SRGAN

EECE 5644, Machine Learning and pattern Recognition

Mayur Bhise, Derek Koechlin

General Adversarial Network (GAN)

Algorithm

While not converged **do**

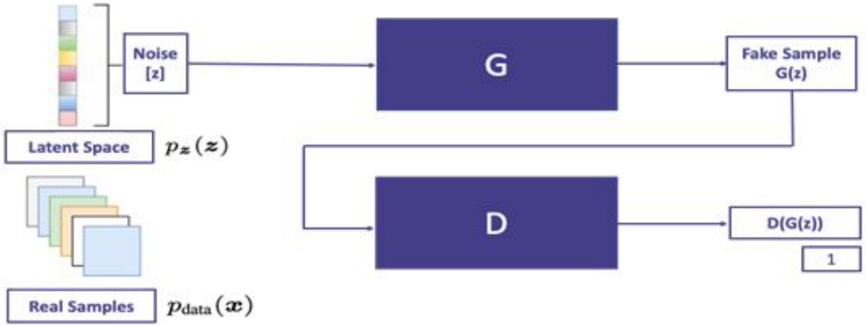
1. **For** k steps **do**
   1.1 Draw B training samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_B\}$ from $p_{data}(\mathbf{x})$
   1.2 Draw B latent samples $\{\mathbf{z}_1, \ldots, \mathbf{z}_B\}$ from $p(\mathbf{z})$
   1.3 Update the **discriminator** D by **ascending** its stochastic gradient:

$$\nabla_{\mathbf{w}_D} \frac{1}{B} \sum_{b=1}^{B} \log D(\mathbf{x}_b) + \log(1 - D(G(\mathbf{z}_b)))$$

2. Draw B latent samples $\{\mathbf{z}_1, \ldots, \mathbf{z}_B\}$ from $p(\mathbf{z})$
3. Update the **generator** G by **descending** its stochastic gradient:

$$\nabla_{\mathbf{w}_G} \frac{1}{B} \sum_{b=1}^{B} \log(1 - D(G(\mathbf{z}_b)))$$

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})} \boxed{[\log D(\boldsymbol{x})]} + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})} [\log(1 - \boxed{D(G(\boldsymbol{z}))})]$$



Super-Resolution GAN (SRGAN)

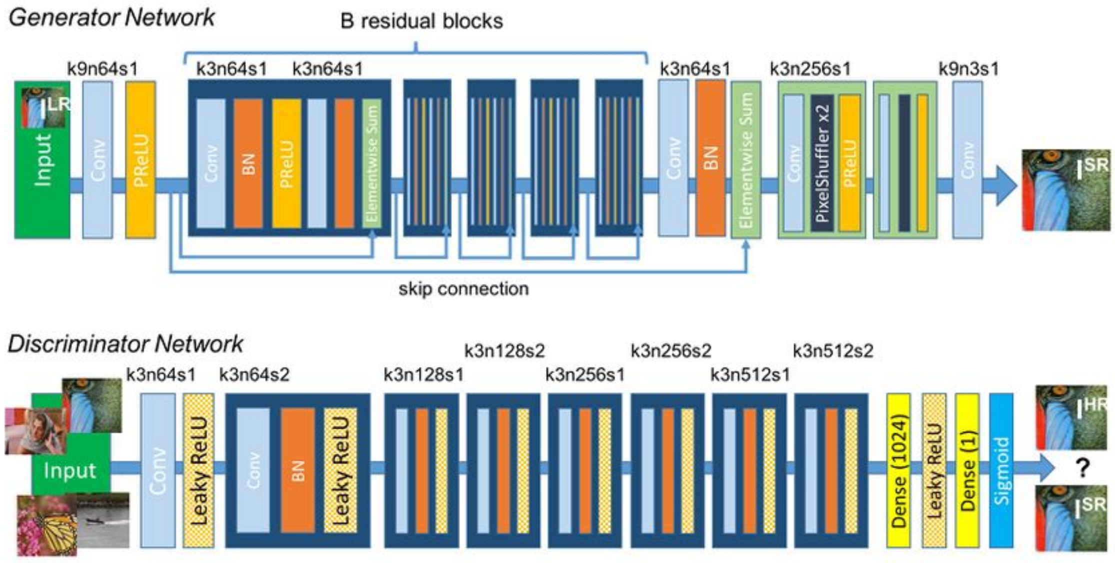
- Given LR image enhances image resolution and reconstruct SR approx. equal to HR image.
- CNN Based

