Deep Learning and its Applications to Signal and Image Processing and Analysis
361.2.1120
Final Project

Eitan Spivak

311391866

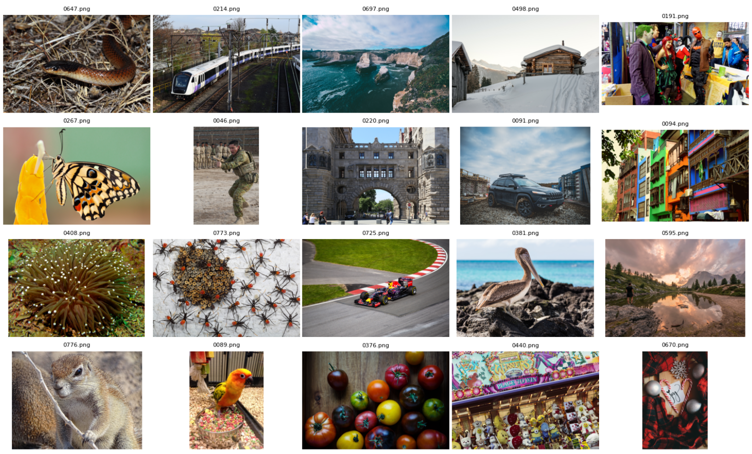
GitHub

Abstract

In this project, I tackled the single-image super-resolution challenge using the DIV2K dataset, where the goal is to reconstruct high-resolution images from their 4× downsampled versions. I implemented and evaluated two models: a standard vanilla SRCNN, and an improved SRCNN with greater depth and explicit bicubic upsampling. Experiments were repeated three times with different random seeds to ensure robustness. The improved SRCNN consistently outperformed the vanilla model, achieving higher average PSNR (27.47 vs. 26.69), SSIM (0.7937 vs. 0.7713), and lower FID (29.13 vs. 34.47). These results demonstrate that architectural enhancements such as deeper layers and learnable upsampling can lead to noticeably better reconstruction quality in super-resolution tasks.

Introduction, Objective, and Data Description

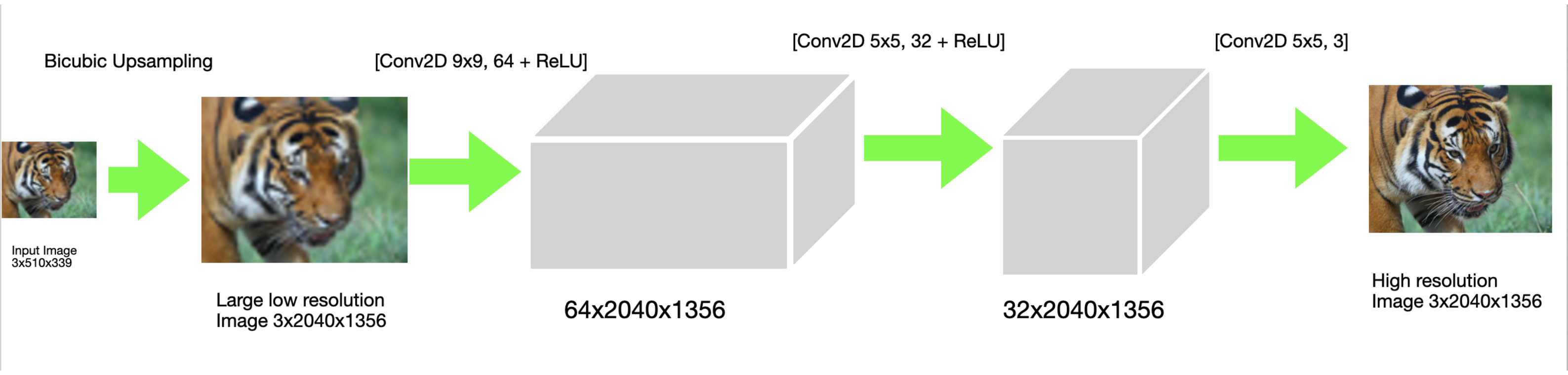
The objective of this project is to enhance low-resolution images by reconstructing high-resolution versions, using the DIV2K dataset as a benchmark. DIV2K provides 800 training and 100 validation image pairs at 2K resolution, with low-res inputs generated via bicubic downsampling (×4). I visualized random examples to verify data quality and consistency. Model performance was evaluated using PSNR and SSIM for similarity to ground truth, and FID to assess perceptual image quality.



Model 1: Vanilla Model

Model Architecture

I implemented the standard SRCNN model for single-image super-resolution. The network consists of three convolutional layers: the first layer extracts features with 64 filters (9×9), the second applies a nonlinear mapping with 32 filters (5×5), and the third reconstructs the high-resolution image with 3 filters (5×5), using ReLU activations after the first two layers.



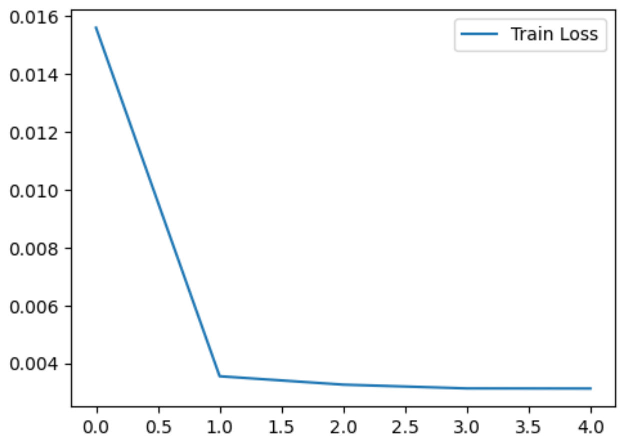
Data Preprocessing

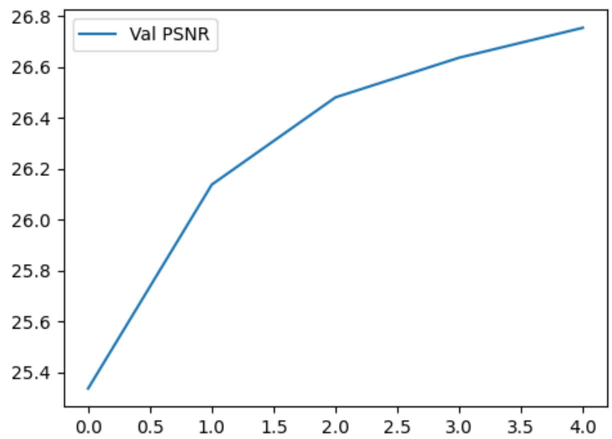
I preprocessed the DIV2K dataset locally by center-cropping or resizing all high-resolution images to a uniform size (2040×1356), ensuring consistent input dimensions. I then created low-resolution images by bicubic downsampling the cropped HR images by a factor of 4, after making their dimensions divisible by 4. These processed images were saved in dedicated folders for streamlined loading in Colab. During training, images were loaded as RGB and normalized to [0, 1]. Random samples were visualized to verify the data pipeline.

