Neural Anomaly Transfer for Wafer Inspection Systems

Karthikeyan Kalaiarasan kkarthi@stanford.edu
Nagendra Kamalapuri theknp@stanford.edu
Songyang Yu syyu2018@stanford.edu
source code link: https://github.com/kkarthi002/Neural-Anomaly-Transfer
KLA-Tencor Corporation

Abstract

In Semiconductor industry, increase in wafer inspection throughput gives better cost of ownership for our customers. Traditionally our wafer inspection systems are a 2-Pass system where we inspect in lower resolution(LR) at a faster rate and the identified defect candidates are re-captured at a higher resolution(HR) which is an inevitable throughput killer. Towards removing the process of recapturing HR images, we present a solution that combines SRGAN and Neural Style transfer by providing references with a HR Golden Image(HRGI) generated during initial setup. We have designed an end to end trainable deep learning model that can generate the HR image with the defect from both LR image and the HRGI.

Introduction:

Wafer defect inspection system detects physical defects (foreign substances called particles) and pattern defects on wafers and obtains the position coordinates (X, Y) of the defects. Defects can be classified into random defects and systematic defects. Random defects are mainly caused by particles that become attached to a wafer surface, so their positions cannot be predicted. On the other hand, systematic defects are caused by the conditions of the mask and exposure process. In some cases, even if there is a valid defect on the wafer, it may not be of importance for the customer and hence will not affect the yield. As a result, it is very important for the customer to manually review all the defects to find out which are the Defects of Interest(DOI) and which are nuisances. Since the size of the defects are in the order of microns and end user needs HR images to validate whether it is a DOI or nuisance and fix the process to increase the yield.

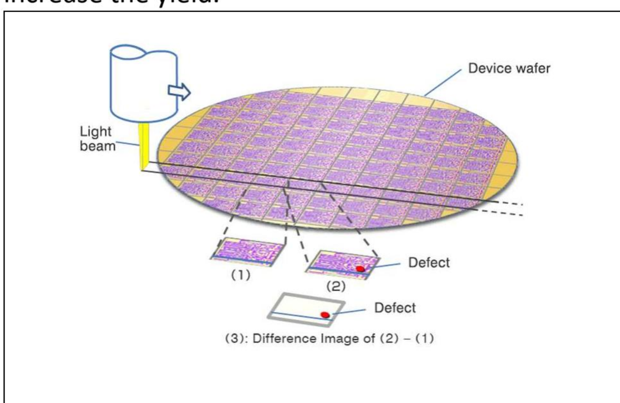

Figure 1. Silicon Wafer

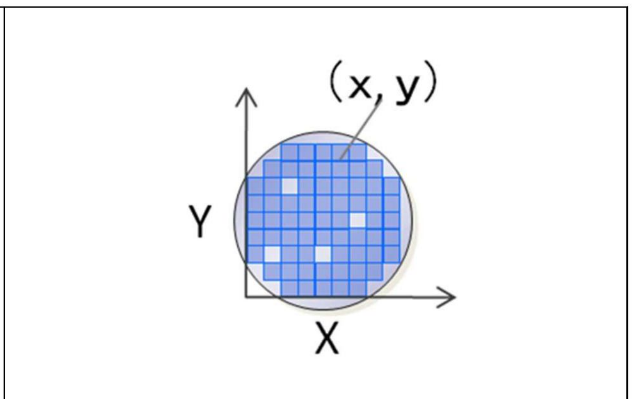

Figure 2. Defective map after inspection

In this project, we try to develop a deep learning model to create HR image (contains defect) without doing a second pass recapture as mentioned before.

2. Related Works

In recent years, deep learning based Single Image Super Resolution(SISR) has shown superior performance in visual quality compared to those non-deep-learning based methods [1][2][3]. After googling our problem statement, we found a research paper "Reference-Conditioned Super-Resolution by Neural Texture Transfer(SRNTT)" [4] that closely matched with what we were looking for. It adaptively transfers textures to the generated Super Resolution image conditioned on the reference image. In our case, we would like to extract and transfer the anomalies from the lower resolution to the higher resolution image using a reference HR Golden Image(HRGI)which doesn't contain the anomaly.

