Neural Anomaly Transfer for Wafer Inspection Systems

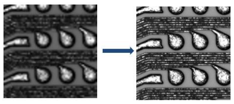
Nagendra Kamalapuri, Songyang Yu, Karthikeyan Kalaiarasan

Fall18 CS230 Final Project
Mentor: Daniel Paul Kunin
{theknp, syyu2018, kkarthi} @Stanford.edu

Stanford University

MOTIVATION

- 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 want to design an end to end trainable deep learning model that can generate the HR image with the defect from both LR image and the HRGI

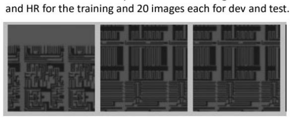


LR HR

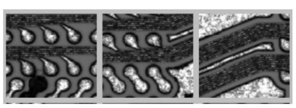
DATA SET

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

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.



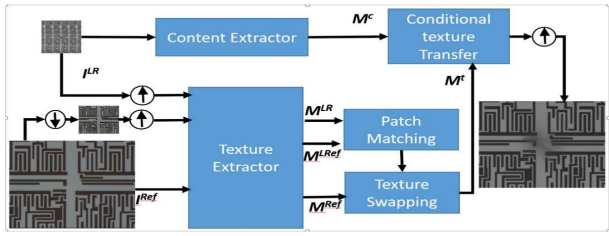
3. 3.We also collected 200 HR images and 200 LR images on the customer data. We used 180 images each for LR and HR For the training and 20 images each for dev and test.



NETWORK MODEL

Architecture:

We use the same network model proposed in [1] as shown below. This is a complex model and there are not enough details in the paper for some of the modules for Patch Matching and Texture swapping. We are learning more and more by the day as we continue to break apart the model module by module.



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

Hyper-parameter	Value
Epoch	2000(public data) , 200(SWD),200(CWD)
Batch Size	16
Weight Decay	0.1
Learning Rate	0.0001

Training:

- Initial training we just tried with SRGAN[2] to start with and we were amazed by the results. We did not have enough time to completely replicate the 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

RESULTS

- After the first training with Yahoo Flickr data, we were able to 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 were able to get very good HR images with MSE <Fix The Validation>.
- 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.

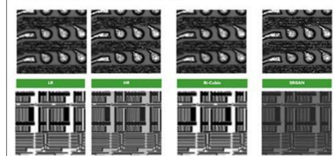


Figure: SWD data(bottom) and customer data(top) evaluated on "Yahoo Flickr25K" data

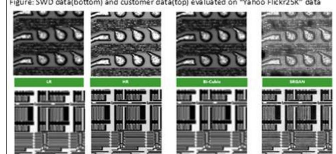


Figure: SWD data(bottom) and customer data (top). Customer data evaluated on a trained model containing SWD only. We could see the SWD style on customer images.

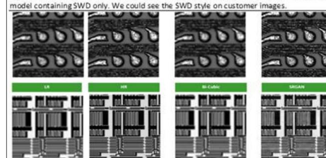


Figure: Customer data(top) and SWD (bottom) is generated using SRGAN

DISCUSSION & FUTURE

- 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 (1) 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.

Presentation video: https://youtu.be/HkZEJODl7Yo

References: [1] Reference-Conditioned Super-Resolution by Neural Texture Transfer, Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi, arXiv:1804.03360v1, 2018. [2] Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,C. Ledig et al., CVPR, 2017