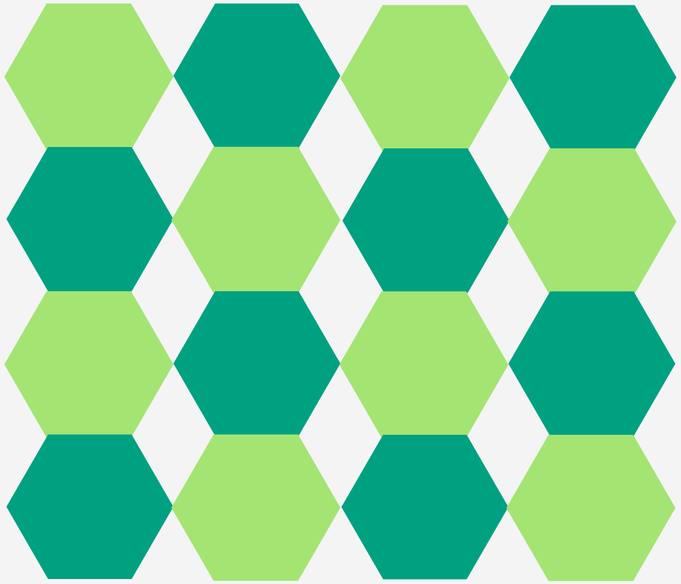
Digital Signal and
Image Management

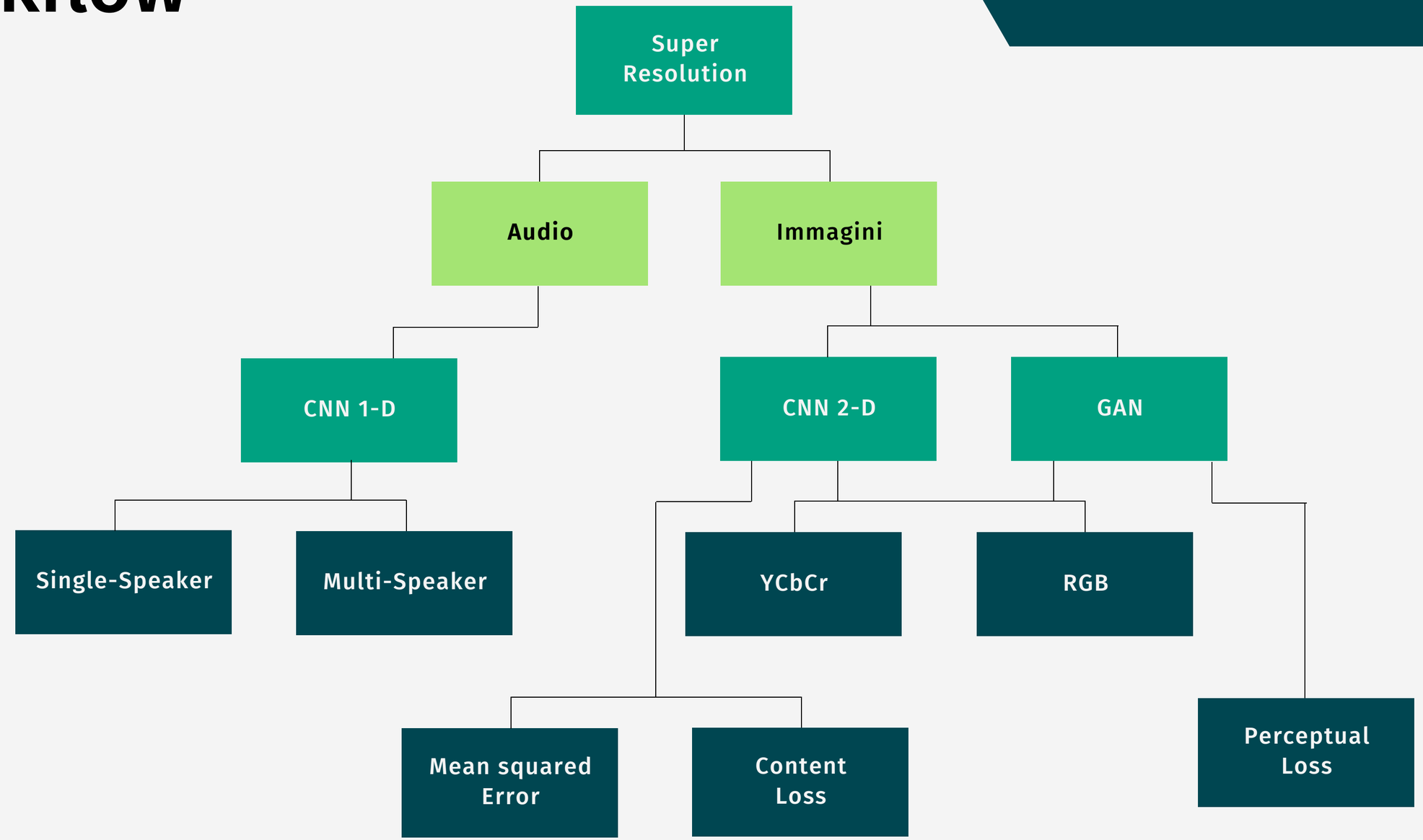
Super-Resolution di segnali 1D e 2D



Convolutional Neural Networks e Generative Adversarial Networks

Gaetano Chiriaco, Riccardo Porcedda, Gianmarco Russo

Workflow

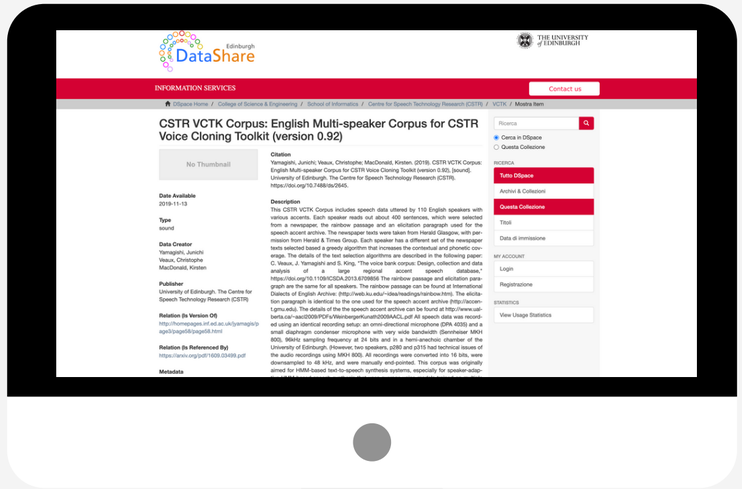

Super Resolution
Audio
Immagini
CNN 1-D
CNN 2-D
GAN
Single-Speaker
Multi-Speaker
YCbCr
RGB
Mean squared Error
Content Loss
Perceptual Loss

Dataset Audio: CSTR-VCTK*

Include frasi pronunciate da 110 English speakers con diversi accenti. Ogni speaker legge circa 400 frasi selezionate da un giornale. Le frasi sono state scelte per massimizzare la copertura fonetica. Ogni frase è stata registrate con 2 diversi microfoni.

- Nel single speaker è stato utilizzato solamente uno speaker (circa 800 file audio).
- Nel multi speaker sono stati utilizzati solo dati di 4 speaker diversi (circa 3000 file audio) a causa di limiti computazionali.

I file audio sono stati divisi in patches, ovvero porzioni di file. In base al valore dello stride le patches possono essere più o meno lunghe e sovrapporsi tra diversi file audio. Selezionare quindi uno stride molto basso significa aumentare la dimensionalità del dataset, in quanto stesse parti di audio verranno incluse in più patches. Al contrario invece, con stride pari alla dimensione dell'audio, ogni patch corrisponde ad un file audio.



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

Architettura Audio

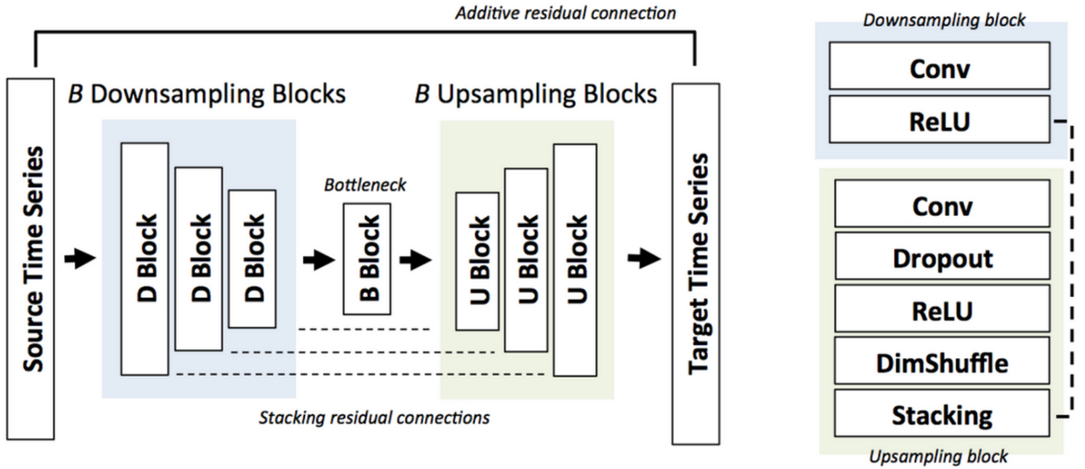
L'architettura utilizzata è di tipo U-net*, formata quindi da una serie di layer di down-sampling e successivamente una serie di up-sampling, questi layer sono collegati tra loro tramite skip connections. La composizione segue quella del paper di riferimento, ma sono stati implementati più layer, fornendo quindi una rete più complessa.

Ad ogni iterazione vengono fornite alla rete coppie di segnali audio composte dalla ground truth (alta risoluzione) e da una suo versione down-sampled x4 e poi interpolata.

