

Comparative Study of Unsupervised Deep Learning Anomaly Detection Techniques for Steel Surface Inspection

School of Mechanical and Control Engineering
Handong Global University

Minwoong Han

Comparative Study of Unsupervised Deep Learning Anomaly Detection Techniques for Steel Surface Inspection

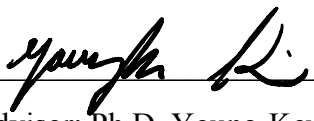
A Bachelor's Thesis

Submitted to the School of
Mechanical and Control Engineering of
Handong Global University

Minwoong Han

June 2024

This certifies that the bachelor's thesis is approved.


Thesis Advisor: Ph.D. Young-Keun Kim

The Dean of Faculty: Ph.D. Chong-Sun Lee

School of Mechanical and Control Engineering

Handong Global University

June 2023

Extended Abstract

Comparative Study of Unsupervised Deep Learning Anomaly Detection Techniques for Steel Surface Inspection

This study investigates recent deep learning techniques utilized for anomaly detection in images, aiming to apply these validated models to a steel dataset. In the steel industry, produced steel can have various defects such as cracks and localized damage. The industry focuses on developing AI models that can automatically detect these defects. By integrating high-accuracy AI models, the goal is to detect surface defects in the produced items and inspect the relevant processes.

There are two major limitations in the field of anomaly detection on steel surfaces. First, abnormal elements occur very rarely during actual steel production, leading to a class imbalance problem where there is a significant difference in the number of normal and abnormal samples. Second, even when abnormal data with surface defects exist, companies generally do not disclose such elements. Consequently, there is a need for deep learning models that can address the class imbalance problem and for open steel datasets that can be applied. There are mainly two methodologies in anomaly detection using deep learning. The first is supervised learning-based methods that learn from both normal and abnormal data. This methodology requires labeled data as it learns the characteristic distributions of both normal and abnormal data to perform classification or anomaly detection later. The second is unsupervised learning-based methods that train solely on normal data. This approach learns the characteristic distributions of normal data and, during inference, addresses the anomaly detection problem by identifying whether the data is abnormal or determining which part of the data contains anomalies. To overcome the aforementioned limitations in the actual industrial field, this study aims to apply unsupervised learning-based methods to an open steel dataset. There are various unsupervised learning methods, including feature embedding techniques and reconstruction-based methods. Feature embedding techniques learn the characteristics of normal data, while reconstruction-based methods map the characteristic information of normal

data to a latent space and then reconstruct the original. This study first implements and evaluates the performance of the representative reconstruction-based method, Convolutional Autoencoder. Then, it applies and compares the performance of the latest methods in feature embedding and reconstruction-based techniques, such as PatchCore and DRAEM models. PatchCore maps the information of normal data to a memory bank space, while the DRAEM model learns to restore arbitrarily generated abnormal data to normal data. Both models can perform not only the normal/abnormal classification task but also anomaly segmentation to determine which part of the data contains anomalies. The study details the structures of these models and compares their performances through practical application.

As a result, this study successfully applied the two models but failed to apply the DRAEM model in practice. The content also covers the analysis of the reasons behind the failure in applying the DRAEM model.

Table of contents

Extended Abstract	1
I. Introduction	4
1.1. Research background	4
1.2. Research purpose	4
II. Selection of unsupervised anomaly detection model	5
2.1. Unsupervised anomaly detection methods	5
2.2. Used deep learning methods	6
2.2.1. Convolutional autoencoder	6
2.2.2. PatchCore	8
2.3.3. DRAEM	9
III. Experiments	11
3.1. Datasets	11
3.2. MVTec dataset evaluation	13
3.3. Severstal dataset evaluation	16
IV. Discussion	17
V. Conclusion	18
Reference	19

I. Introduction

1.1. Research background

In recent times, the steel industry has focused on developing deep learning AI models capable of automatically detecting surface anomalies in produced steel. Even if such models are applied in real-world settings, the inherent variability in the shapes of normal data and the diverse types of abnormal elements pose a limitation, leading to a gap in the expected stability of actual implementation. Additionally, abnormal defects are extremely rare during actual steel production. This results in a significant class imbalance problem, with a large discrepancy between the number of normal and abnormal datasets. Consequently, there is a need for deep learning models that can address these limitations, and related research is actively being conducted. Therefore, this study aims to apply these techniques (unsupervised anomaly detection), evaluate their performance, and determine which model is most advantageous for practical integration.

1.2. Research purpose

This research primarily aims to explore and apply the latest deep learning techniques for anomaly detection in images. The focus is on applying the PatchCore model, a representative feature-embedding technique, to perform anomaly detection. Before applying it to the steel dataset, the model will first be validated using the MVTec standard dataset. Next, the study aims to apply the DRAEM (Discriminatively trained Reconstruction Embedding for surface anomaly detection) model, a representative technique of the reconstruction-based methodology, to the steel dataset. Similarly, this model will also be built and validated using the MVTec standard dataset before its application to the steel data. Following this, the performance of the widely-used Convolutional Autoencoder model will be evaluated. The study will then compare the classification and segmentation performance of each model. Based on these metrics, the study aims to assess which model is most suitable for practical AI integration. If the performance is suboptimal, the study will analyze the underlying reasons for the low performance.

