Image Super-Resolution

Digital Signal and Image Management Project
University of Milano-Bicocca

Matteo Breganni 869549
Francesco Cavallini 920835

Dataset Preparation

Manually created dataset

- Used at the beginning of the project
- **General dataset**, many different types of natural images (people, animals, structures, nature...)
- Training images: 100
- Test images: 50

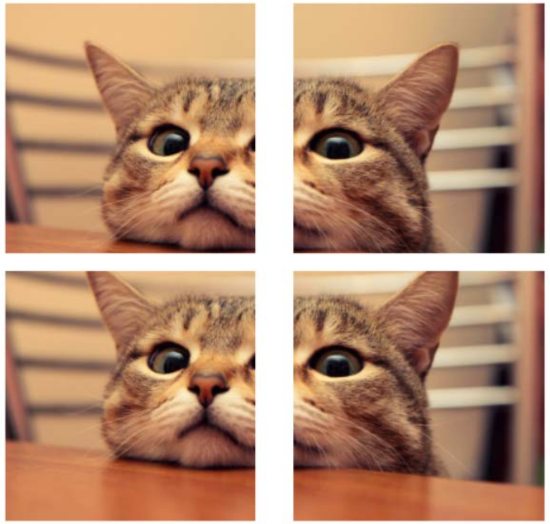
Dataset augmentation

- Original images divided into **256x256 overlapping patches**
- 80/20 train/val split applied
- Training images: 80 → 3752
- Validation images: 20 → 992
- Test images: 50 → 2344
- Images directly saved in colab's runtime to save RAM and have fast access during training

Original image

Size: (1200, 1920, 3)

Split Images



Sizes: (1000, 1000, 3)

Original image

Size: (2160, 3840, 3)

Split Images

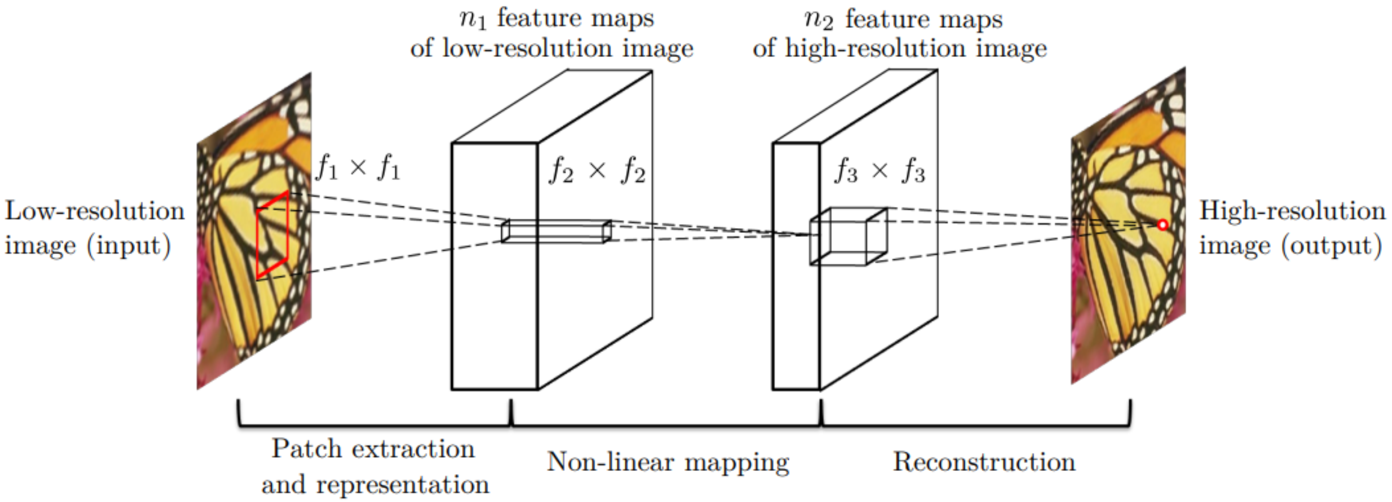
Sizes: (1000, 1000, 3)

(images not part of the dataset)

Super Resolution Convolutional Neural Network (SRCNN)

SRCNN

- **Based on the original 2014 paper** that pioneered the use of CNNs for image super-resolution
- **The input is the low resolution image, upscaled** by the same factor it was downscaled with, to match the output size of the original image (patch).
- 3 channel RGB input works better than YCbCr because of the correlation between channels



- **Input: 33x33x3**
- First layer: 64 filters 9x9 to capture the image's features representations
- Second layer: 32 filters 1x1 to learn non-linear transformations of the features
- Third (and output) layer: 3 filters 5x5 to reconstruct the high-resolution image from the feature maps (output size: 33x33x3, because of padding = 'same')

Evaluation metrics

Peak Signal-to-Noise Ratio (PSNR)

- **Comparing the reconstructed image to the original**
- A **higher PSNR** indicates that the SRCNN-generated image is closer to the original high-resolution image.
- Also computed between the upscaled image using **bicubic interpolation** and the high resolution image, as a **baseline**.

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX\{I\}}{\sqrt{MSE}}\right)$$

Structural Similarity Index Measure (SSIM)

- Used starting with the 4th model
- Calculates the similarity between two images by comparing the luminance (how well the brightness matches between the two images), the contrast (the consistency of pixel intensity variations between the two imaegs) and the strucutral information (compares the structural arrangement of pixels, to make sure that structures and patterns, so textures and edeges, are preserved).
- **The closer to 1, the better**.

First model and issues

- 2x scale factor used

PSNR between LR and HR: 29.85 dB

PSNR between Predicted and HR: 29.70 dB

- Slight quality improvement
- PSNR **worse than the baseline**?

First model and issues

- **Padding='same'** and mantaining input size=output size causes issues around the edges

PSNR between LR and HR: 52.43 dB

PSNR between Predicted and HR: 40.82 dB

- Padding='same' uses 0-padding around the edges, to mantain the spatial dimension
- Introduces **artifacts** around edges and (especially) corners

Solution 1

- **Cut** the affected border, reducing the size of the output

PSNR between LR and HR: 28.78 dB
PSNR between Predicted and HR: 28.97 dB

Solution 2 (second model)

- Use **padding='valid'**
- The model will reduce the size of the output organically (from 33x33 to 21x21)

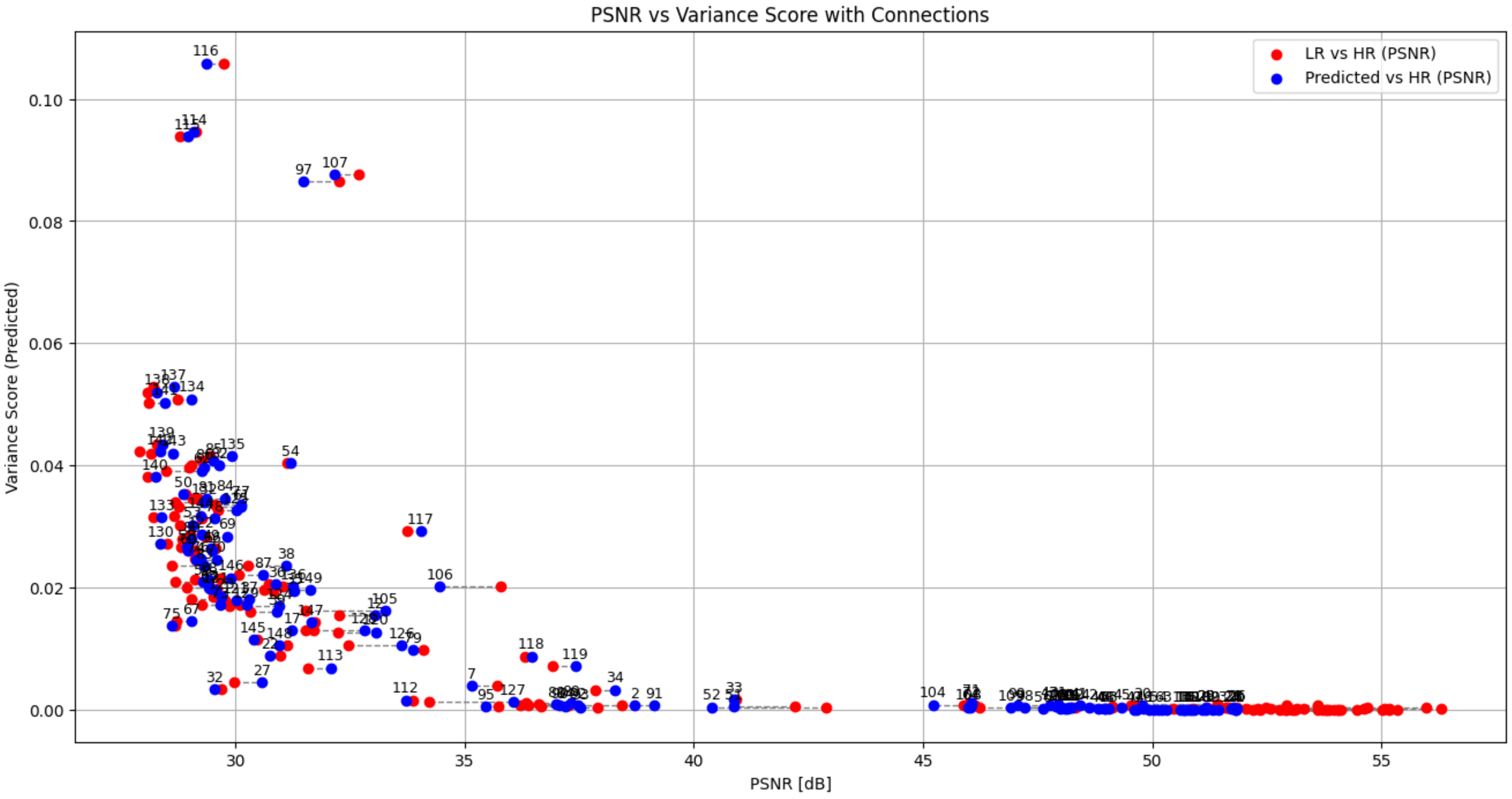
PSNR between LR and HR: 28.79 dB
PSNR between Predicted and HR: 28.96 dB

Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 38.66 dB
Average PSNR between predicted images and high-resolution images: **37.79 dB (was 34.52 dB)**

- **Average PSNR improved but lower than the baseline, why?**

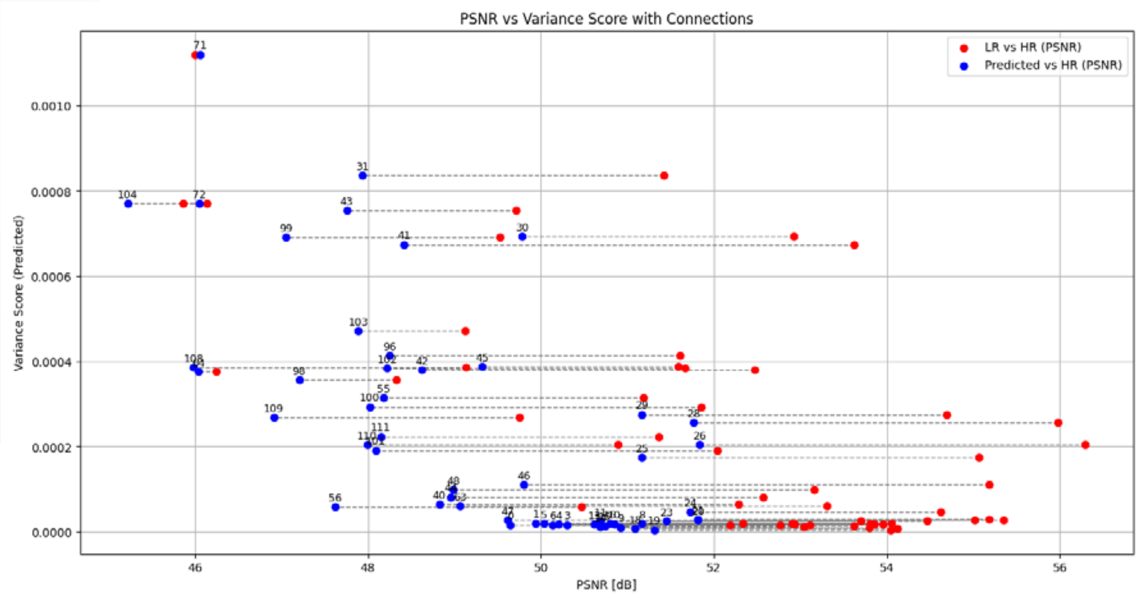
Investigation the issue

Dot plot of **variance**, LR vs HR PSNR and Predicted vs HR PSNR for each image

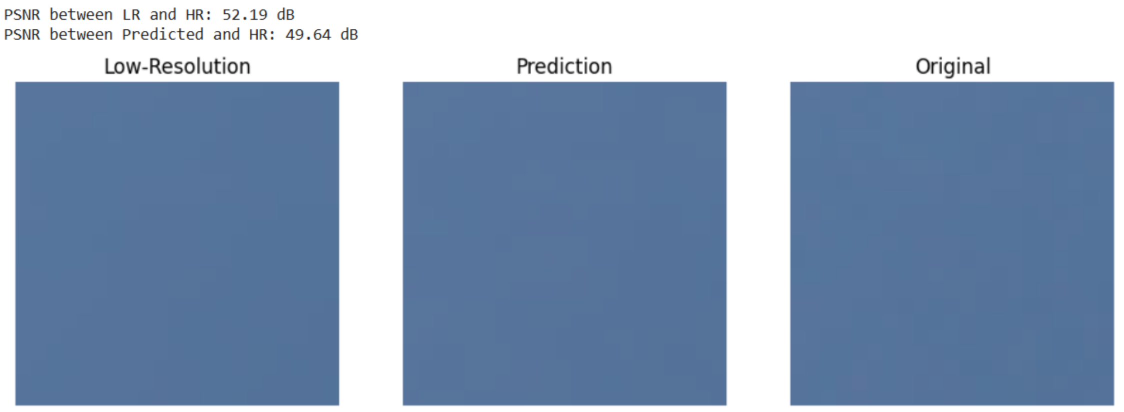


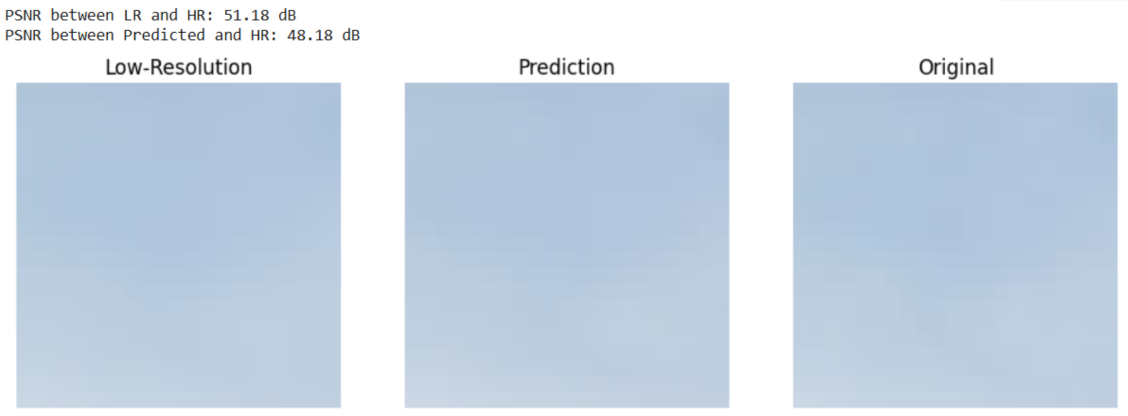
Investigation the issue

Bottom right corner of the previous plot:



- Images with the highest PSNR difference from the baseline are those with **high PSNR and low variance**
- Slight shift of color shade applied by the model
- **High impact** on the test set average PSNR because of the high PSNR values





Excluding this category of images:

Average PSNR (LR-HR): 30.99188956352703

Average PSNR (Pred-HR): 31.51631000379358

- PSNR finally improves

Third model

- Same model architecture, but with **256x256x3 input** and 244x244x3 output

PSNR between LR and HR: 31.03 dB PSNR between Predicted and HR: 31.25 dB

PSNR between LR and HR: 32.04 dB PSNR between Predicted and HR: 32.56 dB

- **Quality visually improved, but PSNR score doesn't reflect it well**

Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 37.92 dB
Average PSNR between predicted images and high-resolution images: 37.42 dB

Fourth model

- Same model architecture, same 256x256x3 input and 244x244x3 output, but **3x scale factor**

PSNR between LR and HR: 28.54 dB

PSNR between Predicted and HR: 28.56 dB

- **Performance visually worse**, as expected
- **PSNR values seem to not reflect the performance drop** compared to the previous model

Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 37.52 dB (was 37.92 dB)
Average PSNR between predicted images and high-resolution images: 37.03 dB (was 37.42 dB)

Introducing Structural Similarity Index

Model 3 performance (2x scale factor):

- Average **PSNR** between low resolution images (bicubic interpolation) and high-resolution images: 37.92 dB
- Average **PSNR** between predicted images and high-resolution images: 37.42 dB
- Average **SSIM** between low resolution images (bicubic interpolation) and high-resolution images: **0.8460**
- Average **SSIM** between predicted images and high-resolution images: **0.9050**

- Inflated by easy images
- Good gap from baseline

PSNR between LR and HR: 28.69 dB
PSNR between Predicted and HR: 28.92 dB
SSIM between LR and HR: **0.64**
SSIM between Predicted and HR: **0.78**

Model 4 performance (3x scale factor):

- Average **PSNR** between low resolution images (bicubic interpolation) and high-resolution images: 37.52 dB
- Average **PSNR** between predicted images and high-resolution images: 37.03 dB
- Average **SSIM** between low resolution images (bicubic interpolation) and high-resolution images: **0.8069**
- Average **SSIM** between predicted images and high-resolution images: **0.8173**

