ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240488

Segmentation-Driven Image Enhancement Based on Deep Reinforcement Learning

Yihong Liu[a,1], **Zishang Chen**[a,1], **Yukang Cui**[a] **and Piji Li**[a,*]

[a]Nanjing University of Aeronautics and Astronautics

Abstract. The rise of large models, often referred to as foundational models, has led to considerable progress in the field of artificial intelligence research. Our empirical findings indicate that the large models might struggle or deliver poor performance when it comes to specific surface segmentation challenges, including the identification and segmentation of defects on strip steel surfaces (S^3D) and the detection of imperfections on magnetic tile surfaces. To apply the large model to defects segmentation, rather than fine-tuning the large model, we propose **S**egmentation-**D**riven **I**mage **E**nhancement (SDIE), using several classic filters to enhance the input images. In this case, the weights of the filters in multiple layers are controlled by reinforcement learning. Then, we test our method on two S^3D datasets with different few-shot settings. Our method accomplishes the task brilliantly compared with other methods for S^3D such as CPANet. We believe that our work not only opens up opportunities for downstream tasks such as segmenting industrial defects using large models, but may also have potential applications in various fields in the future, including medical image processing, remote sensing image analysis, agriculture and more.

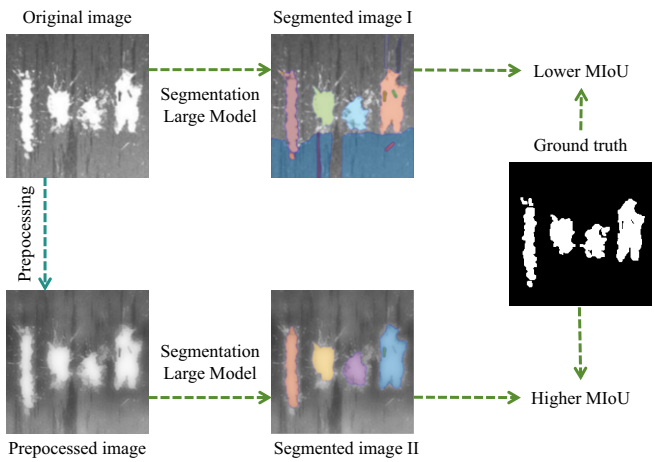

Figure 1. Segmented image I is directly segmented by large model from original image while Segmented image II is segmented from preprocessed image. The noise in the background may mislead large model without preprocessing.

1 Introduction

Surface defect segmentation plays an important role in industrial production. The detection accuracy of surface defects in industrial products can effectively prevent further losses. Manual inspection is one of the main methods in many practical industrial quality inspection processes. However, due to factors such as visual fatigue, manual detection is sometimes unreliable and requires specific expert knowledge. Therefore, early machine learning methods [13], which rely on manually constructing defect features, are proposed to achieve automatic detection of industrial surface defects. However, industrial product surface features with irregular shapes and significant size variations are difficult to accurately characterize. When new defects appear, experts not only need to design multiple additional defect template groups but also need to perform complex feature post-processing. With the rapid development of deep learning, more and more defect detection models based on convolution neural network (CNN) framework have exhibited remarkable performance in surface defect detection [10, 26, 34]. These methods save the cost of manually designing defect features and greatly improve detection accuracy.

Currently, industrial surface defect detection methods mainly include image classification [25], target detection [42] and semantic segmentation [28]. In contrast to the previous two paradigms, methods based on semantic segmentation classify each pixel in a defective image [24] densely, demonstrating a strong capability to predict defective areas on a pixel-by-pixel basis. In recent years, the field of semantic segmentation has witnessed considerable advancements, predominantly driven by the substantial progress in deep learning techniques. However, although existing defect semantic segmentation models have achieved good predictive ability, with the advancement of production technology, the defect rate on the surface of industrial devices has been strictly controlled, which leads to a decrease in the number of accessible defect samples. Therefore, obtaining sufficient defect data is a challenge for researchers. Unfortunately, traditional cnn-based methods require a sufficient amount of annotated data to optimize their large trainable model parameters. In addition, these supervised approaches work effectively only on the defect classes that participate in the training practice phase. In other words, traditional segmentation methods generalize poorly to new defect classes with fewer labeled samples.

To overcome the scarcity of defect samples, Feng [8] has adopted the few-shot semantic segmentation (FSS) approach in their study. The objective of the FSS [36] technique is to efficiently develop segmentation algorithms utilizing a minimal set of annotated instances, enabling swift adaptation to novel defect types with the requirement

* Corresponding author. Email: pjli@nuaa.edu.cn
[1] Equal contribution

of limited additional labeling [31].

Despite the excellent performance of existing FSS methods in non-industrial surface defect segmentation [1, 47], industrial surface defects typically exhibit irregular shapes, significant size variations, inter-class similarity, low contrast and ambiguity between normal and defects, which requires a high level of generalization ability. So we turn our attention to the current popular large models for image segmentation because when using a large number of image libraries from the network for scaling and training, these basic models perform surprisingly well with few shots. However, Chen *et al* [5] has found through testing that SAM [17] doesn't meet expectations in some challenging low-level structure segmentation tasks. We find that SAM makes it difficult to distinguish small defects in the surface of industrial products through experiments, especially when the overall image is blurry.

Hence, a pivotal research inquiry focuses on: how can we utilize the powerful features of models trained on vast datasets and channel their potential to improve the precision and efficiency of defect segmentation in industrial settings? We find that the preprocessing of the images helps a lot for large models. As shown in the lower part of Figure 1, after being processed by our filters, the background noise in the image is reduced. At the same time, the defect border is more prominent and the pixel contrast between the defect and the background is greater. This not only allows the SAM to better identify defects in the image, but also reduces the possibility of its judgment errors. As shown in the upper part of Figure 1, we find that the noise in the background of the defective image can severely affect the segmentation effect of the SAM, thereby misjudging the background as a defect.

