A Computationally Optimized Data Augmentation Framework Utilizing cDCGAN for High-Resolution Package Images Acquisition

Yunbin Chang[1], Wonyong Choi[2], and Keejun Han[1,†]

1 School of Computer Engineering, Hansung University, Seoul, Korea

2 R&D Center, Genesem, Incheon, Korea

† E-mail: keejun.han@hansung.ac.kr

Research Purpose

Training deep learning models (DLs) for the semiconductor industry can be challenging due to the lack of sufficient datasets.

Problem 1

DLs related to packaging require numerous datasets. However, due to the sensitive nature of the semiconductor industry where security is paramount, sharing these datasets can be challenging.

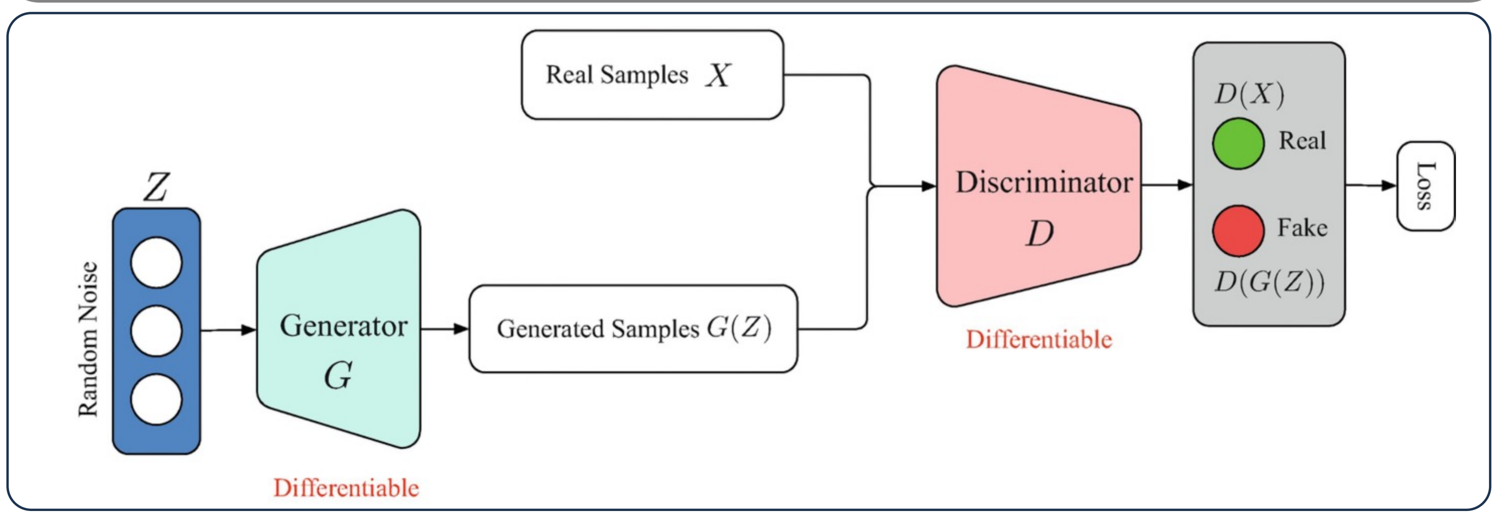
Problem 2

While high-resolution images are necessary for training DLs in the semiconductor industry, the baseline Generative Adversarial Networks (GANs) struggle to generate images of the required high resolution.

Even though baseline models can generate high-resolution images, they are difficult to use in practice due to their high computational requirements and challenges in training. Therefore, we need an augmentation model that requires less computation than the baseline.