PSNR between LR and HR: 28.54 dB
PSNR between Predicted and HR: 28.56 dB
SSIM between LR and HR: **0.55**
SSIM between Predicted and HR: **0.58**

Second dataset and fifth model

- Mammals dataset
- Hoping that similarity can help the model learn more easily and reach better performance




Second dataset and fifth model

- Same model as model 3, 256x256 input, 2x scale factor

Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 31.88 dB
Average PSNR between predicted images and high-resolution images: 31.84 dB
Average SSIM between low resolution images (bicubic interpolation) and high-resolution images: **0.7861**
Average SSIM between predicted images and high-resolution images: **0.7948**

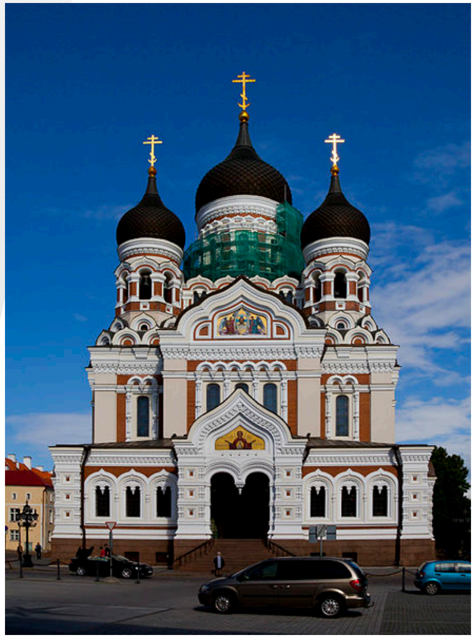
- **Disappointing results**
  - Very close to the baseline
  - Significantly worse than model 3

PSNR between LR and HR: 29.13 dB
PSNR between Predicted and HR: 29.14 dB
SSIM between LR and HR: 0.54
SSIM between Predicted and HR: 0.56

- Very small quality improvement probably due to the **broad dataset** and the **unpredictability** of certain textures like **fur**

Third dataset and sixth model

- **Architecture** dataset
- Different styles of architecture, should share some features like **sharp edges**

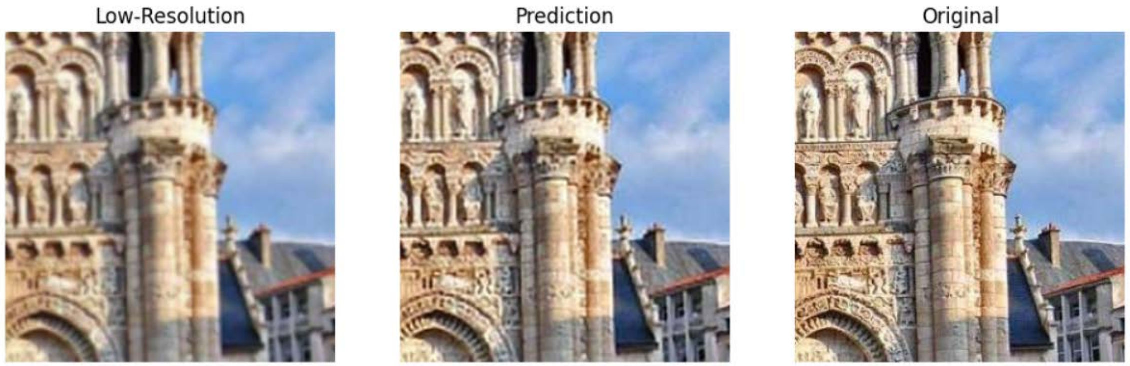


Third dataset and sixth model

- Same model as model 3 and 5, 256x256 input, **2x scale factor**

Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 33.96 dB
Average PSNR between predicted images and high-resolution images: 35.92 dB
Average SSIM between low resolution images (bicubic interpolation) and high-resolution images: **0.8554**
Average SSIM between predicted images and high-resolution images: **0.9477**

- Really good results, SSIM close to 1

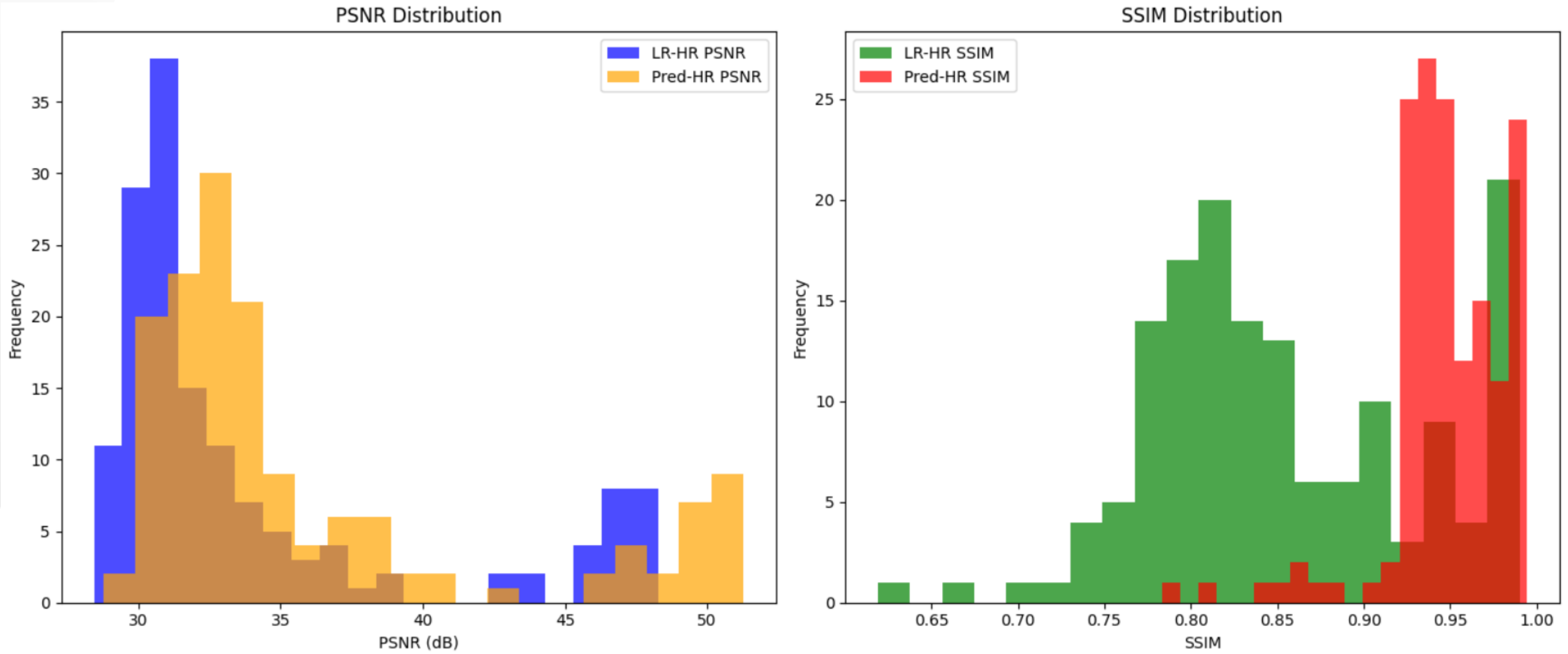


PSNR between LR and HR: 29.79 dB
PSNR between Predicted and HR: 30.31 dB
SSIM between LR and HR: **0.75**
SSIM between Predicted and HR: **0.86**

PSNR between LR and HR: 29.40 dB
PSNR between Predicted and HR: 31.61 dB
SSIM between LR and HR: **0.77**
SSIM between Predicted and HR: **0.94**

