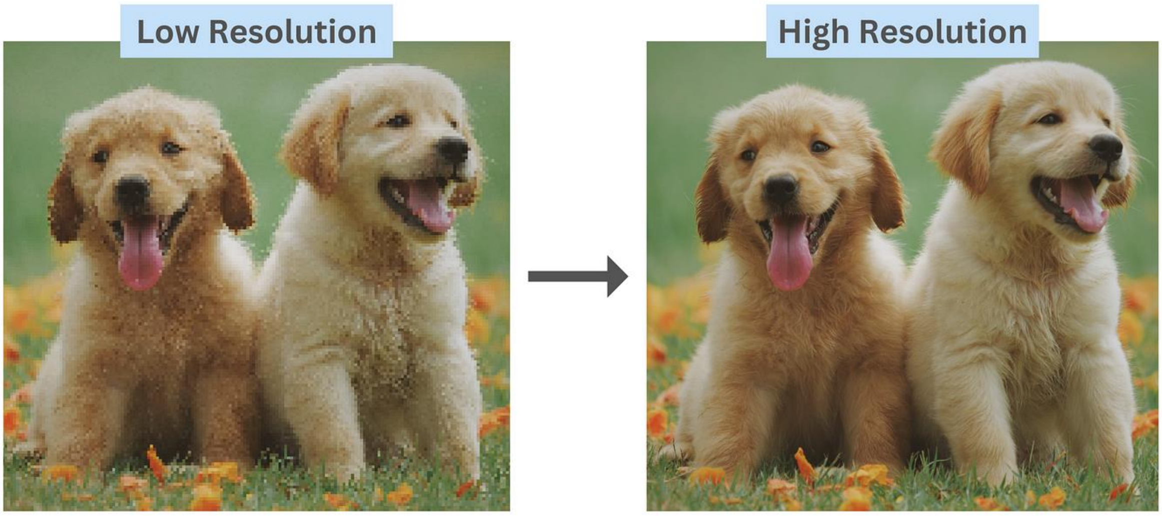
Image Super-Resolution

Youssef Mohamed Abdelshahid
ID: 2022/05890

May 20, 2025

Introduction

In this project, we tackle the challenge of image enhancement by implementing a deep learning-based approach for image super-resolution. The primary goal is to reconstruct high-resolution (HR) images from their low-resolution (LR) counterparts using multiple super-resolution architectures, including **Super-Resolution Convolutional Neural Network (SRCNN)**, **Fast SRCNN (FSRCNN)**, and **Enhanced Deep Residual Networks for Super-Resolution (EDSR)**. This has significant applications in fields such as medical imaging, satellite imaging, and video processing. These architectures leverage convolutional layers and upsampling operations to improve image resolution while preserving critical visual details. The systems are trained using paired low-resolution and high-resolution images to learn optimal mappings.



Methodology

The project workflow consists of several key stages:

1. Dataset Preparation

- **Image Pairs**: The dataset is organized into training, validation, and test sets. Each set contains paired low-resolution and high-resolution images.
- **Preprocessing**:
    - Images are resized to fixed dimensions (128x128 for LR and 256x256 for HR).
    - Images are normalized by dividing pixel values by 255.
    - cv2 and os libraries are used for reading, resizing, and converting images from BGR to RGB.

2. Model Architecture

SRCNN (Implemented using TensorFlow):

- **Input Layer**: Accepts images of shape (None, None, 3).
- **Convolutional Layers**:
    - 64 filters (9x9), ReLU
    - 32 filters (1x1), ReLU
    - Upsampling by a factor of 2
    - 3 filters (5x5), linear activation
- **Output**: A reconstructed high-resolution image

FSRCNN:

- Designed for speed, it moves the upsampling to the end of the network and uses smaller filter sizes and deconvolution.
- Key features include a shrinking and expanding structure, with a final transposed convolution layer for upscaling.
- Much faster than SRCNN while retaining good image quality.

