SEMI3D: Deep Learning Based Defect Finder using Design Data

R M Sayi,Tummalapalli J V
sayit@stanford.edu

Sedhu Madhavan Thirumalaisamy
sedhu@stanford.edu

***Abstract*— Wafer Inspection systems help to identify the defects on the wafer which could eventually lead to failure of the chip. These systems are used in the production cycles of chip manufacturing. Traditionally defects are found by doing Die-Die comparison of adjacent dies to find the abnormalities within a die. But this approach is getting challenged with new advancements in Chip manufacturing. The underlined assumption of all dies in a wafer are identical is no more valid for many of the new use cases. However, design information of each Die that is used during manufacturing is available. We propose a deep supervised model to find the defects on Die based on that design information. This has several steps in it. We propose a Deep network that can convert Binary design image that has polygons into a rendered optical image. Using that rendered optical image and real optical image of dies which is collected using KLA High resolution Inspection tool, we propose deep supervised model that can detect the defects on the optical image of the die.**

Keywords— Semiconductor domain, Defect Finder, Pix2Pix, Faster R-CNN, Normalized cross correlation

I. INTRODUCTION

Wafer Inspection plays a major role in chip manufacturing process. Chip manufacturing involves pipeline of hundreds of steps that involves Lithography, etching and patterning etc. These increase the possibilities of defects on chips as well. Any process variation step can cause defects and non-early detection of these defects can potentially cause very less yield and incurring huge losses for chip manufacturers. Hence, it's very important to detect the defects and their sources. With many advancements in the semiconductor industry, the manufacturing process continuously evolves and poses challenges in finding the defects caused by them. Traditionally, defects are found by comparing adjacent dies. But, with evolution of the process, this assumption is becoming weak and getting challenged. Also, it's not viable solution to come up with new detection algorithm with every new process variation.

In this paper, we propose a novel idea for defect detection based the design template information of the Die itself. Chip manufacturing needs design data as source of the input. As shown in the **Error! Reference source not found.** design image can be obtained from design template of the die and the optical image is the real image of die obtained from KLA high resolution inspection tool. We propose to use the design images and optical images and develop a neural network that can generate ideal rendered optical image of the die. Using this, we propose another deep neural network that can find the defects on the optical image of die based on its rendered optical image.

Rest of the paper is organized as follows. Section II describes the related work that is done for Image generation, object detection and defect detection. Section III explains about the dataset and pre-processing of the data. The model is described in section IV , the Results/Experiments in section V, conclusion/Future work in section VI and contributions in section 7.

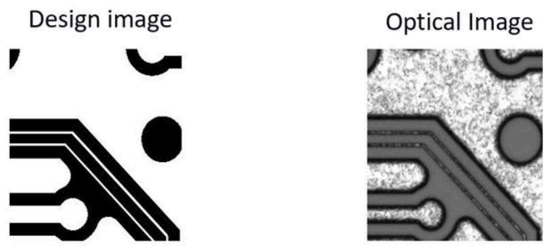

Figure 1: Design Image vs Optical Image

II. RELATED WORK

Of late, lot of research is done in the area of Generative Adversarial Networks (GAN) for Image generation and transformation. Philip et al [1] were the first ones to propose C-GANs called Pix2Pix for Image to Image translation from one domain to other. Limitation of the Pix2Pix Gan is that it needs paired examples for training the network. Jun-Yan Zhu et al [2] proposed CYCLE-GAN that can translate an image from one domain to another in the absence of paired examples. However, it can only translate an image from one domain to other. Younjey Choi et al [3] proposed Star-GAN that can translate an image from one domain to multiple target domains.

Similarly, there is lot of research that is done for the object detection as well. R-CNN proposed by Joseph et al [4] was one of the notable deep learning networks for object detection. Faster R-CNN proposed by Shao qing et al [5] was another notable advancement on R-CNNs where region proposal is directly in the network. Jose et al [6] proposed YOLO which became popular network for object detection.

There is research done in anomaly and defect detection on surfaces. Some of them are in [7],[8]and [9]. But these address defect detection on the un-patterned surfaces. The challenge with pattern surface is network needs to know what the pattern is and what is the defect and pattern varies across the die. There is no known work done for patterned wafers defect detection using the Deep learning and design information.