Le metriche monitorate sono state il Mean Squared Error(MSE) e il Signal to Noise Ratio(SNR).

Il dataset è stato diviso in batch composti da patches e iterato 3 volte per ogni batch con validation split di 0.10.

Nel caso multi-speaker il test set è formato da audio di speaker mai visti duranti il training.



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

Risultati: Audio

Cubic B-Spline

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

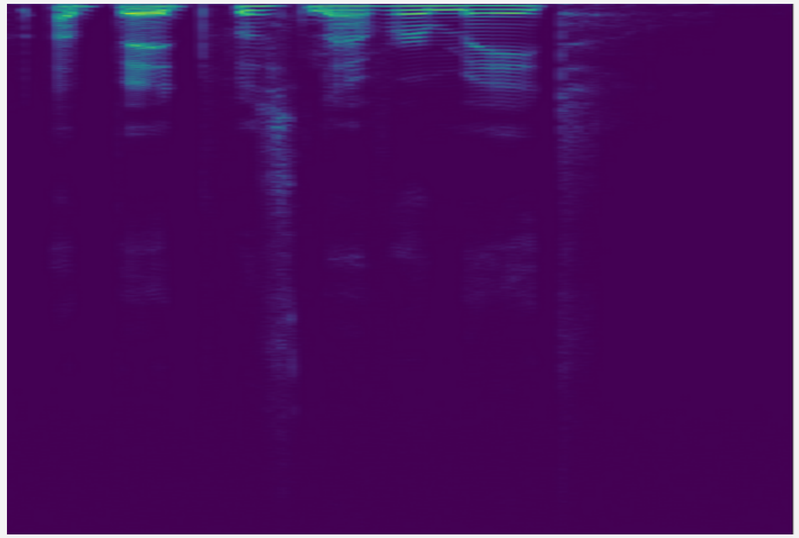
Single Speaker SR

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

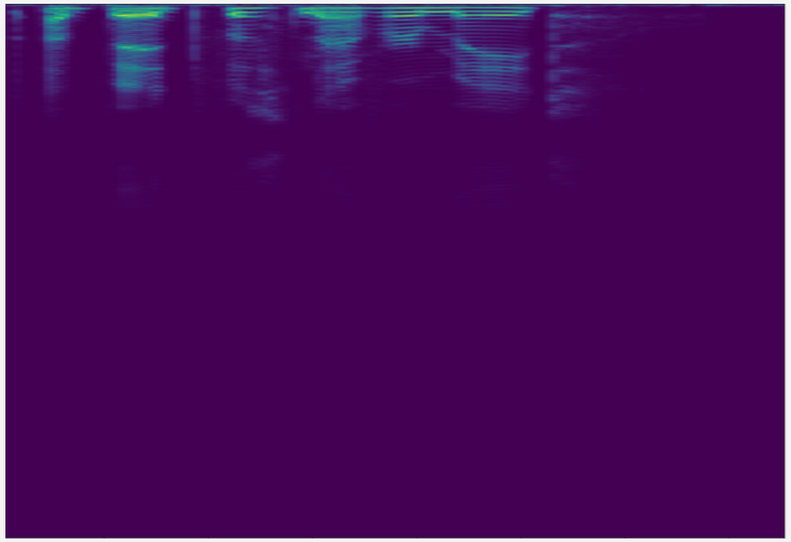
Multi Speaker SR

Stride = 16 non possibile a causa dell'eccessivo aumento della dimensionalità
Stride = 64 --> SNR = 19.22 | MSE = 1.82
Stride = 256 --> SNR = 19.38 | MSE = 1.74

