Exploring the Relationship between Samples and Masks for Robust Defect Localization

Jiang Lin, Hui Xue, Fanxiu Sun, Yaping Yan*

School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
{220215663, hxue, 220246329, yan}@seu.edu.cn

Abstract

Defect detection aims to detect and localize regions out of the normal distribution. The previous approaches often explicitly incorporate the defect detection concept, such as by utilizing self-supervised ground truth or manually defined feature comparison. The aforementioned processes involve modeling the distribution of normal samples, and they rely on the modeled normality for accurate inference. This reliance may hinder their ability to generalize to unseen test scenarios or test sets that deviate from the training distribution. In this paper, we propose a one-stage framework that detects defective patterns directly without the modeling process. This ability is adopted through the joint efforts of three parties: a generative adversarial network (GAN), a newly proposed scaled pattern loss, and a dynamic correction mechanism that allows the network to self-correct. In training, explicit information that could indicate the position of defects is intentionally excluded to prevent the network from adopting any direct mapping. Experimental results show that the proposed method performs superior in comparison with the previous methods in various test scenarios, demonstrating its robustness.

Introduction

Defect detection is a crucial and formidable task in industrial production and automation. Defects exhibit uncertain appearances, often appearing as small anomalies that are partially intermixed with normal regions. The cost of manual annotation and the difficulty of collecting representative samples limit the effectiveness of conventional visual inspection methods, leading recent approaches to adopting unsupervised techniques for addressing this task.

In recent years, the rapid development of generative methods has led to the proposal of several reconstruction-based methods. Many methods employ Generative Adversarial Network (GAN) (Goodfellow et al. 2020) or Autoencoder (Bergmann et al. 2018) for image reconstruction, followed by anomaly level estimation through comparison of the input and the reconstructed images. The aforementioned methods, however, suffer from imperfect reconstructions and noisy outputs resulting from suboptimal distance metrics. Despite attempts (Akcay, Atapour-Abarghouei, and Breckon 2018; Akçay, Atapour-Abarghouei, and Breckon 2019; Deng and Li 2022; Zavrtanik, Kristan, and Skočaj 2021a) to address these issues, they share a similar paradigm: identifying defects by modeling the normal distribution.

However, the training source may not provide sufficient data to adequately model normality, leading to imperfect reconstructions. Variations in environmental factors could also disrupt the distribution of normality, constituting difficulty for generalizing. Beyond that, the purpose of using distance metrics in previous methods is to identify the regions that have undergone certain modifications while being restored to the modeled normality, which also no longer applies if normality changes. Despite these limitations, it appears theoretically infeasible to detect anomalies in object classes without referencing the normal distribution. The main reason is that their standards of normality heavily rely on human definition, which varies among different products. The modeling process upon the given normal samples can be seen as learning the anomaly standard of that particular class. This standard is not transferable across object classes since the same type of anomaly (e.g. rotation) may not be considered anomalous in another class. Texture classes, however, share the same anomaly standard, making the modeling process redundant as defects can be discerned by examining the samples alone.

Motivated by the unique characteristic of texture classes, we aim to propose a method that could directly localize the defective areas without any intermediate process that involves modeling the distribution of the normal samples, thereby avoiding the side effects that prevent the model from generalizing. We commence by examining simple self-supervised methods and promptly discern that they may adopt shortcuts due to the limited number of training samples. This prompted us to adopt an innovative approach that excludes all ground truth information during the learning process, preventing the network from exploiting shortcuts. The fundamental concept is to deliberately exclude explicit information that could signify the location of defects, enabling the network to discern the pattern of defects rather than overfitting certain appearances of defects or normal regions.

Specifically, we adopt a Generative Adversarial Network (GAN) that seeks to find the internal relationship between synthetic anomalies and arbitrary annotation masks, while

*Corresponding author

emphasizing that these two distributions are entirely independent. The generator is expected to find underlying correlations between them and produce annotations relevant to the input. The correlation can be simplified as follows: an annotation mask is characterized by negative and positive regions, possessing a binary nature. Similarly, the generator must recognize the binary nature of inputs, comprising defective and normal regions. Subsequently, it can acquire the ability to associate negative areas with negative annotations and vice versa.