Background

Generative Adversarial Network (GANs)

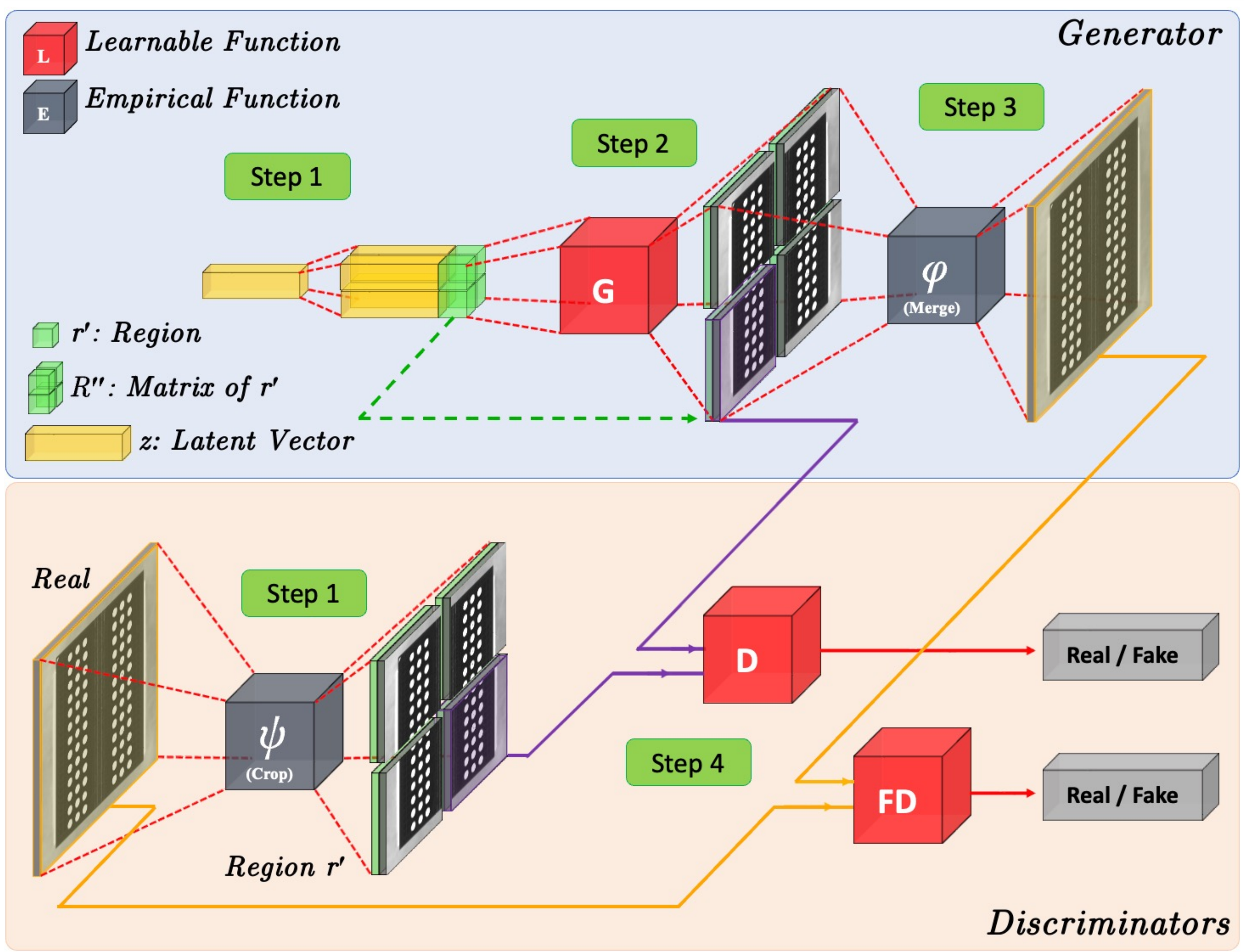


A GAN is a model that augments data to closely resemble real-world instances. To achieve this, We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize log(1 − D(G(z))): In other words, D and G play the following two-player minimax game with value function V (G, D): (Ian J. Goodfellow et al., 2014)

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]$$

It can be extended to a conditional model if both the generator and discriminator are conditioned on some extra information y. y could be any kind of auxiliary information, such as class labels or data from other modalities. (Mehdi Mirza et al., 2014)

Method



Step 01 - Preprocessing

Duplicate the latent vector z into 4 copies and concatenate embeddings of region r'

- The same latent vector is shared for generation across all regions.

