EdgeSR: Image Super-Resolution Using Edge-Guided Diffusion Models

Armine Panosyan[1], Levon Khachatryan[1,2]
[1]Yerevan State University (YSU) [2]Picsart AI Resarch (PAIR)

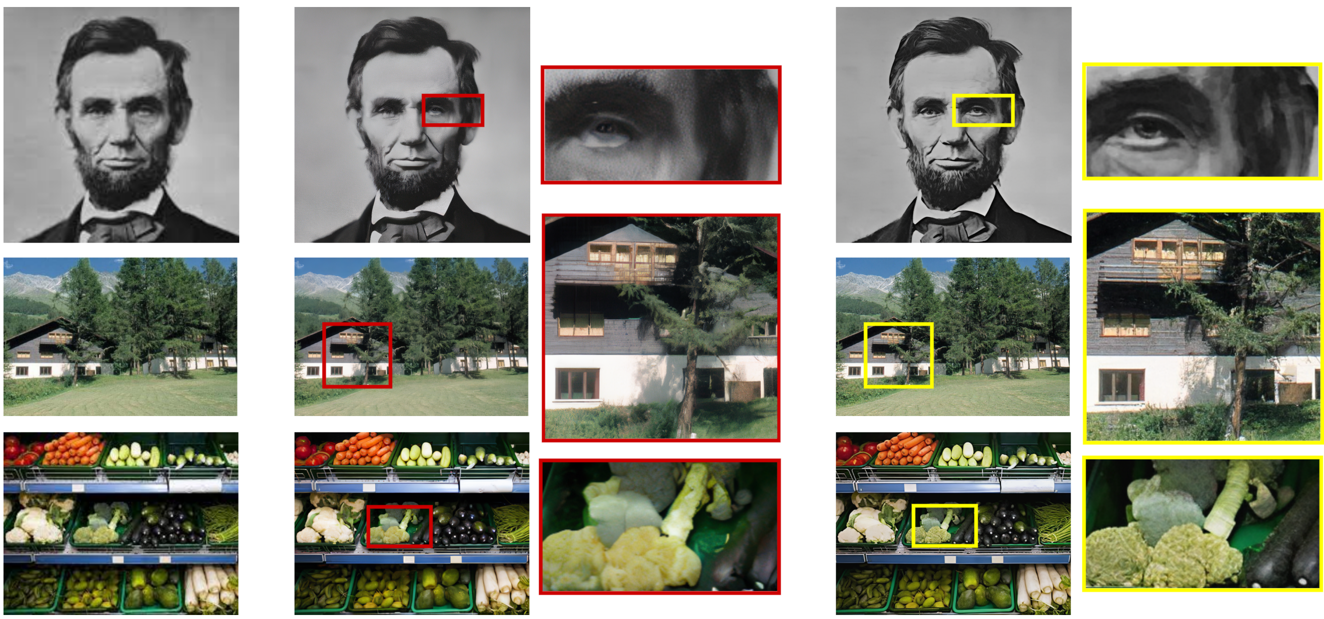

Figure 1. **EdgeSR** is a cutting-edge super-resolution technology designed to enhance the clarity and detail of images at the edge level.Our demonstrations include successful examples of images enhanced using **EdgeSR**, compared to the results produced by the **ResShift** model, which is currently recognized as a state-of-the-art technique for Single Image Super-Resolution (SISR).

Abstract

In the pursuit of advancing image super-resolution (SR) technologies, diffusion models have emerged as a groundbreaking methodology, offering unparalleled potential for generating high-quality, high-resolution images from their lower-resolution counterparts. However, traditional approaches to diffusion-based SR have grappled with significant challenges, notably the computational intensity required for the process. Traditional models necessitate hundreds to thousands of sampling steps to achieve satisfactory results, leading to a slow inference speed that hampers practical application. Additionally, attempts to accelerate this process often result in a noticeable degradation of image quality, producing outputs that lack sharpness and detail. We hypothesize that edge maps are often discernible in low-resolution images, yet after super-resolution, we may lose some edge details. Thus, our research presents a novel plug-and-play module for any diffusion-based image super-resolution (SR) method, which improves image details by incorporating an edge detection algorithm into the reverse diffusion process. By the activation of edge detection in the stages of the reverse diffusion process, we optimize both the efficiency and effectiveness of our model, ensuring exceptional clarity and detail in the final image output.