Hyperparameters and Loss Function

- Optimizer: Adam
- Learning rate: 1e-4
- Batch size: 4
- Epochs: 5
- Loss function: Mean Squared Error (MSE)

Training Loss Curve

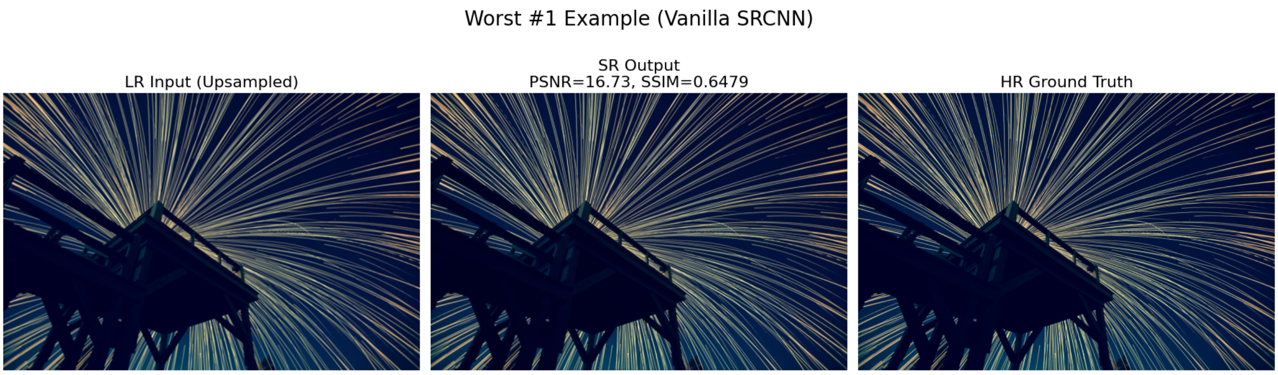




Test Results

Quantitative Evaluation:
The vanilla SRCNN was evaluated on the DIV2K validation set over three runs, achieving an average PSNR of **26.69** (std: 0.08), SSIM of **0.7713** (std: 0.0010), and FID of **34.47** (std: 0.58). These results demonstrate consistent performance across runs and provide a baseline for further improvements.

Qualitative Evaluation:



Model 2: Improved SRCNN

Model Architecture and Modifications
For my improved model, I extended the vanilla SRCNN by increasing the network depth and replacing the standard ReLU activations with PReLU for enhanced nonlinear modeling capacity. The architecture consists of a bicubic upsampling layer (×4), followed by a sequence of convolutional layers:

Conv2D (9×9, 64 filters) + PReLU

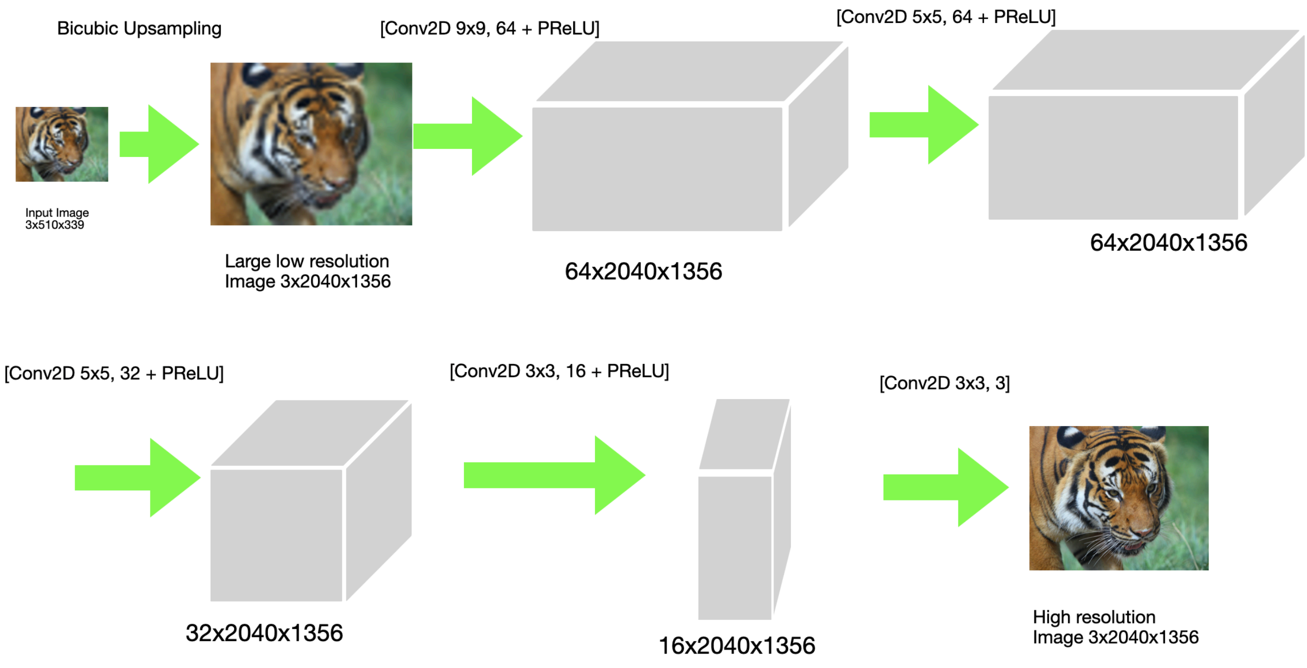
Conv2D (5×5, 64 filters) + PReLU

Conv2D (5×5, 32 filters) + PReLU

Conv2D (3×3, 16 filters) + PReLU

Conv2D (3×3, 3 filters)