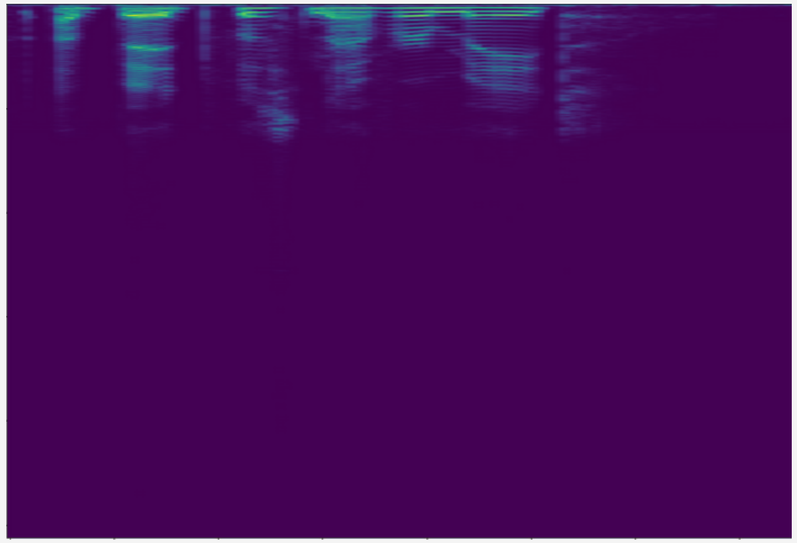
Risultati: Spettrogramma Single-Speaker



Originale

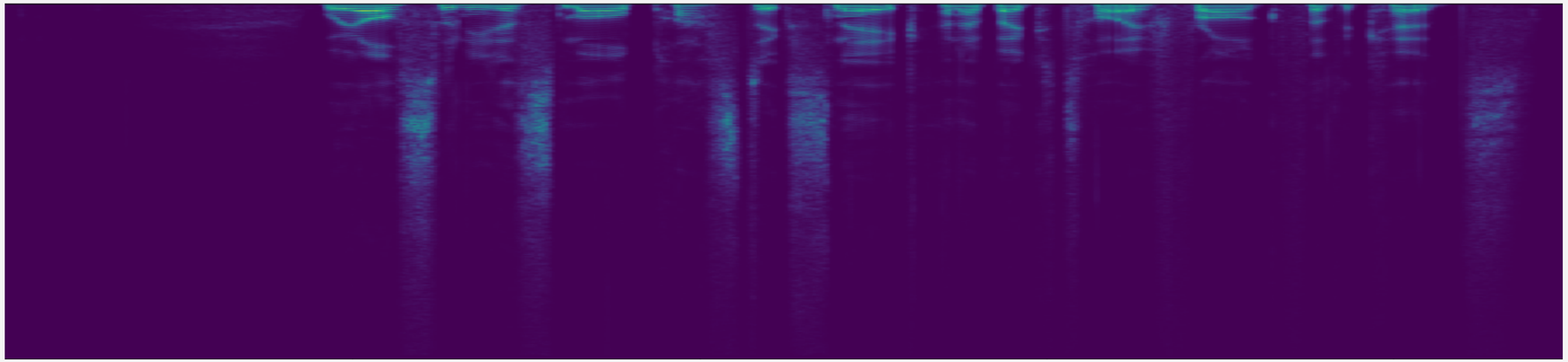


Down-sampled+interpolazione



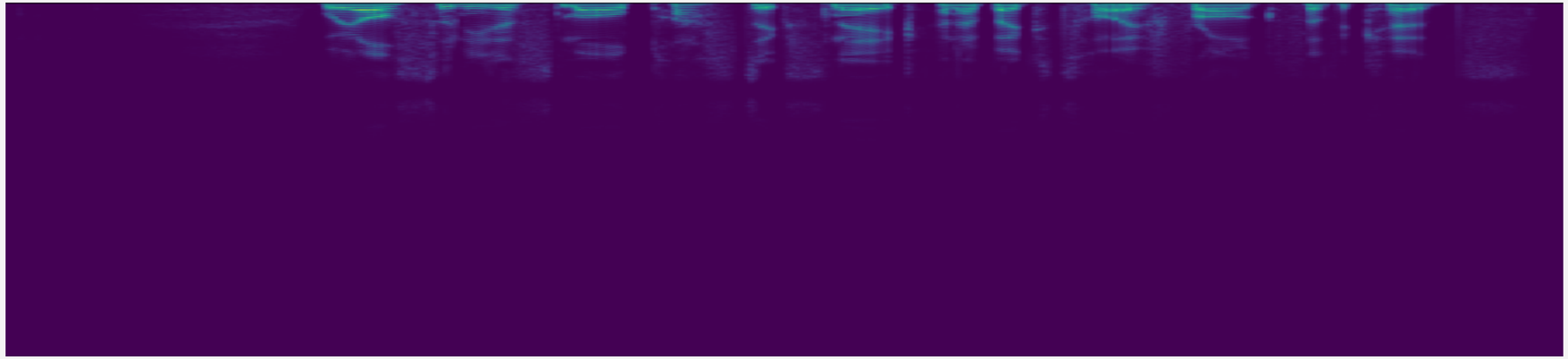
Super Resolution

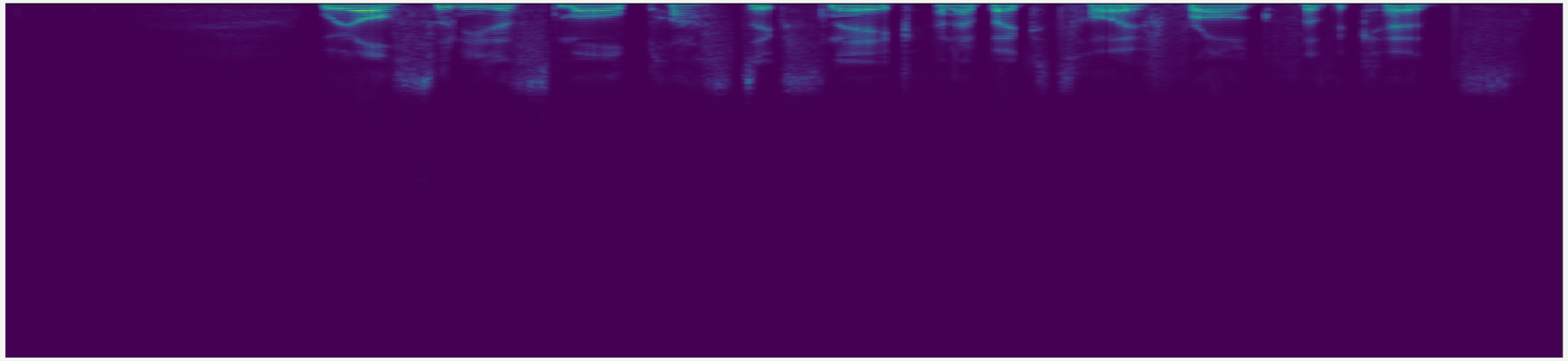
Risultati: Spettrogramma Multi-Speaker



Originale

Downsampled+interpolazione





Super Resolution

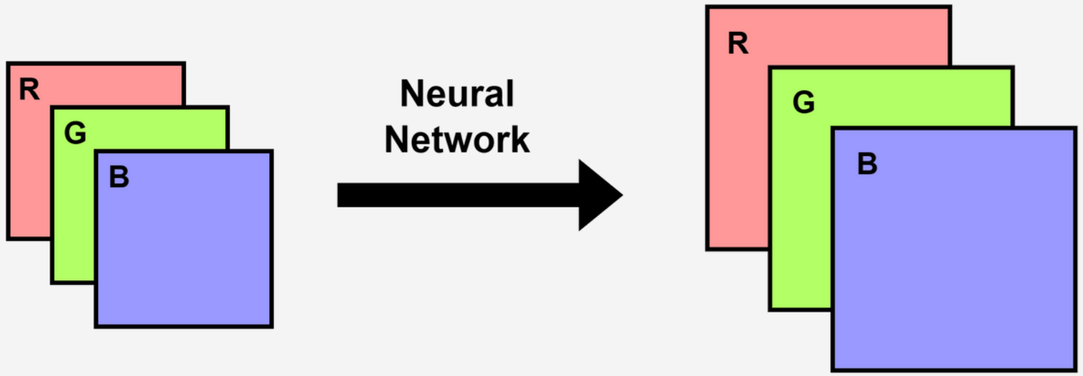
Dataset: Outdoor Scene

- OutdoorScene è un dataset di scene all'aperto composto da 10624 immagini. Sono presenti 7 categorie di immagini: animali, edifici, mare, cielo, erba, piante, montagne

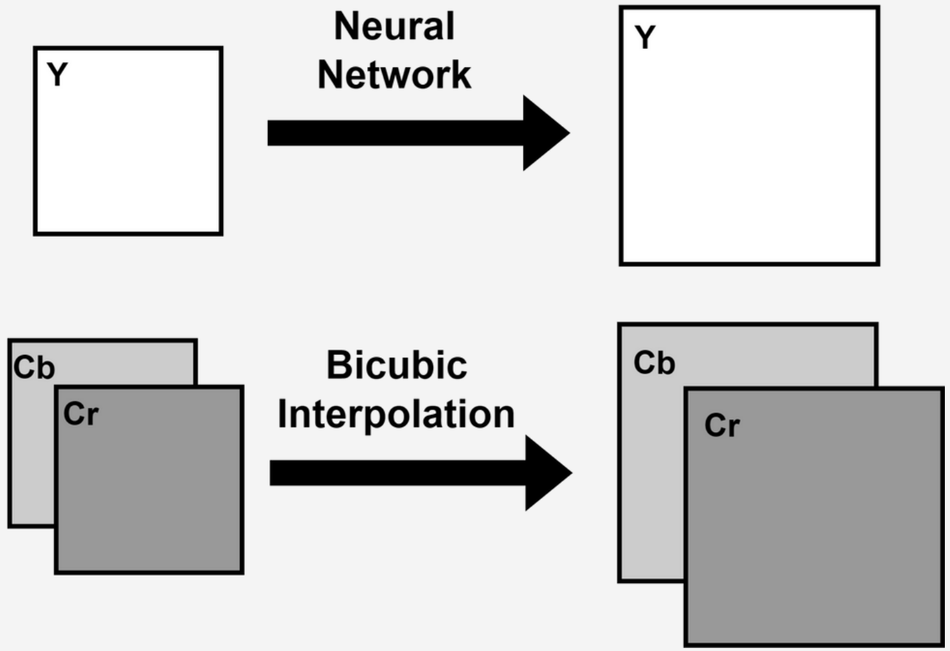
- Per le CNN sono state utilizzate 9408 immagini, dividendole in training set (80%) e validation set (20%), ritagliandole per avere batch di 8 immagini con dimensione fissa 224x224.

- Per le Generative Adversarial Networks è stato utilizzato solo il 10% del dataset, dati i lunghi tempi di addestramento

Super-resolution su RGB e YCbCr



- Gli studi più recenti lavorano sui canali RGB
- L'intero processo di up-scaling viene gestito dalla rete

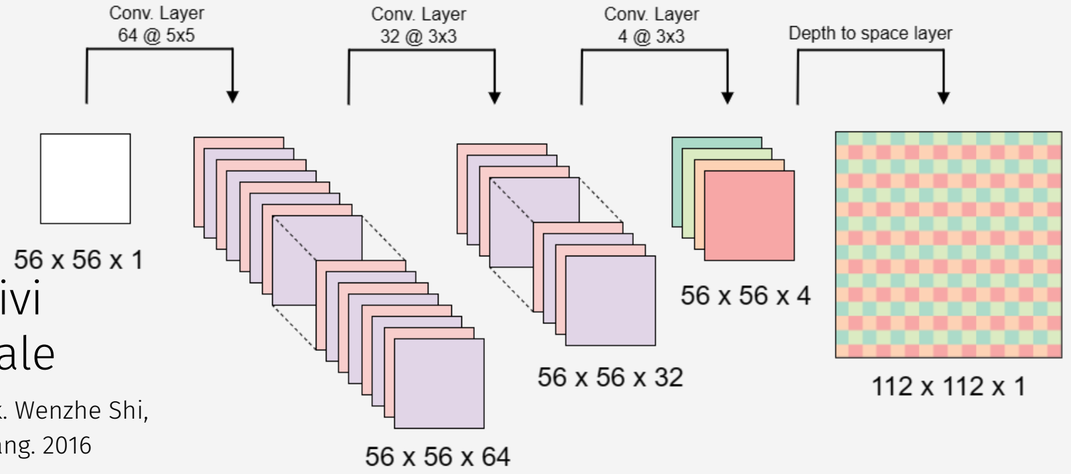


- I primi studi sulla super-resolution utilizzavano gli spazi colore YCbCr
- L'up-scaling del canale Y viene gestito dalla rete, mentre Cb e Cr vengono ingranditi utilizzando metodi deterministici
- Tendono ad essere più efficienti dato il minor numero di canali e parametri

Efficient sub-pixel convolutional neural network

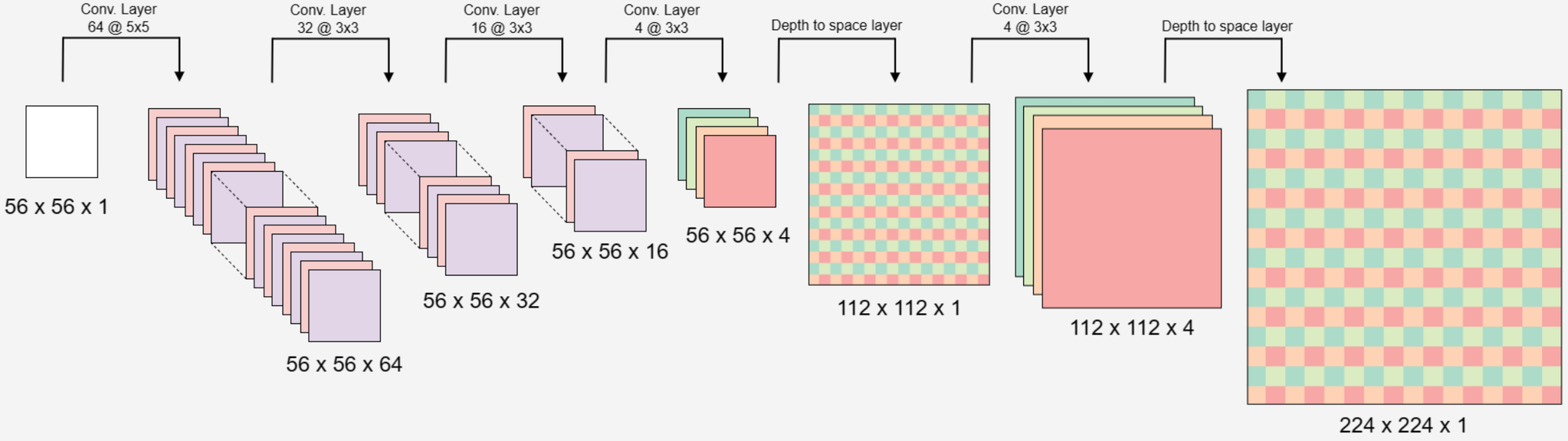
Efficient CNN proposta da W. Shi et al.*:

- Addestrata utilizzando la MSE loss
- Funzione di attivazione Tanh per ogni layer
- Fully convolutional
- La dimensione dei filtri è uguale per tutti i layer successivi al primo e l'immagine viene ingrandita solo nel layer finale