1. Introduction

In the dynamic field of computer vision, image super-resolution (SR) is a crucial challenge that focuses on generating high-resolution (HR) images from low-resolution (LR) ones. Super-resolution is particularly vital in areas such as medical imaging, where it enables clearer diagnostic images; surveillance, where it enhances security footage;

remote sensing and satellite image processing, which require more detailed environmental analysis; film production, which benefits from higher resolution video; scientific research, which needs improved microscopic images; and in video games and virtual reality, where it contributes to more immersive environments [25]. Recently, diffusion models have revolutionized generative modeling, offering significant advancements in image generation. These models are particularly effective at capturing and reconstructing intricate image details, making them highly suitable for super-resolution tasks.

Super-resolution, crucial in enhancing image detail, utilizes advanced methods including Generative Adversarial Networks (GANs) and diffusion models. GANs, particularly through Super-Resolution GAN (SRGAN), employ a generator and discriminator to produce high-resolution images from low-resolution inputs. While GANs are known for generating visually convincing enhancements, they can also introduce artifacts and sometimes produce unnatural results [13]. Diffusion models offer two approaches to super-resolution: inputting low-resolution images directly into models like Denoising Diffusion Probabilistic Models (DDPM) for retraining with high-resolution data, or modifying the reverse generation path of a pre-trained model to upscale images. Both methods, while effective, suffer from inefficiencies during inference. They require extensive sampling steps that can be computationally demanding and often reduce the quality of the resulting images [8, 23]. These challenges underscore the need for innovative diffusion model strategies that balance computational efficiency with high-quality performance in super-resolution tasks [27].

Another innovative approach in the realm of super-resolution (SR) is dubbed ”Resshift,” which reimagines the use of diffusion models for enhancing image resolution [38]. Unlike traditional methods, Resshift begins with a diffusion model that utilizes a shorter Markov chain, specifically designed for the transition between high-resolution (HR) and low-resolution (LR) images. Central to its strategy is the use of a carefully designed transition kernel that efficiently shifts the residual information between the HR and LR states in a stepwise manner. This method capitalizes on the initial condition provided by the LR image, diverging from the standard practice of starting from a Gaussian noise distribution, to iteratively recover the HR image. Such a design notably reduces the number of diffusion steps required, thereby increasing inference efficiency.

Despite the innovative aspects and efficiencies introduced by Resshift, it faces challenges in consistently delivering the highest quality of images. While Resshift significantly reduces the computational load and streamlines the process of image super-resolution, the method still struggles with achieving the pinnacle of image quality. A notable drawback is that images produced by Resshift often have blurred edges, which can detract from the overall clarity and detail that is critical in high-resolution imagery. This issue underscores the need for further refinements and innovations within the approach.

In our advanced approach to super-resolution, we are elevating the capabilities of the Resshift model by incorporating a groundbreaking enhancement that promises to redefine the standards of image quality. Central to our enhanced strategy is the integration of edge-detection algorithms into the reverse diffusion process employed by Denoising Diffusion Probabilistic Models (DDPM). This strategic addition targets the critical weakness of blurred edges in super-resolved images, a common issue with existing methodologies. By embedding precise edge detection into the core of Resshift's process, we aim to dramatically sharpen image details and enhance overall clarity. Here is a concise outline of our method and its potential transformative impacts:

1. **Edge Detection Integration:** We begin by applying an edge detection algorithm, such as the Canny algorithm, to the initial noised low-resolution image. This step creates an edge map that captures essential contours and details. During each denoising step in the reverse diffusion process, we utilize this edge map to guide the enhancement. By calculating the difference between the predicted edges and the actual edges, we apply what we term an ”anti-gradient” to push the image reconstruction towards preserving and accentuating these critical features. This targeted guidance helps in molding the sharper and more defined edges that are pivotal for high-quality high-resolution images.
2. **Optimized Image Quality:** The integration of edge detection is fine-tuned to occur during the latter half of the reverse diffusion sequence, where it can most effectively influence the image's quality without compromising processing efficiency. This adjustment ensures that our approach not only meets but exceeds the existing benchmarks set by conventional super-resolution methods, producing images that are not just higher resolution but also visually superior.

2. Related Work

Super-Resolution with GANs. Generative Adversarial Networks (GANs) have made substantial strides in the field of super-resolution, enhancing the process of upscaling images significantly. The innovative approach of SRGAN (Ledig et al.), which pioneered the use of adversarial networks for super-resolution tasks, demonstrated that GANs could effectively generate high-resolution images from low-resolution inputs, capturing fine details with remarkable precision [12]. Building on this foundational work, Enhanced Super-Resolution GAN (ESRGAN) introduced a more sophisticated architecture and refined training procedures, which led to notable improvements in texture detail and image realism [34].

A key advancement in ESRGAN was the integration of Residual in Residual Dense Block (RRDB) networks, which helped stabilize the training process and improve the quality of the output images [34]. This design allows for deeper network architectures without the risk of vanishing gradients, fostering a more robust learning environment for generating high-quality images.