II. Selection of unsupervised anomaly detection model

2.1. Unsupervised anomaly detection methods

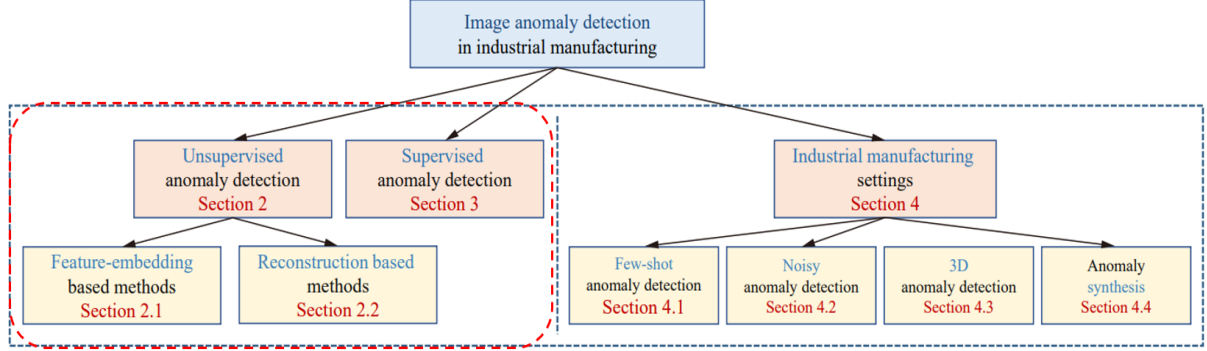


Figure 1. Methodologies of image anomaly detection

There are two main methodologies for image anomaly detection. The first is supervised anomaly detection, and the second is unsupervised anomaly detection. Supervised anomaly detection techniques utilize both normal and abnormal samples to learn the statistical features of each class. In contrast, unsupervised anomaly detection techniques are trained using only normal data. These techniques learn the statistical features of the normal data and then perform tasks such as normal/abnormal classification and abnormal location visualization during the inference stage. In actual industrial settings, the class imbalance problem arises due to the extremely rare occurrence of abnormal samples. Therefore, applying the unsupervised anomaly detection technique, the second methodology, is more appropriate in such scenarios.

Within the unsupervised anomaly detection approach, there are also two methodologies. The first is the feature-embedding methodology, and the second is the reconstruction-based methodology. The feature-embedding methodology involves extracting the characteristic distributions of normal data and mapping them to a specific space. Subsequently, it detects anomalies by mapping the features of abnormal data and identifying differences from the learned characteristics. The reconstruction-based methodology inputs normal data into an encoder to map the feature information to a latent space, and then reconstructs the original data to detect anomalies. The feature-embedding methodology includes four techniques: Teacher-Student Network, One-Class Classification, Distribution Map, and Memory Bank. The methodologies for each technique are as follows.

Table 1. Methodologies of feature-embedding method

Methods	Methodology
Teacher-Student network	Knowledge distillation using backbone network for student model
One-class classification	Finding the hypersphere of the distribution held by normal data
Distribution map	Mapping the feature distribution using Normalizing Flow to a Gaussian distribution
Memory bank	Mapping the distribution held by normal data to a Memory bank

2.2. Used deep learning methods

2.2.1. Convolutional autoencoder

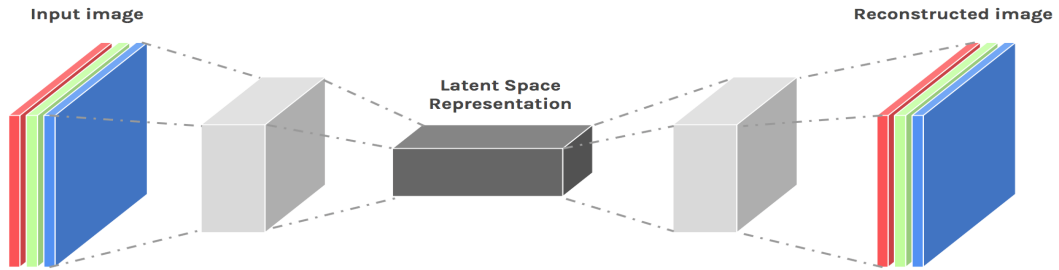


Figure 2. Pipeline of convolutional autoencoder

The Convolutional Autoencoder is an unsupervised learning technique that learns using only normal data. It takes normal images as input, maps them to a latent space, and performs learning. This process begins with an encoder structure. Through multiple convolution operations, the size of the original image is continuously compressed, resulting in the creation of a feature map. Consequently, the model learns the characteristic information of normal data in a low-dimensional space. Following this, the decoder decodes the feature information back to the original image size. Through this process, a reconstructed image is produced. By comparing the original input image with the reconstructed image, an anomaly segmentation task is performed by setting a certain threshold to visualize the areas where anomalies exist.

