



ARTICLE

# Asymmetric Loss Based on Image Properties for Deep Learning-Based Image Restoration

Linlin Zhu, Yu Han, Xiaoqi Xi, Zhicun Zhang, Mengnan Liu, Lei Li, Siyu Tan and Bin Yan\*

Henan Key Laboratory of Imaging and Intelligent Processing, PLA Strategic Support Force Information Engineering University, Zhengzhou, 450000, China

\*Corresponding Author: Bin Yan. Email: ybospace@hotmail.com

Received: 10 September 2023 Accepted: 30 October 2023 Published: 26 December 2023

## ABSTRACT

Deep learning techniques have significantly improved image restoration tasks in recent years. As a crucial component of deep learning, the loss function plays a key role in network optimization and performance enhancement. However, the currently prevalent loss functions assign equal weight to each pixel point during loss calculation, which hampers the ability to reflect the roles of different pixel points and fails to exploit the image's characteristics fully. To address this issue, this study proposes an asymmetric loss function based on the image and data characteristics of the image recovery task. This novel loss function can adjust the weight of the reconstruction loss based on the grey value of different pixel points, thereby effectively optimizing the network training by differentially utilizing the grey information from the original image. Specifically, we calculate a weight factor for each pixel point based on its grey value and combine it with the reconstruction loss to create a new loss function. This ensures that pixel points with smaller grey values receive greater attention, improving network recovery. In order to verify the effectiveness of the proposed asymmetric loss function, we conducted experimental tests in the image super-resolution task. The experimental results show that the model with the introduction of asymmetric loss weights improves all the indexes of the processing results without increasing the training time. In the typical super-resolution network SRCNN, by introducing asymmetric weights, it is possible to improve the peak signal-to-noise ratio (PSNR) by up to about 0.5%, the structural similarity index (SSIM) by up to about 0.3%, and reduce the root-mean-square error (RMSE) by up to about 1.7% with essentially no increase in training time. In addition, we also further tested the performance of the proposed method in the denoising task to verify the potential applicability of the method in the image restoration task.

## KEYWORDS

Deep learning; image restoration; loss function; image properties; super resolution; image denoising

## 1 Introduction

Artificial Intelligence (AI) technology has developed significantly in recent decades and achieved success in many fields [1,2] (e.g., robotics, regression analysis, pattern recognition, etc.). Deep learning, as one of the representative techniques of AI technology, has been rapidly developed in the field of computer vision with the improvement of computational resources, especially in image processing



tasks (e.g., denoising [3], super-resolution [4], segmentation [5], and style conversion [6], etc.), where it has demonstrated good processing results. In order to further improve the accuracy of processing results and the effectiveness of information expression in deep learning methods, a large number of network processing models with promising results have been proposed by relevant researchers. In order to optimize the design for specific problems, related researchers have developed new network architectures [7,8]. Meanwhile, to enhance the interpretability of neural networks, a large number of research works have explored understanding the internal mechanisms of neural networks and their inherent limitations. For example, by developing reverse processing networks [9] or trying to spoof networks with specific inputs [10].

The essential components of deep neural networks include forward propagation, backpropagation, optimization, activation function and loss function [11–14]. Forward propagation allows inputs to be passed from one layer to the next until an output is produced. Backpropagation is an iterative process that determines the contribution of each neuron to the output error based on a chain rule and adjusts the weights of each neuron through the network. Optimization techniques are used to reduce the errors generated during backpropagation, and algorithms such as gradient descent and stochastic gradient descent can be used to optimize the network. The activation function converts inputs into outputs that the neural network can recognize. The loss function is used to measure the neural network's performance after backpropagation and optimization. Combining these components allows deep learning to accept complex inputs and generate accurate predictions for various tasks. The loss function measures the predictive power of the network model based on the network-predicted results. It is a crucial component of deep learning models as it quantifies the difference between the model's predictions and the actual values. The correct choice of the loss function is vital for achieving effective optimization of deep learning models since it directly impacts the effectiveness of model training [15].

Image recovery is a crucial area of research in computer vision. It involves restoring the original image information from a damaged image, which is essential in various practical applications like medical image processing, image enhancement, and video compression [16]. Traditional image recovery methods rely on mathematical models but often struggle with noise and distortion in complex scenes. In contrast, deep learning techniques can automatically learn advanced feature representations through end-to-end models and have significantly improved image restoration tasks. However, for delicate image recovery tasks, such as medical image processing, the credibility of deep learning recovery results still limits its popularization and application [17]. When utilizing deep networks for image restoration, a loss function quantifies the difference between a low-quality or corrupted image and the original labeled image. A suitable loss function is vital in improving the network's ability to recover high-quality images from low-quality inputs [18]. Numerous studies have explored different loss functions for image recovery using neural networks. Some commonly used loss functions include Mean Square Error (MSE) and Mean Absolute Error (MAE). These loss functions have shown good performance in various image-processing tasks. For instance, Wang et al. [19] and Zhang et al. [20] employed MSE and MAE, respectively, for image super-resolution. However, these loss functions also have limitations. For example, the MSE loss function may have a significant gap in measuring human-perceived image quality when dealing with tasks related to image quality [21]. This is because the MSE loss function assumes several factors, such as the independence of noise from local image characteristics. Nevertheless, the human visual system (HVS) is sensitive to noise based on local brightness, contrast, and structure [22]. Generally, selecting an appropriate loss function for a specific deep learning task is challenging, and there is no universal selection scheme. The choice depends on the nature of the task and the type of model being used. Using a conventional loss function for training

imposes equal weight on each pixel point, making it difficult to distinguish edge parts. Consequently, suppressing visual artifacts in the network's output image without compromising the true details becomes a key concern.

With the development of deep learning, many researchers have constructed a variety of super-resolution and denoising network models with different frameworks from the design of network models, and these super-resolution and denoising models have achieved relatively good results in super-resolution and denoising problems. However, in the process of network training and processing, in addition to improving the effective utilization of features extracted in the middle of the network, the network training process can be optimized based on the objective so that the information contained in the labels can be fully utilized to improve the processing capability of the network after the completion of training. Especially for different imaging modes in which the physical principles that cause image degradation are different, it is difficult to utilize a single image recovery mode that is applicable to all scenarios, especially for low-contrast images.

The use of asymmetric loss functions has been proposed by related researchers in multi-label classification tasks, especially when the data in each category is not balanced [23,24]. In 2023, Tang et al. [25] constructed a triple representation in the clustering problem, which was further enhanced by different feature constraints for unbalanced data. The asymmetric loss function solves this problem by giving different weights to the losses of different categories. Vogels et al. [26] proposed a modular convolutional architecture for denoising rendered images. The functionality of the kernel prediction network is extended by using a set of asymmetric loss. Liu et al. [27] proposed an asymmetric exponential loss function to address the crack segmentation task sample bias and dataset bias. Depending on the needs of the task, we can adjust the weights of the loss function to balance the importance of different categories. Asymmetric loss function can overcome the drawbacks of the original network to process information equally and improve the processing of the network.

In this paper, we design an asymmetric loss function based on the characteristics of image processing tasks. The loss function considers the greyscale information of each pixel in the image and balances the pixel information of different greyscale values by applying weights to the original loss. Our method focuses on improving the learning process for pixel locations with poor prediction results in the deep learning model. This is achieved by applying dynamic weights to individual pixel points.

This paper uses the models obtained based on MAE and MSE loss functions as comparison models. During processing, the loss function is further optimized according to the data characteristics and features of the original image. In this paper, we change the asymmetry of the loss function by imposing different weight values. We tested the proposed loss in different image processing tasks, and the experimental results show that the asymmetric loss proposed in this paper can improve image processing quality. At the same time, the loss proposed in this paper can be efficiently fused into other network models.

The main innovations of this paper are as follows:

- 1) This paper designs an asymmetric loss function based on the characteristics of the image itself. Assigning different weights to pixels effectively improves the network's processing effect.
- 2) The method proposed in this paper allows for quick implementation of the asymmetric loss function without complicated configuration and adjustment. It has the advantage of plug-and-play, making it more convenient and efficient in practical applications.

3) The method presented in this paper has been tested in various image-processing tasks. It has consistently shown improvement in the network's processing effect. The method demonstrates good robustness and applicability.