Due to the fact that the defect images of defects obtained in the industry are often taken in low light environments, we come up with the idea of using image enhancement to highlight the defect information instead of fine-tuning the basic model. We have prepared several commonly used filters for image enhancement, and the weights and parameters of these filters can be adjusted using deep reinforcement learning. Specifically, in each training round, we apply different weights and parameters to the filters, which are weighted and summed the processed images of each filter. Then we input the results of multiple rounds of filtering into SAM, using some important metrics in image segmentation field such as mean Intersection over Union (MIoU) as the reference metric for our reinforcement learning. Through experiments, it has been proven that our structure is effective in segmenting defects in many types of industry defects.

The contributions of this work can be summarized as follows:

- We pioneer image enhancement specifically for image segmentation in industry and propose **S**egmentation-**D**riven **I**mage **E**nhancement (SDIE). Under this framework, large model can be easily applied to a specific dataset simply by training our lightweight enhancement module on this dataset. What's more, our method can combine with various segmentation methods because it is a lightweight module that enhances the input image to improve the segmentation ability of the corresponding method.

- We use various classic image enhancement filters and sum the filter results according to the weights controlled by a trained agent. We model the image enhancement problem as a Markov decision problem and prove the Markov property of the process. After that, we adopt deep reinforcement learning to solve the problem.

- We evaluate our method on two S^3D datasets FSSD-12 and Defect-4^i with 1-shot and 5-shot settings. Attributed to the fact that classical filters do not need to be trained and the strong generalization performance of large models, our method performs well under the few-shot situations and a small number of rounds of RL training.

2 Related Work

2.1 Surface Defect Segmentation

In the field of industrial inspection, surface defect segmentation has high accuracy and has received widespread attention in recent years. Huang *et al* [16] use a push network to define and predict the specific location of surface defects through bounding boxes. Nand and Neogi [28] propose a new entropy based defect detection algorithm, which utilizes the local entropy of images to detect defect areas. Tabernik *et al* [33] propose a deep learning architecture specifically designed for surface anomaly detection and segmentation. The first stage implements a segmentation network that performs pixel by pixel localization of surface defects, and the second stage uses an additional network built on top of the segmentation network to perform binary image classification.

Wang *et al* [38] release the first publicly available dataset of few shot defects, NEU-DET, to alleviate the drawback of insufficient defect samples. Xiao *et al* [42] design graph embedding and distribution transformation modules, as well as optimal transmission modules, fully utilizing the interrelationships between features to achieve few shot classification through direct inference. Bao *et al* [1] transform the segmentation problem of metal surface defects with few shots into a semantic segmentation problem of few shots in defect and background regions through triplets and proposes a multi-image inference method to explore the similarity relationship between different images to improve segmentation performance in industrial scenes. They establish a new dataset called "Surface Defect-4^i", which includes Nonindustrial defects such as leather and tile, to further evaluate the segmentation performance of the model. In addition, Wu *et al* [40] introduce a ResMask Generative Adversarial Network (GAN) framework, which is a residual GAN used to expand insufficient defect datasets. Feng *et al* [9] propose a simple but effective few-shot segmentation method named cross position aggregation network (CPANet), which intends to learn a network that can segment untrained S^3D categories with only a few labeled defective samples. Reviewing the current literature, it becomes clear that many existing methodologies are specifically designed for a single material or defect type. In essence, these studies are not equipped to handle a diverse array of defects across different materials. Consequently, the pursuit of a more universal approach to surface defect segmentation, as highlighted in reference [41], is highly significant.

2.2 Multi-modal large model

In recent years, significant progress has been made in multi-modal large-scale models (MLMs) [19, 20, 45, 50]. By increasing the size of the data and model, these MLMs have improved their amazing emergence ability and demonstrated surprising zero/few-shot inference performance in downstream tasks. Liu *et al* [23] propose a pipeline for automatically generating language image instructions to follow data, and based on this, train a multi-modal model LLaVA to complete visual tasks such as classification, detection and segmentation following human intentions. Meta AI Research design a model called Segment Anything (SAM) [17] which consists of a powerful