In the case of the MVTec dataset, which will be introduced later, the original image size is 900x900. For computational efficiency, the images are resized to 256x256 for subsequent

operations. The autoencoder's encoder and decoder structures for the 256x256 MVTec dataset images are established as follows. Both the encoder and decoder are constructed using a total of four convolutional layers. The kernel size for each convolutional layer operation is set to 3, with a stride of 2 and padding of 1. Additionally, the ReLU function is used as the activation function. The construction information for each layer is summarized below.

Table 2. Established layer of convolutional autoencoder

Block	Layer	Type	Input Channel	Output Channel	Kernel size	Stride	Padding	Activation function
Encoder	1	Convolutional layer	3	32	3x3	2	1	ReLU
	2		32	64				
	3		64	128				
	4		128	256				
Decoder	1		256	128				
	2		128	64				
	3		64	32				
	4		32	3				

2.2.2. PatchCore

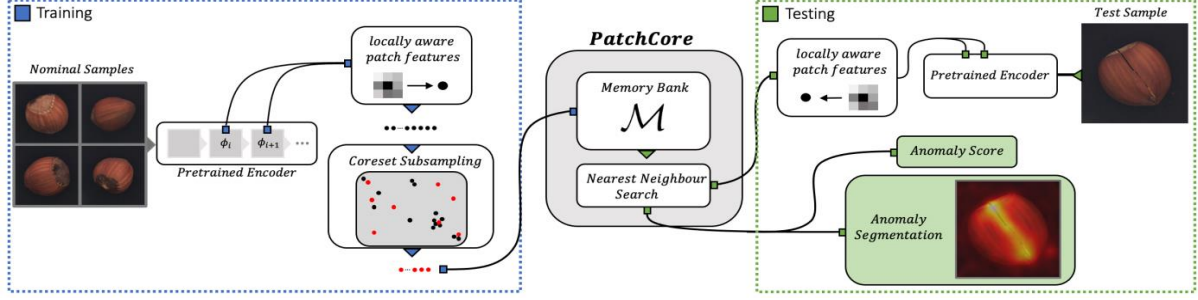


Figure 3. Pipeline of PatchCore model

PatchCore is a method within the Memory Bank technique of the feature embedding methodology. As an unsupervised anomaly detection technique, it extracts the characteristic information of normal data during the training phase. A pretrained model is used to extract the feature information of normal data, with wide resnet-50 (WRN-50) commonly employed for this purpose. This model consists of several layers, and feature information extracted from the second or third layer is typically used in the feature block. There are two main reasons for this approach. First, as the layer depth increases, there is an inevitable loss of spatial information contained in the image. Second, deeper layers introduce a greater bias from the ImageNet classification training process. Therefore, mid-level feature information is utilized. Next, a local patch block is applied to the two feature maps based on a sliding window approach. Through this process, information corresponding to $3 \times 3 \times \text{depth}$ is compressed into a single local patch feature. This operation ultimately forms a local patch feature map for an image. Such a local patch feature map contains all the local feature information of an image, with 3×3 patches commonly applied. This entire process is referred to as 'local patch feature,' and a schematic of the process is shown in Figure 4.

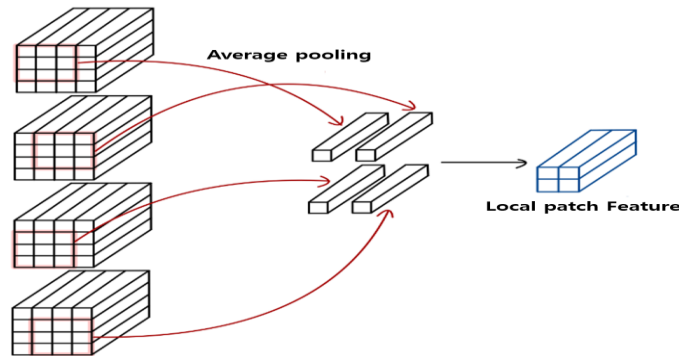


Figure 4. Process of local patch feature

Through the aforementioned computation process, two local patch feature maps are ultimately generated. Since these two feature maps result from convolution operations on different layers, they have different sizes. Therefore, an interpolation process is performed to make their sizes identical. After that, channel-wise global average pooling is applied, creating the final local patch feature map for an image. Next, this feature map is mapped to a low-dimensional space called the memory bank. During this training process, the memory bank is populated with the characteristic information of normal samples. Subsequently, instead of random subsampling, the greedy subsampling method is applied to designate representative feature information for each local area, enhancing computational efficiency. Through this entire process, a memory bank containing the representative characteristic information of normal samples is established. During the inference stage, either normal or abnormal samples can be input. The same local patch feature extraction steps are followed, and the characteristic information of the input image is mapped to the memory bank. For normal samples, all local parts will be distributed close to the learned distribution. However, for abnormal samples, the abnormal parts will be mapped far from the existing learned data distribution, enabling anomaly detection.

