Semiconductor Wafer Defect Detection using Deep Learning

Prof. Rakhi Bharadwaj
Department of Computer Engineering
Vishwakarma Institute of Technology, Pune, Maharashtra, India
rakhi.bharadwaj@vit.edu

Mr. Yashraj Pawar
Department of Computer Engineering
Vishwakarma Institute of Technology, Pune, Maharashtra, India
yashraj.pawar19@vit.edu

Ms. Shruti Pisal
Department of Computer Engineering
Vishwakarma Institute of Technology, Pune, Maharashtra, India
shruti.pisal19@vit.edu

Mr. Vishal Thoke
Department of Computer Engineering
Vishwakarma Institute of Technology, Pune, Maharashtra, India
vishal.thoke19@vit.edu

Mr. Ashutosh Zavar
Department of Computer Engineering
Vishwakarma Institute of Technology, Pune, Maharashtra, India
ashutosh.zawar19@vit.edu

Abstract — Accurately detecting and classifying defects in wafers is a crucial aspect of semiconductor manufacturing. This process provides useful insights for identifying the root causes of defects and implementing quality management and yield improvement strategies. The traditional approach to classifying wafer defects involves manual inspection by experienced engineers using computer-aided tools. However, this process can be time-consuming and less accurate. As a result, there has been increasing interest in using deep learning approaches to automate the detection of wafer defects, which can improve the accuracy of the detection process.

Keywords — Wafer detection, Deep learning, Object detection,Classification

I. INTRODUCTION

Wafer:
In our everyday lives, we use electronic devices that contain an ASIC or an IC, which are made up of silicon dies that perform specific functions. These silicon dies are created from semiconductor wafers, which are thin slices of semiconductor materials, such as gallium arsenide or silicon. These materials are chosen for their ability to conduct electricity in specific conditions while insulating electricity in others. The semiconductor wafers are typically round or rectangular in shape and are created by slicing a block of semiconductor material into thin sheets with a wafer saw. After being cut, the wafers are polished and cleaned to remove any defects or contaminants that could impact the performance of microelectronic devices built on them. A thin layer of photoresist is then applied to the wafer,

and a pattern is created using photolithography. The photoresist areas that have been exposed to light are removed, leaving the desired pattern on the wafer. The wafer is then subjected to various chemical and physical processes to produce tiny electrical components that make up microelectronic devices. These processes include depositing layers of material, etching away unwanted material, and introducing impurities into the semiconductor material to create different types of electrical components. Once the wafer is fabricated, the microelectronic device is cut into individual chips or dice, which are then packaged and tested. These finished chips are used in various electronic devices, such as computers, smartphones, and other similar devices.

Artificial Intelligence:
The field of artificial intelligence (AI) is focused on the creation of machines that can display human-like intelligence and decision-making abilities. This involves the development of algorithms and models that can analyze and process data to achieve specific goals. There are different approaches to building AI systems, such as rule-based systems, decision tree systems, neural networks, and deep learning. Rule-based systems use predetermined rules to perform simple tasks, while decision tree systems use a series of questions and answers to make decisions. The structure of neural networks is in a way that is similar to the human brain.Deep learning, a form of machine learning, utilizes neural networks comprising multiple layers to handle large datasets and accomplish complex tasks like speech and image recognition. While AI has the potential to improve various industries and simplify our lives, it also raises ethical issues and may lead to job displacement. Therefore, research and debate surrounding the development and implementation of AI continue to be active areas of inquiry.

Neural Network:
Machine learning algorithms known as neural networks imitate the structure and function of the human brain.These networks consist of interconnected "neurons" arranged in layers that process and transmit information. Neural networks excel at recognizing patterns and relationships in data and can be trained to perform various tasks, including speech and image recognition, language translation, and decision-making. They are ideal for tasks that require the processing of large datasets and the ability to learn and adapt to new situations. The types of neural networks include feedforward neural networks, convolutional neural networks, and recurrent neural networks, with the selection depending on the task at hand. Training neural networks involves using an optimization algorithm and a vast amount of labeled data to adjust the connections' weights and biases between neurons. Although neural networks are effective at various tasks, they can be computationally demanding and require large amounts of training data. Additionally, the quality of the data used for training affects their sensitivity, and if the training data is not representative of real-world data, they may overfit.