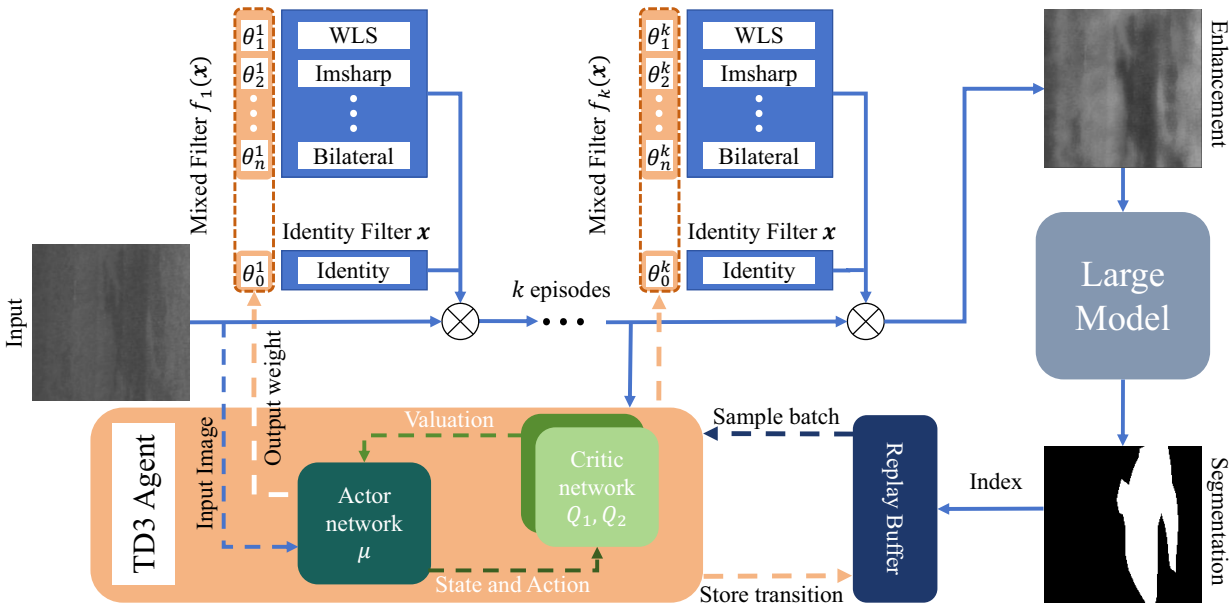

Figure 2. The framework of Segmentation-Driven Image Enhancement. After inputting the original image, it is filtered by the mixed filter in multiple rounds. In each round, the image is also inputted to TD3 agent to decide the weight of the mixed filter. After the enhancement, the enhanced image is sent to large model for segmentation returning an index such as MIoU replay buffer, which is prepared for TD3 agent training. The detailed training process is shown in Section 4.

image encoder, a prompt encoder and a lightweight decoder. SAM shows impressive performance that it can be used out-of-the-box with prompt engineering to solve a variety of tasks involving object and image distributions beyond SAM's training data. On this basis Chen *et al* [5] propose SAM-Adapter, which incorporates domain-specific information or visual prompts into the segmentation network by using simple yet effective adapters to solve the problem of poor performance of SAM in a series of tasks such as hidden object detection. Cheng *et al* [6] presents a framework called Segment And Track Anything (SAM-Track) that allows users to select multiple objects in videos for tracking, corresponding to their specific requirements.

2.3 Image Enhancement

Enhancing images can improve image quality, enhance analysis and recognition accuracy. Performing data pre-processing plays an important role in the fields of computer vision and image processing [3]. Sharma *et al* [32] presents a unified CNN architecture that uses a range of enhancement filters that can enhance image-specific details via end-to-end dynamic filter learning. The approach is capable of improving the performance of all generic CNN architectures. Zheng *et al* [51] propose a paradigm for low-light image enhancement that explores the potential of customized learnable priors to improve the transparency of the deep unfolding paradigm. Wang and Jin [37] propose a "brighten and colorize" enhancement network BCNet for low light images, which includes a multitask encoder and two task specific decoders to decompose low light images into brightness and chromaticity, achieving decoupling enhancement. Li *et al* [21] propose a new paradigm, aesthetics-guided low-light image enhancement (ALL-E), which introduces aesthetic preferences to low-light image enhancement and motivates training in a reinforcement learning framework with an aesthetic reward to make integrating human preferences into image enhancement. Kozlowski *et al* [18] propose a semi-supervised method called Dimma, which replicates scenes captured by specific cameras under extreme lighting conditions using a small set of image pairs, thus maintaining consistency with any camera. This approach enables accurate grading of the dimming factor, which provides a wide range of control and flexibility in adjusting the brightness levels during the low-light image enhancement process. Yu *et al* [46] propose a novel synergistic structure that can balance brightness, color and illumination more effectively in Low-light enhancement tasks.

3 Problem Formulation

3.1 Image Segmentation

S^{3}D task needs the division of an image into a number of disjoint regions based on features such as grayscale, color, spatial texture, geometric shapes, etc. Thus it requires a classifier to predict each pixel's classes in the input image and each class can represent a kind of defect, texture, etc. We modeled this process as follows:

We denote the S^{3}D dataset by $D = \{(M_i, M_i^*)|\forall i \in [1, n]\}$ including c classes, where $M_i, M_i^* \in \mathbb{R}^{w\times h\times r}$ represent the image to be segmented and its ground truth. D is divided into D_{train} and D_{test}. Considering a mapping $S(\cdot) : \mathbb{R}^{w\times h\times r} \to \mathbb{R}^{w\times h\times r}$, image segmentation requires finding a mapping S by the rules of D_{train} to transfer M_i to the prediction $M_i^{'} \in \mathbb{R}^{w\times h\times r}$ aiming to maximize the $Index(\cdot)$ on D_{test} as (1)

$$\max_{S} \sum_{D_{test}} \text{Index}\left(S(M_i|D_{train}), M_i^*\right). \tag{1}$$

$Index(\cdot)$ can be an important metric in the field of image segmentation such as MIoU [30], FB-IoU, etc.

3.2 Segmentation-Driven Image Enhancement

To solve the problem, we propose SDIE which performs multiple rounds of enhancement of an image using combined multiple enhancement filters in a weighted way. This method provides a

lightweight module for large models. After training on a type of dataset (such as defect detection), this module can make the contour of the defect area more prominent, and the noise in the background part smoother, thereby making the large model perform better on this type of dataset.

Our specific design framework for the enhancement module is shown in Figure 2:

Step 1: Use various traditional image enhancement filters and filter these filters to the image respectively.

Step 2: Take a certain weight to calculate the weighted sum of the filtering results as a one-step enhancement image. The decision of weight will be described in detail in section 4.

Step 3: Use the output image of step 2 as the input image of step 1. Repeat steps 1-2 until the preset round is reached.

We also introduce the architecture design of ResNet [15] in step 1: The whole filtering process can be seen as multiple convolutional layers, we add identity filters in step 1, which can significantly increase the ability of the convolutional layers to express the identity map. We use $\mathbf{w}$ to represent the weight of all the filters and define our one-step weighted filtering process on image M as $f_i(M; \mathbf{w}_i)$. We perform T-round filtering on the image, and the whole filtering process can be represented as

$$F(\cdot; W) = f_1(\cdot; \mathbf{w}_1) \circ f_2(\cdot; \mathbf{w}_2) \circ \cdots \circ f_T(\cdot; \mathbf{w}_T). \tag{2}$$

This framework has the following characteristics: various traditional image filters can be seen as our "toolkit". For the image to be enhanced, by controlling the weights, we can select different tools in each round and enhance the image with different weight combinations.

