Comparative study of unsupervised deep learning anomaly detection techniques for steel surface inspection

Min-woong Han

Supervised: Young-keun Kim Ph.D

Introduction

This study introduces the application and performance comparison of deep learning models for anomaly detection on steel. surfaces There are two main limitations in obtaining steel datasets from real industrial sites:

1. Normal data is extremely rare, making it difficult to acquire.

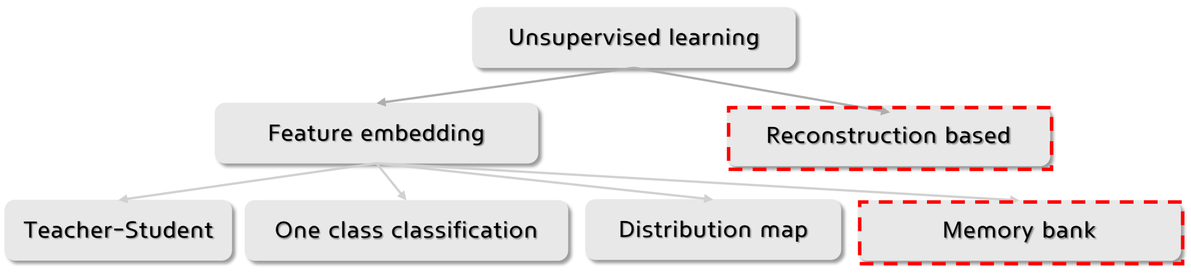
2. Defect details are generally not disclosed.

To overcome these limitations, this study conducted a comparative study of models using the open Severstal steel surface dataset. Anomaly detection methodologies are broadly divided into Unsupervised learning and Supervised learning. To address the aforementioned limitations, I applied Unsupervised learning techniques that train using only normal data. I implemented and compared the performance of classical methods, such as Convolutional Autoencoders, with state-of-the-art techniques.

Unsupervised learning

Figure 1. Pipeline of unsupervised learning

The overall pipeline of Unsupervised learning is illustrated in Figure 1. Although inputs are specified with unlabeled data, most methodologies assume that such data is normal. The key, therefore, is to directly learn the characteristic distributions of the normal data and then, during the inference stage, identify anomalies by analyzing the distribution of the input test set and detecting deviations from the normal distribution.



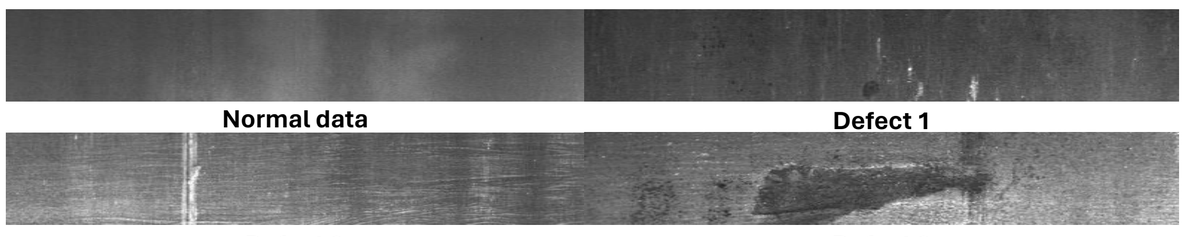
There are two main methodologies in Unsupervised learning: Feature Embedding methods and Reconstruction-based methods. Among the Feature Embedding methods, several techniques exist, with the most notable being the Teacher-Student network, One-class classification, Distribution map, and Memory bank. In this study, I implemented a technique corresponding to the Memory bank method as well as a technique from the Reconstruction-based method.

Datasets

In anomaly detection, the MVTec dataset is one of the most commonly used datasets. This dataset contains various classes such as bottle, cable, and capsule, with each class consisting of a normal dataset and an abnormal test set, which includes anomalies like broken (large and small) and contamination.

Figure 2. MVTec dataset bottle class (good, broken large, small, contamination)

The dataset intended for application to the actual model is the Severstal steel defect dataset. This dataset includes normal data and data for a total of four types of defects. Since classes 1 and 2 do not have clear distinctions, they were combined for use. Additionally, the original images are 1600x256 in size, so for computational efficiency of the model, each image was divided into four 400x256 images.



