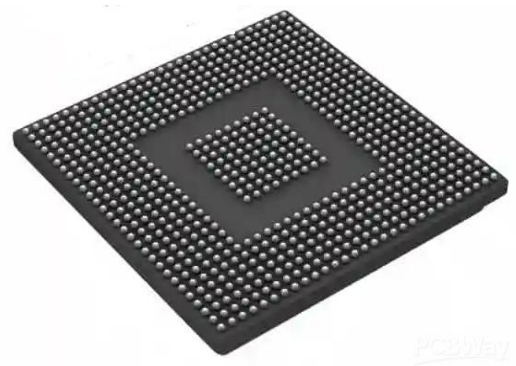
BGA Anomaly detection by exploiting Variational AutoEncoders

Gyeongmin Lee[1], Keejun Han[2], Wonyong Choi[2]
[1]School of Computer Engineering, Hansung University, Seoul, Korea, [2]R&D Center, Genesem, Incheon, Korea
E-mail: gyeongmin@hansung.ac.kr

Background

Ball Grid Array

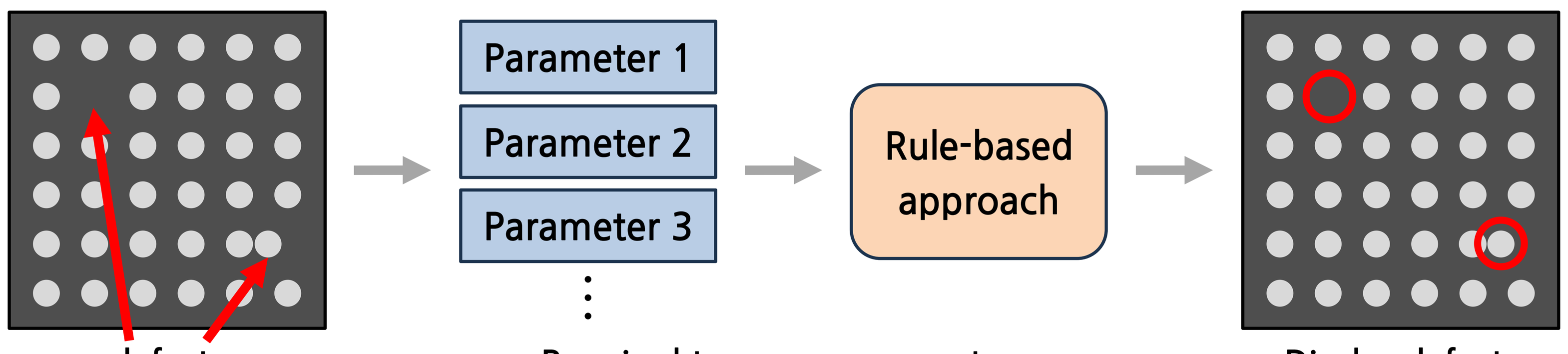
- BGA (Ball Grid Array) is a type of semiconductor package that features hundreds of connection points arranged on its underside.
- Common flaws in BGA include cracks, scratches, missing connections, and bridging, which can adversely affect the semiconductor's electrical performance.



Accurate and efficient inspection of BGA defects is therefore essential.

