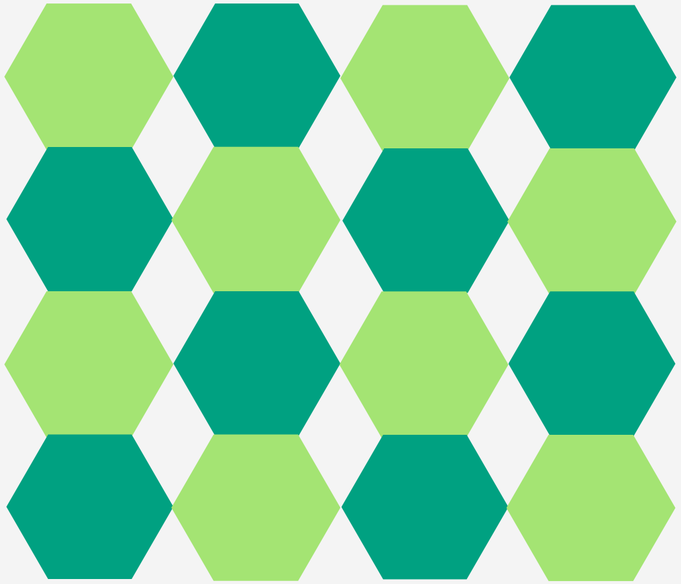
Digital Signal and
Image Management

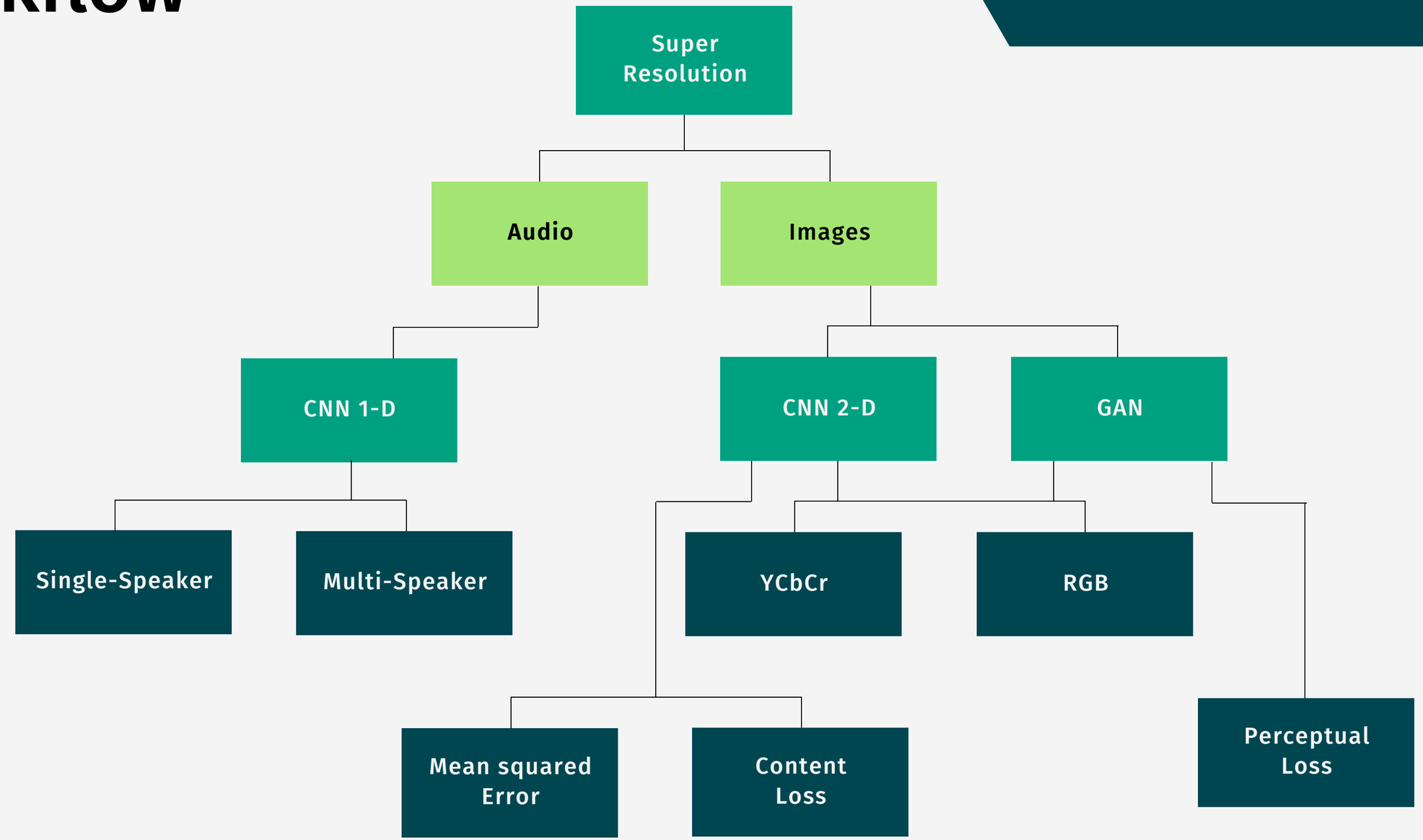
Super-Resolution di segnali 1D e 2D



Convolutional Neural Networks e Generative Adversarial Networks

Gaetano Chiriaco, Riccardo Porcedda, Gianmarco Russo

Workflow

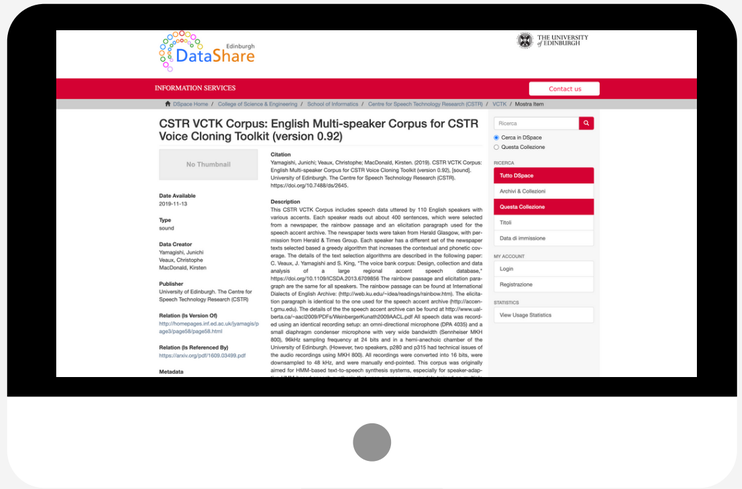

Super Resolution
Audio
Images
CNN 1-D
CNN 2-D
GAN
Single-Speaker
Multi-Speaker
YCbCr
RGB
Mean squared Error
Content Loss
Perceptual Loss

Audio Dataset: CSTR-VCTK*

It includes phrases spoken by 110 English speakers with different accents. Each speaker reads about 400 sentences selected from a newspaper. The sentences were chosen to maximize phonetic coverage. Each sentence was recorded using 2 different microphones.

- In the single speaker dataset, only one speaker was used (approximately 800 audio files).
- In the multi-speaker dataset, data from only 4 different speakers were used (approximately 3000 audio files) due to computational limits.

The audio files were divided into patches, or portions of the file. Depending on the value of the stride, the patches can be longer or shorter and overlap between different audio files. Selecting a very low stride value increases the dataset's dimensionality, as the same parts of the audio will be included in multiple patches. Conversely, a stride equal to the audio's size means that each patch corresponds to an audio file.



*https://datashare.ed.ac.uk/handle/10283/3443

Architettura Audio

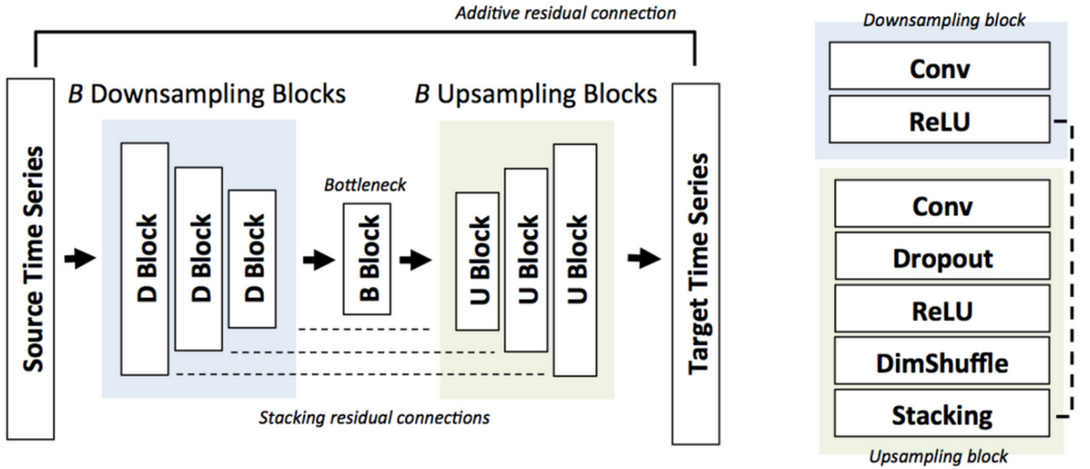
The architecture adopted is a U-net*, consisting of a series of down-sampling layers followed by a series of up-sampling layers, which are connected through skip connections. The composition follows that of the reference paper, but more layers have been implemented, providing a more complex network.

At each iteration, the network is provided with pairs of audio signals composed of the ground truth (high resolution) and its x4 down-sampled version which is then interpolated.

The monitored metrics are Mean Squared Error (MSE) and Signal to Noise Ratio (SNR).

The dataset has been divided into batches composed of patches and iterated 3 times per batch with a validation split of 0.10.

In the case of multi-speaker, the test set consists of audio from speakers who were not seen during training.



