A Performance Comparison of SRCNN and ESPCN for Single-Image Super-Resolution on DIV2K and Smartphone Data

Akif Islam
Dept. of Computer Science & Engineering
University of Rajshahi, Bangladesh
Email: iamakifislam@gmail.com

***Abstract*—Single-image super-resolution (SISR) attempts to reconstruct a high-resolution image from a low-resolution input. Early convolutional models such as SRCNN and ESPCN demonstrated that even shallow neural networks can outperform traditional interpolation. In this study, both SRCNN and ESPCN are implemented and trained for two times upscaling using the DIV2K dataset dataset. The networks are evaluated on synthetic low resolution (LR) images generated from DIV2K and on a small set of author's smartphone photographs converted into LR images. Experiments show that both neural models produce visually sharper images than bicubic interpolation most of the time, although SRCNN generalizes slightly better to real smartphone data. ESPCN achieves marginal improvements on DIV2K but is more sensitive to training configuration. Overall, the work highlights the strengths and limitations of shallow CNN-based super-resolution methods on both ideal and real-world data.**

I. Introduction

Single-image super-resolution (SISR) has been widely studied due to its impact on fields such as medical imaging, surveillance, satellite imaging, and consumer photography. The task is particularly challenging because recovering fine high-frequency details from a single low-resolution input is inherently ill-posed. Classic interpolation approaches, including bicubic interpolation, often fail to reconstruct edges and textures faithfully and tend to produce over-smoothed results.

The introduction of deep learning led to significant improvements in SISR. SRCNN demonstrated that even a simple three-layer convolutional architecture could learn an end-to-end mapping from a bicubic-upsampled low-resolution image to its high-resolution counterpart. ESPCN later introduced a more efficient design that performs most computations in the low-resolution domain and reconstructs the final high-resolution output using a pixel-shuffle layer. Although both models are shallow compared to modern super-resolution networks, they remain important educational baselines.

The purpose of this work is to implement SRCNN and ESPCN from scratch, train them under a unified framework, and evaluate their performance on both synthetic and real images. The comparison highlights the differences between models trained on ideal bicubic-downsampled data and their behavior on real smartphone images, which contain noise, compression artifacts, and non-linear camera processing pipelines.

II. Methodology

A. Code Availability

The complete implementation used in this study, including data preprocessing, model training, evaluation scripts, and visualization routines, is publicly available in a Google Colab notebook. The full code can be accessed at the following link:

Click here to view code.

This notebook contains all steps required to reproduce the SRCNN and ESPCN training pipeline, generate low-resolution inputs, compute PSNR and SSIM metrics, and visualize qualitative results on both DIV2K and smartphone images.

B. Datasets

Although the initial plan was to use our own smartphone images for training, validation, and testing, early warm-up experiments showed that real-world photographs contain substantial noise, compression artifacts, and non-linear camera processing effects. For example, though the figure 2 image was taken with smartphone, it is washed out and contains noise. These factors introduced instability and produced inaccurate results during training with phone images.

TABLE I
Dataset distribution used for training, validation, and testing.

Dataset	Train	Validation	Test
DIV2K (High-Resolution)	800	100	100
Smartphone Images	–	–	10
Total	800	100	110

Since the goal of this assignment is to understand and analyze the behavior of super-resolution models under controlled conditions, we decided to use the DIV2K dataset as the primary source of high-quality images for training and validation. The smartphone images were instead retained as a secondary test set in order to assess how the trained models generalize to real-world data.

The experiments therefore rely on the DIV2K dataset, which provides 800 high-resolution images for training, 100 images for validation, and 100 images for testing. During training,

(a) Red Panda

(b) Church

Fig. 1. Examples of visually rich images used for qualitative super-resolution evaluation from the DIV2K dataset.

Fig. 2. Super-resolution comparison for a test data of the smartphone captured photo where the author holding a persian cat at pet hospital

random patches of size 128×128 are extracted from the high-resolution images. Each patch is converted into a low-resolution version through bicubic downsampling by a factor of two, and then upsampled back to high-resolution size using bicubic interpolation to form the SRCNN input. ESPCN directly receives the downsampled low-resolution patch, while the original high-resolution patch serves as the ground truth for both models. For validation and test evaluation, the full-resolution images are used, with only minimal adjustment to ensure that their dimensions remain divisible by the scaling factor. The SRCNN and ESPCN both were trained for 300 epochs.

C. SRCNN Architecture

SRCNN follows the classic three-layer design. The first layer performs feature extraction from the bicubic-upsampled input. The second layer performs non-linear mapping to produce a refined representation, and the final layer reconstructs the high-resolution output. All convolutions operate with stride 1 and adequate padding to preserve spatial size. Despite its simplicity, SRCNN can correct some of the smoothing introduced by bicubic interpolation and reconstruct moderately sharper textures.

D. ESPCN Architecture

ESPCN processes the image entirely in the low-resolution space. It applies two convolutional layers with non-linear activation functions and produces a multi-channel output encoding the high-resolution structure. The final pixel-shuffle (sub-pixel convolution) operation rearranges the feature maps into a high-resolution image. This design reduces computational cost and often produces sharper edges, although it is also more sensitive to training stability and dataset quality.

E. Training Setup

Both networks are trained using mean squared error loss computed between the predicted output and the ground-truth high-resolution patches. The Adam optimizer with a learning

Fig. 3. Super-resolution comparison for a test data of the smartphone captured photo where the author is standing infront of a conference banner

Fig. 4. Super-resolution comparison for a test data of the smartphone captured photo where the author with his friend took a selfie in a railway station

rate of 1×10^{-4} is used throughout all experiments, and training is carried out for 300 epochs with a batch size of 16. During training, the model that achieves the highest PSNR on the validation set is saved and later used for final evaluation. The upscale factor is fixed at $2\times$, since larger scaling factors require deeper and more expressive architectures than those considered in this study. Although we initially experimented with an $8\times$ upscale factor, the results confirmed that SRCNN and ESPCN are not designed to handle such large magnification levels, and their performance degraded significantly under that configuration.

