Five Deep Learning Recipes for the Mask-making Industry

Ajay Baranwal*[a], Mike Meyer[a], Thang Nguyen[a], Suhas Pillai[a], Noriaki Nakayamada[b], Mikael Wahlsten[c], Aki Fujimura[d], Mariusz Niewczas[d], Michael Pomerantsev[d]
[a]Center for Deep Learning in Electronics Manufacturing, Inc. (cdle.ai);
[b]NuFlare Technology, Inc.; [c]Mycronic AB; [d]D2S, Inc.

ABSTRACT

Deep learning has an increasing impact on our personal and professional lives. Deep learning has the potential to transform mask, semiconductor and electronics manufacturing. This paper reviews key results from the Center for Deep Learning in Electronics Manufacturing's (CDLe's) first year of operation. We consider results from adapting five common types of deep learning recipes to solve key challenges in the manufacture of photomasks, printed circuit boards (PCBs), and flat panel displays (FPDs). These deep learning applications include 1) grouping similar items to automatically categorize mask rule errors; 2) using U-Net architecture to construct fast mask designs; 3) using vision-based object classification to find and classify pick-and-place (PnP) errors on PCB assembly lines; 4) using anomaly detection to improve the quality of FPDs; and 5) using digital twins to create SEM images and optimize Inverse Lithography Technology (ILT). While we compare the relative benefits of these techniques, all show the importance of data to improve the success of deep learning networks and of electronics manufacturing. These applications rely on varying neural network architectures such as autoencoders, segmentation networks, deep convolutional networks, anomaly detection, and generative adversarial networks (GANs).

Keywords: photomask, deep learning, mask simulation, lithography hotspot, mask defect categorization, fault detection

1. INTRODUCTION

Deep learning (DL) has solved problems in a wide range of industries—retail, IT, medical, pharmaceuticals, biotechnological, and autonomous driving to name just a few. Likewise, deep learning recipes for recommendation, segmentation, classification, anomaly detection and digital modeling are highly relevant to the manufacture of photomasks, printed circuit boards (PCBs) and flat panel displays (FPDs). Photomask shops face challenges with mask inspection, as well as detecting and classifying hotspots, faults and defects that impede production. Deep learning has the potential to solve these challenges before they become real problems on the assembly line. Digital twins that model the properties, conditions and attributes of real-world counterparts in electronics manufacturing have significant advantages over real data in simulating the behavior of the system. Digital twins allow us to observe, reproduce, and find faults in the system at a software level, long before they stop or slow down an assembly line.

During the past year, CDLe, with its member companies,[1] has used supervised, unsupervised and reinforcement learning methods to address several mask shop problems. We have worked on more than ten deep learning-based projects with encouraging results for the industry. For this paper, we have chosen five illustrative examples of how DL techniques, or recipes, can effectively solve problems in the photomask industry. For each recipe, we first explain how other industries have

[1] D2S, Inc.; Mycronic AB; and NuFlare Technology, Inc.

Photomask Technology 2019, edited by Jed H. Rankin, Moshe E. Preil, Proc. of SPIE Vol. 11148, 1114809 · © 2019 SPIE · CCC code: 0277-786X/19/$21 · doi: 10.1117/12.2538440

benefited from similar DL techniques, then map out a problem suited to the DL technique in the mask industry, and finally explain how we used the technique to solve the problem. In section seven, we share what we have learned from these case studies. Section eight concludes with our advice on how the industry might benefit most from deep learning.

1.1 Background

The type of problems deep learning can solve include natural language understanding to extract meaningful information from text documents and information retrieval and language translation like Google Translate. In the speech domain, DL has shown tremendous progress in automatic speech recognition, text-to-speech and realistic-speech generation like WaveNet [9]. Related to computer vision, DL offers effective solutions for a multitude of problems, such as detecting objects, segmenting objects in MRI scans, denoising images, extracting text from images, performing image-based searches, improving the quality of images and even creating new images. DL has introduced advancements in finding anomalies in the form of outliers, by learning the accurate distribution of normal data, so that DL can flag any anomalous data. DL even has the ability to help build digital twins to simulate physical environments.

1.2 Semiconductor Manufacturing Problems

Many of the problems in photomask industries such as conventional optical proximity correction (OPC), inverse lithography technology (ILT), lithography hotspot detection, fault detection and classification, automatic mask defect classification and diagnostics, and SEM denoising and contour extraction can benefit from deep learning. We have found effective DL solutions for these photomask industry problems [10]. Sections two through six of this paper focus on five such recipes.

2. RECIPE 1: AUTOENCODING TO AUTOMATICALLY CATEGORIZE MASK RULE ERRORS

2.1 Motivation: Recommendation Systems

An increasing number of companies like Amazon and Netflix are embracing recommendation systems that are based on customers' browsing history [1]. Many of these recommendation systems require learning to distinguish similar items—retail products or movies—to predict which items will have future relevance for users. The need to find like items inspired us to use a similar approach to categorize mask design rule errors.

2.2 Automatic Categorization of Mask Rule Errors

Aspects of recommendation systems where similar items are grouped together to recommend new ones also apply to grouping like design rule errors produced by photomask check tools.

Photomask design rule checking, particularly when applied to anticipated curvilinear mask shapes after simulation, may generate thousands of Edge Placement Errors (EPEs). Because so many similar spacing errors are generated, it is more beneficial to report and analyze errors of similar magnitude together. We found that DL can group together spacing errors with similar surrounding geometries within an interaction region [Fig-3]. Traditionally, this has been done using fuzzy pattern matching. Using a deep learning approach called autoencoder combined with unsupervised clustering, error-grouping tasks are solved faster and more accurately than with fuzzy pattern matching.

2.3 Autoencoders

Autoencoder is a type of neural network that learns a numerical representation (also called encoding) of input data as a series of vectors. The purpose of autoencoders is to learn low-dimensional numerical representations of high-dimensional data. An autoencoder consists of two neural networks—encoding and decoding networks. After sufficiently training the autoencoder, the encoder network can successfully encode input data to a lower dimensional vector representation. To cluster mask clips with similar errors, we use deep autoencoders on input data [Fig-1] to obtain reasonably accurate encoding vectors that define the shapes and preserve similarity information for grouping.

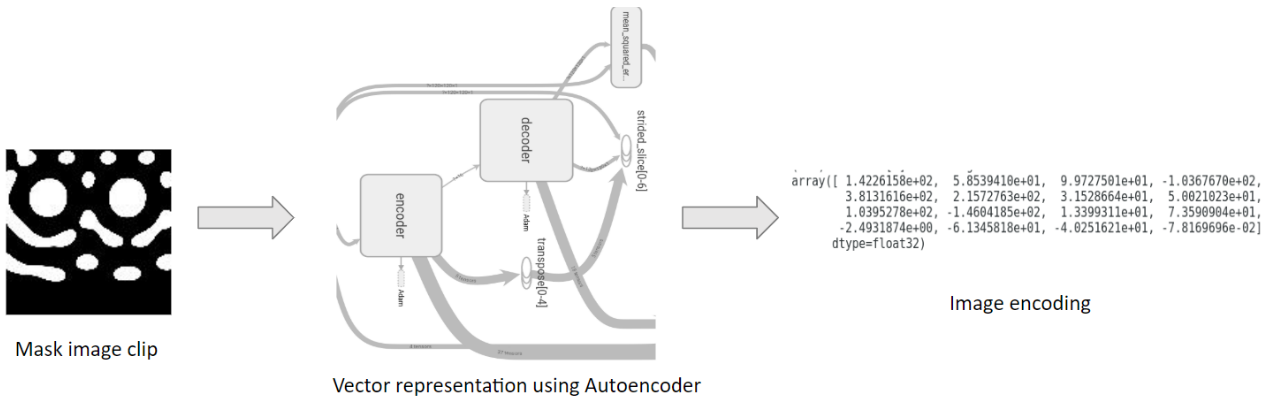


Fig-1: Deep autoencoder creates vectors (series of numbers) representing similar shapes.