Learning and making reasonable predictions in such a simplistic setup is highly challenging, often resulting in the generator producing mostly noise. Only occasionally does it show behavior aligned with our expectations. To facilitate the discovery of the correlation between these two distributions, we propose a concept termed *Invisible Pattern* that conceals pattern information exclusively within the semantic negative area, accompanied by a corresponding loss function called the scaled pattern loss. It establishes a bidirectional pixel-wise mapping, thereby preventing the generator from producing arbitrary outputs. Nevertheless, the adversarial objective often collides with the scaled pattern loss, with an emphasis on one diminishing the effect of the other. Inspired by previous methods (Zhu et al. 2017), we apply a cycle-consistent structure to ensure that the prediction encodes essential information, and successfully stabilizes the generation of the invisible pattern. Upon that, a dynamic correction mechanism is designed to incentivize the network to localize defects more accurately without the aid of ground truth. This unique mechanism allows the network to refine its predictions with its own predictions instead of depending on corrections from external sources. The collaborative efforts of these constraints and catalysts transform the generator from producing noises to achieving accurate precision with high reliability.

In contrast to previous approaches specifically designed for modeling normality distribution or discriminative learning mappings based on ground truth masks, our method is trained without explicit guidance to avoid leaking information that could indicate the location of defects and prevent overfitting. We provide unrelated masks solely just to ensure that the outputs resemble a mask while allowing the network to autonomously learn how to accurately localize defects. Our approach has successfully trained a model that operates seamlessly without the need for intermediate processes, showcasing robust generalization capabilities not only on test sets but also on additional test samples with diverse environmental variations. Experiments also suggest that the performance of our method is superior to the previous state-of-the-art methods while operating with no requirement for a threshold process before actual deployment. During the inference stage, our framework requires only one forward pass in the main generator, resulting in an increase in inference speed compared to previous methods.

Overall, our main contributions are listed as follows:

- A novel approach that leverages correlations between unaligned distributions to acquire the capability of defect localization while performing superior in experiments.
- A scaled pattern loss that enforces a bidirectional pixel-wise mapping between the input and the output.
- A dynamic correction mechanism that could optimize the prediction without any external guidance.
- We bring some new perspectives in assessing the practicability of defect localization methods.

Related Work

We briefly review previous literature on surface defect detection in this section. A primary difficulty in tackling this problem is the acquisition of defective samples. Consequently, recent literature predominantly emphasizes unsupervised or self-supervised methodologies.

A great number of anomaly detection models adopted a reconstructive approach. Generative methods such as GAN (Schlegl et al. 2017, 2019) or autoencoders (Bergmann et al. 2018; Gong et al. 2019) enable powerful reconstruction ability using normal data only. Under the assumption that these models would not successfully recover the defects, the localization map is produced by the reconstruction error between the image and its reconstruction (Sakurada and Yairi 2014; Zavrtanik, Kristan, and Skočaj 2021b). Distance metrics such as l_2 distance (Hadsell, Chopra, and LeCun 2006) or SSIM (Bergmann et al. 2018) are used for better measurement. However, the models might be able to reconstruct defects as well though they were not present in training, leading to low error in the comparison and causing inaccurate localization results. In DRAEM (Zavrtanik, Kristan, and Skočaj 2021a), a discriminative network is introduced to capture subtle differences between the inputs and the reconstructions. Other work (Guo et al. 2023) proposes a template-guided compensation approach to rectify the distorted features and restore them to their anomaly-free state. Besides the over-generalization of defects, another issue is the unsatisfied reconstructions regarding normal regions, leading to inaccurate localization. Some methods seek ways to generate images of higher quality. Bergmann et al. apply structural similarity to autoencoders and improve the reconstruction quality. Deng and Li propose a trainable one-class bottleneck embedding module to acquire more accurate reconstructions while excluding noises. However, changes in environmental conditions, such as light change or chromatism, still impacts the reconstruction results significantly.

In recent works, due to the lack of defect samples, which is essential in training, self-supervised methods are introduced to serve as a solution to the dire need for defect samples. Many methods (DeVries and Taylor 2017; Yun et al. 2019; Zhong et al. 2020) fabricate anomalies by replacing small rectangular regions of the original image with other values. (Li et al. 2021) propose the CutPaste augmentation to generate defect images by cutting a structural image patch from itself and randomly pasting them to other places. It highlights the need to synthesize samples close to the normal image distribution. Zavrtanik, Kristan, and Skočaj proposed an augmentation method by utilizing a Perlin noise mask to combine normal images with random image sources from (Cimpoi et al. 2014). The generated anomalies take various forms and contain both subtle and obvious defects.

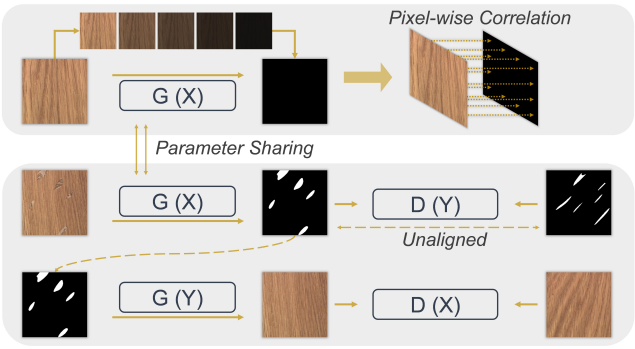

Figure 1: An overview of the proposed framework.