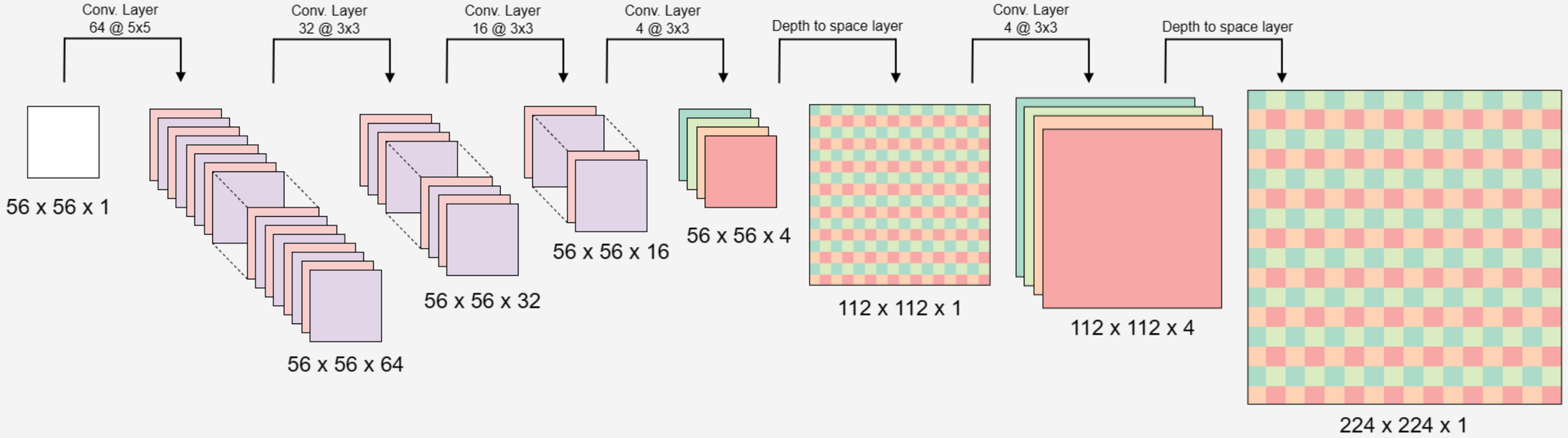
*Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. Wenzhe Shi, Jose Caballero , Ferenc Huszar´ , Johannes Totz , Andrew P. Aitken , Rob Bishop, Daniel Rueckert , Zehan Wang. 2016

La nostra CNN:



Efficient sub-pixel convolutional neural network

La nostra CNN:



- Addestrata utilizzando la **MSE loss** e la **Content Loss**
- **N. di parametri**: 25362 per immagini ad 1 canale e 30028 per immagini a 3 canali
- Funzione di attivazione **Tanh** per ogni layer
- **Fully convolutional**
- L'ingrandimento viene effettuato in due step separati

- Ogni tipo di modello è stato addestrato per 50 epoche
- Circa 30 secondi ad epoca per le reti addestrate con la MSE loss, circa 240 secondi ad epoca per le reti addestrate con la Content loss

*addestramento effettuato con GPU Tesla T4 di Colab

Funzioni di perdita

Errore quadratico medio (MSE):

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

Content Loss:

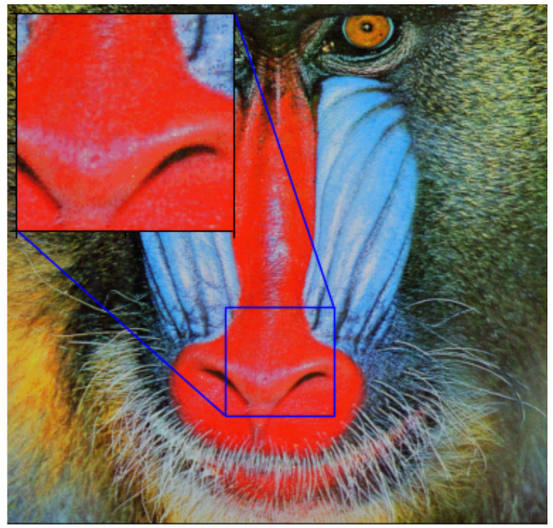
$$l_{VGG/i.j}^{SR} = \frac{1}{W_{i,j}H_{i,j}}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

Metrica

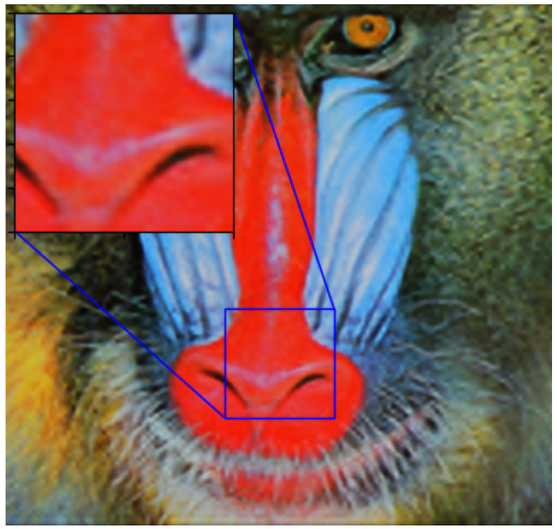
Peak Signal to Noise Ratio (PSNR):