*AUDIO SUPER-RESOLUTION USING NEURAL NETS, Volodymyr Kuleshov, S. Zayd Enam, and Stefano Ermon. ICLR 2017. https://arxiv.org/pdf/1708.00853.pdf

Results: Audio

Cubic B-Spline

Single Speaker --> SNR = 14.80
Multi Speaker --> SNR = 13.0

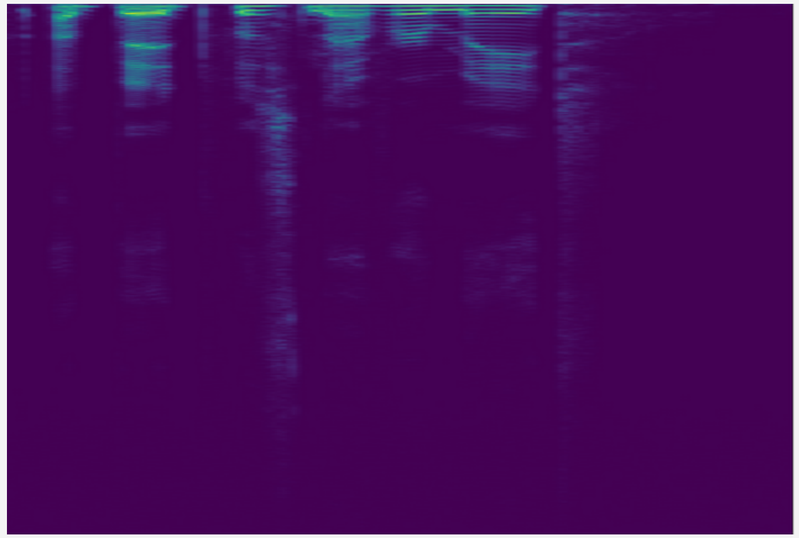
Single Speaker SR

Stride = 16 --> SNR = 16.75 | MSE = 0.35
Stride = 64 --> SNR = 17.13 | MSE = 0.34
Stride = 256 --> SNR = 16.70 | MSE = 0.35

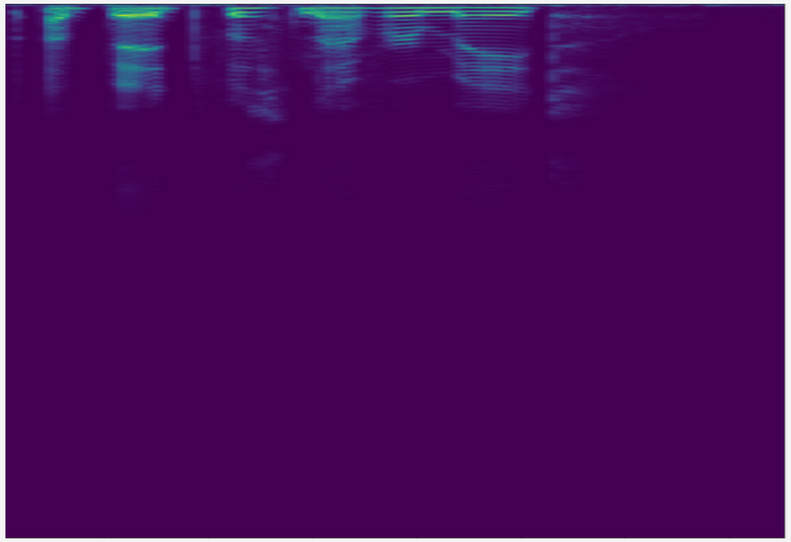
Multi Speaker SR

Stride = 16 not possible due to high dimensionality
Stride = 64 --> SNR = 19.22 | MSE = 1.82
Stride = 256 --> SNR = 19.38 | MSE = 1.74

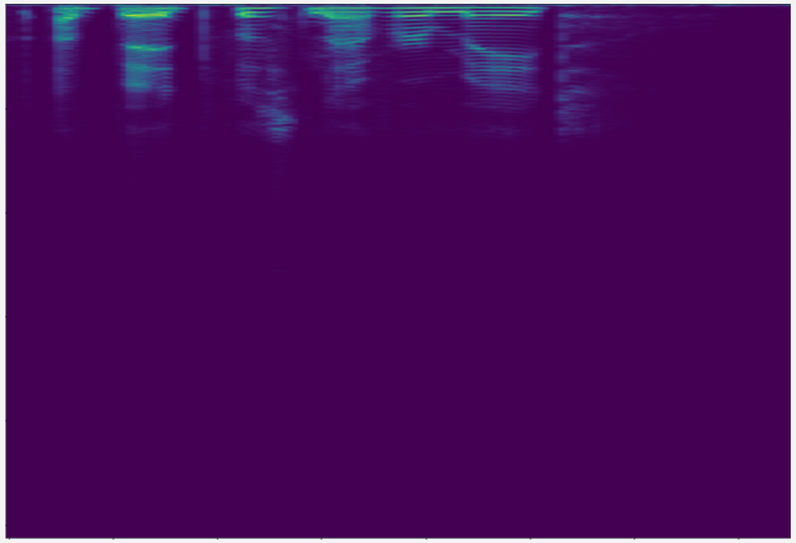
Results: Single-Speaker Spectrogram



Original



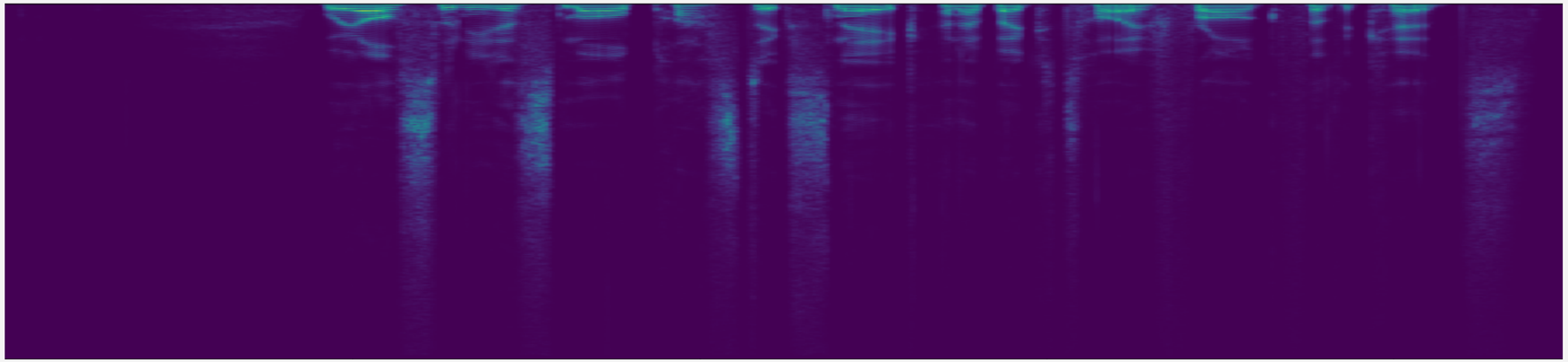
Down-sampled+interpolation



Super Resolution

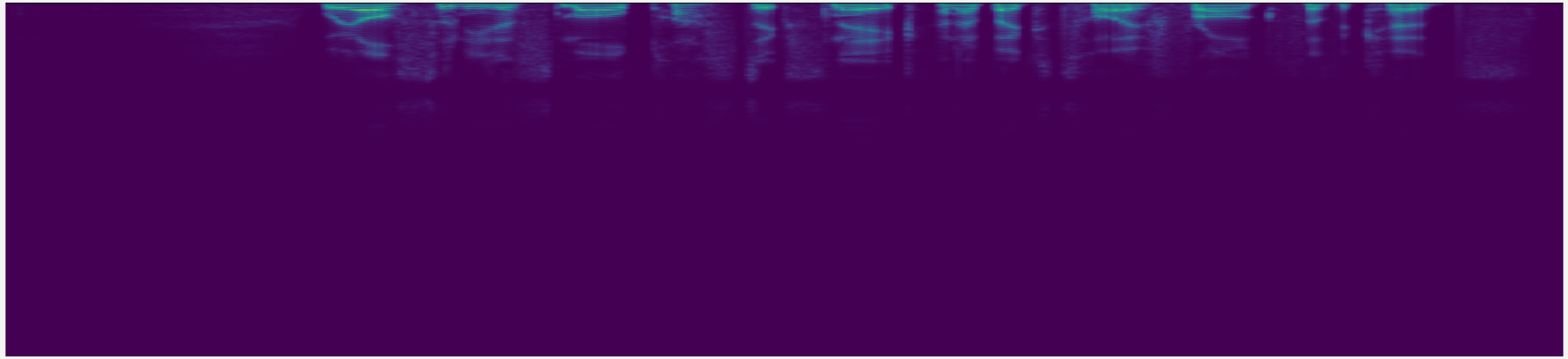
It can be observed that our model is not able to recover frequency bands which are lost during the down-sampling, but it acts on the power of the spectral density of the signal in low-resolution.