The actual source code for the network model was not made public and we tried to replicate most part of the model based on the details provided on the research paper. But due to the lack of details in the paper about some parts of the model, in particular, patch matching and texture swapping, we could not make the model working. So, instead, we then focus our project on a model based on the paper "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network (SRGAN)" [5].

3. Data Set

We used three different data sets.

1. We downloaded the dataset "Yahoo MirFlickr25K" to train with the SRGAN to get an understanding how Super Resolution model works. SRGAN network we used needed a minimum size of 384x384 pixels. We preprocessed the data to filter the images of the required size and used approx. 3500 images for training the model. 3300 images were used for training and 100 images each were used for Dev and Test. This was done for the mid-term milestone.

Figure 3. Yahoo MirFlickr25K sample data

2. We collected 1000 images related to our Standard Silicon Wafer referred to as "Standard Wafer Dataset" (SWD) going forward. Collected 500 images of (64x64) LR images using 4 microns/pixel (μ/p) using our Wafer inspection systems. We also collected the 500 (256x256) HR images using 1 (μ/p) at the same locations. We split the data 460 images each for LR and HR for the training and 20 images each for dev and test.

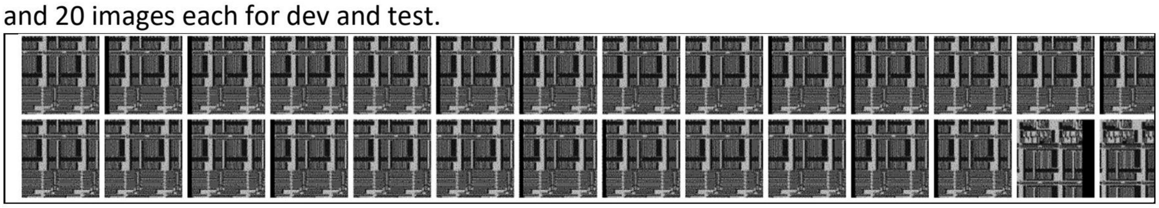

Figure 4. SWD sample images

3. We also collected 200 HR images and 200 LR images on the customer data. We used 180 images each for LR and HR for training and 20 images each for dev and test.

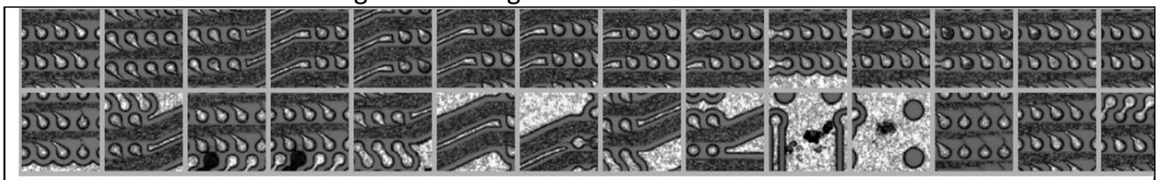

Figure 5. Customer sample images

4. Network Model

The architecture of the model of SRNTT [4] is shown in Figure 6. This is a complex model and we replicated a portion of the model as part of mid-term milestone for some of the modules. Since there are not enough details in the paper for some of the modules such as patch matching and texture swapping, we couldn't make the model working. We list them here for the completeness of the project since it part of our study process.