Moreover, ongoing research in the GAN domain for super-resolution has explored various modifications to the standard GAN architecture and adjustments to loss functions. These enhancements aim to better capture the intricacies of high-resolution image generation. Notably, the adoption of perceptual loss measures has become more prevalent. These measures leverage features extracted by pre-trained deep neural networks, comparing them to assess the similarity between the super-resolved images and their high-resolution counterparts more effectively [10]. This method enhances the perceptual quality of the images, ensuring that they not only look visually appealing but also maintain fidelity to the original high-resolution images.

Super-Resolution with Diffusion Models. Another pivotal method used in super-resolution is the application of diffusion models. Distinct from the adversarial training employed by GANs, diffusion models such as Denoising Diffusion Probabilistic Models (DDPMs) and their variants offer a fundamentally different approach to image generation. These models achieve high-quality image outputs through a process of gradually denoising a noisy signal, which is reversed from how natural diffusion would progress from order to disorder.

Studies like those by Menon et al.[18] and Rombach et al.[21] have pioneered the use of diffusion models conditioned on low-resolution images. This initial approach involves conditioning the diffusion process directly on the degraded input, which guides the denoising steps toward a more accurate high-resolution output. However, such methods can sometimes lead to inconsistencies in image details, particularly when dealing with complex textures or high-frequency information.

Subsequent studies, such as those by Saharia et al. [23], have introduced more sophisticated techniques like masking, where additional contextual information is encoded into the model to guide the reconstruction process more effectively. This approach allows for more precise control over the areas being enhanced, but still struggles with ensuring complete consistency across larger images or sequences of images, leading to potential discrepancies in texture and detail.

More recent innovations in diffusion models for super-resolution involve the integration of cognitive processing capabilities [30]. This framework enhances traditional super-resolution methods by incorporating both image appearance and language understanding to create cognitive embeddings. These embeddings activate prior information from text-to-image diffusion models, significantly enriching the contextual depth and semantic accuracy of the enhanced images. Additionally, the CoSeR framework introduces the ”All-in-Attention” mechanism, which consolidates all conditional information into a single module, ensuring comprehensive and uniform image enhancement.

Overall, while diffusion models have considerably improved the capability to generate high-resolution images from low-resolution inputs, the field continues to face challenges related to consistency, detail preservation, and the avoidance of artifacts. Current techniques, while promising, often struggle to produce high-quality images consistently over long sequences or across diverse image types without some degree of degradation or stagnation in image quality.

Edge Detection methods.Edge detection is a crucial technique in image processing that calculates the image gradient to measure the strength and direction of edges within an image. In edge detection, abrupt changes between adjacent pixel values in an image are identified using classical edge detection operators, which are categorized into first-order and second-order differential operators.These methods are based on gradient change. First-order operators, like Sobel, Prewitt, and Roberts [17, 24, 31] are designed for basic edge detection, while second-order operators, such as Laplace and Canny [3, 33], are optimized for more precise detection in varying conditions. The Canny operator, developed in 1986, is renowned for its robustness to noise and ability to detect subtle edges, making it superior to other methods such as the Roberts operator, which struggles with noise, and the Sobel operator, which enhances edges by integrating Gaussian smoothing.

The next line of methods is based on Gaussian difference. Among these methods are Difference of Gaussian (DoG), FDoG and XDoG algorithms. Difference of Gaussian (DoG) [35] is a technique used for enhancing blurred images, functioning effectively as a band-pass filter by retaining specific frequency information from the original image. The FDoG [11] method enhances DoG by incorporating directional information into the Gaussian convolution, allowing for more accurate edge detection by calculating the Gaussian difference along the edge gradient direction, which effectively suppresses noise and false edges. The XDoG algorithm [6] further refines DoG by introducing a constant that modulates the intensity of Gaussian difference filtering, enabling the transformation of image styles. XDoG also converts the threshold function into a continuous slope, combining Gaussian blur results with Gaussian difference for edge detection that supports more complex styles and improved visual effects.

Multi-scale feature-based edge detection methods address the challenge of detecting edges across varying object sizes and shapes. The gPb algorithm, introduced by Arbeláez et al.[1], combines local cues like brightness, color, and texture with global structural information from spectral clustering for enhanced edge detection. Similarly, the SGD algorithm by Ren et al.[20] leverages sparse coding techniques along with directional gradients and multi-scale pooling to improve the detection and localization of edges in complex images, significantly outperforming traditional methods that rely on manually designed features.

