World Academy of Science, Engineering and Technology
International Journal of Cognitive and Language Sciences
Vol:15, No:2, 2021

Image Ranking to Assist Object Labeling for Training Detection Models

Tonislav Ivanov, Oleksii Nedashkivskyi, Denis Babeshko, Vadim Pinskiy, Matthew Putman

Abstract—Training a machine learning model for object detection that generalizes well is known to benefit from a training dataset with diverse examples. However, training datasets usually contain many repeats of common examples of a class and lack rarely seen examples. This is due to the process commonly used during human annotation where a person would proceed sequentially through a list of images labeling a sufficiently high total number of examples. Instead, the method presented involves an active process where, after the initial labeling of several images is completed, the next subset of images for labeling is selected by an algorithm. This process of algorithmic image selection and manual labeling continues in an iterative fashion. The algorithm used for the image selection is a deep learning algorithm, based on the U-shaped architecture, which quantifies the presence of unseen data in each image in order to find images that contain the most novel examples. Moreover, the location of the unseen data in each image is highlighted, aiding the labeler in spotting these examples. Experiments performed using semiconductor wafer data show that labeling a subset of the data, curated by this algorithm, resulted in a model with a better performance than a model produced from sequentially labeling the same amount of data. Also, similar performance is achieved compared to a model trained on exhaustive labeling of the whole dataset. Overall, the proposed approach results in a dataset that has a diverse set of examples per class as well as more balanced classes, which proves beneficial when training a deep learning model.

Keywords—Computer vision, deep learning, object detection, semiconductor.

I. Introduction

DEEP learning models have become popular in object detection tasks. These models require a large amount of labeled data to effectively train. In many cases, collecting large amounts of data is not difficult, but manually labeling that data can be very tedious and time consuming. Automatic labeling is increasing its applicability [1], [2] but requires a prior labeled dataset to learn from. So, for datasets containing novel objects, manual labeling is necessary. Since only a subset of the data can be labeled in the constraints of cost and time, a way to select this subset is desirable.

Currently, most labelers naively go through as much images as possible proceeding in the order that the images appear in the host directory. As they do that, they become very skilled at detecting classes with high number of examples due to the repetitive exposure to them. Thus, they are more likely to miss examples of a rare class. For RCNNs [3], this presents an extremely large problem as areas that are not classified into a labeled class are treated as background and cause the model to not detect those type of defects [4]. Consequently, the resulting labeled data, which has excess repeats of similar examples and lacks rare examples, is ill-suited for training a model. Moreover, the labeler often receives little feedback on the use of their data as to have any knowledge of how to formally optimize the labeling process. Even if a labeler can be instructed on how to ignore certain images, compliance with such instructions can be difficult and vary when dealing with many labelers. Furthermore, going through every image in a very large dataset in order to select a subset is infeasible for a human. Thus, a computerized approach to aid the labeler in selecting a subset of images for labeling is necessary.

We present a model that provides a ranked list of images for the labeler to label. The model is trained in an active learning fashion where one trains the model on a small initial dataset, uses the model to assist labeling the next batch of data, and then retrains the model including the new data. This multi-staged approach is used to continuously update the order of the images in the dataset to be manually labeled. Our model ranks the images with the most novel examples first, thus, creating a diverse set of labeled data which can be successfully used in the training of an object detector. To perform this ranking, we employ a deep learning segmentation model, based on the UNet architecture, that predicts the areas in an image containing previously unseen data. We focus our experiments on datasets where the objects are distributed randomly, vary in morphology within class, and have different quantity per class. We show that for these datasets we can achieve high accuracy detection models by labeling only an algorithmically chosen subset of the images. Our approach can be expanded to other types of models that use supervised training and to other types of datasets.

II. Prior Work

Automatic labeling of images has been explored in [5] and [6]. In this paradigm, one would run a model trained on previous data to generate the labels and bounding boxes on the desired dataset, and the labeler would simply modify the bounding boxes. In the proprietary material domains, such as semiconductors, automatic labeling approaches fail due to the lack of previously labeled data of the desired classes. Our method addresses this problem by requiring the operator to initially label a small number of images to define the classes for the overall classification task. This is sufficient to jump start our model, but it would not be sufficient to construct an automatic labeling model.