Challenges

- Network architecture
- Loss function
- Training procedure
- Upsampling method
- Real-time

Applications : Medical imaging, Video and Image enhancement, Satellite images, Video Game upscaling, Low Res camera upscaling

SRGAN architecture



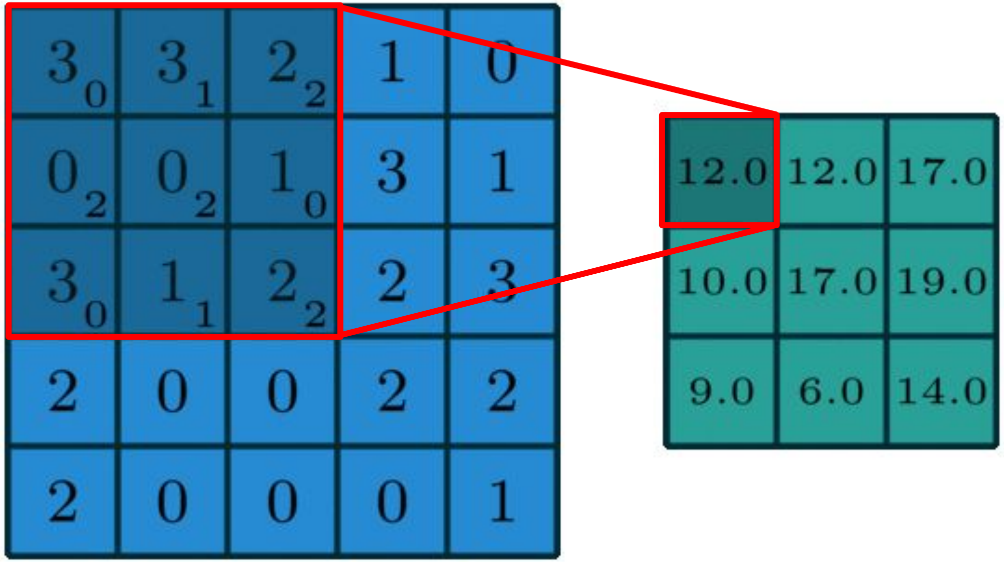
2D Convolution

0+3+4

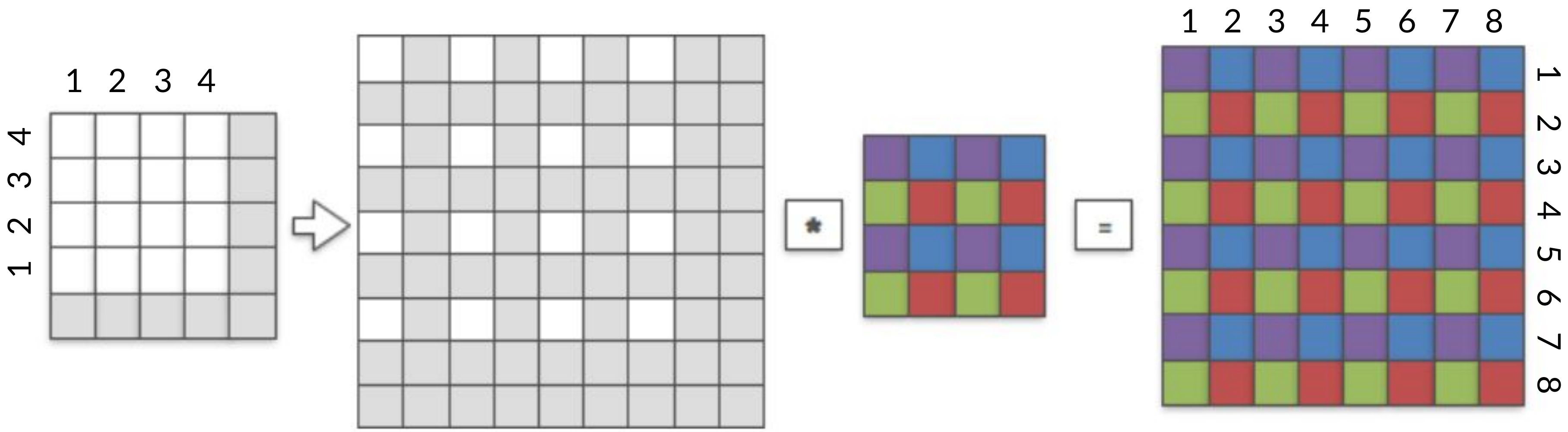
0+0+0

0+1+4

12



Pixel Shuffle



Loss Function

MSE LOSS

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

PERCEPTUAL LOSS

$$l^{SR} = \underbrace{\underbrace{l_{X}^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}}$$