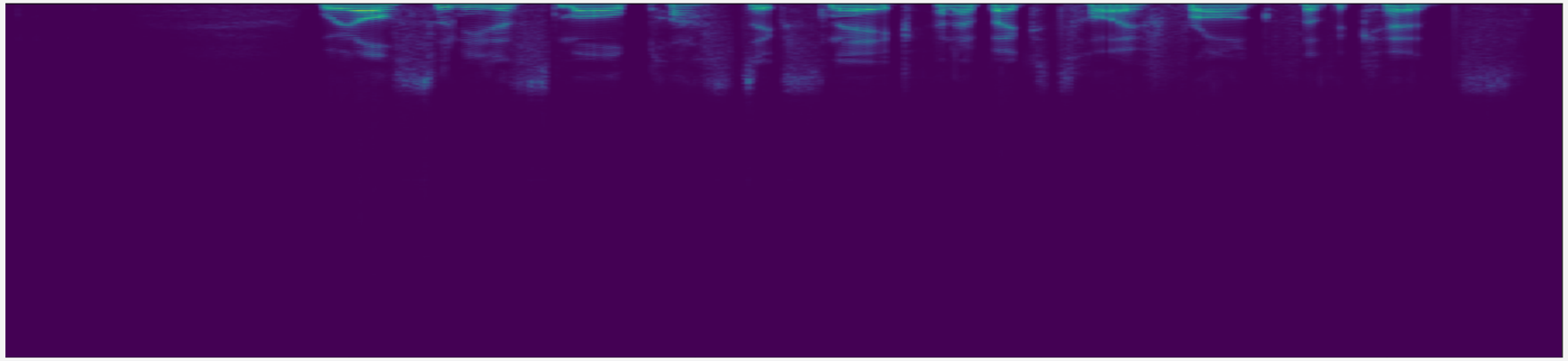
Results: Multi-Speaker Spectrogram



Original

Downsampled+interpolation





Super Resolution

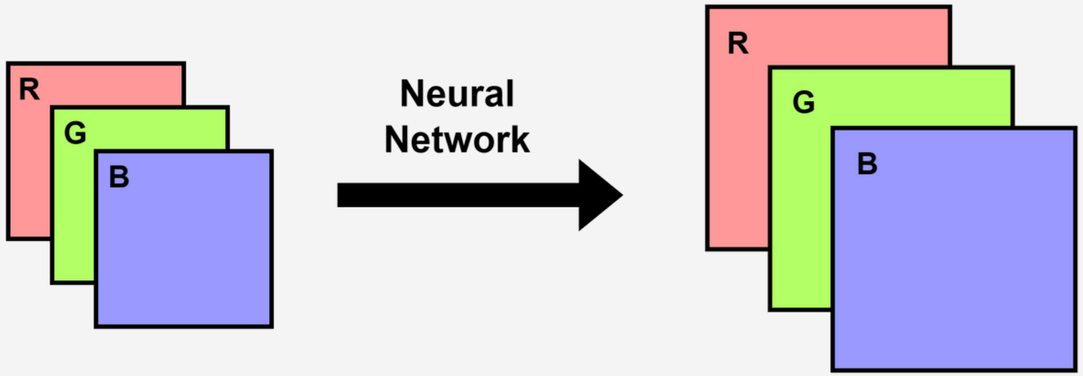
Dataset: Outdoor Scene

- OutdoorScene is a dataset of outdoor scenes composed of 10624 images. There are 7 categories of images: animals, buildings, sea, sky, grass, plants, mountains.

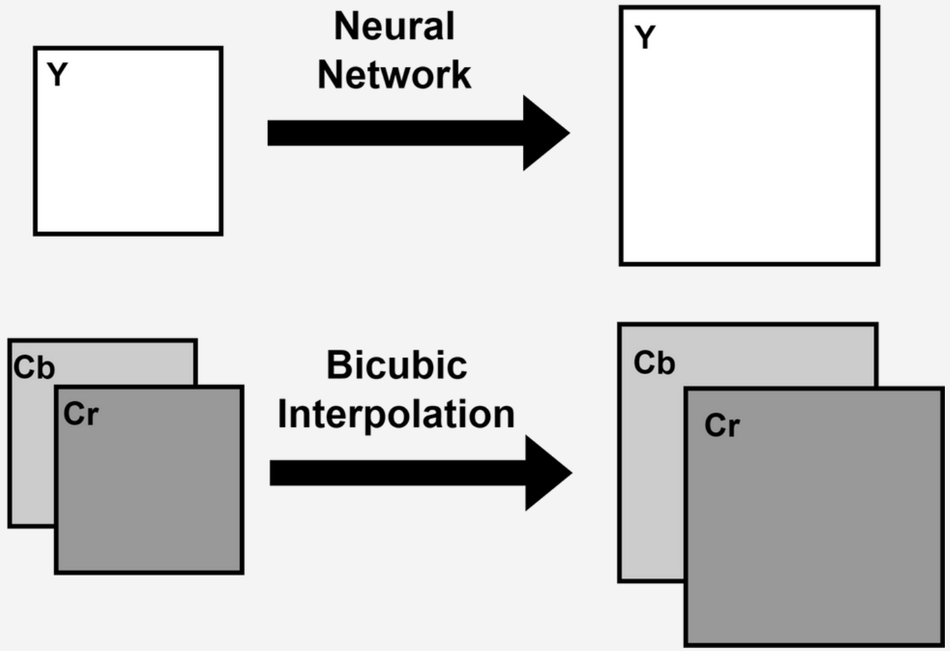
- For the CNN, 9408 images were used, dividing them into a training set (80%) and a validation set (20%), cropped to have batches of 8 images with a fixed size of 224x224.

- For the Generative Adversarial Networks, only 10% of the dataset was used, given the long training times.

Super-resolution on RGB and YCbCr



- The most recent studies work on RGB channels.
- The entire up-scaling process is handled by the network.



- The first studies on super-resolution used the YCbCr color space.
- The up-scaling of the Y channel is handled by the network, while Cb and Cr are upscaled using deterministic methods

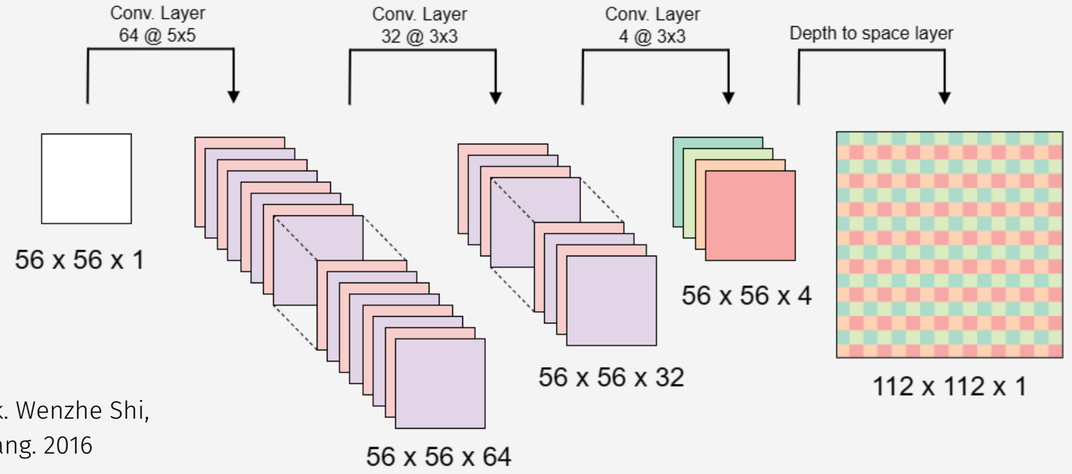
.

- They tend to be more efficient due to the smaller number of channels and parameters.

Efficient sub-pixel convolutional neural network

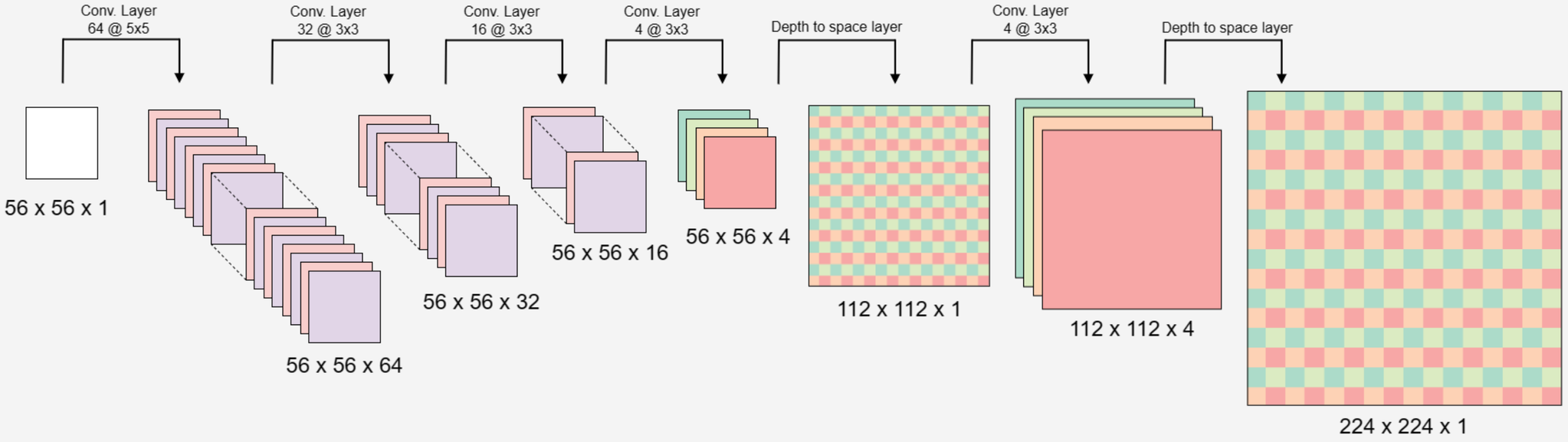
Efficient CNN proposta da W. Shi et al.*:

- Trained using MSE loss
- Tanh activation function is used for each layer
- Fully convolutional
- The filter size is the same for all the layers following the first and the image is only enlarged in the final layer



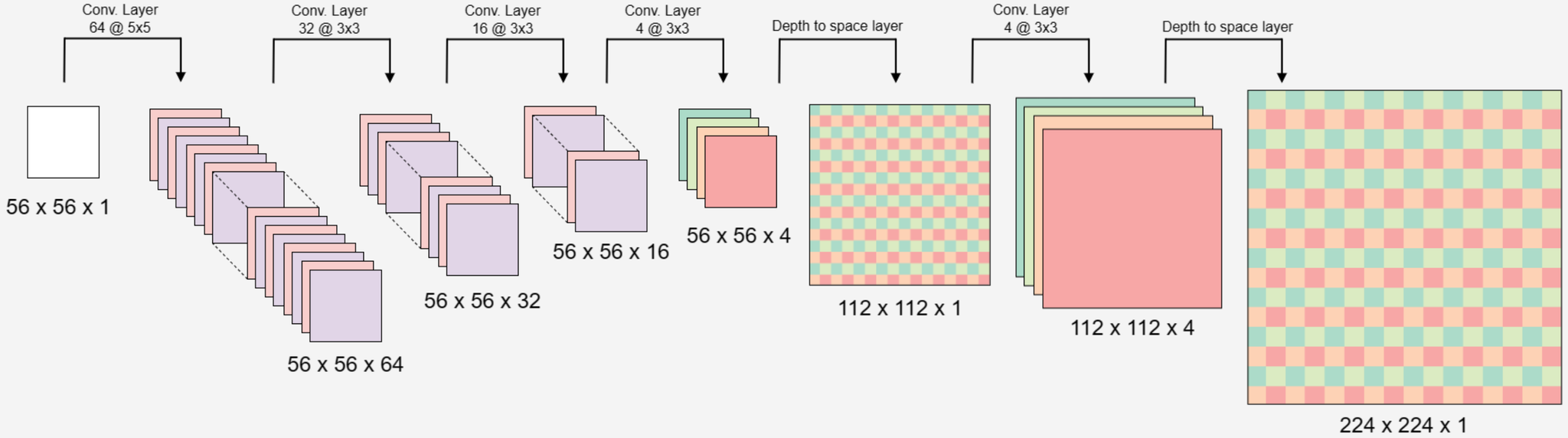
*Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. Wenzhe Shi, Jose Caballero , Ferenc Huszar´ , Johannes Totz , Andrew P. Aitken , Rob Bishop, Daniel Rueckert , Zehan Wang. 2016

