Super-Resolution Using Simple CNNs

RENAUDIN Rémi, TERCHANI Loucas

December 22, 2024

Contents

1 Introduction

Image super-resolution (SR) is a critical technique in the field of image processing that aims to produce high-resolution images from their low-resolution counterparts. With the proliferation of electronic devices such as cameras, smartphones, and surveillance systems, the demand for high-quality images has significantly increased. Unfortunately, in many cases, the captured images are constrained by hardware limitations, such as low-quality sensors or unfavorable capture conditions. In this context, SR becomes a valuable tool for enhancing image quality.

Among the most notable applications of SR are:

- Medical imaging, where it allows for precise details to be obtained from diagnostic examinations like MRIs or CT scans.
- Video surveillance, where it improves the readability of images or videos captured by security cameras.
- Multimedia applications, where it is used to convert old content into optimized versions for modern screens.

Historically, interpolation-based methods, such as bilinear and bicubic interpolation, were the most commonly used approaches for SR. These methods are simple to implement and quick to execute, but they suffer from significant limitations. For instance, they tend to produce blurry images and visual artifacts in complex or textured areas. With the emergence of convolutional neural networks (CNNs), the quality of SR images has been revolutionized, as these models can learn rich and complex representations from large training datasets.

The primary objective of this project is to design and train a specific convolutional network, known as SRCNN (Super-Resolution Convolutional Neural Network). The SRCNN is a simple yet powerful model designed to perform SR tasks by learning from pairs of low- and high-resolution images. The main steps of this project include:

1. Data preparation, using several datasets such as Set5, Set14, DIV2K, BSD500, and T91.
2. Designing and training an SRCNN model to enhance the resolution of low-quality images.
3. Evaluating the results using standard metrics like PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index).

This project aims to demonstrate the effectiveness of convolutional networks for SR while exploring the limitations and potential future improvements in this domain.

2 Methodology

2.1 Datasets

For this project, several datasets were used: Set5, Set14, DIV2K, BSD500, and T91. These datasets are widely adopted in the super-resolution community for their varied qualities and different levels of detail.

- **Set5:** This dataset contains five iconic high-resolution images used for quick qualitative evaluations. It includes natural images such as landscapes and urban scenes, making it useful for preliminary performance analysis of models.
- **Set14:** A set of 14 high-resolution images often used as a benchmark to evaluate super-resolution models. It contains a diversity of images ranging from landscapes to indoor scenes.
- **DIV2K:** The DIVerse 2K (DIV2K) is a large dataset comprising 800 training images and 100 validation images. These images feature a wide range of textures, colors, and details. Due to its size and variety, DIV2K is ideal for training complex models like SRCNN.
- **BSD500:** This dataset, often used for segmentation and super-resolution tasks, contains 500 high-resolution images rich in visual details. Its diversity in terms of visual content makes it an excellent choice for final testing.
- **T91:** This dataset consists of 91 high-resolution images and is particularly used to generate training patches for super-resolution tasks.

Low-resolution (LR) images are generated from high-resolution (HR) versions by applying bicubic interpolation and downscaling them by a predefined scale factor (e.g., x2, x3, or x4). For datasets such as BSD500, Set5, and Set14, the images were converted to the YCbCr color space, and only the luminance channel Y was used for training, following standard practices in SR.

2.2 Model Architecture

The SRCNN architecture used in this project is based on a three-step design that transforms a low-resolution image into a high-resolution version through an end-to-end optimized process. The three main steps are:

1. **Feature extraction:** This initial layer uses 64 filters of size 9x9 to capture low-level information present in the low-resolution image. It produces an initial feature map representing the essential details of the image.
2. **Non-linear mapping:** This intermediate step applies 32 filters of size 1x1 to transform the extracted features into a more complex representation suitable for the final reconstruction.
3. **Reconstruction:** The last layer, composed of 1 filter of size 5x5, combines the transformed features to produce the reconstructed high-resolution image.

The model is optimized using the Adam algorithm with an initial learning rate dynamically adjusted through a callback. The loss function used is the Mean Squared Error (MSE), which measures the average difference between the pixels of the reconstructed and original images. This metric promotes overall accuracy while minimizing visual artifacts.

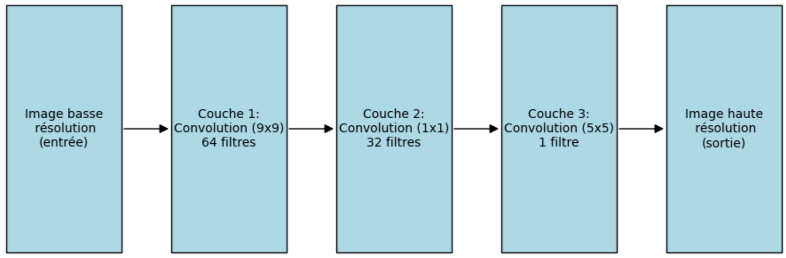

Figure 1: SRCNN model architecture. The data flow illustrates the successive steps of convolution and reconstruction to transform a low-resolution image into a high-resolution image.

2.3 Optimization and Overfitting Prevention

To ensure optimal convergence and prevent overfitting, several techniques were implemented:

- **Learning rate reduction:** A callback monitors the performance on the validation set and reduces the learning rate if the loss does not improve over several consecutive epochs.
- **Cross-validation:** The validation set is regularly used to evaluate performance, allowing hyperparameters to be adjusted and preventing overfitting.