After image enhancement, we use the large model for segmentation. Pretrained large models such as SAM can perform segmentation tasks on many different datasets. Under this framework, our problem is transformed into inputting an image M and solving the sequence $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T$ to maximize the $Index(\cdot)$ of the enhanced image segmented by large model. We formulated our problem as follows.

$$\begin{aligned} &\max_{W} \sum_i Index\left(S(F(M_i; W)), M_i^*\right). \\ &\text{s.t. } W = (\mathbf{w}_i | \forall i \in [1, T]). \end{aligned} \tag{3}$$

Here, S can represent a pretrained large model for segmentation.

4 DRL Based Filters for Enhancement

It is nontrivial to solve Problem (3) for W has many parameters. What's more, the solution of $\mathbf{w}_i$ is related to $\mathbf{w}_j, \forall j \in [0, i)$ which makes the problem more complex. Therefore, we formulate (3) as a deep reinforcement learning (DRL) problem and employ the twin delayed deep deterministic policy gradient (TD3) [12] to solve the problem.

4.1 Markov Decision Process Design

Although the problem has many parameters and is complex to solve, it has the following ideal property. The solution to $\mathbf{w}_i$ is only related to the image at the current step M_i. Therefore, the problem can be transformed into a Markov decision process (MDP) [11] and solved using reinforcement learning (RL) methods. The image enhancement process is defined as follows.

Definition 1. *Considering an enhancement process for image M, in each round $t \in [1, T]$ the agent observes the state s_t embedded from current image M_t, and takes an action a_t as the parameter $\mathbf{w}_t$ of the filter f_t. Then, we filter f_t to image M_t to get enhanced image at $t+1$ round M_{t+1}. The reward r_t is positively corrected with the result of segmentation of image M_{t+1} evaluated by $Index$. The main components of the process are detailed as follows.*

- ***State**: Encoded by $G(\cdot)$ as (4), state s_t is the embedding of the image M_t in round t.*

$$s_t = G(M_t). \tag{4}$$

- ***Action**: Given the state s_t, the agent decides the action a_t as filter's weight. Since we hope the brightness of the enhanced image to be consistent with the original image, we guarantee the sum of elements in w_t equals 1 by $\mathbf{w}_t = N(a_t)$ where*

$$\mathbf{w}_t^i = \frac{a_t^i}{\sum_{i=1}^n a_t^i}. \tag{5}$$

  $\mathbf{w}_t^i$ denote the weight of i-th filter.
- ***State transition**: Under the state s_t, take action a_t, we filter to obtain new image*

$$M_{t+1} = f_t(M_t; \mathbf{w}_t). \tag{6}$$

  The new state s_{t+1} is embedding of the new image $s_{t+1} = G(M_{t+1})$.
- ***Reward**: After filtering, the enhanced image is segmented by a large model. The segmented image is calculated with the ground truth M_t^* to obtain the value index as*

$$r_t = Index(S(f_t(M_t; \mathbf{w}_t)), M_t^*) \tag{7}$$

Lemma 1. *Suppose $\{s_t : t \in T\}$ is a stochastic process, where s_t is a state at moment t and T is a time set.*

Then, for any moment t and any states i, the Markov property can be stated as follows:

$$P(s_{t+1} = i_{t+1} | s_n = i_n) = P(s_{t+1} = i_{t+1} | s_n = i_n, \cdots, s_1 = i_1)$$

A stochastic process is a Markov process if and only if the process satisfies the Markov property.

Theorem 2. *The image enhancement process in definition 1 is a Markov Decision Process.*

Proof. According to (4) and (6), we have the relation between two neighboring states s_{t+1} and s_t.

$$s_{t+1} = G(f_t(G^{-1}(s_t); \mathbf{w}_t). \tag{8}$$

Given the state s_t, the agent decides the action by a policy function formulated as follows.

$$a_t = \mu(s_t). \tag{9}$$

Next, we have

$$\begin{aligned} s_{t+1} &= G(f_t(G^{-1}(s_t); \mathbf{w}_t) \\ &= G(f_t(G^{-1}(s_t); N(a_t)) \\ &= G(f_t(G^{-1}(s_t); N(\mu(s_t))) \\ &= g(s_t). \end{aligned} \tag{10}$$

where $g(\cdot)$ represents the composite function $G(f_t(G^{-1}(\cdot); N(\cdot)))$. It's clear that s_{t+1} is only related with s_t. Therefore, for $\forall i_n$, under any conditions that do not conflict with $s_n = i_n$,

$$P(s_{t+1} = g(i_n)|s_n = i_n) = P(s_{t+1} = i_{t+1}|s_n = i_n, \cdots, s_1 = i_1) = 1. \quad (11)$$

Finally, the sequence $\{s_1, s_2, \cdots, s_n\}$ satisfies Markov property and the image enhancement process is a Markov Decision Process. □

4.2 The agent trained on TD3

The TD3 agent is designed to learn two critic networks, denoted as $Q_{\theta_1}(\mathbf{s}_t, a)$ and $Q_{\theta_2}(\mathbf{s}_t, a)$, which are utilized to assess the value of taking action a in the state $\mathbf{s}_t$. To mitigate the issue of overestimation, the agent selects the lower of the two values provided by these networks as the action valuation. In order to streamline the process of calculating $\max_a Q(\mathbf{s}_t, a)$, an actor network $\mu_\phi(\mathbf{s}_t)$ is concurrently trained, with the goal of maximizing $Q_{\theta_i}(\mathbf{s}_t, a)$. After the critic networks are trained to estimate the Q-target defined in (13), during the evaluation phase, the TD3 agent determines the action as follows:

$$a_t = \mu_\phi\left(\mathbf{s}^t\right) = \arg\max_{a_t} Q(\mathbf{s}_t, a_t). \quad (12)$$

This action valuation signifies the highest expected return the agent can achieve by selecting the filter parameter f_t in the state $\mathbf{s}_t$. Consequently, the optimum value of $Q_{\theta_i}\left(\mathbf{s}^t, a\right)$ is attained when the parameter $\mathbf{w}_t$ is established for f_t at the state s_t.