In reality, a gap exists between the distributions of fabricated data and real-world defects, leading to generalization issues. This significantly impacts the performance of self-supervised methods, prompting us to adopt a novel training scheme that operates without explicit guidance to improve generalization.

Method

The paper presents a new approach for defect localization that significantly diverges from previous methods. The presented method, as shown in Fig. 1, is a dual-path framework with two distinct inputs (normal and defective). Both paths share a common generator G_X, which is updated concurrently through feedback from both paths. The training process relies on both defective samples and normal samples. Given the absence of real-world defective samples, we extend the previous augmentation method (Zavrtanik, Kristan, and Skočaj 2021a) and simultaneously incorporate a new progressive opacity strategy.

Unpaired Transition

The first step, as shown in the middle of Fig. 1, is to establish a GAN network that takes the defective image as input and aims to generate an output resembling a randomly generated mask. This part consists of a generator G_X and a discriminator D_Y, the objective is formulated as:

$$\mathcal{L}_{GAN}(G_X, D_Y, X, Y) = \mathbb{E}[log(1 - D_Y(G_X(x))] + \mathbb{E}[logD_Y(y)], \quad (1)$$

where X and Y represent the domain of defective samples and random masks respectively, and x and y are the samples stochastically drawn from each domain. Both X and Y are simulated samples generated by data augmentation methods. They are intentionally set to be unaligned in training which means y is the wrong annotation for x.

The generator is naturally expected to output random masks. However, in experiments, the generator still produced predictions that were roughly related to the inputs. The generator showed a slight tendency to distinguish defects from normal areas, although the prediction is unstable and easily deteriorates into meaningless noise as the training progresses.

The aim is to utilize this trait to help the generator learn to make predictions without explicit guidance, thus improving generalization in testing. The current association that the generator has established between the inputs and random masks is weak, unstable, and imprecise. Despite having the corresponding self-supervised ground truth, we avoid using it to prevent information leaks and overfitting, which hinder the generator from generalizing effectively across unseen test conditions. Instead, we propose various constraints and catalysts that significantly improve its overall stability and precision, elevating it from producing mostly meaningless noise to making precise predictions.

Scaled Pattern Loss

In this section, we introduce a key concept called the invisible pattern and propose a new loss function termed the scaled pattern (SP) loss. The invisible pattern is a linearly scaled version of the normal samples. It closely resembles an empty prediction, while concealing necessary information. The scaled pattern loss is derived from this. As shown in Fig. 1, it requires another path of normal inputs, and the generator is expected to produce the corresponding invisible patterns. The goal of the SP loss is to create a pixel-wise bidirectional mapping between the input and the output of the generator, preventing the generator from producing arbitrary localizations. Since the invisible pattern is imperceptible to the human eye, its corresponding regions can be directly interpreted as negative predictions during inference. The corresponding formula is shown below:

$$I_s = I_n \times \alpha - \beta \quad (2)$$

$$\mathcal{L}_{sp}(I_p, I_s) = ||I_p - I_s||_1, \quad (3)$$

where I_s is the scaled version of inputs and I_p is the output for normal samples. The symbols α and β control the scale level of the image. Both of them are adjustable hyperparameters, and we recommend setting $\alpha = 0.005$ and $\beta = 0.995$.

Despite being trained only on normal input, the second path exerts a simultaneous influence on the outputs for anomalous images, thereby effectively preserving texture information within its region of negative predictions. The invisible pattern, which is distributed in a data range close to negative annotation, seamlessly integrates with the generator output, thereby enforcing it to produce pixel-wise correlated predictions for defective regions as well. However, this proximity also means that if the network produces a negative prediction instead of the invisible pattern, the SP loss will not increase significantly.

Dynamic Correction

Despite the theoretical feasibility of the above structure, the training process is found to be volatile in practice. As training progresses, the adversarial loss and the scaled pattern loss often collide and fail to coexist, with one potentially dominating and excluding the effect of the other. The dominance of the discriminator will erase the invisible pattern, replacing it with pure negative predictions. Conversely, if the SP loss is trusted with more weights, the generator will convert all input regions into the invisible pattern, thereby impeding the generation of the positive mask.

