ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888690

DN-DR: Discriminative Network with Dual Reconstruction for Image Anomaly Detection

Wen Li
Institute of Information Science
Beijing Jiaotong University
Beijing, China
wenwenli@bjtu.edu.cn

Chune Zhang*
Institute of Information Science
Beijing Jiaotong University
Beijing, China
chezhang@bjtu.edu.cn

***Abstract*—One mainstream of image anomaly detection is based on reconstruction. Such methods still struggle with diverse anomalies, such as near-in-distribution or deformed types. To address the challenge, we propose a Discriminative Network with Dual Reconstruction (DN-DR), consisting of a Memory Reconstructor, a Corrector, and a Discriminator. DN-DR aims to better restore the defective image to its normal state through dual reconstruction, thereby obtaining superior Discriminator performance. Specifically, (1) the Memory Reconstructor is based on training multi-scale codebooks from normal images to rebuild unknown regions in the test images, also named preliminary reconstruction; (2) the Corrector, as a subsequent reconstruction module, addresses false anomalies caused by the patch-level replacement strategy in the Memory Reconstructor, achieving a final refined reconstruction; (3) a U-Net Discriminator follows. Experiments on the challenging MVTec AD dataset demonstrate excellent reconstruction performance and anomaly inspection, including defects of near-in-distribution or deformed types.**

***Index Terms*—anomaly detection, reconstruction-based methods, dual reconstruction.**

I. Introduction

Image anomaly detection is the technique of identifying patterns in images that deviate from the norm to prevent potential risks [1]. A typical scenario is industrial surface defect detection and localization. In large-scale industrial manufacturing, accurately detecting and locating defects is crucial for quality control [2]. However, surface defects are unpredictable and often vary by product category. For example, common defects in carpets are holes and threads, while capsules may have cracks and scratches. Therefore, a large number of abnormal samples are required for defect classification, detection, and segmentation. In actual production, the number of abnormal samples is relatively small, and artificially producing them is impractical. Meanwhile it is time-consuming to label samples. Clearly, traditional supervised methods are not feasible [3].

There is a growing emphasis on unsupervised training methods. Three main trends are synthesizing-based methods [4]–[8], embedding-based methods [2], [5], [9]–[13], and reconstruction-based methods [14]–[18], and so on. Compared to the others, reconstruction-based methods are more intuitive and interpretable. It is assumed that a model trained on only normal data can effectively restore normal regions of the input images but struggles with anomalies, allowing us to identify defects based on reconstruction differences.

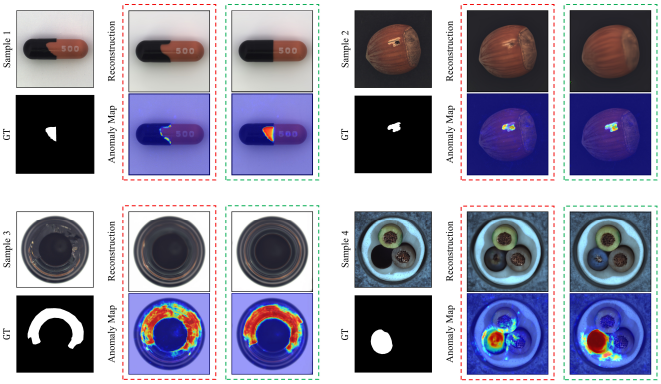

Fig. 1. Impact of improved reconstruction. For each example, from left to right, they are the sample image and its ground-truth mask (GT), the result of DRÆM [4], and the result of DN-DR (ours). DN-DR outperforms DRÆM [4] in restoring sample images, leading to improved anomaly inspection.

However, if the model generalizes too well, even the abnormal regions could be well reconstructed, leading to failures in reconstruction-based methods. Researchers are working to limit the generalization ability of reconstruction models [19]. [14]–[16] use memory banks to limit image reconstruction. Although effective, these methods do not learn anomalies during training, which can lead to potential detection failures in real-world scenarios. [4], [7] use synthetic data to train models for restoring defective images to normal instead of poorly reconstructing anomalies. However, due to limitations in design or anomaly simulation, these models struggle with complex anomalies, such as those near in distribution (i.e., highly similar to normal appearances) or just deformed.