The TD3 algorithm incorporates three key techniques aimed at enhancing the performance of the FL server.

1. Clipped double Q-learning: To address the issue of overestimation in Q-learning, TD3 simultaneously learns two Q-functions, Q_{θ_1} and Q_{θ_2}, by minimizing the mean squared error. Both Q-functions share the same target, and the Q-target is determined by taking the minimum value from the two Q-functions, expressed as follows:

$$y = r + \gamma \min_{i=1,2} Q_{\theta_i'}(s_{t+1}, a_{TD3}(s_{t+1})) \quad (13)$$

2. Delayed policy updates: Empirical evidence suggests that concurrently training the actor and critic networks, without the use of a target network, can result in instability during training. In contrast, fixing the actor network while updating the critic network allows for better convergence. Therefore, the TD3 method updates the actor network less frequently compared to the critic network, specifically updating the policy after every u updates of the critic network.

3. Target policy smoothing: TD3 incorporates a smoothing technique that introduces noise into the target action, which helps prevent the policy from exploiting errors in the Q function by reducing the fluctuations in Q with respect to the action. The target policy smoothing is defined as follows:

$$a_{TD3}(s_{t+1}) = C(\mu_\phi(s_{t+1}) + C(\epsilon, -c, c), a_{low}, a_{high}) \quad (14)$$

Here, ϵ represents noise sampled from a normal distribution, $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$. This technique serves as a regularization method, where c is a constant ensuring that ϵ remains within the bounds of $[-c, c]$, while a_{low} and a_{high} denote the lower and upper limits of the action space, respectively.

In Algorithm 1, the training process of the TD3 agent begins by initializing the replay buffer $\mathcal{B}$ to store transition tuples $(\mathbf{s}, a, r, \mathbf{s}')$, where the agent executes action a in state $\mathbf{s}$, receives reward r, and transitions to the subsequent state $\mathbf{s}'$. A batch of N transition samples $(\mathbf{s}, a, r, \mathbf{s}')$ is then randomly drawn for training. The target Q-function Q_{θ_i}, which takes $\mathbf{s}'$ and $\widetilde{a}$ as inputs is employed to compute the Q-target y. Here, $\widetilde{a}$ is defined as $C(\mu_{\theta'}(\mathbf{s}') + C(\epsilon, -c, c), a_{low}, a_{high})$. Subsequently, the critic parameters are updated by minimizing the loss function:

$$\frac{1}{N}\sum (y - Q_{\theta_i}(\mathbf{s}, a))^2 \quad (15)$$

using gradient descent. Every u iterations, the actor's parameters θ are adjusted through the deterministic policy gradient method:

$$\frac{1}{N}\sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s) \quad (16)$$

Moreover, the target networks are updated at a rate τ as follows:

$$\theta_i' = \tau\theta_i + (1-\tau)\theta_i',\ \phi' = \tau\phi + (1-\tau)\phi'. \quad (17)$$

Algorithm 1: TD3 Agent Training

```
Given initial critic networks θ1, θ2 and actor network φ
Initialize target networks θ1' ← θ1, θ2' ← θ2, φ' ← φ
Initialize replay buffer B and states s0
for t ← 1 to T do
    Each participant selects an action with exploration noise
      a_t ~ π_φ(s_t) + ε.
    Each participant performs a_t, observes reward r_t and new
      state s_{t+1}.
    Store n transition tuples (s_t, a_t, r_t, s_{t+1}) in B
    if Size of B > batch size z then
        Sample z transitions (s, a, r, s') from B
        For each transition:
          a' = C(π_φ'(s') + C(ε, −c, c), a_low, a_high)
        Each y ← r + γ min_{i=1,2} Q_{θ_i'}(s', a')
        Update critic parameter
          θ_i ← arg min_{θ_i} (1/N) Σ (y − Q_{θ_i}(s, a))^2
        if t mod d then
            Update actor φ by the gradient
              (1/N) Σ ∇_a Q_{θ1}(s, a)|_{a=π_φ(s)} ∇_φ π_φ(s)
            Update target Q networks:
            θ_i' ← τθ_i + (1 − τ) θ_i'
            φ' ← τφ + (1 − τ) φ'
        end
    end
end
```

5 Experiment

In this section, We evaluate SDIE based on two datasets: FSSD-12 [9] and Defect-4^i [1]. There are twelve S^3D classes in FSSD-12, including iron-sheet ash, liquid, oxide-scale, oil-spot, water-spot, patch, punching, red-iron sheet, roll-printing, scratch and inclusion, each type of strip steel defect contains 50 defective images, ground truth (GT), and a large number of normal images. The Defect-4^i dataset contains aluminum, steel, rails, and magnetic tiles that belong to common metal surface defects and adds the nonmetal classes (leather and tile) as extensions to further prove generalization ability. Moreover, all images from both datasets have undergone grayscale processing, and their sizes have been standardized to 200 x 200 to ensure consistency. Furthermore, the number of samples in each defect class is limited to 50 to avoid a long-tailed distribution. In our experiment, we use SAM [17] as our base large model.

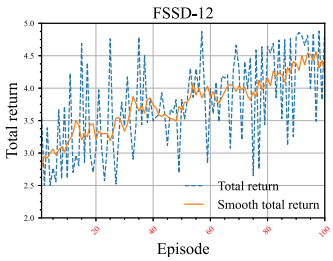

Figure 3. TD3 agent training on 5-shot FSSD-12.