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

Ground truth

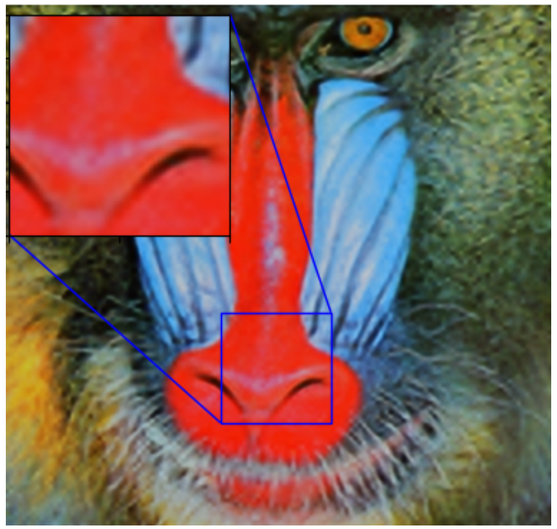


RGB MeanSquaredError



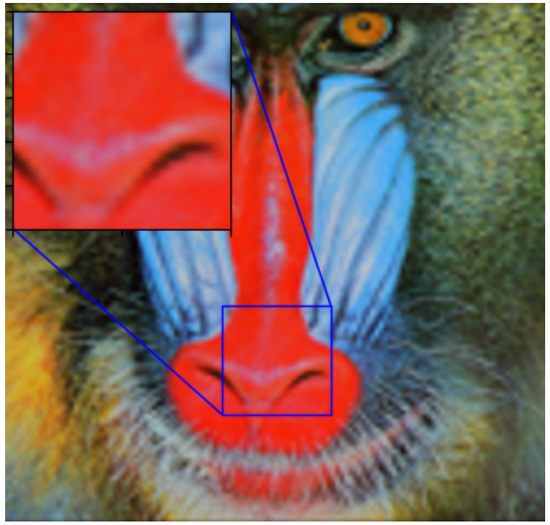
PSNR: 20.51

YCbCr MeanSquaredError



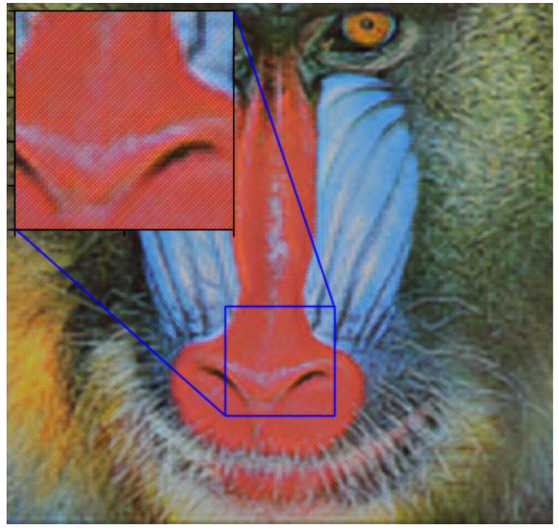
PSNR: 20.44

Bicubic interpolation



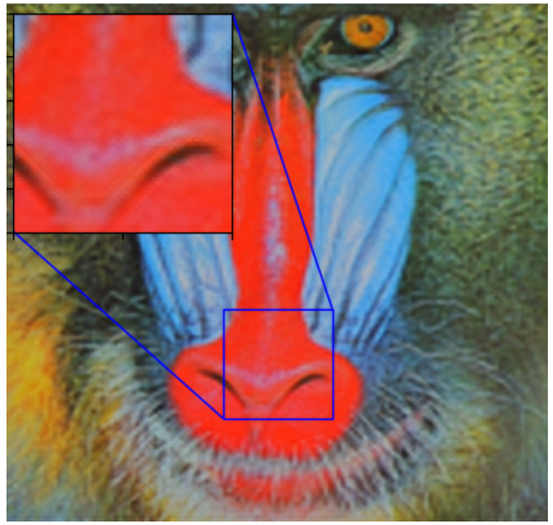
PSNR: 20.23

RGB Content Loss



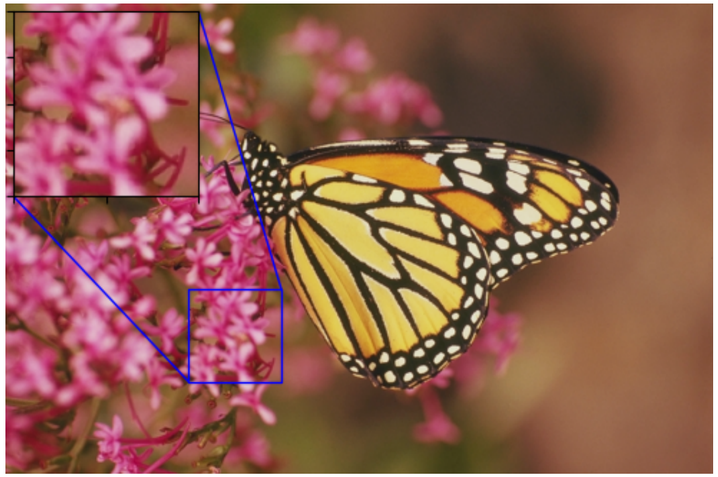
PSNR: 17.25

YCbCr Content Loss

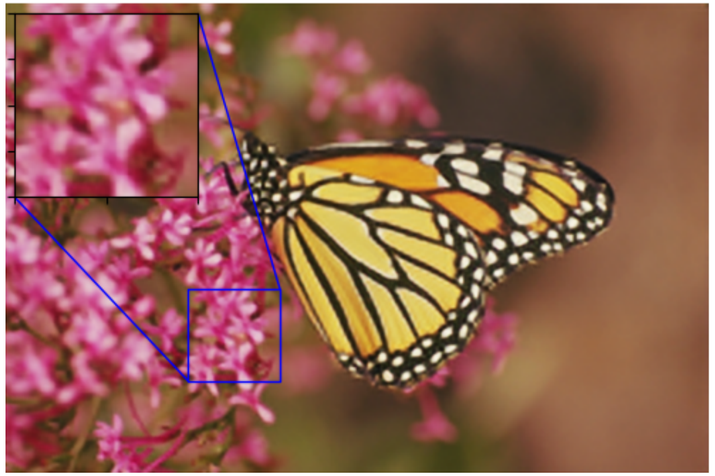


PSNR: 19.48

Ground truth

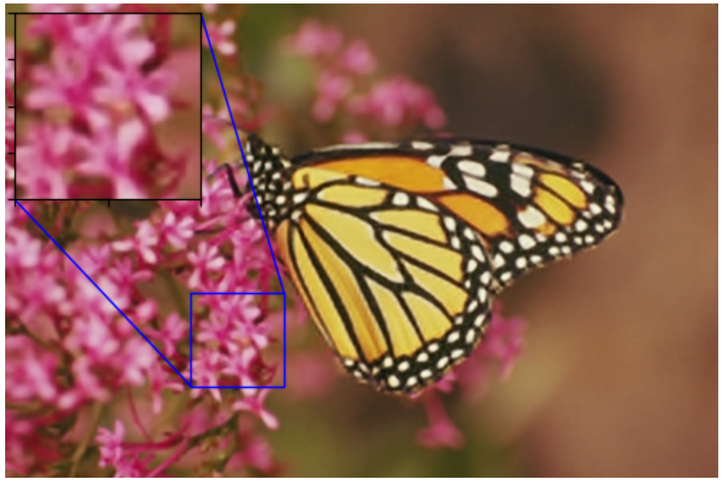


RGB MeanSquaredError



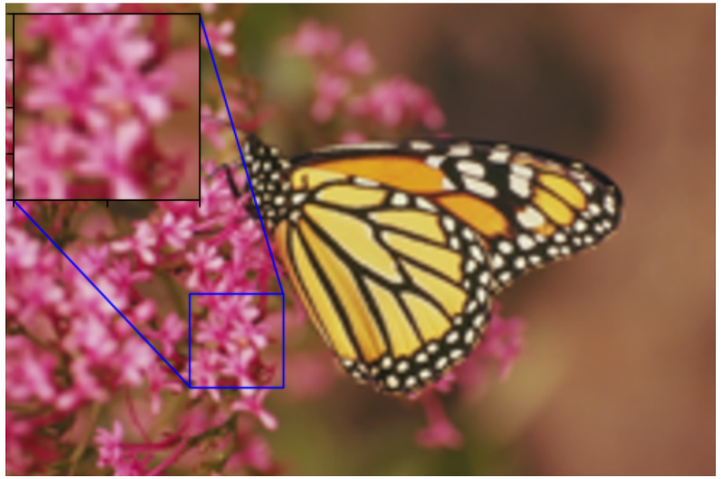
PSNR: 27.91

YCbCr MeanSquaredError



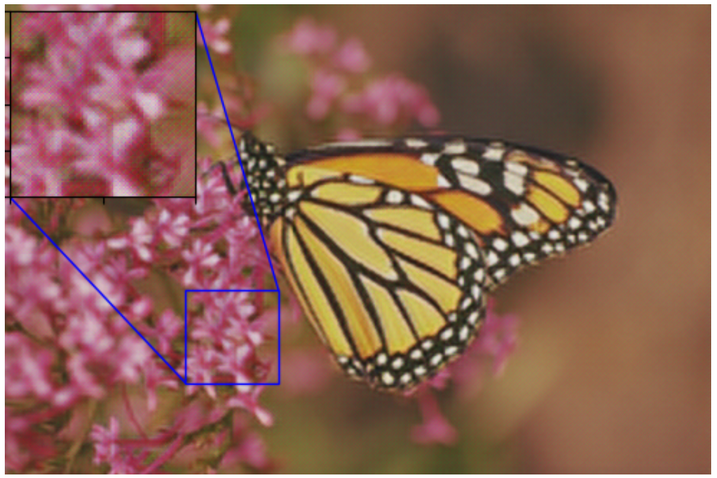
PSNR: 27.85

Bicubic interpolation



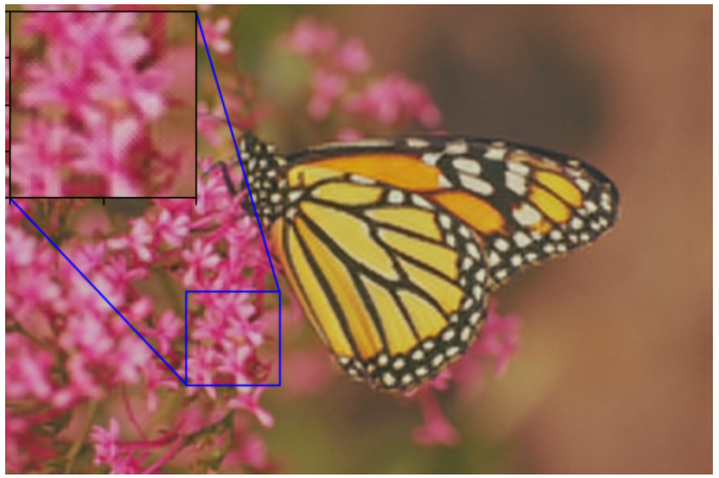
PSNR: 26.24

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

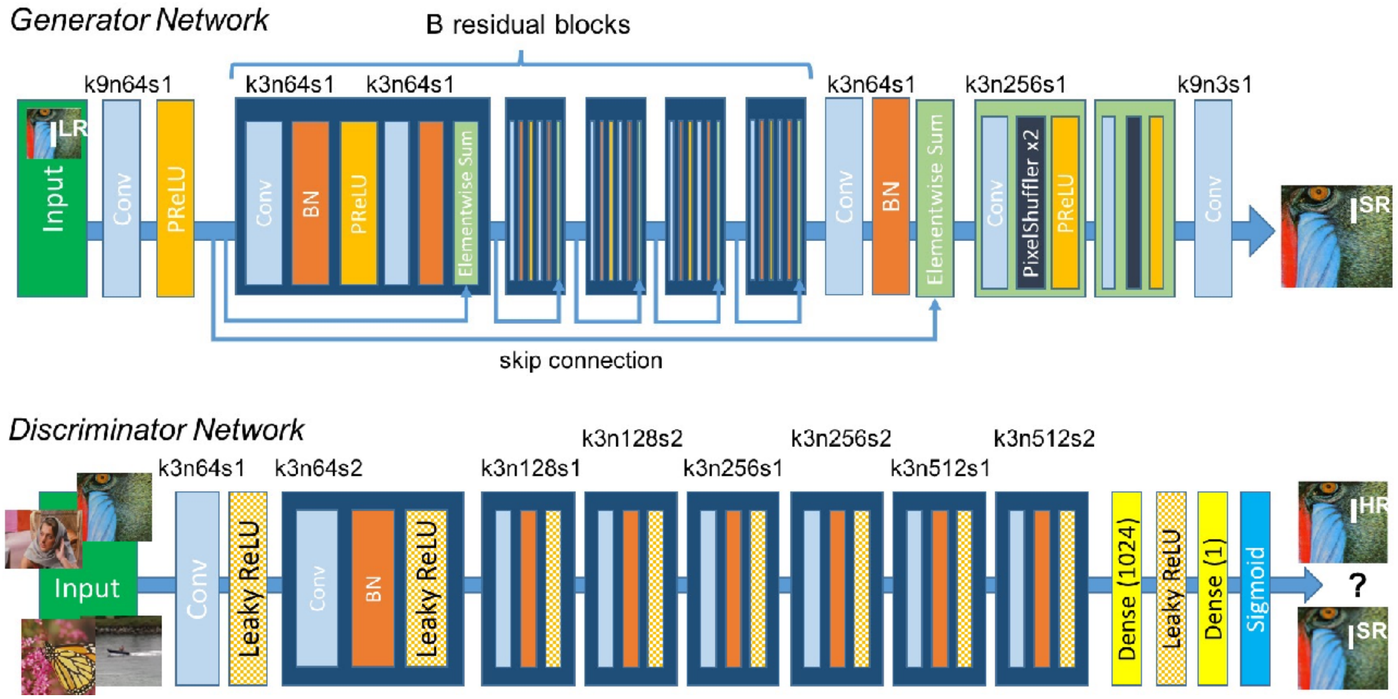
YCbCr Content Loss

PSNR: 18.87

Architettura GAN proposta da C. Ledig et al.*

- Addestrata utilizzando la **Perceptual Loss**
- **N. di parametri**: 1,554,883 (Generator) - 107,455,297 (Discriminator)
- **Generator Fully convolutional**
- Gli ultimi due blocchi del generator effettuano due up-scale x2

- ~ 500 epoche di addestramento per i due modelli (RGB e YCbCr)
- ~ 90 secondi per epoca



Perceptual Loss:

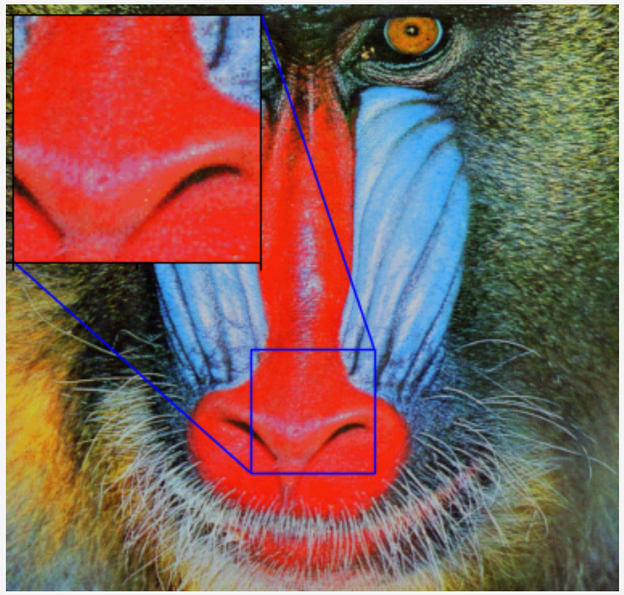
$$l^{SR} = \underbrace{l_{\mathrm{X}}^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}}$$

perceptual loss (for VGG based content losses)

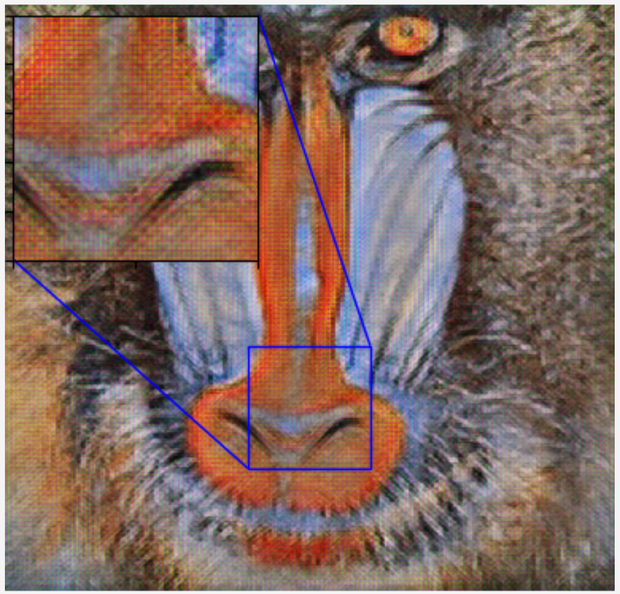
$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

*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

La SRGAN addestrata sulle immagini RGB ha prodotto risultati di super-resolution apprezzabili, ma si è palesato un problema di shifting dei colori.

In particolare sembra esserci una tendenza del modello a generare immagini con tonalità seppia.

La prima ipotesi è stata che la causa fosse un numero troppo limitato di epoche di addestramento o un numero troppo ridotto di osservazioni, ma abbiamo verificato che non era questo il caso.

Approfondendo la letteratura scientifica e thread github sulle SRGAN, sono stati identificati possibili problemi non riportati nel paper di C. Ledig et al.

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.

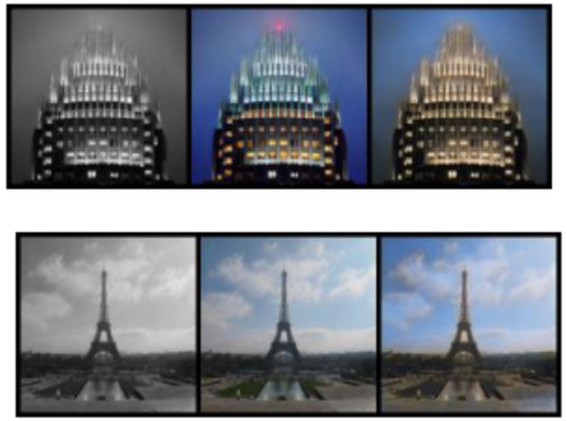


Figure 6: Converting Gray-Scale to Color (Tower)
Left: black and white, Center: original, Right: output.

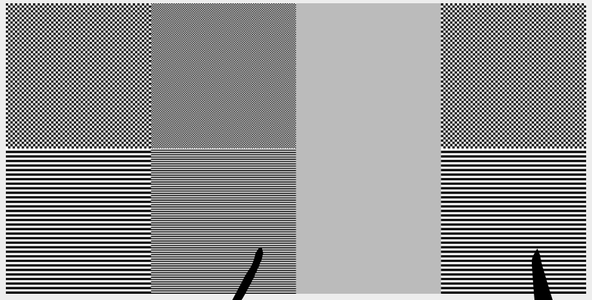
5.1.2. Dining Room

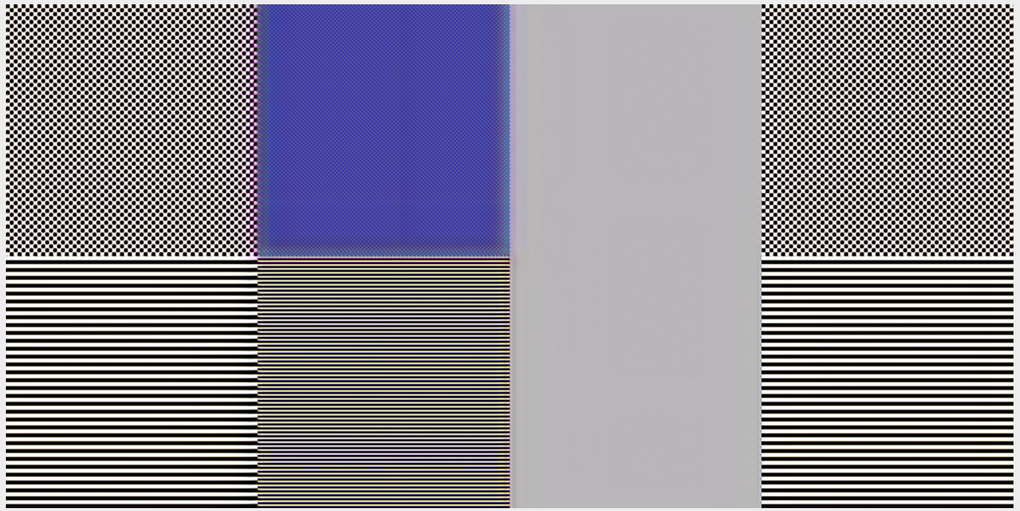
Figure 5 shows the result of our experiment for dining room coloring. Many training images contain tables and chairs, and very often if the color of the table is white, the chairs are brown, and vice versa. Two examples of those color combinations are presented. It appears the network detects the texture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

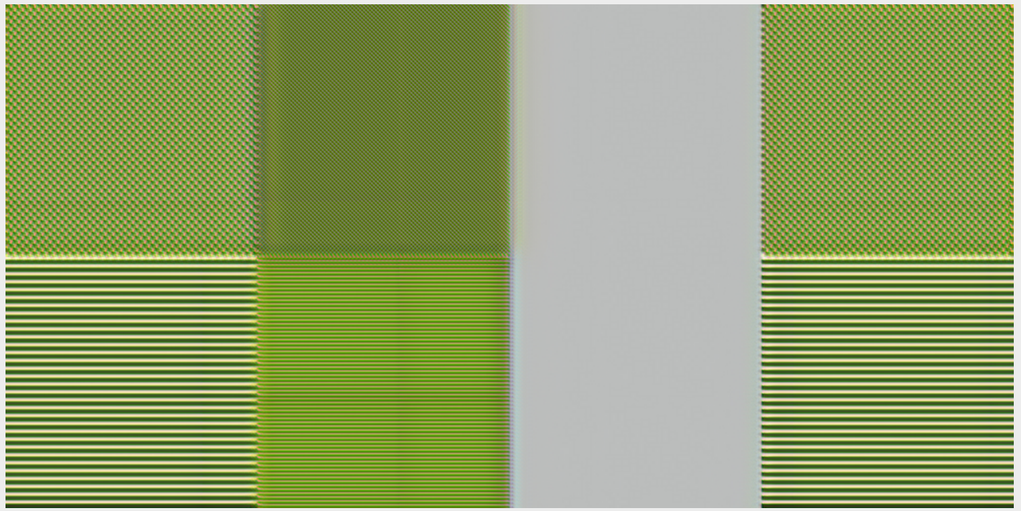
5.1.3. Tower

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

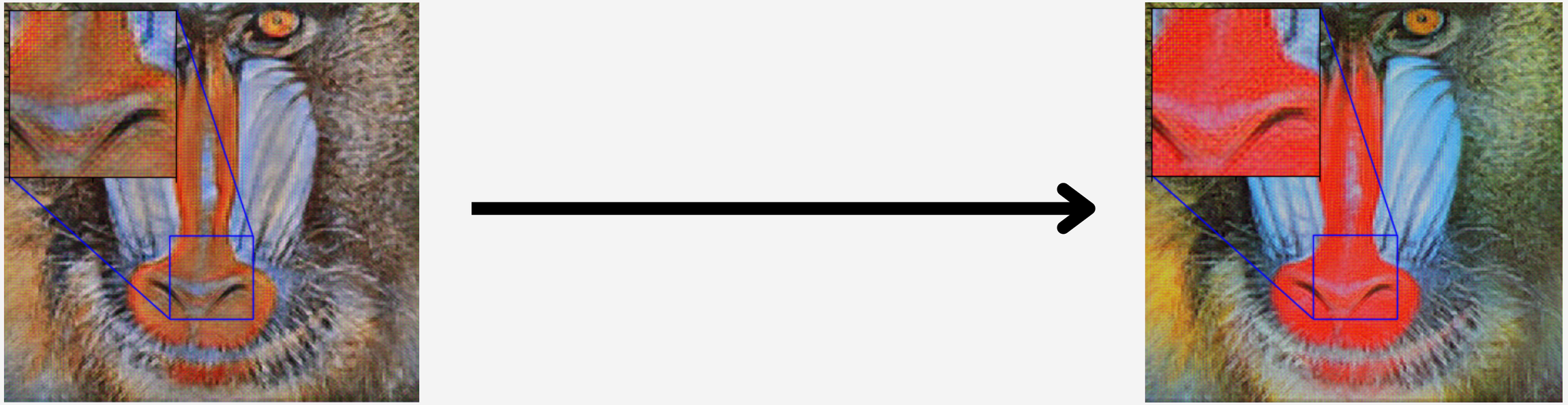
devernay commented on Sep 18, 2020 • edited