2.4 Results

Fig-2 shows eight mask-error clips from a curvilinear mask design rules checker [11]. Each error clip is extracted from the inspected design by taking a window of a specific size (to include the interaction region). To understand the dynamics at play, it is useful to group error clips with similar geometry patterns and similarly sized EPE errors.

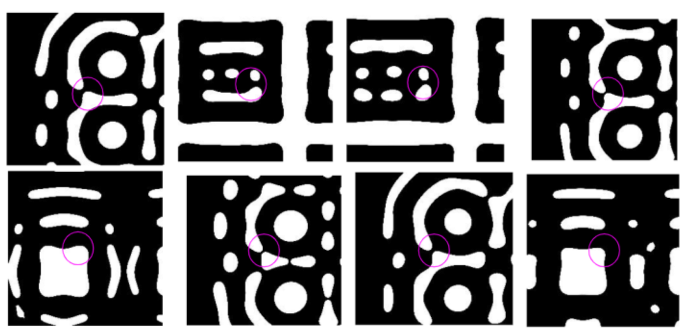


Fig-2: Eight Sample Mask-error Clips. Top left, top right, bottom two middle clips have spacing errors with similar surrounding geometries, thus fall into the same category. Top two middle mask clips also have spacing errors with similar surrounding geometries, yet they are grouped separately. Same is the case with bottom left and bottom right.

After defining similar mask-error clips using the autoencoder, we can group together or cluster the error clips with similar patterns as in Fig-3.

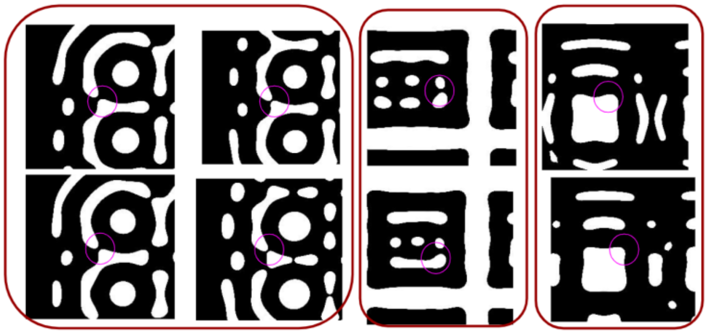


Fig-3: After deep autoencoder finds similar patterns, errors are grouped in three clusters.

To validate the accuracy of the groupings, all error clips in each group of errors were summed as in Fig-4(a). Notice the bottom image is not as sharp as the upper three in Fig-4(a). As you can see in Fig-4(b), the interaction regions in that last group are not as similar as those in other groups. The fuzzy focus of the sum indicates the existence of varying patterns inside the cluster. In this case, users may further nest clusters within that group.

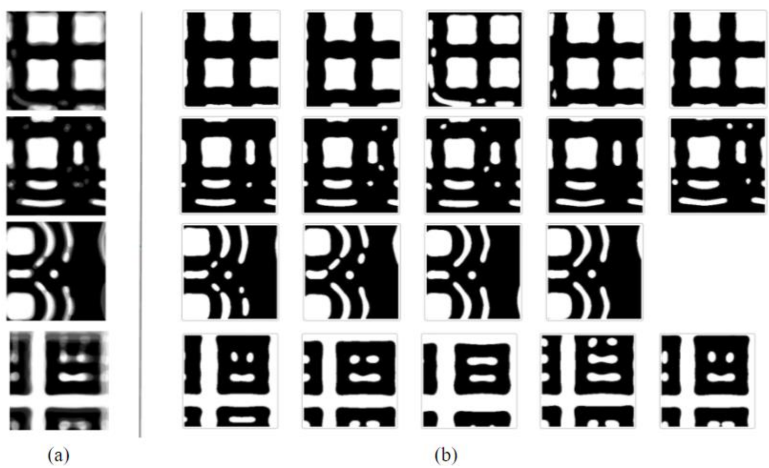


Fig-4: Typical Mean Cluster Images (a) and Error Clips in Each Cluster (b)

3. RECIPE 2: U-NET ARCHITECTURE TO CONSTRUCT FAST MASK DESIGNS

3.1 Motivation: MRI Image Segmentation to Diagnose Brain Tumors

The goal of biomedical image processing is to segment out objects and texture from images of the body to diagnose medical problems, for example visually detecting a tumor in an MRI (Magnetic Resonance Imaging) scan of the brain. In 2015, a new type of neural network called U-Net [2] was proposed to help diagnose problematic regions in the brain based on MRI scan images. The U-Net network is able to segment images to the same degree of accuracy as a human expert. Moreover, U-Net architecture showed great strength in other segmentation applications and image processing tasks [12, 13, 14]. Inspired by these successes, we employed U-Net architecture to construct photomask designs faster.

3.2 Fast Mask Design Construction

The predominant technology for forming geometric patterns on a leading-edge mask reticle today is the variable shaped beam (VSB) eBeam mask writer, where shots or doses of electrons having essentially Manhattan rectangular shapes are exposed to a resist-coated mask-reticle surface to form geometric patterns on the mask. By simulating the effect that the VSB shots will have on the mask before production begins, a shot list and corresponding dose map can be planned with overlapping VSB shots, so that the patterns are written with fewer shots and higher accuracy. However, run-time required for an iterative optimization based on simulation is time consuming. An acceleration using deep learning is possible.

To construct photomask designs faster with deep learning, we used the idea of image segmentation to predict an optimized overlapping VSB shot list for the mask, and based on it, a related output dose map. This approach can be extended to image-to-image translation to speed up physical model-based simulation. In simple terms, Fig-5 shows the problem expressed as a conversion of rasterized overlapping shots to a simulated dose map. With deep learning, the process involves applying image-to-image translation techniques with U-Net to detect areas having overlapping shots on the input image and converting them to a dose map for the mask design.

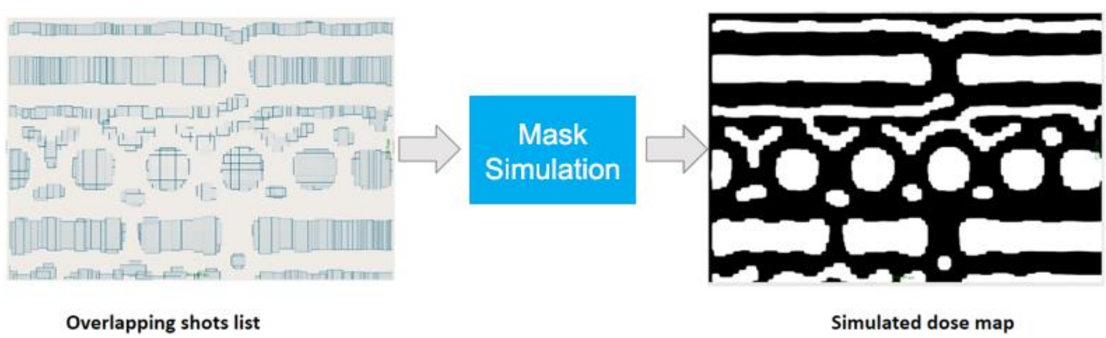


Fig-5: Translation of an Overlapping Shot List to a Simulated Dose Map

3.3 U-Net Architecture

U-Net is a popular neural network architecture in the deep learning field [Fig-6]. Initially used to perform segmentation on biomedical data, U-Net can be extended to various image translation tasks. The U-Net network consists of three main components: the contraction path, the bottleneck and the expansion path. The contraction path is made of many contraction blocks, which use two convolutional layers with kernel size 3x3, followed by a max pool layer of 2x2 for down-sampling. The bottleneck layer applies two convolutional layers followed by an up-convolutional layer, and it connects the contraction and expansion paths. The novel idea of U-Net, however, rests with the expansion path. Like the contraction path, the expansion path uses blocks consisting of two convolutional layers with kernel size 3x3. However, those blocks are followed by a 2x2 up-sampling layer. The input for the expansion path is a concatenated-feature map of the previous layer, with corresponding feature maps from the contraction layer.

