanced feature-maps can be obtained for instance segmentation. We demonstrate that our ISD-42 significantly outperforms state-of-the-art DenseNet-42 in terms of both accuracy (9.73% more accurate) and speed (3 times faster) at mAP@50 in validation. Also, our ISD-42 improves 217 times faster in speed and 3.45% higher accurate than state-of-the-art ResNet-38 at mAP@0.75 in validation.

In addition to backbone network of Mask R-CNN framework, the ISD can be applicable to many instance segmentation architecture. We believe that it can be useful to many future instance segmentation research efforts in diverse industry domain which is requiring real time and good performance with only smaller training dataset.

REFERENCES

[1] M. Tiwari, S. S. Lamba, and B. Gupta, "An image processing and computer vision framework for efficient robotic sketching," Procedia Computer Science, vol. 133, pp. 284-289, 2018.
[2] J. Byung-Wan and L. Yun-Sung, "Computer Vision-Based Bridge Displacement Measurements Using Rotation-Invariant Image Processing Technique," Sustainability, vol. 10, no. 6, p. 1785, 2018.
[3] P. K. Saha, "Tensor scale: A local morphometric parameter with applications to computer vision and image processing," Computer Vision and Image Understanding, vol. 99, no. 3, pp. 384-413, 2005.
[4] N. Gonçalves, V. Carvalho, M. Belsley, R. M. Vasconcelos, F. O. Soares, and J. Machado, "Yarn features extraction using image processing and computer vision – A study with cotton and polyester yarns," Measurement, vol. 68, pp. 1-15, 2015.
[5] C. Ng, C. Wu, W. Ip, C. Chan, and T. Ho, "A real time quality monitoring system for the lighting industry : a practical and rapid approach using computer vision and image processing (CVIP) tools," International journal of engineering business management, vol. 3, no. 4, pp. 14-21, 2011
[6] D. L. B. R. Jurjo, C. Magluta, N. Roitman, and P. Batista Gonçalves, "Analysis of the structural behavior of a membrane using digital image processing," Mechanical Systems and Signal Processing, vol. 54-55, pp. 394-404, 2015.
[7] S. Sunoj et al., "Sunflower floral dimension measurements using digital image processing," Comput. Electron. Agric., vol. 151, pp. 403-415, 2018.
[8] H.-S. Choi, J.-H. Cheung, S.-H. Kim, and J.-H. Ahn, "Structural dynamic displacement vision system using digital image processing," NDT and E International, vol. 44, no. 7, pp. 597-608, 2011.
[9] A. Iswardani and W. Hidayat, "Mammographic Image Enhancement using Digital Image Processing Technique," arXiv.org, 2018.
[10] S. Nur Mutiara, R. Pola, and D. Tresna, "Vision-Based Pipe Monitoring Robot For Crack Detection Using Canny Edge Detection Method as an Image Processing Technique," Kinetik, vol. 2, no. 4, pp. 243-250, 2017.
[11] L. Liu et al., "Deep Learning for Generic Object Detection: A Survey," arXiv:1809.02165, 2018. [Online] Available: https: //arxiv.org/abs/1809.02165.
[12] S. Agarwal and F. Jurie, "Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks," arXiv.org, 2019.
[13] D. Jia, D. Wei, R. Socher, L. Li-Jia, L. Kai, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," ed, 2009, pp. 248-255.
[14] P. Sinno Jialin and Y. Qiang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, 2010.
[15] S. Xia et al., "Transferring Ensemble Representations Using Deep Convolutional Neural Networks for Small-Scale Image Classification," IEEE Access, vol. 7, no. 99, pp. 168175-168186, 2019.
[16] W. Cui, G. Zheng, Z. Shen, S. Jiang, and W. Wang, "Transfer Learning for Sequences via Learning to Collocate," arXiv.org, 2019.
[17] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan, "A theory of learning from different domains," Machine Learning, vol. 79, no. 1-2, pp. 151-175, 2010.