Wavelet Transform:
Wavelet transform is a mathematical technique commonly used in image processing to break down an image into different frequency components, enabling it to be analyzed at different resolutions or scales. This is particularly useful for tasks such as image compression, denoising, and feature extraction. To perform wavelet transform on an image, the image is first divided into small blocks or pixels, and each block is then transformed into the frequency domain using wavelet transform. The resulting wavelet coefficients represent the amplitude and phase of the different frequency components of the image at varying scales.

One important advantage of wavelet transform in image processing is its ability to provide multiresolution analysis of the image, which enables the image to be analyzed at different levels of detail. This can be particularly useful for tasks such as denoising, where it is important to retain fine details while removing noise. Wavelet transform is also utilized in image compression, where it can be used to eliminate redundancy and irrelevance in the image data. By preserving only the most significant wavelet coefficients, it is possible to reconstruct an approximation of the original image with a lower data rate. Overall, wavelet transform is a powerful tool in image processing, with broad applications across various fields.

II. Literature Review

In [1], the authors discussed the importance of exploring different combinations of features that enhance the accuracy of Convolutional Neural Networks (CNNs) on large datasets. While some features may only be effective for certain models or problems or may be limited to smaller datasets, there are universal features that can be applied to most models, tasks, and datasets. Examples of such features include Weighted Residual Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT),

Mish-activation, Batch Normalization, Mosaic data augmentation, DropBlock regularization, and CIoU loss. In order to attain cutting-edge results, the authors used these universal features as well as a mixture of several of them, including WRC, CSP and CIoU loss. In [2], the authors introduced the Cross-Stage-Partial Network (CSPNet) that mitigates the problem of high inference computations required by previous works from a network architecture perspective.The proposed network topology reduces calculations by 20% while preserving or enhancing accuracy on the ImageNet dataset by combining feature maps from the beginning and end of a network stage. On the MS COCO object detection dataset, it also beats current state-of-the-art methods in terms of AP50.The ResNet, ResNeXt, and DenseNet-based designs can be handled by the CSPNet because it is simple to implement and sufficiently generic. In [3], The SPP-net, a novel network topology that does away with the need for a fixed-size input picture for current deep CNNs, was proposed by the authors.The SPP-net is resistant to object deformations since it consistently represents images of any size or scale. The authors demonstrated that the SPP-net improves the accuracy of several CNN architectures, regardless of their unique designs, through studies on the ImageNet 2012 dataset. In [4], The authors developed an intelligent indoor positioning system devoid of infrastructure that relies solely on visual data. The suggested plan makes use of smartphones as client premises equipment and readily available environmental items as location landmarks.Directional pedestrian signs are detected and recognised using the Google Object Detection framework. The testing findings show that the suggested method can identify pedestrian directional signs with up to 98% accuracy. In [5], the authors presented a survey on image classification using deep learning as well as transfer learning approaches, discussing unsupervised, semi-supervised, and supervised learning strategies.

III. PROPOSED ALGORITHM

3.1 Image or Data Augmentation

Data augmentation is a technique that involves generating additional data based on existing data. In this project, we utilized data augmentation to improve the performance of our object detector YOLO8 model by augmenting the image datasets. There are several ways to augment an image dataset, and we employed the following techniques:

Flipping: This involves horizontally or vertically flipping images to produce additional data. This technique is particularly useful for object detection tasks, where the same object may appear in different orientations in the image.

Rotating: Images can be rotated to generate additional data, which is beneficial for tasks such as object recognition where the same object may appear at different angles in the image.

Scaling: Scaling images up or down can also be used to create additional data. This technique can be helpful for object detection tasks, where the size of the objects in the image may vary.

Adding noise: Adding noise to images can be an effective way to generate additional data. This technique is especially useful for image classification tasks, where the model needs to be able to handle noise in the input data.

Overall, data augmentation is a valuable tool for improving the performance of machine learning models, and the various techniques employed can help to create more diverse and robust datasets.In this project we had around 30 base images which we augmented using the above techniques to about 933 images.

3.2 Model training:

To train a YOLOv8 model, we have gathered a large dataset of annotated images that are labeled with the bounding boxes and class labels of the objects in the image. Then data is split into a training ,validation and testing set, and uses the training set to train the model using an optimization algorithm. After the model has been trained, we have evaluated its performance on the testing set to see how well it generalizes to new data.

