Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation

Chin-Chia Tsai[1], Tsung-Hsuan Wu[1], and Shang-Hong Lai[1,2]

s108065530@m108.nthu.edu.tw, th.wu@mx.nthu.edu.tw, shlai@microsoft.com

[1]Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
[2]Microsoft AI R&D Center, Taipei, Taiwan

In this supplementary materials, we provide some additional ablation study on MVTec AD dataset and the experimental results on the challenging dataset, Magnetic Tile Surface Defects. We also illustrate our network architectures of the encoders and make an analysis on the time complexity of our model.

1. Ablation Study

1.1. Different CNN Architectures

We compare model train-from-scratch, model pretrained on VGG16 without fine-tuning, and model pretrained on different CNN architectures with fine-tuning (Table 1) in Table 2. It is obviously that without pretrained CNN architectures, the performance significantly drops for all classes. The result of model pretrained on VGG16 without fine-tuning can be regarded as the baseline performance for validating the benefit of representation learning. Our study shows that deeper CNN architectures, such as ResNet, do not produce better results. This may be caused by the simple structure of the images in the MVTec AD dataset. However, the light-weight models, such as AlexNet, do not lead to better results, either. Even though the image structure is simple, the semantics of the patches and the relationship between the patches still require an adequate depth of CNN architecture for a better feature extraction.

1.2. Different K Groups of K-means Clustering

The selection of the parameter K in the K-means clustering is another important issue in our model. In Table 3, we study the influence of difference K's through experimental evaluation of MVTec AD dataset. We observe that $K = 50$ is the sweet spot for all classes. If we set a smaller K, the clusters may not be enough to distinctly separate patch embeddings in the feature space. On the contrary, a larger K can cause the embeddings disperse too much. However, the result does not show significant difference for object classes with different K's in our model. We attribute it to the contributions of other losses, which is justified in our ablation study for each loss function as we discussed earlier.

Table 1. Comparison between different pretrained CNN architectures. The average inference time per image is calculated on MVTec Dataset.

	Model Size (MB)	Depth	Inference Time (Sec.)
AlexNet	244.80	8	1.17
MobileNet	213.78	88	2.81
VGG16 (Ours)	814.18	23	2.29
VGG19	860.43	26	2.51
ResNet18	127.35	18	2.23
ResNet50	472.52	50	3.53

Table 2. Study of the image-level anomaly detection and the pixel-level anomaly segmentation performance with model train-from-scratch, pretrained on VGG16 without fine-tuning, and pretrained on different CNN architectures with fine-tuning on MVTec AD dataset. The results are reported with AUROC%.

Class	All Texture Classes		All Object Classes		All Classes	
Task	**det.**	**seg.**	**det.**	**seg.**	**det.**	**seg.**
Train-from-scratch	94.6	89.7	87.1	95.0	89.6	93.2
VGG16 w/o fine-tune	95.5	87.4	89.7	89.6	91.6	88.9
AlexNet	96.0	94.9	93.2	97.6	94.1	96.7
MobileNet	96.2	97.0	96.5	98.1	96.4	97.7
VGG16 (Ours)	97.7	**97.6**	**98.4**	**98.4**	**98.1**	**98.1**
VGG19	95.5	94.4	97.6	98.2	96.9	96.9
ResNet18	**98.8**	96.6	94.5	97.7	95.9	97.4
ResNet50	97.3	96.1	93.4	97.3	94.7	96.9

Table 3. Study of the image-level anomaly detection and the pixel-level anomaly segmentation performance with different K groups of K-means on MVTec AD dataset. The results are reported with AUROC%.

Class	All Texture Classes		All Object Classes		All Classes	
Task	**det.**	**seg.**	**det.**	**seg.**	**det.**	**seg.**
K=10	97.0	95.0	97.6	98.0	97.4	97.0
K=30	95.3	95.9	96.8	97.6	96.3	97.0
K=50 (Ours)	**97.7**	**97.6**	**98.4**	**98.4**	**98.1**	**98.1**
K=100	96.2	95.7	97.9	98.3	97.3	97.4

Table 4. Comparison of our models with the SOTA methods for both the image-level anomaly detection and the pixel-level anomaly localization performance on Magnetic Tile Surface Defects dataset. The results are reported with AUROC%.

	det.	seg.
MCuePushU [2] (supervised)	-	**98.5**
DifferNet [3] (unsupervised)	97.7	-
Ours (unsupervised)	**98.7**	74.0

2. Magnetic Tile Surface Defects

The Magnetic Tile Surface Defects dataset is released by [2] in IEEE. It is a challenging dataset with 1344 grayscale images, which contains 952 defect-free images and 392 defective images. The defect types are $Uneven$, $Crack$, $Fray$, $Blowhole$, and $Break$ respectively. All the images have different resolution and illuminations. We resize all the images to 128×128 in our experiments.