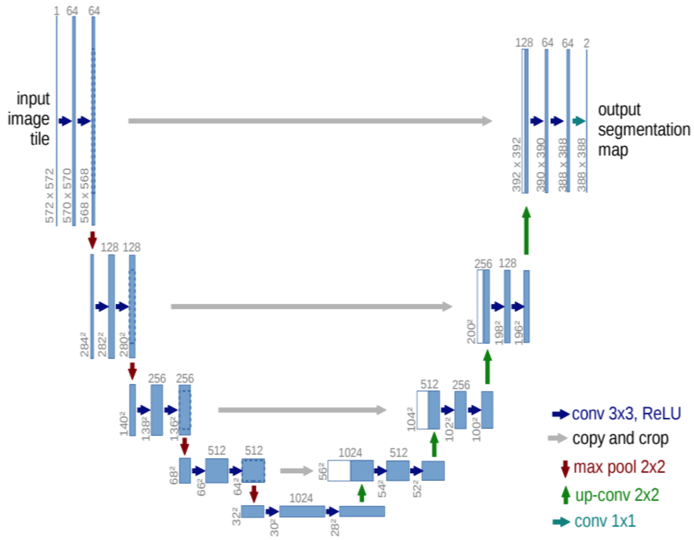


Fig-6: U-Net Architecture

3.4 Results

Applying a U-Net deep neural network, we predicted a mask-shape dose map from an overlapping shot list faster [Fig-7]. To verify the accuracy of the predicted dose map, we employed a technique to compare the contours.

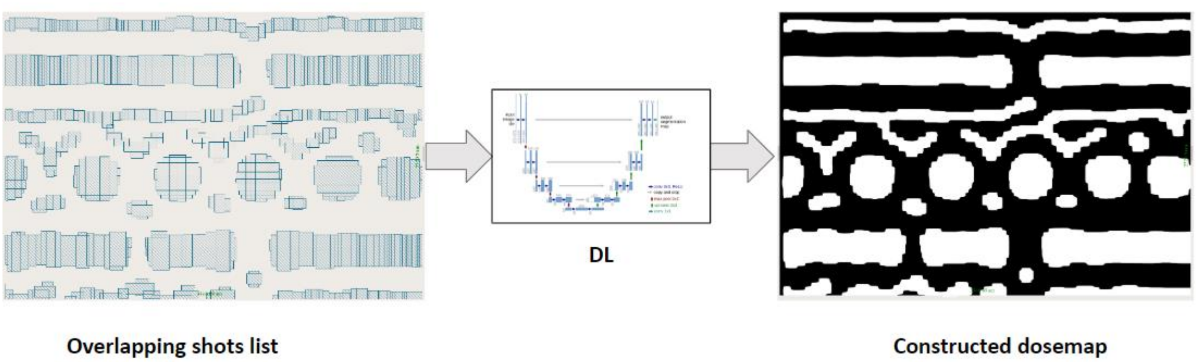


Fig-7: Using deep U-Net, we saw 2.5x improvements in predictions from shots to mask shape.

As shown in Fig-8, first we retrieved the simulated contours and predicted (constructed) contours from the simulated dose map and the predicted (constructed) dose map, respectively. Second, we overlapped the two types of contours on the same coordinates to obtain EPE errors to compare them. In Fig-8, simulated contours appear in red and predicted contours are in green so that we can compare regions with discrepancies between the predicted and simulated contours. Fig-8 shows the EPE distribution of the 1000 worst EPE errors between the compared contours. While the maximum magnitude of EPE error is less than 3nm, typical EPE errors are well within a 0-1 nm range, showing that the network has great potential in various applications.

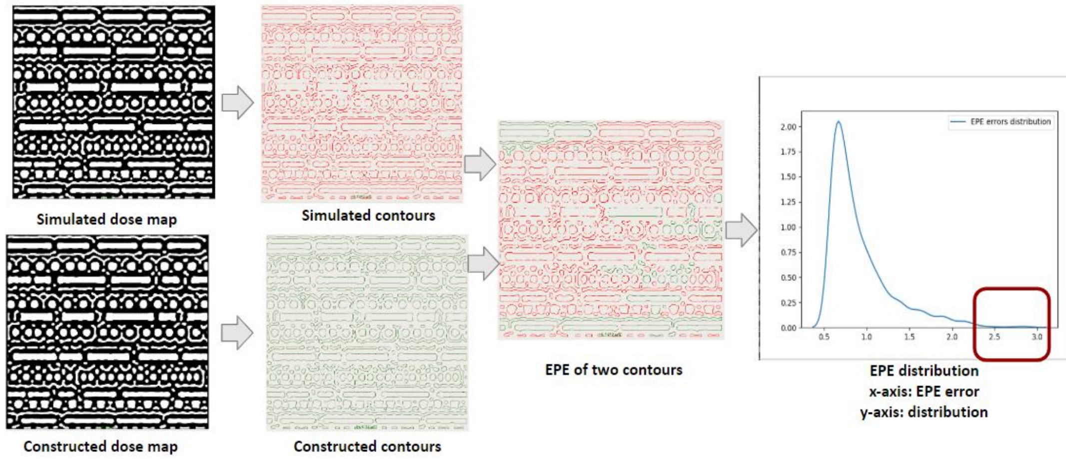


Fig-8: Maximum EPE error on constructed dose map is less than 3 nm.

4. RECIPE 3: OBJECT CLASSIFICATION TO FIND AND CLASSIFY PICK-AND-PLACE (PNP) ERRORS ON PCB COMPONENTS

4.1 Motivation: Automatic Checkout and Self-driving Cars

Object classification is one of the most widely used deep learning techniques in a number of tasks including, but not limited to, automatic checkout in retail stores like Amazon Go, self-driving cars, inventory management, monitoring machines etc. At CDLe, we used computer vision-based object detection and classification to improve the vision judgement system in a PCB-component pick-and-place machine. Instead of object classification, we applied our recipe to error classification.

4.2 PCB Component Pick-and-place Error Classification

A typical PCB may contain chips, transistors, buses and other electronic parts or components. With surface mount technology (SMT), all these components are mounted on the surface of the board using solder paste, making it possible to use smaller components and create more compact PCBs (as compared to through-hole components). Sophisticated robotic machines called PnP machines mount the components [Fig-9. As part of SMT component placement process, the PnP robot arm picks up an electronic component from a magazine of components, observes an image of the picked component, and decides whether to place it on the PCB or discard it. The decision to place the component is made by a vision judgement system, which analyses and decides if the pick is correct, then decides whether to mount or discard the component. Identifying incorrectly picked components is crucial to the successful operation of PCB assembly line and thus requires high precision.

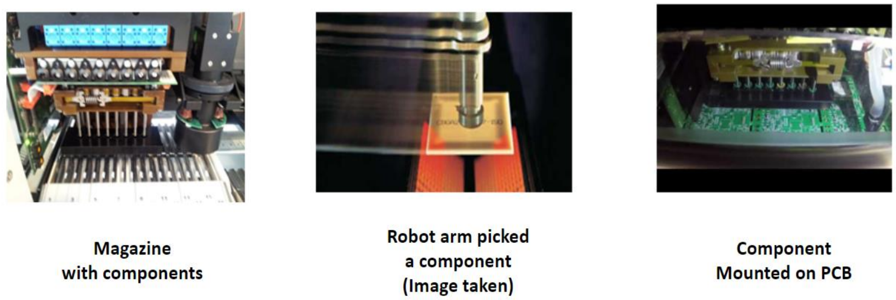


Fig-9: Pick-and-place Machine Flow

To accurately identify incorrectly picked components, a camera inside the machine takes pictures of picked components, and a vision judgement system decides whether to mount or discard. This judgement system usually relies on a classical computer-vision software program to classify whether the component was correctly picked or not, based on the picture taken. As shown in Fig-10, Picked components fall into categories: Ok, Billboarded, Corner pick, Damaged, Not picked, etc. This is a straightforward error classification problem.