## 2 Method

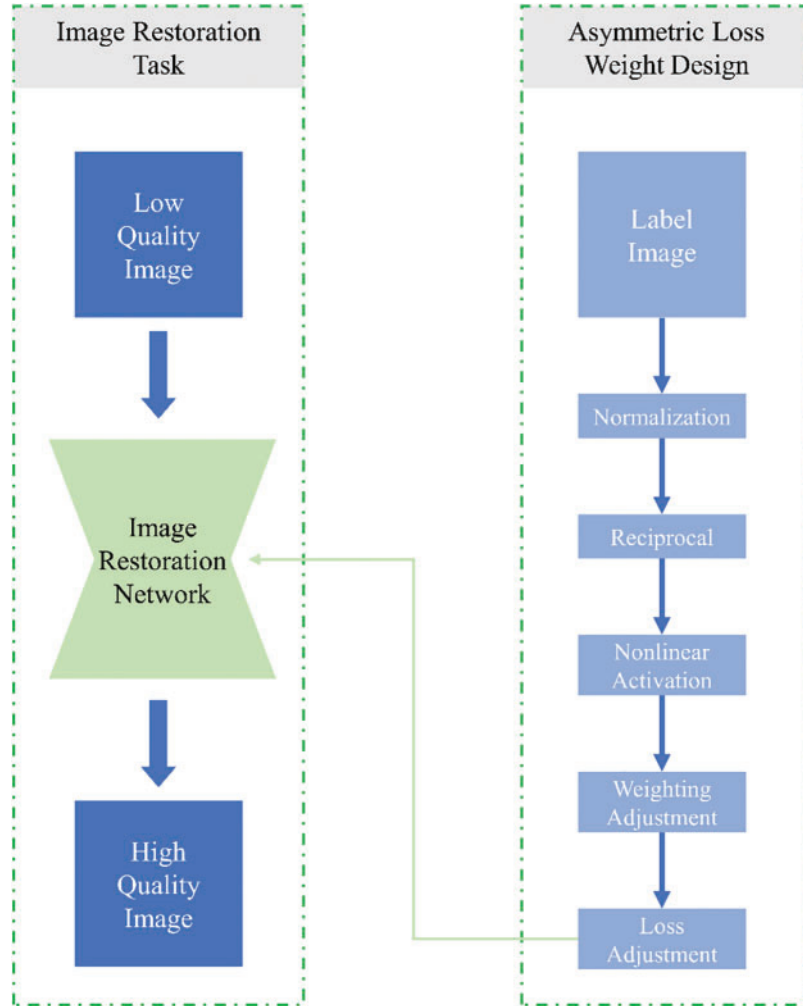
The different levels of greyscale information in an image reflect the brightness of various areas within the image. In a greyscale image, each pixel has a unique greyscale value that indicates the brightness level of that pixel's location. A higher greyscale value corresponds to a lighter pixel in the image. Therefore, analyzing the greyscale values in an image provides insight into its overall brightness distribution and the differences in brightness between different regions. When training the output image using a deep network with a conventional loss function, each pixel carries the same weight, leading to similar deviations across pixels of different greyscale values. However, it is essential to note that deviations among greyscale values of varying magnitudes are significantly different. The optimization effect of individual pixels can be adjusted by assigning weights to address this issue. In this study, we propose an asymmetric loss function that considers the different levels of greyscale. By increasing the loss weights for pixels with large greyscale values, we aim to enhance the image processing effect. These weight adjustments optimize the network parameters and improve the overall performance of the image processing system.

Traditional loss treats each pixel point equally and each pixel point as having the same weight, i.e., a weight of 1 is imposed. The current loss applied in image processing treats each pixel point equally. The total loss  $Err$  is the sum of the deviation values  $err$  of all the pixel points,  $Err = \sum_1^N err_i$ . Here, the deviation value  $err_i$  is the product of the loss of each pixel point and the weight (in this case, 1) of that pixel point. Under this assumption, the difference between the final output of the network optimized based on this loss function and the label should be the minimum total loss  $Err_{min}$  herefore, the bias value  $err$  for each pixel point can be calculated by dividing this minimum total loss by the total number of pixels  $N$ , i.e.,  $err = \frac{Err_{min}}{N}$ . However, in an actual image, the greyscale values of the different pixel points are not consistent. This leads to a significant difference in the performance of these pixel points on the image after generating the same deviation value  $err$ . Specifically, this is since different proportions of changes in pixel grey values can lead to changes in the visual perception of the image, especially in low-contrast regions. This is because in these regions, small changes in grey scale can have a significant impact, thus altering the overall visual appearance of the image.

To enhance the reliability of information in the low-contrast region, asymmetric weights can be used. During the calculation process, weights are added to the low-contrast region. In this way, even a tiny attenuation produces a more significant loss. As a result, in the final result, the magnitude of the information difference of the pixel points with small grey values decreases, enhancing the low-contrast region's information reliability. Specifically, introducing asymmetric weights changes the degree of deviation of each pixel point, making the degree of deviation of the grey values of different pixel points closer to each other instead of being numerically similar. In this way, we can optimize the whole image more effectively. Fig. 1 illustrates the associated asymmetric loss weight making flowchart.

To solve this problem, an asymmetric loss function can be introduced to adjust the optimization effect for pixel points with different grey value sizes. The asymmetric loss function can be designed based on the degree of deviation between the output value and the actual value. Their corresponding loss weight can be increased for pixel points with smaller grey values. Conversely, their corresponding loss weight can be decreased for pixel points with larger grey values. This approach better reflects the

difference in importance of pixel points with different grey value sizes and thus improves the optimization of network parameters. Fig. 2 shows the flowchart of asymmetric loss weight computation based on a specific image.



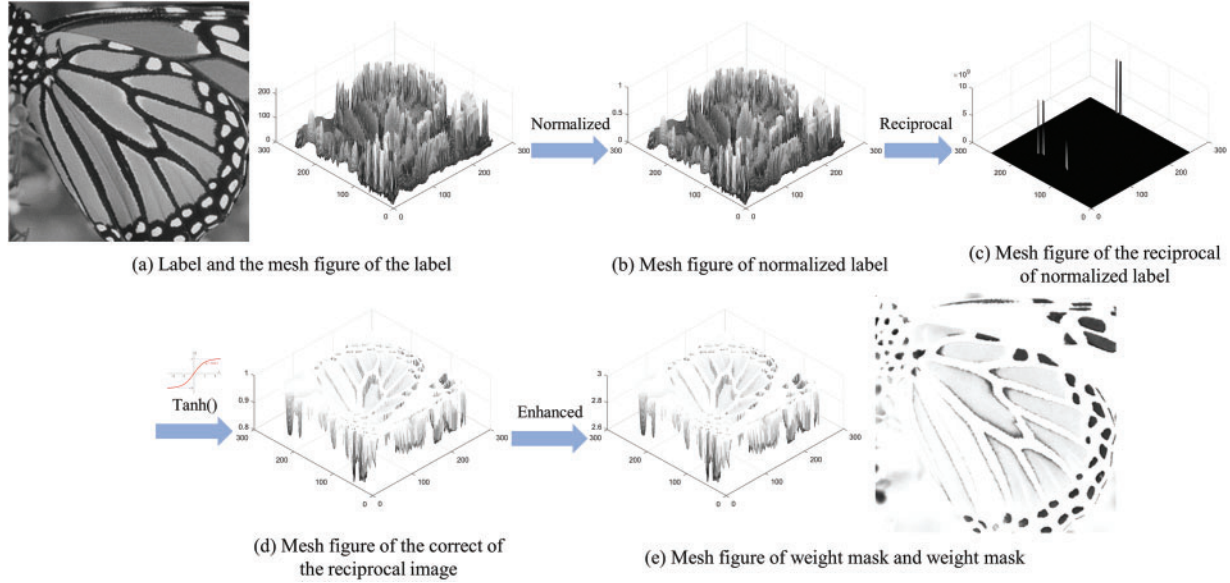
**Figure 1:** The flowchart of the proposed scheme

First, for the given images  $I_{input}$  and  $I_{label}$  with the resolution of  $H \times W \times 3$ , grey value normalization is performed to normalize the grey range to the  $[0,1]$  range.

$$I_n = \frac{(I - I_{min})}{I_{max} - I_{min}}. \tag{1}$$

We have designed an asymmetric weight mask based on the normalized label image,  $I_{l-n}$ , to fully utilize the valid information in the truthful labels. This weight mask converts the original loss into an asymmetric weighted loss by multiplying it with a weighting variable function,  $M$ . This pixel-level mapping,  $M \in R^{H \times W \times 1}$ , is constructed based on the greyscale information of the truth image. Each element of  $M(i,j)$  represents the weights applied to the corresponding pixel coordinate points. In calculating the weight mask for the normalized label image, we specifically focus on pixel points with

smaller grey values. These points tend to impact the results more, so we assign them higher weights in calculating the loss weights. Conversely, we reduce their respective loss weights for pixel points with larger grey values. This modification allows the network to prioritize pixel points that have a greater impact on the results.



