Improvement in MetaF2N

Raj Agrawal (210809)

Project Overview

In the MetaF2N project, I trained The model on the aligned FFHQ and DF2K multiscale image datasets. I employed the first-order MAML (Model-Agnostic Meta-Learning) approach to minimize several loss functions, including:

- **LPIPS (Learned Perceptual Image Patch Similarity)**: This loss function measures the perceptual similarity between the output and the ground truth (GT) images during training.
- **Fidelity loss ($\mathcal{L}_1$)**: This function computes the absolute difference between the output and the GT images.

Motivation

While my model effectively minimizes losses comparing the output with the GT images, none of the loss functions assess the quality of the output images themselves. This observation motivated me to explore loss functions that evaluate the quality of the output images. Recognizing that real-world images often exhibit similarities between nearby patches, I sought a method to enhance the realism of my output images. I discovered a paper that utilizes a loss function called SSL (Self-Similarity Loss), which measures the similarity between adjacent patches in an image.

New Idea

1. Image Self-Similarity

Natural images often display repetitive patterns, a phenomenon known as image self-similarity. This property has been leveraged to enhance image restoration performance. I adopted the Exponential Euclidean distance to quantify self-similarity. For two patches I_p and I_q centered at pixels μ_p and μ_q in an image $I \in R^{H\times W\times C}$, I compute the squared Euclidean distance as follows:

$$d^2(I_p, I_q) = \frac{1}{C(2f+1)^2}\sum_{i=1}^{C}\sum_{j=-f}^{f}(\mu_i^{(p+j)} - \mu_i^{(q+j)})^2, \tag{1}$$

where f denotes the patch radius, H, W, and C are the height, width, and channel number of the image, respectively (with $C = 3$ for RGB images). The similarity $S(I_p, I_q)$ is defined as:

$$S(I_p, I_q) = e^{-\frac{d^2(I_p, I_q)}{h}}, \tag{2}$$

where $h > 0$ is a scaling factor. The similarity values range from 0 to 1, indicating that as the Euclidean distance $d^2(I_p, I_q)$ approaches 0, the similarity $S(I_p, I_q)$ approaches 1, suggesting high similarity between the two patches.

2. Mask Generation

Using the self-similarity measure defined above, we can evaluate the similarity of a given patch with all other patches in the image, thereby constructing a Self-Similarity Graph (SSG). However, calculating self-similarity for each patch is computationally intensive due to the size of the SSG, which would be $H^2 \times W^2$. Given that the challenges of Real-ISR (Image Super-Resolution) are primarily found in areas with edges and textures rather than in smooth regions, we create a mask to identify edge/texture pixels, limiting the SSG calculation to these areas. To generate this mask, we first compute an edge map $E \in R^{H\times W}$ by applying the Laplacian operator L to the ground truth (GT) image $I_{HR} \in R^{H\times W\times C}$:

$$E = L \otimes I_{HR}. \tag{3}$$

we then obtain a binary mask $M \in R^{H\times W}$ by thresholding E:

$$M_{i,j} = \begin{cases} 0, & \text{if } E_{i,j} \le t \\ 1, & \text{if } E_{i,j} > t \end{cases} \tag{4}$$

where t is a threshold, set empirically to 20 to retain most true edge pixels while filtering out smooth and trivial features. The mask M is computed offline to avoid repetitive calculations in each iteration. Next, we normalize $S(I_p, I_q)$ as follows:

$$\bar{S}(I_p, I_q) = \frac{1}{\epsilon}S(I_p, I_q), \tag{5}$$

where $\epsilon = \sum_{q\in I_{Ks}} S(I_p, I_q)$ serves as the normalization factor.
For each edge pixel in the mask, we locate the corresponding pixels in the GT image and ISR image, defining a search area centered at them. A local sliding window is used to compute the similarity between the patch centered at the central pixel and patches in the search area. The values of $\bar{S}(I_p, I_q)$ collectively form the SSG of image I, representing the inherent structural similarity distribution within the image.

