PROBA-V Project Report

Rizandi Gemal Parnadi

July 2019

1 Problem

1.1 Problem Definition

PROBA-V is an earth observation satellite designed to map land cover and vegetation growth across the entire globe. It provides images with two spatial resolutions:

- 300m resolution images every day
- 100m "high resolution" images, roughly every 5 days

The goal of this challenge is to construct such high-resolution images by fusion of the more frequent 300m images. The images provided for this challenge are not artificially degraded, but are real images recorded from the very same scene, just at different resolutions and different times.

This process of combining multiple low resolution images to reconstruct a high resolution image is known as **Multi-Image Super-Resolution (MISR)**.

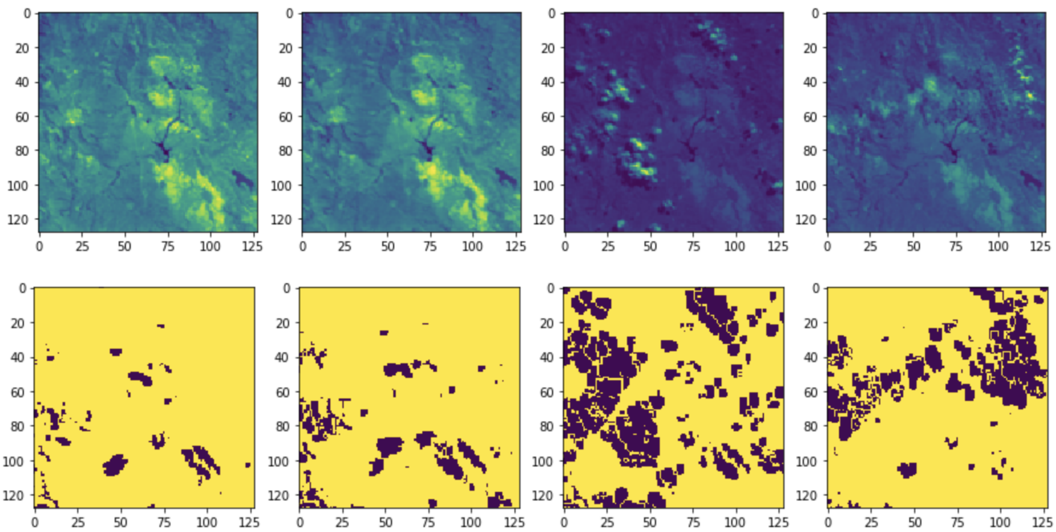


Figure 1: Low resolution input maps (top) and their corresponding quality maps (bottom)

1.2 Data

Input: Multiple 128x128 images per scene. At least 9 images per scene, 19 images on average. Quality maps indicating clear pixel (from clouds, shadows, etc) for each image.

Target: One 384x384 image per scene. Status map indicating clear pixels.

1.3 Scoring

We assume that the pixel-intensities are represented as real numbers $\in [0,1]$ for any given image. Furthermore, for any given image I with a corresponding quality map, we define $clear(I)$ as the set of pixel coordinates that are indicated as clear for image I. For every possible u, v, we first compute the bias in brightness b as follows:

$$b = \frac{1}{\left|clear\left(HR_{u,v}\right)\right|}\left(\sum_{x,y\in c\,\mathrm{loar}\left(HR_{u,v}\right)} HR_{u,v}(x,y) - SR(x,y)\right)$$

Next, we compute the corrected clear mean-square error $cMSE$ of SR w.r.t. $HR_{u,v}$

$$\mathrm{cMSE}\left(HR_{u,v}, SR\right) = \frac{1}{\left|c\,\mathrm{lear}\left(HR_{u,v}\right)\right|}\sum_{x,y\in clear\left(HR_{u,v}\right)} \left(HR_{u,v}(x,y) - (SR(x,y) + b)\right)^2$$

which results in a clear Peak Signal to Noise Ratio of

$$\mathrm{cPSNR}\left(HR_{u,v}, SR\right) = -10 \cdot \log_{10}\left(\mathrm{cMSE}\left(HR_{u,v}, SR\right)\right)$$

Let $N(HR)$ be the baseline cPSNR of image HR as found in the file norm.csv. The individual score for image SR is

$$z(SR) = \min_{u,v\in\{0,\ldots,6\}}\left\{\frac{N(HR)}{cPSNR\left(HR_{u,v}, SR\right)}\right\}$$

finally, the overall score of the submission is

$$Z(\mathrm{submission}) = \frac{1}{|\mathrm{submission}|}\sum_{SR\in\mathrm{submission}} z(SR)$$

1.4 Literature Search

Single Image Super-Resolution with Deep Learning is relatively well researched. On the other hand, using Deep Learning for Multi-Image Super-Resolution is an under-researched area, with only one very recently published architecture (EvoNet[6]).