Hyperparameter Selection

- General Adversarial Networks have a massive amount of hyperparameters to choose

- Specific model hyperparameters were selected according to IEEE research into GANs and SRGANs

- Additional hyperparameters include:
  - Input size - 64x64
  - Output size - 256x256
  - Residual Blocks - 16
  - Epochs - 750 & 4000
  - Batch Size - 8 & 1
  - Learning Rate - .0001

Dataset

- Mirflickr Dataset with 25k images of different sizes.
- Other benchmark datasets: Set5, BSD100, Set14

Preprocessing and Augmentation

- LR images with size 64 x 64
- LR images 4x downscaled (Bicubic)
- 0-1 Normalization

Metrics

- MSE

$$MSE = \frac{1}{NxM}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[X(i,j) - Y(i,j)]^2$$

- PSNR

$$PSNR = 10.\log(\frac{{MAX_I}^2}{MSE})$$

- SSIM

$$\text{SSIM}(x,y) = \left[l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}\right]$$

Results :

Metrics	SR-ResNet-L2	SRGAN (authors)	SRGAN (ours)$_{\text{at 4k epochs}}$
MSE	280	N/A	68.16
PSNR	21.97	29.4	29.75
SSIM	0.47	0.84	0.72

Training Time:

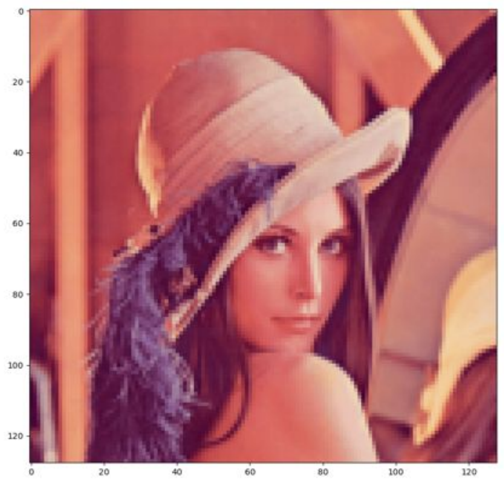
Epochs: 750	Batch Size: 8	Time: 11 Hours
Epochs: 4000	Barch Size: 1	Time: 15 Hours