Deep learning has revolutionized edge detection with specialized architectures addressing various challenges. The CASENet by Yu et al.[37] merges multi-label learning with deep semantic edge detection, using ResNet-based connections for improved accuracy in edge classification. RINDNet, introduced by Pu et al.[19], detects multiple edge types simultaneously using separate decoders for reflectance, illumination, normal, and depth edges, enhancing detection through a sophisticated fusion of spatial cues and attention modules. COB, developed by Maninis et al.[16], integrates contour detection with hierarchical segmentation, employing a novel sparse boundary representation for superior edge detection. DeepEdge by Bertasius et al.[2] adopts a top-down multiscale approach, using a bifurcated network structure to enhance contour detection by combining features across scales. HED by Xie and Tu [36] utilizes deeply supervised learning within a fully convolutional network to refine edge detection outputs progressively. Lastly, DexiNed by Soria et al.[28] introduces an up-sampling block in its network to produce finely detailed edge maps, integrating features from multiple encoders for comprehensive edge detection.

3. Preliminaries

3.1. Latent Diffusion Model (LDM)

Rombach et al. introduced Latent Diffusion Models (LDMs)[21], designed to decrease the computational demands of Diffusion Probabilistic Models (DPMs), making it feasible to train them with limited computational resources while maintaining their quality and versatility. The development of LDMs involves a two-stage training process:

1. **Perceptual Image Compression:** During this initial stage, an autoencoder is trained to create a lower-dimensional representational space that is perceptually comparable to the original data space. This step ensures a more efficient encoding of the image data.
2. **Latent Diffusion:** In the subsequent stage, the DPM is trained within this compact latent space instead of the traditional high-dimensional pixel space. This approach not only makes the training process more scalable but also allows for efficient image generation directly from the latent space in a single pass through the network.

LDM operates in the latent space of an autoencoder, typically using architectures like VQ-GAN [5] or VQ-VAE [32], where the encoder ($\mathcal{E}$) and decoder ($\mathcal{D}$) play crucial roles. For an input image Im, the encoder ($\mathcal{E}$) transforms it into a latent tensor $x_0 \in \mathbb{R}^{h \times w \times c}$, initiating the forward diffusion process. During this process, Gaussian noise is iteratively added to x_0 according to the formula:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I),\ t = 1, .., T \quad (1)$$

where $\{\beta_t\}_{t=1}^T$ are hyperparameters governing the noise level, and the process aims to transform x_0 into Gaussian noise x_T. The objective of the Latent Diffusion Model (LDM) is to establish a reverse process, represented as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (2)$$

for $t = T, \dots, 1$.This process effectively reconstructs the original signal x_0 from the noise-distributed x_T. This backward process allows for the reconstruction of the final image from the latent space with a single pass through the decoder: $Im = \mathcal{D}(x_0)$.

After mastering the reverse diffusion process detailed in DDPM [8], a deterministic sampling technique known as DDIM [26] can be employed. This technique is mathematically represented as:

$$\begin{aligned} x_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta^t(x_t)}{\sqrt{\alpha_t}} \right) + \\ \sqrt{1-\alpha_{t-1}}\epsilon_\theta^t(x_t), \quad t = T, \dots, 1, \end{aligned} \quad (3)$$

where $\alpha_t = \prod_{i=1}^t (1-\beta_i)$ and

$$\epsilon_\theta^t(x_t) = \frac{\sqrt{1-\alpha_t}}{\beta_t} x_t + \frac{(1-\beta_t)(1-\alpha_t)}{\beta_t} \mu_\theta(x_t, t). \quad (4)$$

In applications that convert text to images, the SD model orchestrates the diffusion sequence guided by a text prompt τ. In the specific case of DDIM sampling, the update formula modifies to:

$$\begin{aligned} x_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta^t(x_t, \tau)}{\sqrt{\alpha_t}} \right) + \\ \sqrt{1-\alpha_{t-1}}\epsilon_\theta^t(x_t, \tau), \quad t = T, \dots, 1. \end{aligned} \quad (5)$$

Within the framework of LDM, the function $\epsilon_\theta^t(x_t, \tau)$ is realized via a neural network architecture resembling a UNet [22], which integrates convolutional layers along with self and cross-attention mechanisms. Here, x_T denotes the latent code of the original signal x_0, and a specific deterministic process called DDIM inversion [4] is utilized to restore x_T from x_0.