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 -
```

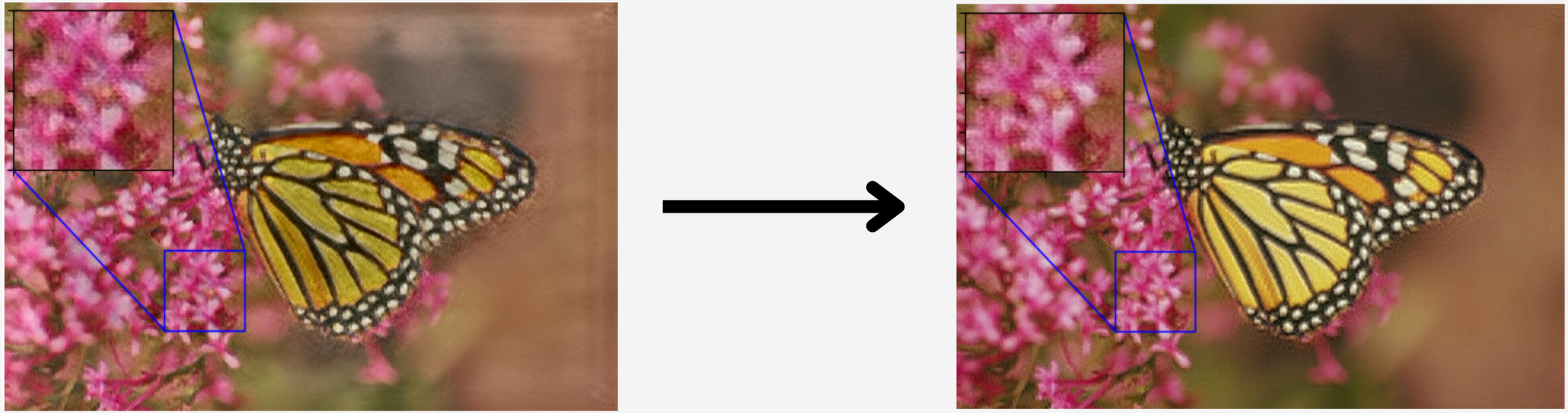
SRGAN RGB Color and Level Correction



Per risolvere il probelma del color shift, si procede effettuado un down-scaling dell'immagine SR, si computa la differenza con l'immagine LR e si esegue l'up-scaling della matrice differenze che viene sommata all'immagine SR.

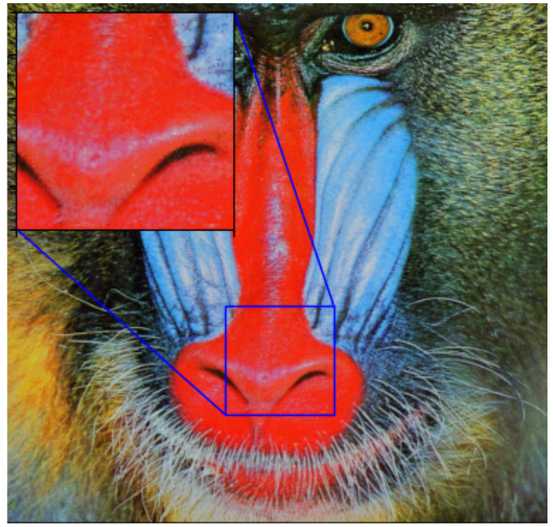
Successivamente operiamo una trasformazione su tutti i livelli della luminosità e del contrasto con correzione gamma

SRGAN YCbCr Luminance Correction

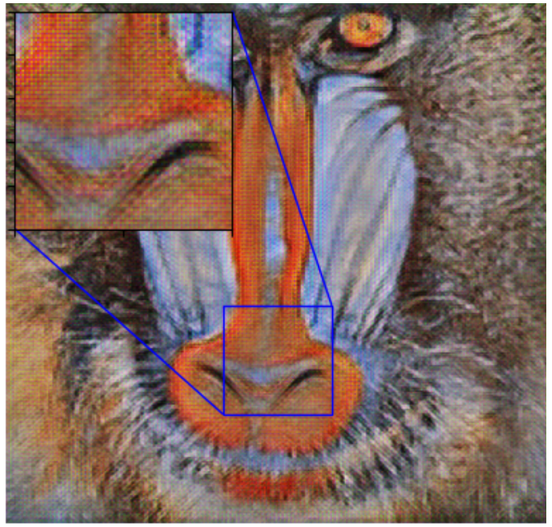


Si effettua la stessa correzione per il canale della luminanza
nelle immagini generate dalla SRGAN YCbCr

Ground truth

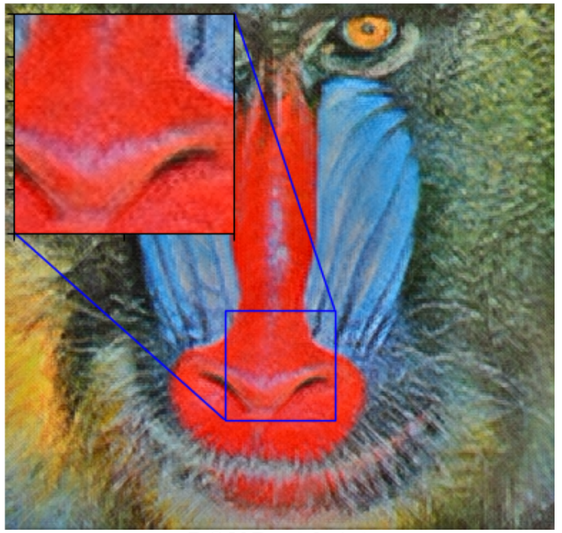


RGB GAN Perceptual Loss



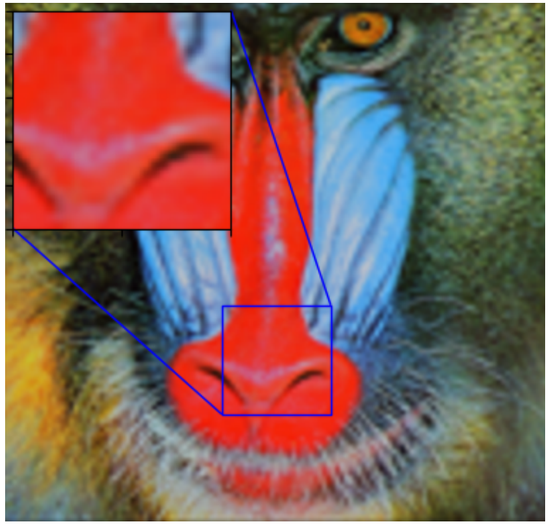
PSNR: 14.71

YCbCr GAN Perceptual Loss



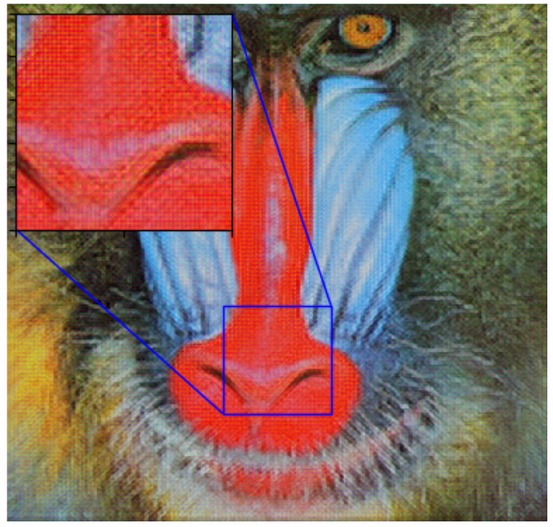
PSNR: 16.41

Bicubic interpolation



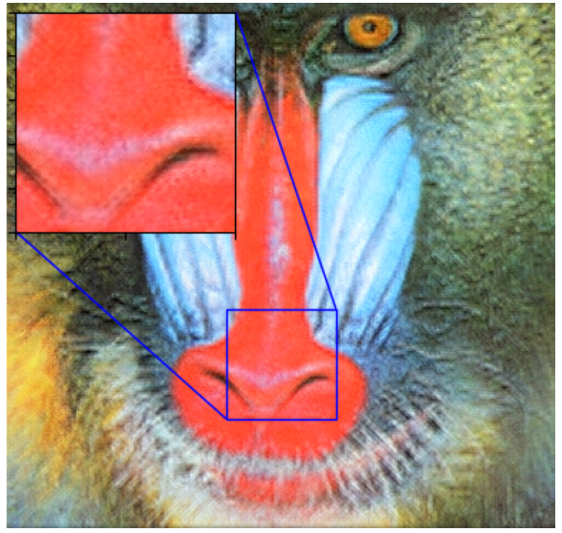
PSNR: 20.23

RGB GAN Perceptual Loss w/ correction



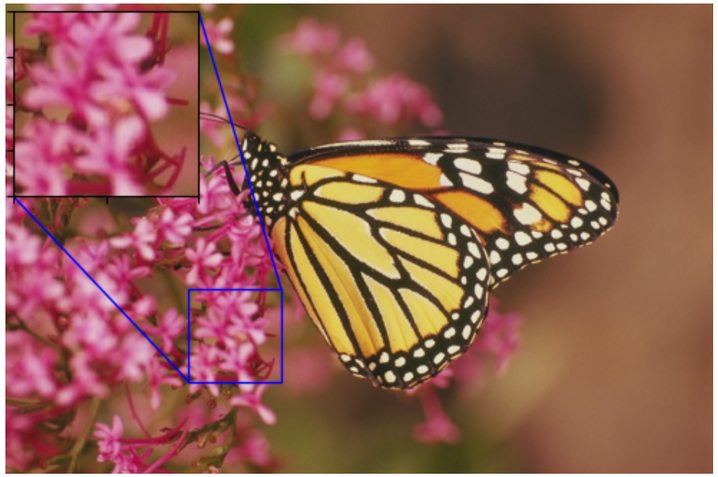