In close relation to our work, [7] performs an active learning using a latent SVM to select successive batches of data for

T. Ivanov, O. Nedashkivskyi, D. Babeshko, V. Pinskiy, and M. Putman are with Nanotronics Imaging, Brooklyn, NY, USA (e-mail: {tivanov, onedashkivskyi, dbabeshko, vpinskiy, matthew}@nanotronics.co).

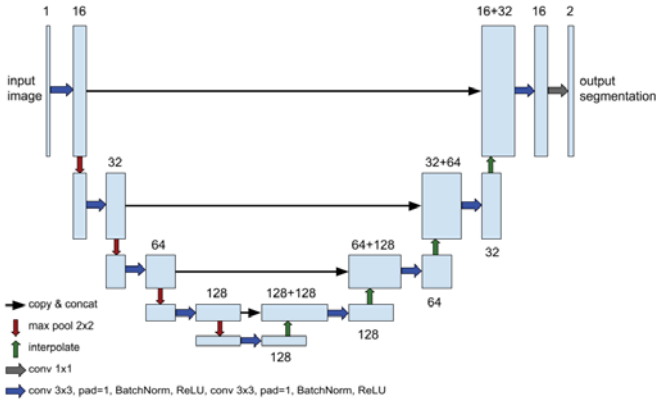

Fig. 1 Schematic of the deep-learning UNet based model

labeling. We perform the same procedure but use a CNN model. However, we do not present a bounding box with a class label to the labeler but rather highlight a suggested area for the labeler to look at. Nonetheless, both approaches aim to select the most uncertain or unseen unlabelled data.

It is known that CNN detection models suffer from bias when trained on a imbalanced dataset [8]. Some approaches modify the learning algorithm of the detector to combat the class imbalance problem already present in the labeled data. For example, [9] introduces a Class Rectification Loss (CRL) regularising algorithm to the CNN learning process by performing minority class hard samples mining. Reference [10] uses the Generalized Dice overlap as a loss function. Unlike these approaches, we curate the training data to include more rare class examples in the first place, thus, achieving better class balance in the training dataset. Even further, our method can be used in conjunction with the aforementioned training methods to combat the remaining class imbalance.

III. Methods

We use a Convolutional Neural Network based on the UNet architecture [11] to sort images for labeling. The major modification to the UNet was to remove the final softmax layer that produces the segmentation masks so that we obtain directly the pixel probabilities per class. We also changed the convolutional layer padding to match our input size and we changed the number of feature maps used by the convolutional layers. The architecture is shown in Fig. 1.

The UNet model is trained on an initial small subset of images (∼5% of the total set) to establish a baseline where all objects of interest are labeled with rectangular bounding boxes. All the pixels inside the bounding boxes of all labeled objects get grouped into one category "foreground," and the rest of the image pixels are treated as "background." Using this classification, two input segmentation masks are generated - one for the background and one for the foreground. We do not need to use tight segmentation of the foreground objects because we do enlarge the bounding boxes of the foreground objects by 10 pixels to eliminate ambiguity of classifying pixels on the border of bounding boxes. The trained UNet model produces two probability maps the same size as the input image: one giving the probability of each pixel belonging to the background and the other - belonging to the foreground. Using these two maps, we compute a measure that a pixel is unseen before S_{unseen} i.e. neither part of the background nor the observed foreground. Pixels that belong to examples of a new class not yet labeled or to novel-looking examples of a previously labeled class will have a high unseen score.

$$S_{unseen}(x,y) = 1 - P((x,y) \in bg) - P((x,y) \in fg)$$

Using the per-pixel unseen scores, an overall image metric is computed and used for ranking images. A high image metric would mean that the model has detected that there is something unseen before in the image, indicating that this image deserves labeling priority. However, it's unclear which image should have higher priority: an image that has few high scoring pixels or an image that has plenty of low scoring pixels. Therefore, we compute two metrics: threshold metric M_{thresh} and alpha metric M_{alpha}. Threshold metric equals to the number of pixels that have unseen score above some threshold t. In this metric, low scoring pixels will have no influence on the metric.

$$M_{thresh} = \sum_{(x,y)} S_{unseen}(x,y) > t$$

The Alpha metric equals to the sum of all unseen scores at power α. In this metric, all pixels are accounted for but lower scoring pixels have a lesser influence on the score.

$$M_{alpha} = \sum_{(x,y)} S_{unseen}(x,y)^{\alpha}$$

After the images are ranked using one of these metrics, the next batch of images to be labeled is produced. The process iterates in an active learning fashion: the new images are labeled, the UNet model is retrained, including the newly available labeled images, and the model is run to produce the next batch of images to be labeled. After a sufficient amount of data is labeled, one can train a successful detection model. This model can also be used to automatically label the rest of the dataset.