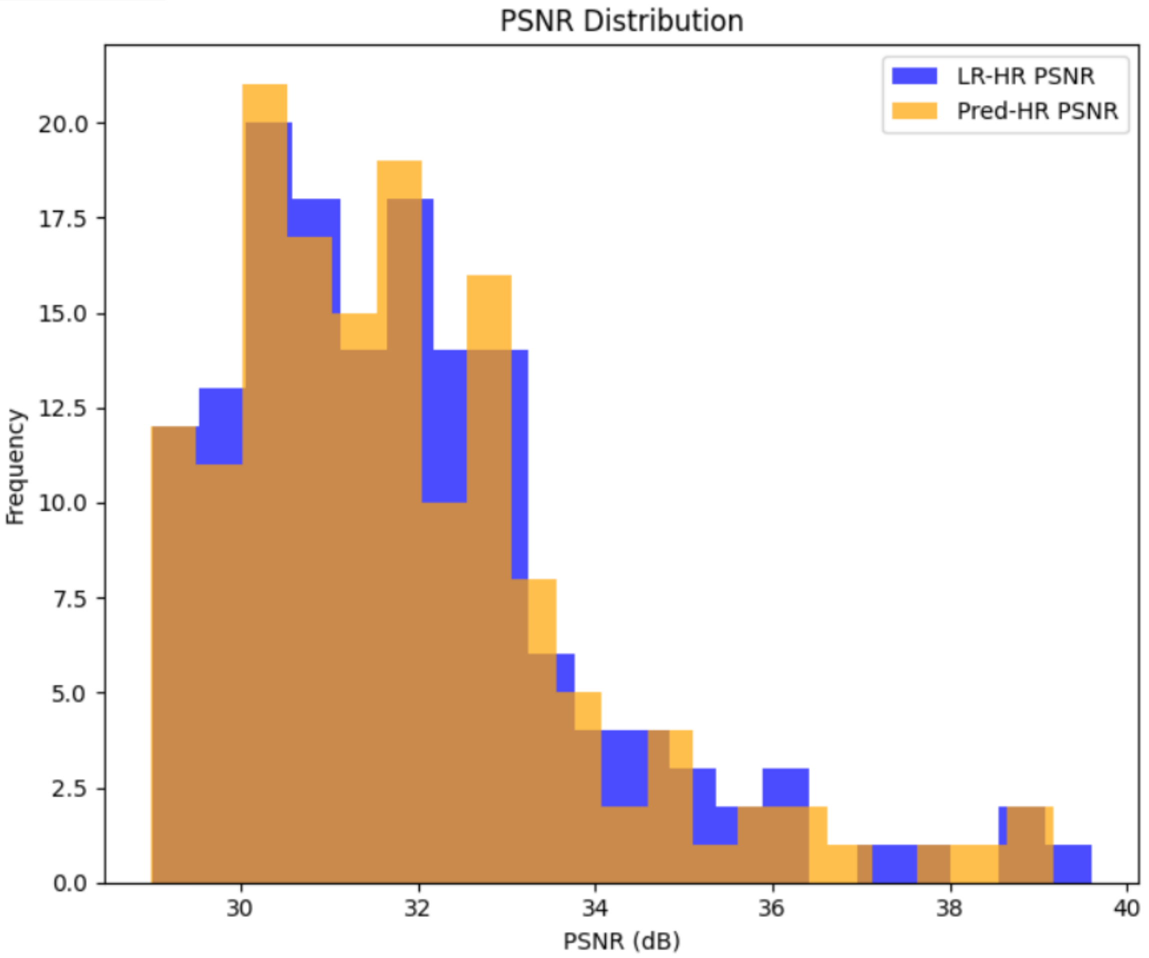
Distribution of quality metrics

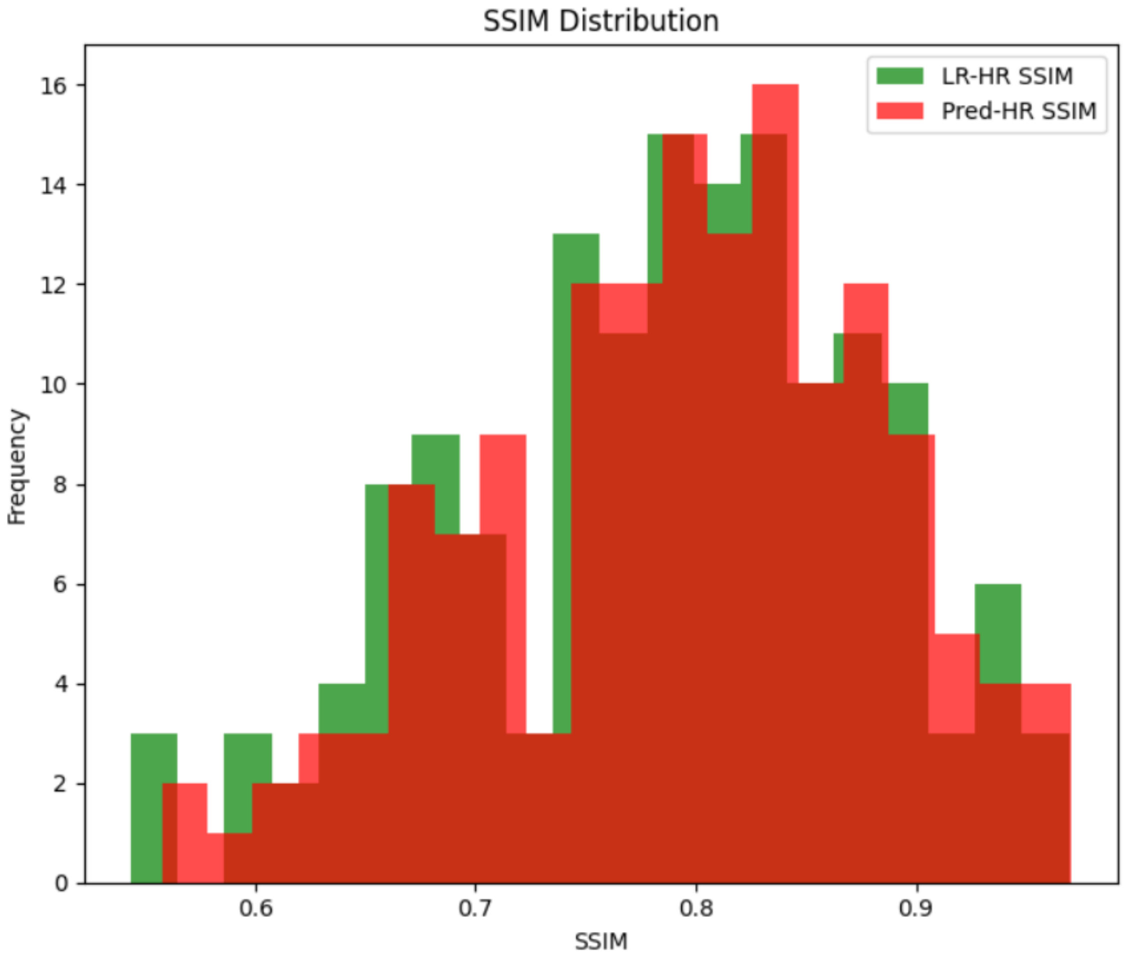
- **Model 6** (previous model, architecture dataset, 256x256 input size, 2x scale factor)



Distribution of quality metrics

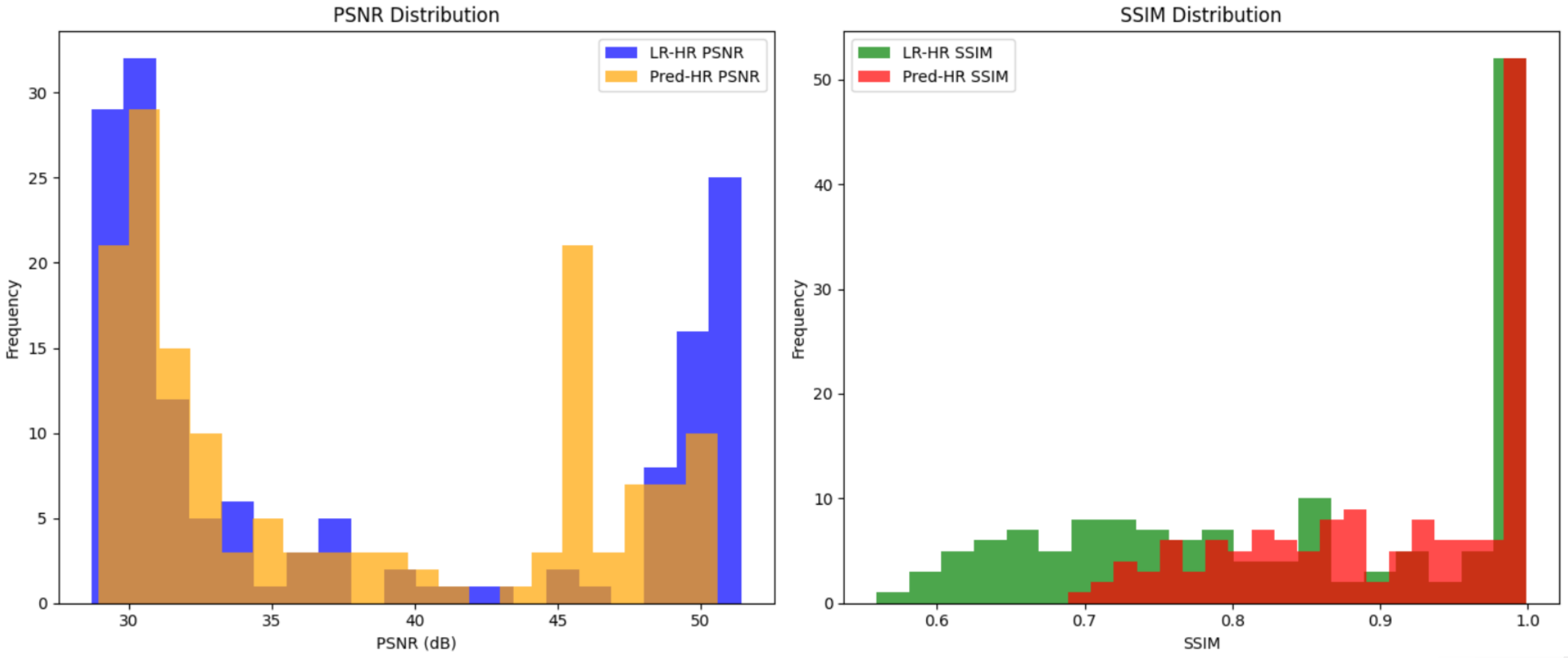
- **Model 5** (**mammals dataset**, 256x256 input size, 2x scale factor)