La nostra CNN:



Efficient sub-pixel convolutional neural network

La nostra CNN:



- Trained using both **MSE loss** and **Content Loss**
- **N. of parameters**: 25362 for single-channel images and 30028 for three-channel images
- **Tanh** activation function is used for each layer
- **Fully convolutional**
- The upscaling is performed in two separate steps

- Each type of model was trained for 50 epochs.
- Approximately 30 seconds per epoch for the networks trained with MSE loss, and approximately 240 seconds per epoch for the networks trained with Content loss*

*training performed using Colab's Tesla T4 GPU

Loss functions

Mean Squared Error (MSE):

$$l_{MSE}^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y})^2$$

Content Loss:

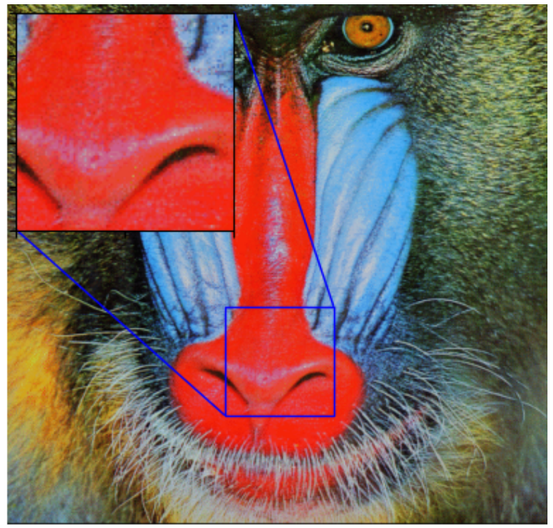
$$l_{VGG/i.j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

Performance metric

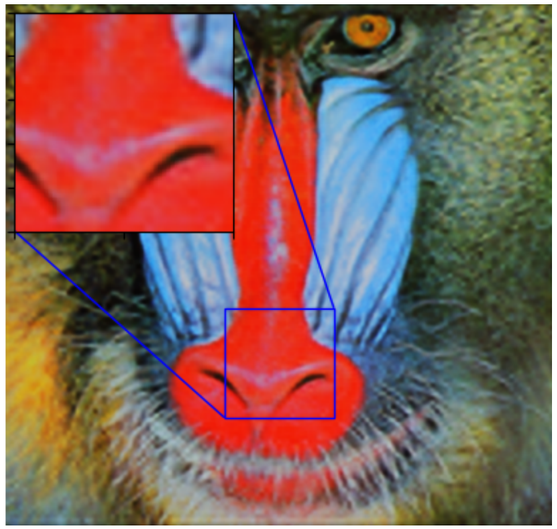
Peak Signal to Noise Ratio (PSNR):