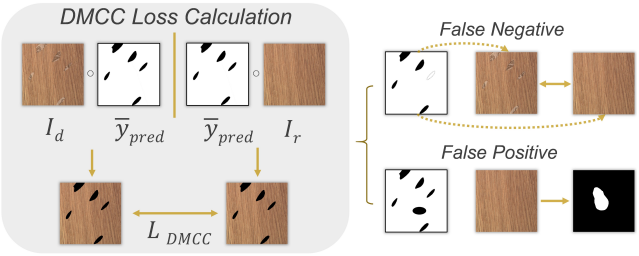

Figure 2: Illustration of the dynamic correction process.

Though the prediction is pixel-wise correlated with the input, whether the generator could precisely transform different regions into its corresponding annotations remains uncertain. Our design methodology here mainly focuses on two aspects. First, we ensure that the normal regions are correctly transformed into invisible patterns through information retrieval. Second, we punish the false negative prediction by enforcing a dynamic correction mechanism. Inspired by previous works(Zhu et al. 2017), we adopt the encoder-decoder paradigm and perceive the prediction of the generator as an encoded version of the input. A cycle-consistent structure is applied to recover the normal regions of the input from the invisible pattern, thereby preventing the invisible pattern lost. Specifically, the prediction y_{pred} is processed by a second generator G_Y to retrieve the original input from the invisible pattern, and a corresponding D_X discriminates the recovered image I_r with the original input I_d. This demand for pattern information drives the generation of the invisible pattern, improving overall stability. However, we observed a slight decline in performance and promptly recognized that this was a side effect of the defective areas that had been simultaneously recovered. Since the positive annotation itself contains no information, the defective areas could not be faithfully recovered, which hampers further convergence and induces network turbulence.

A rather straightforward solution is to utilize the mask obtained in the defect synthesis process to exclude the defective region during recovery loss calculation. Still, it leaks the defect location to the network. To avoid reliance on ground truth information, we propose a dynamic correction mechanism. It could stabilize the training and further improve the localization precision without necessitating prior knowledge of defect locations. Instead of the ground truth, we utilize the prediction from generator G to mask the defective regions in the inputs and the corresponding area in the recovered image x_r as in Fig. 2. As localization accuracy improves, the previous turbulence from the unsuccessfully recovered defective region will gradually diminish. Conversely, false negative predictions will create mismatch areas from unsuccessfully recovered areas, increasing the $DMCC$ loss. This establishes a positive feedback loop. On the other hand, false positives in predictions result in the occurrence of positive masks in the normal path, thereby increasing the SP loss. Together, these modules impose various constraints and catalysts, allowing the main generator to identify relationships between uncorrelated distributions for precise defect localization without any ground truth supervision. The mask process is denoted as M and the objective for the part can be formulated as:

$$\begin{aligned}\mathcal{L}_{DMCC}(G_X, G_Y) = \mathbb{E}[||M(G_Y(G_X(x)) - M(x)||_1] \\ + \mathbb{E}[||G_X(G_Y(y)) - y||_1].\end{aligned} \tag{4}$$

To maintain the balance between the losses, we assign proper weight to each loss and the full objective is:

$$\begin{aligned}\mathcal{L}(G_X, G_Y, D_X, D_Y) = \mathcal{L}_{GAN}(G_X, D_Y, X, Y) \\ + \mathcal{L}_{GAN}(G_Y, D_X, Y, X) \\ + \lambda_1 \mathcal{L}_{DMCC}(G_X, G_Y) \\ + \lambda_2 \mathcal{L}_{SP}(I_p, I_s),\end{aligned} \tag{5}$$

where λ_1 and λ_2 are set to 10 and 0.4 respectively, while reasonable adjustments could be made for the overall balance of the network.

Data Augmentation

In training, our method requires the presence of both defective and normal images. With only normal samples being accessible, augmentation methods could serve as an alternative to provide the defect samples needed. We adopt the synthesis technique proposed by (Zavrtanik, Kristan, and Skočaj 2021a) to generate defective samples while utilizing another Perlin (Perlin 1985) noise mask M_p generated during this process as the example for the discriminator D_Y. Note that, M_p are unaligned with M_s which is the mask that synthesis defects, which means no paired samples are utilized during training. Given the unique training method, we conjecture that the final performance of the model will be heavily influenced by the quality of the simulated samples. Transparency, in particular, plays a key role, and experiments show that a model trained with high-transparency samples is more sensitive to subtle defects. However, introducing more transparent defects results in unstable training due to potential failure in perceiving defective areas in the early training stage.