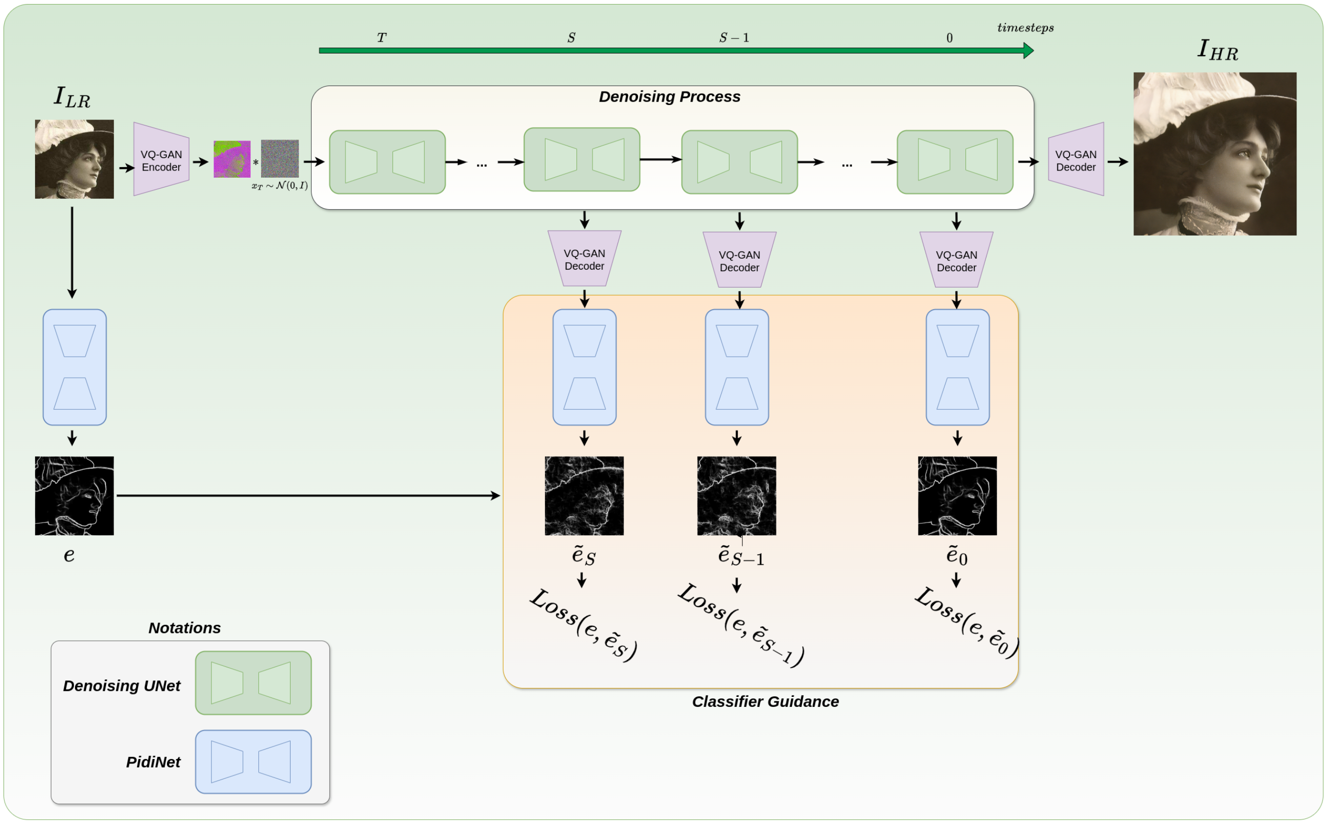

Figure 2. The overall pipline of EdgeSR: For the first step is generated latent represantation of LR image. In reverese process, starting from S step z_s is decoded into spatial image using a VQ-GAN decoder. PiDiNet architecture is then applied to image for edge detection, outputting a predicted edge map $\tilde{e}$. The loss $L(\tilde{e}, e) = ||\tilde{e} - e||^2$ quantifies the discrepancy between the predicted and ground truth edge maps. Following this, the anti-gradient is computed to adjust the latent representation serving as the initial state for the subsequent denoising phase in the diffusion process.

3.2. Classifier Guidance

This innovative method, initially presented in the paper **Diffusion Models Beat GANs on Image Synthesis** [4] involves conditioning a pre-trained diffusion model using the gradients of a classifier. This method harnesses the classifiers trained on noisy images to guide the image synthesis. A classifier $p_\phi(y|x_t, t)$ is trained on noisy images x_t at various diffusion stages. This classifier is designed to recognize specific features or attributes (like edges) in the images. The training involves noisy versions of the images to mimic conditions encountered during the diffusion process.