Research Purpose

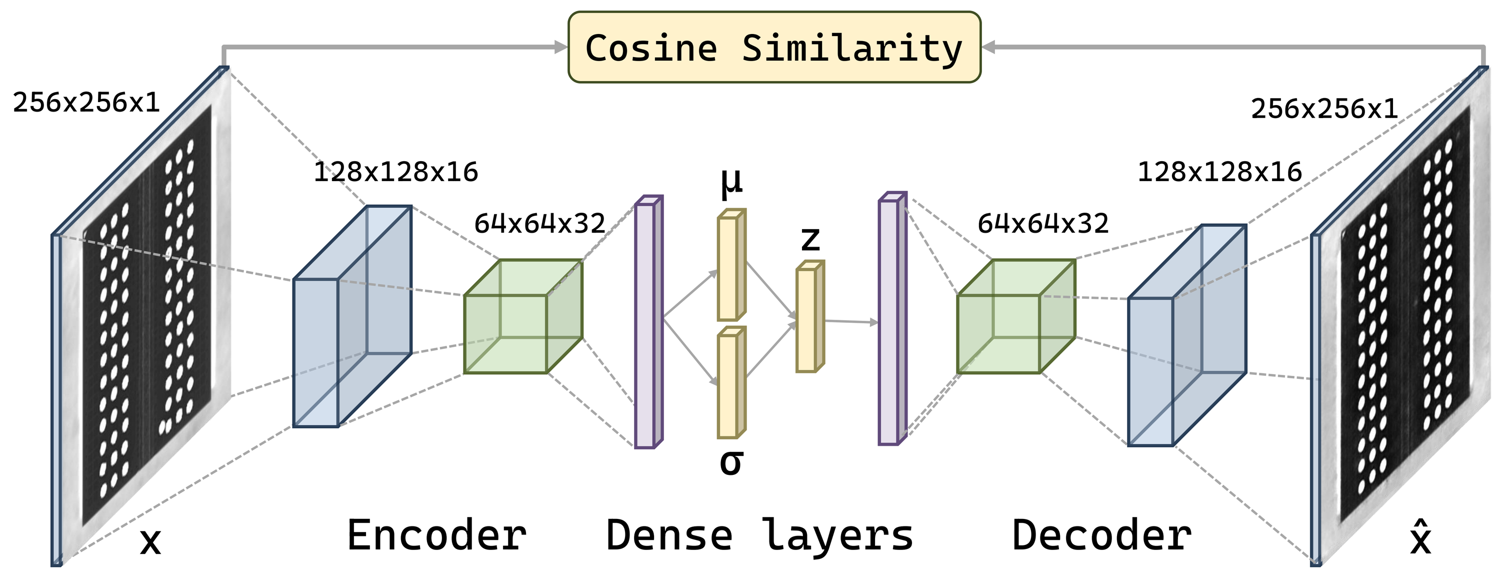
- Most existing methods for BGA defect detection rely on rule-based approaches, making them dependent on pre-set parameters by the user.
- This poses challenges as material changes would require frequent reconfiguration.



defects | Required too many parameters | Display defects

Need an inspection method that doesn't require parameter configuration.

Method



Step 01 - Encoder Network

- The encoder network takes an input x and maps it to a latent space represented by parameters μ (mean) and $\log(\sigma^2)$ (log variance).
- Typically consists of convolutional layers for image data or recurrent layers for sequence data.
- Two vectors μ and $\log(\sigma^2)$ define the distribution of the latent variables.

Step 02 - Reparameterization Trick

- The reparameterization trick is used to sample from the latent variable z in in a differentiable way, which allows backpropagation to pass through.
- Sample ϵ from a standard normal distribution $N(0, I)$.
- Compute $z = \mu + \sigma \odot \epsilon$.

Step 03 - Decoder Network

- The decoder network takes the sampled latent variable z and maps it back to the original data space.
- Typically the mirror image of the encoder network, using transposed convolutional layers for image data or recurrent layers for sequence data.
- The reconstructed data $\tilde{x}$, which is as close as possible to the original input x.

Step 04 - Loss Function

Reconstruction Loss
- Measures the difference between the original data x and the reconstructed data $\tilde{x}$.
- $Reconstruction\ Loss = ||x - \hat{x}||^2$

KL Divergence
- Measure the difference between the posterior distribution $q(z|x)$ of the random variable z and the standard normal distribution $N(0, I)$.
- $KL\ Divergence = \frac{1}{2}\sum_i(\mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1)$

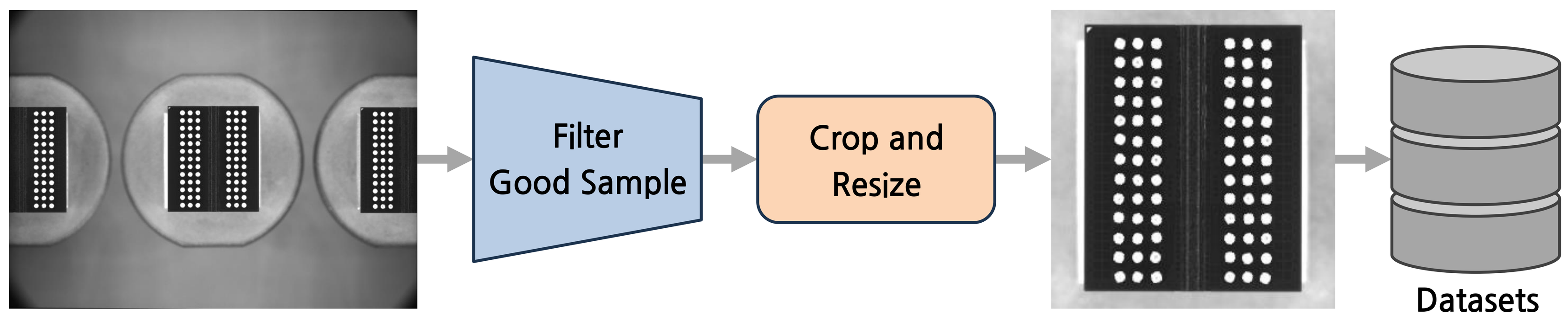
Total Loss
- $Loss = Reconstruction\ Loss + KL\ Divergence$

Step 05 - Cosine Similarity

- Use trained VAEs to generate images based on the original input image.
- Compute the Cosine similarity between the original input image and the reconstructed image. And This is utilized as an Anomaly Score.
- In the case of cosine similarity for defective data, it is lower than that for good data.