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{dB}$$

Ground truth

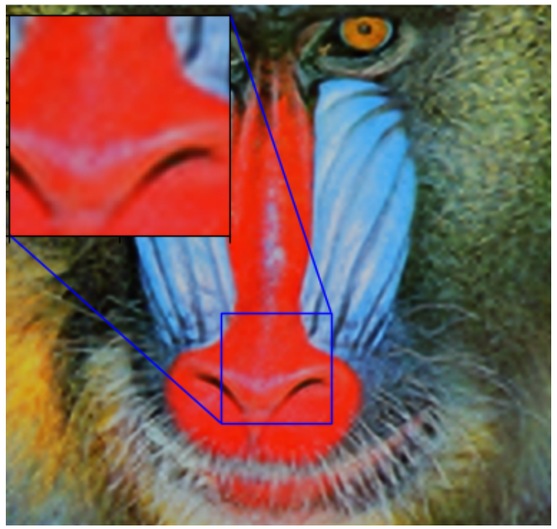


RGB MeanSquaredError



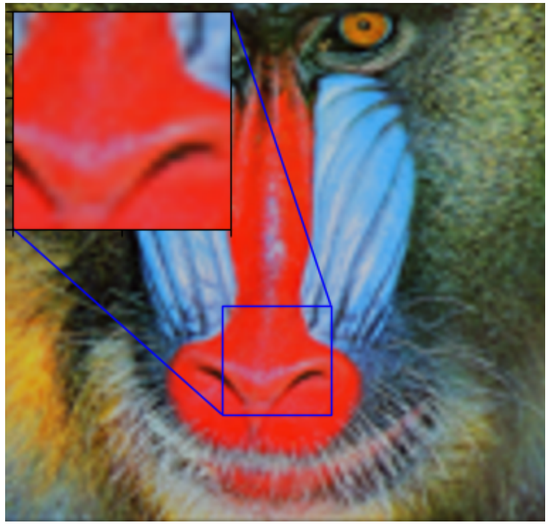
PSNR: 20.51

YCbCr MeanSquaredError



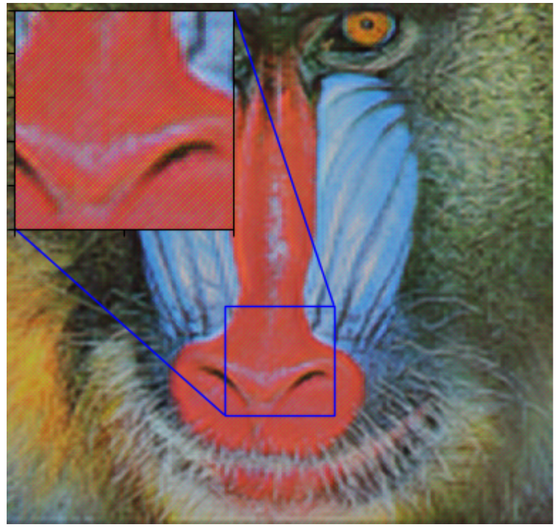
PSNR: 20.44

Bicubic interpolation



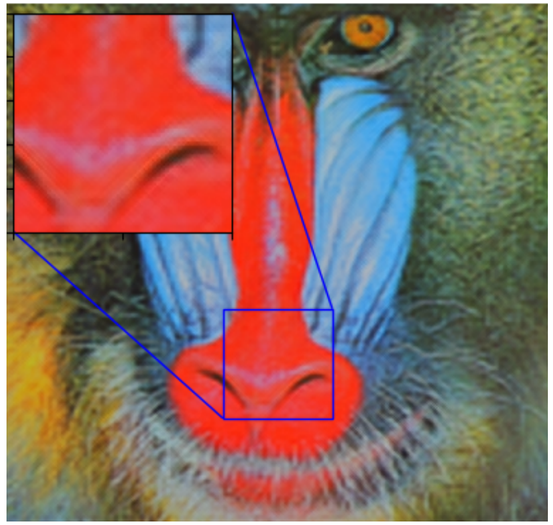
PSNR: 20.23

RGB Content Loss



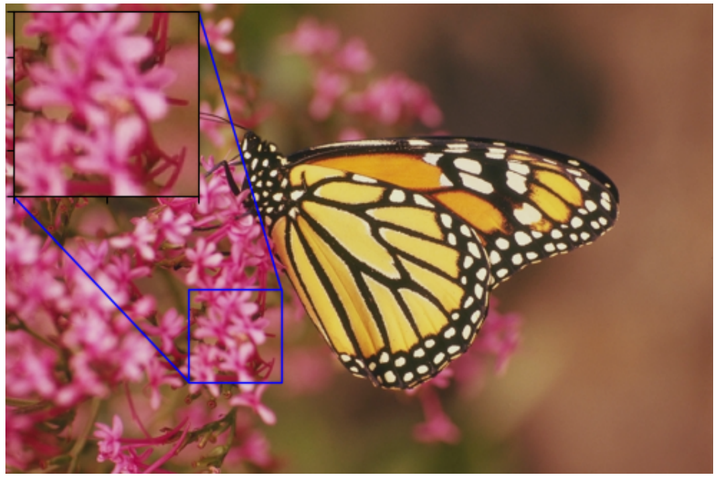
PSNR: 17.25

YCbCr Content Loss

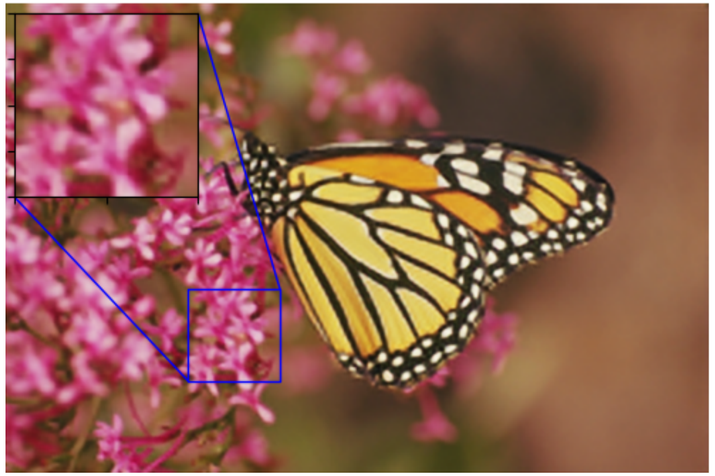


PSNR: 19.48

Ground truth

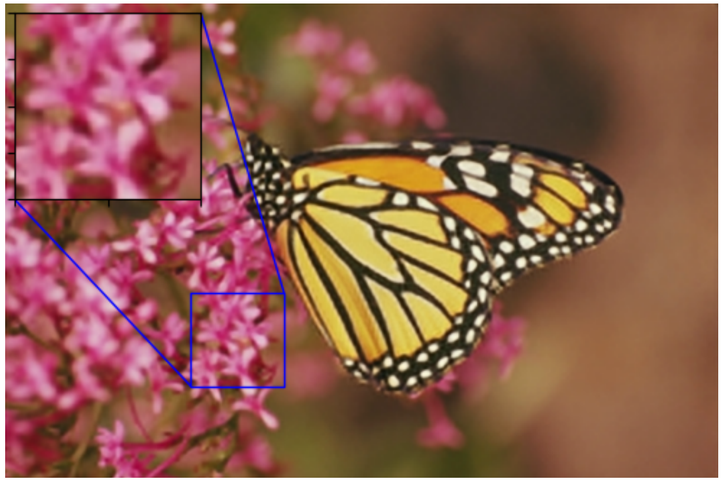


RGB MeanSquaredError



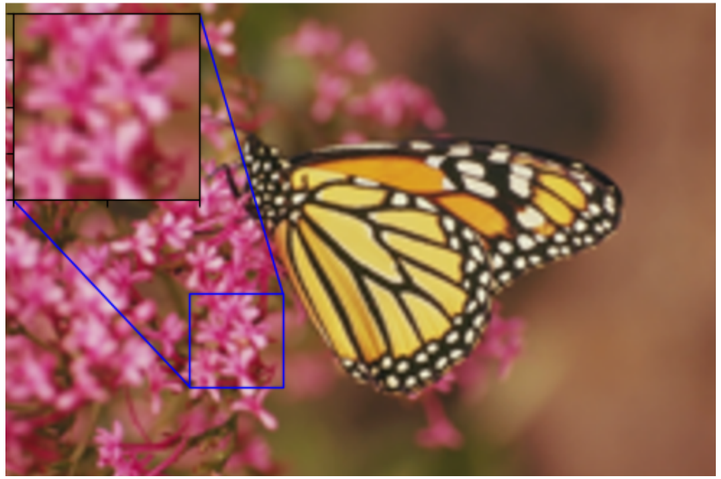
PSNR: 27.91

YCbCr MeanSquaredError



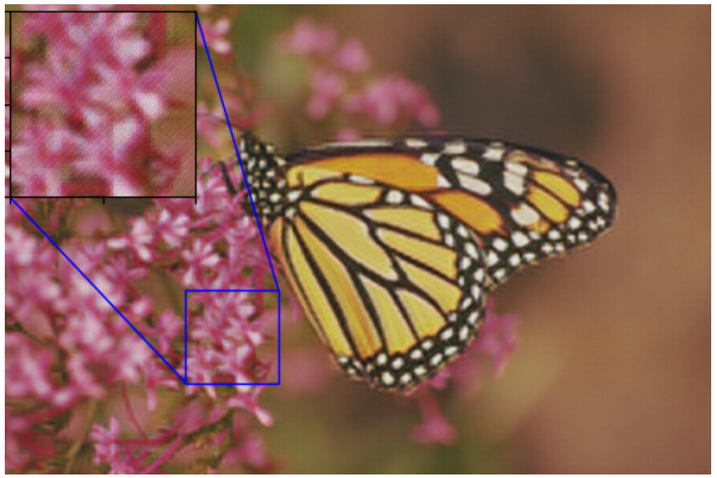
PSNR: 27.85

Bicubic interpolation



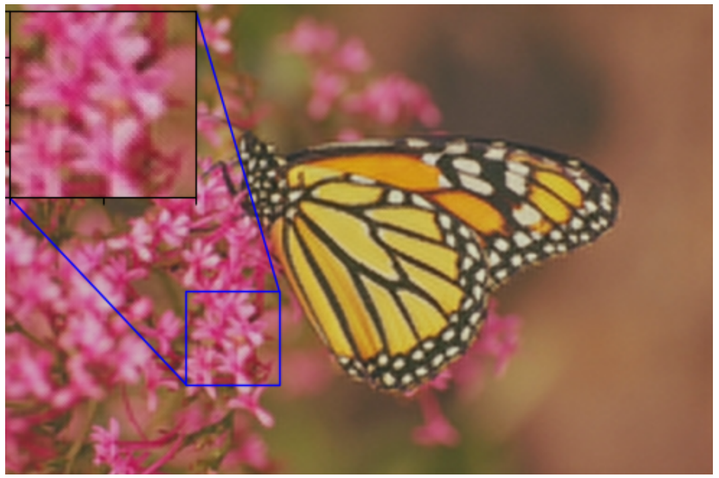
PSNR: 26.24

RGB Content Loss



PSNR: 19.89

YCbCr Content Loss



PSNR: 23.59

Ground truth

RGB MeanSquaredError

PSNR: 20.99

YCbCr MeanSquaredError

PSNR: 20.92

Bicubic interpolation

PSNR: 20.25

RGB Content Loss

PSNR: 17.19

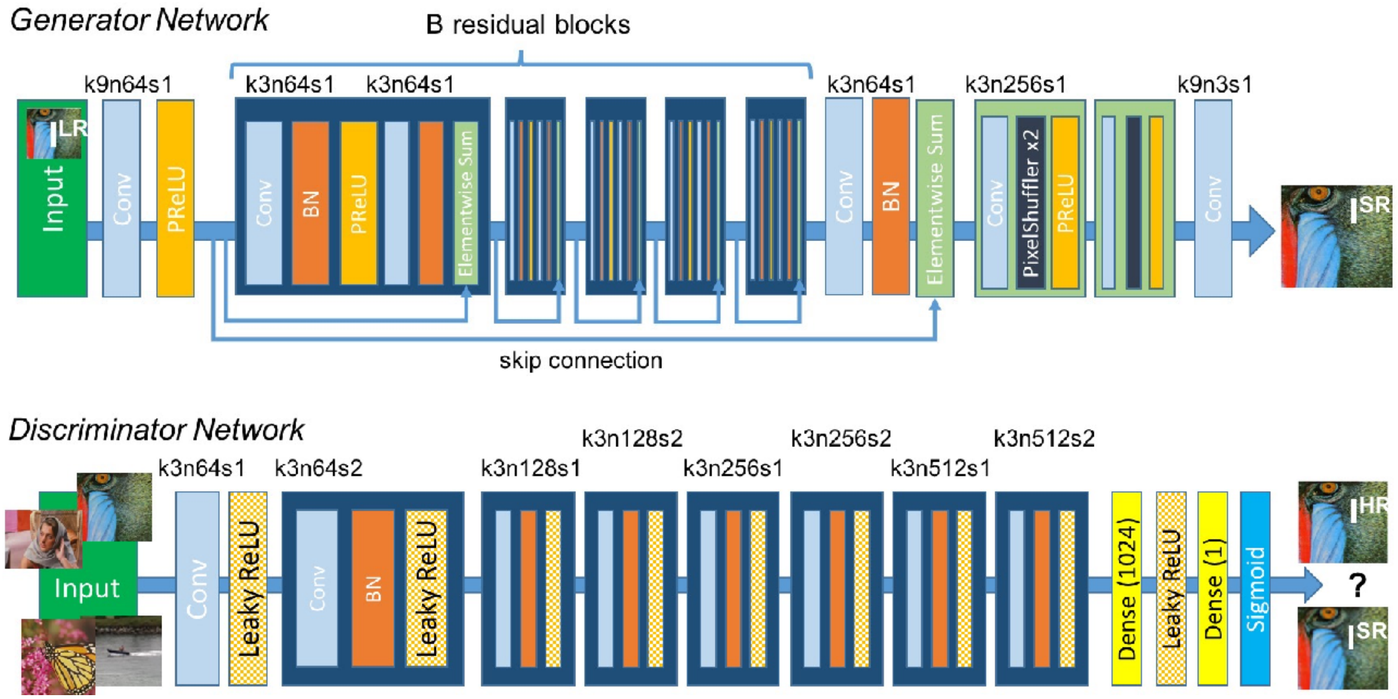
YCbCr Content Loss

PSNR: 18.87

GAN architecture proposed by C. Ledig et al.*

- Trained using **Perceptual Loss**
- **N. of parameters**: 1,554,883 (Generator) - 107,455,297 (Discriminator)
- **Fully convolutional Generator**
- The last two blocks of the generator perform two 2x up-scales.

- ~500 training epochs for both models (RGB and YCbCr)
- ~90 seconds per epoch



Perceptual Loss:

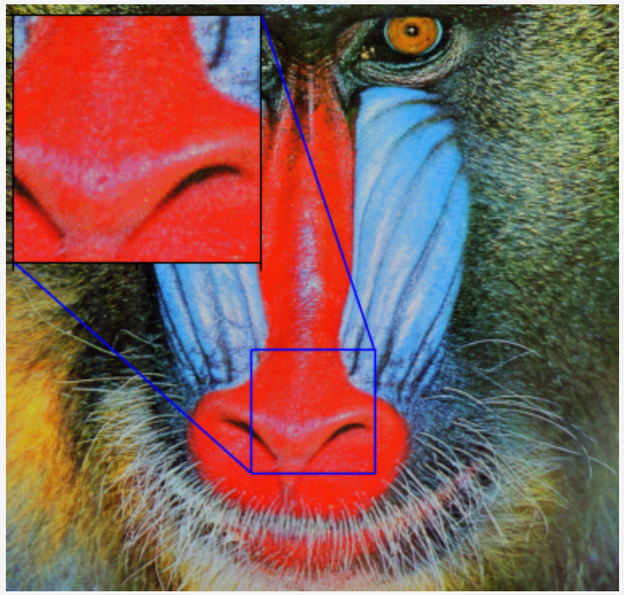
$$l^{SR} = \underbrace{l_{\mathrm{X}}^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}}$$

perceptual loss (for VGG based content losses)

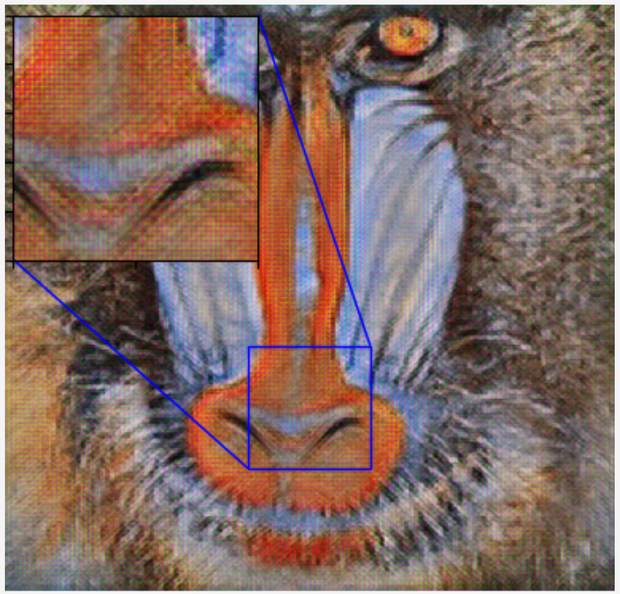
$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

*Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, ´ Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi. 2017.