This paper addresses that gap by combining image transformation and object detection. We choose Pix2Pix Gan for Image transformation due to its simplicity and effectiveness for our use case and we choose Faster R-CNN over YOLO for better accuracy for smaller defects.

III. DATSET AND FEATURES

Dataset was collected using multiple customer wafers and their design data. Below is the process we followed for dataset collection

For each wafer,

- obtain the design information for each die of the wafer
- Construct the binary image of the die using the design information
- For each of these binary images, obtain two sets of real optical image of the die using KLA's proprietary high-resolution wafer inspection tool. First set containing the optical image of dies without defects and second set containing optical images of dies with the defects

This process is repeated on several customer wafers and obtained dataset of size 500.

This data set is further divided into train set (400), dev set (50) and test set (50)

IV. METHODS

Image generation and object detection are two separate research areas in which lot of work has been done. Our proposal is to come up with Novel idea that combines best of both the worlds to do the defect detection.

Rest of this section is divided into following sub-sections. Section A describes the overall model and training steps for the detection, Section B explains how to use the model for inference, section C and D details out the rendered image generation and defect detection respectively.

A. *Model*

Our Model is summarized in Figure 2. Below are the steps we follow for training the model

- Design information of the Die(chip) is available as text file that contains the details of polygons
- Convert that design file to Binary image.
- Collect optical image of the Die without the defects using KLA proprietary high-resolution Inspection tool
- Train Pix2Pix network using the real optical image to convert the binary image to rendered optical image
- Obtain the rendered optical image from design file using Inference on Pix2Pix trained network
- Train Faster R-CNN with real optical images with defects and their corresponding rendered optical images (obtained from pix2pix Gan) to detect the defects and to draw the bounding boxes