[18] Y. Ganin et al., "Domain-Adversarial Training of Neural Networks," arXiv.org, 2016.
[19] P. Sinno Jialin, I. W. Tsang, J. T. Kwok, and Y. Qiang, "Domain Adaptation via Transfer Component Analysis," IEEE Transactions on Neural Networks, vol. 22, no. 2, pp. 199-210, 2011.
[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," arXiv.org, 2014.
[21] R. Girshick, "Fast R-CNN," Microsoft Research, arXiv: 1504.08083, 2015. [Online] Available: https://arxiv.org/abs/1504.08083.
[22] R. Shaoqing, H. Kaiming, R. Girshick, and S. Jian, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 2017.
[23] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," arXiv.org, 2016.
[24] W. Liu, D. Anguelov, C. Szegedy, S. Reed, F. Cheng-Yang, and A. Berg, "SSD: Single Shot MultiBox Detector," vol. 9905, ed. Ithaca, 2016.
[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," vol. 2016-, ed, 2016, pp. 779-788.
[26] L. Tsung-Yi, P. Goyal, R. Girshick, H. Kaiming, and P. Dollar, "Focal Loss for Dense Object Detection," vol. 2017-, ed, 2017, pp. 2999-3007.
[27] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection," arXiv.org, 2016.
[28] S. Bell, C. Zitnick, K. Bala, and R. Girshick, "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks," arXiv.org, 2015.
[29] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse Connection with Objectness Prior Networks for Object Detection," arXiv.org, 2017.
[30] P. Chao et al., "MegDet: A Large Mini-Batch Object Detector," arXiv.org, 2018.
[31] B. Singh and L. Davis, "An Analysis of Scale Invariance in Object Detection - SNIP," arXiv.org, 2018.
[32] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation Networks for Object Detection," arXiv.org, 2018.
[33] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into High Quality Object Detection," arXiv:1712.00726, 2017. [Online] Available: https://arxiv.org/abs/1712.00726.
[34] H. Xu, X. Lv, X. Wang, R. Zhou, N. Bodla, and R. Chellappa, "Deep Regionlets for Object Detection," arXiv.org, 2018.
[35] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A Real-Time Object Detection System on Mobile Devices," arXiv:1804.06882, 2018. [Online] Available: https://arxiv.org/abs/1804.06882.
[36] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640-651, 2017.
[37] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for Object Segmentation and Fine-grained Localization," arXiv.org, 2015.
[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," arXiv.org, 2016.
[39] Y. Fisher and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," arXiv.org, 2016.
[40] J. Wang and A. K. Asundi, "A computer vision system for wineglass defect inspection via Gabor-filter-based texture features," Information Sciences, vol. 127, no. 3, pp. 157-171, 2000.
[41] J. Wang, Y. Liu, D. Zhang, H. Peng, and Y. Zhu, "A new computer vision based multi-indentation inspection system for ceramics," An International Journal, vol. 76, no. 2, pp. 2495-2513, 2017.
[42] M. Quintana, J. Torres, and J. M. Menendez, "A Simplified Computer Vision System for Road Surface Inspection and Maintenance," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 3, pp. 608-619, 2016.
[43] A. l. Gonzalez et al., "Automatic Traffic Signs and Panels Inspection System Using Computer Vision," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 2, pp. 485-499, 2011.
[44] R. G. Saeidi, M. Latifi, S. S. Najar, and A. G. Saeidi, "Computer Vision-Aided Fabric Inspection System for On-Circular Knitting Machine," Textile Research Journal, vol. 75, no. 6, pp. 492-497, 2005.
[45] J. Lopez, M. Cobos, and E. Aguilera, "Computer-based detection and classification of flaws in citrus fruits," Neural Computing and Applications, vol. 20, no. 7, pp. 975-981, 2011.
[46] R. Seulin, F. Merienne, and P. Gorria, "Simulation of Specular Surface Imaging Based on Computer Graphics: Application on a Vision Inspection System," EURASIP Journal on Advances in Signal Processing, vol. 2002, no. 7, pp. 1-10, 2002.
[47] L. Taylor and G. Nitschke, "Improving Deep Learning using Generic Data Augmentation," arXiv.org, 2017.
[48] D. Han, Q. Liu, and W. Fan, "A new image classification method using CNN transfer learning and web data augmentation," Expert Systems With Applications, vol. 95, pp. 43-56, 2018.
[49] R. Takahashi, T. Matsubara, and K. Uehara, "Data Augmentation using Random Image Cropping and Patching for Deep CNNs," IEEE Transactions on Circuits and Systems for Video Technology, pp. 1-1, 2019.
[50] H. Huang, H. Zhou, X. Yang, L. Zhang, L. Qi, and A.-Y. Zang, "Faster R-CNN for marine organisms detection and recognition using data augmentation," Neurocomputing, vol. 337, pp. 372-384, 2019.
[51] M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, and S. W. Baik, "Multi-grade brain tumor classification using deep CNN with extensive data augmentation," Journal of Computational Science, vol. 30, pp. 174-182, 2019.
[52] C. Shorten and T. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," Journal of Big Data, vol. 6, no. 1, pp. 1-48, 2019.
[53] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84-90, 2017.
[54] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors," arXiv.org, 2017.
[55] K. He, G. Gkioxari, P. Doll´ar, and R. Girshick, "Mask R-CNN," Facebook AI Research (FAIR), arXiv:1703.06870, 2018. [Online] Available: https://arxiv.org/abs/1703.06870.
[56] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv.org, 2015.
[57] C. Szegedy et al., "Going Deeper with Convolutions," arXiv:1409.4842, 2014. [Online] Available: https://arxiv.org/abs/1409.4842.
[58] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," arXiv.org, 2016.
[59] G. Huang, Z. Liu, and K. Weinberger, "Densely Connected Convolutional Networks," arXiv.org, 2018.
[60] Y. Chen, J. Li, H. Xiao, X. Jin, and J. Feng, "Dual Path Networks," arXiv.org, 2017.
[61] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park, "PVANET: Deep but Lightweight Neural Networks for Real-time Object Detection," arXiv.org, 2016.
[62] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," arXiv.org, 2016.
[63] T. Nakazawa and D. V. Kulkarni, "Wafer Map Defect Pattern Classification and Image Retrieval Using Convolutional Neural Network," IEEE Transactions on Semiconductor Manufacturing, vol. 31, no. 2, pp. 309-314, 2018.
[64] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "Object Detection from Scratch with Deep Supervision," arXiv:1809.09294, 2018. [Online] Available: https://arxiv.org/abs/1809.09294.
[65] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," arXiv.org, 2017.
[66] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," arXiv:1611.05431, 2016. [Online] Available: https://arxiv.org/abs/1611.05431.
[67] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv.org, 2015.