III. Results

A. Validation Performance

The validation PSNR of both networks increases steadily during the early epochs. SRCNN generally converges slightly faster because it operates on bicubic-upsampled inputs, which already resemble the high-resolution images. ESPCN performs most of its computation in the low-resolution space and occasionally exhibits slower improvement. After approximately 100 epochs out of 300 epochs, both models begin to stabilize.

TABLE II
DIV2K Test Set Performance for $2\times$ Super-Resolution.

Method	PSNR (dB)	SSIM
Bicubic	**31.58**	0.9150
SRCNN	31.51	0.9104
ESPCN	31.53	0.9152

B. DIV2K Test Set Results

The trained SRCNN and ESPCN models are evaluated on the 100-image DIV2K test set using full-resolution inputs. Table II reports the average PSNR and SSIM values. All three methods perform within a narrow range, with differences of less than 0.1 dB. Bicubic interpolation provides a strong

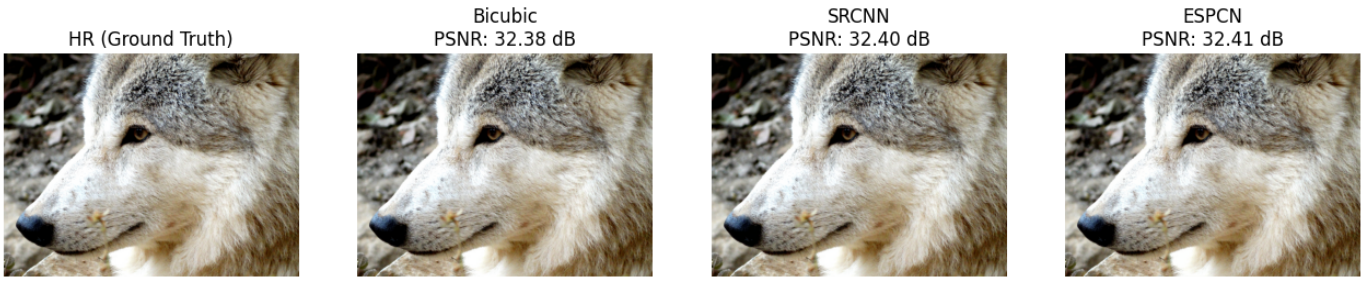

Fig. 5. Super-resolution comparison on DIV2K image `0805.png`, showing HR ground truth, bicubic upsampling, SRCNN output, and ESPCN output along with their PSNR values.

baseline, ESPCN attains the highest SSIM, and SRCNN remains competitive despite its shallow architecture. Because $2\times$ super-resolution leaves limited room for numerical improvement, the gains over bicubic are modest; however, visual inspection still reveals that CNN-based models can better preserve certain textures, as illustrated in Figure 5, where ESPCN performs best overall.

C. Performance on Smartphone Images

To examine how well the trained models generalize beyond the synthetic DIV2K degradation process, the networks were further evaluated on a separate collection of real photographs captured using a smartphone (shown in figure 2,3,4. These images were first converted into low-resolution inputs through bicubic downsampling and were then super-resolved using bicubic interpolation, SRCNN, and ESPCN. The original high-resolution photographs served as the reference for computing PSNR. Unlike the clean DIV2K images, the smartphone photos contain noticeable sensor noise, compression artifacts, and nonlinear processing effects introduced by the camera pipeline. As a result, this test set provides a more realistic but also more challenging evaluation scenario. In all cases of smartphone images, the CNN based model showed better performance than bicubic interpolation result.

TABLE III
AVERAGE PSNR ON THE SMARTPHONE IMAGE SET.

Method	PSNR (dB)
Bicubic	37.95
SRCNN	**38.19**
ESPCN	38.00

IV. DISCUSSION

Although SRCNN and ESPCN are expected to outperform bicubic interpolation, the improvements observed in this experiment remain modest. Across the DIV2K test set, the three methods produce mixed results: in some cases the CNN-based models achieve slightly higher PSNR or SSIM, while in others bicubic interpolation performs marginally better. This inconsistency is surprising, given that both networks were trained for 300 epochs under standard settings, and it suggests that shallow architectures such as SRCNN and ESPCN have limited capacity to deliver meaningful gains. The narrow performance gap also reflects the fact that bicubic downsampling is a relatively simple degradation model, making it easier for interpolation-based methods to approximate the ground truth.

Despite these limitations, the models still demonstrate practical value. Qualitative inspection shows that SRCNN and ESPCN can restore certain textures and edge structures that appear smoother in bicubic outputs, particularly in complex regions such as fur, grass, and fine patterns. Such properties make these models suitable for enhancing old photographs, archival materials, maps, and artistic or cultural heritage images where moderate sharpening can improve perceived clarity.

In contrast to the DIV2K results, the smartphone evaluation set exhibits more consistent behavior. Here, both SRCNN and ESPCN outperform bicubic interpolation in all tested cases. Because real smartphone images contain noise, compression artifacts, and nonlinear camera processing, they differ substantially from the synthetic DIV2K degradation model. Yet, the CNN-based models generalize better than bicubic interpolation, suggesting that their learned priors provide a meaningful advantage when dealing with real-world image imperfections. This indicates that even shallow CNN super-resolution models retain practical usefulness for everyday photographic enhancement, especially in scenarios where interpolation methods fail to reconstruct fine details.