Distribution of quality metrics

- **Model 3** (general dataset, 256x256 input size, 2x scale factor)

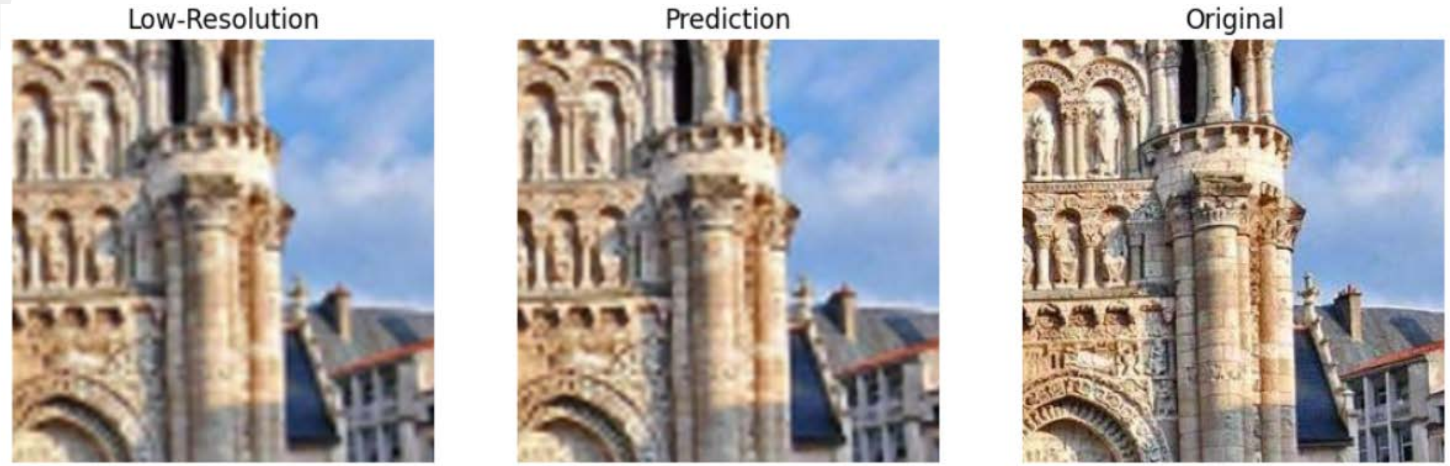


Seventh model

- Same as the previous model: third dataset, 256x256 input, but with **3x scale factor**

Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 33.46 dB
Average PSNR between predicted images and high-resolution images: 33.55 dB
Average SSIM between low resolution images (bicubic interpolation) and high-resolution images: **0.8005**
Average SSIM between predicted images and high-resolution images: **0.8176**

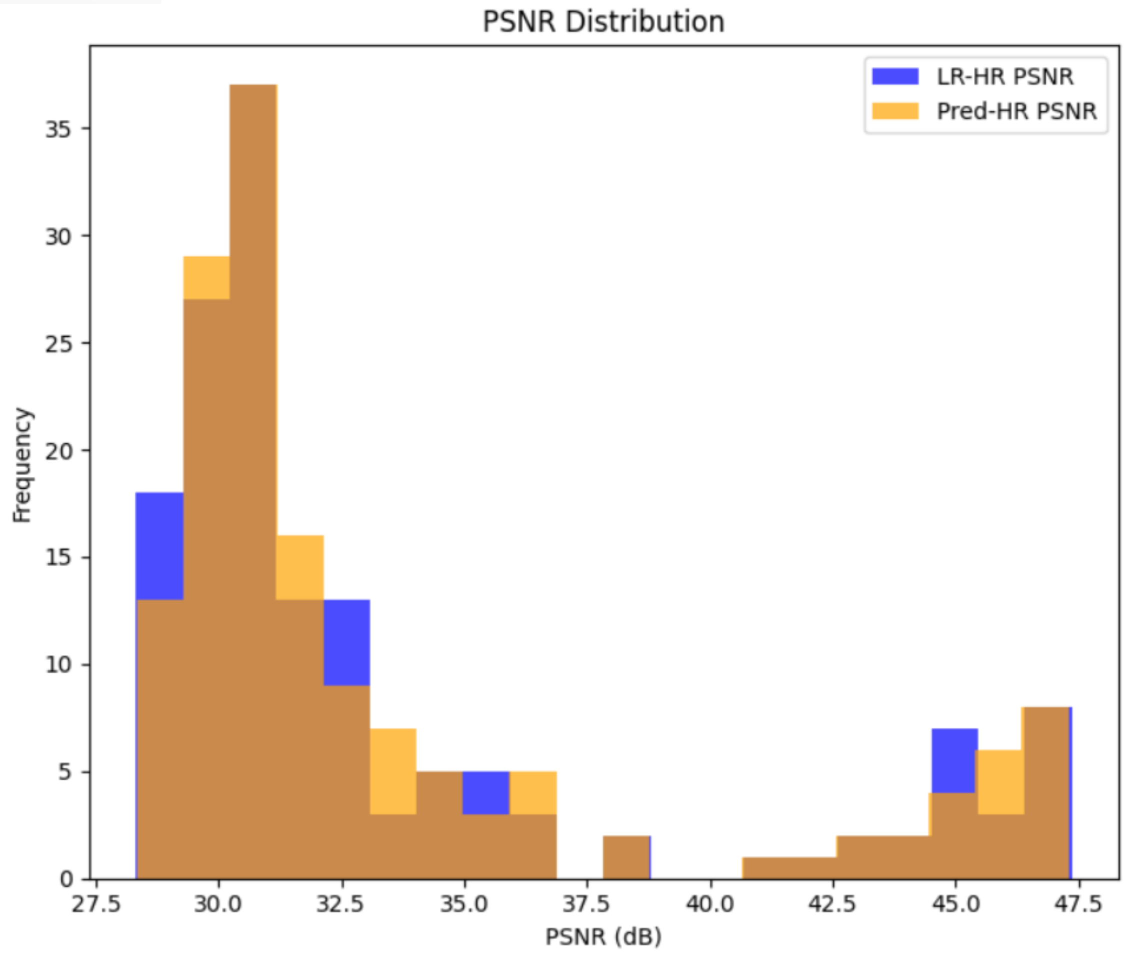
- Performance very similar to model 4
- **The model itself could be the chokepoint**



PSNR between LR and HR: 29.66 dB
PSNR between Predicted and HR: 29.71 dB
SSIM between LR and HR: 0.69
SSIM between Predicted and HR: 0.71

Distribution of quality metrics

- **Model 7** (architecture dataset, 256x256 input size, 3x scale factor)