• Training Data: The dataset with 933 images is divided into training ,validation and test data, with training data accounting for 80 percent of the dataset i.e around 765 images. Again, data augmentation is used for this data because it artificially increases the data by three or more times.

• Validation Data: Validation data accounts for 20 percent of the dataset which resulted in about 150 images. Dividing the dataset during the training phase allows you to see how the algorithm performs on the training data. However, it cannot use data augmentation in this case and must instead use raw images.

• Testing data: We kept 18 images to test and check the accuracy and speed of the model for inference or testing its performance

Hyperparameters are a set of parameters that are defined before training a machine learning model, and their values determine how the model performs and behaves. They are frequently used to adjust the model's complexity, and can significantly influence its performance.

One of the most important hyperparameters is the learning rate, which controls the step size of the optimization algorithm that updates the model's parameters during training. While a larger learning rate can lead to faster convergence, it may also make the optimization process more unstable. In our model, we have set the learning rate to 0.001.

Another important hyperparameter is the batch size, which determines the number of training examples used in each iteration of the optimization algorithm. A larger batch size can improve the efficiency of the optimization process, but it may also increase the memory requirements of the model. In this model, we have used batch sizes of 64.

The number of epochs is one more important hyperparameter that controls the number of times the optimization algorithm will iterate over the training data. Increasing the number of epochs can allow the model to learn more complex features, but it may also increase the risk of overfitting. In our model, we have trained the model for about 50 epochs on the best weight which we got when we trained a small data for about 20 epochs.

Finally, the number of anchors is a hyperparameter that determines the number of bounding boxes used by the model to predict the locations of objects in an image. A larger number of anchors can improve the accuracy of the model, but it may also increase its complexity.

3.3 YOLO v8

YOLOv8 is the latest version of YOLO by Ultralytics.YOLOv8, being a state-of-the-art (SOTA) model, incorporates new features and enhancements to build upon the success of its previous versions. This results in improved performance, flexibility, and efficiency. YOLOv8 supports a full range of vision AI tasks, including detection, segmentation, pose estimation, tracking, and classification. This versatility allows users to leverage YOLOv8's capabilities across diverse applications and domains.

1. Firstly, YOLOv8 is highly accurate as measured by COCO and Roboflow 100, which attests to its precision and reliability.
2. Secondly, YOLOv8 offers a host of developer-convenience features, including an intuitive CLI and a well-structured Python package that simplifies coding tasks.
3. Thirdly, YOLOv8 boasts a large and rapidly growing community of computer vision enthusiasts who can provide assistance and guidance whenever needed.
4. For instance, YOLOv8's medium model (YOLOv8m) attains a remarkable 50.2% mAP on COCO, outperforming YOLOv5 on Roboflow 100. Moreover, YOLOv8's CLI makes training models easier than many other models that require executing various Python files. Additionally, YOLOv8's Python package facilitates seamless coding.
5. Lastly, YOLOv8's popularity and reputation in the computer vision community make it a reliable model to use, with many online guides and resources available to assist users.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)
YOLOv8n	640	37.3	80.4	0.99
YOLOv8s	640	44.9	128.4	1.20
YOLOv8m	640	50.2	234.7	1.83
YOLOv8l	640	52.9	375.2	2.39
YOLOv8x	640	53.9	479.1	3.53