	# Train	# Test	Image size
MVTec	209	20 for each class	256x256
Severstal	228	20 for each class	400x256

Convolutional autoencoder

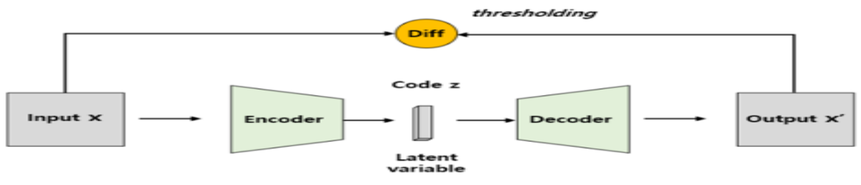


Figure 1. Pipeline of convolutional autoencoder

The Convolutional Autoencoder is a representative Reconstruction-based model. This method involves a structure where the input data undergoes feature extraction through an encoder and is then reconstructed through a decoder. The results of classification anomaly segmentation performed on the Severstal dataset using this model are as follows.

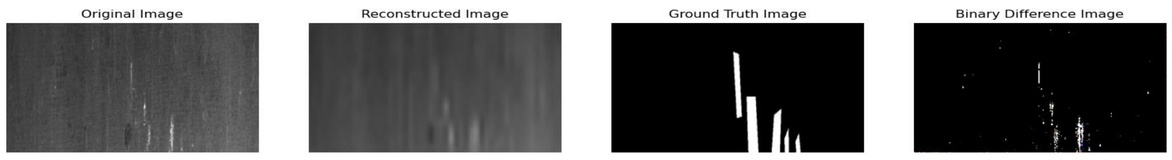


Figure 3. Anomaly segmentation performance of convolutional autoencoder

	Classification accuracy [%]	Pixel AUROC [%]
Good	80	-
Class 1	80	94.81
Class 2	85	90.36
Class 3	85	83.65

PatchCore (Memory bank)

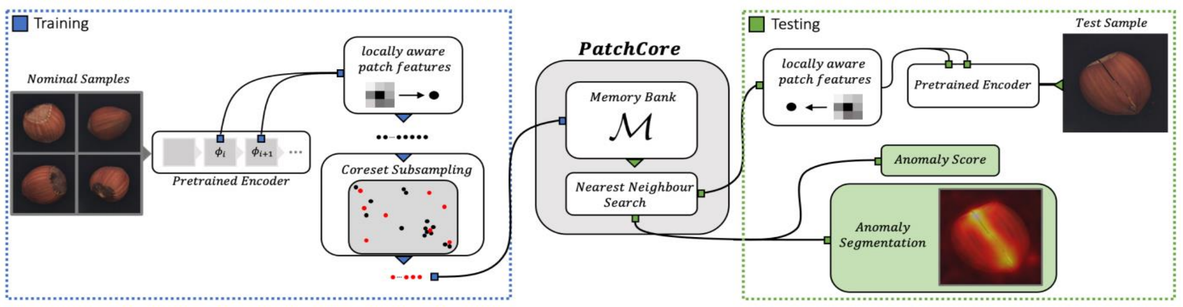


Figure 5. Pipeline of PatchCore model (Memory bank)

In the case of the Patchcore model, a pretrained model is used to extract features from the normal data, forming local patch features. Then, this feature information is mapped to construct a memory bank that contains information about the distribution of features in the normal data. Subsequently, the same process is applied to the test set to extract queries, and an anomaly score is measured by calculating the distance between the query and the existing information.

