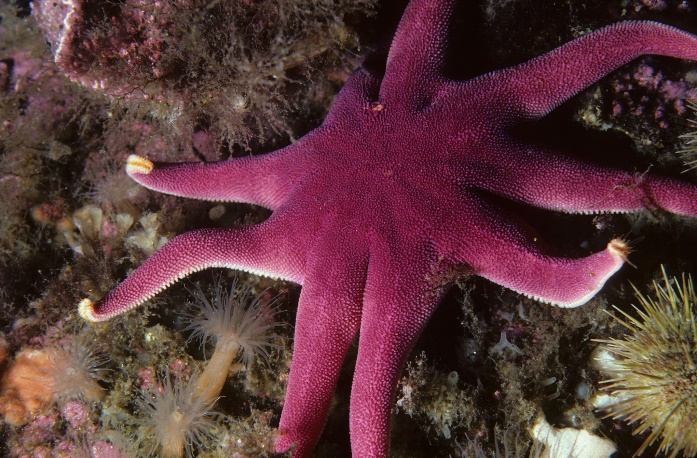
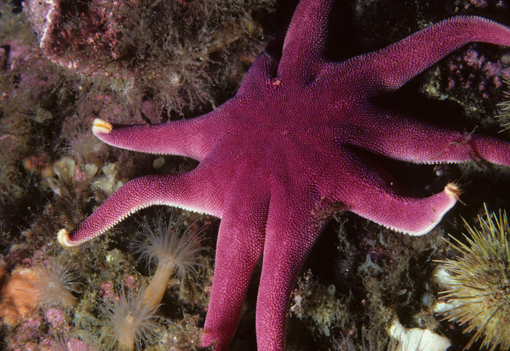
Data Explanation:

CODE ADAPTED FROM: <https://github.com/krasserm/super-resolution>

* Average Downsampling is to average the pixels in a sample area and then substitute the entire area with the average pixel color at the specified resolution.
* Bicubic Downsampling is to use a weighted average to decide pixel color. This method usually downsamples better than Average Downsampling as far as the quality of downsampled images is concerned. Bicubic Downsampling is the most precise method, but it takes time to yield the smoothest tonal gradations.
* Subsampling is to choose a pixel in the center of the sample area and then substitute the entire area with the chosen pixel at the specified resolution. This method reduces the conversion time as compared with downsampling, but generates images of less smoothness and continuity.

<http://www.pdflogic.com/chm/index.html?ImageProcessing>

 LOW RES (Bicubic Downsampled) HIGH RES (X4)

LOW RES (Bicubic Downsampled) HIGH RES (X4)

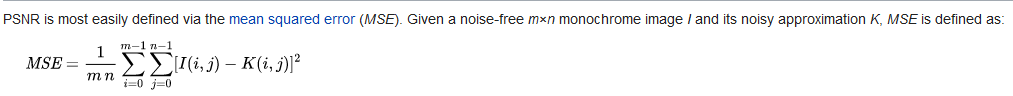


ZOOMED IN EXAMPLE

**PSNR and Loss function**

**PSNR is our metric of goodness here.** [**https://en.wikipedia.org/wiki/Peak\_signal-to-noise\_ratio**](https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio)

Measuring quality of reconstruction:



Looks like it’s pixel wise mean square error / divergence.

The Loss noted by the model during training is mean absolute error for both EDSR and WDSR.

Adam Optimizer w/ learning rate scheduler -> PiecewiseConstantDecay

**DATA:**

Use only images w/ bicubic downsampling and X4 upscaling resolution.

* 800 low res images (train)
* 800 high res images (test)
* 100 low res images (validation)
* 100 low res images(validation)

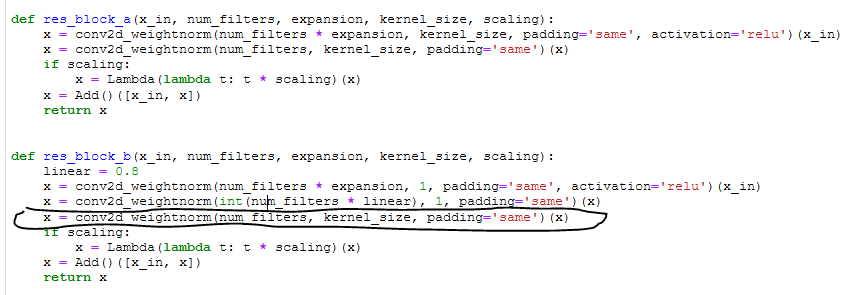
train images go through a data augmentation pipeline (random flips/rotations/crops)

**Models:**

Three models. EDSR, WDSRa, WDSRb

EDSR is the benchmark, and was SotA at the time of our paper’s release. This paper proposes lower PSNR loss with few parameters for better real time SISR performance.

Subtle differences between WDSRa and WDSRb in the form of residual block. Line 50: wdsr.py. wdsra uses resblock a, wdsrb uses resblock b. Only difference I’m seeing.



\*I had to rewrite some of the WDSR models to remove conv2D weightnorm which has a tensorflow-addon dependency that is only available for Linux/Mac. E.g., there’s no layer-wise weight normalization in these models. RE: edsr.py. All conv2d\_weightnorm operations changed to be just Conv2D

* We’ll use the same number of resblocks for all three models for valid comparison of PSNR and model parameters. Resblocks as per the paper.
* During training, 10 validation images will be used to gauge test loss (PSNR) @ every 1000 steps. The models run for 10000 steps. **The paper does not specify this step. The Git Repo we’re using defaults to 300k steps, but this is computational inefficient (Would take weeks to train on my PC).**
* Model checkpointers to save best performing PSNR model weights.

PSNR will be evaluated on the remaining test set.