Table 1. Comparison of different versions of v8
source: docs.ultralytics.com

1. Flow Chart I

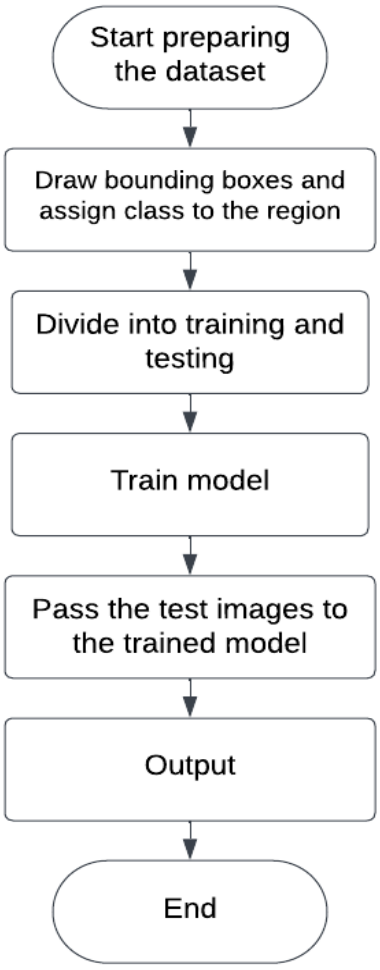

Fig1. The process of training YOLO model

2. Flow Chart II

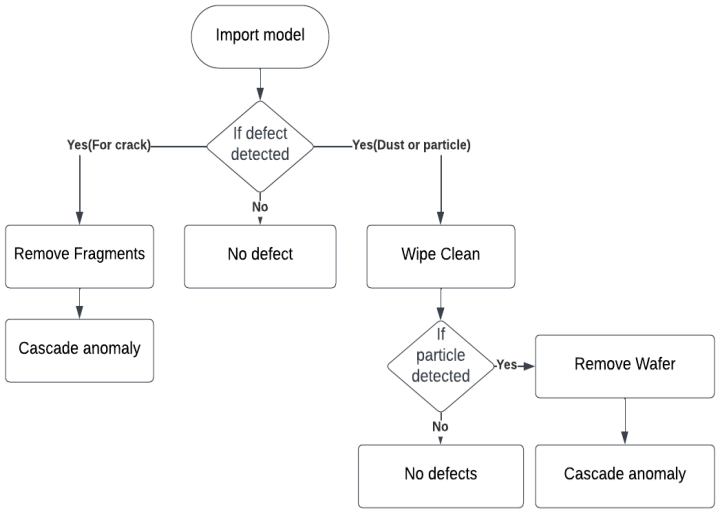

Fig2. Experimental flow chart

IV. EXPERIMENT AND RESULT

4.1 Analysis

For this project, we compiled a vast dataset that underwent several image-processing procedures before being inputted into our model. As depicted in Figure Yolov8,Our object detection technology is able to spot microscopic cracks, long strip cracks, and gaps with cracks. In these test samples, the likelihood of missed detection is extremely low because the system has a tendency to somewhat over-detect. The potential risks mainly fall into four categories, namely:

1. The target frame covering a large portion of the target
2. The existence of hidden cracks at a pixel-level
3.The likelihood of large lines being detected excessively throughout the entire image and
4. Detection of certain controversial gaps or cracks.

Overall, Yolov8 has exhibited excellent robustness for industrial crack detection.

The geometry of the silicon wafer crack defect is unpredictable throughout the detection procedure, and line markings might cause severe interference. To address this issueWe used the Yolov8 target detection algorithm, which can respond to varying fracture sizes and lengths while also making sure that deep and shallow line marks are not over-detected, to find cracks on an industrial silicon chip. The algorithm is quite robust and can recognise silicon wafer cracks in test samples with great accuracy. Uncertain crack defects on silicon wafers can cause severe interference, as can line marks.

To evaluate the effectiveness of target detection, we used the evaluation index on the test set, and the results are as follows:

4.2 MAP:

MAP or mean Average Precision is a metric commonly used to assess object detection models. It calculates the average precision of a model across all classes and measures the model's ability to accurately identify and classify objects in an image. To calculate mAP, the model is first applied to a test dataset, and the predicted bounding boxes are compared to the ground truth bounding boxes. Then, the precision of the model is determined for each class, and the mAP is calculated as the average of the precision values across all classes.
The precision of a model is determined by dividing the number of true positive predictions made by the model by the total number of positive predictions made by the model. A true positive prediction is a prediction that is both accurate and has a high overlap with the ground truth bounding box.

4.3 RECALL:

Recall is a crucial metric for evaluating the performance of a machine learning model, particularly in classification tasks. It measures the number of true positive predictions made by the model in relation to the total number of positive examples in the dataset. True positive predictions are predictions that are both accurate and have a high overlap with the ground truth labels. This overlap is typically assessed using a threshold, such as the Intersection over Union (IoU) metric.
The recall metric is important because it evaluates the ability of the model to correctly identify all of the positive examples in the dataset. A model with a high recall score can correctly identify most or all of the positive examples, while a model with a low recall score may miss many of the positive examples. Therefore, recall is a critical metric that provides insight into how well a model can accurately identify positive examples.

4.4 Test results

The test results are as follows:

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	149	335	0.93	0.927	0.942	0.604
nodefect	149	18	1	0.955	0.995	0.953
spot	149	222	0.849	0.859	0.863	0.331
crack	149	95	0.942	0.968	0.967	0.528

Table 2. Test results displaying map value, precision and recall.