Model Structure Diagram

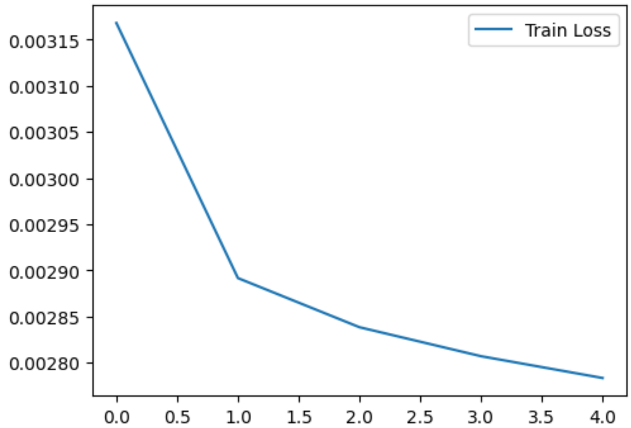


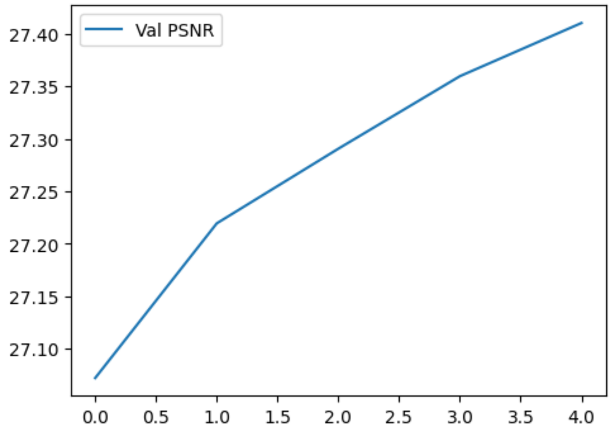
Data Preprocessing
Data preprocessing was identical to Model 1: images were center-cropped or resized to a uniform size, then low-resolution versions were generated via bicubic downsampling (×4). During training, images were normalized to [0, 1] and visualized to verify consistency.

Training Regime, Hyperparameters, and Loss

- Optimizer: Adam
- Learning rate: 1e-4
- Batch size: 4
- Epochs: 5
- Loss function: Mean Squared Error (MSE)

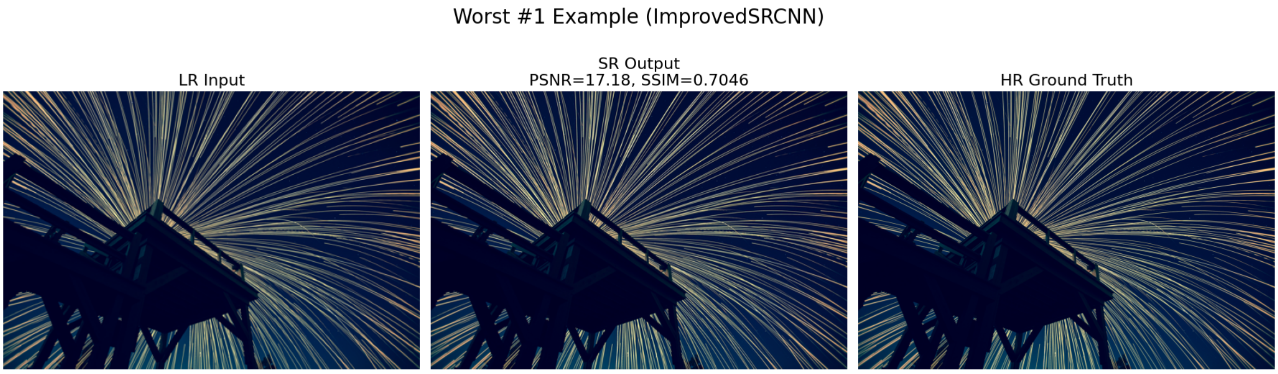
Training Loss Curve





Test Results

The ImprovedSRCNN demonstrated consistently better performance than the vanilla model across three runs, achieving an average PSNR of **27.47** (std: 0.04), SSIM of **0.7937** (std: 0.0009), and FID of **29.13** (std: 0.22). Qualitatively, the improved model produced noticeably sharper and more detailed reconstructions, particularly in regions with complex textures.