Loss Function

1. Loss Function-Original

For training MetaF2N, I calculated the outer loop loss $\mathcal{L}^{T_i}$, which is composed of fidelity loss ($\mathcal{L}_1$), LPIPS loss, and GAN loss. Specifically, the fidelity loss is

$$\mathcal{L}_1 = \|I_{SR} - I\|_1, \tag{6}$$

and the LPIPS loss is defined by

$$\mathcal{L}_{\text{LPIPS}} = \|\phi(I_{SR}) - \phi(I)\|_2, \tag{7}$$

where ϕ is the pre-trained AlexNet feature extractor for calculating LPIPS.
The adversarial loss follows the setting of Real-ESRGAN , which is defined by

$$\mathcal{L}_{\text{adv}} = -E[\log(D(I_{SR}))], \tag{8}$$

where the discriminator D is iteratively trained along with the SR network, i.e.,

$$\mathcal{L}_D = -E[\log(D(I)) - \log(1 - D(I_{SR}))]. \tag{9}$$

To improve the numerical stability and avoid gradient exploding/vanishing, I constrain the MaskNet f_m via a regularization term, which is defined by

$$\mathcal{L}_{\text{reg}} = \|m - 1\|_2. \tag{10}$$

In summary, the learning objective of my MetaF2N (i.e., the outer loop loss) is

$$\mathcal{L}^{T_i} = \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_{\text{LPIPS}} + \lambda_3\mathcal{L}_{\text{adv}} + \lambda_4\mathcal{L}_{\text{reg}}, \tag{11}$$

where the hyperparameters λ_1, λ_2, λ_3, and λ_4 are empirically set to 1, 0.5, 0.1, and 0.002, respectively.

2. Loss Function-Updated

To calculate the SSL loss between the output image and the Ground Truth image, we use the Kullback-Leibler (K-L) divergence between the normalized self-similarity graphs $\bar{S}_{HR}$ (for the high-resolution Ground Truth) and $\bar{S}_{SR}$ (for the super-resolved output). The K-L divergence provides a similarity measure, which is equal to zero when both distributions are identical. To regularize this loss, we also add an L_1 regularizer. Thus, the loss $\mathcal{L}_{SSL}$ is defined as:

$$\mathcal{L}_{SSL} = D_{KL}(\bar{S}_{HR} \,||\, \bar{S}_{SR}) + \lambda\|\bar{S}_{SR} - \bar{S}_{HR}\|_1$$

where:

- $D_{KL}(\bar{S}_{HR} \,||\, \bar{S}_{SR})$ denotes the K-L divergence between $\bar{S}_{HR}$ and $\bar{S}_{SR}$,
- $\|\bar{S}_{SR} - \bar{S}_{HR}\|_1$ represents the L_1 norm of $\bar{S}_{SR}$ - $\bar{S}_{HR}$ for regularization,
- λ is the regularization coefficient that balances the contribution of the L_1 regularizer.

This formulation ensures that the loss function captures similarity while controlling the regularization effect.
The total loss is expressed as:

$$\mathcal{L}_{total} = \mathcal{L}^{T_i} + \beta\mathcal{L}_{SSL} \tag{12}$$

$$\mathcal{L}_{total} = \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_{\text{LPIPS}} + \lambda_3\mathcal{L}_{\text{adv}} + \lambda_4\mathcal{L}_{\text{reg}} + \beta(D_{KL}(\bar{S}_{HR} \,||\, \bar{S}_{SR}) + \lambda\mathcal{L}_1) \tag{13}$$

Now combining terms and rewriting the equation we get:

$$\mathcal{L}_{total} = \lambda_2\mathcal{L}_{\text{LPIPS}} + \lambda_3\mathcal{L}_{\text{adv}} + \lambda_4\mathcal{L}_{\text{reg}} + \lambda_5\mathcal{L}_1 + \beta(D_{KL}(\bar{S}_{HR} \,||\, \bar{S}_{SR})) \tag{14}$$

where $\mathcal{L}^{T_i}$ represents the original loss function, and β is weighting factors for the additional losses.

Advantage

The newly introduced loss function incorporates the similarity within the output image itself, providing an additional assessment of the quality of the generated image. By minimizing this total loss, the model learns to produce outputs that are not only closer to the ground truth but also have enhanced internal consistency, especially in regions with similar nearby patches. This approach makes the model more accurate and robust, leading to more realistic, visually coherent output images.

References

[1] Du Chen, Zhengqiang Zhang, Jie Liang, and Lei Zhang. ***SSL**: A Self-similarity Loss for Improving Generative Image Super-resolution.* Available at `https://arxiv.org/pdf/2408.05713`.

[2] Zhicun Yin, Ming Liu, Xiaoming Li, Hui Yang,Longan Xiao, Wangmeng Zuo. ***MetaF2N**: Blind Image Super-Resolution by Learning Efficient Model Adaptation from Faces.* Available at `https://arxiv.org/pdf/2309.08113`.