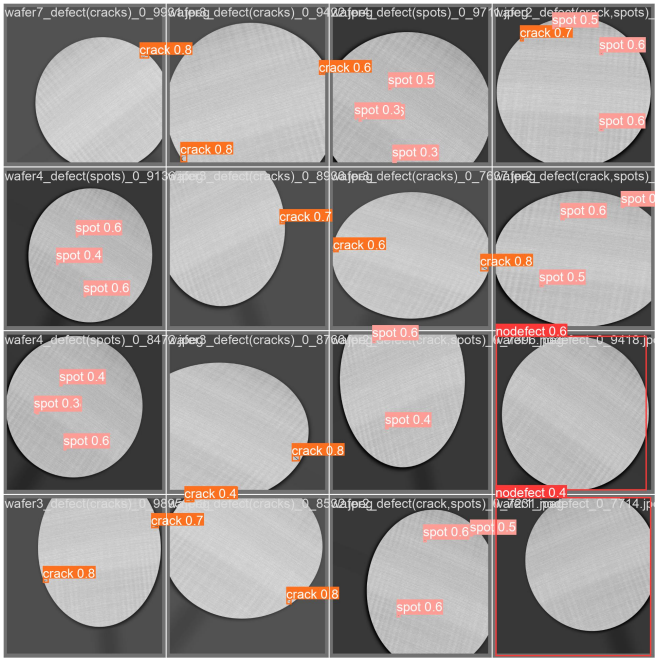

Fig 3. Classification of defects into different classes

4.5 Result of the project

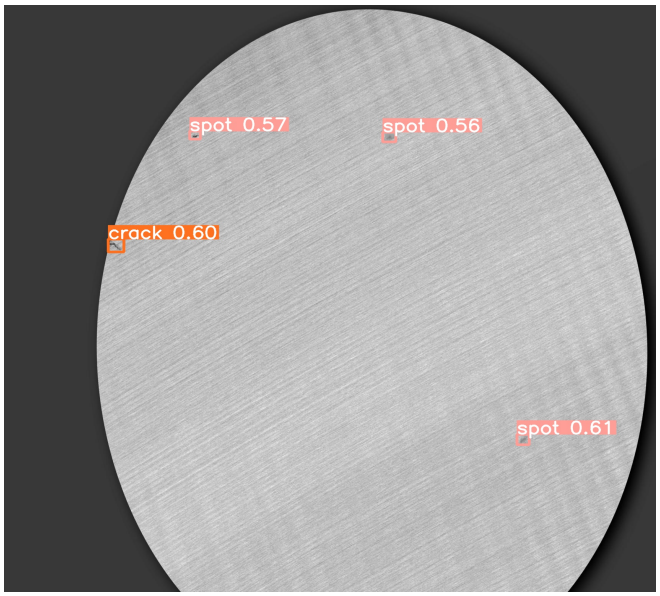

Fig4. Defect detection with displaying confidence interval

4.6 Result analysis graph of training and testing dataset:

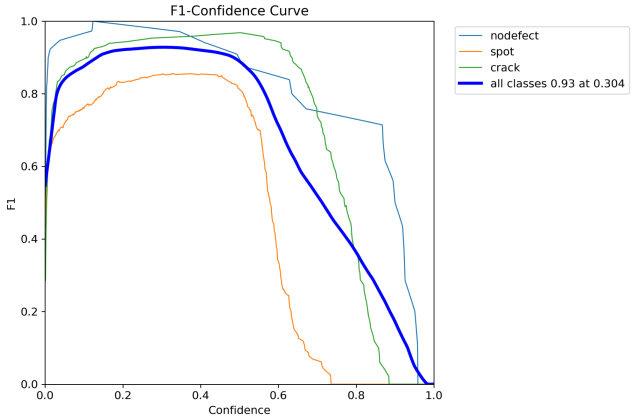

Fig5. F1 - Confidence Curve

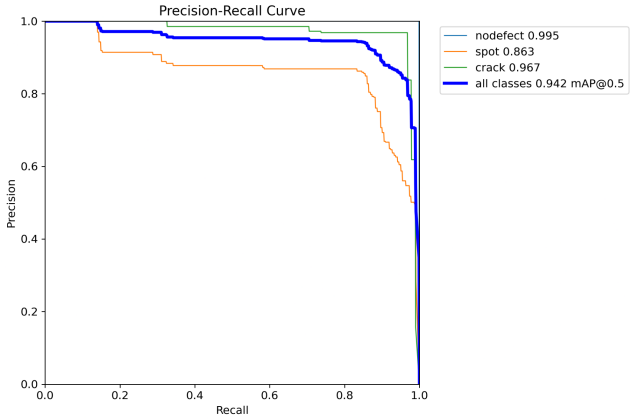

Fig6. Precision-Recall curve

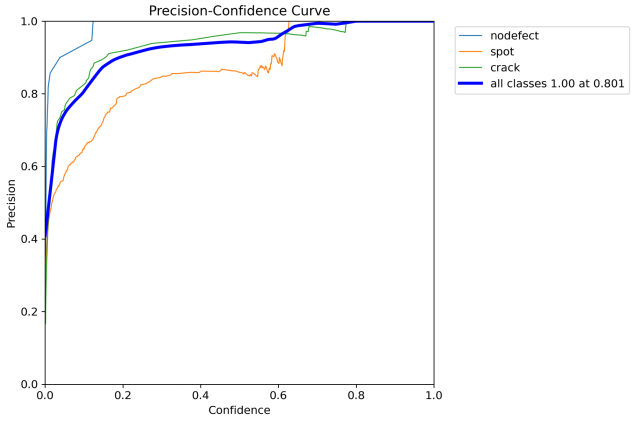

Fig7. Precision-Confidence curve

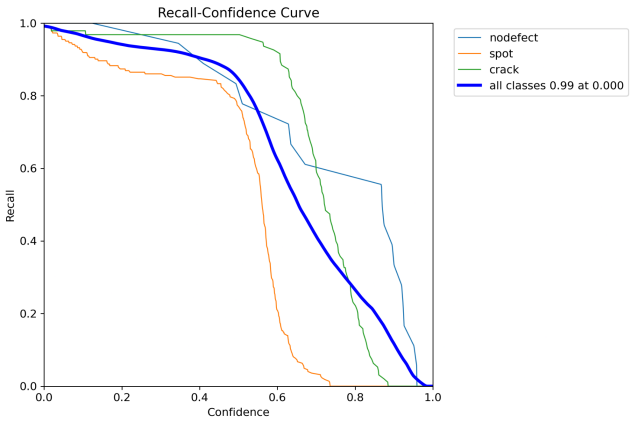

Fig8. Recall-Confidence curve

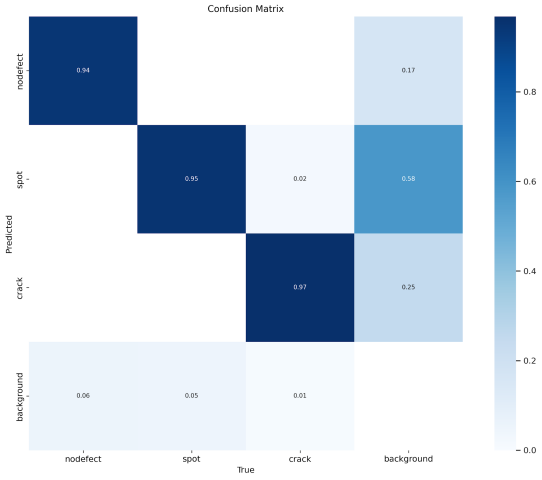

Fig9. Confusion Matrix

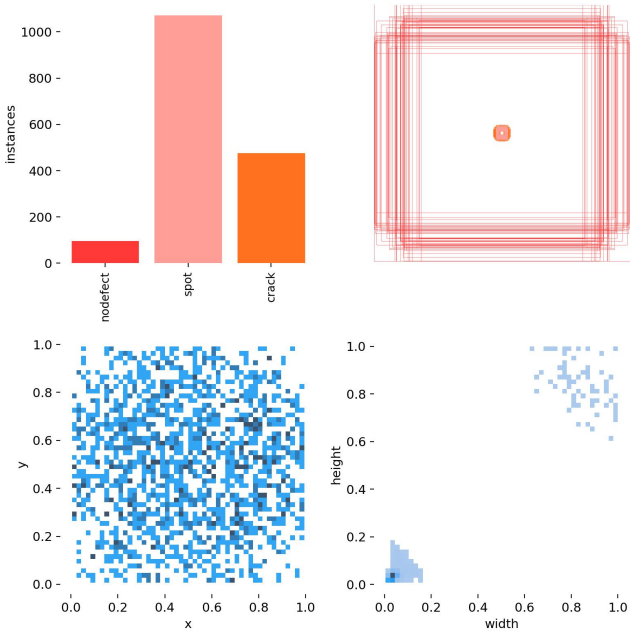

Fig10. Instances vs classes

V. Conclusion

This project has implemented a deep learning method that is YOLOv8 for the automated detection and classification of defects in semiconductor wafers. The approach focused on the classification of defects that most significantly damaged the wafer during the manufacturing process. The defect classification is done in 3 classes that are allspot, crack and no defect. We have divided the dataset in 70 percent of training data, 20 percent of validation dataset, and 10 percent of testing dataset. The four-class approach achieved an overall accuracy of 96 percent, which can be further improved by increasing the training dataset. The larger the dataset the better the accuracy.

VI. Future Scope