IV. Experiments

We applied our approach on semiconductor data to showcase the broad applicability to proprietary materials where historical examples of target classes are limited. We removed partial and blank images from all datasets since such images are simply an artifact of the imaging system. We tested on two types of semiconductor wafers - device wafer, which contain tiles of devices, and bare wafers, which don't have any devices. These two types of wafers represent the morphological extremes of white-light based conventional inspection microscopy. The image data was captured on the Nanotronics nSpec® optical inspection system. Each sample contained approximately 200 images.

We selected the first 10 images as they appeared in the host directory and labeled all the defects in each image as it is conventionally done. Our labeler was a subject matter expert who determined which types of the objects were considered

defects. The same expert was used for labeling all datasets and for evaluating the predictions of the model. The choice of a single labeler minimized interpersonal bias when comparing different datasets.

We then trained our UNet model and used it to select the next 10 images for subsequent labeling using the alpha metric for ranking. We iterated through this process two times. Finally, we trained a Faster-RCNN [12] object detection model on all 30 labeled images. Note, that a different detection architecture can be used and we obtained similar results with CenterNet [13].

For comparison with the conventional approach for labeling, we trained a Faster-RCNN model on sequentially labeled images. We labeled as many images in the same time that it took to label the algorithmically selected images, resulting in approximately the same total number of bounding boxes labeled. Also, we trained a RCNN model on the entire labeled dataset where available. All models were evaluated on the same set-aside and labeled test set. The hyperparameters of the RCNN were kept the same for all experiments. We trained on each dataset for 100 epochs. We did not tune parameters to obtain the optimal RCNN model, but our results are sufficient for a comparison.

The metrics we used to measure RCNN performance were precision, recall, and F1 score. Both metrics were calculated on pre-class basis. The precision equals to the number of bounding boxes that overlap with the ground truth bounding boxes more than 50% over the total number of RCNN output bounding boxes for that class. The recall is the number of bounding boxes that overlap with the ground truth bounding boxes more than 50% over the total number of bounding boxes in the ground truth for that class. The F1 score is computed using the formula below.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

V. Results

We evaluated our model on both types of semiconductor wafers, including single and multi-class data. We compared performance against a sequentially labeled subset and a fully labeled dataset. The amount of total labels in the entire dataset was far greater than that used by our model.

The results on the device single class dataset are shown in Table I. The devices on the wafer are considered native and part of the background. The classification is focused solely on the particles regardless of their location on the wafer. This single-class labeling is commonly performed for gross surface contamination detection and overall particle counting. The precision and F1 score of the Faster-RCNN model trained on the initial 10 images start out roughly 20% lower compared to the fully labeled dataset. The precision and F1 score steadily increase over the next two iterations, resulting in a difference of just 10% with the full model. Each iteration includes 10 new images selected by our method for a total of 30 images in the last. The full dataset consisted of 194 images. Thus, we achieved a good performance using only 15% of all training images. The precision and F1 score are expected to naturally improve with more iterations, but future experimentation will assist in quantifying the number of iterations required to match or exceed the performance of the full model. Also, the model shows roughly a 10% increase in the precision and F1 score compared to sequential labeling on similar amounts of labels as shown in Figure II. Thus, sorting the images using our approach outperforms sequential labeling. We expect similar results if we compare against random image labeling. Note that for device images in general manual detection of rare objects is very difficult as the repetitive pattern of the devices creates a complex background that is harder to differentiate than a empty bare wafer. This is seen in the low overall precision and F1 score.

TABLE I
Performance on Single-Class Dataset (Device Wafer) of RCNN Model Trained with Data Labeled Using Our Approach and with the Full Dataset Labeled

Num. Images	Init	Iter1	Iter2	Full
Precision	0.44	0.51	0.59	0.69
Recall	0.86	0.85	0.86	0.87
F1 score	0.58	0.64	0.70	0.77

The results show a steady improvement in the precision and F1 score as iterations increase.