2.2.3. DRAEM

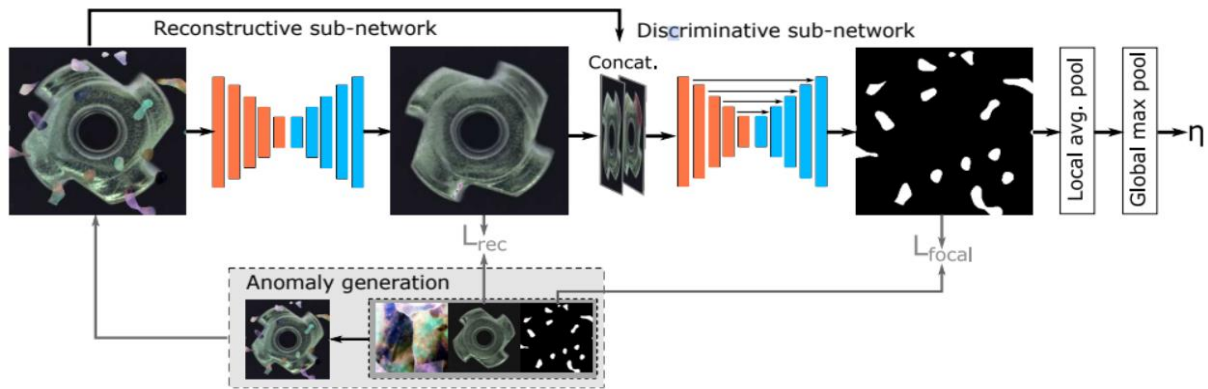


Figure 5. Pipeline of DRAEM model

The core components of the DRAEM model are composed of three parts: Anomalous Image Generation, Reconstructive Sub-network, and Discriminative Sub-network. This model is also an unsupervised anomaly detection model, utilizing only normal data as input. First, after receiving normal samples as input, the model generates abnormal data. Anomalous sample images are created by adding Perlin noise. The entire process is as follows. P refers to a random

Perlin noise image, M_a is the binarized version of this image, and A corresponds to a random image used to add anomalous characteristics to a normal sample. I represents the normal sample that is provided as input to the model.

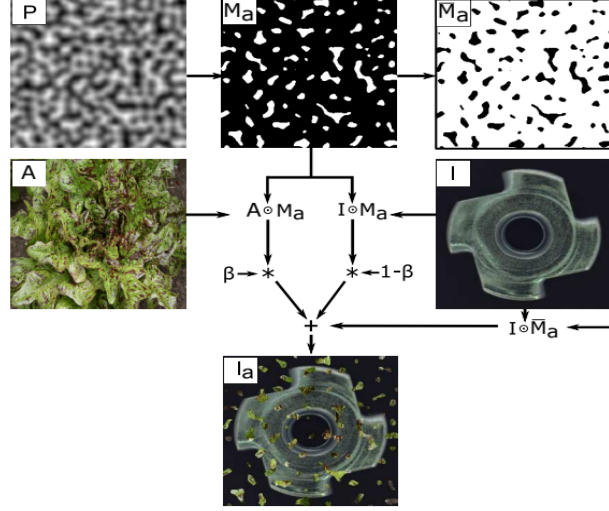


Figure 6. Process of generating anomalous image

$$I(a) = B \times (A \odot M_a) + (1 - B) \times (I \odot M_a) + I \odot M_a'$$

Next, the randomly generated anomalous image is provided as input to the reconstructive sub-network. In this process, the network learns the technique of restoring the abnormal image to a normal one. The difference between the original noise-free image I and the randomly noise-added image I_a is used as the loss. In addition to the L2 loss between the two images, the SSIM (Structural Similarity Index Measure) loss, which defines structural similarity, is also defined. Thus, the loss for the reconstructive sub-network is defined as follows.

$$L_{reconstructive} = L_2(I_a, I) + \alpha \cdot SSIM(I_a, I)$$

Next, the discriminative sub-network is trained to enhance anomaly segmentation performance based on the differences between the noisy image and the image restored by the reconstructive sub-network. Here, the network's output generates a segmentation mask. This result (M) is compared with the ground truth mask (M_a) using focal loss. Loss of discriminative sub-network and the total loss of model are defined as follows.

$$L_{discriminative} = L_{focal}(M_a, M)$$

$$L_{total} = L_{reconstructive} + L_{discriminative}$$

III. Experiments

3.1. Datasets

The MVTec dataset is the most commonly used dataset for validating model performance in the anomaly detection field. This dataset includes normal and abnormal class images, as well as ground truth images, for various items such as carpets, bottles, and wires. The original image size is 900x900, but it is typically resized to 256x256 for use with models. Each item category contains approximately 200 images in the normal dataset and around 20 images for each test class.