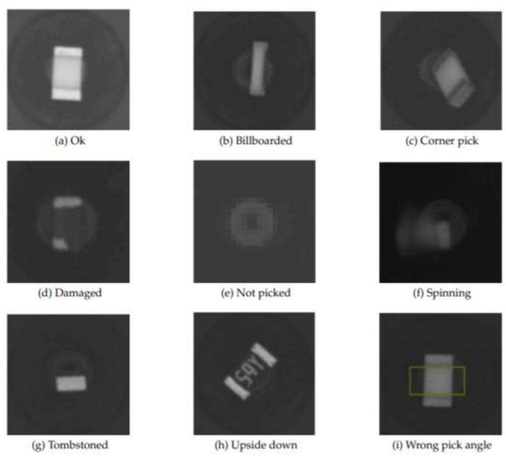


Fig-10: Images Taken and Labels Assigned by a PnP Machine

Classical computer-vision software does not sufficiently address this error classification problem because the pictures taken vary [Fig-11]. The pictures might have different resolutions or component sizes or be taken by different cameras installed on different PnP machines. Classical computer-vision algorithms are based on hand-crafted feature engineering that cannot generalize to account for those varying conditions. To avoid these shortcomings, we applied a deep-neural-network-based convolutional network like VGG16 [8], which is the de facto method for solving an image classification problem. A deep convolutional network automatically learns important features for classification tasks from training data using convolutional filters.

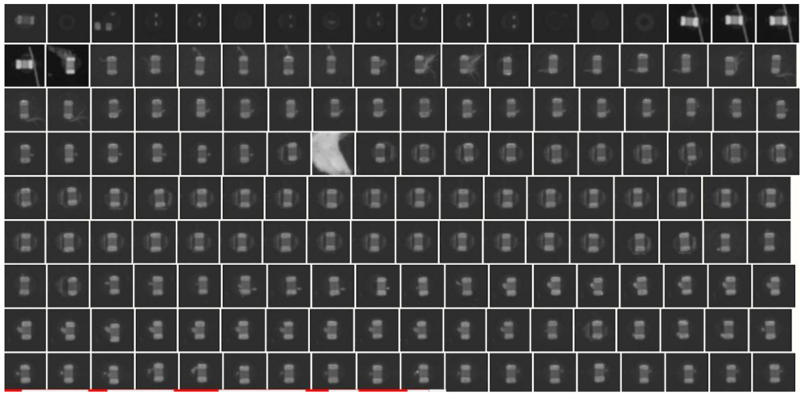


Fig-11: Various Images of Like Objects Taken by a PnP Machine

4.3 Deep Convolutional Architecture (VGG16)

VGG16 is a pioneering neural network architecture for image classification. It has 16 layers, consisting mainly of convolutional layers with 3x3 filters [Fig-12]. The first convolutional layer has 64 filters, doubling with every down-sample layer. Down-sample layers reduce the spatial dimension of input. For example, we can down sample from 112x112 input to 56x56. The output of the final convolutional layer then passes through fully connected layers to give the predicted probabilities of the input image belonging to the predefined class labels.

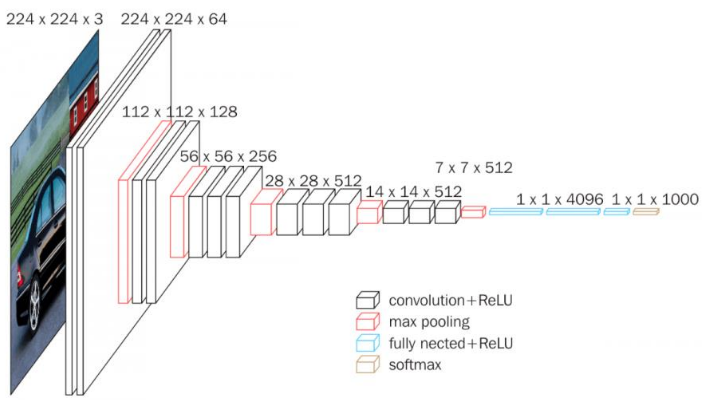


Fig-12: VGG16 Network [15]

4.4 Results

By using a neural network architecture like VGG16, we easily achieved more than 99% accuracy on a hold-out test set [Fig-13]. However, when we tested the network on a new machine with a different camera, the network did not generalize, and the results were worse than for the test set. The root cause of this discrepancy was a change in image domain between the new machine's images and the old machine's images. For example, lighting, contrast and pixel density were different. This problem is formally defined as domain adaptation problem. At CDLe, we tackled this error classification problem by combining a convolutional neural network with transfer learning and domain adaptation techniques to achieve an accuracy of 99.76%, which is better than a trained human's accuracy ranging from 90% to 95%.

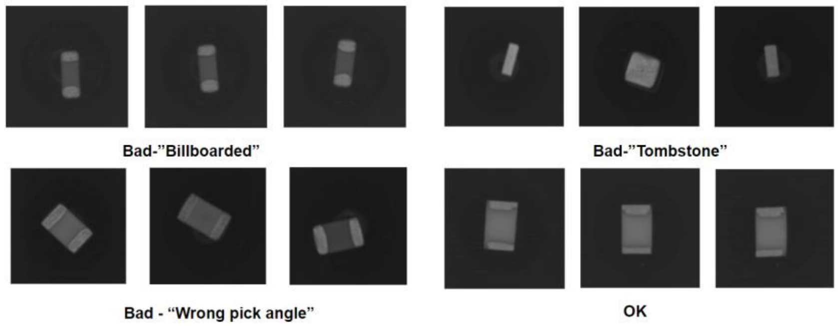


Fig-13: Examples of Classified Images by DL Model

5. RECIPE 4: SURFACE LOG ANOMALY DETECTION TO IMPROVE FPDS

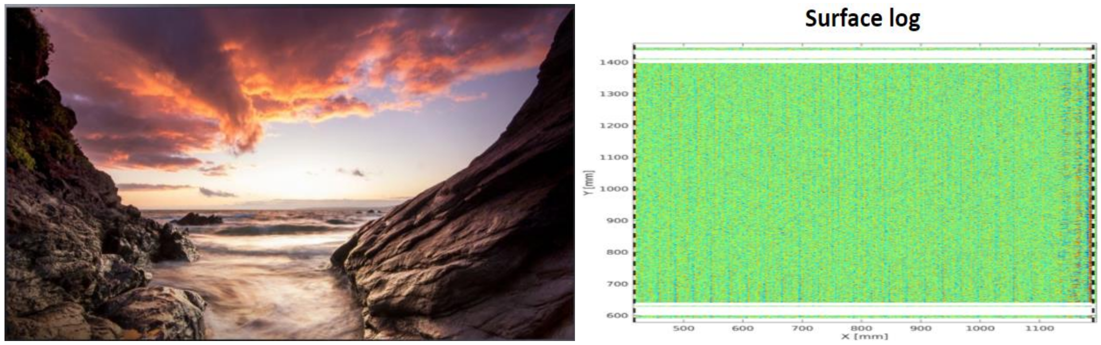
5.1 Motivation: Detecting Credit Card Fraud

Fraud detection is an absolute necessity in banking and the finance industry due to large financial losses involved. Anomaly detection is one of the primary methods to identify fraudulent activity.