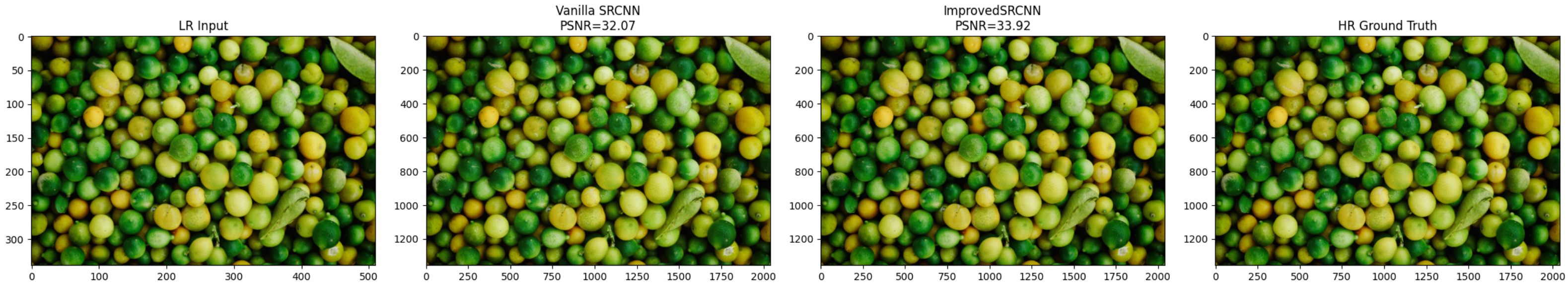


3.6 Results

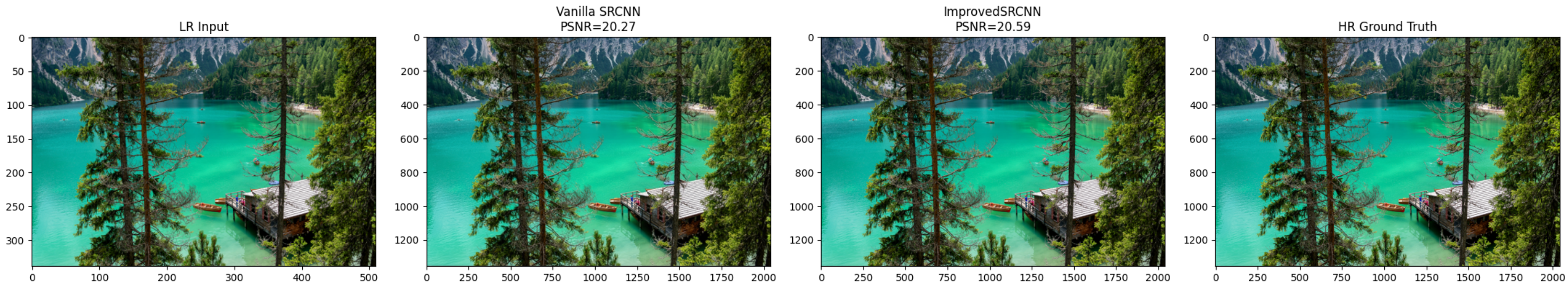
Model	PSNR	SSIM	FID
Vanilla SRCNN	26.69 ± 0.08	0.7713 ± 0.0010	34.47 ± 0.58
Improved SRCNN	27.47 ± 0.04	0.7937 ± 0.0009	29.13 ± 0.22

3.6.1

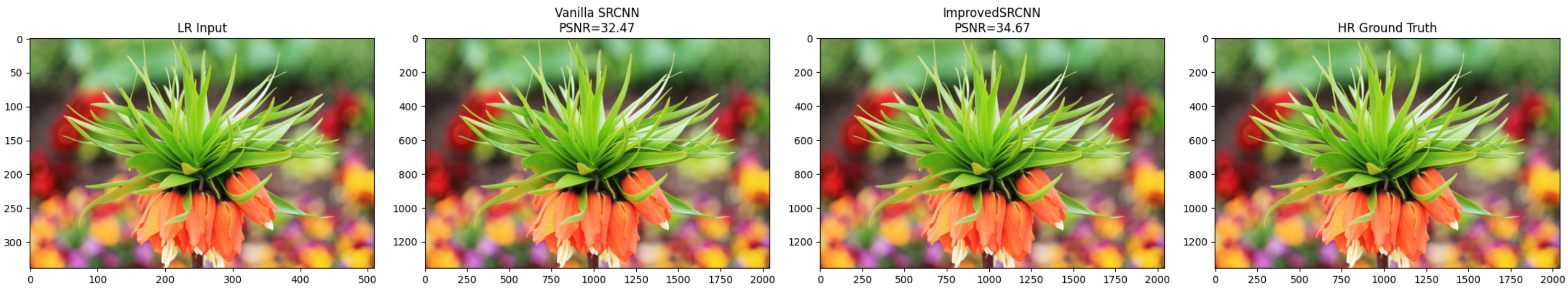
Both Good Example (Validation Index 1)



Both Bad Example (Validation Index 6)



Improved Better Example (Validation Index 2)



There are no examples where Model 1 performed well and Model 2 did not

Deep Learning and its Applications to Signal and Image Processing and Analysis
361.2.1120
Final Project

Eitan Spivak

311391866

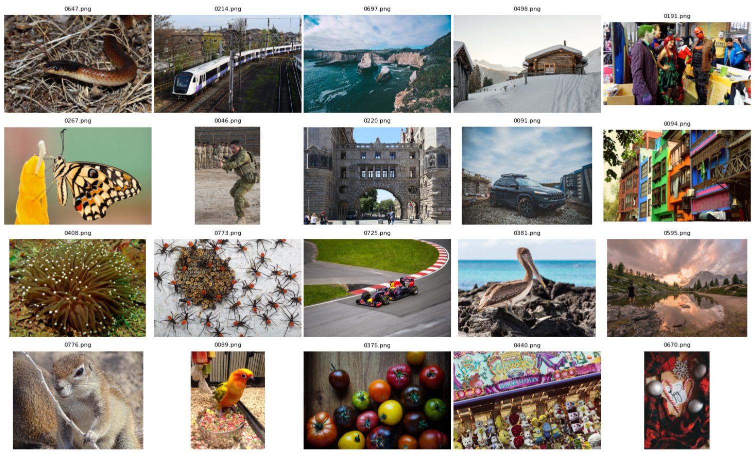
https://github.com/eitanspi/super-resolution-div2k

Abstract

In this project, I tackled the single-image super-resolution challenge using the DIV2K dataset, where the goal is to reconstruct high-resolution images from their 4× downsampled versions. I implemented and evaluated two models: a standard vanilla SRCNN, and an improved SRCNN with greater depth and explicit bicubic upsampling. Experiments were repeated three times with different random seeds to ensure robustness. The improved SRCNN consistently outperformed the vanilla model, achieving higher average PSNR (27.47 vs. 26.69), SSIM (0.7937 vs. 0.7713), and lower FID (29.13 vs. 34.47). These results demonstrate that architectural enhancements such as deeper layers and learnable upsampling can lead to noticeably better reconstruction quality in super-resolution tasks.

Introduction, Objective, and Data Description

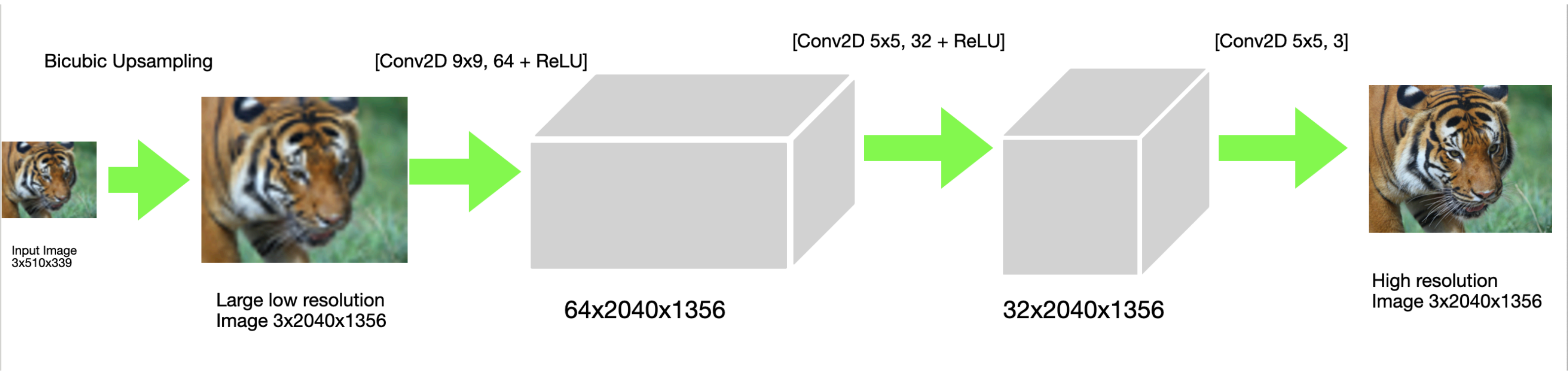
The objective of this project is to enhance low-resolution images by reconstructing high-resolution versions, using the DIV2K dataset as a benchmark. DIV2K provides 800 training and 100 validation image pairs at 2K resolution, with low-res inputs generated via bicubic downsampling (×4). I visualized random examples to verify data quality and consistency. Model performance was evaluated using PSNR and SSIM for similarity to ground truth, and FID to assess perceptual image quality.