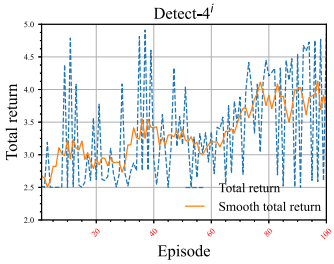

Figure 4. TD3 agent training on 5-shot Detect-4^i.

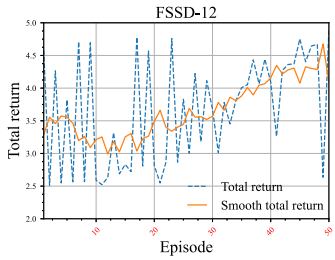

Figure 5. TD3 agent training on 1-shot Detect-4^i.

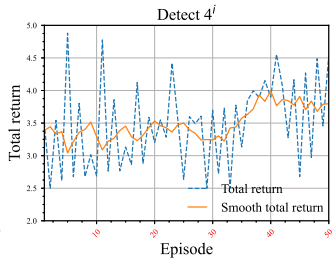

Figure 6. TD3 agent training on 1-shot Detect-4^i.

Table 1. Class MIoU and FB-IoU results on FSSD-12 and Defect-4^i of 1-shot and 5-shot setting.

Dataset	Index	CANet [49]	PGNet [48]	PMMs [44]	PFENet [35]	HSNet [27]	TGRNet [1]	CPANet [9]	Ours
FSSD-12	MIoU	54.4	52.5	50.4	56.0	54.7	58.5	62.6	**62.8**
(5-shot)	FB-IoU	69.2	70.1	67.2	74.0	71.6	75.1	76.3	**77.6**
Defect-4^i	MIoU	21.27	21.32	23.64	31.66	34.82	40.56	42.18	**43.96**
(5-shot)	FB-IoU	53.10	50.01	61.32	54.06	53.93	61.61	59.75	**68.32**
FSSD-12	MIoU	52.3	52.2	50.1	55.3	48.4	57.7	61.5	**61.7**
(1-shot)	FB-IoU	67.8	67.5	66.7	70.3	67.9	73.6	76.1	**76.3**
Defect-4^i	MIoU	19.34	20.25	20.59	27.49	32.80	39.58	39.48	**39.77**
(1-shot)	FB-IoU	49.40	48.89	58.87	53.46	55.85	57.46	56.78	**66.23**

5.1 Agent training

Evaluation Metrics: Following the approach of previous minimal training segmentation studies, the Mean Intersection-over-Union (MIoU) [27, 35, 39] is utilized as the primary metric due to its fairness and all-encompassing nature. For a particular defect category C, the MIoU is determined by the subsequent method:

$$\text{MIoU} = \frac{1}{C}\sum_{c=1}^{C}\text{IoU}_c \tag{18}$$

where IoU_c represents the IoU of defect class c. The Foreground-and-Background Intersection over Union (FB-IoU) disregards the class-specific details, and it was introduced solely to ensure an equitable comparison. The calculation of FB-IoU is as follows:

$$\text{FB-IoU} = \frac{1}{2}(\text{IoU}_f + \text{IoU}_b) \tag{19}$$

where IoU_f and IoU_b denote foreground and background IoU in the target fold, respectively.

Base filter setting: We test various filters such as gamma, median, and Gaussian. Finally, we select the five most effective filters including wls-filter [7], bilateral-filter [29], imsharp-filter [4, 22, 43], guided-filter [14], and [2] to filter the image separately. The results of all filters are weighted and summed up with identity maps to obtain the image after one round of image enhancement. By testing multiple sets of convolution kernels of different sizes, 3, 5, 7, 9, and 11, we ultimately select the size of the convolution kernel as 7.

Reinforcement Learning Settings: We perform 5 rounds of enhancement in each round, and select the round with the highest FB-IoU as the final enhancement result.

In order to better encode images, we use the ResNet18 network architecture to extract image features, concatenate them with weight actions and input multi-layer linear layers as the Q network. We use the ResNet18 network architecture as the architecture of the policy network. During the training process, update the policy network every two updates of the Q network. For 5-shot and 1-shot settings, we select 5 images and 1 image from each S^3D class to form training set. We train TD3 agent on FSSD-12 and Defect-4^i with 1-shot and 5-shot settings as is shown in Figure 3,4,5 and 6 repectively. It's evident from the plots that the return progressively increases and eventually converges.

5.2 Comparison Experiment

Quantitative Result: Table 1 presents the segmentation effectiveness of our SDIE in comparison with other current Fine-Scale Semantic Segmentation (FSS) methods on the FSSD-12 dataset. Whether in the setting of one shot or five shots, Our methods have achieved the best results. Our method achieves 1.2% MIoU and 1.3% FB-IoU improvements over the previous best general segmentation method CPAnet on FSSD-12 in the five shots setting. While in the one shot setup, our method achieves a 0.2% improvement on both MIoU and FB-IoU.

In addition, we also compare our SDIE with the above method on the Defect-4^i dataset under the same settings, and the results are shown in Table 1. From the table, we can see that our method still has significant advantages compared to other advanced FSS methods on the defect dataset. Besides, SDIE has 3.4% MIoU improvement and 6.71% FB-IoU improvement over surface defect segmentation method TGRNet on Defect-4^i in the 5-shot setting. It is worth noting that in the setting of one shot, our method still leads in performance, which has 0.17% MIoU improvement and 8.77% FB-IoU improvement over TGRNet, the model that currently performs average well on several tasks. Additionally, through the above experiments, we can find that our SDIE has strong generalization segmentation performance in the setting of few lenses. Whether in the dataset FSSD-12

