Lab2
Super Resolution

313510164　陳緯亭　電子碩一

October 7, 2024

Contents

1 Screenshot

1.1 task-1

PSNR: 23.47 dB

```
Epoch 16  Train Loss: 0.0051 | Val Loss: 0.0054 | PSNR: 22.79 dB
Model saved
Epoch 17  Train Loss: 0.0050 | Val Loss: 0.0051 | PSNR: 23.03 dB
Model saved
Epoch 18  Train Loss: 0.0049 | Val Loss: 0.0050 | PSNR: 23.09 dB
Model saved
Epoch 19  Train Loss: 0.0048 | Val Loss: 0.0049 | PSNR: 23.14 dB
Model saved
Epoch 20  Train Loss: 0.0047 | Val Loss: 0.0048 | PSNR: 23.24 dB
Model saved
Epoch 21  Train Loss: 0.0046 | Val Loss: 0.0047 | PSNR: 23.34 dB
Model saved
Epoch 22  Train Loss: 0.0045 | Val Loss: 0.0047 | PSNR: 23.40 dB
Model saved
Epoch 23  Train Loss: 0.0045 | Val Loss: 0.0046 | PSNR: 23.43 dB
Model saved
Epoch 24  Train Loss: 0.0045 | Val Loss: 0.0046 | PSNR: 23.50 dB
Model saved
Epoch 25  Train Loss: 0.0044 | Val Loss: 0.0046 | PSNR: 23.51 dB
```

Fig. 1: Traning result

Fig. 2: Testing result

1.2 task-2

PSNR: 24.01 dB

```
Epoch 91/100, PSNR: 23.89 dB, Loss: 0.0357
New best model saved with PSNR: 23.89 dB
Epoch 92/100, PSNR: 23.89 dB, Loss: 0.0355
New best model saved with PSNR: 23.90 dB
Epoch 93/100, PSNR: 23.90 dB, Loss: 0.0354
New best model saved with PSNR: 23.90 dB
Epoch 94/100, PSNR: 23.90 dB, Loss: 0.0354
New best model saved with PSNR: 23.90 dB
Epoch 95/100, PSNR: 23.90 dB, Loss: 0.0354
New best model saved with PSNR: 23.90 dB
Epoch 96/100, PSNR: 23.90 dB, Loss: 0.0354
Epoch 97/100, PSNR: 23.90 dB, Loss: 0.0354
New best model saved with PSNR: 23.91 dB
Epoch 98/100, PSNR: 23.91 dB, Loss: 0.0354
New best model saved with PSNR: 23.91 dB
Epoch 99/100, PSNR: 23.91 dB, Loss: 0.0353
New best model saved with PSNR: 23.91 dB
Epoch 100/100, PSNR: 23.91 dB, Loss: 0.0353
```

Fig. 3: Traning result

Fig. 4: Testing result

2 In task-2

2.1 What model I choose?

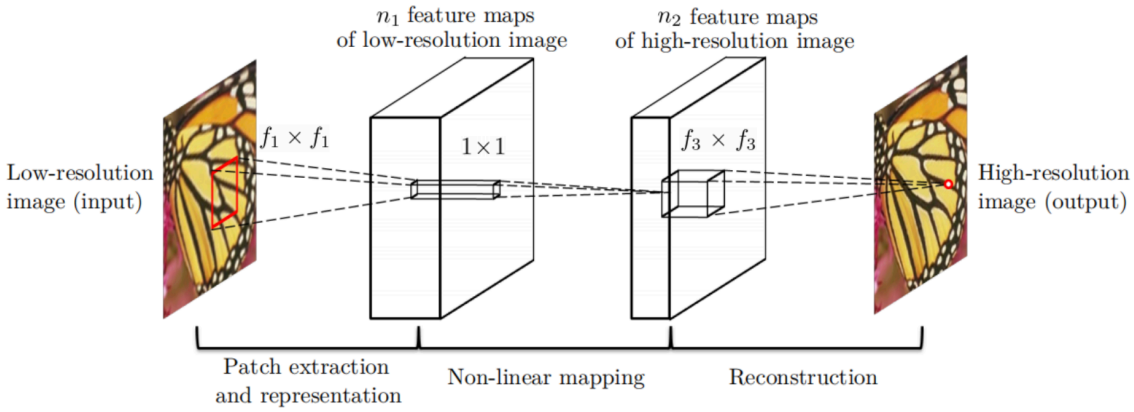


Fig. 5: SRCNN

```
SRCNN(
  (conv1): Conv2d(3, 64, kernel_size=(9, 9), stride=(1, 1), padding=(4, 4))
  (conv2): Conv2d(64, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
  (conv3): Conv2d(32, 3, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
  (relu): ReLU()
  (upsample): Upsample(scale_factor=4.0, mode='bicubic')
)
```

Fig. 6: Task2 SRCNN Model

For Task 2, I've selected the SRCNN model, which features a simple yet effective architecture. It consists of three convolutional layers for feature extraction, followed by a bicubic interpolation layer to handle upsampling.