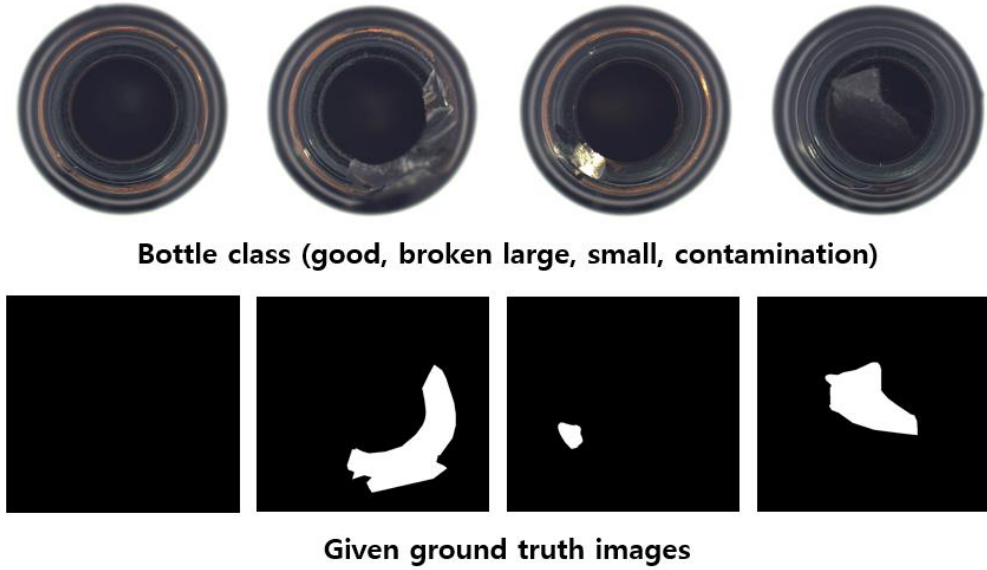


Figure 7. Example image of MVTec dataset (bottle)

The MVTec dataset is used for model validation, but practical application requires the models to be applied to the Severstal dataset. The original Severstal dataset consists of approximately 6,000 normal images and 8,000 abnormal images. The abnormal data is categorized into four classes, including localized defects such as cracks and vertical cracks, with each image having a resolution of 1600x256 pixels. While reducing the resolution to 256x256 pixels works for the MVTec dataset without losing the clarity of the defects, this approach is not suitable for the Severstal dataset. Due to the localized nature of the defects in the Severstal dataset, reducing the resolution makes it difficult to distinguish between normal and abnormal areas. To address this issue, the original 1600x256 pixel images were divided

into four 400x256 pixel images, preserving the original defect shapes. Thus, one original normal or abnormal dataset image was split into four smaller images to ensure the defects remained clearly distinguishable. This resulted in approximately 228 normal dataset images and several smaller images for the abnormal dataset. Unlike the MVTec dataset, the Severstal dataset does not provide ground truth images. Instead, encoded pixel information specifying the location of defects is available in the accompanying train.csv file. These encoded pixel data were decoded to generate ground truth images for evaluating model performance. Figure 7 shows the original form of the Severstal dataset, while Figure 8 depicts the form used for actual model application along with the corresponding decoded ground truth images.