5.2 Servo (Surface) Log Based "*Mura*" Identification

The quality of the display [Fig-14] is a critical issue for FPD manufacturing. Visual defects or irregularities such as *muras* can occur due to errors in the mask or the printing process.[2] A *mura* is the appearance of a low contrast non-uniformly bright region on the display panel. To control the quality of the display panels, visual defects must be found on the mask before it is used for printing displays. Traditionally, human experts visually inspect the mask to identify any *muras*. This remains the most common approach to identify *muras. Muras* are also identified using analytic methods with surface logs [Fig-14 (b)]. In these cases, a surface log is created by color coding the parameters recorded during mask writing for every x, y position of the flat panel. Experts empirically set thresholds for the average parameter values to be used across the plate. These thresholds work in most cases. However, our ability to identify *muras* using surface logs is limited. Some *muras* fit in the accepted threshold range and thus are not detected. Using deep anomaly detection techniques, we were able to accurately identify even these difficult-to-discern *muras.*

[2] *Mura* is a Japanese word referring to an irregularity or unevenness.



(a) (b)

Fig-14: A Flat Panel Display and Corresponding Surface Log [16]

5.3 Deep Anomaly Detection

Anomaly detection, also known as outlier detection, is a challenging problem in deep learning with important applications in finance, medicine, traffic flow and cyber security, to name a few. The key technique used to identify anomalies is to learn the accurate distribution of normal data and use that distribution to flag any abnormality. Autoencoders offer a promising method to learn normal data distribution. We applied state-of-the-art deep anomaly detection techniques, estimating the normal data distribution with an estimation network on the encoded vectors and reconstructed errors from an autoencoder [5]. The deep autoencoding Gaussian mixture model (DAGMM) architecture that we came up with for unsupervised anomaly detection is shown in Fig-15.

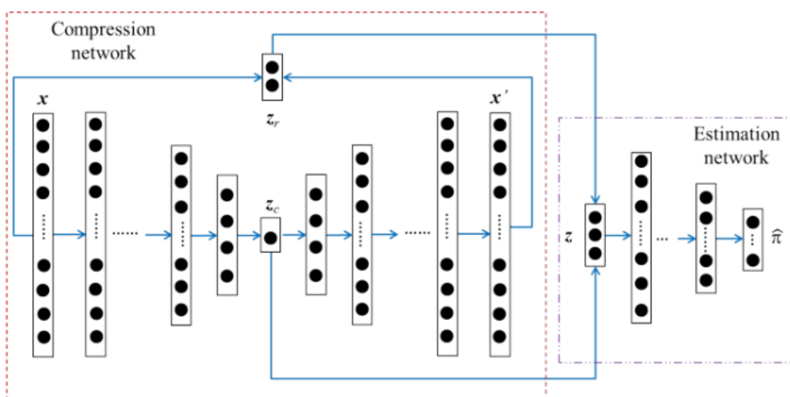


Fig-15: Deep Autoencoding Gaussian Mixture Model (DAGMM) [5] for Unsupervised Anomaly Detection

5.4 Results

With this deep anomaly detection recipe, we generated a mapping function for the FPD's mask to show regions having anomalies. The surface log is cropped into small images that are fed to the learned network to obtain energy values (energy values indicate how often the network has seen similar image crops). Then, abnormal image crops (anomalous regions in the surface log) are filtered out based on surpassing an energy threshold. Normal regions fall below this value. Based on these energy values, one can draw a heatmap to visualize the regions that are detected as anomalies by the neural network. Fig-16 shows surface logs (on the left) and anomalies (bright regions) identified from heatmap images (on the right).

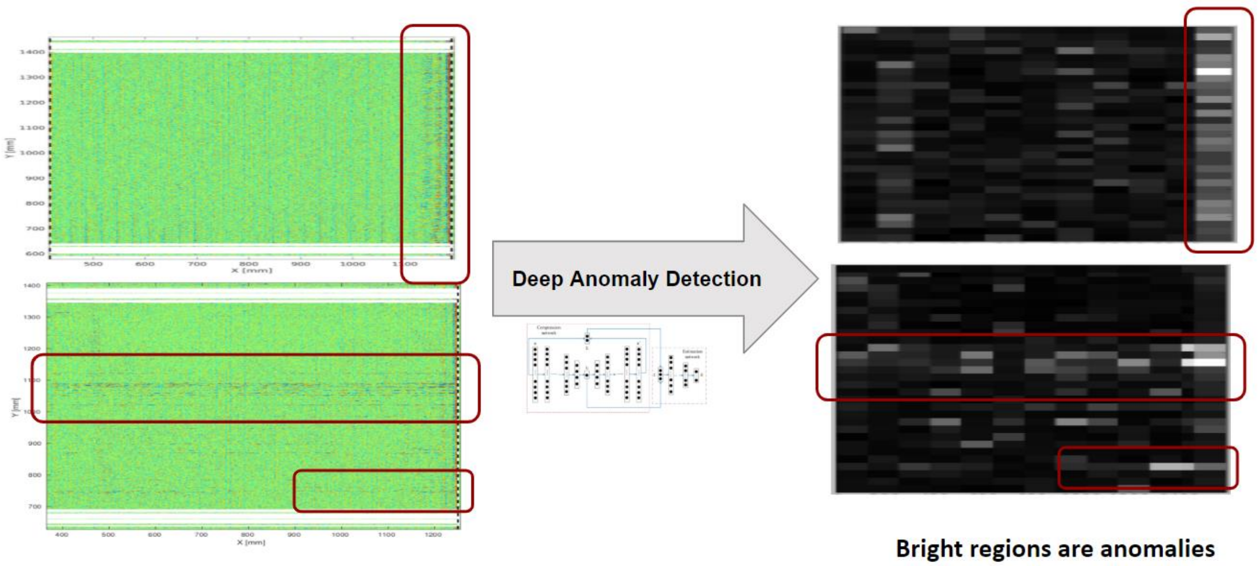
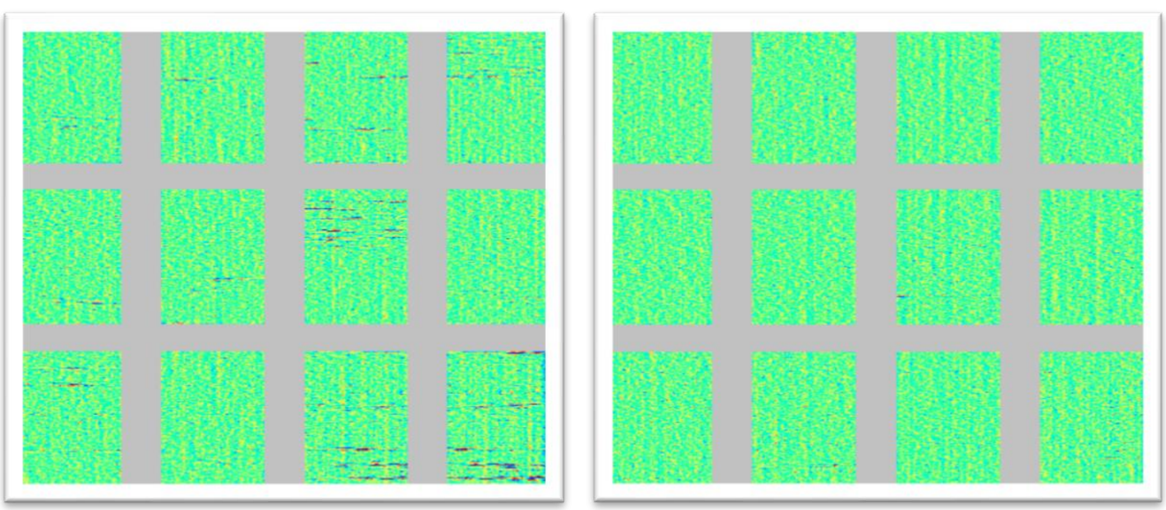


Fig-16: *Muras* Identified by DL Model

To investigate our capability to detect anomalous regions using surface logs, we can plot regions with lowest energy and highest energy. Fig-17 (a) shows regions of a surface log with highest energy. These regions contain several abnormal patterns that fall out of normal distributions. In contrast, Fig-17 (b) shows only normal crops from the surface log that are similar to each other.