Model 1: Vanilla Model

Model Architecture

I implemented the standard SRCNN model for single-image super-resolution. The network consists of three convolutional layers: the first layer extracts features with 64 filters (9×9), the second applies a nonlinear mapping with 32 filters (5×5), and the third reconstructs the high-resolution image with 3 filters (5×5), using ReLU activations after the first two layers.



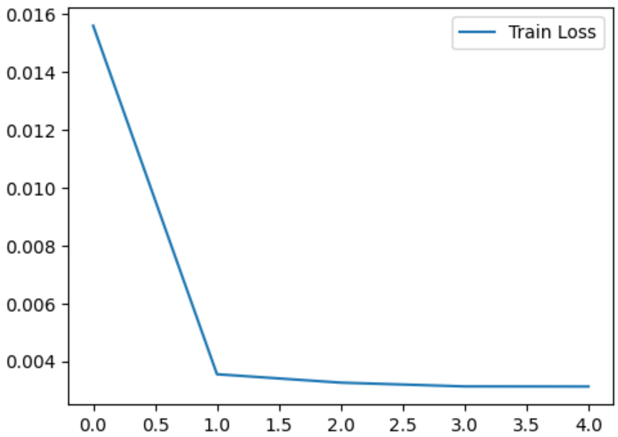
Data Preprocessing

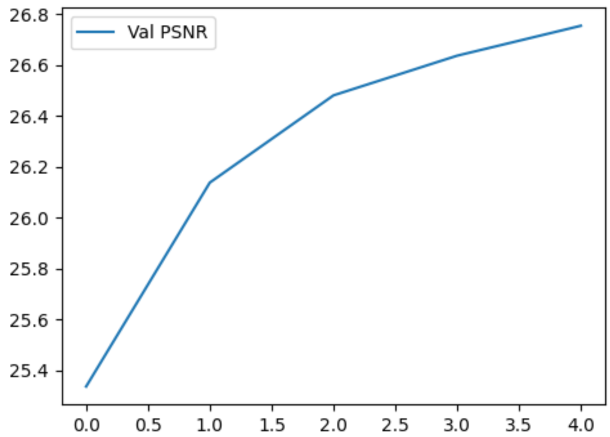
I preprocessed the DIV2K dataset locally by center-cropping or resizing all high-resolution images to a uniform size (2040×1356), ensuring consistent input dimensions. I then created low-resolution images by bicubic downsampling the cropped HR images by a factor of 4, after making their dimensions divisible by 4. These processed images were saved in dedicated folders for streamlined loading in Colab. During training, images were loaded as RGB and normalized to [0, 1]. Random samples were visualized to verify the data pipeline.

Hyperparameters and Loss Function

- Optimizer: Adam
- Learning rate: 1e-4
- Batch size: 4
- Epochs: 5
- Loss function: Mean Squared Error (MSE)

Training Loss Curve

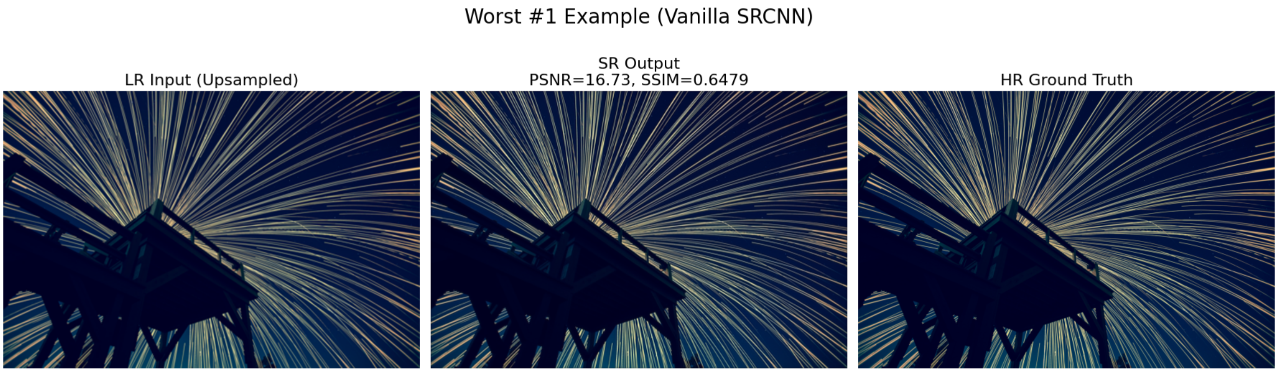




Test Results

Quantitative Evaluation:
The vanilla SRCNN was evaluated on the DIV2K validation set over three runs, achieving an average PSNR of **26.69** (std: 0.08), SSIM of **0.7713** (std: 0.0010), and FID of **34.47** (std: 0.58). These results demonstrate consistent performance across runs and provide a baseline for further improvements.