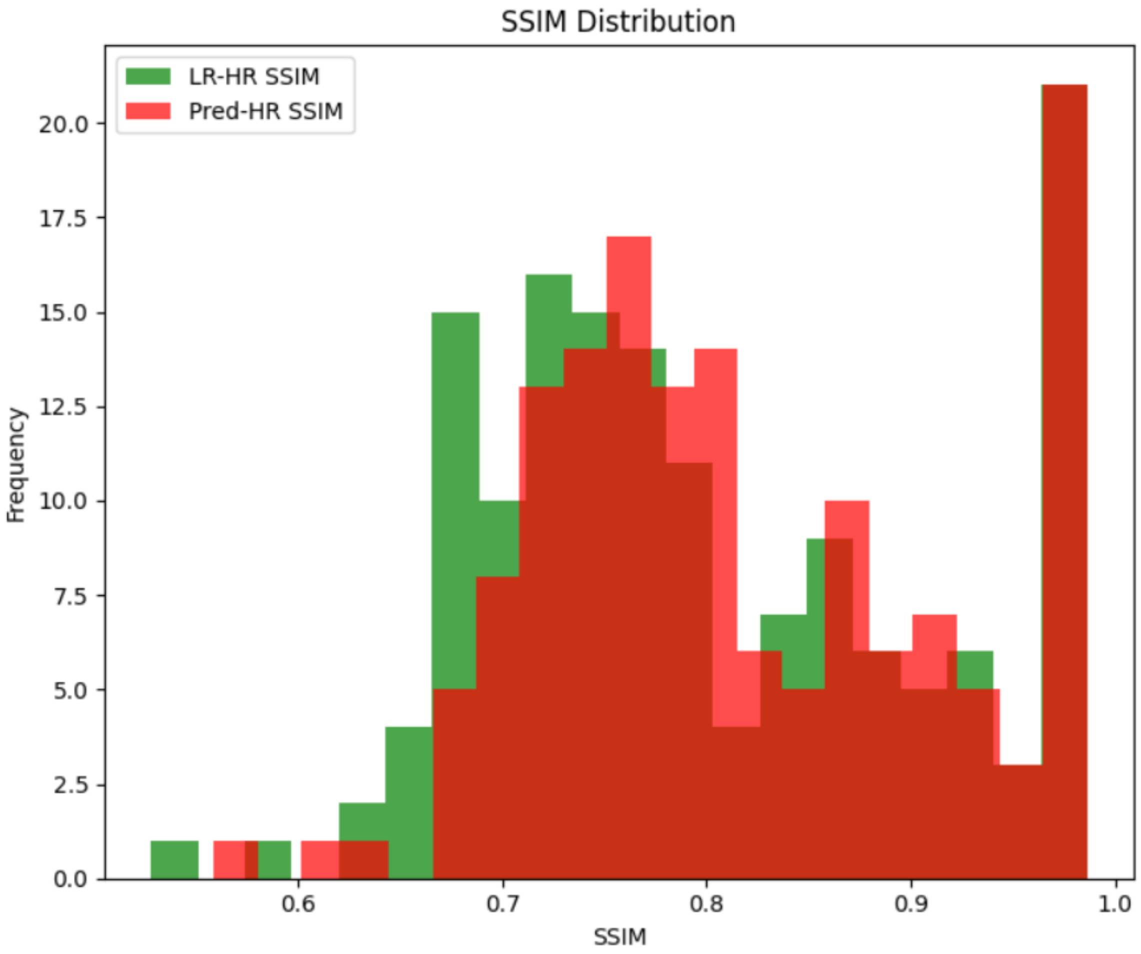
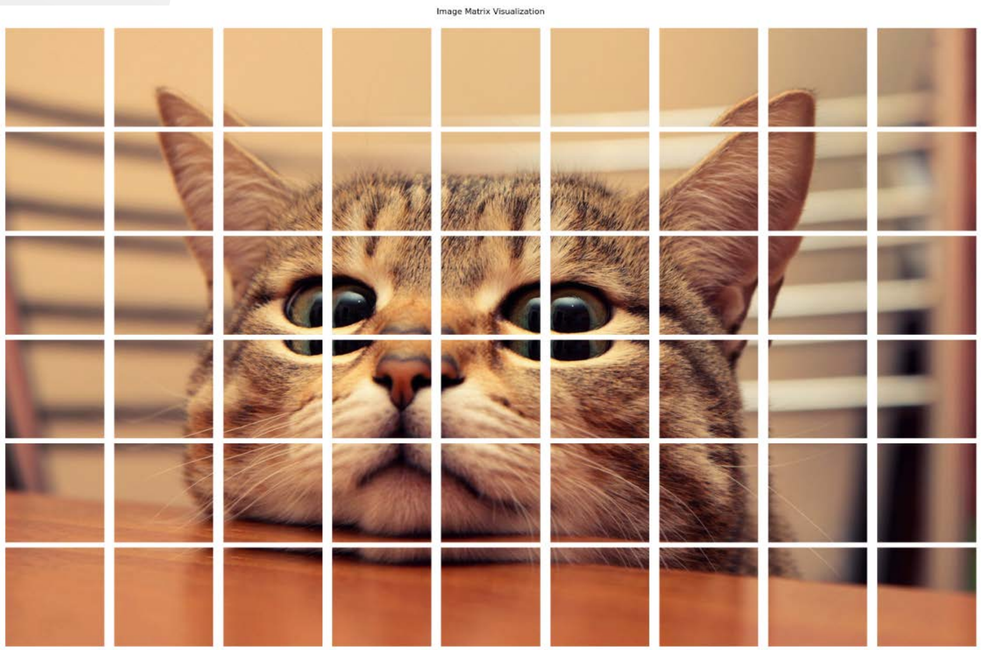
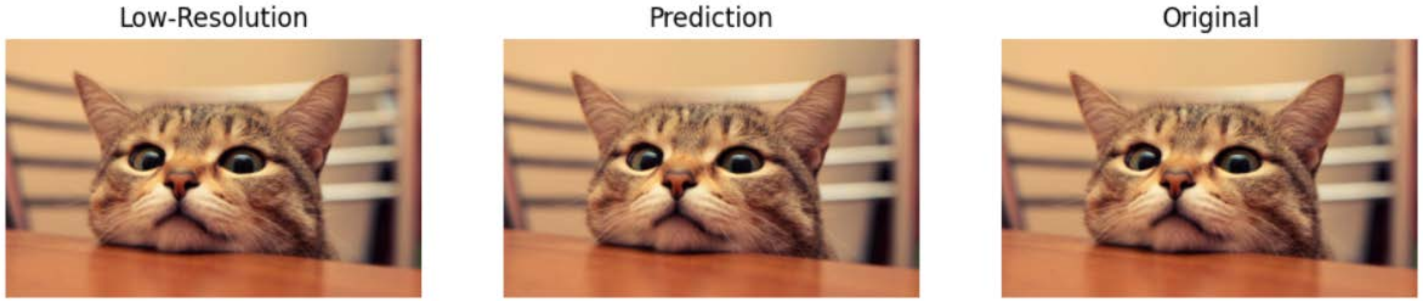


Image reconstruction on HighResTestSet

Image reconstruction

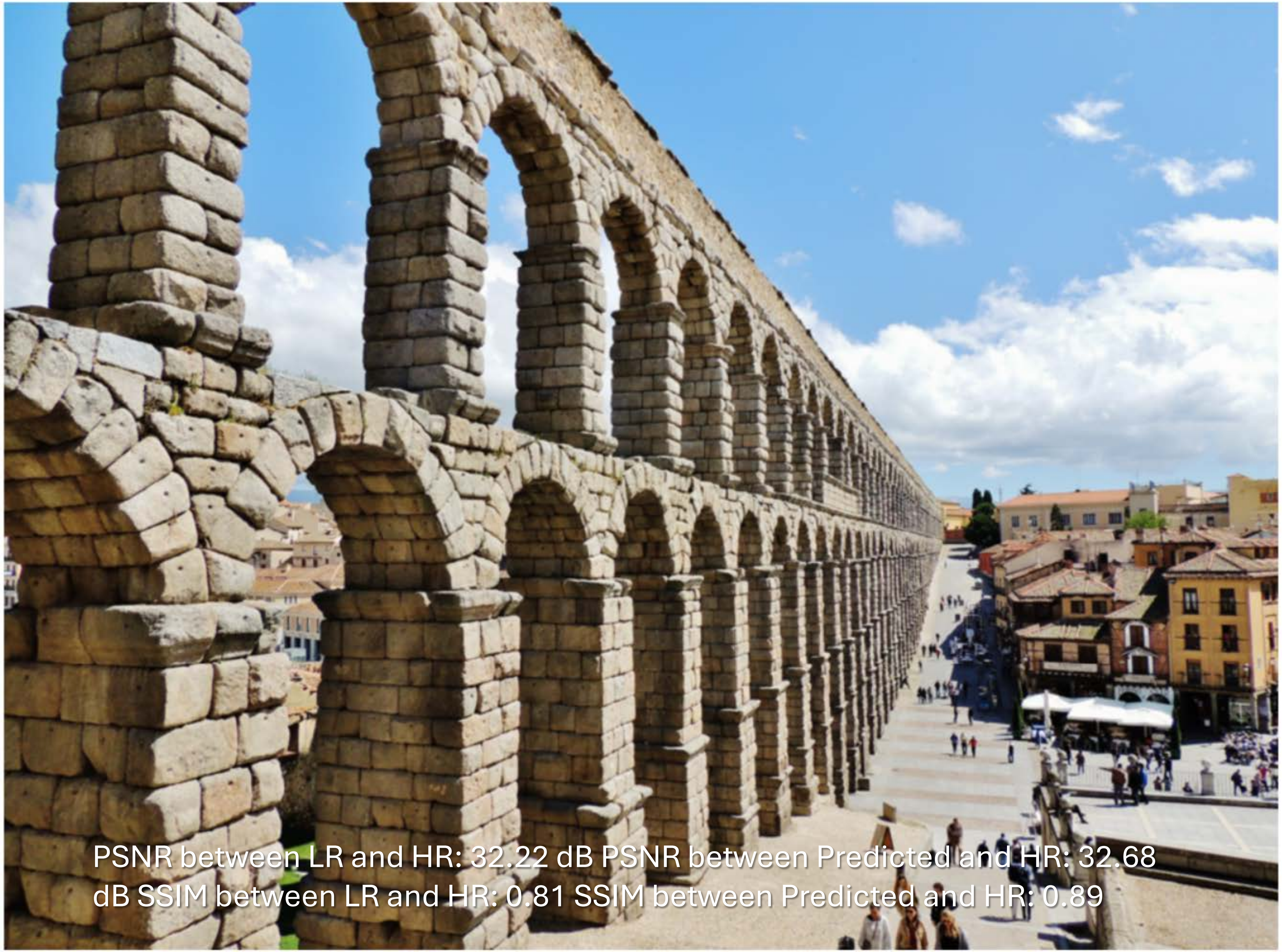
- The high resolution **images are split** by the size needed to process each patch with the SRCNN model (considering both input and output sizes)
- **Each patch is upscaled** with the 6th model (2x scale factor)
- **The patches are stitched together** for all three output images (low-resolution upscaled with bicubic interpolation, predicted image, original)
- The performance metrics are calculated directly on these resulting images