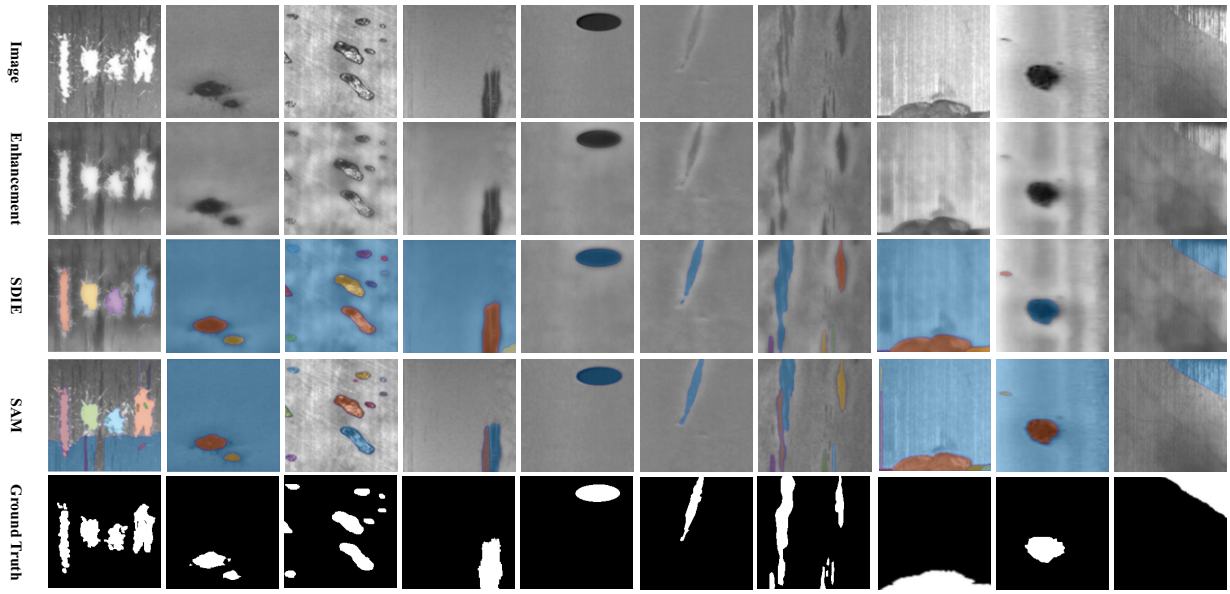

Figure 7. Visualize the experiments results of SDIE. We save the enhanced image with the highest MIoU score and input it into the SAM demo to obtain the segmentation image of our method. At the same time, we also input the unenhanced image into the SAM demo to obtain the corresponding segmentation image. From top to bottom, each row represents base input images, enhancement images, SDIE output, SAM output and corresponding GT.

with all categories of steel strips or in the dataset Defect-4^i with non-industrial defects such as leather and tile, we have achieved relatively good results. This demonstrates the potential application prospects of SDIE in more scenarios in the future.

Ablation study: We conduct ablation experiment between our SDIE and SAM on FSSD-12 and Defect-4^i. These experiments allow the impact of the Image Enhancement module to be evaluated. The Image Enhancement Module can significantly improve the pixel contrast between the defective parts and the background and improve the segmentation performance. The Table 2 shows that the segmentation performance of SDIE will perform 17.6% MIoU increase on FSSD-12 and 8.03% MIoU increase on Defect-4^i over the large model even in the one-shot setup. Of course, after five rounds of shooting training, our model performs better compared to SAM, which has 19.1% MIoU improvement and 12.22% FB-IoU improvement. We also find that without an image enhancement module trained in the one-shot setup, FB-IoU and MIoU decreased by 9.2% and 0.51% on FSSD-12 and Defect-4^i, respectively. Analysis shows that compared to directly handing images over to SAM for segmentation, the significant pixel contrast caused by the image enhancement module can make it easier for SAM to detect defects.

Method	FSSD-12		Defect-4^i	
	MIoU	FB-IoU	MIoU	FB-IoU
SDIE (5-shot)	62.8	77.6	43.96	68.32
SDIE (1-shot)	61.3	76.3	39.77	66.23
SAM	43.7	67.1	31.74	65.72

Table 2. MIoU and FB-IoU of ablation study for image enhancement module trained in 1-shot and 5-shot setup.

5.3 Visualization

As shown in Figure 7, we visualize the segmentation results to analyze better effectiveness of our SDIE. From top to bottom, each row represents base input images, enhancement images, SDIE output, SAM output and corresponding GT. Our SDIE can achieve excellent segmentation performance in most defect classes. We can see that in most images, the difference in segmentation performance between SDIE and SAM models is relatively small. However, from the segmentation results of the first and eighth images, we can see that SDIE does not misclassify the background as a defect, which demonstrates the superiority of our image enhancement module.

6 Conclusion

In this article, we propose a simple and effective method to address the challenges in industrial defect segmentation. Our SDIE consists of a filter module that parameters and weights and determined by deep reinforcement learning and a large SAM model. We conducted comparative and ablation experiments on FSSD-12 and Defect-4^i, and our SDIE achieved state-of-the-art results. However, our SDIE has some failures in segmenting complex defects, such as the segmentation boundaries being not smooth enough. This is likely due to the lack of background knowledge of the dataset and the loss of some information after binarizing the output of the SAM model. We hope that our work can provide some positive insights for future work on existing related challenges.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No.62106105), the CCF-Baidu Open Fund (No.CCF-Baidu202307), and the High Performance Computing Platform of Nanjing University of Aeronautics and Astronautics.

References

[1] Y. Bao, K. Song, J. Liu, Y. Wang, Y. Yan, H. Yu, and X. Li. Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021.

[2] S. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1158–1163 vol. 2, 2005. doi: 10.1109/CVPR.2005.330.

[3] B. D. Brabandere, X. Jia, T. Tuytelaars, and L. V. Gool. Dynamic filter networks, 2016.

[4] A. Chakrabarti. A neural approach to blind motion deblurring, 2016.

[5] T. Chen, L. Zhu, C. Ding, R. Cao, S. Zhang, Y. Wang, Z. Li, L. Sun, P. Mao, and Y. Zang. Sam fails to segment anything?–sam-adapter: Adapting sam in underperformed scenes: Camouflage, shadow, and more. *arXiv preprint arXiv:2304.09148*, 2023.