This paper further studies the reconstruction-based methods and proposes a Discriminative Network with Dual Reconstruction (DN-DR), which consists of a Memory Reconstructor, a Corrector, and a Discriminator. DN-DR restores images well, thus enhancing anomaly detection and localization. Our method is based on three interactive considerations. **First,** anomalies are unpredictable, which makes it hard to cover all scenarios during training. The task becomes easier when unknown problems are transformed into known ones through a medium. Therefore, we use multi-scale codebooks to learn limited patterns from normal samples, aiming to replace all unknown patterns in the test images. This is the preliminary reconstruction, performed by a Memory Reconstructor.

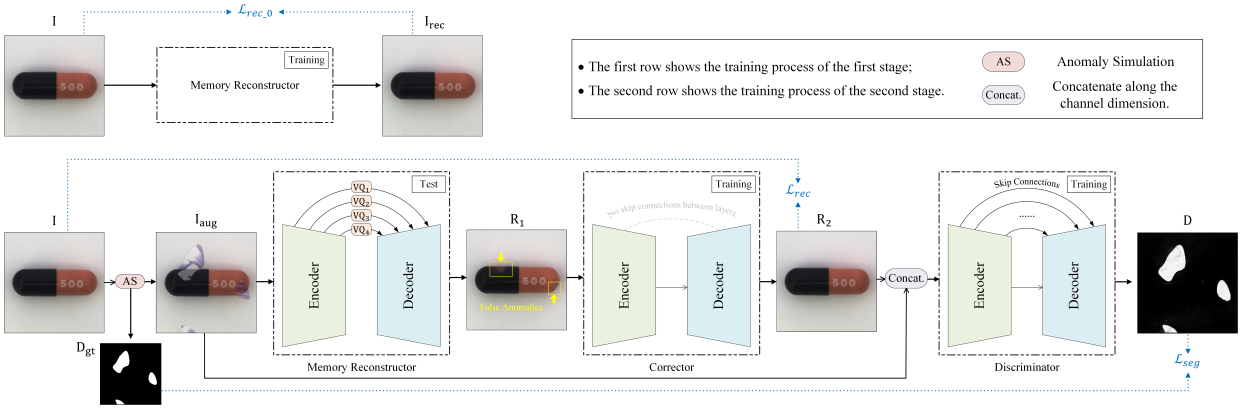

Fig. 2. Overview of the proposed method. The two rows show the two-stage training of DN-DR. Stage 1: The Memory Reconstructor is trained on original normal images. Stage 2: The Corrector and the Discriminator are trained on synthetic anomalous images. We adopt the idea of [4] for anomaly simulation.

Second, the initial reconstruction focuses on local features rather than individual pixels for more effectively removing anomalies. However, this strategy is limited and may lead to problems such as incoherence between regions and deviations in details, called false anomalies. So it is necessary to perform further fine-grained reconstruction through a Corrector. **Third,** understanding false anomalies improves the Corrector, and existing anomaly simulation strategies contribute to this. We design a simple Corrector for practicality. Experiments on the MVTec AD dataset [20] show DN-DR's excellent performance in reconstruction as well as in anomaly detection and localization. Fig. 1 illustrates the impact of improved reconstruction.

II. DN-DR

As shown in Fig. 2, DN-DR comprises three main modules: a Memory Reconstructor, a Corrector, and a Discriminator. The Memory Reconstructor and the Corrector work together for dual reconstruction. The Discriminator outputs the results of anomaly detection and localization by comparing the refined reconstructions with the test images.

A. Memory Reconstructor

Fig. 3 shows the Memory Reconstructor in training mode. We extract multi-scale global feature maps $\mathrm{E}_i \in \mathbb{R}^{H_i \times W_i \times C_i}$ from the input image $\mathrm{I} \in \mathbb{R}^{H \times W \times 3}$, where $1 \leq i \leq L$ denotes the i-th scale. A separate codebook is trained at each scale for high-resolution reconstruction, and multi-scale information is aggregated. The process flow remains consistent across all scales (see Fig. 4). Specifically, (1) the global feature map E_i is divided into local feature maps, which are then flattened into one-dimensional vectors; (2) each one-dimensional vector is replaced with the most similar embedding vector from the codebook; (3) the inverse of the first step is performed to obtain a new global feature map $\mathrm{Q}_i \in \mathbb{R}^{H_i \times W_i \times C_i}$ in preparation for multi-scale feature fusion. In the following, we refer to the above steps as the vector quantization (VQ) [7], [21] process. Key operations are detailed below.