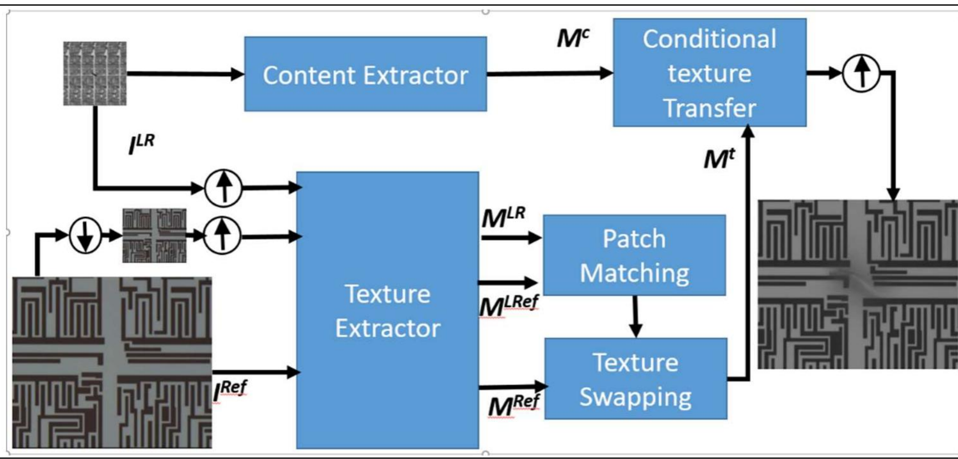

Figure 6. Overall structure of SRNTT

Content Extractor:

#	Layer name(s)	Output size
0	Input	H × W × 3
1	Conv, ReLU	H × W × 64
2-17	Residual blocks (Conv,BN, ReLU, Conv, BN)	H × W × 64
18	Conv, BN	H × W × 64
19	#1 + #18 Mc:	H × W × 64

Conditional texture Transfer

#	Layer name(s)	Output size
0	Input	Mc: H × W × 64 Mt: H × W × 256
1	Concatenate	H × W × 320
2	Conv, ReLU	H × W × 64
3-18	Residual blocks (Conv,BN, ReLU, Conv, BN)	H × W × 64
19	Conv, BN	H × W × 64
20	Mc + #19	H × W × 64

Upsample

#	Layer name(s)	Output size
0	Input	H × W × 64
1	Conv	H × W × 256
2	Sub-pixcel, ReLU	2H × 2W × 64
3	Conv	2H × 2W × 256
4	Sub-pixcel, ReLU	4H × 4W × 64
5	Conv, tanh	ISR : 4H × 4W × 3

Discriminator

#	Layer name(s)	Output size
0	Input	I^{HR} or I^{SR}: 160×160×3
1	Conv, BN, LReLU	160 × 160 × 32
2	Conv, BN, LReLU	80 × 80 × 32
3	Conv, BN, LReLU	80 × 80 × 64
4	Conv, BN, LReLU	40 × 40 × 64
5	Conv, BN, LReLU	40 × 40 × 128
6	Conv, BN, LReLU	20 × 20 × 128
7	Conv, BN, LReLU	20 × 20 × 256
8	Conv, BN, LReLU	10 × 10 × 256
9	Conv, BN, LReLU	10 × 10 × 512
10	Conv, BN, LReLU	5 × 5 × 512
11	Flatten	12800
12	FC, LReLU	1024
13	FC	1

The loss function of the SRGAN [5] model we used is defined as:

$$l^{SR} = \underbrace{\underbrace{l_X^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}}$$

Where is the content loss function contains two parts, MSE loss and VGG loss, which are defined as the following :

$$l_{MSE}^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} \left(I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y}\right)^2$$

$$l_{VGG/i,j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\emptyset_{i,j}(I^{HR})_{x,y} - \emptyset_{i,j} G_{\theta_G}(I^{LR})_{x,y}\right)^2$$

The adversarial loss function is defined for the whole training samples as:

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

5. Training

In initial training, we just tried with SRGAN to start with and we were amazed by the results. We did not have enough time to completely replicate the SRNTT network as per the reference paper. So, we continued to use SRGAN for our project. The LR images are obtained through our inspection systems are of the size 64 × 64. The corresponding reference is fed with the HR size, 256×256.

Training was performed on "Yahoo MirFlickr25K" for a total of 2000 epochs with a mini-batch of 16 using the Adam optimization algorithm. A learning rate of $1e^{-3}$ was used for the first 100 epochs, and the learning rate is decayed by 0.1 for each 100 epochs. Weights were initialized with random normal distribution initializer with standard deviation 0.02. From SWD dataset we added 460 images each of LR and HR and added it to the training set and initialized the network weights from the previous training. We trained for a total of 200 epochs with the additional input data. Further we added the customer data set images to the training set and trained the model for additional 200 epochs

6. Results and Remarks

After the first training with Yahoo Flickr data, we could generate HR images from the LR images. we evaluated the model with the SWD data set and customer data set to see the results. It worked very well on the customer data even though it did not see the images as part of the training data set whereas SWD data set performed very poorly which we can see from the images below. We further trained the network by adding input images from both the data set we could get very good HR images. Initially the network performed well on the test data generating the HR images for the test LR images. We got excited and collected 100 more images at a different location of the wafer but the result was totally wrong. The images we originally collected were from a location that only covered portions of the die and the network performed well only on those images from that portions of the die. We collected 200 more images that covered all the die locations and then retrained the network to get a better result.