To tackle this shortcoming, I decided to improvise and develop my own custom Deep Learning architecture for Multi-Image Super-Resolution, taking inspiration from related research.

Other challenges of this project:

- Small dataset (~1000 training images).
- Cannot easily use/modify/do transfer learning from pre-trained models because of the nature of the images.

1.4.1 Single Image Super-Resolution (SISR)

- SRCNN[1] (2014) https://arxiv.org/abs/1501.00092
- ESPCN[3] (2016) https://arxiv.org/abs/1609.05158
- SRGAN[4] (2016) https://arxiv.org/abs/1609.04802
- SRResNet[4] (2016) https://arxiv.org/abs/1609.04802
- EDSR[5] (2017) https://arxiv.org/abs/1707.02921

1.4.2 Multiple Image Super-Resolution (MISR)

- EvoNet[6] (2019) https://arxiv.org/abs/1903.00440

2 Approach

2.1 Workflow

Development followed roughly the framework by Josh Tobin. Key strategies:

- Start with simple baseline, use sensible defaults
- Gradually ramp up complexity
- Once each model runs, overfit a single batch to make sure it can learn (no model bugs)
- Git branch each experimental architecture

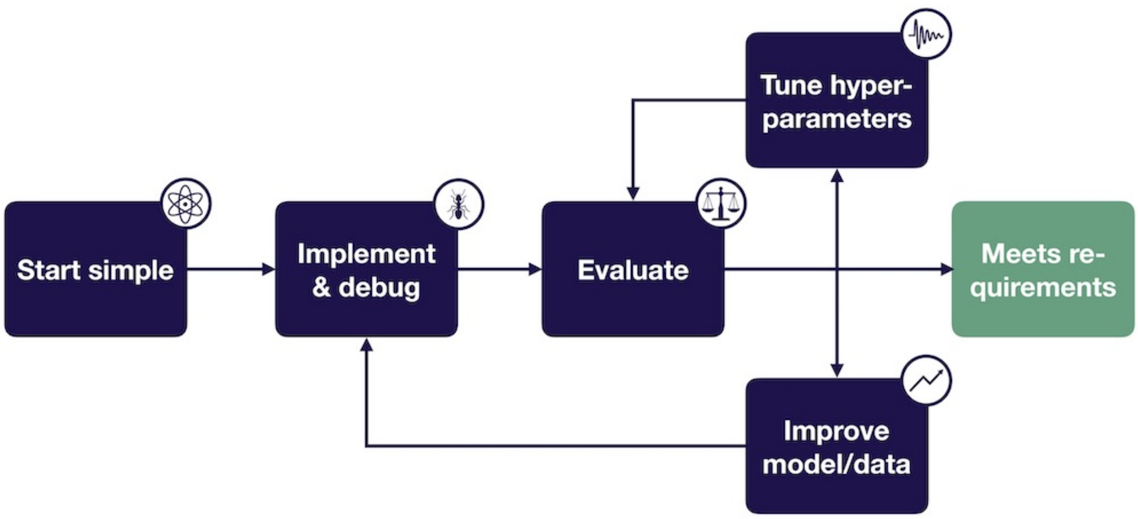


Figure 2: Workflow

I developed and trained 4 architectures (see git branches):

- Baseline: simple upsampling
- SRCNN[1]
- U-Net[9]-based architecture with residual blocks
- Custom architecture (see section 2.2)

2.2 Custom Architecture

I developed a custom deep learning architecture specifically for this task. The architecture is made up of 3 main parts: the head block, the middle (mini U-Net) block and the upsample block:

- The head block is two series of 3x3 Convolutions followed by ReLU and Batch Normalization.
- The middle block is a small U-Net[9]-inspired architecture.
- The upsample block is a 3x3 Transposed Convolution with stride 3, followed by regular convolutions.
- Each Encode/Decoder/Center Block of the mini U-Net is made up of residual building blocks.