Results - Images

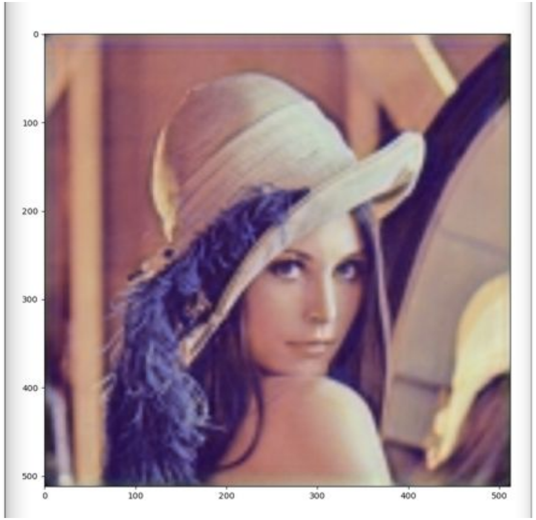
Low Res - Generated - Original

Results - Images

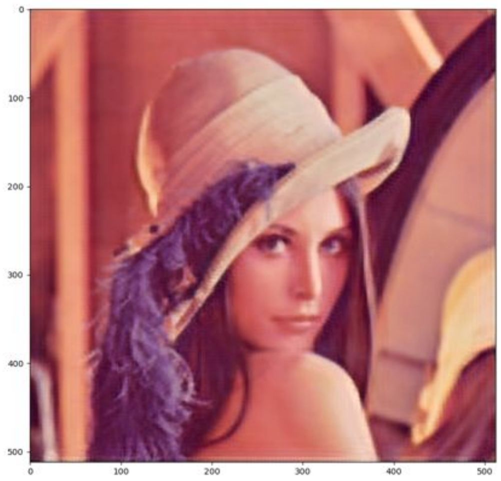
Input 128 x 128



SR 512 x 512, 750E



SR 512 x 512, 4000E

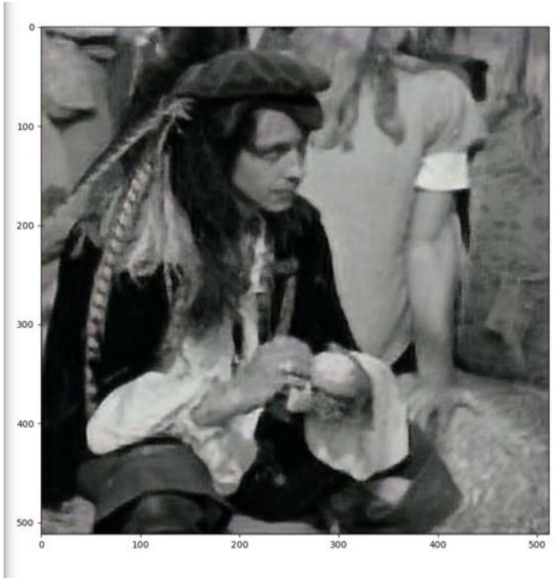


Results - Images

Input 128 x 128

SR 512 x 512, 750E

SR 512 x 512, 4000E



Results - Images

Input 128 x 128

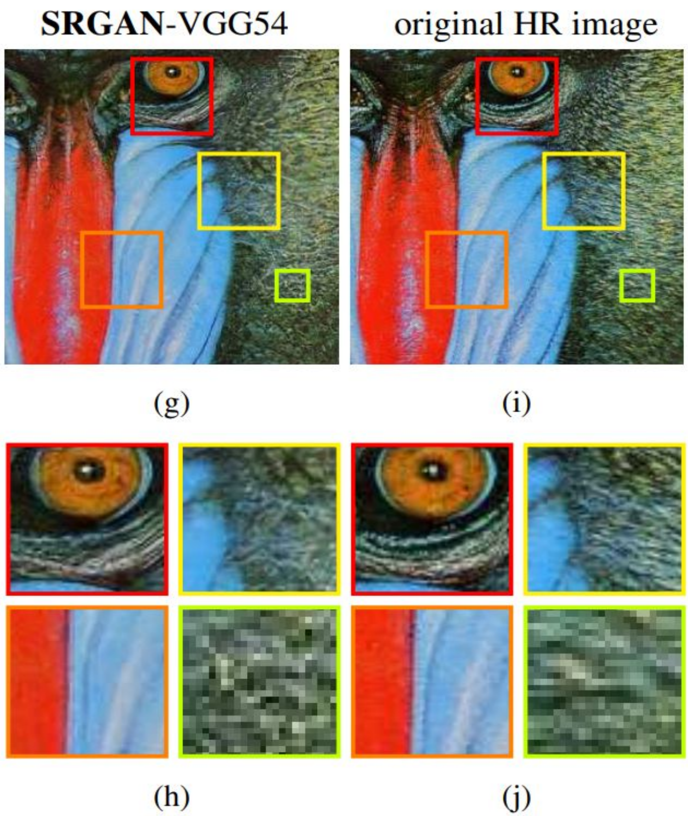
SR 512 x 512, 750E

SR 512 x 512, 4000E

Optimal SRGAN

- GANs require massive datasets in order to generate quality images
- The optimal training would use the ImageNet dataset containing over 1,000,000 images
- Longer training times also improve the accuracy of the model. Using the ImageNet dataset, training could take multiple days

Optimal Results



Code links and References

[1] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi (Twitter) - Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, https://arxiv.org/abs/1609.04802v5

[2] Ian J. Goodfellow, Jean Pouget-Abadie∗, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio D´epartement d'informatique et de recherche opérationnelle Université de Montréal arXiv:1406.2661v1 [stat.ML] 10 Jun 2014 Montreal, QC H3C 3J7, https://arxiv.org/abs/1406.2661

[3] https://theaisummer.com/skip-connections/

[4] https://pytorch.org/docs/stable/generated/torch.nn.PixelShuffle.html?highlight=pixel%20shuffle#torch.nn.PixelShuffle

[5] https://towardsdatascience.com/srgan-a-tensorflow-implementation-49b959267c60

[6] https://uni-tuebingen.de/fakultaeten/mathematisch-naturwissenschaftliche-fakultaet/fachbereiche/informatik/lehrstuehle/autonomous-vision/lectures/deep-learning/

[7] https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

[8] https://arxiv.org/abs/1609.07009