Once trained the classifier's gradients $\nabla_{x_t} \log p_\phi(\mathrm{y} \mid x_t, t)$ are used to guide the diffusion sampling process. These gradients provide directional cues that steer the noise reduction steps toward enhancing the desired attributes within the images. The integration of classifier gradients modifies the standard diffusion sampling process. Typically, the diffusion model employs a Gaussian distribution to predict subsequent image states, defined as:

$$p_\theta(x_t \mid x_{t+1}) = \mathcal{N}(\mu, \Sigma) \tag{6}$$

The classifier's influence is introduced by adjusting this distribution using the gradient information, effectively shifting the Gaussian distribution's mean:

$$\mu' = \mu + \Sigma \cdot g \tag{7}$$

where g is the gradient of the log probability with respect to x_t, given by:

$$g = \nabla_{x_t} \log p_\phi(y \mid x_t, t) \tag{8}$$

at the mean image $x_t = \mu$.

The modified noise prediction integrates the classifier's gradients to alter the diffusion trajectory:

$$\log\left(p_\theta(x_t \mid x_{t+1}) p_\phi(y \mid x_t)\right) \approx \mathcal{N}(\mu + \Sigma g, \Sigma) \tag{9}$$

This approach redefines the mean of the Gaussian distribution to include the effect of the gradients, aligning the process more closely with the desired attribute enhancements. The influence of the classifier can be adjusted by scaling the gradients. A larger scale increases the attribute specificity, enhancing the desired features more strongly but potentially reducing output diversity. This trade-off is managed by selecting an appropriate scale factor s, where the gradient term becomes $s \cdot \nabla_{x_t} \log p_\phi(y \mid x_t, t)$.

3.3. Pixel Difference Network(PiDiNet)

Pixel Difference Convolution (PDC): Pixel Difference Convolution (PDC) modifies the standard convolution process by utilizing pixel differences instead of direct pixel values, enhancing the model's ability to capture gradient information crucial for edge detection. Unlike traditional convolutions that use pixel values, PDC operates on the differences between pixels within a convolutional kernel's coverage area [29]. This approach is illustrated in the equation:

$$y = f(\nabla \mathbf{x}, \theta) = \sum_{(x_i, x_i') \in \mathcal{P}} w_i \cdot (x_i - x_i'), \tag{10}$$

where (x_i, x_i') are the pixel pairs in the set $\mathcal{P}$, and w_i are the weights assigned to each pixel difference in the convolution kernel. PDC can be categorized based on how pixel pairs are selected:

- **Central PDC (CPDC):** Focuses on differences between centrally located pixel pairs.
- **Angular PDC (APDC):** Utilizes angular relationships among pixels.
- **Radial PDC (RPDC):** Captures radial differences from a central point.

These variants leverage the Extended Local Binary Pattern (ELBP) methodology [14] to encode pixel differences, thereby enhancing the convolution operation's ability to discern textural and edge information.

By embedding PDC within CNN architectures, the network learns to emphasize important gradient features for edge detection, resulting in improved activation responses during training. In a 3x3 APDC configuration, eight pixel pairs are selected and their differences are convolved with kernel weights to produce the output feature map.

Once training is complete, PDC can be converted back to standard convolution to reduce computational overhead. This is achieved by adjusting the kernel weights to directly incorporate pixel differences, maintaining inference efficiency:

$$\begin{aligned} y &= w_1 \cdot (x_1 - x_2) + w_2 \cdot (x_2 - x_3) + w_3 \cdot (x_3 - x_6) + \dots \\ &= (\hat{w}_1 \cdot x_1 + \hat{w}_2 \cdot x_2 + \hat{w}_3 \cdot x_3 + \dots) = \sum \hat{w}_i \cdot x_i \end{aligned} \tag{11}$$

This streamlined explanation retains the key aspects of PDC's role in enhancing edge detection capabilities within PiDiNet, focusing on its innovative approach and practical implementation.

PiDiNet Architecture: PiDiNet features a small model size, high operational efficiency, and the ability to train effectively with limited datasets. Inspired by [7] and [9], the backbone of PiDiNet is a streamlined, depth-wise separable convolutional structure with shortcuts to enhance inference speed and simplify training. This structure is organized into four stages, each containing multiple residual blocks that utilize depth-wise followed by point-wise convolutional layers, optimizing for efficiency and size. The stages are designed to progressively increase in channel capacity, scaling from C to 4×C channels.

To capture detailed edge features, PiDiNet includes a side structure that employs a Compact Dilation Convolution Module (CDCM) to process multi-scale information from each stage. This module is complemented by a Compact Spatial Attention Module (CSAM) to focus on relevant features by reducing background noise. The processed features are then scaled down through a 1×1 convolution and upscaled back to the original dimensions using interpolation and a sigmoid activation to form the edge maps. These maps are combined to produce the final edge detection output through a series of concatenations and convolutions.