[68] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," vol. 15, ed, 2011, pp. 315-323.
[69] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.
[70] V. Santhanam and L. S. Davis, "A Generic Improvement to Deep Residual Networks Based on Gradient Flow," IEEE Transactions on Neural Networks and Learning Systems, pp. 1-10, 2019.
[71] A. L. KABADE and D. V. G. Sangam, "Canny edge detection algorithm," International Journal of Advanced Research in Electronics and Communication Engineering, vol. 5, no. 5, May 2016.
[72] O. R. Vincent and O. Folorunso, "A Descriptive Algorithm for Sobel Image Edge Detection," Proceedings of Informing Science & IT Education Conference (InSITE) 2009.
[73] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," arXiv.org, 2016.

JUNGHEE HAN is currently pursuing the Ph.D. degree with Graduate School of Convergence Science and Technology (GSCST), Seoul National University, Korea. He is currently a principal research engineer with Korea Aerospace Industries, LTD. His research interests include computer vision, deep learning, object detection and instance segmentation.

SEONGSOO HONG is currently a professor with Department of Electrical and Computer Engineering at Seoul National University and the associate dean of Graduate School of Convergence Science and Technology at Seoul National University. Prof. Hong received the B.S. and M.S. degrees in computer engineering from Seoul National University, Korea in 1986 and 1988, respectively. He received the Ph.D. in computer science from the University of Maryland, College Park in 1994. His current research interests include embedded real-time systems design, real-time operating systems, embedded middleware, and software tools and environments for embedded real-time systems. Prof. Hong served as a general co-chair of IEEE RTCSA 2006 and CASES 2006 and as a program committee co-chair of IEEE RTAS 2005, RTCSA 2003, IEEE ISORC 2002 and ACM LCTES 2001. He has served on numerous program committees including IEEE RTSS and ACM OOPSLA. He is currently a senior member of the IEEE and a senior member of the ACM.