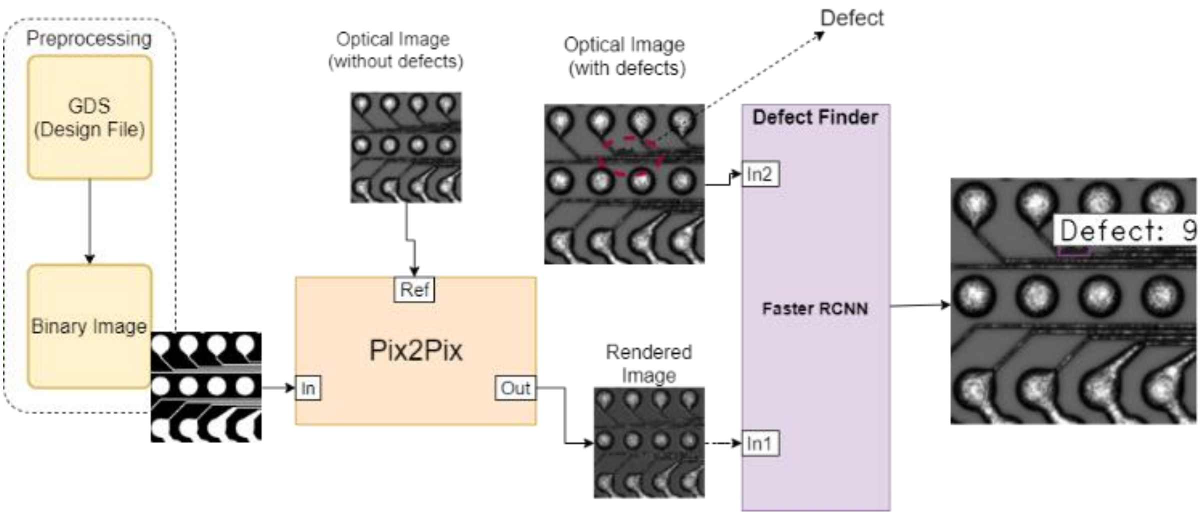

Figure 2: End to End Model for Defect Detection

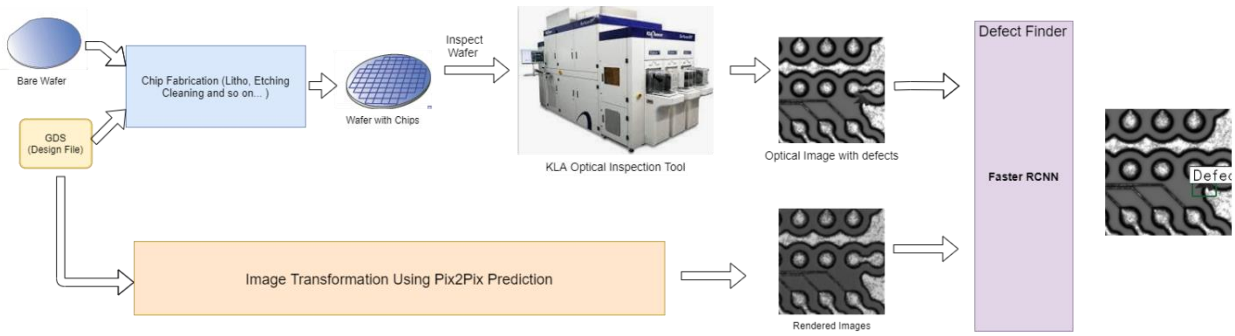

Figure 3; Defect Detection Flow using the Deep Learning model

B. Model Inference

Once the model is trained as mentioned in section A, we can use the model for anomaly/defect detection by following the steps mentioned in Figure 3. Same is summarized below.

- Convert the design file to Binary image.
- Obtain rendered optical image from binary image using inference on Pix2Pix network
- Collect the real optical images of the dies across the wafer with KLA proprietary high-resolution Inspection tool
- Feed the real optical image and rendered optical image to the Faster-RCNN to find the anomalies or defects on the dies of the wafer

C. Rendered optical Image generation

We use pix2pix Gan to generate rendered optical image from the real image. Below are the hyper parameters of the network that we changed.

- Input - Binary image
- Output - Rendered optical image.
- Train set – 400 images
- Dev set – 50
- Test set - 50
- Number of epochs – 800
- Batch size – 32

We used same learning rate and loss function as given in the pix2pix paper [1].

D. Defect Detection

For defect detection, we use Faster R-CNN network as shown in Figure 4. Below are the hyper parameters that we changed.

- Input - Rendered image and optical image
- Output – Optical image with bounding box drawn on defects
- Number of epochs – 500
- Train set – 400
- Dev set – 50
- Test set – 50
- Anchor box scale – 32,64,128 pixels
- Anchor box ratio – [1,1], [1,2], [2,1]
- Pretrained conv layer – First 5 layers of VGG
- Concat layer is shared by both RPN and ROI pooling layers

Network is trained to learn the difference between the rendered optical image and real optical image, detect the defects and draw bounding boxes.

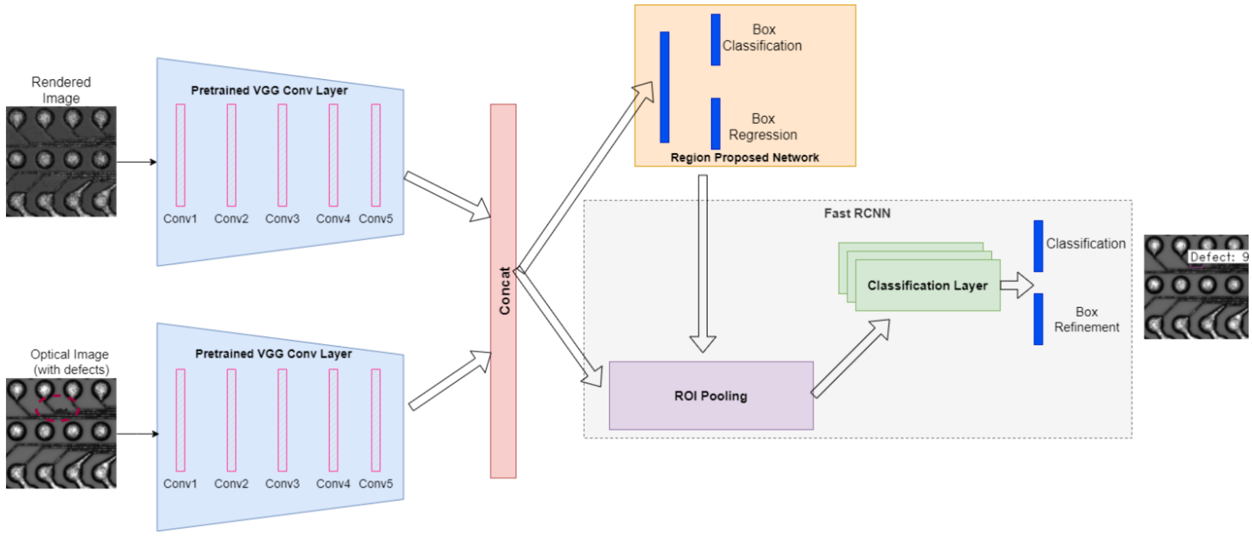

Figure 4: Faster RCNN for Defect Detection

V. Experiments and Results

A. Experiments and Network Standardization

The following experiments were performed for pix2pix Gan for rendered optical image generation