Division and Flattening of Feature Maps. For each global feature map E_i, we divide it into $n = r_h r_w$ local feature maps $\mathrm{F}_i^j \in \mathbb{R}^{\frac{H_i}{r_h} \times \frac{W_i}{r_w} \times C_i}$ $(j = 1, 2, \ldots, n)$, where r_h and r_w are the division rates along height and width respectively.

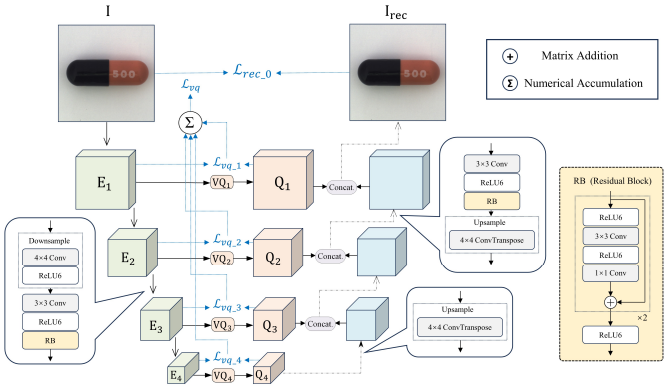

Fig. 3. Details of the Memory Reconstructor. Identical arrows indicate the same operations.

To simplify subsequent processing, each local feature map is flattened into a one-dimensional vector, also named query vector $q_i^j \in \mathbb{R}^{\frac{H_i W_i C_i}{r_h r_w}}$ $(j = 1, 2, \ldots, n)$.

Details of Codebooks. The codebook $\mathcal{K}_i \in \mathbb{R}^{N \times D_i}$ is defined as a real-valued matrix containing N embedding vectors $e_i^k \in \mathbb{R}^{D_i}$ $(k = 1, 2, \ldots, N)$, where D_i matches the dimension of the query vector q_i^j. Note that codebooks are not shared across scales due to varying local feature map sizes.

Method of Replacement. It is formulated as:

$$(q_i^j)' = e_i^l = \underset{e_i^k \in \mathcal{K}_i}{\arg\min}(\|q_i^j - e_i^k\|_2), \tag{1}$$

where $(q_i^j)'$ is the result of replacing q_i^j. $\|\cdot\|_2$ denotes the Euclidean distance between two vectors, with smaller values indicating higher similarity.

Fusion of Multi-scale Features. After processing with the VQ module (see Fig. 4), we obtain new global feature maps $\{\mathrm{Q}_1, \mathrm{Q}_2, \ldots, \mathrm{Q}_L\}$. We begin by upsampling Q_L to match the size of Q_{L-1} and then concatenate them along the channel dimension to form $\mathrm{Q}_{\text{concat}(L-1,L)}$. Next, we apply a series of operations (see Fig. 3) to halve the channels of $\mathrm{Q}_{\text{concat}(L-1,L)}$, completing the initial fusion. Similar steps are repeated for the remaining scales until all global feature maps are fused.

B. Corrector

As stated above, codebooks store local feature maps, not individual pixels. While the former captures rich local information and is more representative of patterns in the image, it

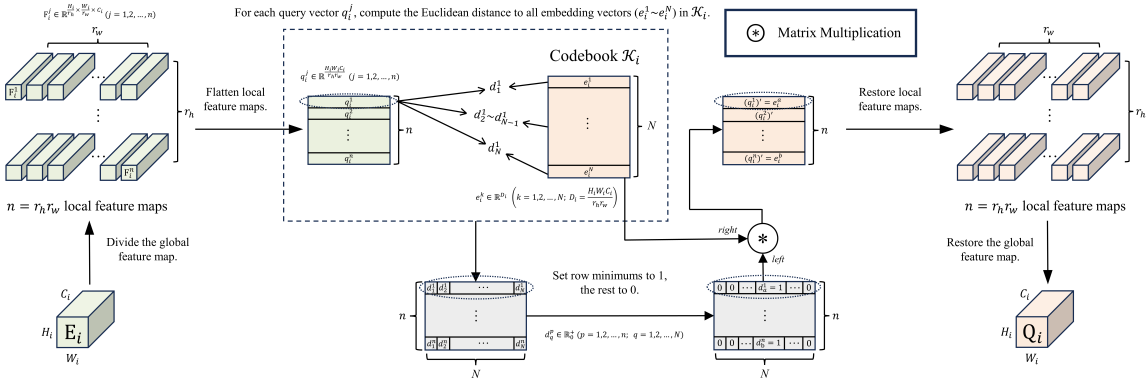

Fig. 4. Details of the overall VQ process.

is less likely that anomalies and normality share the same patterns [16]. Actually, the patch-level replacement strategy has limitations. Specifically, during VQ, specific-sized patches are directly replaced in E_i, which can result in detail deviations. Moreover, since each patch is replaced independently, this may lead to incoherence between regions of Q_i. Such limitations prevent us from using the preliminary reconstruction R_1 as final. Therefore, a Corrector is used to refine the reconstruction.