**Figure 2:** Flowchart of asymmetric loss weight generation

We use the reciprocal of each pixel point of  $I_{l-n}$  as a weighting benchmark. Since some pixel points have zero grey values, a direct operation of taking the reciprocal may result in a divide-by-zero error. To avoid this, we add a minimal value ( $1 \times e^{-10}$ ) to each pixel point's corresponding labelled image data before calculating its loss weight. This tiny increment ensures that no divide-by-zero error occurs during the computation and maintains the computation's accuracy and stability. Due to the large value of grey scale values close to zero after taking the reciprocal, directly using this value as a weighting factor for calculating the loss value may cause the network to pay too much attention to these pixel points. To solve this problem, we used a hyperbolic tangent function ( $Tanh()$ ) to correct the value after taking the reciprocal. The  $Tanh()$  function is a commonly used nonlinear function centred at the origin and can compress larger values into a range between  $-1$  and  $1$ . By processing the values after taking the inverse through the  $Tanh()$  function, we can make these values more numerically compact, thus better reflecting the importance of the pixel points.

$$M = 0.5 \times \text{Tanh} \left( 1 / (I_{l-n} + 1 \times e^{-10}) \right) + 0.5 \quad (2)$$

We input the normalized  $I_{i-n}$  into the model to get the output  $F(I_{i-n})$  after network processing. Next, we calculate the MSE or MAE between the output and actual values as the basic loss function.

The expression for the MSE loss function is:

$$L_{MSE} (F(I_{i-n}), I_{l-n}) = (F(I_{i-n}) - I_{l-n})^2 \quad (3)$$

The expression for the MAE loss function is:

$$L_{MAE} (F(I_{i-n}), I_{l-n}) = |F(I_{i-n}) - I_{l-n}| \quad (4)$$

Finally, the corrected weight mask is applied to the original loss to obtain the result of the loss calculation after applying the weight mask.

$$L_{Asy}(F(I_{i-n}), I_{i-n}) = M \circ L_{loss} \quad (5)$$

Based on the above description, [Table 1](#) gives the pseudo-code of the method of this paper.

**Table 1:** Pseudo-code of the method

---

Pseudo-code for asymmetric weight calculation:

---

```

//Input: label image
//Output: asymmetric weight values
1: Weight_Mask = Q1 / (label+ 1e-10)
//Calculate weight mask (Weight_Mask), 1e-10 prevents division by 0. Q1 denotes the
smaller quartile, which is equal to the 25th percentile of all grey-scale value values in the labeled
image in descending order.
2: Weight_Mask = TH(Weight_Mask)
//Calculate weight_mask non-linearly, here TH() is a function to judge and process
according to the value of weight_mask.
3: W_out = mul(Weight_Mask, out)
//Multiply the output (out) with the weight mask (Weight_Mask) to get the weighted output
(W_out_train).
4: W_label = mul(Weight_Mask, label)
//Multiply the label value (label) with the weight mask (Weight_Mask) to get the weighted
label (W_label).
5: loss = criterion_loss(W_out, W_label)
//Calculate the loss (loss) between the weighted output (W_out_train) and the weighted target
(W_target_train).

```

---

### 3 Results

#### 3.1 Evaluation Metrics

In this paper, the numerical variability between the processed and ideal reference images is used to evaluate the image quality quantitatively. To quantitatively analyze the corrected image quality of the proposed method in this paper, Peak Signal Noise Ratio (PSNR), Structural Similarity Index Measure, SSIM), and Root Mean Square Errors (RMSE) to measure the accuracy of the output results. The expressions for PSNR, SSIM and RMSE are given below:

$$PSNR = 10 \log_{10} \left( \frac{MAX^2(f)}{\frac{1}{N} \sum_{i=1}^N |f_{Ref}(i) - f(i)|^2} \right) \quad (6)$$

$$SSIM(f_{Ref}, f) = [l(f_{Ref}, f)]^\alpha \cdot [c(f_{Ref}, f)]^\beta \cdot [s(f_{Ref}, f)]^\gamma \quad (7)$$

$$l(f_{Ref}, f) = \frac{2u_{f_{Ref}}u_f + C_1}{u_{f_{Ref}}^2 + u_f^2 + C_1}, \quad c(f_{Ref}, f) = \frac{2\sigma_{f_{Ref}}\sigma_f + C_2}{\sigma_{f_{Ref}}^2 + \sigma_f^2 + C_2}, \quad s(f_{Ref}, f) = \frac{\sigma_{f_{Ref}f} + C_3}{\sigma_{f_{Ref}}\sigma_f + C_3} \quad (8)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N |f_{Ref}(i) - f(i)|^2}{N}} \quad (9)$$

Here  $f_{Ref}$  denotes the ideal reference image,  $f$  denotes the result image after processing by different methods,  $i$  denotes the index value of each pixel in the image, and  $N$  denotes the total number of pixels in the image.  $\alpha > 0$ ,  $\beta > 0$ , and  $\gamma > 0$  represent the weights of luminance  $l(f_{Ref}, f)$ , contrast  $c(f_{Ref}, f)$ , and structural information  $s(f_{Ref}, f)$  in the SSIM calculation, respectively.  $u_{f_{Ref}}$  and  $u_f$  are the mean values of the ideal reference image  $f_{Ref}$  and the result image  $f$  after processing by different methods, respectively, which are used to reflect the brightness information of the image;  $\sigma_{f_{Ref}}$  and  $\sigma_f$  are the standard deviation of the ideal reference image  $f_{Ref}$  and the resultant image  $f$  processed by different methods, respectively, which are used to reflect the contrast information of the image;  $\sigma_{f_{Ref}f}$  is the correlation coefficient of the ideal reference image  $f_{Ref}$  and the resultant image  $f$  processed by different methods, which is used to reflect the similarity of structural information.  $C_1$ ,  $C_2$ , and  $C_3$ , all of which are constant quantities greater than zero, are used to prevent the results from appearing unstable due to a very small or zero denominator.

The closer the RMSE is to zero, the less the numerical variability of the processing result from the ideal labelled image. When the signal-to-noise ratio of the processing result is higher, the larger the PSNR and SSIM value is.

### 3.2 Super-Resolution Task

Firstly, we verify the effectiveness and applicability of the proposed method in this paper on the super-resolution problem in deep learning-based image restoration tasks. SRCNN [28] and EDSR [29] are chosen as the baseline models for the super-resolution task, respectively. MAE and MSE are the baseline losses during the network training process. The network training is completed and tested by introducing asymmetric loss weights designed in this paper based on MAE and MSE losses, respectively. The test dataset is shown in Fig. 3. All these images involved in the test are not included in the training dataset.



**Figure 3:** The ground-truth of the image used for super-resolution task testing

#### 3.2.1 SRCNN

Referring to the parameter settings in the literature [28], we set up the network and selected the BSD300 [30] dataset as the training dataset. The images of the test dataset were obtained after four sets



of trained networks with corresponding super-resolution results. We evaluated the test results based on PSNR, SSIM and RMSE metrics and calculated the corresponding metric values. The obtained results are shown in Table 2. The corresponding metric values show that adding the weights proposed in this paper, the asymmetric loss optimization improves the super-resolution of the network.