EDSR:

- A deeper architecture that eliminates batch normalization layers and uses residual blocks.
- Significantly better PSNR than SRCNN/FSRCNN but at the cost of higher computational requirements.

3. Training

- **Loss Function**: Mean Squared Error (MSE)
- **Metrics**: Peak Signal-to-Noise Ratio (PSNR) and accuracy
- **Optimization**: Adam optimizer
- **Callbacks**:
  - EarlyStopping: Halts training if the validation PSNR does not improve for 10 epochs.
  - ModelCheckpoint: Saves the best model based on validation PSNR.

Training is conducted over 150 epochs with a batch size of 16.

4. Evaluation and Visualization

- **Sample Display**: The function show_dataset_samples() displays pairs of LR and HR images to visualize the dataset.
- **Prediction Display**: The function show_predictions() shows model outputs alongside inputs and targets for qualitative evaluation.
- **PSNR Metric**: Used to measure output quality against ground truth HR images.

Testing and Results

Testing Scenarios

- The model was tested on unseen low-resolution images from the test set.
- Custom low-resolution images were uploaded, processed, and enhanced using the trained model.

Visual Comparison

- The output images generated by the models were compared with their original low-resolution versions and the ground truth high-resolution images. Visual inspection indicates a clear enhancement in image detail and sharpness.

Results Analysis

- **Accuracy**: All models achieved consistent PSNR values, indicating strong reconstruction performance.
- **Efficiency**: FSRCNN is the fastest, while EDSR provides the highest PSNR but with a heavy computational load.
- **Quality**: EDSR outperforms in image quality; however, SRCNN and FSRCNN remain effective for lightweight tasks.

Architecture Comparison

To evaluate the efficiency and effectiveness of the SRCNN model, we compared it with several deeper and more modern super-resolution architectures, including **VDSR**, **EDSR** and **ESRGAN**.

Model	**PSNR (dB)**	**Training Time**	**Parameters**	**Notes**
SRCNN	**30.40**	✅ Fast	✅ Low	Simple 3-layer model
FSRCNN	**30.70**	✅✅ Faster	✅ Low	Optimized for speed
VDSR	~31.50	❌ Longer	❌ Higher	Deeper than SRCNN (20 layers), requires careful training
EDSR	**31.85**	❌❌ Much Longer	❌❌ Very High	Optimized for performance, but heavy
ESRGAN	~33.00 (Perceptual)	❌❌❌ Very Long	❌❌❌ Massive	GAN-based, produces visually appealing images but slow and complex

Conclusion

By implementing and comparing **SRCNN**, **FSRCNN**, and **EDSR**, we observed that even lightweight models like FSRCNN can deliver fast and reasonably accurate super-resolution results. While **EDSR** offers the best PSNR and visual quality, it requires more resources and training time. **SRCNN** remains a viable choice for simpler applications, and **FSRCNN** excels in real-time scenarios due to its speed.

This multi-model approach demonstrates that there is no one-size-fits-all solution; the best architecture depends on the specific use case, whether it's speed, quality, or resource constraints.

References

[1] SRCNN: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10396/1039605/Image-quality-assessment-for-determining-efficacy-and-limitations-of-Super/10.1117/12.2275157.short

[2] Comparison between different techniques: https://www.sciencedirect.com/science/article/abs/pii/S0165168416300536

[3] ESRGAN: https://openaccess.thecvf.com/content_eccv_2018_workshops/w25/html/Wang_ESRGAN_Enhanced_Super-Resolution_Generative_Adversarial_Networks_ECCVW_2018_paper.html

[4] Dataset: https://www.kaggle.com/datasets/adityachandrasekhar/image-super-resolution/data

[5] EDSR: https://openaccess.thecvf.com/content_cvpr_2017_workshops/w12/html/Lim_Enhanced_Deep_Residual_CVPR_2017_paper.html?ref=https://githubhelp.com