SRGAN RGB Color Shift



Ground truth



SRGAN RGB

The SRGAN trained on RGB images produced appreciable super-resolution results, but a color shifting problem has arisen.

In particular, there seems to be a tendency for the model to generate images with sepia tones.

The first hypothesis was that the cause was a limited number of training epochs or a the small number of observations, but we verified that this was not the case.

By deepening the scientific literature and github threads on SRGAN, possible problems not reported in C. Ledig et al.'s paper were identified.

SRGAN RGB Color Shift

SRGAN: Training Dataset Matters

Nao Takano
nao.takano@ucdenver.edu
Computer Science and Engineering
University of Colorado Denver

Gita Alaghband
gita.alaghband@ucdenver.edu
Computer Science and Engineering
University of Colorado Denver

Figure 5: Converting Gray-Scale to Color (Dining Room)
Left: black and white, Center: original, Right: output.

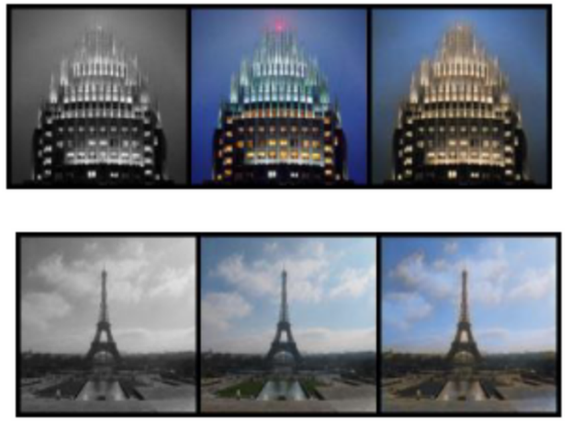


Figure 6: Converting Gray-Scale to Color (Tower)
Left: black and white, Center: original, Right: output.

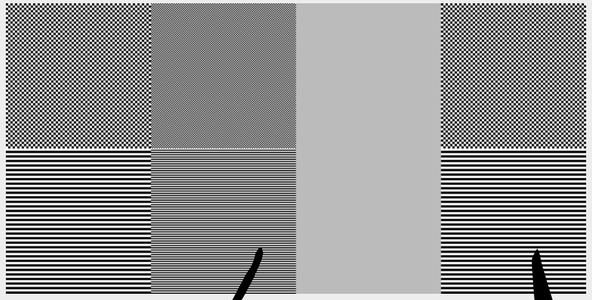
5.1.2. Dining Room

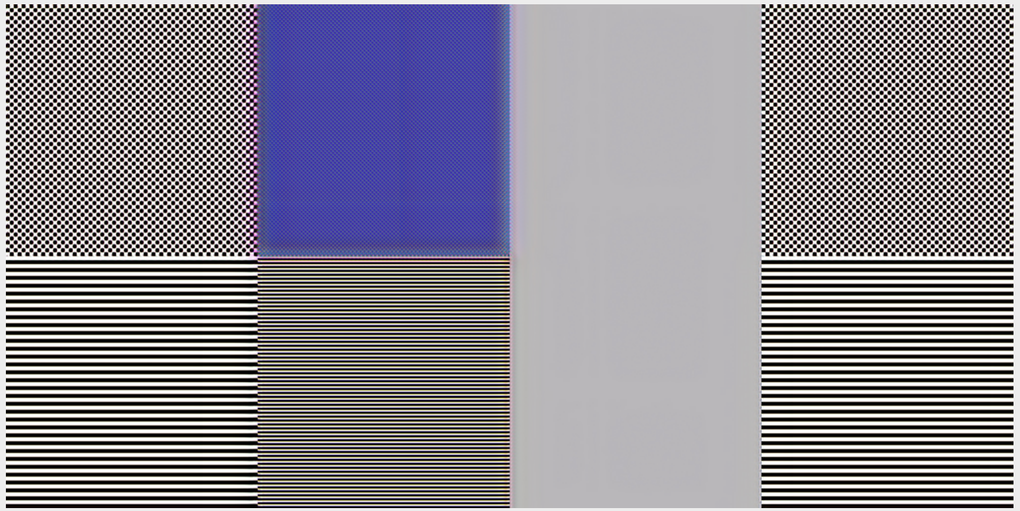
Figure 5 shows the result of our experiment for dining room coloring. Many training images contain tables and chairs, and very often if the color of the table is white, the chairs are brown, and vice versa. Two examples of those color combinations are presented. It appears the network detects the texture of materials to determine the color of furniture.

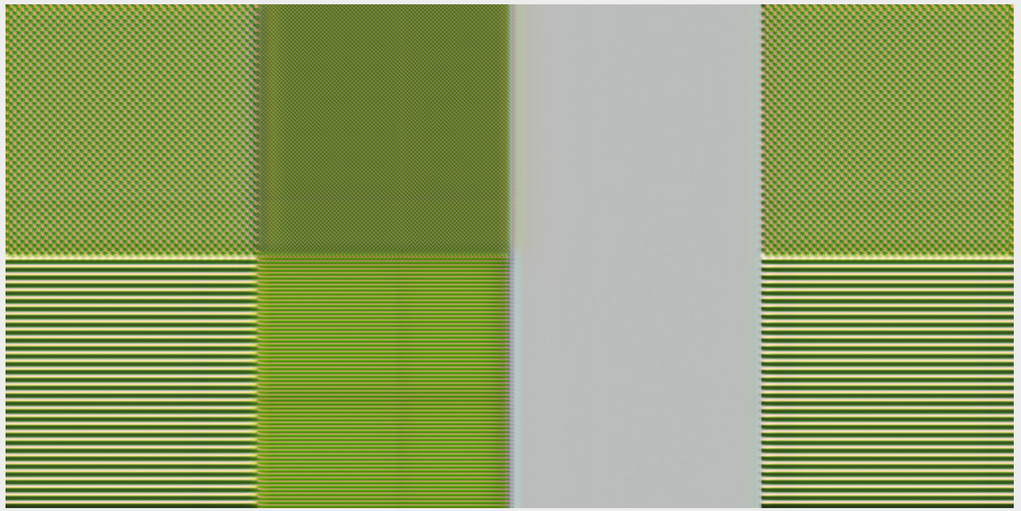
5.1.3. Tower

As shown in Figure 6, the buildings in the dataset do not exhibit a common color, but other elements, such as the sky or illumination of the building at night, will be regenerated in the output, which indicates that the network picks up colors of the most common denominator found in the training images.

SRGAN RGB Color Shift







Closed **Color shift with srgan model** #30

fjallraven opened this issue on Oct 20, 2019 · 7 comments

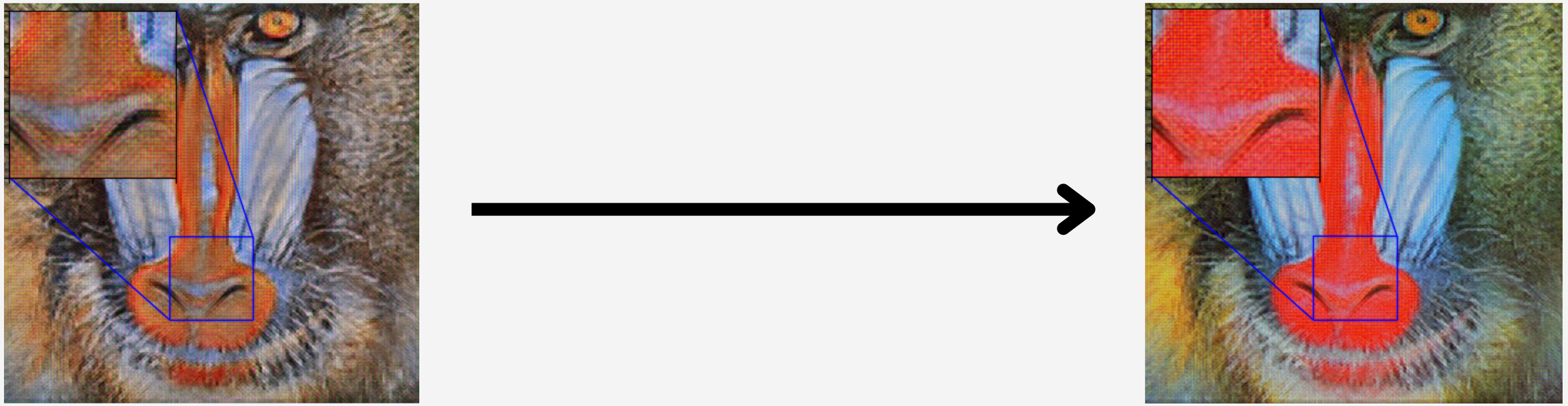
devernay commented on Sep 18, 2020 • edited

I had the same issue with https://github.com/Tencent/Real-SR so I guess that color shift is not (yet) considered a real issue in SR methods but the community (only PSNR takes into account color shift, SSIM and LPIPS consider it a minor disturbance).