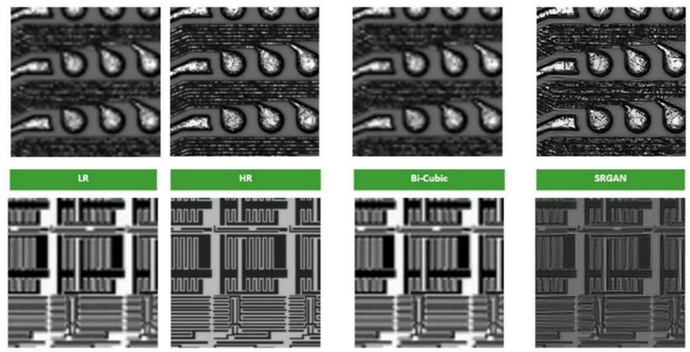

Figure 7. SWD data(bottom) and customer data(top) evaluated on "Yahoo Flickr25K" data

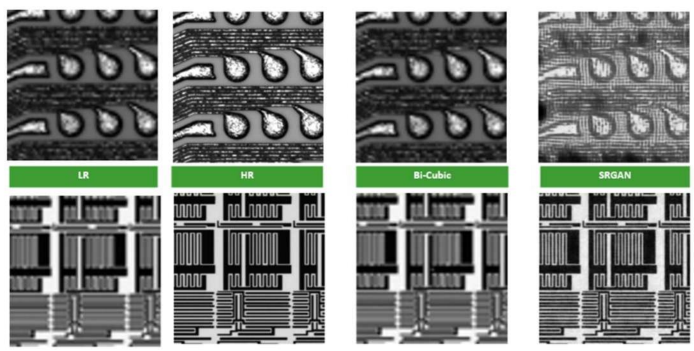

Figure 8. SWD data(bottom) and customer data (top). Customer data evaluated on a trained model containing SWD only. We could see the SWD style on customer images.

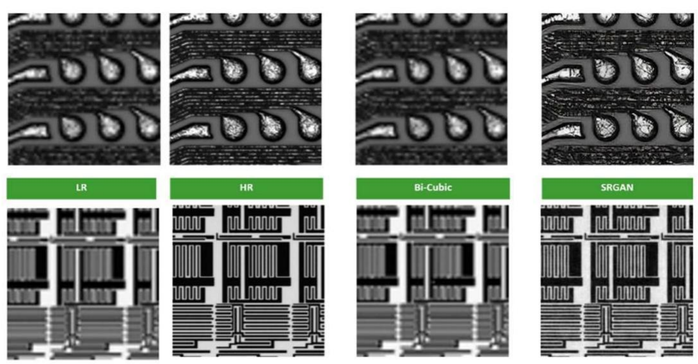

Figure 9. Customer data(top) and SWD (bottom) is generated using SRGAN

We started with the goal of using a deep learning model to work for different types of wafers. With this current approach, we must retrain the network for different type of wafers. We still believe that SRNTT [4] will help us achieve our goal and we will continue to work on the model which we shelved after mid-term milestone due to time constraints.

7. Acknowledgement

A special shout out to Daniel Paul Kunin our TA for guiding us throughout the project.

Reference:

1. Learning a deep convolutional network for image super-resolution, Dong C., Loy, C.C, He, K., Tang, X., European Conference on Computer Vision. Pp.184-199, Springer (2014)
2. Deeply-recursive convolutional network for image super-resolution. Kim, J., Kwon Lee, J., Mu Lee, K., Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1637-1645 (2016)
3. Enhanced deep residual networks for single image super-resolution, Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., IEEE Conference on Computer Vision and Pattern Recognition Workshops, July 2017
4. Reference-Conditioned Super-Resolution by Neural Texture Transfer, Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi, arXiv:1804.03360
5. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, arXiv:1609.04802