- We Started training Pix2Pix Gan with binary image and optical image pair
- After training it with 400 set of pairs, we ran the inference randomly on few images from the dev set.
- We could see visibly that some of the images inference didn't give output which is close to real image as shown in the Figure 5

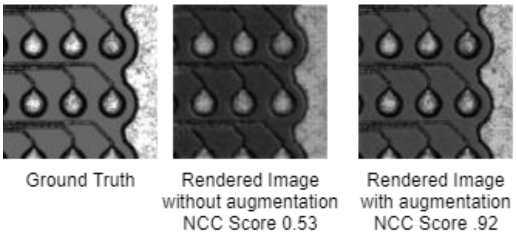

Figure 5: High bias problem in Pix2Pix GAN

- We then ran inference on entire dev set, collected output images and performed Normalized cross correlation (NCC) on the output images(y^) and real optical images(y) and considered average of that as accuracy metric.
- NCC score on Dev set was .53.
- After individually looking at several of the images which had low NCC score, we realized that train set didn't have enough samples that could cover the examples of Dev set and also background was similar for most of the training set examples which is contributing to the noise in the image as shown in Figure 5. We ran into High-Bias problem.
- We reshuffled the data set and generated new train, dev and test sets and repeated the same again. It didn't improve the score much. Then we did data augmentation for the set that had less examples. We created more examples by removing some of the polygons and by shuffling polygons in them.
- We then repeated same exercise and we got average NCC score as .97
- We repeated same exercise with test set and NCC score was .92. Figure 6 shows the rendered optical image obtained from pix2pix trained network for different dies.
- Figure 8 shows how Pix2Pix Gan has learned to generate image over the number of epochs. As shown in the figure, NCC score has improved over the number of epochs and we can confirm the same by visibly observing the rendered images across the number of epochs.

Once Pix2Pix GAN is trained, we generated rendered optical images using that and trained Faster R-CNN with rendered and real optical images for the defect detection.

Below are the observations

- As shown in the Figure 7, all four losses of the Faster-R CNN – RPN classification loss, RPN regression, detection classification loss and detection regression loss have started converging from 400 epochs for our training set of 400 image pairs.
- VGG net, Inception and Res Net are the options for the encoding layer in Faster R-CNN. We tried all and all of them gave similar results for our use case. Hence, for simplicity we chose VGG Net.
- For gray scale images, we have tried with 3 channels - one being rendered optical image from Pix2Pix Gan and other two are optical image with defects. Our model was able to capture the defects even with this input. Hence, for gray scale images we can use 3 channels as mentioned above as input.