(a) (b)

Fig-17: Regions with Highest Energy (a) and Lowest Energy (b)
(Energy values increase from left to right, top to bottom.)

6. RECIPE 5: DIGITAL TWINS TO SYNTHESIZE SEM IMAGES AND INVERSE LITHOGRAPHY TECHNIQUES (ILT)

6.1 Motivation: Digital Twins

A digital twin [Fig-18] digitally replicates physical entities. In general, digital twins are created by digitally modeling the properties, conditions and attributes of the real-world counterpart [6] by using math and physics modelling and a large amount of data collected from real world systems. With enough data from real world systems, deep learning can assist in creating digital twins of real-world processes and equipment. Keeping this idea in mind, we are using deep learning to build "digital twins" for the photomask industry to automate and/or optimize the efficiency of equipment and processes.

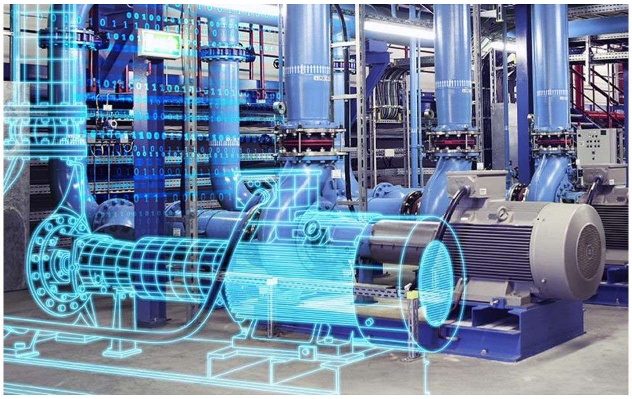


Fig-18: Example of a Digital Twin for a Machine [17]

6.2 Digital Twins for Photomask Applications

Defects in semiconductor manufacturing are important; however, real defects on masks and wafers rarely occur due to robust efforts by manufacturers to avoid them. To work effectively, deep learning algorithms need in the order of ten thousand defects for adequate training data. Finding real defects and collecting data related to them is tedious work and takes time. By creating a digital twin for photomask applications/process, we can study process behavior more efficiently. Digital twins allow us to create enough relatively rare defect data for deep learning training.

We built two types of digital twins: 1) digital twins to synthesize SEM images from the CAD data defining mask shapes which is described in section 6.3, 2) digital twins to construct curvilinear masks with assist features using ILT (described in section 6.4).

6.3 Digital Twins for Generating SEM Images from CAD Data

To create realistic looking SEM images, we used techniques from image-to-image translation literature. A recent popular deep learning architecture for image-to-image translation is pix2pix [2]. pix2pix is an image-to-image translation model that uses conditional Generative Adversarial Networks (GANs). Studies have shown the effectiveness of pix2pix to synthesize photos from label maps, reconstruct objects from edge maps and colorize images [2]. The model we used has a similar structure to GAN-based models. It works in this way: a generator generates images that can fool the discriminator and the discriminator distinguishes between real images and the images generated by the generator. The generative network is similar to image translation networks like U-Net [3] and the discriminator is a regular convolutional neural network (CNN), which classifies whether the image is real or generated by a generator. Rather than generating images at random, we can condition the generator to generate specific images [4]. Fig-19 shows an overview of the network architecture.

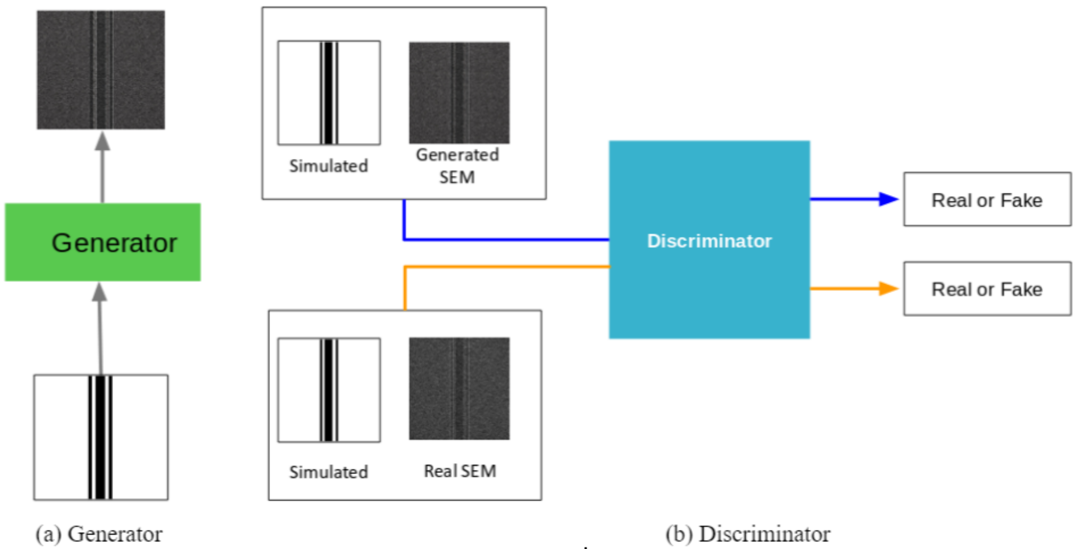
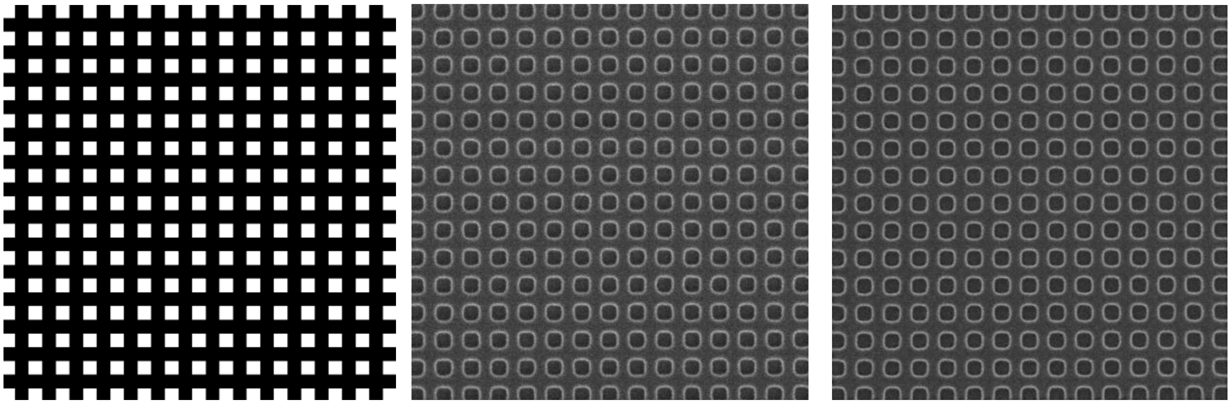
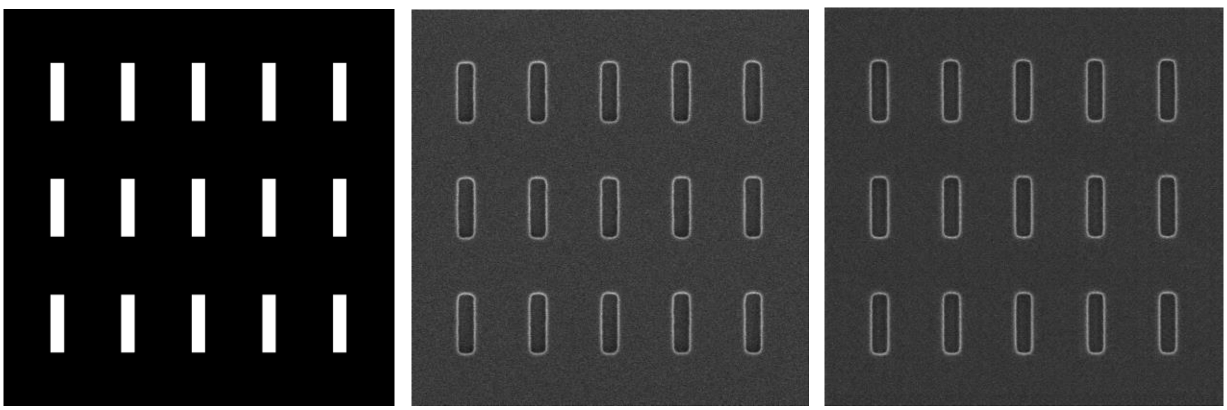