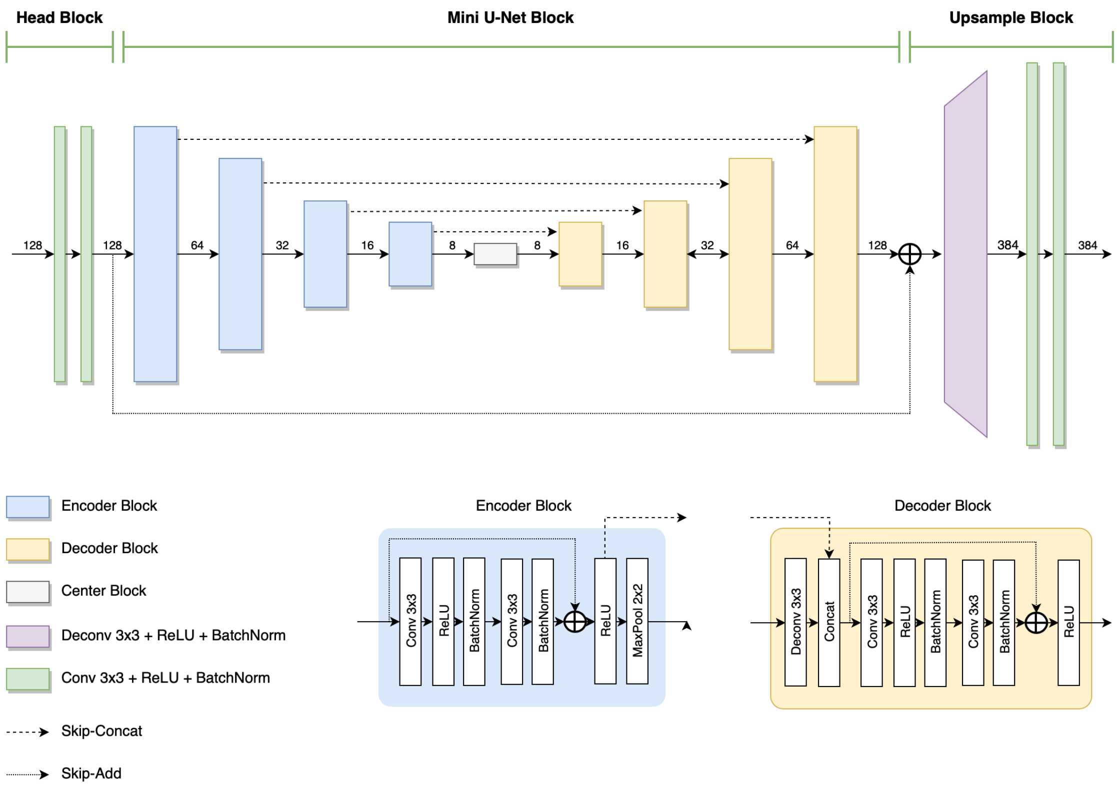


Figure 3: Custom Architecture

- A long skip connection skipping the entire middle block.

Justifications for this architecture:

- Baseline performance is achieved by simply upsampling the input image, better performance is likely achieved with small detail perturbations of this upsampled image.
- From experience, U-Net learns well even with small datasets, skip connections help information flow.
- **In the worst case**, the skip connections enable the entire middle block to learn to be an identity function, simply passing information through. So performance will never be worse than baseline. This helps training tremendously with such a deep network.
- **In the best case**, learned detail will be added by the network through **residual learning**.

The network has around 8.4 million parameters.

I decided against using GANs for this task, since they tend to make up/'hallucinate' details that don't actually exist in the input, instead of reconstructing actual details.

2.3 Data Pipeline

For each scene, we take 4 low resolution images with maximum clearance, feeding it to the network as a 128x128x4 tensor. This is augmented with non-random rotations.

2.4 Training

2.4.1 Cyclical Learning Rates & Superconvergence

Cyclical Learning Rates (CLR)[7] is a scheme to accelerate the training of neural networks. Instead of monotonically decreasing the learning rate, this method lets the learning rate cyclically vary between reasonable boundary values. Training with cyclical learning rates instead of fixed values achieves improved classification accuracy without a need to tune and often in fewer iterations.