To address this, we propose a progressive opacity strategy (Pos) that gradually increases transparency throughout the training process, ultimately yielding nearly imperceptible defects. The training data is regenerated every 50 epochs based on a progressive lower opacity value provided. With Pos instead of random opacity, the model could reach a stable state faster by initially using apparent defects, and it is gradually optimized to detect more subtle defects.

Experiments

In this section, we benchmark the proposed method on a diverse set of texture defect detection datasets. The generalizability of the proposed method is also evaluated under simulated environment conditions.

Benchmarks

We compare our method qualitatively with recent works PatchCore (Roth et al. 2022), PyramidFlow(Lei et al. 2023), Revisiting Reverse Distillation (Tien et al. 2023) and PNI (Bae, Lee, and Kim 2023).

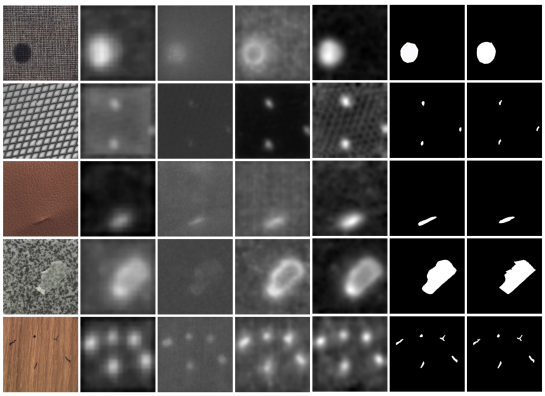

Figure 3: Comparison of anomaly maps (no threshold). The anomalous images and the ground truth are shown in the first and last columns. The middle five columns show results from comparison methods (PatchCore, PF, RRD, PNI) and our proposed method.

Datasets. The MVTec anomaly detection dataset (Bergmann et al. 2019) consists of 5 texture classes and 10 object classes. Our proposed method is specifically designed for detecting texture defects, thus the 5 texture classes are utilized in the evaluation. To further enrich the test source, we also benchmark on the DTD-Synthetic (Aota, Tong, and Okatani 2023) and the Woven Fabric Textures (Bergmann et al. 2018).

Metrics. AUROC is a commonly used metric that measures performance across all thresholds. In imbalanced problem settings, it tends to overlook false positives and excessively penalize false negatives. However, defect localization results should not be over ambiguous, as the presence of defects in specific regions is already determined by classification tasks. Methods with vague localization results often achieve a high AUROC, while offering obvious deficiencies in their results, as shown in Fig. 3. Practically, the defect localization results eventually need a threshold to determine the final detected defective region. Current methods often struggle to find an optimal threshold to acquire a performance that matches its high AUROC. Even when ground truth is available for threshold selection, which is clearly infeasible in real-world practice, the results still fall short, highlighting the inadequacy of AUROC in this context. The F1-Score, a threshold-based metric tailored for imbalanced problems, is a more suitable alternative.

Experimental settings. The images are resized to a predetermined resolution (256×256) and then normalized before being fed into the network. The generator (He et al. 2016) and discriminator (Isola et al. 2017) adopt structures that are similar to previous methods (Zhu et al. 2017; Kim et al. 2019). The model is trained for 1000 epochs using a batch size of 1 on an RTX 4090 GPU. An Adam optimizer (Kingma and Ba 2014) with $\beta = (0.5, 0.999)$ is applied with a learning rate of 0.0002. In training, the data augmentation method mentioned is utilized to generate a total of 300 simulated defective samples and masks in each regeneration.

Class	PC	PF	RRD	PNI	Ours
Wood	47.2	48.6	64.6	66.6	**70.7**
Carpet	59.8	54.1	51.1	54.3	**66.1**
Tile	63.6	70.6	66.1	71.9	**86.2**
Leather	45.0	46.4	50.6	41.4	**67.1**
Grid	18.5	32.9	53.0	**58.0**	55.8
AVG	46.8	50.5	57.1	58.5	**69.2**

Table 1: The F1-Score of the anomaly localization results on the MVTec (Bergmann et al. 2019) dataset.

Class	PC	PF	RRD	PNI	Ours
Texture_1	71.7	37.3	76.6	**76.8**	62.6
Texture_2	67.8	39.2	75.3	**76.3**	52.1
Blotchy_099	62.3	46.2	66.6	69.8	**85.9**
Fibrous_183	52.1	43.9	59.3	64.6	**84.3**
Marbled_078	59.3	46.1	60.3	66.0	**78.4**
Matted_069	55.2	52.8	67.1	65.7	**78.8**
Mesh_114	41.7	23.1	45.3	53.6	**62.1**
Perforated_037	45.5	22.4	41.1	58.8	**64.3**
Stratified_154	52.1	54.3	58.4	60.1	**73.4**
Woven_001	42.2	40.6	54.7	54.7	**78.5**
Woven_068	45.1	**66.9**	55.2	54.4	61.2
Woven_104	52.4	15.6	54.1	**60.0**	47.1
Woven_125	59.7	51.3	61.3	63.2	**76.5**
Woven_127	58.3	43.4	57.0	**63.6**	28.6
AVG	56.7	41.7	59.4	63.4	**66.7**

Table 2: The F1-Score of the anomaly localization results on the DTD-Synthetic (Aota, Tong, and Okatani 2023) and the Woven Fabric Textures (Bergmann et al. 2018).