TABLE II
Performance on Single-Class Dataset (Device Wafer) of RCNN Model Trained with Data Labeled Using Our Approach and Same Amount of Data Labeled Sequentially

	Our Model	Sequential Model
Precision	0.59	0.47
Recall	0.86	0.84
F1 score	0.70	0.60

The results on the bare wafer multi-class dataset are shown in Table III. The dataset contains three classes of different particle types on the substrate wafer. As seen in Fig. 2, the classes contain irregular defects of various types, which are quite different in intensity and morphology. These objects originate in different parts of the production pipeline and are grossly different. Their detection is expected to be uncorrelated. For the initial 10 selected images, the F1 score starts out 13% and 14% less for class A and B as compared to the fully labeled set. Over the next two iterations, the F1 score increases for class A and gets to 1.0 which is 0.42 higher compared to the fully labeled set. For class B, a similar pattern is shown. We suppose that the poor performance of the full model is due to the excess number of repetitive examples that cause the model to overfit. Also, there are a greater number of labeling errors present in the full dataset which can confuse the model. For class C, the F1 score increases from the nominal 0, due to the lack of adequate labels in the first 10 images, to 73% which is 8% less compared to that of the fully labeled dataset. The F1 score is also expected to increase with additional iteration adding new examples of this rare class.

The Faster-RCNN performance can be further increased through fine tuning of the loss function and data augmentation of the labeled cases. This work focused exclusively on unbiased comparison of performance as a function of labels