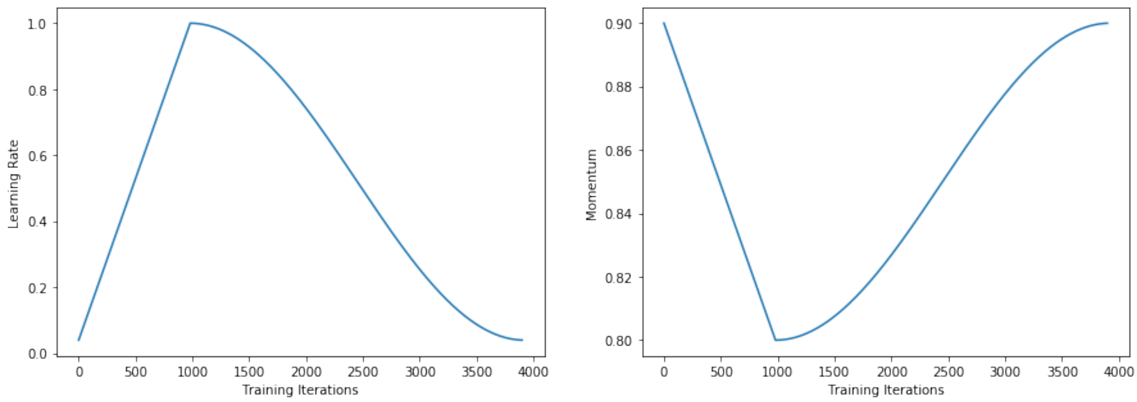


Figure 4: Learning rate (left) and momentum (right)

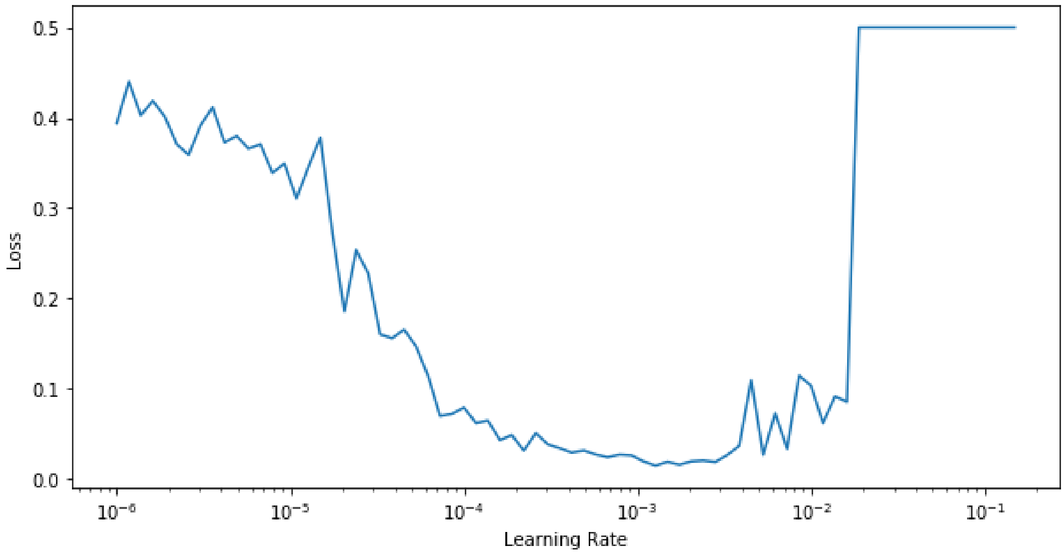


Figure 5: "Learning rate finder": loss against learning rate

Super-convergence[8] is a phenomenon where we can train a neural network in far fewer training iterations, up to one order of magnitude faster compared to conventional training methods. We achieve this in a similar way to CLR, but with a "Once Cycle Policy": the learning rate starts low then increases to a maximum, then gradually lowers again. Meanwhile, momentum goes through the same cycle in inverse. Here we use a linear increase for the first 25% of training, then gradually decrease with cosine annealing.

To find the maximum learning rate, we perform gradient descent by incrementally increasing the learning rate, while plotting the losses against it. The maximum learning rate in order to achieve super-convergence is the highest we can go before the losses start to 'explode'.

Benefits of training this method for training are:

- Much faster convergence, in up to one order of magnitude fewer iterations.
- Removes the need to find optimal learning rate and schedules.
- Adds a slight regularization effect.

It's particularly suitable for quick iterative projects like this one.