The model uses an annotator-robust loss function [15] to train the edge detection framework, ensuring that it adapts based on the clarity of the annotations. The loss for the i-th pixel in the j-th edge map with value p_i^j is defined as follows:

$$l_i^j = \begin{cases} \alpha \cdot \log(1 - p_i^j) & \text{if } y_i = 0 \\ 0 & \text{if } 0 < y_i < \eta \\ \beta \cdot \log(p_i^j) & \text{otherwise} \end{cases} \tag{12}$$

where y_i is the annotated edge probability, η is a threshold for annotator agreement, and α and β adjust the loss based on the balance of positive and negative samples. This nuanced loss function allows PiDiNet to train effectively even with limited data and annotations.

4. Method

In this section, we outline the primary stages of the proposed Image Super Resolution using Edge-Guided Diffusion model (*EdgeSR*) approach. While our technique is versatile and can function as a plug-and-play block for any diffusion-based SISR method, we opted to prioritize implementation with the ResShift [38], which is currently recognized as state-of-the-art technique for SISR.

The key idea of our method is to guide the inference process of a pretrained SISR diffusion model using an edge predictor. This approach encourages the edges of the reconstructed image to align with a reference edge map derived from the low-resolution input image.

Methods	Metrics				
	PSNR↑	**SSIM↑**	**LPIPS↑**	**CLIPIQA**	**ESSIM↑**
ResShift [38]	31.4	0.76	0.069	0.936	0.73
EdgeSR	31.66	0.77	0.085	0.936	0.76

Table 1. Quantitative comparison of ResShift and EdgeSR methods on 40 images taken from test set RealSet65

4.1. EdgeSR:Edge-Guided Image Super-Resolution

Given a low-resolution image I_{LR} and an edge-map e, our goal is to reconstruct a detailed high-resolution image I_{HR}. Figure ? illustrates the proposed edge-guidance described in detail below.

We start with a latent image representation z_T, which is the noised version of the low-resolution image I_{LR}. Typically, the DDPM synthesis involves T consecutive denoising steps $z_t \to z_{t-1}$, comprising the reverse diffusion process, with z_0 representing the final, encoded output image. During each denoising step from $t = T$ to 1, a density score gradient estimation $\epsilon(z_t, t)$ is computed. Based on this gradient and a specific sampler algorithm, the subsequent sample z_{t-1} is determined. To enhance edge fidelity in the diffusion process, at each step-t, an edge predictor is applied to z_t, producing an edge map $\tilde{e}$. The similarity between this predicted edge map and the reference edge map e is then quantified by:

$$L(\tilde{e}, e) = ||\tilde{e} - e||^2 \tag{13}$$

For this purpose, the PiDiNet architecture is utilized, leveraging its capabilities to accurately guide the refinement of edge details by influencing the diffusion steps according to the computed edge loss $L(\tilde{e}, e)$.

Similarly to the external classifier gradient guidance in [4], we evaluate the *anti-gradient* $-\nabla_{z_t} L$. Intuitively, this anti-gradient pushes an intermediate sample z_t to have edges closer to the target. Now we replace the next-step sample prediction z_{t-1} with $\tilde{z}_{t-1} = z_{t-1} - \alpha \cdot \nabla_{z_t} L$, where α controls the edges guidance strength. In practice, the impact of this gradient depends on its relative magnitude to the original model step, hence, we normalize it with:

$$\alpha = \frac{||z_t - z_{t-1}||^2}{||\nabla_{z_t} L||^2} \cdot \beta \tag{14}$$

with β being a constant throughout the reconstruction process. Once being reconstructed with the guidance from the objective L, the model produces a high-resolution image characterized by intricate details and sharp edges.

4.2. Implementation of Edge Guidance

In the diffusion-based image super-resolution process, the initial steps typically do not yield visually meaningful results due to high levels of noise. Therefore, we focus on applying edge guidance at later stages when the image features start to become discernible. Specifically, edge guidance is implemented from step S down to the first step, where S is strategically chosen based on the progress of the denoising process. Commonly, we select $S = 0.5T$, meaning that edge guidance begins at the midpoint of the diffusion process. This selection ensures that the guidance is applied only when the images have sufficiently progressed towards clarity, maximizing the effectiveness of the edge enhancement while avoiding the less coherent stages of the reconstruction.

After determining the optimal start step S for edge guidance, our method progresses with a detailed procedure to enhance edge fidelity in the super-resolved images. The process begins at selected diffusion step S, where the latent representation z_s is decoded into a spatial image format using a VQ-GAN decoder [5]: $\hat{I}_s = \mathcal{D}(z_s)$, where $\hat{I}_s$ represents the decoded image from the latent representation.