PSNR: 17.16

YCbCr GAN Perceptual Loss w/ correction

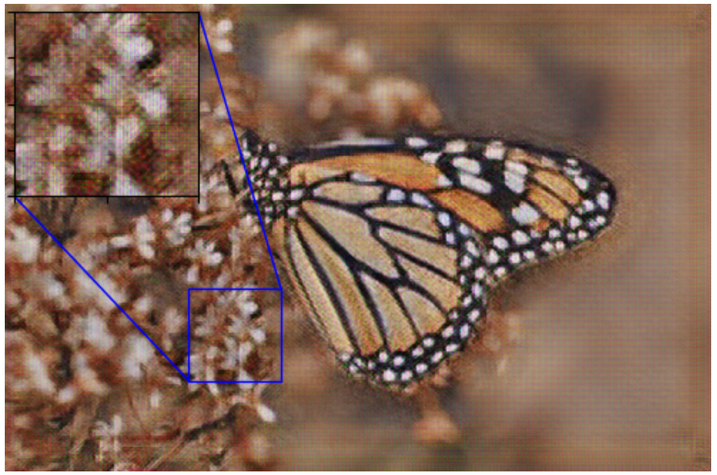


PSNR: 15.03

Ground truth

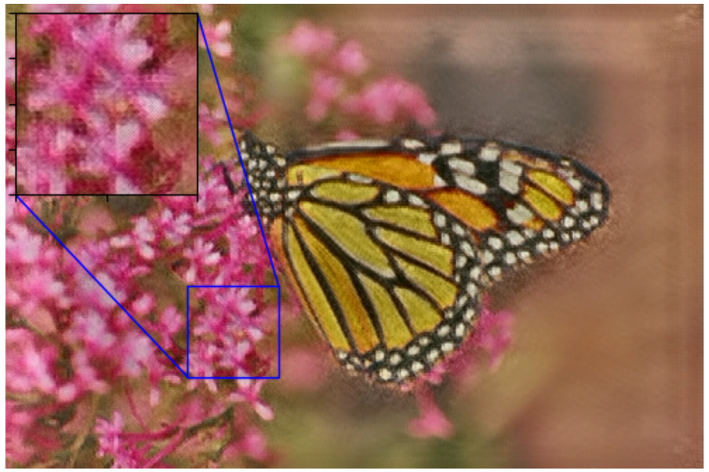


RGB GAN Perceptual Loss



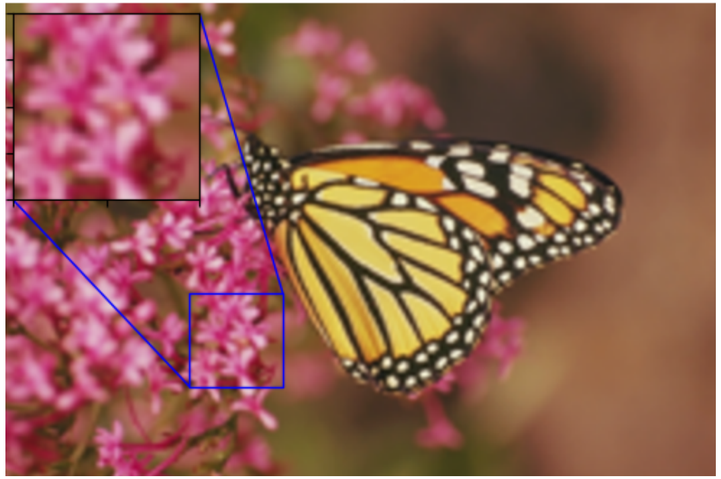
PSNR: 16.63

YCbCr GAN Perceptual Loss



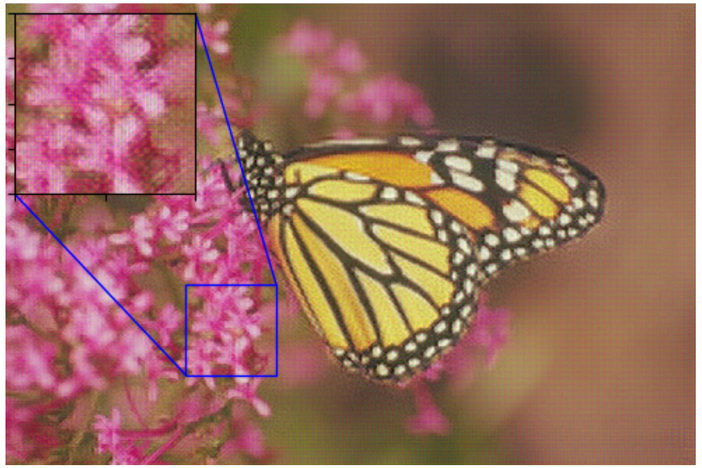
PSNR: 17.69

Bicubic interpolation



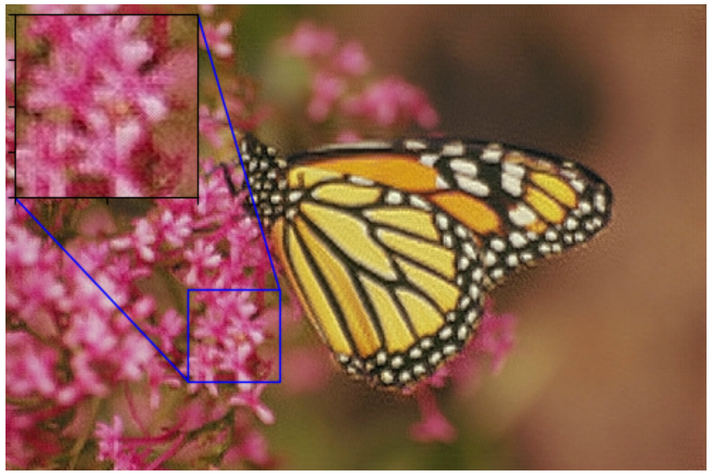
PSNR: 26.24

RGB GAN Perceptual Loss w/ correction



PSNR: 19.50

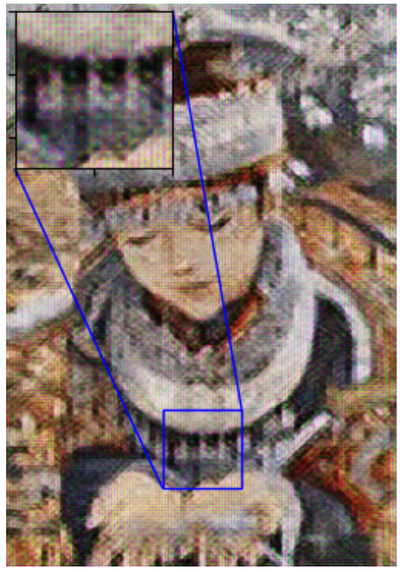
YCbCr GAN Perceptual Loss w/ correction



PSNR: 23.49

Ground truth

RGB GAN Perceptual Loss



PSNR: 14.26

YCbCr GAN Perceptual Loss

PSNR: 14.19

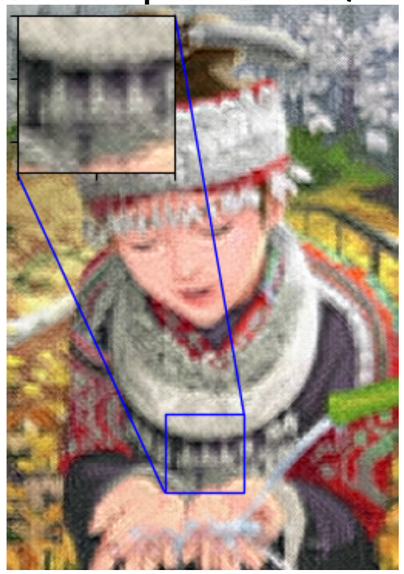
Bicubic interpolation

PSNR: 20.25

RGB GAN Perceptual Loss w/ correction

PSNR: 16.89

YCbCr GAN Perceptual Loss w/ correction



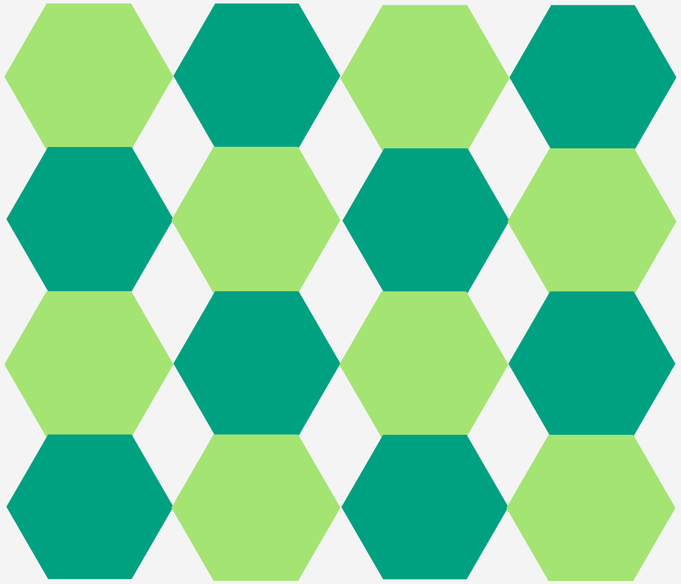
PSNR: 19.11

Valori PSNR:

	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

Digital Signal and
Image Management

Grazie per l'attenzione!



Gaetano Chiriaco, Riccardo Porcedda, Gianmarco Russo