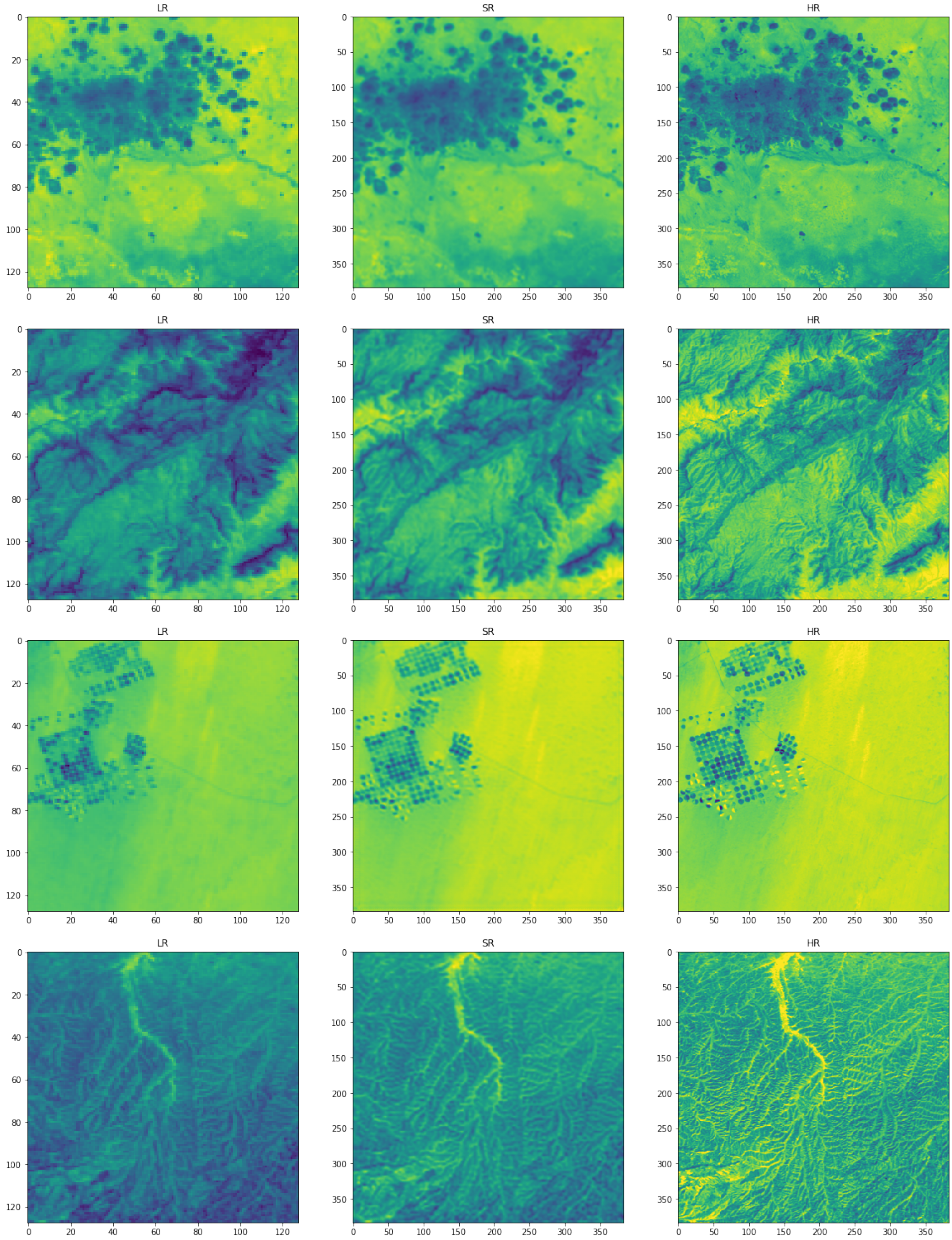
3 Results

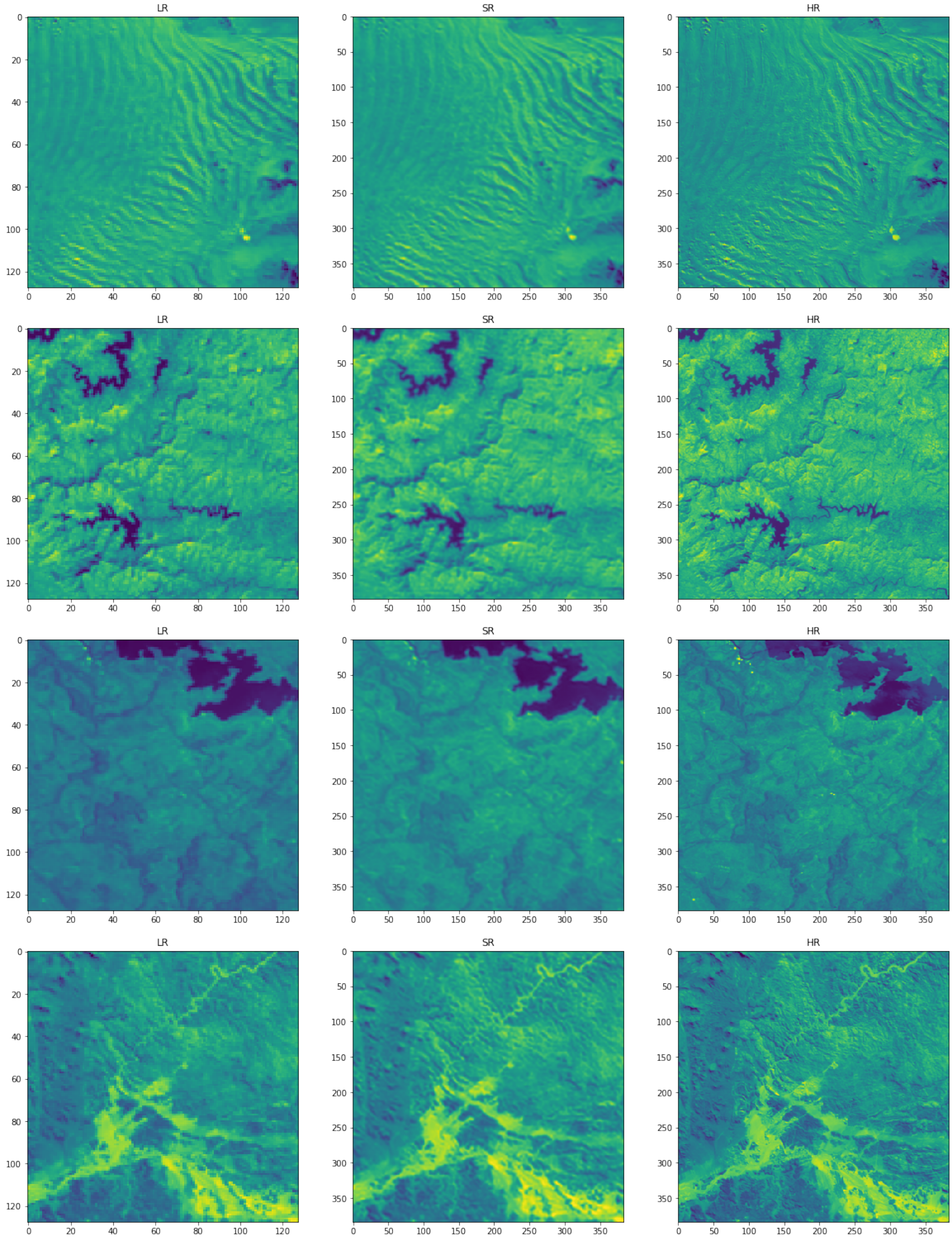
Scores (Z) against validation set (lower is better):

- Baseline: 1.0199454626478197