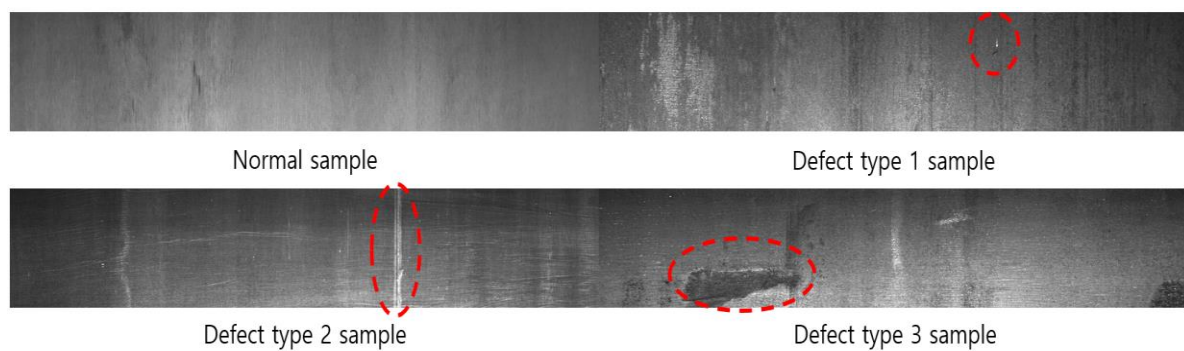


Figure 8. Example image of original Severstal dataset

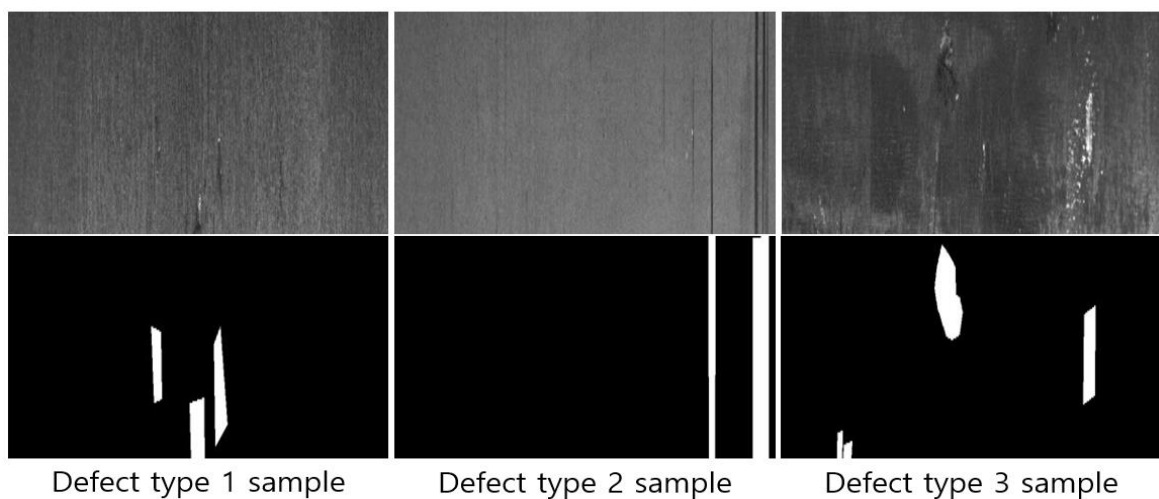


Figure 9. Example image of each defect type image and decoded ground truth image

3.2. MVTec dataset evaluation

Now, the results and performance comparison of each model applied to the MVTec dataset are presented. First, we evaluate the classification and anomaly segmentation performance of the Convolutional Autoencoder. For the Convolutional Autoencoder, as shown in Figure 9, the difference between the input image and the reconstructed image is calculated. This difference image is then binarized using a specific threshold. Next, the binarized image is compared pixel-by-pixel with the ground truth image to determine the accuracy of the match. This process evaluates the anomaly segmentation performance. Subsequently, the classification performance, which determines the presence or absence of anomalies, is evaluated. To assess classification performance, the proportion of pixels identified as defects out of the total pixels is used to determine whether an image is classified as normal or abnormal. The average classification performance is calculated by averaging the accuracy for normal images and each defect type (type 1, type 2, type 3).

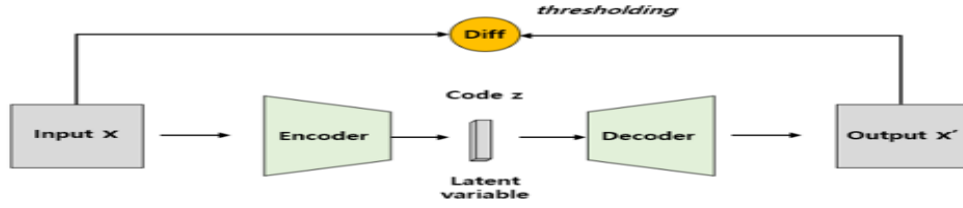


Figure 10. Process of anomaly segmentation using convolutional autoencoder

For models other than the Convolutional Autoencoder, specifically PatchCore and DRAEM, the performance evaluation concept of PixelAUROC (Area Under the Receiver Operating Characteristic Curve at the pixel level) is applied. When using PatchCore and DRAEM models for anomaly segmentation, the output is a probabilistic mask prediction indicating the likelihood of defects in certain areas. However, comparing these probability distributions directly with the ground truth is not feasible. Therefore, a TPR (True Positive Rate) curve is plotted, with the true positive rate on the y-axis and the false positive rate on the x-axis. This curve is generated by applying various thresholds to the image. A performance of 0.5, corresponding to the middle diagonal line, represents a random classifier. The closer the curve is to the y-axis, the smaller the area under the curve, indicating poor performance and an unusable model. Conversely, the closer the curve is to the y-axis, the better the segmentation performance, as the area under the curve increases. The threshold that maximizes this area is then used as the binarization threshold for image segmentation performance evaluation.