**Table 2:** Quantitative results (PSNR/SSIM/RMSE) of the test dataset processing results based on SRCNN with different losses

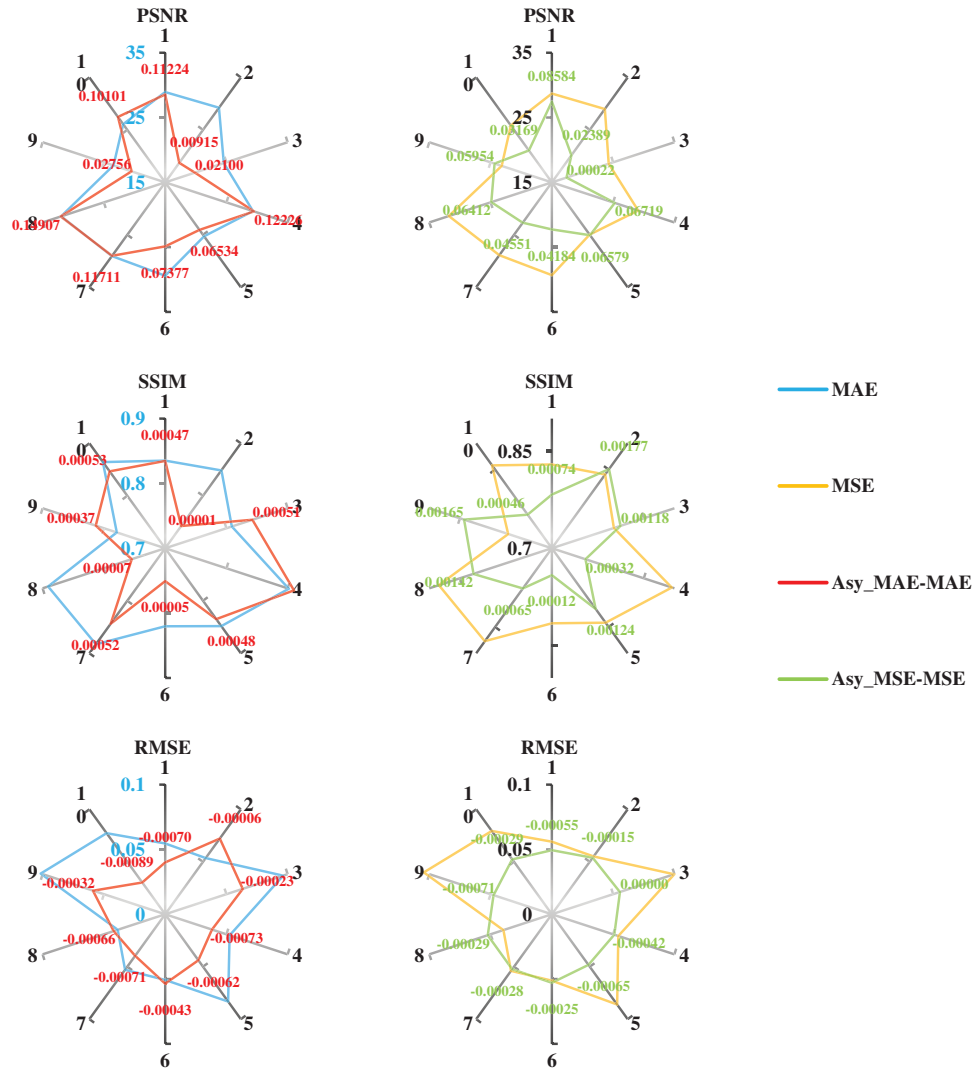
Method No.	MAE	MSE	Asy_MAE	Asy_MSE
1	28.92492/0.83519/0.05455	28.70239/0.82921/0.055968	29.03716/0.83566/0.05385	28.78823/0.82996/0.05542
2	29.22410/0.84854/0.05361	28.97919/0.84070/0.055147	29.23325/0.84855/0.05356	29.00308/0.84246/0.05499
3	24.58361/0.80891/0.09628	24.26785/0.80113/0.09984	24.60460/0.80942/0.09605	24.26807/0.80231/0.09984
4	29.34065/0.90189/0.05256	29.07492/0.89560/0.054195	29.46290/0.90274/0.05183	29.14211/0.89596/0.05378
5	25.27809/0.84832/0.08318	24.97863/0.84177/0.08610	25.34343/0.84879/0.08256	25.04442/0.84301/0.08545
6	29.41628/0.82028/0.05070	29.32586/0.81562/0.05123	29.49005/0.82034/0.05028	29.36769/0.81574/0.05099
7	29.06536/0.88302/0.05294	28.90163/0.87721/0.05395	29.18247/0.88353/0.05224	28.94714/0.87785/0.05367
8	31.98799/0.89101/0.03878	31.81156/0.88441/0.03957	32.13706/0.89109/0.03812	31.87569/0.88583/0.03928
9	23.43703/0.77886/0.10162	23.18690/0.77127/0.10459	23.46459/0.77923/0.10130	23.24645/0.77292/0.10387
10	26.07676/0.86434/0.07734	25.84178/0.85817/0.07946	26.17776/0.86487/0.07645	25.87347/0.85863/0.07917

To comprehensively assess the effectiveness of the asymmetric loss weights proposed in this paper in terms of network processing effect enhancement, we processed the calculated indicator values. Firstly, we use the metric values of the network output results based on MAE and MSE loss as the benchmark. We plotted a radar chart to visualize the numerical results of the processing results for the 10 images. In Fig. 4, the light blue and orange lines represent the evaluation values of the network models trained based on MAE and MSE loss for the super-resolution results of 10 images in the test set, respectively. The red line represents the improvement part of the evaluation values of the network models trained based on asymmetric weighted MAE loss compared to those trained based on MAE loss after the super-resolution processing of 10 images in the test set. The green line represents the improvement part of the evaluation values of the network models trained based on asymmetric weighted MAE loss compared to those trained based on MAE loss after the super-resolution processing of 10 images in the test set.

From Fig. 4, we can see that the model introducing asymmetric weights can improve the metrics performance of the images after super-resolution when super-resolution the images in the test set. By using asymmetric weights, different weights can be assigned to different regions during the super-resolution process, thus better preserving the image's detail information and texture features. Compared to the traditional uniform weight assignment method, asymmetric weighting can capture important details in the image more effectively and reduce noise and distortion. This results in a more metrics performance of the super-resolution image.

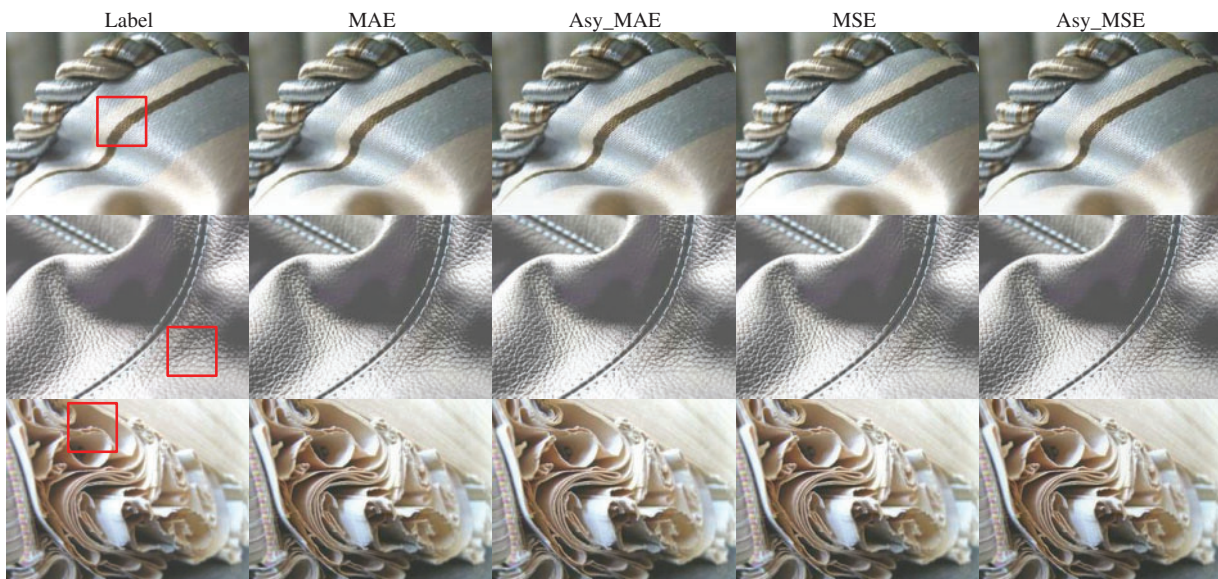
Fig. 5 shows the output results after the training is completed. To facilitate the comparison of the output results after adding asymmetric weights with the original loss, we place the results based on the same loss in neighboring columns. By comparing the entire image, we can find that the super-resolution results of the network with asymmetric gradient weights are visually closer to the label image. To better understand this result, we enlarged the region of interest marked in red in Fig. 6. In Fig. 6, we have marked the locations where the different results are inconsistent with red arrows. Through this comparison, we can find that after adding asymmetric gradient weights, the network has a certain improvement effect on the edge clarity of super-resolution images. This means that the

network can better capture the detailed information in the image, thereby improving the contrast of the restored image.