- SRCNN: 1.0160809862117985
- U-Net-based: 1.0081685029538871
- **Custom architecture: 0.9989938039605468**

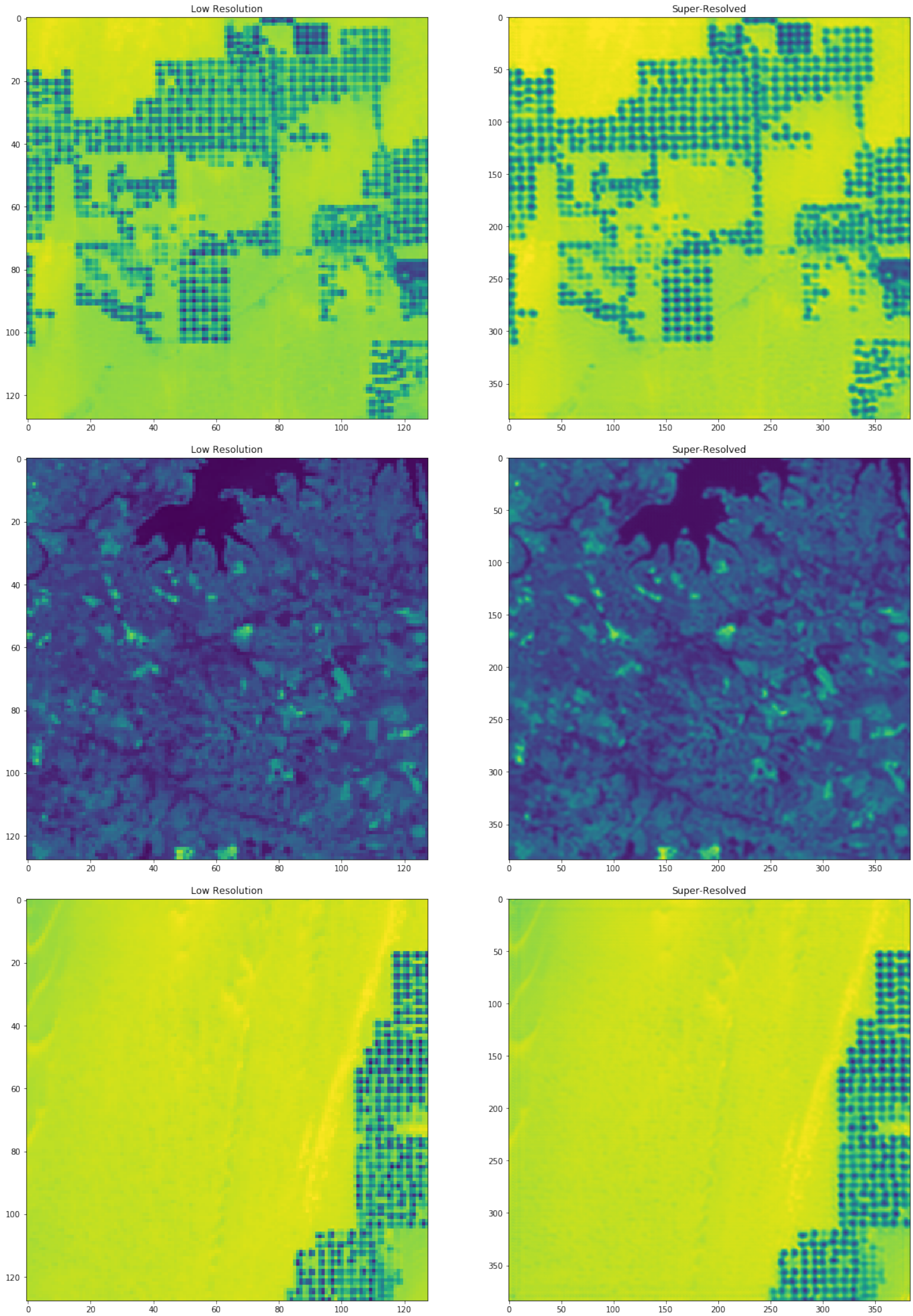
3.1 Results on Validation Set: Custom Architecture