Here's how I remove the color-shift from super-resolved images: downscale the SR image, compute the difference with the LR image, Gaussian blur, upscale, add to the SR image. All computation is done in linear color space.
This gives this one-line G'MIC script (edited to work also with older versions of gmic):

```
gmic image.jpg image_4x.jpg -srgb2rgb [1] -resize[1] 25%,25%,[0],[0],2 -sub[0,1] -blur[0] 2 -resize[0] 400%,400%,[0],[0],3 -
```

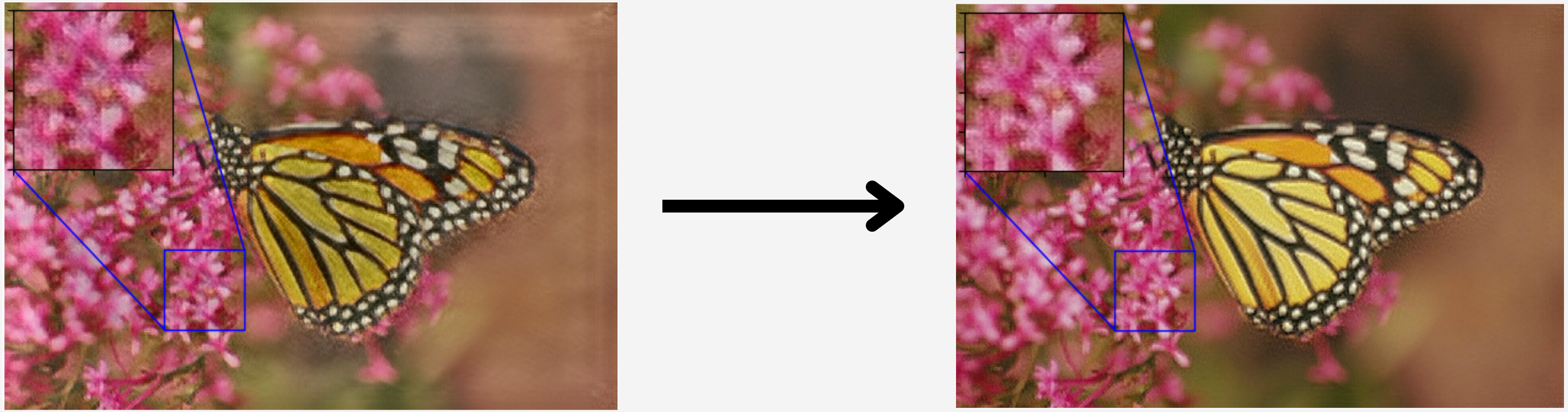
SRGAN RGB Color and Level Correction



To solve the problem of color shift, we perform a downscaling of the SR image, compute the difference with the LR image, and then upscale the difference matrix which is added to the SR image.

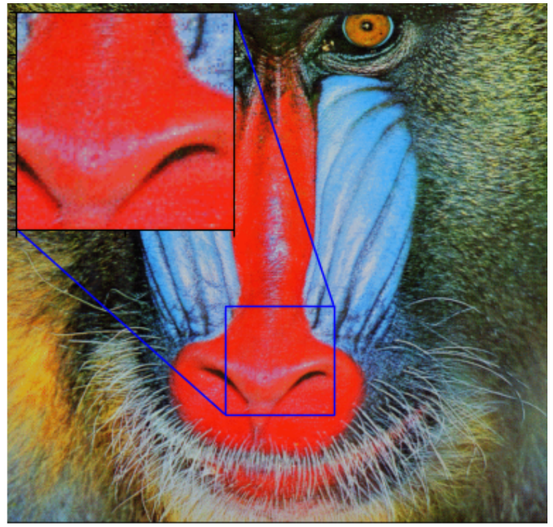
Then, we apply a transformation to all levels of brightness and contrast with gamma correction.

SRGAN YCbCr Luminance Correction

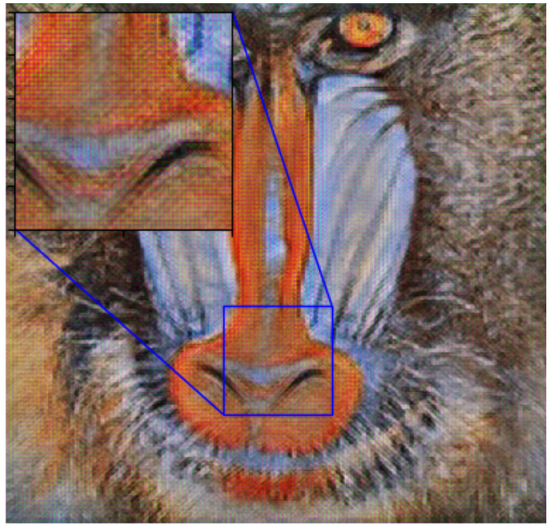


The same correction is applied to the luminance channel in the images generated by the YCbCr SRGAN.