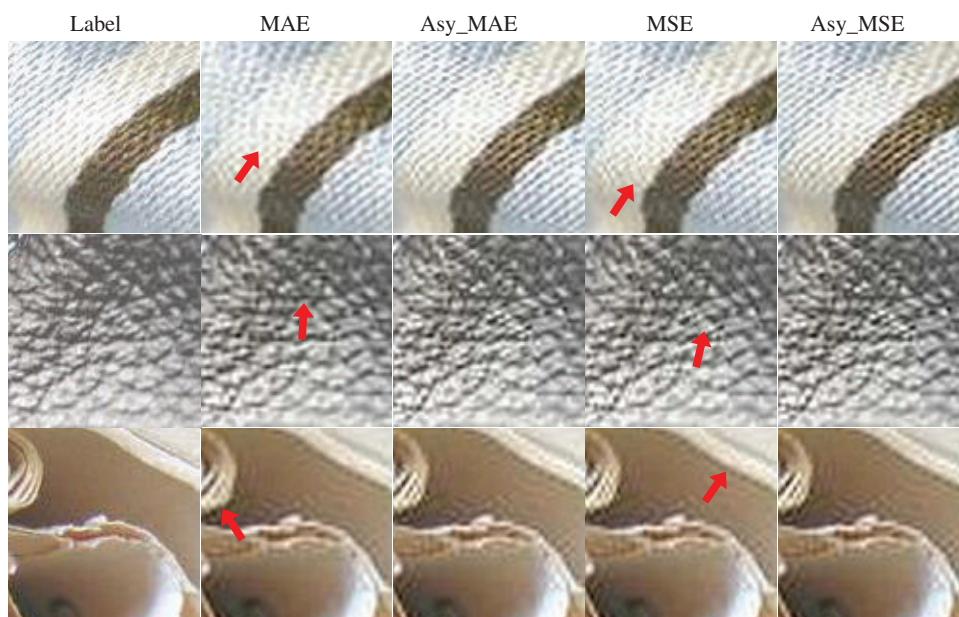


**Figure 4:** Quantitative difference of the results of the test dataset processing by SRCNN based on different losses. The light blue and orange bar charts represent the metric values of the results of 10 images after super-resolution the MAE loss-based and MSE loss-based network models, respectively. The red and green line graphs represent the difference between the metric values of the super-resolution results of the model after testing of the 10 images with the introduction of asymmetric weights for the MAE loss and the introduction of asymmetric weights for the MSE loss, respectively

By introducing asymmetric gradient weights during network training, we successfully improved the super-resolution performance of the network. This makes the output visually closer to the label image and improves the edge clarity and contrast of the image. These improvements are significant for image processing tasks as they can help improve image quality and application value.



**Figure 5:** The output images generated by SRCNN models based on different loss function



**Figure 6:** The corresponding magnified images of the region of interest by the red marked locations in Fig. 5

### 3.2.2 EDSR

Referring to the parameter settings in the literature [29], we set up the network and selected the BSD300 dataset as the training dataset. The images of the test dataset were obtained after four sets of trained networks with corresponding super-resolution results. We evaluated the test results based on PSNR, SSIM and RMSE metrics and calculated the corresponding metric values. The obtained results

are shown in Table 3. In order to better analyze and interpret the impact of the method proposed in this paper on the network performance, we visualized the data in Table 3.

**Table 3:** Quantitative results (PSNR/SSIM/RMSE) of the test dataset processing results based on EDSR with different losses

Method No.	MAE	MSE	Asy_MAE	Asy_MSE
1	26.51871/0.82646/0.07196	13.42654/0.53437/0.32489	28.58646/0.84265/0.05672	16.52055/0.61927/0.22753
2	26.12353/0.85759/0.07661	13.49726/0.59760/0.32781	28.80974/0.86317/0.05623	18.72635/0.70140/0.17954
3	24.37370/0.82594/0.09863	12.80365/0.51736/0.37371	25.15233/0.82774/0.09018	17.11179/0.61507/0.22757
4	26.39623/0.88009/0.07377	13.42304/0.55654/0.32851	28.72694/0.90457/0.05641	17.27729/0.64743/0.21078
5	24.78057/0.85987/0.08809	13.11221/0.50552/0.33755	25.86106/0.86268/0.07778	16.23997/0.61721/0.23548
6	26.32489/0.78169/0.07238	13.52573/0.50731/0.31591	28.39762/0.81880/0.05701	15.46339/0.58417/0.25275
7	26.15808/0.87147/0.07399	13.38455/0.60064/0.32200	28.42011/0.89070/0.05703	16.96121/0.68008/0.21332
8	27.23225/0.88793/0.06704	13.54082/0.59661/0.32427	30.38947/0.89705/0.04661	17.78649/0.68617/0.19890
9	23.21293/0.78917/0.10427	13.02648/0.50109/0.33690	23.95918/0.79568/0.09569	16.95150/0.59956/0.21441
10	24.94953/0.85841/0.08806	13.22688/0.55174/0.33956	26.23638/0.86952/0.07594	17.19596/0.65075/0.21501

In Fig. 7, the light blue lines represent the measurement values of the results of 10 images after super-resolution of the network model based on MAE loss. The organ line represents the difference between the measurement values of the super-resolution results of the network model with asymmetric weights and the measurement values of the super-resolution results obtained based on MAE losses after testing 10 images. The red lines represent the measurement values of the results of 10 images after super-resolution of the network model based on MAE loss. The green line represents the difference between the measurement values of the super-resolution results of the network model with asymmetric weights and the measurement values of the super-resolution results obtained based on MAE losses after testing 10 images.

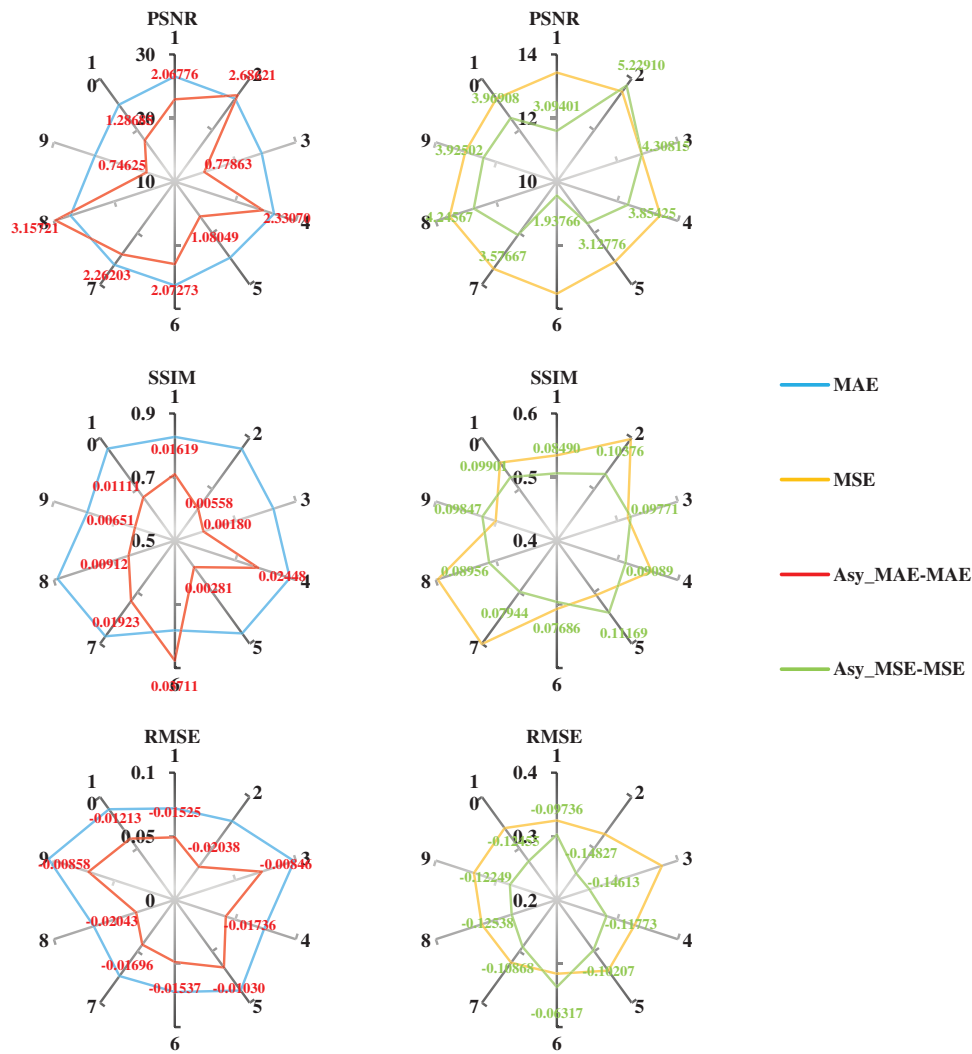
In the PSNR, SSIM and RMSE subplots of Fig. 7, by introducing asymmetric weights, we can see that the output outcome metrics are evaluated better, i.e., the obtained values are higher than the mean values in PSNR and SSIM, while in RMSE, the obtained values are lower than the mean values. This result shows that by using asymmetric loss weights, the model can be guided to pay more attention to grey scale information, which helps to improve the performance of the model.