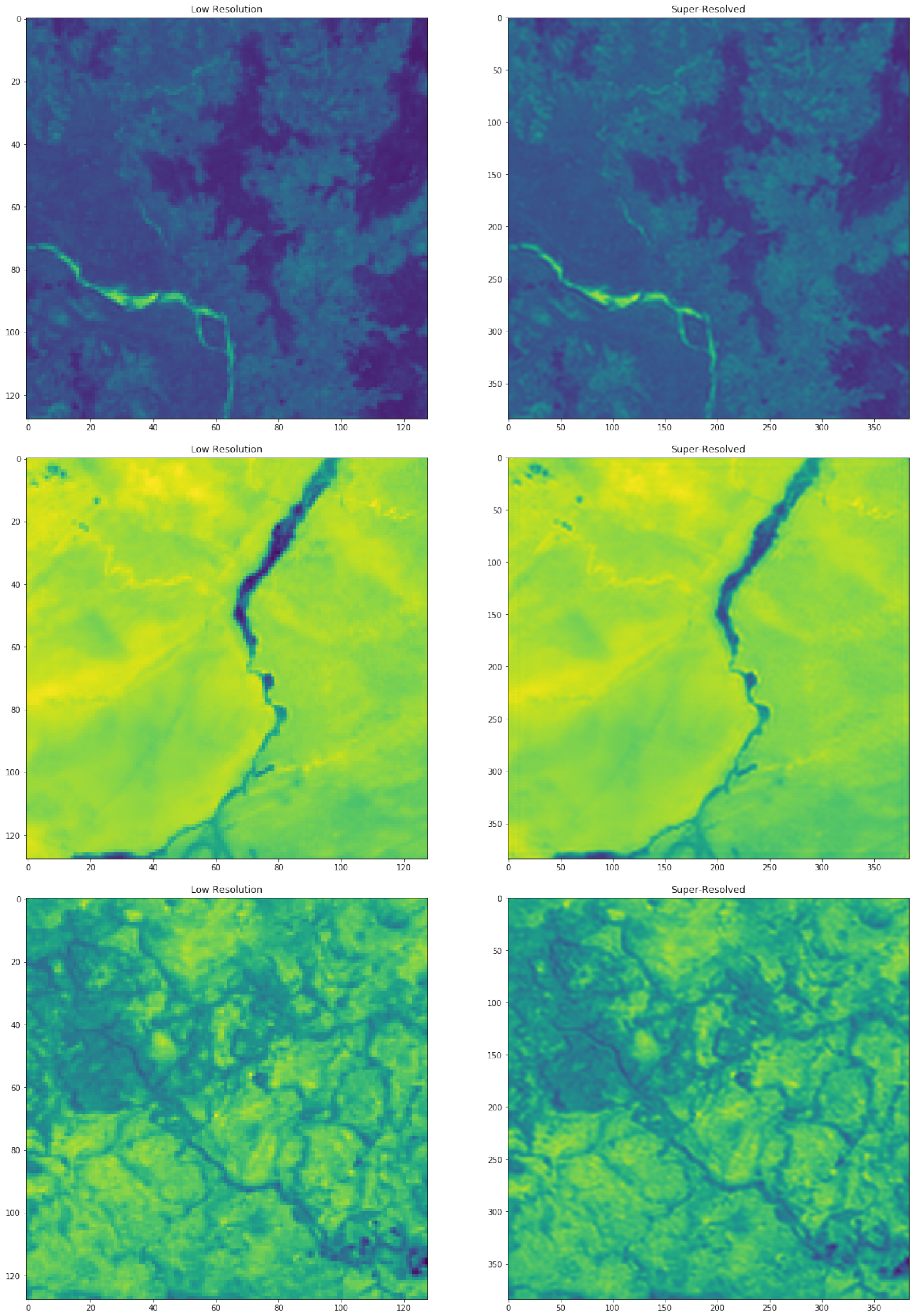
LR: low resolution input image. **SR**: super-resolved image. **HR**: ground truth high resolution image.