3 Experimentation

3.1 Data Preparation

To validate the performance of the SRCNN model, the data was divided into three sets:

- **Training:** Consists of high- and low-resolution images used to adjust the model weights through backpropagation.
- **Validation:** Monitors performance during training and prevents overfitting by dynamically adjusting hyperparameters.
- **Testing:** Includes images exclusively used for final evaluation, ensuring an unbiased measure of performance.

For each image, a low-resolution version was generated by downscaling the original size with a predefined scaling factor. The low-resolution images were then upscaled using bicubic interpolation to match the size of the high-resolution versions, facilitating their processing by the SRCNN model.

A key aspect of data preparation involves converting images into the YCbCr color space, a technique used to isolate luminance information (Y channel), which is crucial for the perceived quality of reconstructed images.

3.2 Model Training

The SRCNN model was trained over multiple epochs with a batch size of 128 to balance efficiency and memory usage. The initial learning rate was set to 10^{-4} and dynamically adjusted based on performance on the validation set. A total of 1000 epochs was used for the Set5 and Set14 datasets, while BSD500 was trained for 500 epochs. For the T91 and DIV2K datasets, only 50 epochs were performed due to their more complex nature.

Training was conducted using a GPU to accelerate computations, significantly reducing the time required to process image batches. Training and validation losses were recorded after each epoch to monitor convergence and detect potential overfitting.

3.3 Evaluation Metrics

Two primary metrics were used to evaluate the model's performance:

- **PSNR (Peak Signal-to-Noise Ratio):** A quantitative metric measuring the overall quality of the reconstructed image. A higher PSNR indicates more accurate reconstruction.

- **SSIM (Structural Similarity Index):** A perceptual metric that evaluates the structural similarity between the reconstructed and original images, considering luminance, contrast, and structural information.

4 Results

4.1 Performance Metrics

The performance of the SRCNN model was evaluated on multiple datasets using the PSNR and SSIM metrics. Below is a summary of the results obtained:

Dataset	PSNR (dB)	SSIM
Set5	22.08	0.7414
Set14	26.49	0.8255
BSD500	12.05	0.0173
T91	28.00	0.8454
DIV2K	18.60	0.7501

Table 1: PSNR and SSIM metric results for different datasets.

4.2 Visual Comparison

A visual comparison of the reconstructed images was conducted to illustrate the capabilities of the SRCNN model. Examples of reconstructed images from the Set14 dataset, compared to their low-resolution and high-resolution counterparts, are presented in Figure 2.

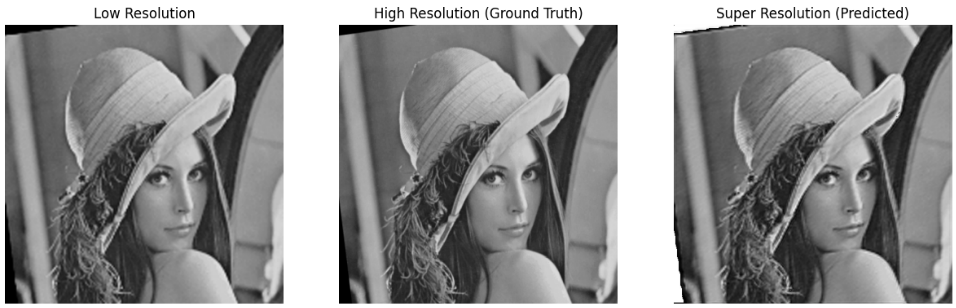

Figure 2: Visual comparison of images from the Set14 dataset: low resolution, SRCNN reconstruction, and high resolution (ground truth).

4.3 Loss Evolution Curves

The evolution of training and validation losses over epochs for the Set5 dataset is shown in Figure 3. These curves demonstrate the progressive convergence of the SRCNN model.

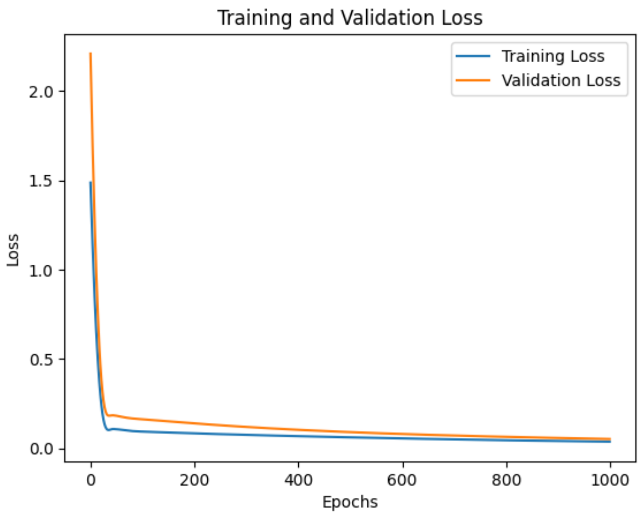

Figure 3: Loss evolution curves for training and validation on the Set5 dataset.

5 Conclusion

The SRCNN architecture demonstrated a significant improvement in the quality of low-resolution images, as confirmed by the obtained PSNR and SSIM values. However, the visual results remain imperfect. The integration of techniques such as dynamic learning rate reduction facilitated better model convergence, limiting overfitting.

Nevertheless, training time remains high, particularly for large datasets like DIV2K, and performance on complex datasets such as BSD500 leaves room for improvement.

For future work, exploring advanced architectures based on residual or attention networks could enhance these results. Additionally, incorporating additional regularization techniques, such as dropout or batch normalization, could strengthen the model's robustness. Finally, testing SRCNN on specific datasets, such as those in medical imaging or video surveillance, could open new avenues for exploration.