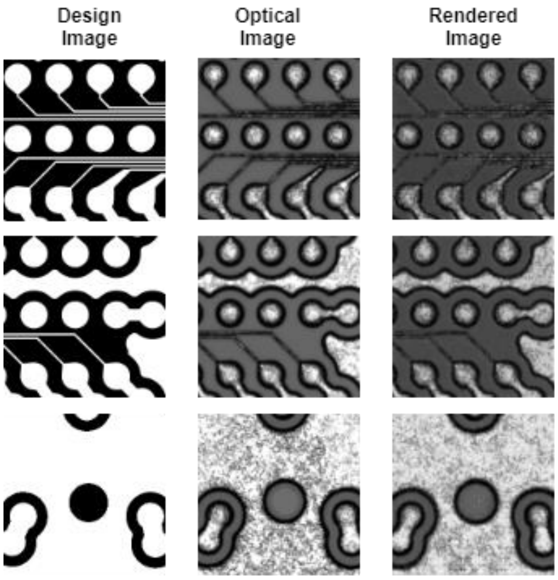

Figure 6: Rendered optical Image generation from pix2pix GAN

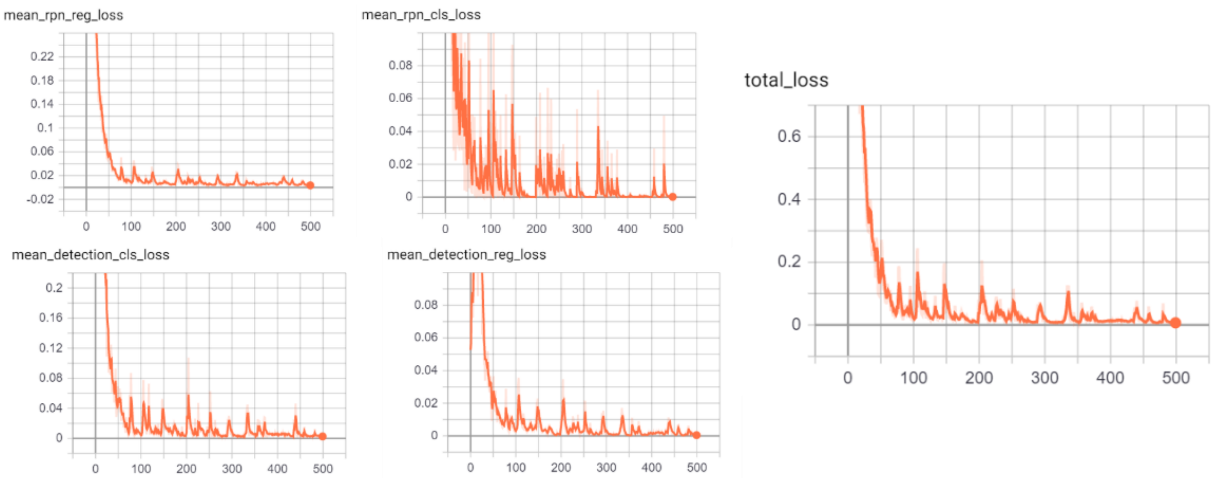

Figure 7: Loss curves from Faster R-CNN Training

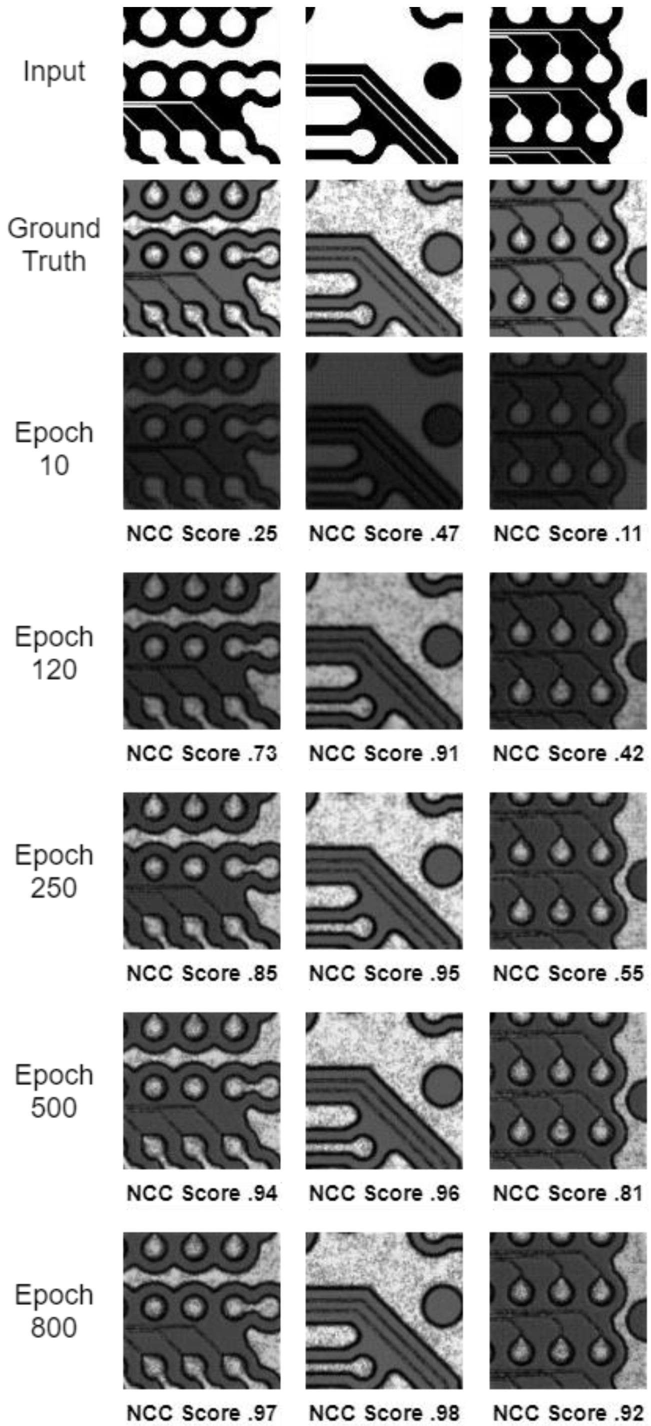

Figure 8: NCC score improvement over the Epochs

Below are the results from our experiments on Faster RCNN network

- Training accuracy – 85%
- Test accuracy – 81%
- No false positives but all the accuracy errors are from false negatives

Figure 9 shows the inference from Trained Faster RCNN network. As shown in the figure, the network is able to detect the defects based on the difference between rendered and optical image.

The test set had examples with defects of different shape, dimension, and locations. It also had examples with different die surface and patterns. Network could detect all these variations with high accuracy.

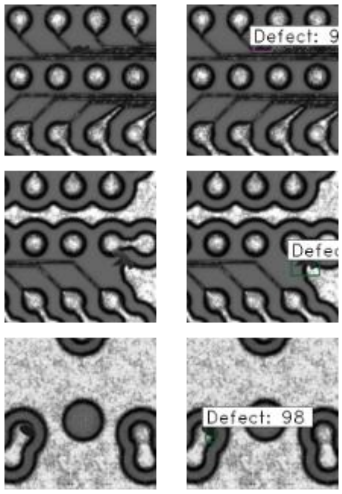

Figure 9: Defect Detection using Faster R-CNN

VI. CONCLUSIONS AND FUTURE WORK

By using image transformation and object detection techniques, Pix2Pix Gan and Faster RCNN we showed that defect detection on the optical image is possible with its design template alone.

This approach shall form a solid baseline for next generation defect detection techniques in semiconductor domain. With the limited number of samples of die images of specific material layer itself, we were able to achieve our goal of defect detection. This proves our hypothesis and our approach.

This work can be extended to multi-layer design templates and to chips of different materials and patterns. Other challenges like suppressing noise from the optical sources shall also be considered further.

VII. CONTRIBUTIONS

Sayi Tummalapalli worked on pre-processing of the data, generating binary image from the design files and training and optimizing Pix2Pix GAN for rendered optical image generation