Qualitative Evaluation:



Model 2: Improved SRCNN

Model Architecture and Modifications

For my improved model, I extended the vanilla SRCNN by increasing the network depth and replacing the standard ReLU activations with PReLU for enhanced nonlinear modeling capacity. The architecture consists of a bicubic upsampling layer (×4), followed by a sequence of convolutional layers:

Conv2D (9×9, 64 filters) + PReLU

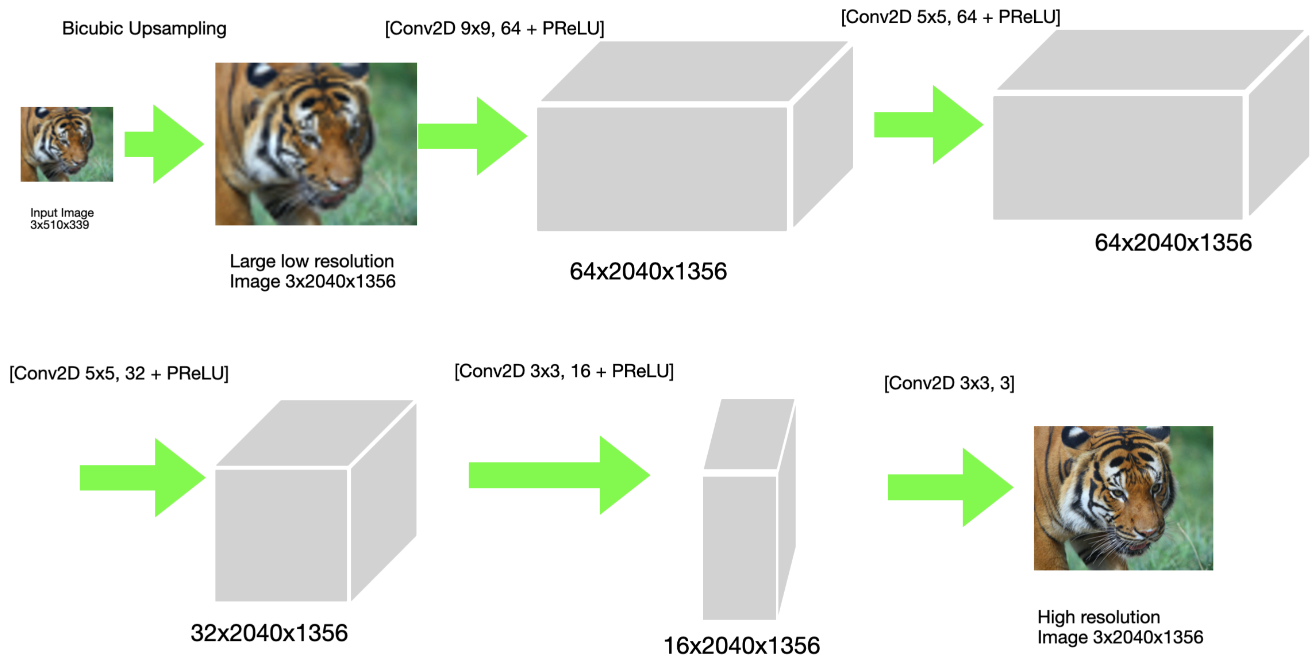
Conv2D (5×5, 64 filters) + PReLU

Conv2D (5×5, 32 filters) + PReLU

Conv2D (3×3, 16 filters) + PReLU

Conv2D (3×3, 3 filters)

Model Structure Diagram

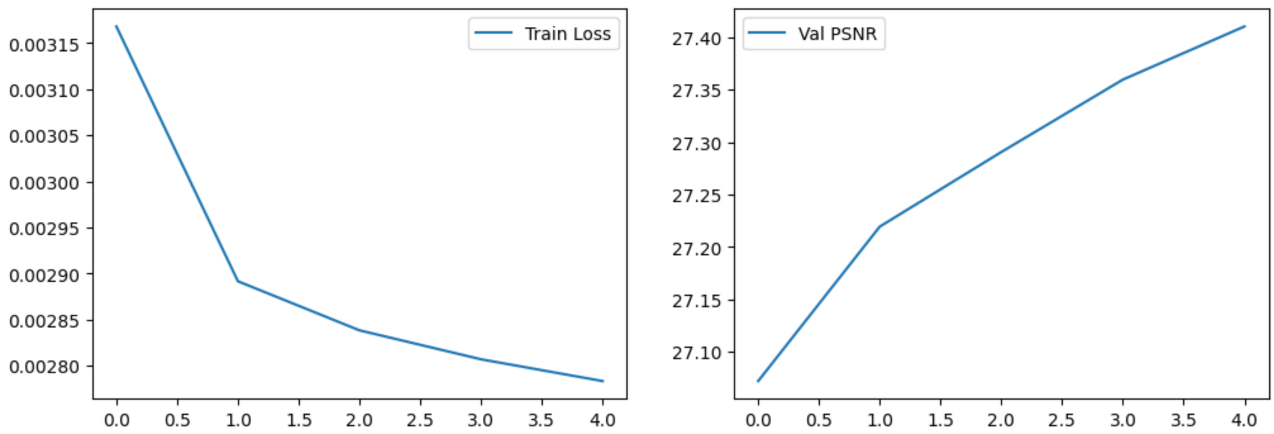


Data Preprocessing
Data preprocessing was identical to Model 1: images were center-cropped or resized to a uniform size, then low-resolution versions were generated via bicubic downsampling (×4). During training, images were normalized to [0, 1] and visualized to verify consistency.

Training Regime, Hyperparameters, and Loss

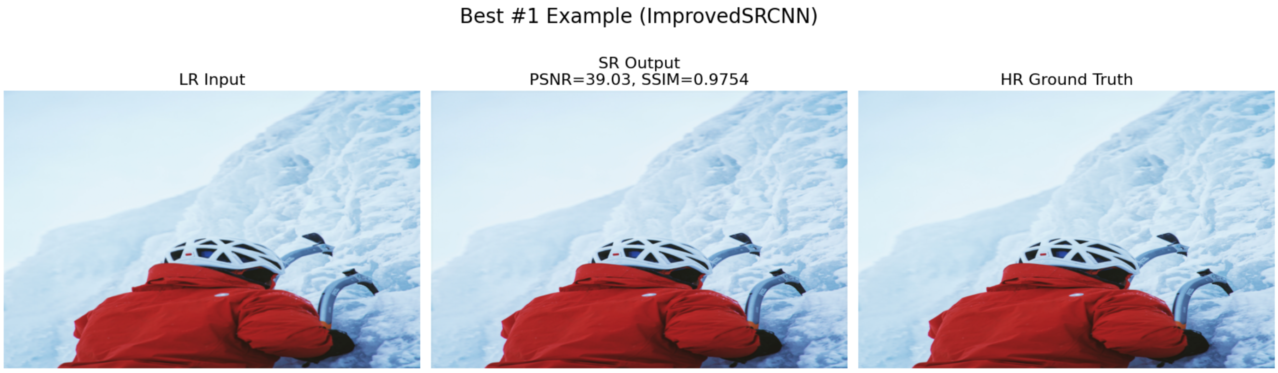
- Optimizer: Adam
- Learning rate: 1e-4
- Batch size: 4
- Epochs: 5
- Loss function: Mean Squared Error (MSE)