Streamlined manufacturing process can be done to make it possible to automate the entire semiconductor wafer manufacturing process, reducing the requirement for human intervention and boosting productivity. Improved defect detection, deep Learning algorithms can be trained on extensive datasets to detect even the slightest defects in wafers that may be undetectable through human inspection. This can lead to greater yields and less waste in the manufacturing process. Real-time monitoring of the manufacturing process and prediction of potential defects can be achieved with the use of Deep Learning algorithms. This could lead to enhanced efficiency and decreased downtime. Also, measuring the length of the crack to show that the defect is present with the length given can be thought of as a future work.

ACKNOWLEDGMENT

We would like to express our sincere appreciation to Vishwakarma Institute of Technology (VIT), Pune, for providing us with an excellent platform to undertake this project. We extend our gratitude to our esteemed Director, Prof. Dr. Rajesh M. Jalnekar, for his invaluable guidance and inspiration throughout the project. Additionally, we would like to thank the Head of the Computer Engineering Department, Prof. (Dr.) Sandeep Shinde, for his

unwavering support and assistance and Prof. Rakhi Bharadwaj, our project guide, for providing us with valuable support and guidance throughout the project and in writing this paper.

REFERENCES

[1] B. Alexey, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” in arXiv, arXiv:2004.10934, Apr. 2020.

[2] I-H. Yeh, C.-Y. Wang, H-Y. M. Liao, Y. -H. Wu and P.-Y. Chen, J.-W. Hsieh, “CSPNET: A NEW BACKBONE THAT CAN ENHANCE LEARNING CAPABILITY OF CNN,” in arXiv, arXiv:1911.11929, Nov. 2019