[Prediction example: Broken small class]



[Prediction example: Contamination class]



Figure 11. Prediction result of Convolutional autoencoder on MVTec dataset (Bottle)

[Prediction example: Broken large class]

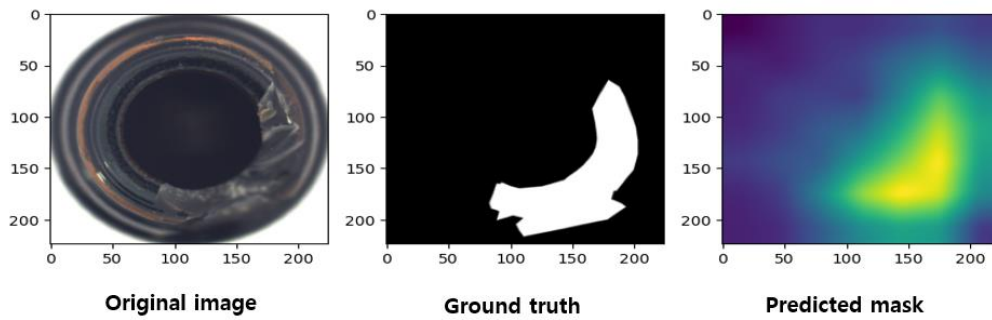


Figure 12. Prediction result of PatchCore on MVTec dataset (Bottle)

[Prediction example: Broken large class]



[Prediction example: Broken small class]



[Prediction example: Contamination class]



Figure 13. Prediction result of DRAEM on MVTec dataset (Bottle)

The results and performance comparison of the three models applied to the bottle class in MVTec dataset are as follows.

	Classification [%]	Pixel AUROC [%]
Convolutional autoencoder	86	90.42
PatchCore	97.62	93.68
DRAEM	98.3	98.26

3.3. Severstal dataset evaluation

Through the aforementioned process, the anomaly detection validation for each model on the MVTec dataset has been completed. The results of applying these models to the Severstal steel defect dataset are as follows. The models are presented in the order of Convolutional Autoencoder and PatchCore. In the case of the DRAEM model, the loss did not converge as expected, and thus, meaningful results could not be obtained. This issue will be addressed in the discussion section.

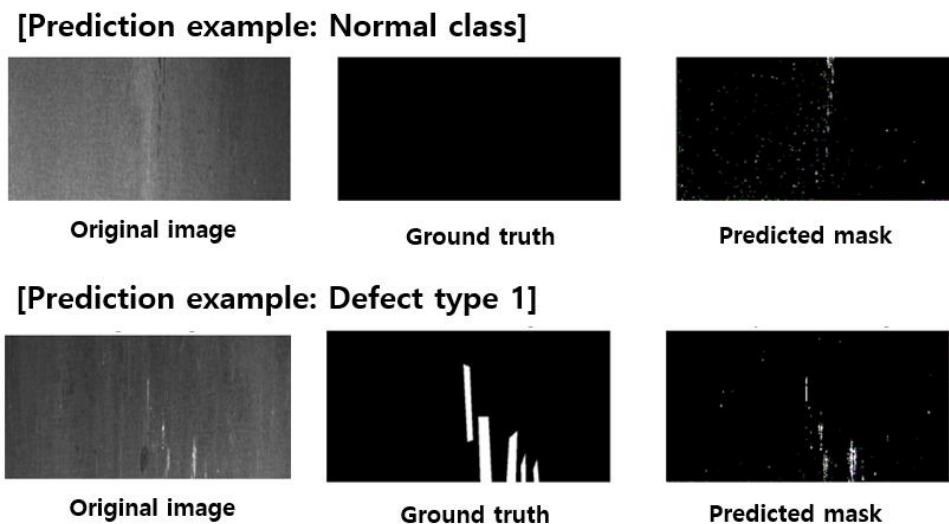


Figure 14. Prediction result of Convolutional autoencoder on Severstal dataset

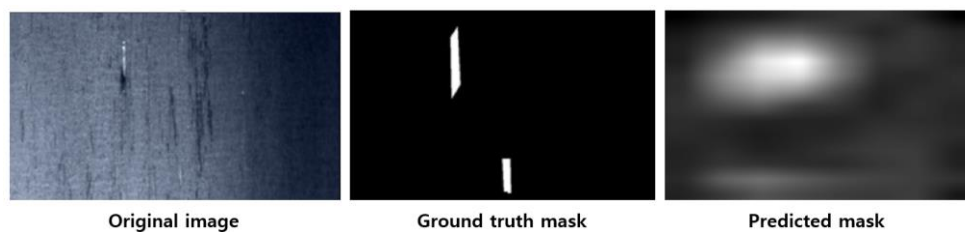


Figure 15. Prediction result of PatchCore model on Severstal dataset

	Classification [%]	Pixel AUROC [%]
Convolutional autoencoder	82.5	87.65
PatchCore	88.3	87.6
DRAEM	62.1	56.7

IV. Discussion

The main discussion focuses on the issue of decreased prediction performance in the DRAEM model. The final loss of the DRAEM model is defined as the sum of the L2 loss and SSIM loss from the reconstructive subnetwork, and the focal loss from the discriminative subnetwork. For the MVTec dataset, training is conducted for a total of 700 epochs. On the MVTec training data, the total loss converges to a range of 0.05 to 0.1 within approximately 50 epochs, demonstrating high performance. However, for the Severstal dataset, even after more than 500 epochs, the loss does not converge to the same range and continues to fluctuate. Consequently, the training performance, specifically the reconstruction and segmentation performance, appears to be low. The cause of this phenomenon might require attempts to change the dataset's structure, but such modifications could reduce the model's generalization performance. Therefore, alternative solutions are necessary.