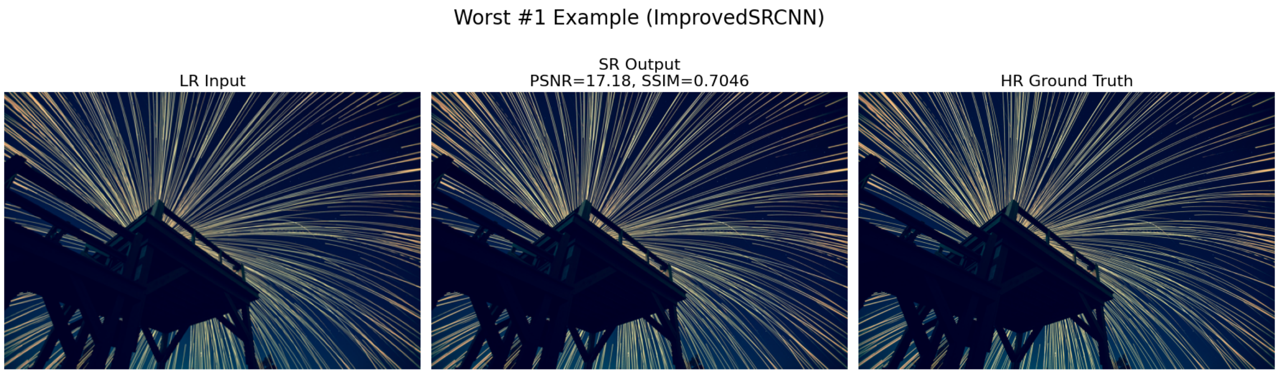
Training Loss Curve



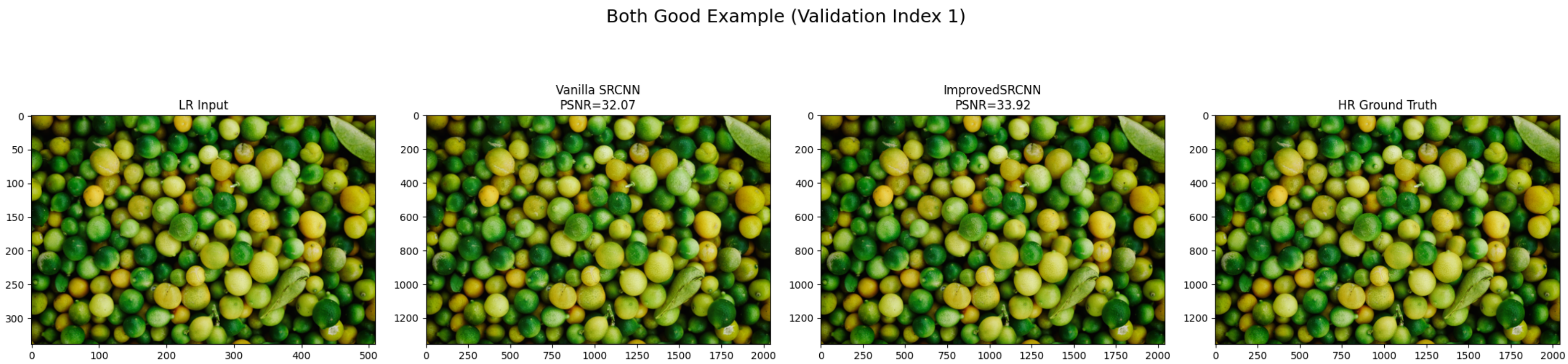
Test Results

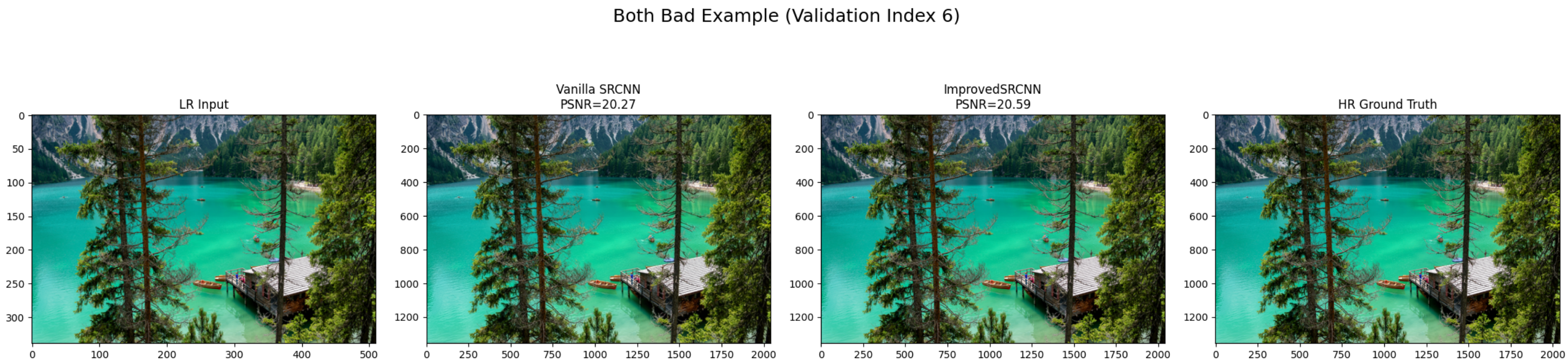
The ImprovedSRCNN demonstrated consistently better performance than the vanilla model across three runs, achieving an average PSNR of **27.47** (std: 0.04), SSIM of **0.7937** (std: 0.0009), and FID of **29.13** (std: 0.22). Qualitatively, the improved model produced noticeably sharper and more detailed reconstructions, particularly in regions with complex textures.