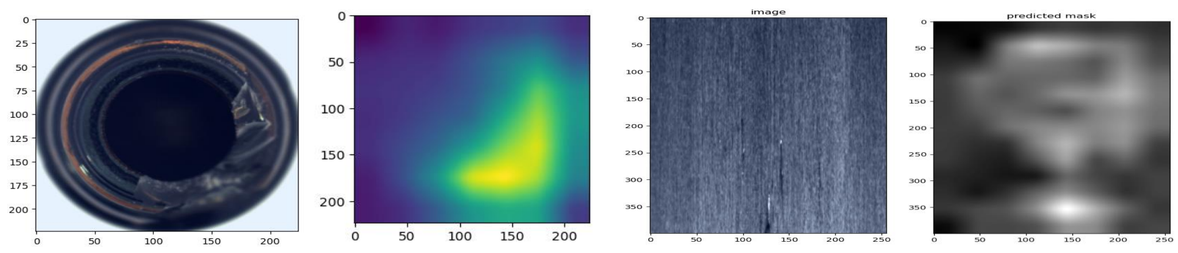
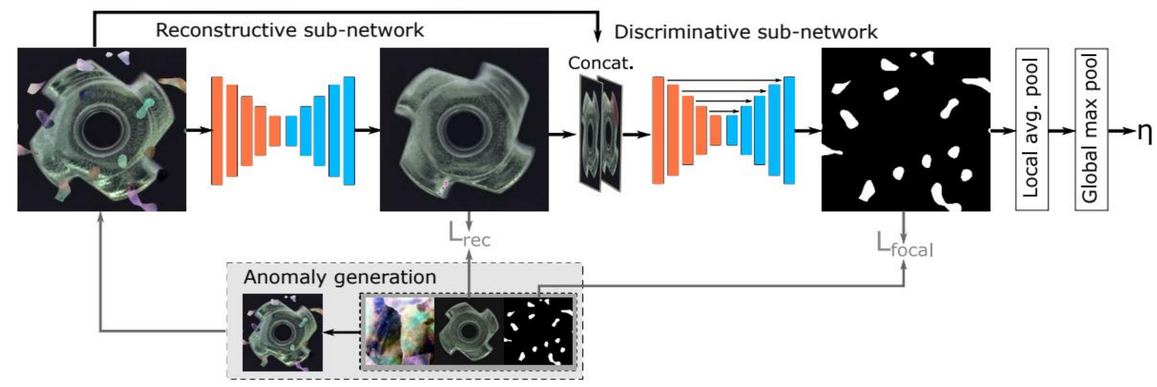
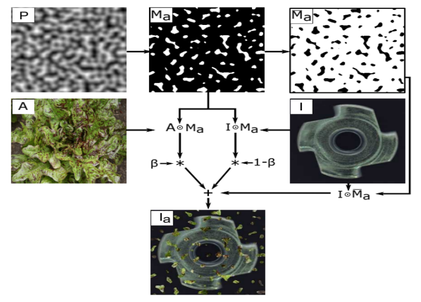


Figure 6. Anomaly detection performance of PatchCore model

	Class 1	Class 2	Class 3
Pixel AUROC [%]	87.31	84.78	81.31

DRAEM (Reconstruction)





One of the main features of the DRAEM model is that it directly generates the anomaly source and uses it as input to the model. By utilizing Perlin noise, it creates noisy images, which are then added to normal samples to generate abnormal samples.

$$I_a = B \cdot (A \times M_a) + (1 - B) \cdot (I \times M_a)$$

The DRAEM model consists of a **Reconstructive network** that restores abnormal samples to normal samples and a **Discriminative network** that performs anomaly segmentation. Let I be the sample train image, I_r be the reconstructed image, M be the final output mask, and M_a be the ground truth mask. The losses for the model are defined as follows.

$$L_{reconstructive} = \lambda L_{SSIM}(I, I_r) + l_2(I, I_r)$$

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

$$L_{total} = \lambda L_{SSIM}(I, I_r) + l_2(I, I_r) + L_{focal}(M_a, M)$$

And the following result with DRAEM model is as follows.

	Class 1	Class 2	Class 3
Pixel AUROC [%]	70.31	68.37	65.37

Conclusion

As a result of the performance evaluation, in the case of actual industrial data, the shapes of normal and defective data are extremely diverse, leading to lower performance compared to when applied to the MVTec dataset. However, by adding equally diverse training data, the model can learn the distribution of various normal data, potentially leading to further performance improvements.