GNR 638: Image Super-Resolution

Aditya Anand (21D070007) & Shounak Das (21D070068)

1 Introduction

In this Kaggle competition project, we tackle image super-resolution on a custom dataset of gaming images, where each low-resolution image has a corresponding high-resolution ground truth. The goal is to enhance details, textures, and visual fidelity using a deep learning model, specifically the Enhanced Deep Super-Resolution (EDSR) model, a convolutional neural network tailored for super-resolution tasks.

2 Model Details

The Enhanced Deep Super-Resolution (EDSR) [1] model is a convolutional neural network (CNN) designed for single-image super-resolution (SISR). It builds on residual learning and efficient upsampling to reconstruct high-resolution images from low-resolution inputs. Here, we provide a technical breakdown of its three components—head, body, and tail—aligned with a class-based implementation such as `EDSR`.

Note: We implemented the architecture from scratch and trained it on the dataset from scratch, without using any pretrained weights.

2.1 Head: Feature Extraction

The head processes the low-resolution image $I_{LR} \in R^{H\times W\times 3}$ using a 2D convolution with a 3x3 kernel and padding of 1, extracting initial features:

$$F_0 = W_{head} * I_{LR} + b_{head}$$

with $W_{head} \in R^{256\times 3\times 3\times 3}$ and $F_0 \in R^{H\times W\times 256}$. In code, this is typically implemented as: `torch.nn.Conv2d(in_channels=3, out_channels=256, kernel_size=3, padding=1)`.

2.2 Body: Residual Learning Core

The body consists of 32 residual blocks (ResBlocks) that refine features through residual learning, which mitigates vanishing gradients.

Each ResBlock performs:

1. **First Convolution:**

$$F_1 = \text{ReLU}(W_1 * F_{in} + b_1)$$

2. **Second Convolution:**

$$F_2 = W_2 * F_1 + b_2$$

3. **Skip Connection:**

$$F_{out} = F_{in} + F_2$$

Here, F_{in} and F_{out} have 256 channels, with kernels of size 3x3. The body is usually implemented as a `ModuleList` of 32 such blocks.

2.3 Tail: Upscaling and Reconstruction

The tail upscales the feature maps to the target high-resolution size, here a 4x scale factor, using PixelShuffle.

- **Upscaling Block:** For a scale factor $r = 2$, an initial convolution increases channels by 4, followed by:

  $$F_{\text{up}} = \text{PixelShuffle}(2)(W_{up} * F_{body} + b_{up})$$

  For 4x upscaling, this block is applied twice.

- **Final Convolution:**

  $$I_{SR} = W_{tail} * F_{\text{up-final}} + b_{tail}$$

  where $I_{SR} \in R^{4H \times 4W \times 3}$.

The PixelShuffle technique effectively rearranges a tensor of shape $H \times W \times (C \cdot r^2)$ into $rH \times rW \times C$, reducing artifacts compared to traditional upsampling methods.

This condensed explanation outlines the EDSR's three main parts while preserving the key mathematical and operational details for a class-based implementation.

3 Training Details

3.1 Training Configuration

Training is conducted with:

- Batch size: 32
- Epochs: 10

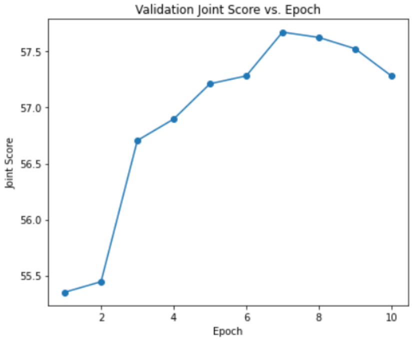

Figure 1: Joint validation score ($40 \times \text{SSIM} + \text{PSNR}$) versus epoch number. The model achieving the highest score was selected for generating on test images.

- Learning rate: 5×10^{-5}
- Optimizer: Adam
- Loss function: L1 Loss
- Scheduler: ReduceLROnPlateau, patience of 5

4 Evaluation

To determine the best-performing model, we adopted a joint metric defined as $40 \times \text{SSIM} + \text{PSNR}$. Throughout the training process, which spanned 10 epochs, we tracked this joint validation score for each epoch using the validation set. The model that achieved the highest joint validation score was selected as the best model and subsequently used for inference on the test set.

To visualize the model's performance progression, we plotted the joint validation score ($40 \times \text{SSIM} + \text{PSNR}$) against the epoch number. This plot, shown in Figure 1, illustrates how the score evolved over the training period, highlighting the improvement in image quality.

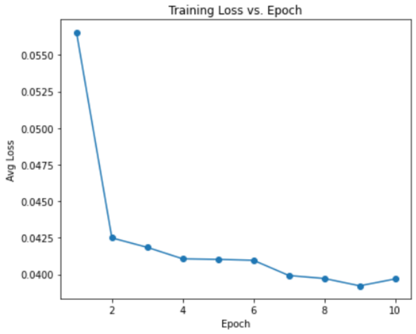

Figure 2: Training loss curve over 10 epochs.

5 Loss Curves and Results

The best model, selected by the joint metric, was saved for inference.

References

[1] Lim, Bee, et al. ”Enhanced deep residual networks for single image super-resolution.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.