Sedhu Madhavan worked on setting up environment with GPUS for training the models, data collection of real optical images using KLA high resolution Inspection tool, training and optimizing Faster R-CNN for defect detection on optical images of the die.

ACKNOWLEDGMENT

We would like to acknowledge Pavan Kumar and Gong Jie for their valuable inputs and directions and our mentor Shahab Mousavi for technical guidance and support

REFERENCES

[1] Phillip Isola, Jun-Yan Zhu Image-to-Image Translation with Conditional Adversarial Networks - 10.1109/CVPR.2017.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros," Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", arXiv:1703.10593

[3] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo,"StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation" - IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8789-8797

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik," Rich feature hierarchies for accurate object detection and semantic segmentation", arXiv:1311.2524

[5] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun," Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", arXiv:1506.01497

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi," You Only Look Once: Unified, Real-Time Object Detection", arXiv:1506.02640

[7] Domen Tabernik, Samo Šela, Jure Skvarč, Danijel Skočaj," Segmentation-Based Deep-Learning Approach for Surface-Defect Detection", arXiv:1903.08536

[8] Manpreet Singh Minhas, John Zelek," Semi-supervised Anomaly Detection using AutoEncoders", arXiv:2001.03674

[9] Jakob Božič, Domen Tabernik, Danijel Skočaj,R. Nicole, "End-to-end training of a two-stage neural network for defect detection," arXiv:2007.0767

[10] Collin Arnett, YoungJin Kim,"Keras Implementation of Faster R-CNN", https://github.com/you359/Keras-FasterRCNN