- **First Convolutional Layer**
  The input is a 3-channel RGB image. This layer uses a 9x9 kernel to output 64 feature maps. The large kernel size helps capture more global context from the input image.

- **Second Convolutional Layer**
  This layer uses a 5x5 convolutional kernel and reduces the number of feature maps to 32, further refining the feature extraction.

- **Third Convolutional Layer**
  his layer reduces the 32 feature maps back to 3 channels, matching the input image's

channel dimension.

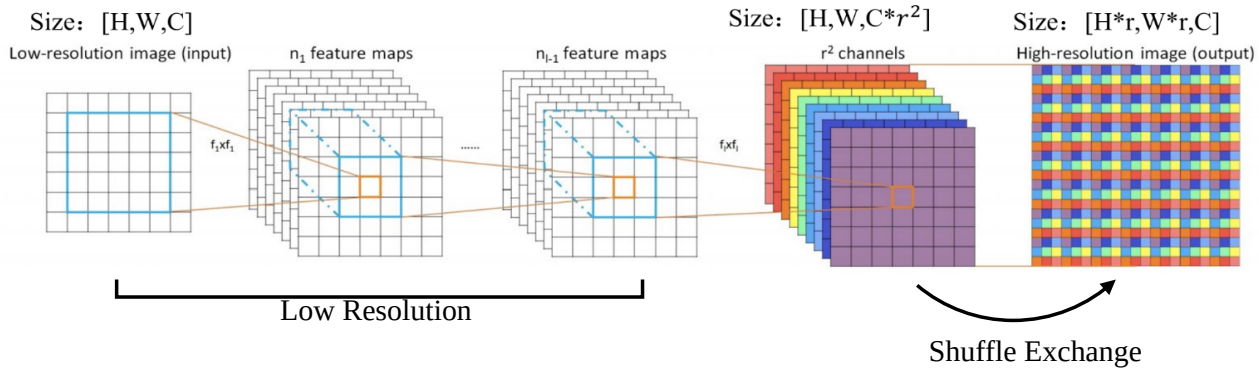
- **Upsampling Layer**
  The upsampling layer scales the image up by a factor of 4 using bicubic interpolation. This step is crucial in super-resolution tasks to output higher-resolution images.

- **Charbonnier loss function** We have abandoned both L2 and L1 loss functions as they tend to produce overly smooth images, lacking the perceptual realism needed for visually convincing results. Instead, we opted for the Charbonnier loss function, which is a more stable and robust alternative.

2.2 The advantage of chosen model

- Lightweight architecture.
- Superior performance compared to state-of-the-art methods.
- High robustness.
- Achieves fast speed for practical on-line usage even on a CPU.

3 Explain what is PixelShuffle



Input: $(*, C_{in}, H_{in}, W_{in})$: * is zero or more batch dimension
Output: $(*, C_{out}, H_{out}, W_{out})$

$$C_{out} = C_{in} \div upscale\_factor^2$$
$$H_{out} = H_{in} \times upscale\_factor$$
$$W_{out} = W_{in} \times upscale\_factor$$

Called ”Sub-Pixel Convolutional Neural Network”

3.1 Functions

Given a low-resolution input image of size $H \times W$, apply a sub-pixel operation with a stride of $\frac{1}{r}$ to upscale it to a high-resolution image of size $rH \times rW$. This involves reshaping the feature map from $[*, C \times r^2, H, W]$ to $[*, C, H \times r, W \times r]$.

3.2 Advantages

1. Effectively mitigates the checkerboard artifacts commonly produced by transposed convolution operations.

2. Demonstrates superior performance compared to conventional upsampling techniques.
   In standard transposed convolution, there are typically many regions filled with zeros, which can negatively impact the output. Pixel shuffle, on the other hand, reconstructs the high-resolution image by rearranging sub-pixels through a sub-pixel convolution.

3. Optimized for image super-resolution tasks.

4. Each pixel in the feature map corresponds to sub-pixels in the new feature map, enabling resolution enhancement.

3.3 Pros

1. **High Accuracy:** By accounting for spatial relationships, this method enables more precise recovery of fine image details, resulting in higher fidelity reconstructions.

2. **Efficient Computation:** The neural network's reorganization reduces the computational load, enabling faster upsampling with lower resource consumption.

3. **Versatility:** This approach is suitable for a wide range of image upsampling tasks, including natural images, medical imaging, remote sensing images, and more.