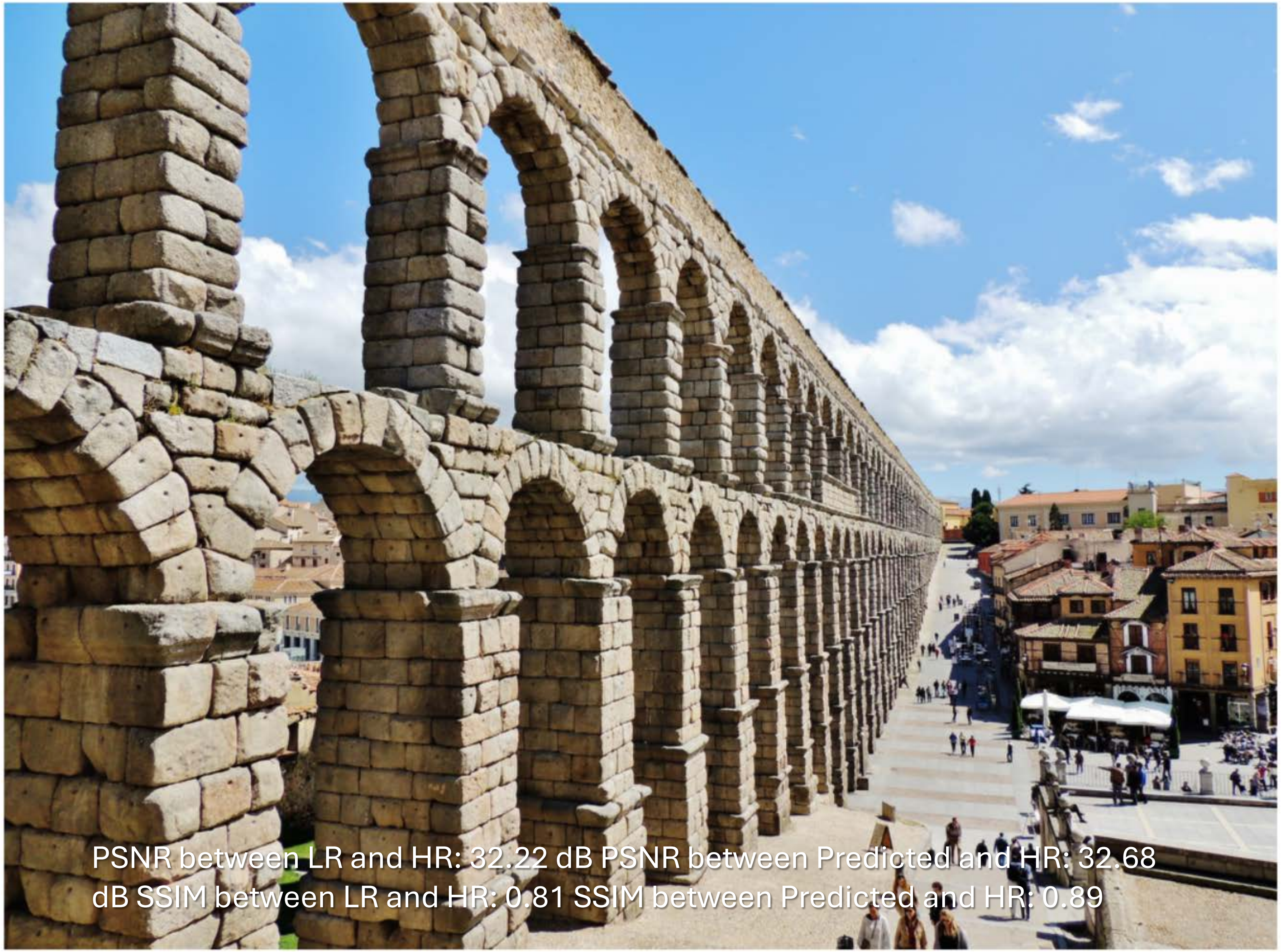


PSNR between LR and HR: 35.64 dB
PSNR between Predicted and HR: 37.16 dB
SSIM between LR and HR: 0.90
SSIM between Predicted and HR: 0.94

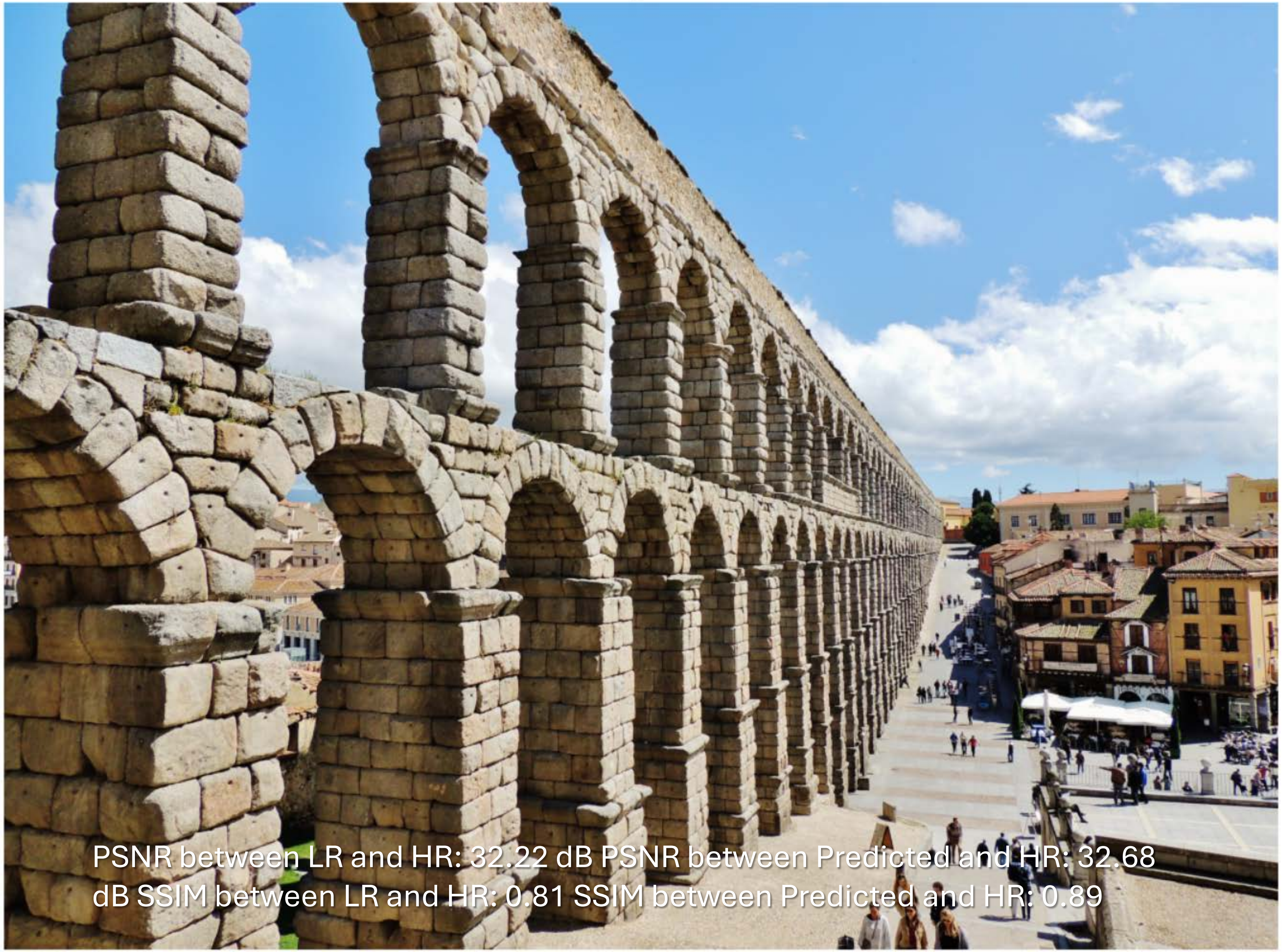
Low-Resolution



Prediction


PSNR between LR and HR: 32.22 dB PSNR between Predicted and HR: 32.68
dB SSIM between LR and HR: 0.81 SSIM between Predicted and HR: 0.89

Original


PSNR between LR and HR: 32.22 dB PSNR between Predicted and HR: 32.68
dB SSIM between LR and HR: 0.81 SSIM between Predicted and HR: 0.89

Low-Resolution

PSNR between LR and HR: 32.99 dB
PSNR between Predicted and HR: 33.54 dB
SSIM between LR and HR: 0.84
SSIM between Predicted and HR: 0.90

Prediction

PSNR between LR and HR: 32.99 dB
PSNR between Predicted and HR: 33.54 dB
SSIM between LR and HR: 0.84
SSIM between Predicted and HR: 0.90