[6] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023.

[7] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph.*, 27, 08 2008. doi: 10.1145/1360612.1360666.

[8] H. Feng, K. Song, W. Cui, Y. Zhang, and Y. Yan. Cross position aggregation network for few-shot strip steel surface defect segmentation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–10, 2023. doi: 10.1109/TIM.2023.3246519.

[9] H. Feng, K. Song, W. Cui, Y. Zhang, and Y. Yan. Cross position aggregation network for few-shot strip steel surface defect segmentation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–10, 2023.

[10] X. Feng, X. Gao, and L. Luo. X-sdd: A new benchmark for hot rolled steel strip surface defects detection. *Symmetry*, 13(4):706, 2021.

[11] J. A. Filar and K. Vrieze. Competitive markov decision processes. 1996. URL https://api.semanticscholar.org/CorpusID:122474594.

[12] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods, 2018.

[13] S. Ghorai, A. Mukherjee, M. Gangadaran, and P. K. Dutta. Automatic defect detection on hot-rolled flat steel products. *IEEE Transactions on Instrumentation and Measurement*, 62(3):612–621, 2012.

[14] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013. doi: 10.1109/TPAMI.2012.213.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[16] Y. Huang, C. Qiu, and K. Yuan. Surface defect saliency of magnetic tile. *The Visual Computer*, 36(1):85–96, 2020.

[17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[18] W. Kozłowski, M. Szachniewicz, M. Stypułkowski, and M. Zięba. Dimma: Semi-supervised low light image enhancement with adaptive dimming. *arXiv preprint arXiv:2310.09633*, 2023.

[19] J. Li, R. R. Selvaraju, A. D. Gotmare, S. Joty, C. Xiong, and S. Hoi. Align before fuse: Vision and language representation learning with momentum distillation, 2021.

[20] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[21] L. Li, D. Liang, Y. Gao, S.-J. Huang, and S. Chen. All-e: aesthetics-guided low-light image enhancement. *arXiv preprint arXiv:2304.14610*, 2023.

[22] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang. Joint image filtering with deep convolutional networks, 2019.

[23] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning, 2023.

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[25] Q. Luo, X. Fang, Y. Sun, L. Liu, J. Ai, C. Yang, and O. Simpson. Surface defect classification for hot-rolled steel strips by selectively dominant local binary patterns. *IEEE Access*, 7:23488–23499, 2019.

[26] X. Lv, F. Duan, J.-j. Jiang, X. Fu, and L. Gan. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors*, 20(6):1562, 2020.

[27] J. Min, D. Kang, and M. Cho. Hypercorrelation squeeze for few-shot segmentation, 2021.

[28] G. K. Nand, N. Neogi, et al. Defect detection of steel surface using entropy segmentation. In *2014 Annual IEEE India Conference (INDICON)*, pages 1–6. IEEE, 2014.

[29] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. 2009. doi: 10.1561/0600000020.

[30] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression, 2019.

[31] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017.

[32] V. Sharma, A. Diba, D. Neven, M. S. Brown, L. Van Gool, and R. Stiefelhagen. Classification-driven dynamic image enhancement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4033–4041, 2018.

[33] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj. Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*, 31(3):759–776, 2020.

[34] R. Tian and M. Jia. Dcc-centernet: A rapid detection method for steel surface defects. *Measurement*, 187:110211, 2022.

[35] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia. Prior guided feature enrichment network for few-shot segmentation, 2020.

[36] Z. Tian, X. Lai, L. Jiang, S. Liu, M. Shu, H. Zhao, and J. Jia. Generalized few-shot semantic segmentation, 2022.

[37] C. Wang and Z. Jin. Brighten-and-colorize: A decoupled network for customized low-light image enhancement. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8356–8366, 2023.

[38] H. Wang, Z. Li, and H. Wang. Few-shot steel surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2021.

[39] H. Wang, Z. Li, and H. Wang. Few-shot steel surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022. doi: 10.1109/TIM.2021.3128208.

[40] X. Wu, L. Qiu, X. Gu, and Z. Long. Deep learning-based generic automatic surface defect inspection (asdi) with pixelwise segmentation. *IEEE Transactions on Instrumentation and Measurement*, 70:1–10, 2020.

[41] X. Wu, L. Qiu, X. Gu, and Z. Long. Deep learning-based generic automatic surface defect inspection (asdi) with pixelwise segmentation. *IEEE Transactions on Instrumentation and Measurement*, 70:1–10, 2021. doi: 10.1109/TIM.2020.3026801.

[42] W. Xiao, K. Song, J. Liu, and Y. Yan. Graph embedding and optimal transport for few-shot classification of metal surface defect. *IEEE Transactions on Instrumentation and Measurement*, 71:1–10, 2022.

[43] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia. Deep edge-aware filters. In *International Conference on Machine Learning*, 2015.

[44] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye. Prototype mixture models for few-shot semantic segmentation, 2020.

[45] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models, 2022.

[46] N. Yu, H. Shi, and Y. Han. Joint correcting and refinement for balanced low-light image enhancement. *IEEE Transactions on Multimedia*, 2023.

[47] R. Yu, B. Guo, and K. Yang. Selective prototype network for few-shot metal surface defect segmentation. *IEEE Transactions on Instrumentation and Measurement*, 71:1–10, 2022.

[48] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9586–9594, 2019. doi: 10.1109/ICCV.2019.00968.

[49] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5212–5221, 2019. doi: 10.1109/CVPR.2019.00536.

[50] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. URL http://arxiv.org/abs/2303.18223.

[51] N. Zheng, M. Zhou, Y. Dong, X. Rui, J. Huang, C. Li, and F. Zhao. Empowering low-light image enhancer through customized learnable priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12559–12569, 2023.