### 3.3 Denoising Task

In order to verify the applicability of the method proposed in this paper to the image restoration problem, the paper is further tested on a deep learning-based denoising model. The denoising problem was addressed using the DnCNN [31] and DPHSIR [32] models as baseline models. The baseline losses were measured using MAE and MSE during the network training process. The network training was completed and tested by incorporating the asymmetric loss weights introduced in this paper. The test dataset is shown in Fig. 8. All these images involved in the test are not included in the training dataset.

#### 3.3.1 DnCNN

Referring to the parameter settings in the literature [31], we set up the network and selected the DIV2K [33] dataset as the training dataset. The DIV2K dataset is a newly proposed high-quality (2K resolution) image dataset for image restoration tasks.



**Figure 7:** Quantitative difference of the results of the test dataset processing by EDSR based on different losses. The light blue and orange bar charts represent the metric values of the results of 10 images after super-resolution the MAE loss-based and MSE loss-based network models, respectively. The red and green line graphs represent the difference between the metric values of the super-resolution results of the model after testing of the 10 images with the introduction of asymmetric weights for the MAE loss and the introduction of asymmetric weights for the MSE loss, respectively

The images of the test dataset were obtained after four sets of trained networks with corresponding denoising results. We evaluated the test results based on PSNR, SSIM and RMSE metrics and calculated the corresponding metric values. The obtained results are shown in [Table 4](#).

For the convenience of observation and analysis, we further processed and visualized the obtained indicator values. We plotted a radar chart to visually display the numerical results of the processing results of 10 images. In [Fig. 9](#), the light blue and orange lines represent the evaluation values of the network models trained based on MAE loss for the super-resolution results of 10 images in the test set. The red line represents the improvement part of the evaluation values of the network models trained

based on asymmetric weighted MAE loss compared to those trained based on MAE loss after the super-resolution processing of 10 images in the test set. The green line represents the improvement part of the evaluation values of the network models trained based on asymmetric weighted MAE loss compared to those trained based on MAE loss after the super-resolution processing of 10 images in the test set.



**Figure 8:** The ground-truth of the image used for denoising task testing

**Table 4:** Quantitative results (PSNR/SSIM/RMSE) of the test dataset processing results based on DnCNN with different losses

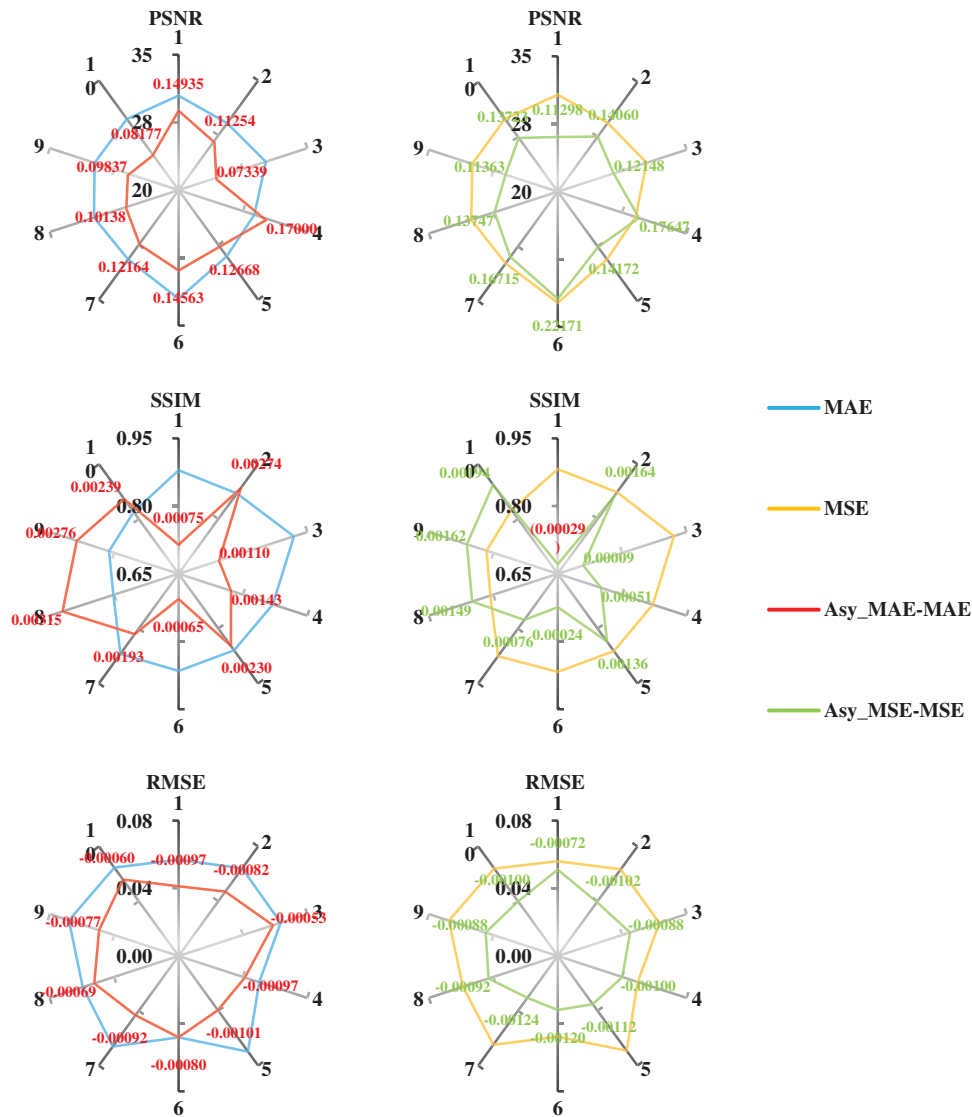
Method No.	MAE	MSE	Asy_MAE	Asy_MSE
1	30.47316/0.87933/0.05695	30.61518/0.88183/0.05603	30.6225/0.88008/0.05598	30.72817/0.88155/0.05530
2	29.17037/0.87025/0.06402	29.27422/0.87332/0.06326	29.28291/0.87298/0.06320	29.41482/0.87496/0.06225
3	30.16740/0.91865/0.06341	30.21514/0.92076/0.06306	30.24078/0.91975/0.06287	30.33662/0.92085/0.06218
4	28.82701/0.86976/0.05009	28.90846/0.87148/0.04962	28.99701/0.87119/0.04912	29.08492/0.87199/0.04862
5	29.05744/0.85938/0.06981	29.13556/0.86136/0.06918	29.18413/0.86168/0.06880	29.27727/0.86272/0.06806
6	32.04077/0.86530/0.04809	32.11249/0.86745/0.04769	32.1864/0.86596/0.04729	32.33421/0.86769/0.04649
7	29.48723/0.86928/0.06607	29.65442/0.87550/0.06481	29.60887/0.87121/0.06515	29.82157/0.87626/0.06357
8	29.85495/0.80191/0.05948	29.94678/0.80684/0.05886	29.95633/0.80506/0.05879	30.08425/0.80834/0.05793
9	29.78228/0.81200/0.06812	29.83843/0.81615/0.06769	29.88065/0.81476/0.06736	29.95206/0.81777/0.06681
10	29.73983/0.81943/0.06452	29.81981/0.82456/0.06393	29.82160/0.82182/0.06392	29.95705/0.82650/0.06293

By analyzing the radar charts, introducing the asymmetric loss weights proposed in this paper can significantly improve the index values of the processing results compared to the results of MAE and MSE. By processing the calculated index values and analyzing them by drawing radar plots, we can see the effectiveness of asymmetric loss weights in improving the processing results of the network. This indicates that asymmetric loss weighting significantly improves the network processing effect.

### 3.3.2 DPHSIR

Referring to the parameter settings in the literature [32], we set up the network and selected the DIV2K dataset as the training dataset. We have performed the same processing operation on the test

results of the DPHSIR network with reference to the processing of the output results of the DnCNN network in Section 3.3.1. The calculated PSNR, SSIM, and RMSE results are shown in Table 5.