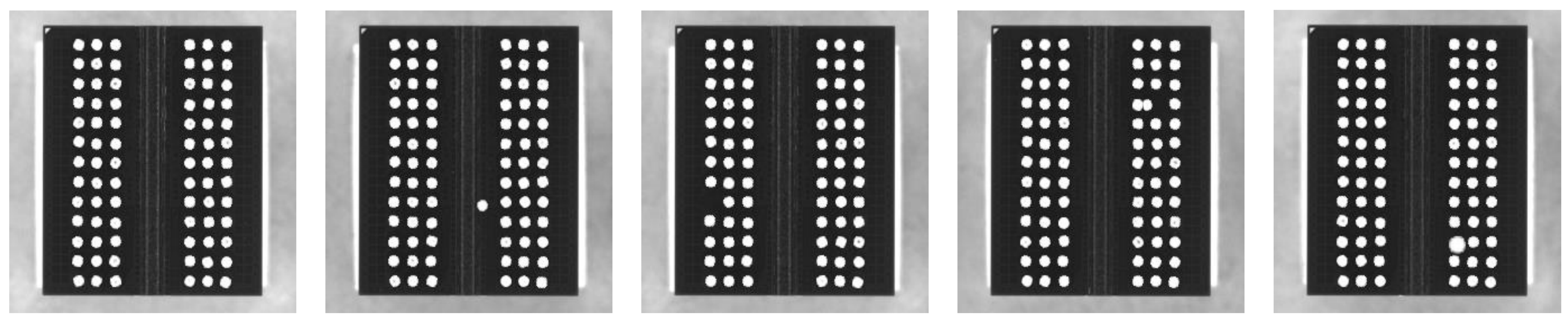
Experiment

01 Data Preprocessing



- BGA dataset collected from actual industrial fields (1024 images)
- The package area has been cropped and resized to a resolution of 256x256.
- These are train datasets, and all the images are good quality.