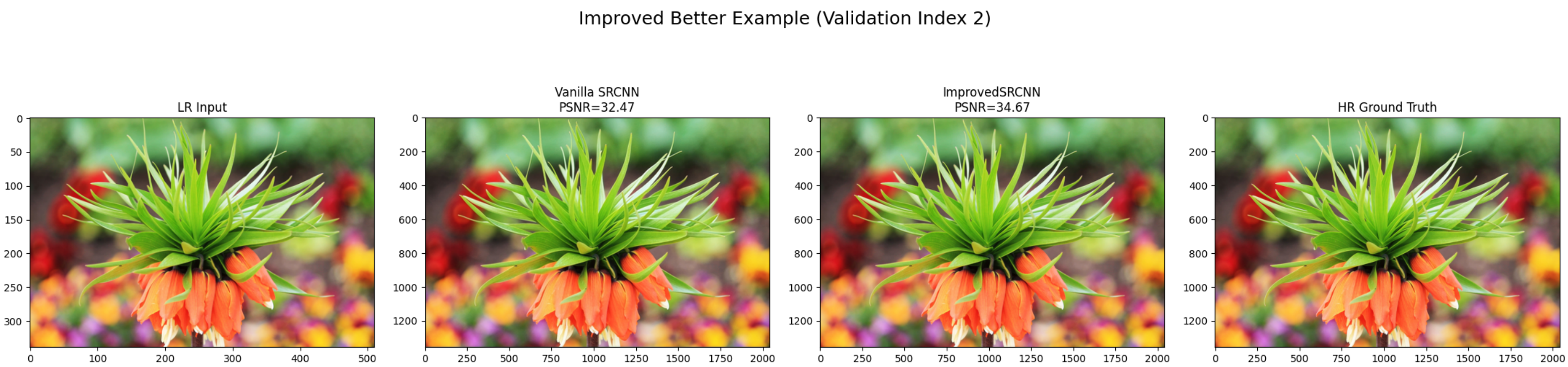




3.6.1







There are no examples where Model 1 performed well and Model 2 did not

3.6.2 Results

Model	PSNR	SSIM	FID
Vanilla SRCNN	26.69 ± 0.08	0.7713 ± 0.0010	34.47 ± 0.58
Improved SRCNN	27.47 ± 0.04	0.7937 ± 0.0009	29.13 ± 0.22

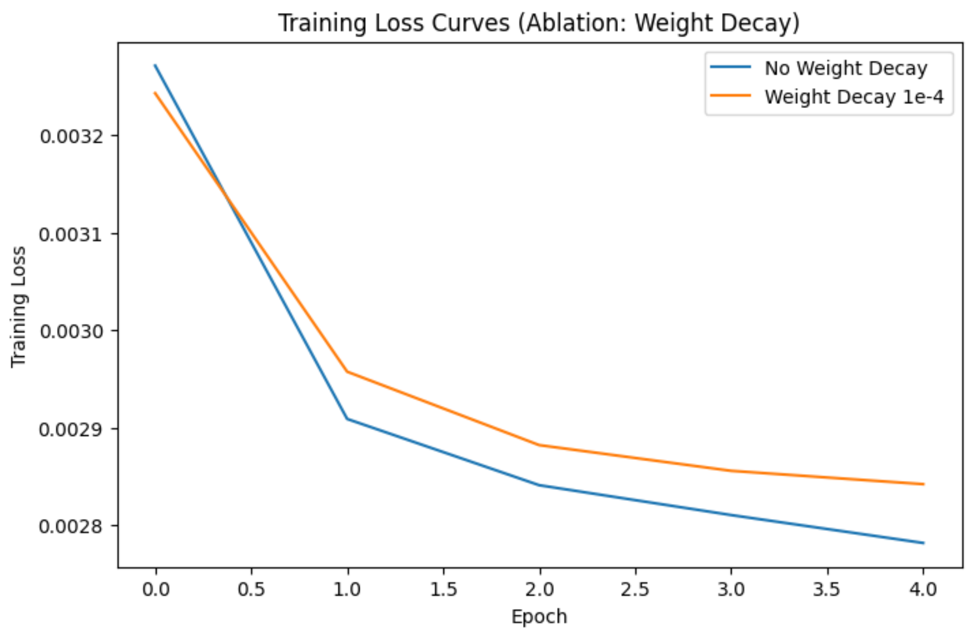
3.7 Ablation Studies

3.7.1 Ablation Setup and Motivation

I evaluated the effect of regularization by training ImprovedSRCNN with and without weight decay (L2 regularization, 1e-4) in the Adam optimizer. This tested whether weight decay could help prevent overfitting or improve generalization.

3.7.2 Training Loss Curves

Both models showed stable convergence, but the no-weight-decay version reached a marginally lower final training loss.

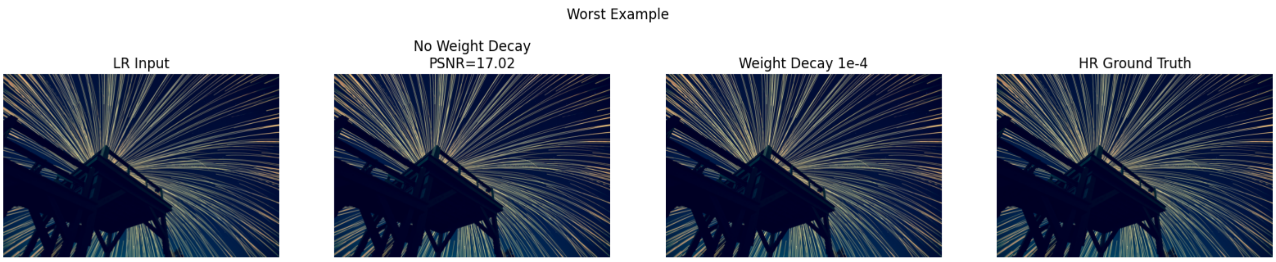


3.7.3 Quantitative Metrics

Setting	Final Val PSNR	Final Train Loss
No Weight Decay	27.40	0.00278
Weight Decay 1e-4	27.27	0.00284

3.7.4 Qualitative Examples

Best and worst-case outputs are shown below, comparing models with and without weight decay. In both easy and challenging cases, the visual results are similar, with no clear qualitative advantage for weight decay. The model without weight decay achieved slightly higher PSNR, but differences in output quality are minor.



3.7.5 Insights

Weight decay at this scale did not benefit the model and slightly reduced performance. The architecture appears robust enough for this task without additional regularization.

3.8 Conclusions and Summary

In this project, I implemented and compared two deep learning models for single-image super-resolution using the DIV2K dataset. The improved SRCNN, which incorporated greater depth and PReLU activations, consistently outperformed the vanilla SRCNN in terms of PSNR, SSIM, and FID. Ablation studies showed that adding moderate weight decay did not improve results, indicating the model's robustness without additional regularization. Overall, the findings demonstrate that architectural enhancements and careful selection of activation functions can meaningfully improve image reconstruction quality in super-resolution tasks.

3.9 References

No external references were used in this report.