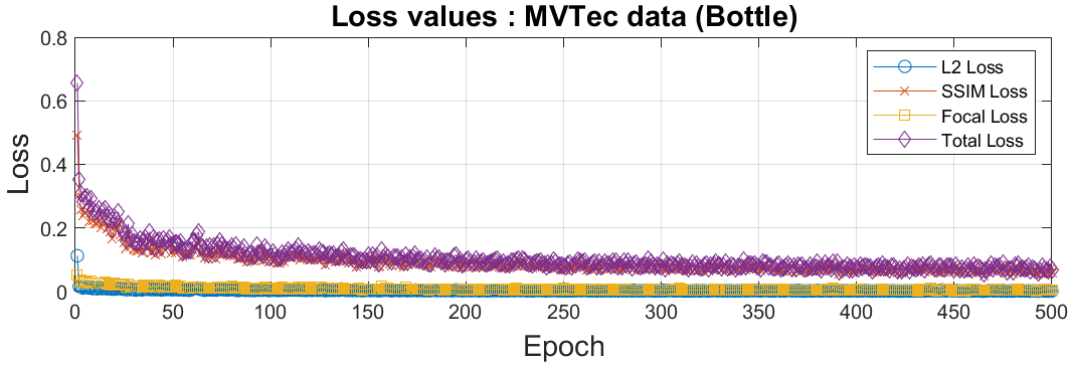


Figure 16. Learning loss convergence on the MVTec dataset

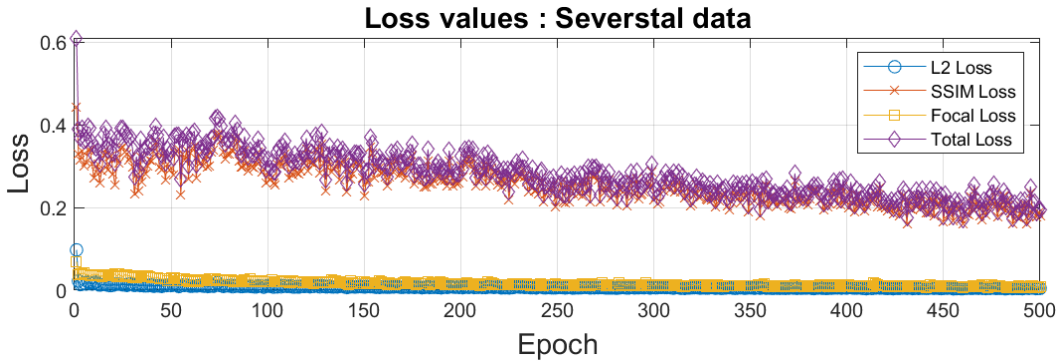


Figure 17. Learning loss convergence on the Severstal dataset

V. Conclusion

The Convolutional Autoencoder is a representative unsupervised learning-based method. This approach has the advantage of being relatively lightweight during both training and inference processes. However, it has the disadvantage of unstable classification or segmentation performance. In contrast, the PatchCore model demonstrated relatively stable segmentation performance on the same training dataset. To achieve higher performance, it would be necessary to increase the training data, but this leads to the problem of an overwhelming amount of feature information and original data to extract. Finally, the DRAEM model showed the highest performance when applied to the MVTec dataset, but it failed to produce proper results when applied to the Severstal dataset. However, with proper identification of the cause and an appropriate training process, it is possible to achieve the best performance. The DRAEM model has the advantage of high accuracy, but during the actual training and inference process, it consumed real-time GPU resources of 8GB and 12GB, respectively, with a batch size of 8. As such a heavy model, there may be limitations in deploying it in practical applications.

Reference

- [1] Liu, J., Xie, G., Wang, J., Li, S., Wang, C., Zheng, F., & Jin, Y. (2024). Deep Industrial Image Anomaly Detection: A Survey. *Machine Intelligence Research*, 21(1), 104-135. <https://doi.org/10.1007/s11633-023-1459-z>
- [2] Roth, K., Pemula, L., Zepeda, J., Scholkopf, B., Brox, T., & Gehler, P. (2022). Towards total recall in industrial anomaly detection. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). <https://doi.org/10.1109/cvpr52688.2022.01392>
- [3] Zavrtanik, V., Kristan, M., & Skocaj, D. (2021). DRÆM - a discriminatively trained reconstruction embedding for surface anomaly detection. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). <https://doi.org/10.1109/iccv48922.2021.00822>