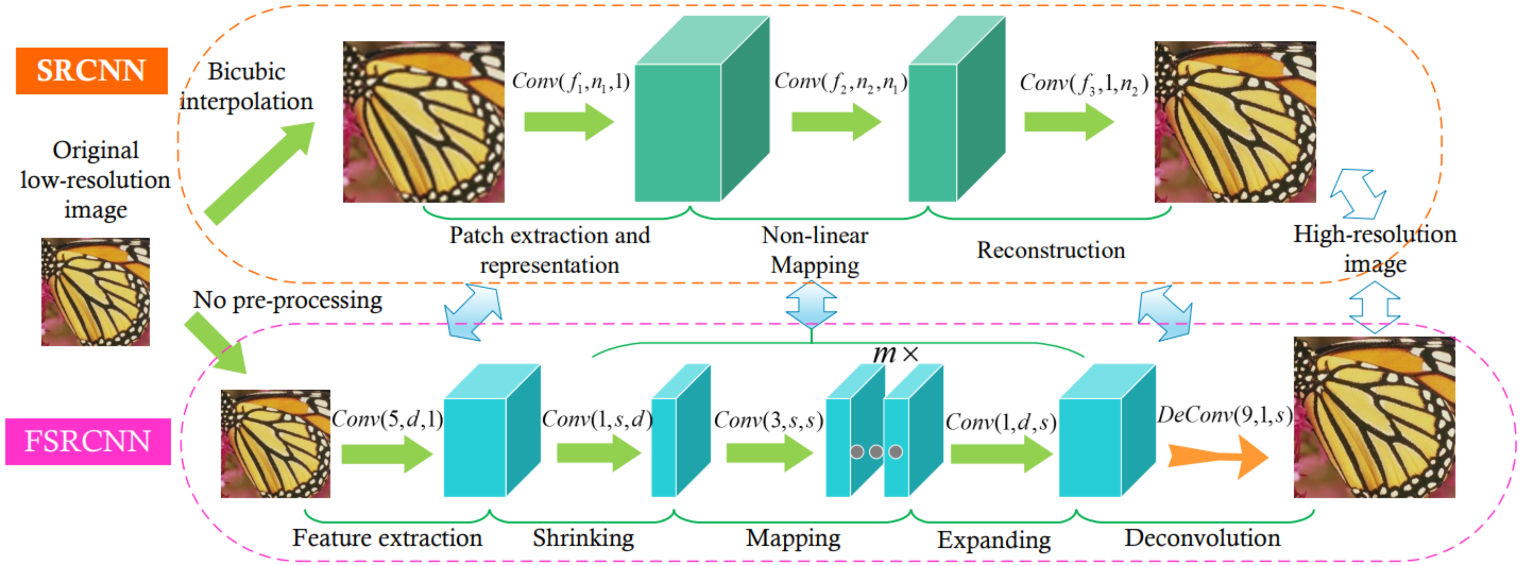
Original

PSNR between LR and HR: 32.99 dB
PSNR between Predicted and HR: 33.54 dB
SSIM between LR and HR: 0.84
SSIM between Predicted and HR: 0.90

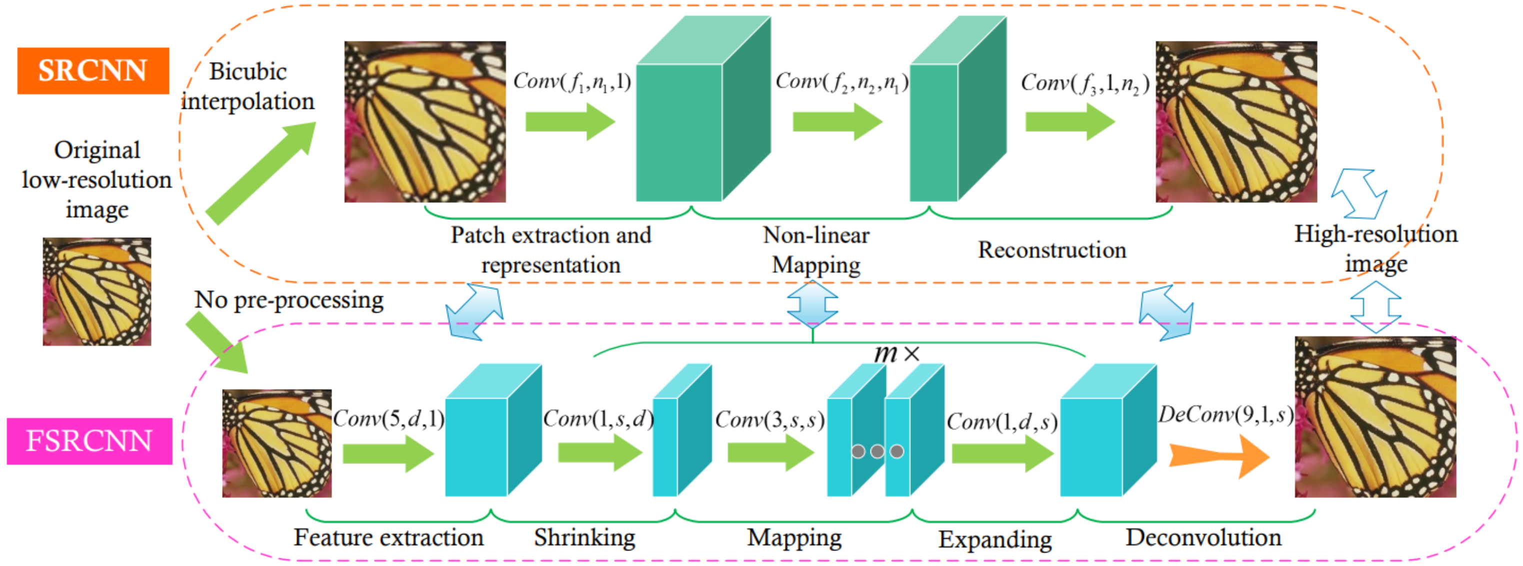
Fast Super Resolution Convolutional Neural Network (FSRCNN)

FSRCNN vs SRCNN

- Faster execution and better reconstruction
- **Takes as input the low resolution image** without any upscaling needed
- Output size bigger than the input size
- **Uses deconvolution layers** to increase the spatial dimension



FSRCNN



Feature extraction layer: extracts shallow features from the input image by applying convolutions (d=56 filters).

Shrinking layer: reduces the dimensionality of the feature maps to reduce computation and model size (s=12 filters).

Mapping layers: perform non-linear transformations in the reduced feature space to learn mappings for super-resolution (s=12 filters for every m=4 maps).

Expanding layer: expands the reduced feature maps back to higher dimensions, before the reconstruction.

Deconvolution layer: upscales the spatial resolution of the feature maps to the desired high-resolution output, based on the chosen scale factor.

First FSRCNN model

- Recreates 244x244 images (just like the previous models that took 256x256 input images) with a **2x upscale factor**, on the third dataset

Sixth model performance, for comparison:

Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 33.96 dB
Average PSNR between predicted images and high-resolution images: 35.92 dB
Average SSIM between low resolution images (bicubic interpolation) and high-resolution images: **0.8554**
Average SSIM between predicted images and high-resolution images: **0.9477**
Execution time: 18.63 seconds

First FSRCNN model:

Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 34.42 dB
Average PSNR between predicted images and high-resolution images: 34.16 dB
Average SSIM between low resolution images (bicubic interpolation) and high-resolution images: **0.8768**
Average SSIM between predicted images and high-resolution images: **0.9409**
Execution time: 17.98 seconds

- Difficult to compare the speed given the high variability between executions
- Lowering the number of filters can help make it faster

Second FSRCNN model

- **3x upscale factor**

Seventh model performance, for comparison:
Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 33.46 dB
Average PSNR between predicted images and high-resolution images: 33.55 dB
Average SSIM between low resolution images (bicubic interpolation) and high-resolution images: 0.8005
Average SSIM between predicted images and high-resolution images: **0.8176**

Second FSRCNN model:
Average PSNR between low resolution images (bicubic interpolation) and high-resolution images: 33.35 dB
Average PSNR between predicted images and high-resolution images: 32.60 dB
Average SSIM between low resolution images (bicubic interpolation) and high-resolution images: 0.7646
Average SSIM between predicted images and high-resolution images: **0.8460**

- **Good improvement from the SRCNN**, still not amazing though

- **Some improvement is now visible**

PSNR between LR and HR: 29.63 dB
PSNR between Predicted and HR: 29.92 dB
SSIM between LR and HR: **0.68**
SSIM between Predicted and HR: **0.78**

PSNR between LR and HR: 29.03 dB
PSNR between Predicted and HR: 29.48 dB
SSIM between LR and HR: **0.61**
SSIM between Predicted and HR: **0.77**

PSNR between LR and HR: 30.11 dB
PSNR between Predicted and HR: 30.42 dB
SSIM between LR and HR: **0.71**
SSIM between Predicted and HR: **0.80**

PSNR between LR and HR: 30.14 dB
PSNR between Predicted and HR: 30.56 dB
SSIM between LR and HR: **0.64**
SSIM between Predicted and HR: **0.80**

Thanks for Your Attention