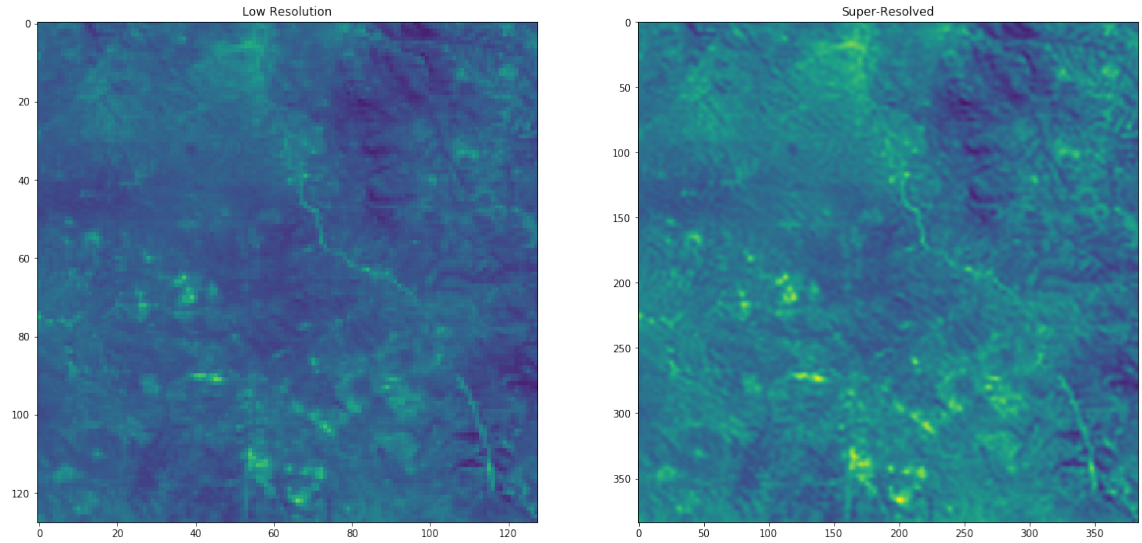



LR
SR
HR
LR
SR
HR
LR
SR
HR
LR
SR
HR

3.2 Results on Test Set: Custom Architecture




Low Resolution
Super-Resolved
Low Resolution
Super-Resolved
Low Resolution
Super-Resolved
0 20 40 60 80 100 120
0 50 100 150 200 250 300 350



4 References

[1] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In ECCV, 2014.

[2] J. Kim, J. K. Lee, and K. M. Lee. Accurate image superresolution using very deep convolutional networks. In CVPR, 2016.

[3] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In CVPR, 2016.

[4] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Te- jani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In CVPR, 2017.

[5] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In CVPRW, 2017.

[6] M. Kawulok, P. Benecki, S. Piechaczek, K. Hrynczenko, D. Kostrzewa, and J Nalepa. Deep Learning for Multiple-Image Super-Resolution. In IEEE Geoscience and Remote Sensing Letters, 2019.

[7] L. N. Smith. Cyclical Learning Rates for Training Neural Networks. In WACV 2017.

[8] L. N. Smith, N. Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. 2017.

[9] O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In MICCAI 2015.