Performance. The performance of our method is compared to the recent approaches in Table 1 and Table 2, with the best results highlighted in boldface. Our method outperforms the previous SOTA by 10.7% on the MVTec dataset and by 3.3% on the combination of the DTD-Synthetic dataset and the Woven Fabric Textures dataset. A visual comparison with previous methods is presented in Fig. 3, and our method provides precise localizations with less noise. The average prediction times of RRD (Tien et al. 2023) and PNI (Bae, Lee, and Kim 2023) are 0.239 and 55.930 seconds respectively, whereas our method achieves a prediction time of 0.006 seconds under the same test environment, greatly increasing the inference speed.

Generalizability

Previous approaches have demonstrated consistent advances in performance. The generalizability, regarding the accuracy of test samples under different environmental conditions, has not yet been given enough attention. The appearance of texture in the real world can vary significantly due to factors such as lighting and chromatism, which highlights the importance of assessing generalizability in unseen scenarios to determine the applicability of models.

In this subsection, we conduct additional experiments in simulated environments to evaluate the generalizability of our approach. Image attributes, such as Contrast, Brightness,

	General Setting					Hard Setting				
Class	PatchCore	PyramidFlow	RRD	PNI	Ours	PatchCore	PyramidFlow	RRD	PNI	Ours
Wood	44.1	11.6	62.4	**64.5**	62.7	14.2	9.4	**60.4**	61.6	50.0
Carpet	51.2	5.2	52.9	54.7	**63.2**	52.3	11.0	49.5	17.8	**57.2**
Tile	61.3	21.1	63.1	67.6	**75.4**	55.6	36.2	62.7	**64.2**	63.5
Leather	44.9	12.4	**49.9**	18.8	49.8	19.3	1.9	**50.1**	34.7	45.3
Grid	7.3	2.6	26.6	16.8	**47.7**	8.9	3.3	23.4	15.5	**48.2**
AVG	41.8	10.6	51.0	44.5	**59.8**	30.1	12.4	49.2	38.8	**52.8**

Table 3: Results for generalization experiments under general setting (left) and hard setting (right).

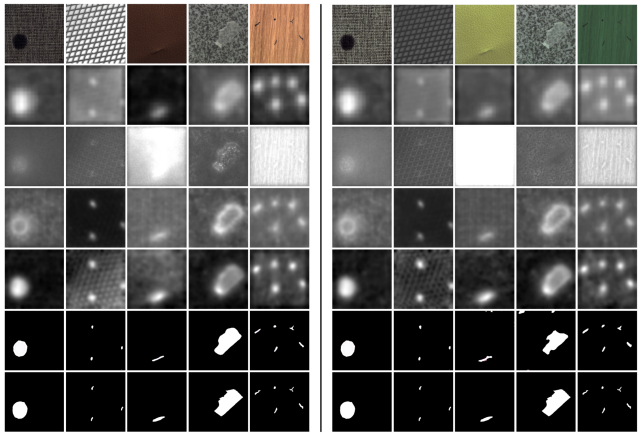

Figure 4: The first row shows the source images for the general (left) and hard (right) settings. The following four rows display localization results for PatchCore, PyramidFlow, RRD, PNI, and our proposed method, respectively. The ground truth is shown at the bottom.

and Hue are utilized to create a simulated illumination environment of two levels of settings. The general setting adjusts both brightness and contrast, whereas the hard setting additionally modifies hue. As could be compared between Fig. 4 and Fig. 3, previous methods already perform unstable in the general setting, and experience more significant impacts in the hard setting. The statistics presented in Table 3 demonstrate the overall performance in both settings, wherein certain methods exhibit significant degradation across multiple classes when subjected to unseen test scenarios. Our method is capable of maintaining certain performance in the visuals and delivering superior performance in both the general setting and the hard setting.