The Corrector is important in DN-DR and has a simple design. It consists only of multi-layer symmetric encoders and decoders, trained simultaneously. The training aims to minimize the reconstruction loss $\mathcal{L}_{rec}$, which measures the discrepancy between the final refined reconstruction R_2 and the target (i.e., the original input I), as detailed in (4).

C. Discriminator

To adaptively distinguish the differences between the final refined reconstruction R_2 and the test image I_{aug}, we concatenate them along the channel dimension and feed the result into a U-Net [22] Discriminator. This architecture retains detailed information through skip connections for finer segmentation. The Discriminator directly outputs a pixel-level anomaly score map $\mathrm{D} \in \mathbb{R}^{H \times W}$. To determine the image-level anomaly score $s \in \mathbb{R}$, we smooth the map with a 21×21 mean filter and then take the maximum value, i.e., $s = \max(Avgpool_{21\times 21}(\mathrm{D}))$.

D. Loss Function and Training Procedure

The training of DN-DR involves two stages. In the **first stage**, we train the codebook, encoder and decoder of each layer of the Memory Reconstructor on normal images I. In codebook training, traditional methods use a loss term to further assist in updating embedding vectors, while we use EMA (Exponential Moving Average) [21]. EMA replaces the current value with a weighted average of data from the previous K periods, where weights decay exponentially as the time interval increases. This makes updating faster and more stable. The overall loss function here consists of two losses: reconstruction loss $\mathcal{L}_{rec\_0}$ and quantization loss $\mathcal{L}_{vq}$, i.e.,

$$\begin{aligned}\mathcal{L}_{stage1} &= \mathcal{L}_{rec\_0} + \lambda \mathcal{L}_{vq} = [\mathcal{L}_{mse}(\mathrm{I}, \mathrm{I}_{\text{rec}}) + \mathcal{L}_{ssim}(\mathrm{I}, \mathrm{I}_{\text{rec}})] \\ &+ \lambda \sum_{i=1}^{L} \{\mathcal{L}_{mse}(\mathrm{E}_i, sg[\mathrm{Q}_i]) + \boxed{\mathcal{L}_{mse}(sg[\mathrm{E}_i], \mathrm{Q}_i)}\},\end{aligned} \tag{2}$$

where the loss term in the $\boxed{\text{box}}$ is discarded due to the use of EMA, λ controls the proportion of the two losses, $sg[\cdot]$ is the stop-gradient operator [7], [21], $\mathcal{L}_{mse}(\cdot,\cdot)$ is the MSE (Mean Square Error) loss function, and $\mathcal{L}_{ssim}(\cdot,\cdot) = 1 - ssim(\cdot,\cdot)$ where $ssim(\cdot,\cdot)$ measures the structural similarity [23], [24]:

$$ssim(\mathrm{I}, \mathrm{I}_{\text{rec}}) = \frac{[2E(\mathrm{I})E(\mathrm{I}_{\text{rec}}) + \alpha][2Cov(\mathrm{I}, \mathrm{I}_{\text{rec}}) + \beta]}{[E(\mathrm{I})^2 + E(\mathrm{I}_{\text{rec}})^2 + \alpha][D(\mathrm{I}) + D(\mathrm{I}_{\text{rec}}) + \beta]}. \tag{3}$$

Here, $E(\cdot)$, $D(\cdot)$, and $Cov(\cdot,\cdot)$ stand for the mean, variance, and covariance, respectively. Constants α and β ensure numerical stability. The combined use of the loss terms in (2) helps effectively generate an output I_{rec} close to the input I, while ensuring high-quality VQ at each scale.