Following the decoding step, the PiDiNet architecture [29] is applied to the decoded images to perform edge detection. PiDiNet utilizes a series of convolutional neural networks that have been specifically trained to identify and enhance edges within the image. This network processes the decoded image and outputs a predicted edge map $\tilde{e}$, which represents the detected edges at this particular step of the diffusion process.

After predicting the edge map $\tilde{e}$ from the decoded image $\hat{I}_s$ using PiDiNet, you calculate the loss L to quantify the discrepancy between the predicted edge map $\tilde{e}$ and the ground truth edge map e. This loss provides a measure of the performance of the edge detection at each step of the diffusion process and guides the network in learning to produce more accurate predictions. The loss function for edge detection would be formulated as: $L(\tilde{e}_s, e) = ||\tilde{e}_s - e||^2$, where e is the ground truth edge map e, $\tilde{e}_s$ is the predicted edge map.

For the next step we compute the anti-gradient (negative gradient), which is used to adjust the latent representation z_s directly: $\tilde{z}_{s-1} = z_s - \alpha \nabla_{z_s} L(\tilde{e}_s, e)$, where α, as mentioned above, is a scaling factor that controls the magnitude of the update step, ensuring that the edge guidance does not overpower the natural progression of the diffusion process.

This adjusted latent representation $\tilde{z}_{s-1}$ is then used as the starting point for the next denoising step in the diffusion process.

5. Experiments

5.1. Implementation Details

5.2. Qualitative Results

5.3. Quantitative Results

5.4. Ablation Study

6. Conclusion

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33:898–916, 2011. 4

[2] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection, 2015. 4

[3] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8:679 – 698, 1986. 3

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 4, 5, 7

[5] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 4, 7

[6] Bruce Gooch, Erik Reinhard, and Amy Gooch. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph.*, 23:27–44, 2004. 3

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 4

[9] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. 6

[10] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *Proceedings of the European conference on computer vision*, pages 694–711, 2016. 3

[11] Henry Kang, Seungyong Lee, and Charles Chui. Coherent line drawing. pages 43–50, 2007. 3

[12] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2

[13] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2

[14] Li Liu, Lingjun Zhao, Yunli Long, Gangyao Kuang, and Paul Fieguth. Extended local binary patterns for texture classification. *Image and Vision Computing*, 30:86–99, 2012. 6

[15] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. 2016. 6

[16] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. *Convolutional Oriented Boundaries*, page 580–596. Springer International Publishing, 2016. 4

[17] D. Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, 207:187–217, 1980. 3

[18] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models, 2020. 3

[19] Mengyang Pu, Yaping Huang, Qingji Guan, and Haibin Ling. Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth, 2021. 4

[20] X. Ren and L. Bo. Discriminatively trained sparse code gradients for contour detection. *Advances in Neural Information Processing Systems*, 1:584–592, 2012. 4

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 3, 4

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 4

[23] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021. 2, 3

[24] G.T. Shrivakshan and Chandramouli Chandrasekar. A comparison of various edge detection techniques used in image processing. *International Journal of Computer Science Issues*, 9:269–276, 2012. 3

[25] Amanjot Singh and Jagroop Singh. Super resolution applications in modern digital image processing. *International Journal of Computer Applications*, 150:6–8, 2016. 2

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 4

[27] Yang Song, Chenlin Meng, and Stefano Ermon. Fast and flexible diffusion model training with denoising diffusion implicit models. *arXiv preprint arXiv:2102.05379*, 2021. 2

[28] Xavier Soria, Edgar Riba, and Angel D. Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection, 2020. 4

[29] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection, 2021. 6, 7

[30] Haoze Sun, Wenbo Li, Jianzhuang Liu, Haoyu Chen, Renjing Pei, Xueyi Zou, Youliang Yan, and Yujiu Yang. Coser: Bridging image and language for cognitive super-resolution. *arXiv preprint arXiv:2311.16512*, 2023. 3

[31] Rui Sun, Tao Lei, Qi Chen, Zexuan Wang, Xiaogang Du, Weiqiang Zhao, and Asoke Nandi. Survey of image edge detection. *Frontiers in Signal Processing*, 2:826967, 2022. 3

[32] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 4

[33] Xin Wang. Laplacian operator-based edge detectors. *IEEE transactions on pattern analysis and machine intelligence*, 29:886–90, 2007. 3

[34] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018. 3

[35] Holger Winnemoeller, Sven Olsen, and Bruce Gooch. Realtime video abstraction. *ACM Trans. Graph.*, 25:1221–1226, 2006. 3

[36] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. 2015. 4

[37] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1761–1770, 2017. 4

[38] Jianyi Wang Zongsheng Yue and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 6, 7