23b2444

Harsh kumar singh SOC 2025

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Introduction~

This research explains that **super-resolution (SR)** aims to recover high-resolution (HR) images from low-resolution (LR) ones, which is a very **challenging and ill-posed problem**, especially at large upscaling levels (e.g., 4×). Traditional methods often use **mean squared error (MSE)** as the loss function because it boosts **PSNR**, a popular metric. However, MSE and PSNR fail to preserve **fine texture details** and do not reflect how humans perceive image quality.To overcome this, the authors propose **SRGAN**, a **Generative Adversarial Network** that uses a **deep ResNet with skip connections** and a **perceptual loss.**

This section explains why traditional **pixel-wise loss functions like MSE** are not ideal for super-resolution. MSE tends to **average out all possible high-detail variations**, resulting in **blurry and smooth images** that lack realistic textures. That's because it treats every pixel equally and doesn't account for **how humans perceive image quality**. To fix this, researchers started using **GANs (Generative Adversarial Networks)**, which can generate images that lie closer to the **natural image distribution**, making them look more realistic.Instead of just comparing raw pixels, newer methods use **feature-based losses** especially from deep networks like **VGG19** which focus on **high-level image features**. This allows the model to better preserve **textures and structures**

Method~

To recover a **high-resolution (HR) image** I{SR} from a given **low-resolution I{LR}**

I{LR} has shape:
W×H×C
where:

- W,H width and Height of the LR image
- C: Number of color channels (e.g., 3 for RGB)

I{HR} and I{SR} have shape:
rW×rH×C

because they are upscaled by a factor r.

📐 **Optimization Formula:**

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} \ell_{SR}(G_{\theta_G}(I_{LR}^{(n)}), I_{HR}^{(n)})$$

- **The model is trained by minimizing the average loss between the predicted super-resolved image G(I{LR}) and the true high-resolution image I{HR} , across all N training samples.**
- **argmin means we are looking for the best set of parameters θ{G} that minimizes this average loss.**

Goal of Generator G: Generate super-resolved images from low-resolution inputs that **fool the discriminator**.

Goal of Discriminator D: Accurately classify real high-resolution images I{HR} as real, and generated ones G(I{LR}) as fake.

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR} \sim p_{\text{train}}(I^{HR})}[\log D_{\theta_D}(I^{HR})] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR}))]$$

Generator Architecture (SRGAN Generator)

Uses a **deep CNN** with **B residual blocks**, each made of:

- Two 3×3 convolutional layers
- 64 feature maps
- **Batch normalization**
- **Parametric ReLU (PReLU)** activations

Upsampling is done via **two sub-pixel convolution layers** [48], which efficiently increase image size without interpolation

Discriminator Architecture:

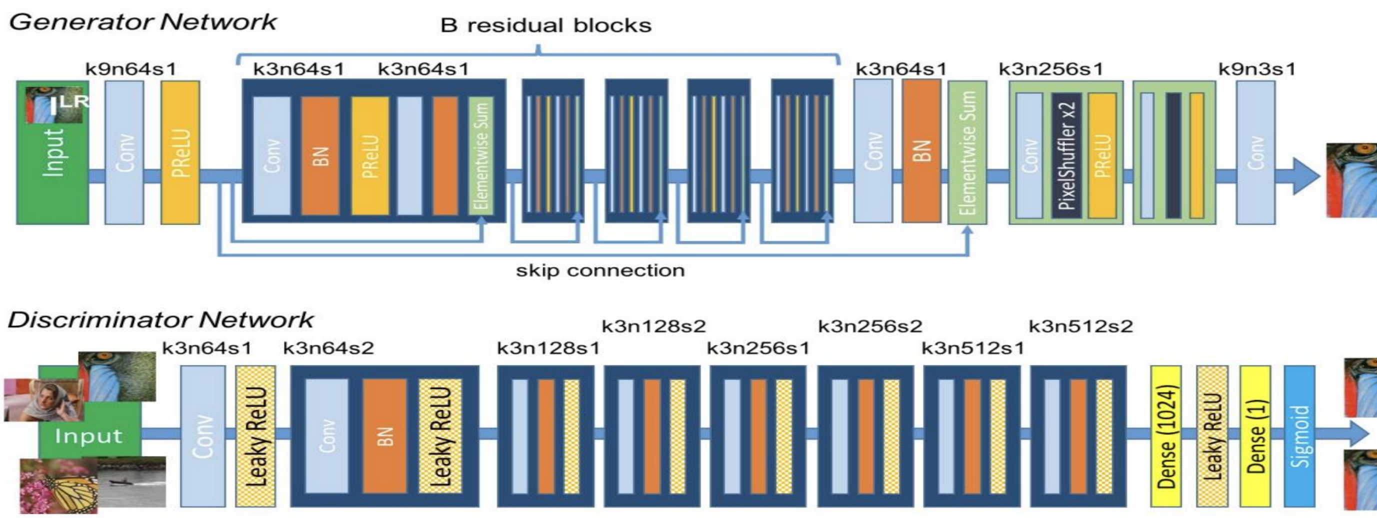
Discriminator is designed following **DCGAN-style guidelines** [44]:

- Uses **Leaky ReLU** activation (α = 0.2)
- Avoids **max pooling**

- Has **8 convolutional layers** with 3×33 \times 33×3 filters
- Number of filters increases from 64 to 512 (doubles every few layers)
- Uses **strided convolutions** to downsample instead of pooling

Final part:

- Outputs 512 features
- Passes through **2 dense (fully connected) layers**
- Ends with a **sigmoid activation**, giving a probability (real vs fake)



Perceptual Loss

Instead of using just MSE (which looks at raw pixel differences), the authors create a **better loss function** made of two parts:

$$\ell_{SR} = \underbrace{\ell^{X}_{SR}}_{\text{Content Loss}} + 10^{-3} \cdot \underbrace{\ell^{Gen}_{SR}}_{\text{Adversarial Loss}}$$

Content Loss ℓSRX

1. Measures how **similar the generated image is to the real image** in terms of meaningful features (like texture or structure).
2. Often computed using features from a **VGG network** (a deep CNN trained for image classification).

3. This is more aligned with **how humans judge image quality**, unlike MSE which looks only at per-pixel differences.
4.

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

$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2 \qquad (5)$$

Here $W_{i,j}$ and $H_{i,j}$ describe the dimensions of the spective feature maps within the VGG network.

Adversarial Loss ℓSRGen

- Comes from the **discriminator** in the GAN setup .
- Encourages the generator to make images that look **realistic and natural**, so they can **fool the discriminator**.
- It's multiplied by a small factor (**0.001**) to keep it from overpowering the content loss

is defined based on the probabilities of the discriminator $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ over all training samples as:

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \qquad (6)$$

Here, $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ is the probability that the reconstructed image $G_{\theta_G}(I^{LR})$ is a natural HR image. For better gradient behavior we minimize $-\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$ instead of $\log[1 - D_{\theta_D}(G_{\theta_G}(I^{LR}))]$ [22].

Model testing and results

Metrics:

- **PSNR (Peak Signal-to-Noise Ratio):** Measures pixel-wise accuracy. **Higher is better**, but not always aligned with visual quality.
- **SSIM (Structural Similarity Index):** Measures perceptual similarity (structure, contrast). **Higher is better.**
- **MOS (Mean Opinion Score):** Human-rated visual quality. Scores range from 1 (bad) to 5 (excellent). **Higher is better** and reflects **true perceptual quality**.

  - **SRResNet** is **technically most accurate** (best PSNR/SSIM) but visually looks smooth/blurry (lower MOS).
  - **SRGAN** generates images that **look far better to humans**, scoring **much higher in MOS**, despite lower PSNR.
  - So, **SRGAN > SRResNet** for *realistic and perceptual quality*, even though it sacrifices some numerical precision.