A key difference in our method is that it localizes defective regions directly without reference to the normality. In training, there is also no explicit information to signify the location of defects. Our framework is trained without the presence of ground truth, and no manually designed procedures are used for determining the existence of a defect, such as comparing feature similarity with normal samples or measuring reconstruction differences. Therefore, the network will not associate the prediction with certain appearances and could adapt well to unseen scenarios as shown in Fig. 4. Our network learns to localize defects by continuously attempting predictions and being corrected by the dynamic correction mechanism, and this process does not contain explicit goals for defect localization. After the training, we believe the network learns to identify the defect based on a global observation of the inputs. The prediction no longer relies on fixed normality, thereby avoiding assumptions about the input distribution and demonstrating enhanced adaptability to environmental variations.

Ablation Study

In this section, we investigate the individual contributions of each module and the impacts of data choice.

Architecture

The components are reassembled to form different structures. These components consist: (*1*) GAN represents only the basic GAN structure(*2*) Scaled denotes the scaled pattern loss with another path of normal inputs. (*3*) Cyc is the auxiliary network with an additional generator and discriminator, enabling cycle-consistent loss. (*4*) Mask refers to the dynamic correction mechanism. (*5*) Pos is the progressive opacity strategy that we applied in data generation. A combination of initials, such as $G.S.C$, represents the enabled components and the content within the parentheses indicates the opacity setting.

Table 4 shows that $G.S.C.M(Pos)$ demonstrates optimal performance compared to others, while the simplest form G only produces a result of 2.8%. Upon closer inspection, the table reveals a general increasing trend as each component is added. Contrary to the overall trend, the performance of $G.S.C$ has deteriorated with the addition of the Cyc component, which is due to the unsuccessful recovery for defective regions. G alone only delivers a result of 2.8%, which is mostly noise. With additional components, whether in $G.S$ or $G.C.M$, the prediction reaches a functional point. Unlike other methods, these components are not part of a linear process, and their joint effort falls onto the generator alone. In $G.S.C.M$ and $G.S.C.M(Pos)$, there are two more major rises in the performance. The performance improvement would not be as significant if other components were removed, such as a comparison between $G.C.M$ and $G.C.M(Pos)$. These components serve different purposes, and simply stacking components will not result in a significant performance increase each time.

Method	Architecture				Opacity					AVG
	GAN	Scaled	Cyc	Mask	0(0.1)	0(0.5)	0(0.9)	O(0.1-0.9)	Pos	
G	✓							✓		2.8
G.S	✓	✓						✓		37.2
G.C.M	✓		✓					✓		27.4
G.S.C	✓	✓	✓					✓		35.4
G.S.C.M	✓	✓	✓	✓				✓		51.1
G.S.C.M(0.1)	✓	✓	✓	✓	✓					15.9
G.S.C.M(0.5)	✓	✓	✓	✓		✓				40.4
G.S.C.M(0.9)	✓	✓	✓	✓			✓			30.9
G.S.C.M(Pos)	✓	✓	✓	✓					✓	**69.2**

Table 4: Anomaly localization results of ablation study on architecture (left) and opacity (right).

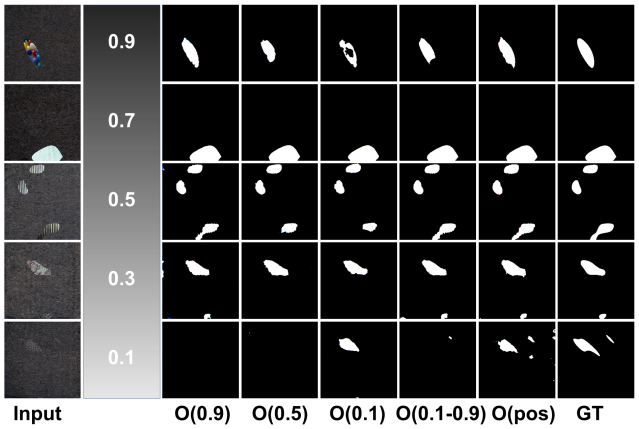

Figure 5: Comparisons of the opacity strategies on dataset(Božič, Tabernik, and Skočaj 2021). The first two columns display simulated defects and a bar indicating opacity. The remaining columns show localization results for models trained with different opacity settings, along with the ground truth.

Data Source

In this framework, the model is not guided by any explicit information and needs to navigate the correct way of localizing defects. The opacity value significantly affects defect visibility and plays a crucial role in determining recognition difficulty, making it a pivotal training factor.