4 PSNR

4.1 Explain what is PSNR.

The full name is Peak Signal-to-Noise Ratio (PSNR), which is primarily used to calculate the difference between two images based on the concept of mean square error (MSE).

$$MSE = \frac{1}{mn}\Sigma_{i=0}^{m-1}\Sigma_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

I: represent the original image
K the compressed image
i and j: the position of each pixel in the image
m and n: the width and height of the image.
In simple terms, MSE calculates the difference between each pixel in the images, sums these differences, and then averages them.

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \end{aligned}$$

MAX : the maximum value of the image signal.
In simple terms, if each sample point of the image is 8 bits, the maximum value would be 255, and so on.

$$PSNR \propto \frac{1}{MSE}$$

4.2 Discuss why is not the only metric used for evaluating super-resolution.

- **Limitations:** Each model evaluation method has its own limitations.
- **Different Aspects of Quality:** Image evaluation metrics encompass a wide range of aspects, and different criteria exist from various perspectives. Each evaluation standard has its own strengths and weaknesses
- **Task-Specific Requirements:** Ideal images are generally difficult to obtain, making it important to use different evaluation methods.

4.3 Give some other metrics that provide different perspectives on image quality.

- **SSIM (structural similarity)**: To compare images in terms of brightness, contrast, and structure using statistical metrics like average, standard deviation and cross-covariance.

$$SSIM(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma}$$

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

- **LPIPS (Learned perceptual image patch similarity)** It is more aligned with human perception compared to PSNR and SSIM.

$$d(x, x_0) = \Sigma_l \frac{1}{H_l W_l} \Sigma_{h,w} ||w_l \odot (\hat{y}^l_{hw} - \hat{y}^l_{0hw}||_2^2$$

l : Index for the layer in the neural network.
w_l: A learned weight for layer l indicating its importance in the final similarity score.
$H_l \times W_l$: The spatial dimensions of the feature map at layer l.
$\hat{y}^l_{hw}, \hat{y}^l_{0hw}$: The normalized feature vectors at spatial position (h, w) in layer l for two input image x and x_0
$\odot$: Element-wise multiplication.
$||\cdot||_2^2$: The squared L2 norm (Euclidean distance) between the normalized feature vectors at each spatial location.

- **IS (Inception Score)**

$$IS(G) = exp\left(\frac{1}{N}\Sigma_{i=1}^{N} D_{KL}(p(y|x^{(i)})||\hat{p}(y))\right)$$

Using Inception model and KL divergence(Kullback-Leibler divergence), measures how one probability distribution diverges from a second expected probability distribution.

- **FID (Frechet Inception Distance)**

$$FID = ||\mu_r - \mu_g||^2 + T_r\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right)$$

μ_r: Mean feature vector of real images
μ_g: Mean feature vector of generated images
Σ_r : Covariance matrix of features from real images
Σ_g : Covariance matrix of features from generated images

Optimizing IS is crucial because it primarily assesses the clarity and diversity of generated images while entirely overlooking the influence of real data. Moreover, IS has inherent limitations. In contrast, FID emphasizes the relationship between generated images and real images. This algorithm is computed using the Inception model, with the final fully connected layer, which is used for classification, removed.

- **KID**
  KID provides an unbiased estimate using a cubic kernel, which aligns more consistently with human perception

- **Recall(Sensitivity or true positive rat)**

$$Recall = \frac{TP}{TP + FN}$$

- **Precision**

$$Recall = \frac{TP}{TP + FP}$$

TP (True Positive): The number of FTU (Functional Tissue Units) pixels correctly classified as FTU.
FP (False Positive): The number of background pixels incorrectly classified as FTU (usually due to misalignment).
FN (False Negative): The number of FTU pixels incorrectly classified as background.
TN (True Negative): The number of background pixels correctly classified as background.

		Ground truth	
		FTU (1)	Background (0)
Prediction	FTU (1)	TP	FP
	Background (0)	FN	TN

- **MSE（Mean Square Error）**
- **MAE（Mean Absolute Error）**
- **MAE（Mean Absolute Error）**
- **Entropy**

5 Improvement

Anything I do to improve the quality of the output photos.

- I use data augmentation to increase the number of training samples, which helps the model generalize better. For instance, I rotate, flip, and crop the images.
- Add the Gradient Clipping technique to prevent exploding gradients during training.

6 Challenges I faced

- Focusing solely on improving PSNR may not always result in visually pleasing images.
- SRResNet can take a long time to converge. Therefore, I use T4 GPU to speed up the training process.
- There are many architectures to choose from in task 2, and selecting the right one can be challenging.

7 References

1. [電腦視覺] 圖像衡量指標 PSNR、SSIM 介紹

2. 图像评价常用指标 (PSNR、SSIM、LPIPS 、IS、FID、Precision、Recall)

3. Understanding Evaluation Metrics in Medical Image Segmentation

4. 超分辨率模型开源推荐

5. 電腦視覺-超解析度-論文回顧