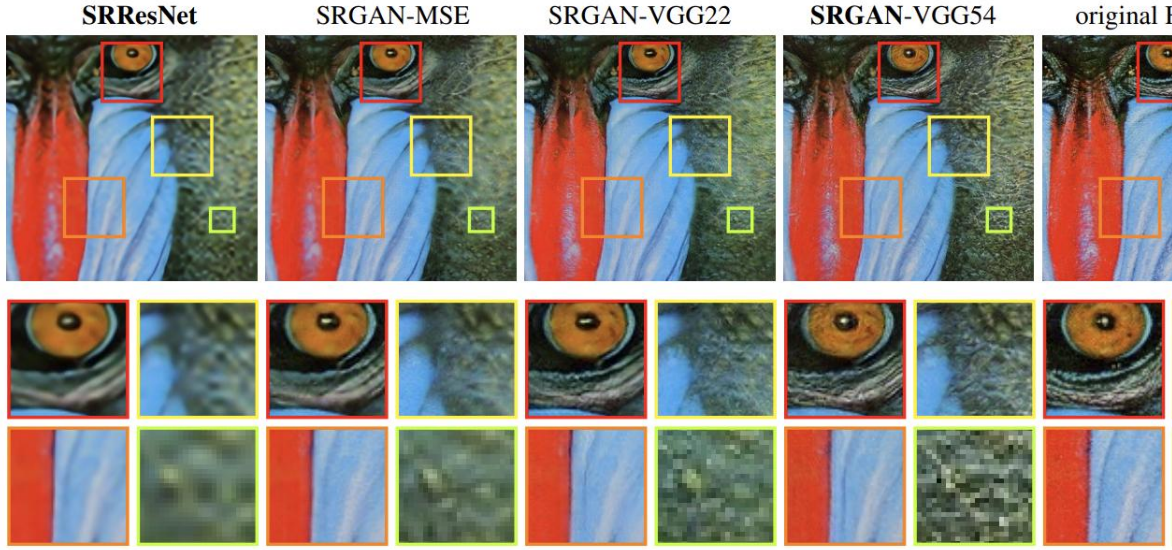


Figure 6: **SRResNet** (left: a,b), SRGAN-MSE (middle left: c,d), SRGAN-VGG2.2 (middle: e,f) and **SR**
(middle right: g,h) reconstruction results and corresponding reference HR image (right: i,j). [4× upscaling]

Table 2: Comparison of NN, bicubic, SRCNN [9], SelfExSR [31], DRCN [34], ESPCN [48], **SRResNet**, **SR**
and the original HR on benchmark data. Highest measures (PSNR [dB], SSIM, MOS) in bold. [4× upscaling

Set5	nearest	bicubic	SRCNN	SelfExSR	DRCN	ESPCN	**SRResNet**	**SRGAN**	HR
PSNR	26.26	28.43	30.07	30.33	31.52	30.76	**32.05**	29.40	∞
SSIM	0.7552	0.8211	0.8627	0.872	0.8938	0.8784	**0.9019**	0.8472	1
MOS	1.28	1.97	2.57	2.65	3.26	2.89	3.37	**3.58**	4.3
Set14									
PSNR	24.64	25.99	27.18	27.45	28.02	27.66	**28.49**	26.02	∞
SSIM	0.7100	0.7486	0.7861	0.7972	0.8074	0.8004	**0.8184**	0.7397	1
MOS	1.20	1.80	2.26	2.34	2.84	2.52	2.98	**3.72**	4.3
BSD100									
PSNR	25.02	25.94	26.68	26.83	27.21	27.02	**27.58**	25.16	∞
SSIM	0.6606	0.6935	0.7291	0.7387	0.7493	0.7442	**0.7620**	0.6688	1
MOS	1.11	1.47	1.87	1.89	2.12	2.01	2.29	**3.56**	4.4

- Shows zoomed-in image patches from different models.
- **SRResNet** looks smooth and lacks texture.
- **SRGAN-VGG54** (final version of SRGAN) produces **sharp and detailed textures**, much closer to the **original HR image**.
- **SRGAN-MSE** is closer to SRResNet and looks blurrier than VGG-based SRGANs.
- **SRGAN-VGG22** is intermediate—better than MSE, but not as good as VGG54.

The best visual result is clearly from **SRGAN-VGG54**, capturing sharp edges and textures that mimic the HR image, while SRResNet and MSE-based outputs are visually smoother and less realistic.