Fig-19: Process for Generating a SEM Image with pix2pix Architecture

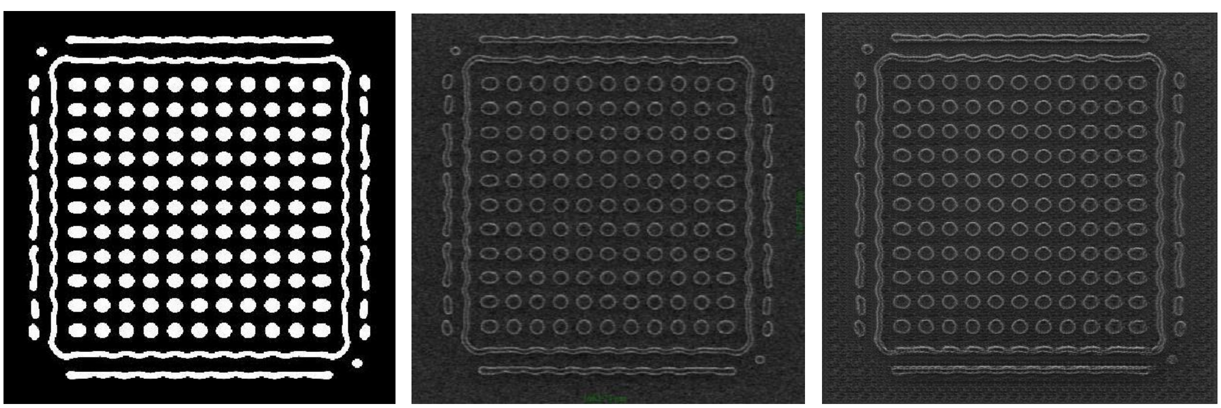
To generate realistic SEM images, we need a pair of training images, i.e., CAD data and a corresponding real SEM to train the model. The generator takes the CAD data as input and tries to create a realistic SEM image from it. The discriminator network takes both the real SEM and generated SEM image and the CAD data as input. The discriminator then classifies whether the SEM image is generated (fake) or real. Fig-20 compares CAD, real and generated SEM images.



a. CAD Image b. Real SEM Image c. Generated SEM Image



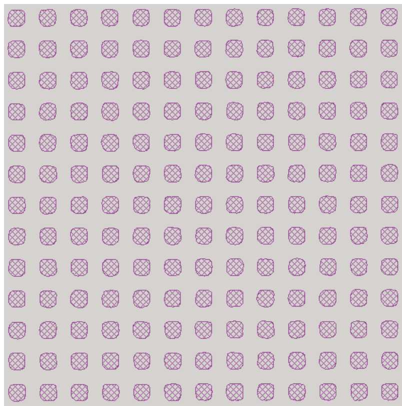
d. CAD Image e. Real SEM Image f. Generated SEM Image

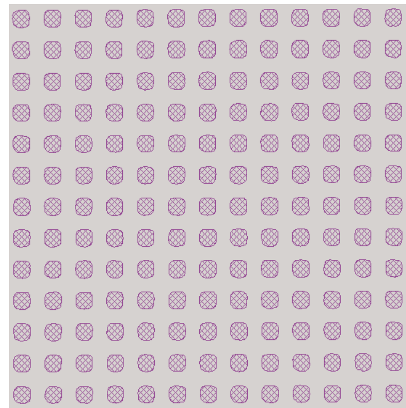


g. CAD Image h. Real SEM Image i. Generated SEM Image

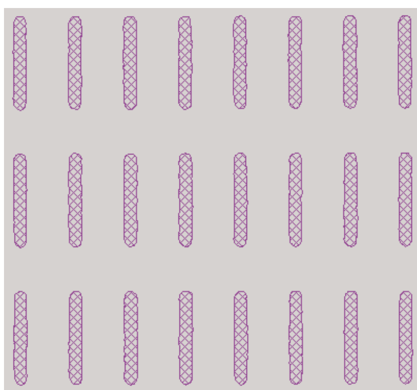
Fig-20: Examples of Generated SEM Images

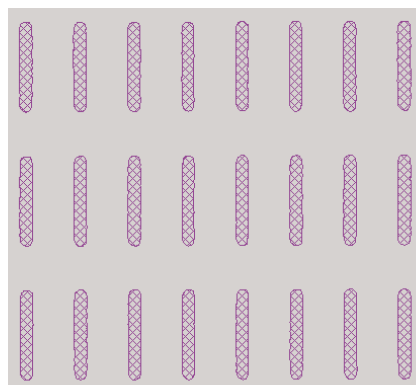
To evaluate the quality of the SEM images we generated, we used aBeam Technologies' contour-extraction software to extract contours from real and generated SEM images. As shown in Fig-21, the extracted contours from real and generated SEM images are very similar.





a. Contour Extraction for Real SEM b. Contour Extraction for Generated SEM





c. Contour Extraction for Real SEM d. Contour Extraction for Generated SEM

Fig-21: Examples of Extracted Contours of Real and Generated SEM images using aBeam

6.4 Digital Twins for ILT

Curvilinear masks have many advantages over rectilinear masks for ILT [7], but adoption requires support from the whole mask-making ecosystem, including mask writers, metrology, inspection, review and repair. Mask rule checks (MRCs), equipment and processes that work for rectilinear do not easily carry over to curvilinear. Despite improvements in turnaround time for generating curvilinear ILT-based masks, the time and computing resources needed are still significant. An ILT digital twin can generate masks that have similar curvilinear characteristics, yet they run more than an order of magnitude faster. This digital twin can generate excellent examples of curvilinear mask shapes with assist features, but it cannot be used for wafer production. Deep learning is a statistical method, so wafer quality will not be as good as its real twin. But the DL-based digital twin is made to generate a legal mask that serves as a very good test case generator of curvilinear masks for mask shops and products that serve the mask shops.

The ILT digital twin is trained through supervised learning with a target wafer shape dose map as input and the real ILT-based mask as the label. A modified U-Net [3] architecture (like other image-to-image translation applications described in this paper) is used. The digital twin rasterizes the CAD data, runs inferences with the trained model, then does MRC cleanup before contouring and writing the mask data. The MRC cleanup is necessary, because deep learning is a statistical process. While the training data may appear MRC clean, the inferred mask may have small violations that must be corrected.

Fig-22 shows two target wafers and related mask shapes with assist features that were constructed using a digital twin.