[3] I K. He, X. Zhang and S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep ConvolutionaL Networks for Visual Recognition,” in arXiv, arXiv:1406.4729, Apr. 2015.

[4] Yang Boyu, ”Design and Implementation of Hospital Fire Escape Guidance System Based on Image Recognition Indoor Positioning Technology: A Case Study of Oriental Memorial Hospital,” Master of Information and Communication Engineering, Department of Communication Engineering, Oriental Institute of Technology, 2021

[5] Krishna, Sajja Tulasi, and Hemantha Kumar Kalluri. ”Deep learning and transfer learning approaches for image classification.” International Journal of Recent Technology and Engineering (IJRTE) 7.5S4 (2019): 427-432.

[6] Huang, L., Y. Zhou, and L. Zhang, “Automated Defect Detection and Classification on Wafer Fabrication Using Machine Learning", Journal of Sensors, 2021.

[7] Kim, H., et al. , “Real-Time Defect Detection on Semiconductor Wafer Using Machine Learning-Based Approaches", IEEE Transactions on Industrial Informatics, 2018.

[8] Ng, K. C., et al., "Defect Detection on Semiconductor Wafers Using Convolutional Neural Networks", IEEE Transactions on Semiconductor Manufacturing, 2017.

[9] Zhang, T., et al., "Deep Learning-Based Defect Detection for Semiconductor Wafer Inspection", IEEE Transactions on Industrial Electronics, 2019.

[10] Wang, W., et al., "Automatic Defect Detection in Semiconductor Wafer Images Using Deep Convolutional Neural Networks", Journal of Electronic Testing, 2020.