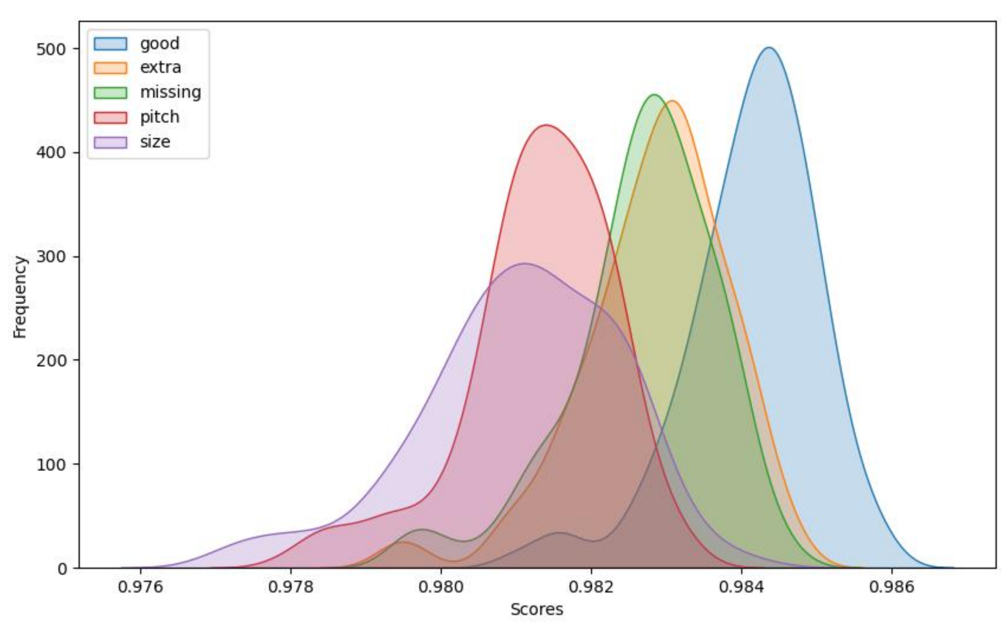
02 Generating Abnormal Data

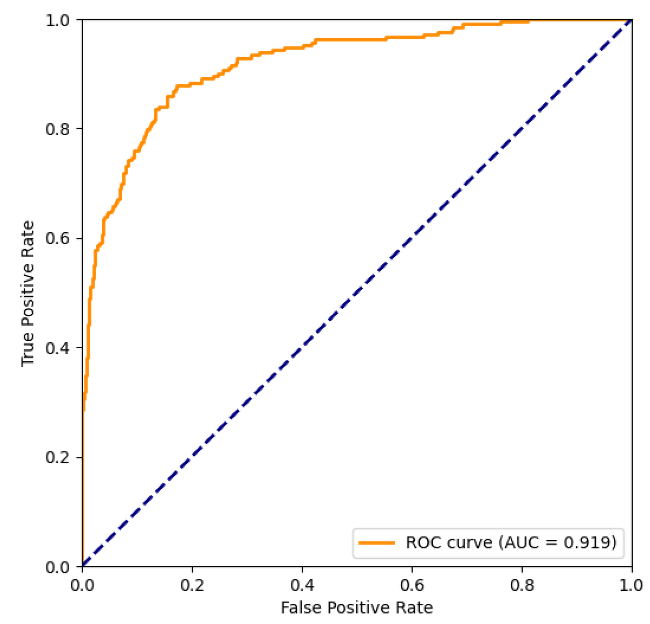


Good | Extra Ball | Missing Ball | Ball Pitch | Ball Size

- In the industrial field, the occurrence of defective data is exceedingly rare.
- The original image data is segmented into Ball and Package areas using morphology techniques.
- Balls are placed on the package according to a given probability.
- These are the test datasets, and similar to the actual defective image

03 Performance of our framework

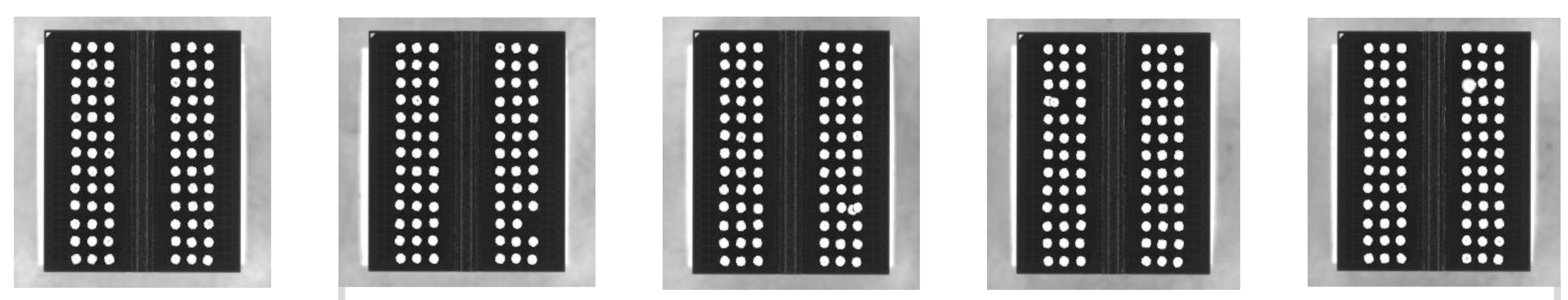




Method	Accuracy	Precision	Recall	Specificity	F1-Score	AUC
Ours (VAE)	0.849	0.958	0.946	0.859	0.898	0.919

- When looking at the density plot of anomaly scores by category, it is clear that good-quality images and defects are distinctly separated.
- After binarizing error items and good items and then inspecting for defects, it shows a significantly accurate inspection.
- Our approach doesn't require any preset parameters and demonstrates a sufficiently fast inspection speed.

Method	Speed	Pre-set Parameter
Rule-based	0.975ms	36
Ours (VAE)	3.424ms	0



Classified as good | Classified as abnormal

04 Limitation

- Small errors are not well detected, so a model that responds more sensitively to errors is needed. Due to the inherent unexplainable nature of deep learning, it's hard to explain the reasons for each classification.

Conclusion and future work

In this study, we propose a method based on VAE, eliminating the need for pre-configuring inspection parameters. By running our experiment, the proposed model outperforms rule-based methods. Utilizing this model will enable defect detection without the need for pre-set parameters, thus providing a more user-friendly machine vision user interface.

Future work

01 Introduce highly sensitive model for detecting even the smallest errors in images.

02 Implement an image classification model to categorize images based on the types of errors found.