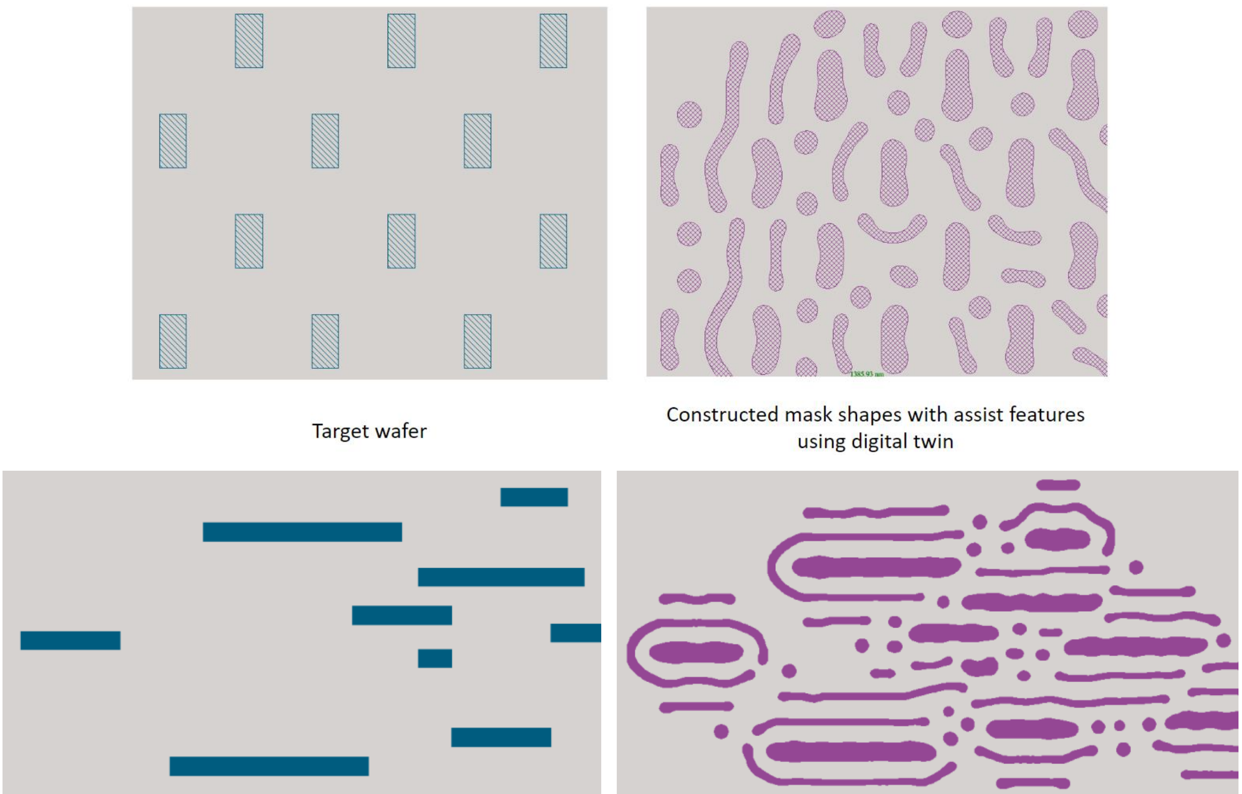


Fig-22: Target Wafers and Constructed Mask Shapes

7. LEARNING FROM THE FIRST YEAR

7.1 Not every problem is suitable for deep learning.

Deep learning techniques are great for specific types of problems. However not all problems can be solved using deep learning. In the photomask industry, tasks that require extremely high accuracy may not be suitably deep learned due to the statistical nature of deep learning. For example, computers can learn to map VSB shots to a mask dose map for fast mask design construction (Section 3, Recipe 2); however, DL suffers from a higher maximum EPE (around 3 nm) compared to detailed simulation.

Machine tasks like error classification for PnP machines (Section 4, Recipe 3) have a much higher level of accuracy than a human performing the same task.

7.2 Deep learning can be used to efficiently generate training data for other deep learning projects.

Deep learning techniques are data hungry, whether they are supervised or unsupervised methods. While unsupervised methods do not need labels, a large amount of unlabeled data is required. Semiconductor manufacturing requires an enormous amount of data which is good for unsupervised methods. However, for supervised methods and semi-supervised methods such as anomaly detection, anomalous data must be created, just as it must be to realize autonomous driving. Being able to generate required training data, particularly for anomalous conditions, is critical to the success of deep learning projects in our industry. Fortunately, simulation-based digital twins already exist in our industry. It may take time to compute, but once training data are generated, they can be used repeatedly for multiple training runs. And when using the resulting deep learning network for production use, inferencing is very fast. We made two deep-learning-based digital twins to further speed up the training process by generating test cases using deep learning. This technique of using deep learning to efficiently generate training data for other deep learning projects is an important technique for all deep learning projects in our industry.

8. CONCLUSIONS

These deep learning recipes cover four different problems in the mask industry and one related to PCB assembly lines. These recipes use at least five types of neural networks: 1) deep autoencoders to capture similar information for clustering errors, 2) U-Net architecture for image-to-image translation, 3) deep convolutional neural networks for classifying errors, 4) deep anomaly-detection networks that use autoencoders with a Gaussian Mixture Model, and 5) pix2pix and mixture models to generate digital twins.

Our observation is that, in general, deep learning associated with log and data analysis is useful for solving problems where a human expert makes a decision based on visual inspection. Tedious and error-prone processes that human operators perform today are good candidates for deep learning. It is difficult for human-based processes to scale to adequately inspect and analyze the big data necessary in the photomask industry. If a human can say, “I know how to flag a problem by looking at it, but it’s hard for me to express an exact algorithm for how I know it is an error.” And if a human can give deep learning all the data on which that judgement is based, we likely have a great deep learning problem. Deep learning can do things normal programming has not been able to. If an engineer finds it hard to express an exact algorithm to solve a particular problem, it is difficult to write a regular software program to solve it. Given enough data, deep learning can automatically figure out what features to care about to make a decision, sometimes without even understanding the context. Problems such as defect (or fault) detection and classification, mask inspection, and the abilities to simulate and replicate the behavior of photomask systems are very well-suited to deep learning.

REFERENCES

[1] Carlos A. Gomez-Uribe and Neil Hunt, Netflix. “The Netflix Recommender System: Algorithms, Business Value, and Innovation” ACM Transactions on Management Information Systems (TMIS).

[2] Isola, Phillip, et al. “Image-to-image translation with conditional adversarial networks.” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[3] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional networks for biomedical image segmentation.” *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

[4] Gauthier, Jon. “Conditional generative adversarial nets for convolutional face generation.” *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester* 2014.5 (2014): 2.

[5] Zong, Bo, et al. “Deep autoencoding gaussian mixture model for unsupervised anomaly detection.” ICLR 2018.

[6] Leo, Pang, et al. “Digital Twins for a Curvilinear World.” eBeam Initiative 2019.

[7] Pearman, Ryan, et al. “Adopting curvilinear shapes for production ILT: challenges and opportunities.” SPIE 2019.

[8] Karen Simonyan, et al “Very deep convolutional networks for large-scale image recognition” arXiv 2015.

[9] A. van den Oord, et al. “WaveNet: A generative model for raw audio.” arXiv pre-print, 2016.

[10] Ajay Baranwal, et al. “Deep Learning Primer: Data is the new source code (invited paper)” SPIE 2018.

[11] D2S introduced GPU-accelerated deep learning kit tailored for photomask and semiconductor manufacturing R&D teams, https://design2silicon.com/wp-content/uploads/2019/04/D2S-TrueMask-DLK-Release-FINAL2.pdf

[12] Yao, Wei; Zeng, Zhigang; Lian, Cheng; Tang, Huiming (2018-10-27). "Pixel-wise regression using U-Net and its application on pansharpening". Neurocomputing. 312: 364–371.

[13] Çiçek, Özgün; Abdulkadir, Ahmed; Lienkamp, Soeren (2016). "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation". arXiv:1606.06650

[14] Iglovikov, Vladimir; Shvets, Alexey (2018). "TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation". arXiv:1801.05746

[15] https://neurohive.io/en/popular-networks/vgg16/

[16] https://webobjects2.cdw.com/is/image/CDW/5036714?wid=1142&hei=818&resMode=bilin&fit=fit,1

[17] https://www.attendmia.com/blog/2019/02/27/explore-the-latest-manufacturing-technology/