Crop the real images into 4 regions and concatenate embeddings of region r'

- Crop the real images into 4 regions (top-left, top-right, bottom-left, bottom-right) with empirical function $\psi(x, r')$ and concatenate the embedding of region r' of them

Step 02 - Generating

Feed the latent vectors and regions R'' to Generator

- When generating images of sub regions, the same latent vector is used to generate visually smooth and globally coherent full images.
- Sub-images are generated for each individual sub-region r'. It's essential to retain the region r' as it allows the discriminator to ascertain if a generated image meets the given condition r'.

Step 03 - Merging

- Merge the sub images into a full image with empirical function $\varphi(G(z, R''))$.

Step 04 - Discriminating

Full Discriminator : we added an additional discriminator to check the fully merged images, making the full image globally smooth and coherent.

- A Discriminator distinguishes if the partial image looks real and is a partial image of region r'
- Our full discriminator distinguishes if the merged full image looks real or fake and looks natural.

Step 05 - Training

$$\min_G \max_{D,FD} V(G,D,FD) = \mathbb{E}_{x\sim p_{data}(x)}[log\, D(\psi(x,r'))] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z,r')))]$$
$$+\mathbb{E}_{x\sim p_{data}(x)}[log\, FD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-FD(\phi(G,z,R'')))]$$

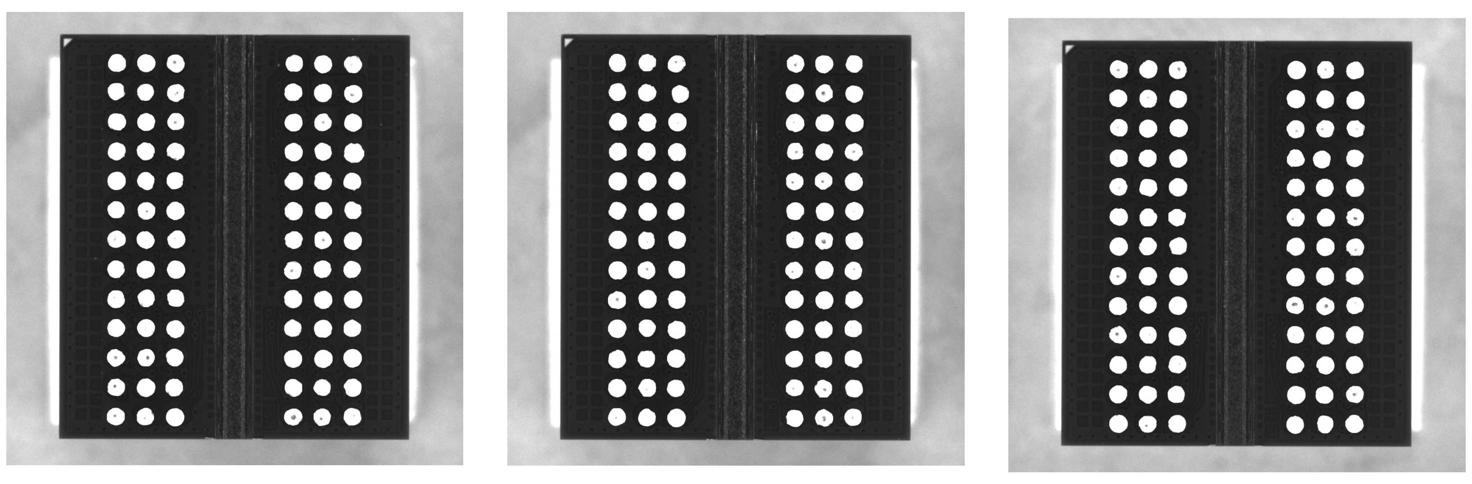
Train our framework according to the above our objective function.

We train D and FD to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize $log(1-D(G(z, r')))$ and $log(1-FD(\varphi(G(z, R''))))$. In summary, G is trained to deceive both D and FD.

Experiment

01 Datasets

- Dataset : BGA Bottom Vision (1566 images), Collected in the actual industrial field.
- The package area was cropped and resized from images with a resolution of 4000x3000.