Ground truth

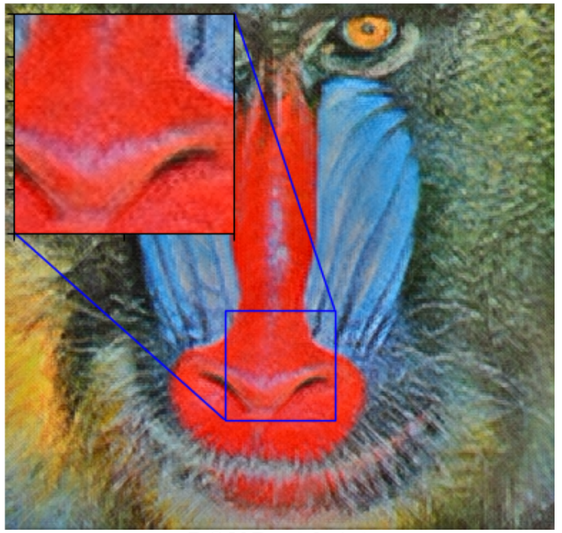


RGB GAN Perceptual Loss



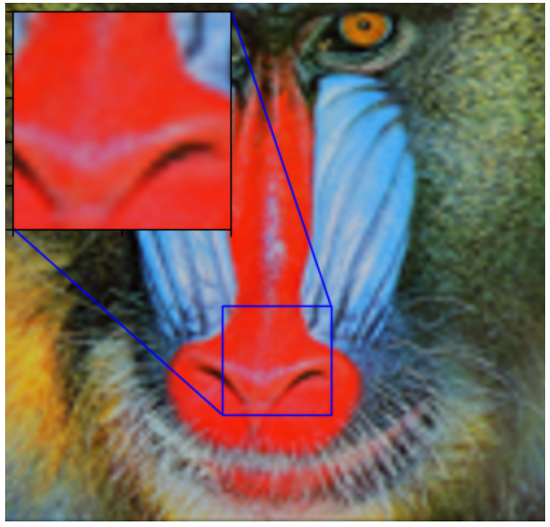
PSNR: 14.71

YCbCr GAN Perceptual Loss



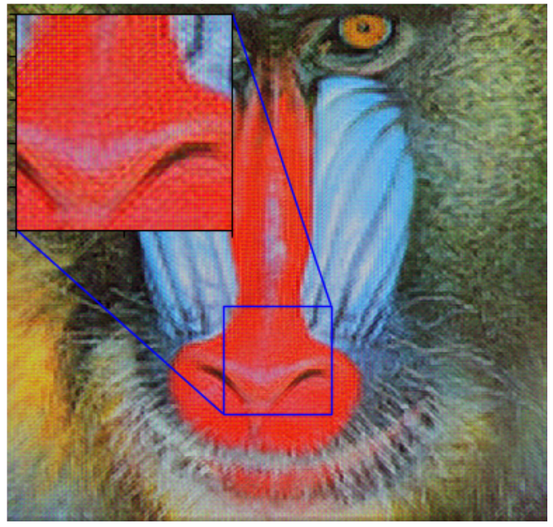
PSNR: 16.41

Bicubic interpolation



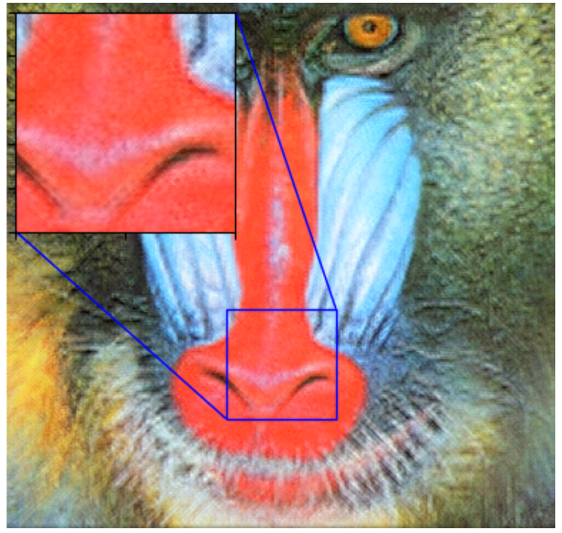
PSNR: 20.23

RGB GAN Perceptual Loss w/ correction



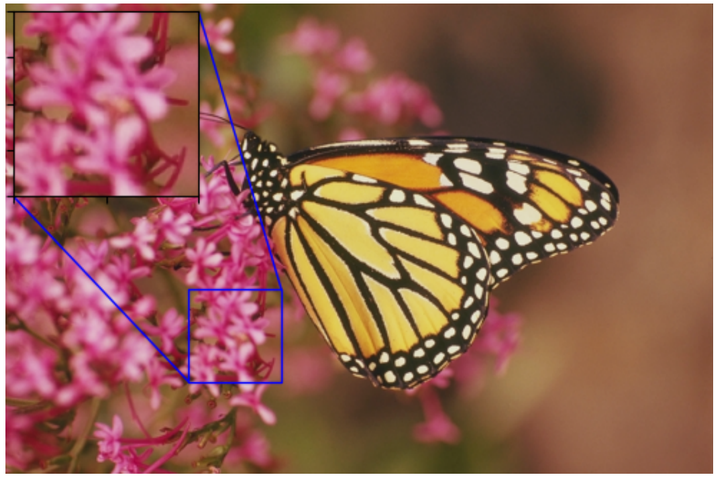
PSNR: 17.16

YCbCr GAN Perceptual Loss w/ correction

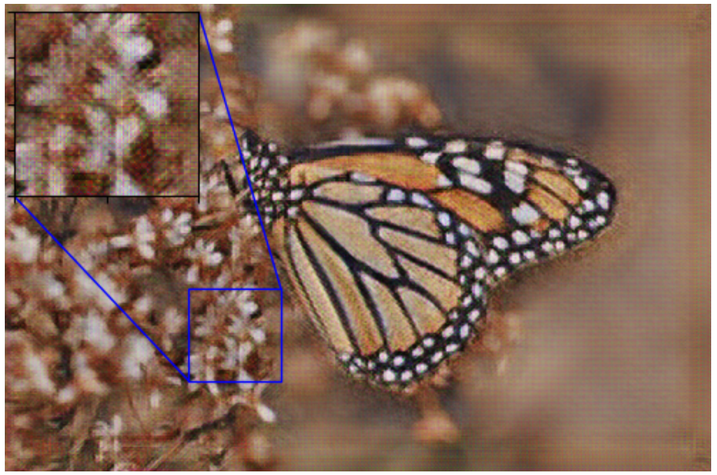


PSNR: 15.03

Ground truth

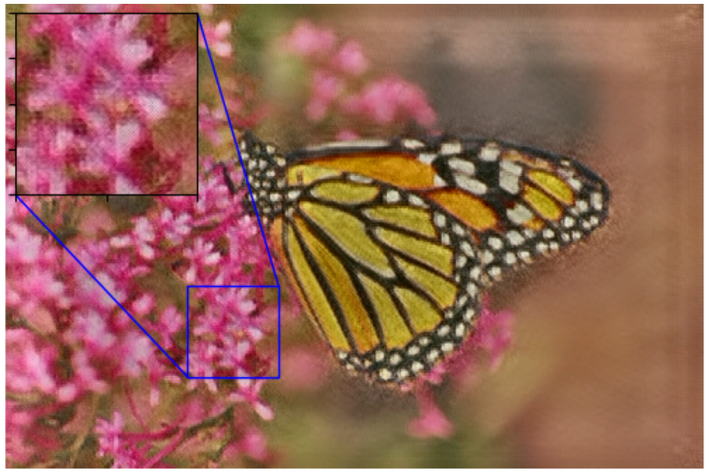


RGB GAN Perceptual Loss



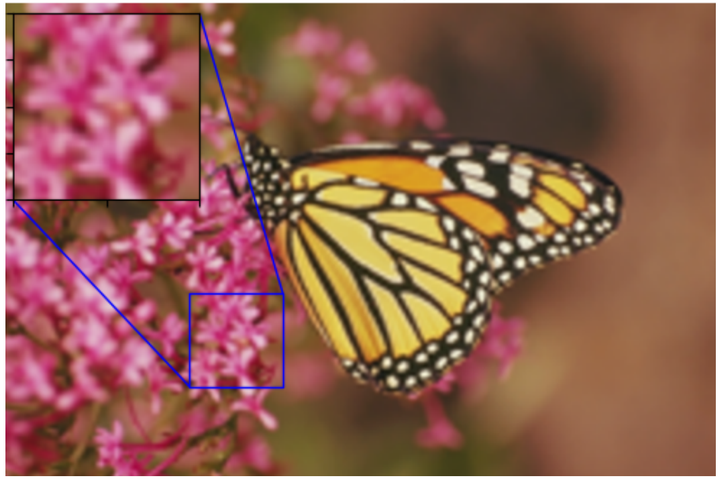
PSNR: 16.63

YCbCr GAN Perceptual Loss



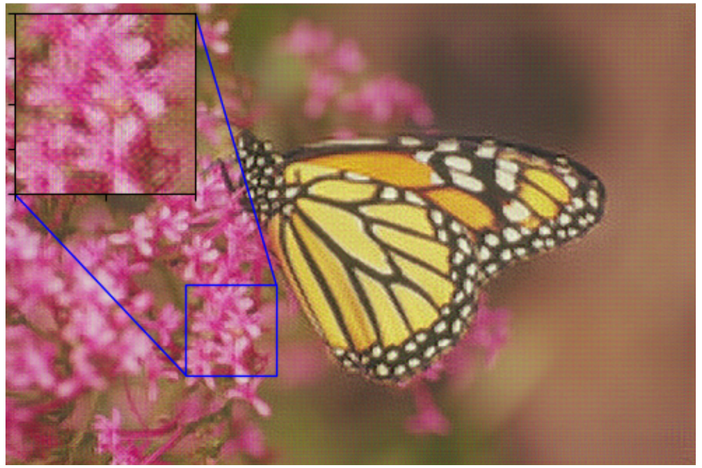
PSNR: 17.69

Bicubic interpolation



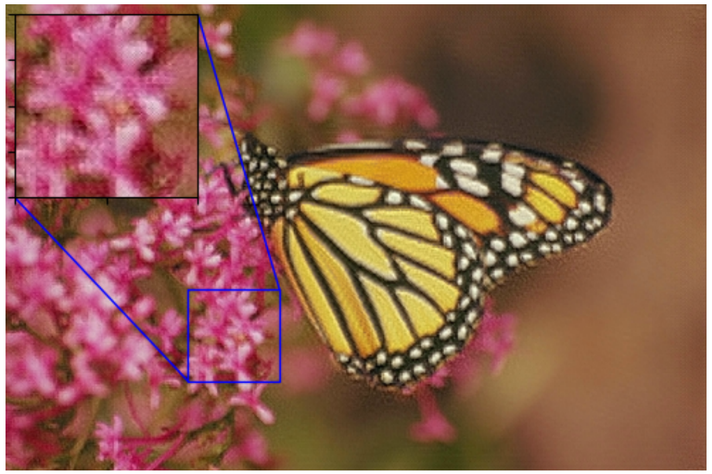
PSNR: 26.24

RGB GAN Perceptual Loss w/ correction



PSNR: 19.50

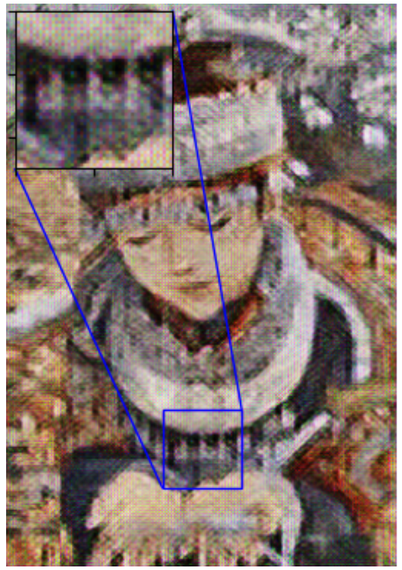
YCbCr GAN Perceptual Loss w/ correction



PSNR: 23.49

Ground truth

RGB GAN Perceptual Loss



PSNR: 14.26

YCbCr GAN Perceptual Loss

PSNR: 14.19

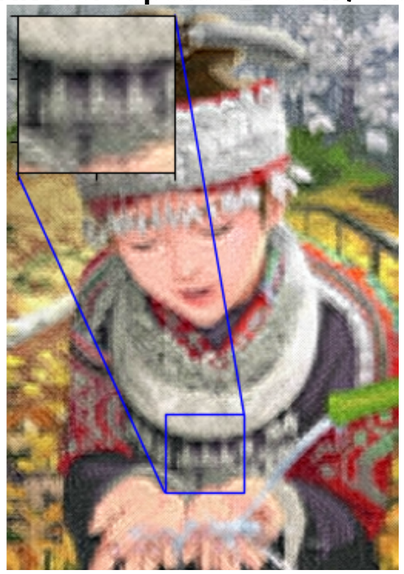
Bicubic interpolation

PSNR: 20.25

RGB GAN Perceptual Loss w/ correction

PSNR: 16.89

YCbCr GAN Perceptual Loss w/ correction



PSNR: 19.11

PSNR values:

	RGB	YCbCr
Bicubic interpolation	24.41	24.41
CNN (MSE)	24.93	24.83
CNN (Content Loss)	19.56	20.85
GAN (Perceptual Loss)	14.49	14.99
GAN (Perceptual Loss) w/ Color Correction	18.31	20.47

Conclusions

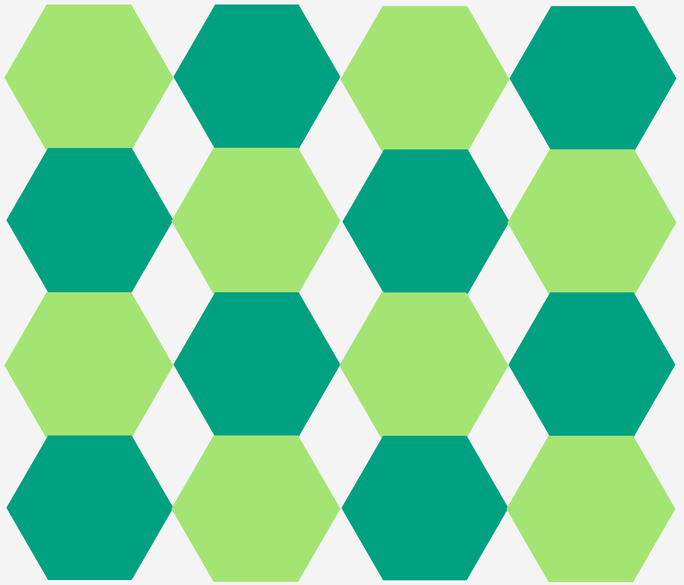
The use of the depth-to-space layer generates a noticeable checkerboard artifact, but the super-resolution is visually efficient.

The PSNR values seem to favor the deterministic method of Bicubic Interpolation, but the literature on the subject of super-resolution raises many doubts about this metric: user tests or other metrics more related to human perception are preferred.

PSNR is not only uncorrelated with human perception, but is also highly influenced by transformations such as mean shifting, rotations, noise, etc. (https://videoprocessing.ai/metrics/ways-of-cheating-on-popular-objective-metrics.html)

Digital Signal and
Image Management

Thanks for your attention!



Gaetano Chiriaco, Riccardo Porcedda, Gianmarco Russo