In the **second stage**, we train both the Corrector and the Discriminator. First, we perform anomaly simulation [4] on normal images I, adding anomalies randomly. Those without anomalies are used as positives. Later, the synthetic data I_{aug} is sequentially processed through the Memory Reconstructor (in test mode), Corrector, and Discriminator. Parameter updates are guided by $\mathcal{L}_{stage2}$, which also consists of two losses: reconstruction loss $\mathcal{L}_{rec}$ and segmentation loss $\mathcal{L}_{seg}$, i.e.,

$$\begin{aligned}\mathcal{L}_{stage2} = \mathcal{L}_{rec} + \mathcal{L}_{seg} &= [\mathcal{L}_{mse}(\mathrm{I}, \mathrm{R}_2) + \mathcal{L}_{ssim}(\mathrm{I}, \mathrm{R}_2)] \\ &+ \mathcal{L}_f(\mathrm{D}, \mathrm{D}_{\text{gt}}),\end{aligned} \tag{4}$$

where $\mathrm{D}_{\text{gt}} \in \mathbb{R}^{H \times W}$ is the ground-truth mask obtained from anomaly simulation. The focal loss [25] $\mathcal{L}_f(\cdot,\cdot)$ emphasizes hard-to-detect instances to increase segmentation robustness:

$$\mathcal{L}_f(\mathrm{D}, \mathrm{D}_{\text{gt}}) = -\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (1 - p_{hw})^{\gamma} \log(p_{hw}), \tag{5}$$

$$p_{hw} = \mathrm{D}_{\text{gt}}^{(h,w)} \mathrm{D}^{(h,w)} + (1 - \mathrm{D}_{\text{gt}}^{(h,w)})(1 - \mathrm{D}^{(h,w)}), \tag{6}$$

where $\mathrm{D}_{\text{gt}}^{(h,w)}$ denotes the pixel value of D_{gt} at (h, w), and $\mathrm{D}^{(h,w)}$ has a similar definition. γ is the focusing parameter.

III. Experiment

A. Experimental Settings

Dataset. The MVTec AD dataset [20] includes 15 categories (5 textures, 10 objects). The training set contains only normal images. The test set has both normal and abnormal images.

Training Details. For each category, we perform a two-stage training (see Section II-D). In the first stage, training is set to 200 epochs with a batch size of 16. The codebook size N is experimentally set to 1024. r_h and r_w are both set to 8. The learning rate η_1 is set to 0.0002 and is multiplied by 0.1 after 160 epochs. In the second stage, to enhance model robustness,

TABLE I
RESULTS (I-AUROC% / P-AP%) ON THE MVTEC AD DATASET. BEST AND SECOND BEST SCORES ARE **HIGHLIGHTED** AND UNDERLINED.

Method	Category															*Average*
	bottle	capsule	grid	leather	pill	tile	transistor	zipper	cable	carpet	hazelnut	metal nut	screw	toothbrush	wood	
MemAE [14]	95.4/-	83.1/-	94.6/-	61.1/-	88.3/-	63.0/-	79.3/-	87.1/-	69.4/-	45.4/-	89.1/-	53.7/-	**99.2**/-	97.2/-	96.7/-	80.2/-
US [11]	99.0/74.2	86.1/25.9	81.0/10.1	88.2/40.9	87.9/**62.0**	99.1/65.3	81.8/27.1	91.9/36.1	86.2/48.2	91.6/52.2	93.1/57.8	82.0/83.5	54.9/7.8	95.3/37.7	97.7/53.5	87.7/45.5
RIAD [17]	**99.9**/76.4	88.4/38.2	99.6/36.4	**100**/49.1	83.8/51.6	98.7/52.6	90.9/39.2	98.1/63.4	81.9/24.4	84.2/61.4	83.3/33.8	88.5/64.3	84.5/43.9	**100**/50.6	93.0/38.2	91.7/48.2
PaDiM [10]	**99.9**/77.3	91.3/46.7	96.7/35.7	**100**/53.5	93.3/61.2	98.1/52.4	**97.4/72.0**	90.3/58.2	92.7/45.4	**99.8**/60.7	92.0/61.1	98.7/77.4	85.8/21.7	96.1/54.7	99.2/46.3	95.4/55.0
DAAD+ [16]	97.6/-	76.7/-	95.7/-	86.2/-	90.0/-	88.2/-	87.6/-	85.9/-	84.4/-	86.6/-	92.1/-	75.8/-	98.7/-	99.2/-	98.2/-	89.5/-
CutPaste [6]	98.2/-	98.2/-	**100**/-	**100**/-	94.9/-	94.6/-	96.1/-	99.9/-	81.2/-	93.9/-	98.3/-	**99.9**/-	88.7/-	99.4/-	99.1/-	96.2/-
DRÆM [4]	99.2/86.5	**98.5**/49.4	99.9/65.7	**100/75.3**	**98.9**/48.5	99.6/92.3	93.1/50.7	**100/81.5**	91.8/52.4	97.0/53.5	**100/92.9**	98.7/**96.3**	93.9/**58.2**	**100**/44.7	99.1/77.7	**98.0**/68.4
Ours	99.3/**90.4**	97.6/**53.5**	99.8/**66.3**	**100**/74.3	92.1/49.7	**100/95.6**	95.4/55.2	99.7/63.3	**94.0/55.3**	98.1/**69.2**	**100**/89.5	99.2/92.8	86.8/44.6	99.4/**57.2**	**99.6/77.9**	97.4/**69.0**