02 Performance of our framework

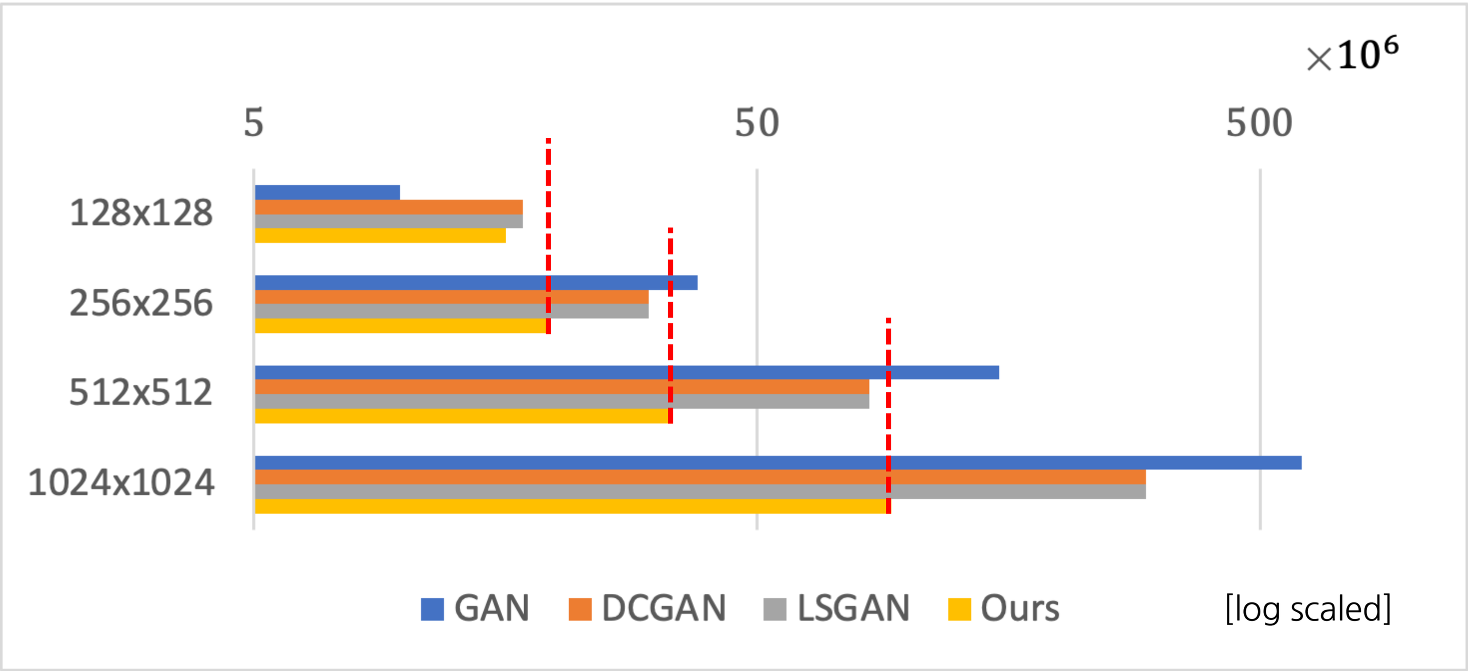
Fréchet inception distance (FID)

Resolution	GAN	DCGAN	LSGAN	**Ours**
128x128	99.712801	82.338656	79.351856	72.395561
256x256	107.631893	68.677863	58.333951	52.028937
512x512	*	59.357004	44.244527	34.310581
1024x1024	*	*	*	24.49373

* Execution failed due to insufficient memory

- All models except for Ours failed to generate images at a resolution of 1024x1024 due to insufficient memory.
- Ours showed the lowest FID score at all resolutions.