**Figure 9:** Quantitative result enhancement values of the test dataset processing results based on DnCNN with different losses. The light blue and orange bar charts represent the metric values of the results of 10 images after super-resolution the MAE loss-based and MSE loss-based network models, respectively. The red and green line graphs represent the difference between the metric values of the super-resolution results of the model after testing of the 10 images with the introduction of asymmetric weights for the MAE loss and the introduction of asymmetric weights for the MSE loss, respectively

Fig. 10 gives the resultant radar plots of PSNR, SSIM and RMSE metrics for the test results of the test data. In Fig. 10, the light blue and orange lines represent the evaluation values of the network models trained based on MAE loss for the super-resolution results of 10 images in the test set. The red line represents the improvement part of the evaluation values of the network models trained based

on asymmetric weighted MAE loss compared to those trained based on MAE loss after the super-resolution processing of 10 images in the test set. The green line represents the improvement part of the evaluation values of the network models trained based on asymmetric weighted MAE loss compared to those trained based on MAE loss after the super-resolution processing of 10 images in the test set.

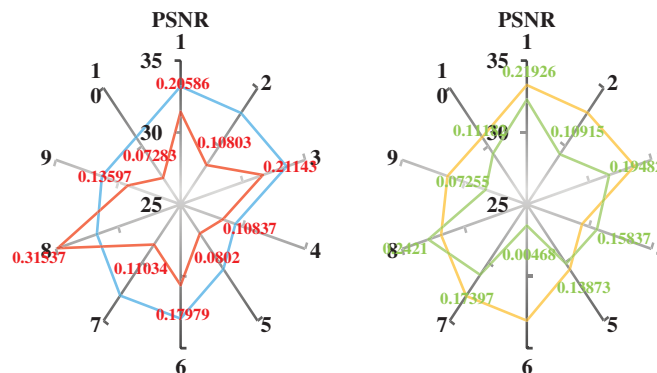
**Table 5:** Quantitative results (PSNR/SSIM/RMSE) of the test dataset processing results based on DPHSIR with different losses

Method No.	MAE	MSE	Asy_MAE	Asy_MSE
1	33.20231/0.85755/0.04807	33.29210/0.86228/0.04758	33.40817/0.86133/0.04695	33.51136/0.86291/0.04639
2	32.84947/0.89071/0.05936	32.87506/0.89098/0.05918	32.95750/0.89188/0.05862	32.98421/0.89102/0.05844
3	33.50950/0.93315/0.06314	33.64608/0.93252/0.06215	33.72094/0.93448/0.06162	33.84090/0.93323/0.06077
4	29.40432/0.82005/0.06823	29.43716/0.82345/0.06797	29.51269/0.82365/0.06738	29.59553/0.82600/0.06674
5	30.56785/0.84959/0.07763	30.58365/0.85225/0.07749	30.64805/0.85161/0.07692	30.72238/0.85345/0.07626
6	32.94595/0.87258/0.03721	33.08510/0.87443/0.03662	33.12574/0.87364/0.03645	33.08978/0.87563/0.03660
7	32.83670/0.86446/0.06611	32.87069/0.86539/0.06585	32.94704/0.86531/0.06527	33.04465/0.86670/0.06454
8	31.71205/0.82028/0.03576	31.87406/0.82429/0.03510	32.02742/0.82319/0.03448	32.11616/0.82540/0.03413
9	31.30851/0.80959/0.04633	31.36823/0.81272/0.04601	31.44448/0.81205/0.04561	31.44078/0.81362/0.04563
10	30.76400/0.85245/0.07317	30.78315/0.85461/0.07301	30.83683/0.85422/0.07256	30.89497/0.85463/0.07208

In the PSNR, SSIM and RMSE subgraphs of Fig. 10, we observed that the measurement of loss training results was better than the results processed by the network without introducing asymmetric weights. This result indicates that the asymmetric loss weight proposed in this article will positively impact loss calculation and network optimization processes.

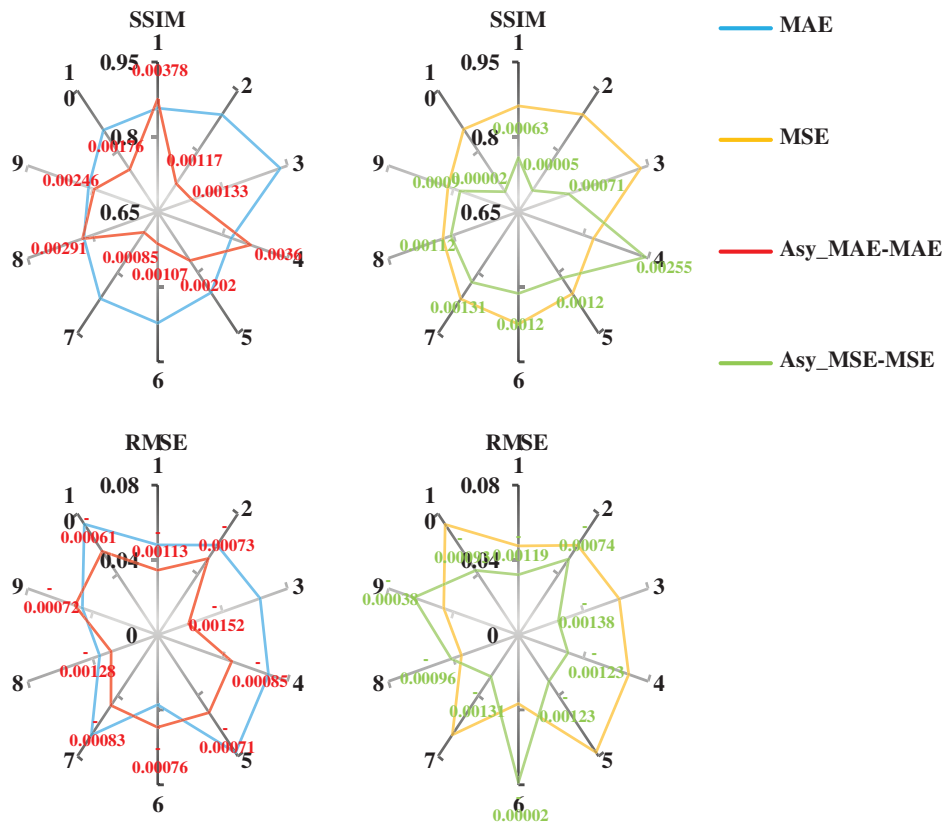
Asymmetric weights can also help networks better understand the importance of different pixels with different greyscale values. In traditional loss functions, it is usually assumed that the greyscale values of all pixels are the same, but this is not the case in reality. Therefore, introducing asymmetric weights can help the network distinguish pixels of different greyscale values, thereby achieving better network optimization after training.

Precisely, the introduction of asymmetric weights adjusts the attention level of the network to different pixel points during the training process, enabling the network to better focus on essential areas in the image, thereby improving the learning efficiency of the network. Meanwhile, due to the design of asymmetric weights, the network can better adapt to various image processing tasks, including super-resolution, denoising, etc.



**Figure 10:** (Continued)





**Figure 10:** Quantitative result enhancement values of the test dataset processing results based on DPHSIR with different losses. The light blue and orange bar charts represent the metric values of the results of 10 images after super-resolution the MAE loss-based and MSE loss-based network models, respectively. The red and green line graphs represent the difference between the metric values of the super-resolution results of the model after testing of the 10 images with the introduction of asymmetric weights for the MAE loss and the introduction of asymmetric weights for the MSE loss, respectively

#### 4 Discussion

This paper’s asymmetric loss function is designed for deep network-based image regression problems. Image regression is a problem of predicting the corresponding value of an image by processing it, which is widely used in the fields of image super-resolution, image denoising and so on. The asymmetric loss function is a loss function that improves the network processing effect by setting different weight values for different pixel points. In the traditional uniform loss function, all pixel points are given the same weight, which often fails to utilize the image information fully. Because each pixel point in an image contains unequal information, pixel points with more critical information should be given more weight.

A deep network is a multilayer neural network with strong expressive ability and generalization performance, which is increasingly widely used in image processing. Deep networks can recover the information of the image better. However, the subsequent processing tasks have a higher and higher demand for the recovery accuracy and detailed information of the image. The rich information contained in the truth label can improve the training effect of the network, but the current widely

used loss function has the same weight for each pixel point, which makes it difficult to mention the role of different pixel points.

To solve this problem, this paper proposes an asymmetric loss function. The asymmetric loss function differentiates the weight settings for different pixel points according to the truth labels' characteristics and the processing task's needs, which better achieves the training optimization of the network. Specifically, the asymmetric loss function weights the loss of each pixel point so that the pixel points with more important information occupy a more significant weight in the loss calculation. This can induce the network to pay more attention to the pixel points with more critical information in the image during the training process, thus improving the processing effect of the network.