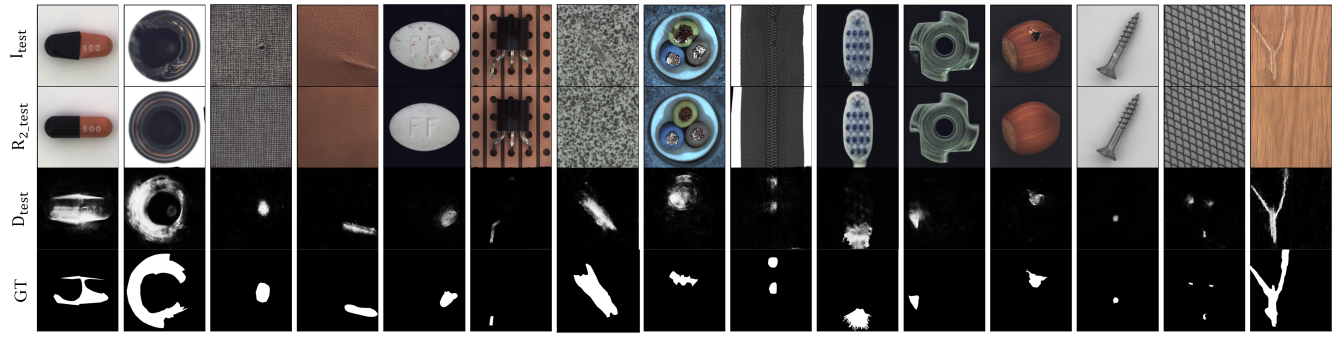

Fig. 5. Qualitative results. From top to bottom, they are the test images, the final refined reconstructions, the pixel-level anomaly score maps, and the ground-truth masks. In some categories, especially with repeating textures (e.g., leather, tile), DN-DR will blur information that interferes with detection.

we randomly rotate the training images in the range of [-90, 90] degrees. This stage trains for 1,000 epochs with a batch size of 4. The initial learning rate η_2 is 0.0001, and it decays by a factor of 10 at 600 and 800 epochs, respectively.

B. Experimental Results

The widely used image-level AUROC (I-AUROC) metric is applied to evaluate the anomaly detection performance. For anomaly localization, we calculate the pixel-level AP (P-AP), which is suitable for highly imbalanced categories [7], [26].

Qualitative Reconstruction. As shown in Fig. 6, we compare the reconstructions from DN-DR with those from existing state-of-the-art methods [4], [7]. The comparison indicates that DN-DR is more effective in restoring defective images, including images with near-in-distribution defects or deformed types of anomalies, to their normal state. This improvement is particularly noticeable in the reconstruction instances of capsules, bottles, hazelnuts, transistors, and more.

Anomaly Detection and Localization. The quantitative results of anomaly detection and localization on the MVTec AD dataset [20] are summarized in Table I, while qualitative results are shown in Fig. 5. To ensure a fair comparison, our DN-DR is primarily compared with advanced reconstruction-based methods and synthesizing-based methods, as we do not use large-scale pre-trained models. DN-DR achieves the best results in several image categories. The advantages of DN-DR are more clearly demonstrated in anomaly localization, where its average pixel-level AP across 15 categories reaches 69.0%, outperforming other methods.