We compare our model with MCuePushU [2] and DifferNet [3] in Table 4. MCuePushU [2] performs a good anomaly segmentation result since it uses the supervised learning approach, and our model adopts the unsupervised learning method instead. It can be observed that our model performs well on the anomaly detection task yet we struggle on the anomaly segmentation task. This is due to the different illuminations among images. We do not have a specific strategy to make our model pay attention to those illuminations, and thus, our model tends to recognize the bright regions as anomalies in most cases. Figure 1 shows some example results of well detected cases and failure cases.

3. Network Architecture

The detailed architectures of our encoder Enc_{64}, Enc_{32}, and Enc_{16} are shown in Figure 2. The convolutional kernels are different layer by layer. The paddings are all set to 1 and we do not apply dilation. Every convolutional layer is followed by a leaky rectified linear unit (LeakyReLU) activation function.

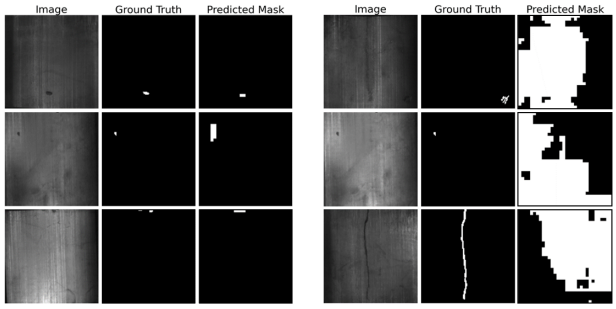

Figure 1. Examples results from Magnetic Tile Surface Defects dataset. The left part shows the well detected examples and the right part is the failure cases. The predicted anomaly segmentation maps are obtained by applying the proposed system. We compare our predicted masks with MCuePushU [2] and DifferNet [3].

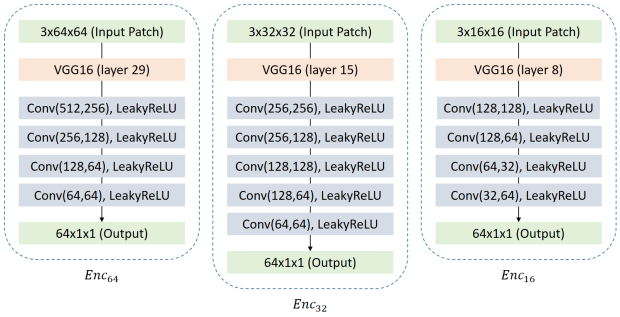

Figure 2. The detailed architecture of our encoders, Enc_{64}, Enc_{32}, and Enc_{16}. The selection of the layer in VGG16 is according to different patch size. Note that all encoders output $64 \times 1 \times 1$ feature for the same architecture of classifier C.

Our classifier C is constructed by 3 fully-connected layers followed by LeakyReLU except the last layer.

Table 5. Average inference time of testing images on the MVTec AD dataset with a CPU Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz and GPU GeForce GTX 1080 Ti. The inference time is calculated as $total\ inference\ time/number\ of\ images$.

	PaDiM [1]	Patch SVDD [4]	Ours (2-scale)	Ours (3-scale)
seconds per image	0.95	1.09	1.81	2.29

4. Time Complexity

During the training procedure, our model is easy to train since it uses pretrained CNN architectures and the encoders are tiny with simple architectures. For the inference phase, our model takes longer time than the previous state-of-the-art method, PaDiM [1], and the method Patch SVDD [4], as shown in Table 5. This is mainly caused by our multi-scale patches architecture. With more different scales or smaller sizes of patches, our inference time increases. It

can be regarded as a trade-off between the accuracy and the time complexity.

References

[1] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. *ICPR*, 2020.

[2] Yibin Huang, Congying Qiu, Yue Guo, Xiaonan Wang, and Kui Yuan. Surface defect saliency of magnetic tile. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pages 612–617, 2018.

[3] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2021.

[4] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.