03 Memory efficiency of our framework

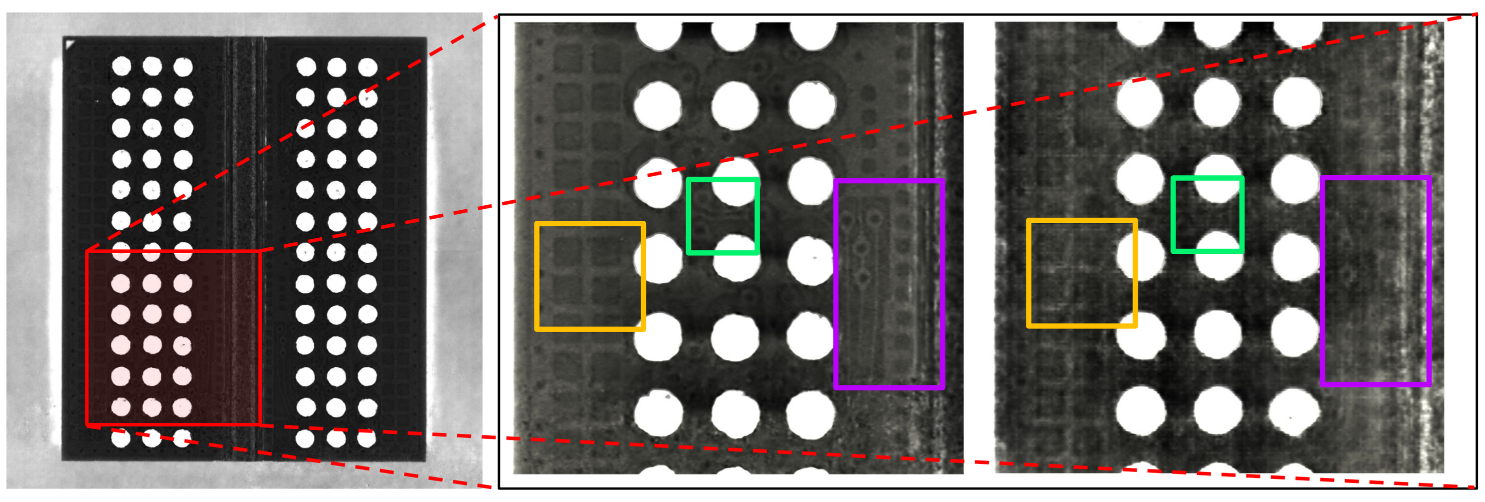


Number of parameters (rounding down to 1,000)

Model		128x128	256x256	512x512	1024x1024
GAN	Generator	1,307	4,502	17,281	68,399
	Discriminator	8,463	33,629	134,292	536,945
	Total	**9,770**	**38,131**	**151,574**	**605,345**
DCGAN	Generator	15,218	28,293	80,591	289,782
	Discriminator	1,922	2,119	2,905	6,051
	Total	**17,141**	**30,412**	**83,496**	**295,833**
LSGAN	Generator	15,218	28,293	80,591	289,782
	Discriminator	1,922	2,119	2,905	6,051
	Total	**17,141**	**30,412**	**83,496**	**295,833**
Ours	Generator	11,991	15,382	28,948	83,212
	Discriminator	1,937	1,986	2,183	2,969
	Full Discriminator	1,922	2,119	2,905	6,051
	Total	**15,851**	**19,488**	**34,037**	**92,233**

- Despite the addition of the full discriminator, our model uses far fewer parameters than other models to generate images of the same resolution.

04 Result



[Ours 1024x1024] [Zoom 512x512 by Ours] [Zoom 512x512 by DCGAN]

- We were successful in augmenting semiconductor package images of a high-resolution (1024x1024) that we could not augment on the same computer.
- We augmented images with higher detail and quality using our model compared to images augmented by other models at the same resolution.

05 Limitation

It can increase memory efficiency and generate images with higher resolution and fidelity, but there may be consistency issues in each region.

- Unnatural lines can appear at the boundary where the 4 sub-regions meet.
- When each of the sub-regions are generated and separately merged, sometimes it may not be coherent.

Conclusion and future work

We have demonstrated that our proposed framework outperforms other comparative models in terms of fidelity and memory efficiency. It generated 1024x1024 images, which other comparative models could not generate, using 69~84% fewer parameters. In addition to quantitative evaluation, it also showed better results in visual evaluation.

Future work

01 The approach to comprehensively resolve the consistency issue will be researched.

02 The approach to soften the boundaries where the sub regions meet will be researched.