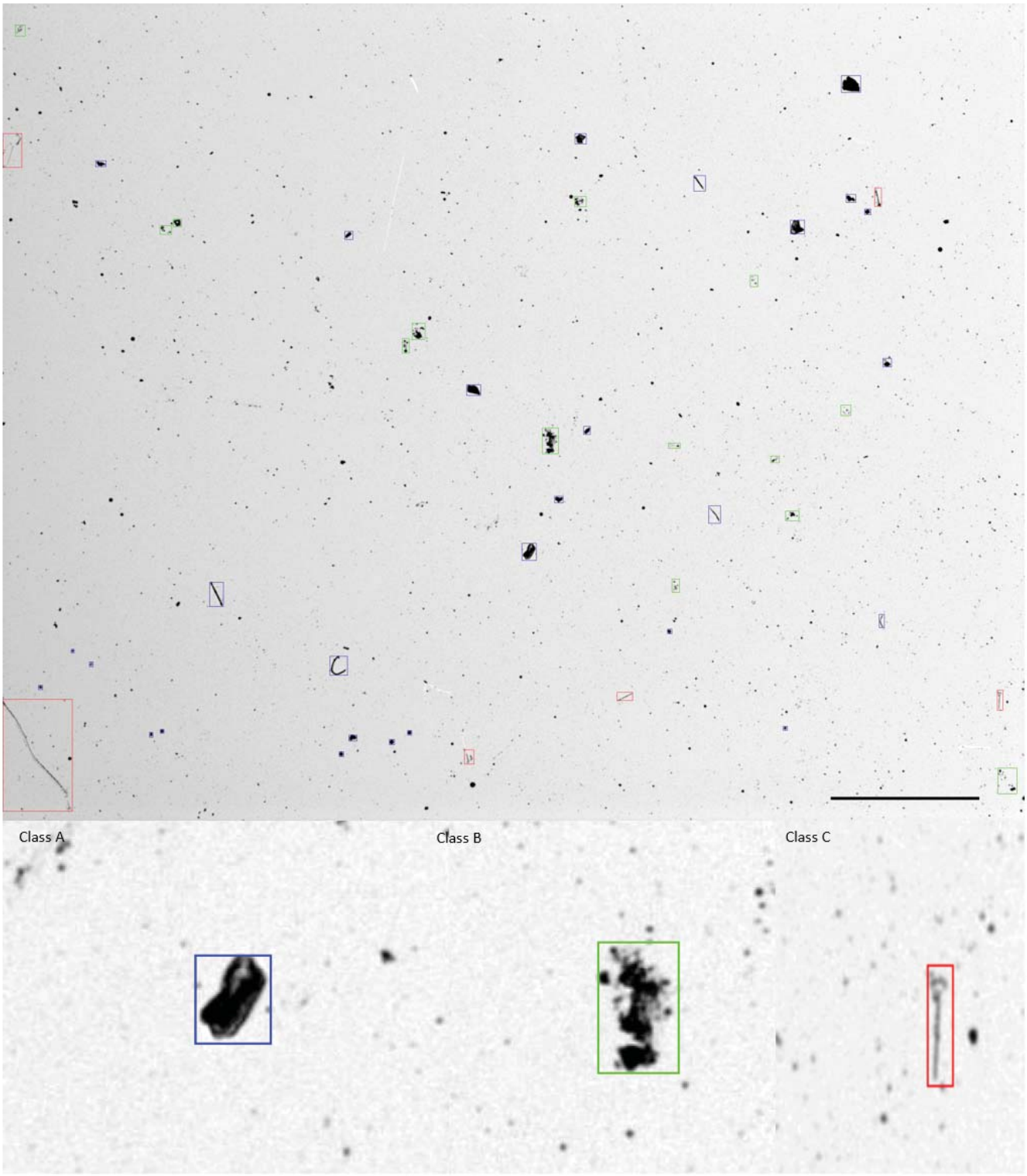

Fig. 2 Bare wafer, multi-class dataset with labeled defects. Classes are shown on the bottom row of the figure. The defects vary by type and are separated by morphology in different classes. The scale bar is 400 microns

TABLE III
PERFORMANCE ON MULTI-CLASS DATASET (BARE WAFER) OF RCNN MODEL TRAINED WITH DATA LABELED USING OUR APPROACH AND WITH THE FULL DATASET LABELED

Class	Metric	Init	Iter1	Iter2	Full
A	Precision	0.67	0.67	1.0	0.4
	Recall	0.34	1.0	1.0	1.0
	F1 score	0.45	0.8	1.0	0.58
B	Precision	0.4	0.34	1.0	0.4
	Recall	0.4	0.4	0.8	0.8
	F1 score	0.4	0.37	0.89	0.54
C	Precision	0	0.67	0.8	0.95
	Recall	0	0.09	0.67	0.71
	F1 score	0	0.15	0.73	0.81

and did not concentrate on optimization of the detection model. The same Faster-RCNN hyperparameters were used for all cases.

The time overhead added by training and running the UNet model for image ranking was on average across the datasets 17 min per iteration. This overhead is insignificant given the time to label a batch of images, which was on average 3 hours across datasets. Moreover, in a sparse dataset, the overhead of skipping through images that contain no or few examples can be considerable. Our approach will show the labeler the images that contain a lot of defects which is a big time saver.

In addition to ranking images, the model can highlight the specific areas of previously unseen data in each image to aid the labeler. This will decrease the bias of the labeler towards already seen examples and make it easier to spot the rare class examples that are so vital for training a successful detection model.

VI. FUTURE WORK

Future work will focus on outlining and sorting for labeling the regions of interest (ROIs) inside each image, which will aid the labeler when given images with complex background conditions. Such an enhancement to the current approach can be done by overlaying the unseen score probability map on top of the image, which highlights the suggested regions of the image to be labeled as shown in Fig. 3. Higher intensity denotes greater importance of the region. This overlay can help with class balancing and detection of rare examples. This visual aid would be integrated into the labeling application to drive the labelers focus on the important parts of the image. Such a probability map would also improve the overall speed of the image labeling. Additionally, one can also overlay domain-specific elements such as the device boundaries.

Another improvement to our approach is to calculate a better image metric for ranking images. We can perform morphology to combine unseen pixels that belong to a single example. This will produce a measure that is not dependent on the size of the examples. We would also want to dynamically adjust the trade off between a few unseen examples with high confidence versus many unseen examples with lower confidence based on the class imbalance of the dataset.

Our approach will provide a great benefit when future development of the RCNN model enable it to use partially labeled images. Then, the labeler can avoid exhaustively labeling the entire image containing many common defects and instead only label the unseen/rare defects highlighted by our UNet model. This process will quickly lead to a high-performance detection model.

VII. CONCLUSION

Manual image labeling and classification is the underpinning of all image processing pipelines. Although automatic labeling continues to improve, it is yet to fully replace manual labeling and cannot be applied in all domains. The presented model focused on sorting images for manual labeling and identifying only select images to be manually labeled. We show that in a single-class experiment the presented approach requires a small amount of the fully labeled dataset to achieve similar detection performance. We also show that in the case of the multi-class model on semiconductor devices excessive labeling of the full dataset produces degraded results compared to two iterations of images labeled using our approach. Labeling of only selected images decreases overfitting and allows for high detection rates. This work shows how a UNet style model can be used to assist manual labeling by decreasing the amount of effort while improving overall performance of deep learning pipelines trained on the labeled data.