## 5 Conclusion

This paper uses an asymmetric loss function to solve the deep network-based image regression problem. In image processing, the image regression problem is an important task that aims to predict the corresponding output values based on a given image. However, the traditional loss function has some limitations in dealing with such problems and cannot fully explore the image information.

The emergence of asymmetric loss function provides a new way to solve these problems. The asymmetric loss function is a loss function that differentiates different pixel points by setting different weight values. In the specific implementation, the asymmetric loss function assigns different weights to each pixel point according to the importance of the pixel point. In the network training process, the asymmetric loss function can make better use of the image information and improve the network's processing effect and training optimization.

In order to verify the effectiveness of the asymmetric loss function, we conducted experimental tests in image super-resolution and image denoising tasks. The experimental results show that the deep network model based on the asymmetric loss function exhibits significant advantages in these tasks. The asymmetric loss function can better highlight the image's detailed information, improve the image's visual effect, and make it easier to analyze and process the subsequent tasks.

In addition, the asymmetric loss function proposed in this paper is mainly realized by setting the weight values. This design idea makes combining with other loss functions easy to form a more powerful loss function. This design idea is essential for designing loss functions in other tasks and can be extended to other image-processing or non-image-processing tasks.

In the future, we can further study more complex and diverse image properties and explore other effective asymmetric loss function design methods. For example, in computer vision, we can introduce more feature extraction methods and deep learning models to improve the quality of image recovery. In addition, we can design and optimize different asymmetric loss functions for different tasks and datasets to adapt to the needs of different scenarios. In addition to the field of computer vision, asymmetric loss functions can also be applied to other fields, such as natural language processing, speech recognition, etc. Of course, applying asymmetric loss function to other fields also needs to consider its applicability and feasibility. We need to choose the appropriate asymmetric loss function according to the specific task and the characteristics of the dataset and conduct the corresponding experiments and verification. At the same time, we also need to pay attention to issues such as the interpretability and robustness of the model to ensure the stability and reliability of the model.

**Acknowledgement:** None.

**Funding Statement:** This work was supported by the National Natural Science Foundation of China (62201618).

**Author Contributions:** Conceptualization, L.Z. and Y.H.; methodology, Y.H.; software, L.Z.; validation, L.Z., Y.H. and X.X.; formal analysis, L.Z. and Y.H.; investigation, L.Z.; resources, B.Y.; data curation, Z.Z., M.L. and X.X.; writing-original draft preparation, L.Z. and Y.H.; writing-review and editing, S.T., L.L., B.Y. and X.X.; visualization, L.Z., Z.Z. and M.L.; supervision, B.Y.; project administration, X.X.; funding acquisition, Y.H. and L.L. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The data and the code used for the manuscript are available for researchers on request from the corresponding author.

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] B. Zhang, J. Zhu and H. Su, "Toward the third generation artificial intelligence," *Science China Information Sciences*, vol. 66, no. 2, pp. 121101, 2023.
- [2] Y. Xu, X. Liu, X. Cao, C. Huang, E. Liu *et al.*, "Artificial intelligence: A powerful paradigm for scientific research," *The Innovation*, vol. 2, no. 4, pp. 100179, 2021.
- [3] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo *et al.*, "Deep learning on image denoising: An overview," *Neural Networks*, vol. 131, pp. 251–275, 2020.
- [4] Z. Wang, J. Chen and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3365–3387, 2021.
- [5] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz *et al.*, "Image segmentation using deep learning: A survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1, 2021.
- [6] P. Isola, J. Y. Zhu, T. Zhou and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 5967–5976, 2017.
- [7] H. C. Burger, C. J. Schuler and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp. 2392–2399, 2012.
- [8] L. Xu, J. S. J. Ren, C. Liu and J. Jia, "Deep convolutional neural network for image deconvolution," in *Proc. of the 27th Int. Conf. on Neural Information Processing Systems*, Cambridge, MA, USA, pp. 1790–1798, 2014.
- [9] M. D. Zeiler, D. Krishnan, G. W. Taylor and R. Fergus, "Deconvolutional networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp. 2528–2535, 2010.
- [10] A. Nguyen, J. Yosinski and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 427–436, 2015.
- [11] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss functions for neural networks for image processing," 2015. [Online]. Available: <https://arxiv.org/abs/1511.08861> (accessed on 26/10/2023)
- [12] G. Xie and J. Lai, "An Interpretation of Forward-Propagation and Back-Propagation of DNN," in *Chinese Conf. on Pattern Recognition and Computer Vision*, Guangzhou, China, pp. 3–15, 2018.
- [13] R. Y. Sun, "Optimization for deep learning: An overview," *Journal of the Operations Research Society of China*, vol. 8, no. 2, pp. 249–294, 2020.
- [14] B. Ding, H. Qian and J. Zhou, "Activation functions and their characteristics in deep neural networks," in *Chinese Control and Decision Conf.*, Shenyang, China, pp. 1836–1841, 2018.

- [15] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza and E. A. Chavez-Urbiola, "Loss functions and metrics in deep learning. A review," 2023. [Online]. Available: <https://arxiv.org/abs/2307.02694> (accessed on 26/10/2023)
- [16] J. Su, B. Xu and H. Yin, "A survey of deep learning approaches to image restoration," *Neurocomputing*, vol. 487, pp. 46–65, 2022.
- [17] A. A. Laghari, V. V. Estrela and S. Yin, "How to collect and interpret medical pictures captured in highly challenging environments that range from nanoscale to hyperspectral imaging," *Current Medical Imaging*, vol. 19, pp. 1–10, 2023.
- [18] Y. Tian, D. Su, S. Lauria and X. Liu, "Recent advances on loss functions in deep learning for computer vision," *Neurocomputing*, vol. 497, pp. 129–158, 2022.
- [19] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung *et al.*, "A fully progressive approach to single-image super-resolution," 2018. [Online]. Available: <https://arxiv.org/abs/1804.02900> (accessed on 26/10/2023)
- [20] Y. Zhang, Y. Tian, Y. Kong, B. Zhong and Y. Fu, "Residual dense network for image super-resolution," 2018. [Online]. Available: <https://arxiv.org/abs/1802.08797> (accessed on 26/10/2023)
- [21] L. Zhang, L. Zhang, X. Mou and D. Zhang, "A comprehensive evaluation of full reference image quality assessment algorithms," in *IEEE Int. Conf. on Image Processing*, Orlando, FL, USA, pp. 1477–1480, 2012.
- [22] 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.
- [23] X. Zhou, X. Liu, J. Jiang, X. Gao and X. Ji, "Asymmetric loss functions for learning with noisy labels," 2021. [Online]. Available: <https://arxiv.org/abs/2106.03110v1> (accessed on 26/10/2023)
- [24] S. R. Hashemi, S. S. Mohseni Salehi, D. Erdogmus, S. P. Prabhu, S. K. Warfield *et al.*, "Asymmetric loss functions and deep densely-connected networks for highly-imbalanced medical image segmentation: Application to multiple sclerosis lesion detection," *IEEE Access*, vol. 7, pp. 1721–1735, 2019.
- [25] Y. Tang, J. Huang, W. Pedrycz, B. Li and F. Ren, "A fuzzy clustering validity index induced by triple center relation," *IEEE Transactions on Cybernetics*, vol. 53, no. 8, pp. 5024–5036, 2023.
- [26] T. Vogels, F. Rousselle, B. McWilliams, G. Röthlin, A. Harvill *et al.*, "Denoising with kernel prediction and asymmetric loss functions," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 124:1–124:15, 2018.
- [27] F. Liu, J. Wang, D. Chen, C. Shen and F. Xu, "Asymmetric exponential loss function for crack segmentation," *Multimedia Systems*, vol. 29, no. 2, pp. 539–552, 2023.
- [28] C. Dong, C. C. Loy, K. He and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conf. on Computer Vision*, Zurich, Switzerland, pp. 184–199, 2014.
- [29] B. Lim, S. Son, H. Kim, S. Nah and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Honolulu, HI, USA, pp. 1132–1140, 2017.
- [30] D. Martin, C. Fowlkes, D. Tal and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE Int. Conf. on Computer Vision*, Vancouver, BC, Canada, vol. 2, pp. 416–423, 2001.
- [31] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [32] Z. Lai, K. Wei and Y. Fu, "Deep plug-and-play prior for hyperspectral image restoration," *Neurocomputing*, vol. 481, pp. 281–293, 2022.
- [33] R. Timofte, E. Agustsson, L. V. Gool, M. H. Yang, L. Zhang *et al.*, "Challenge on single image super-resolution: Methods and results," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Honolulu, HI, USA, pp. 1110–1121, 2017.