C. Ablation Study

To verify the performance and role of each module in DN-DR, we conduct ablation studies (see Table II). Experiments

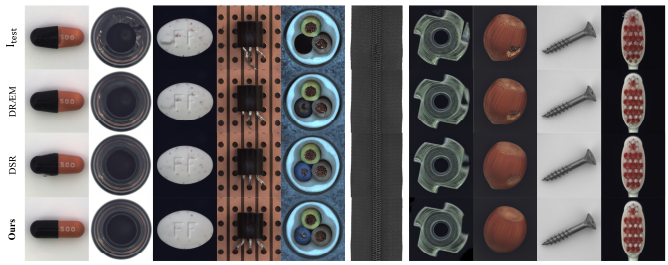

Fig. 6. Qualitative reconstruction results on the MVTec AD dataset [20]. The first row is the test images. The second, third, and fourth rows are the final reconstructions from DRÆM [4], DSR [7], and DN-DR (ours), respectively. DN-DR demonstrates its superior image restoration capability.

TABLE II
ABLATION STUDIES. DET. AND LOC. REFER TO THE AVERAGE I-AUROC% AND AVERAGE P-AP%, RESPECTIVELY.

	Memory Reconstructor	Corrector	Discriminator	*Det. / Loc.*
1	✓		✓	94.8 / 61.8
2		✓	✓	97.0 / 67.5
3			✓	96.4 / 57.7
4	✓	✓	✓	**97.4 / 69.0**

1 to 4 show that the complete DN-DR system performs best in both anomaly detection and anomaly localization. The Discriminator is indispensable, so we keep it in all experiments.

IV. CONCLUSION

In this paper, we propose a Discriminative Network with Dual Reconstruction (DN-DR) for image anomaly detection and localization. The Memory Reconstructor collaborates with the Corrector to achieve dual reconstruction, generating refined reconstructed images that improve the Discriminator's performance. Experiments on the MVTec AD dataset [20] demonstrate the excellent performance of DN-DR.

REFERENCES

[1] O. Rippel, P. Mertens, and D. Merhof, "Modeling the distribution of normal data in pre-trained deep features for anomaly detection," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 6726–6733.

[2] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 14 318–14 328.

[3] S. Akcay, D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja, and U. Genc, "Anomalib: A deep learning library for anomaly detection," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 1706–1710.

[4] V. Zavrtanik, M. Kristan, and D. Skočaj, "DRÆM — a discriminatively trained reconstruction embedding for surface anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8330–8339.

[5] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, "SimpleNet: A simple network for image anomaly detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 402–20 411.

[6] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "CutPaste: Self-supervised learning for anomaly detection and localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9664–9674.

[7] V. Zavrtanik, M. Kristan, and D. Skocaj, "DSR — a dual subspace re-projection network for surface anomaly detection supplementary material," 2022.

[8] X. Zhang, M. Xu, and X. Zhou, "RealNet: A feature selection network with realistic synthetic anomaly for anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 699–16 708.

[9] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," *arXiv preprint arXiv:2005.02357*, 2020.

[10] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *International Conference on Pattern Recognition*. Springer, 2021, pp. 475–489.

[11] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed Students: Student-teacher anomaly detection with discriminative latent embeddings," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4183–4192.

[12] X. Zhang, S. Li, X. Li, P. Huang, J. Shan, and T. Chen, "DeSTSeg: Segmentation guided denoising student-teacher for anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3914–3923.

[13] K. Batzner, L. Heckler, and R. König, "EfficientAD: Accurate visual anomaly detection at millisecond-level latencies," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 128–138.

[14] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1705–1714.

[15] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 372–14 381.

[16] J. Hou, Y. Zhang, Q. Zhong, D. Xie, S. Pu, and H. Zhou, "Divide-and-Assemble: Learning block-wise memory for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 8791–8800.

[17] V. Zavrtanik, M. Kristan, and D. Skočaj, "Reconstruction by inpainting for visual anomaly detection," *Pattern Recognition*, vol. 112, p. 107706, 2021.

[18] Z. You, L. Cui, Y. Shen, K. Yang, X. Lu, Y. Zheng, and X. Le, "A unified model for multi-class anomaly detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4571–4584, 2022.

[19] N.-C. Ristea, N. Madan, R. T. Ionescu, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, "Self-supervised predictive convolutional attentive block for anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 576–13 586.

[20] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD — a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9592–9600.

[21] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[23] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," *arXiv preprint arXiv:1807.02011*, 2018.

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[26] L. Zhao, Y. Chai, Q. Zhang, and H. R. Karimi, "Self-supervised anomaly detection based on foreground enhancement and autoencoder reconstruction," *Signal, Image and Video Processing*, vol. 18, no. 1, pp. 343–350, 2024.