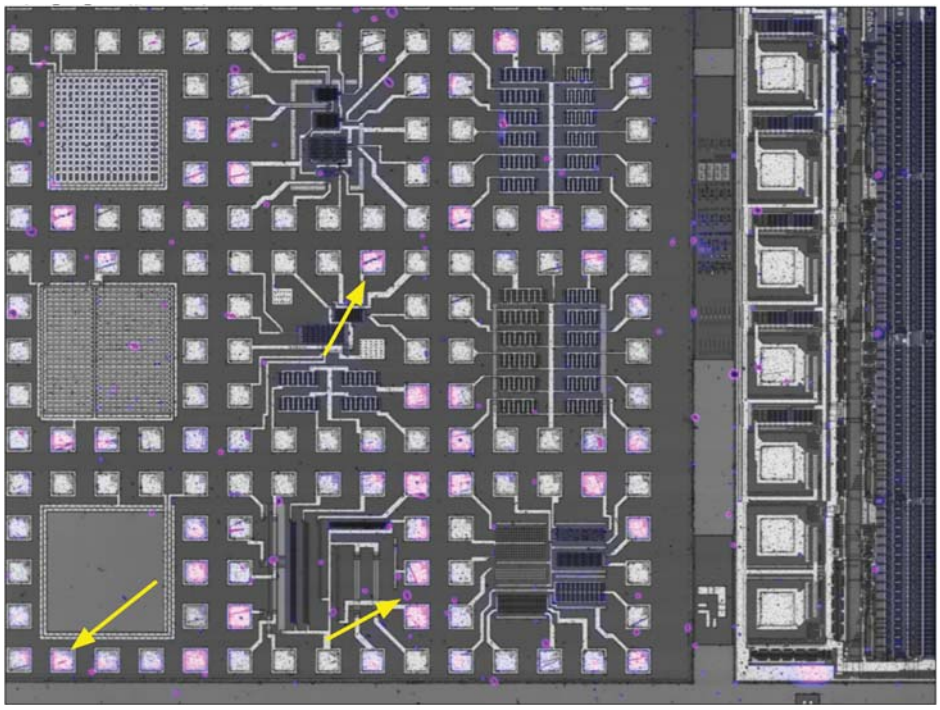

Fig. 3 Probability Map of the Device. The device image is shown in monochrome. The overlaid Unet probability map is shown as a purple heatmap. Higher intensity indicates a higher probability. The yellow arrows point to sample defects in the image

REFERENCES

[1] A. Braun and A. Borrmann, "Combining inverse photogrammetry and bim for automated labeling of construction site images for machine learning," *Automation in Construction*, vol. 106, p. 102879, 2019.
[2] G. H. Weber, C. Ophus, and L. Ramakrishnan, *Automated Labeling of Electron Microscopy Images Using Deep Learning*. IEEE, 2018.
[3] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
[4] H. Alhammady and K. Ramamohanarao, "Using emerging patterns and decision trees in rare-class classification," in *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pp. 315–318, IEEE, 2004.
[5] L.-C. Chen, S. Fidler, A. L. Yuille, and R. Urtasun, "Beat the mturkers: Automatic image labeling from weak 3d supervision," in *Proceedings of CVPR*, 2014.
[6] L. Zhang, Y. Tong, and Q. Ji, "Active image labeling and its application to facial action labeling," in *Proceedings of ECCV*, pp. 706–719, 2008.
[7] K. Okuma, E. Brochu, D. G. Lowe, and J. J. Little, "An adaptive interface for active localization," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, pp. 248–258, 2011.
[8] A. More, "Survey of resampling techniques for improving classification performance in unbalanced datasets," *arXiv preprint arXiv:1608.06048*, 2016.
[9] Q. Dong, S. Gong, and X. Zhu, "Class rectification hard mining for imbalanced deep learning," *CoRR*, vol. abs/1712.03162, 2017.
[10] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *CoRR*, vol. abs/1707.03237, 2017.
[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.
[13] X. Zhou, D. Wang, and P. Krahenbuhl, "Objects as points," *arXiv preprint arXiv:1904.07850v2*, 2019.