In Table 4, both $G.S.C.M(0.1)$ and $G.S.C.M(0.9)$ deliver unsatisfying results. In experiments, training with obvious defects allows the network to learn defect localization faster and more stably. On the other hand, the presence of more transparent defects encourages the model to discern subtle defects. However, the training can be unstable because the generator struggles to identify transparent simulated defects in the initial stage. The proposed progressive opacity strategy facilitates early training by simulating obvious defects and then gradually increasing the transparency of synthetic defects to promote the recognition of subtle defects. Experimental results have demonstrated its effectiveness, with performance significantly outperforming fixed opacity values and random opacity settings.

The impact of opacity is further examined through experiments on simulated test sources with opacities ranging from 0.1 to 0.9, assessing how the model handles defects with varying transparency levels. As shown in Fig. 5, models trained with high opacity fail to localize nearly transparent defects. Conversely, models trained with low opacity perform adequately on the presented samples but show inconsistent results elsewhere. For instance, $G.S.C.M(0.1)$ achieves only half the performance of $G.S.C.M(0.9)$ in Table 4. Models trained with Pos demonstrate consistent performance across simulated test sources with different opacities and achieve optimal results in Table 4. These findings suggest that the performance of this method could be further enhanced by training with real samples exhibiting similar characteristics. Additionally, the model trains with unpaired samples, enabling it to directly utilize unlabeled real-world defects—something other methods often find impractical. The accessibility to real-world defects could potentially lead to improved quality of the model.

Conclusion

In this paper, we pointed out that over-reliance on the modeled distribution may limit the applicability of defect localization methods. Our hypothesis posits that by removing explicit guidance during the training process, the model could generalize better, which enables better performance in various test scenarios. Motivated by this, we proposed a novel GAN-based framework, leveraging its adversarial nature while incorporating various constraints and catalysts to guide defect localization effectively. The proposed invisible pattern, along with the dynamic correction mechanism, enables the generator to self-correct without relying on external guidance. The proposed method attains optimal performance on the benchmark datasets, with additional experiments demonstrating its robustness. The potential of our framework can be further enhanced by leveraging raw defect samples from real-world scenarios to improve its performance, a practice uncommon in other methods that typically require defective samples accompanied by ground truth.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant 62201142.

References

Akcay, S.; Atapour-Abarghouei, A.; and Breckon, T. P. 2018. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, 622–637. Springer.

Akçay, S.; Atapour-Abarghouei, A.; and Breckon, T. P. 2019. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Aota, T.; Tong, L. T. T.; and Okatani, T. 2023. Zero-shot versus many-shot: Unsupervised texture anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 5564–5572.

Bae, J.; Lee, J.-H.; and Kim, S. 2023. PNI: Industrial anomaly detection using position and neighborhood information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6373–6383.

Bergmann, P.; Fauser, M.; Sattlegger, D.; and Steger, C. 2019. MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9592–9600.

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

Božič, J.; Tabernik, D.; and Skočaj, D. 2021. Mixed supervision for surface-defect detection: from weakly to fully supervised learning. *Computers in Industry*.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3606–3613.

Deng, H.; and Li, X. 2022. Anomaly Detection via Reverse Distillation from One-Class Embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9737–9746.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Gong, D.; Liu, L.; Le, V.; Saha, B.; Mansour, M. R.; Venkatesh, S.; and Hengel, A. v. d. 2019. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1705–1714.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Guo, H.; Ren, L.; Fu, J.; Wang, Y.; Zhang, Z.; Lan, C.; Wang, H.; and Hou, X. 2023. Template-guided hierarchical feature restoration for anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6447–6458.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, 1735–1742. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Kim, J.; Kim, M.; Kang, H.; and Lee, K. 2019. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lei, J.; Hu, X.; Wang, Y.; and Liu, D. 2023. PyramidFlow: High-Resolution Defect Contrastive Localization using Pyramid Normalizing Flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14143–14152.

Li, C.-L.; Sohn, K.; Yoon, J.; and Pfister, T. 2021. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9664–9674.

Perlin, K. 1985. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3): 287–296.

Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; and Gehler, P. 2022. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14318–14328.

Sakurada, M.; and Yairi, T. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, 4–11.

Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Langs, G.; and Schmidt-Erfurth, U. 2019. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54: 30–44.

Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Schmidt-Erfurth, U.; and Langs, G. 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, 146–157. Springer.

Tien, T. D.; Nguyen, A. T.; Tran, N. H.; Huy, T. D.; Duong, S.; Nguyen, C. D. T.; and Truong, S. Q. 2023. Revisiting reverse distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 24511–24520.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the*

IEEE/CVF international conference on computer vision, 6023–6032.

Zavrtanik, V.; Kristan, M.; and Skočaj, D. 2021a. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8330–8339.

Zavrtanik, V.; Kristan, M.; and Skočaj, D. 2021b. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112: 107706.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2020. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 13001–13008.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.