Comparação de Arquiteturas GAN para Super-Resolução de Imagens: SRGAN, ESRGAN e Real-ESRGAN

Daniel Henry
danielhenry@id.uff.br
Universidade Federal Fluminense
Rio das Ostras, RJ, Brasil

Resumo

Este trabalho compara três arquiteturas baseadas em Redes Generativas Adversariais (GANs) para super-resolução de imagens: SRGAN, ESRGAN e Real-ESRGAN. Utilizando um subconjunto do dataset DIV2K (100 imagens de treino e 20 de validação), com imagens de 128x128 pixels e um batch size de 4, avaliamos as diferenças arquiteturais e de desempenho. As métricas PSNR e SSIM foram usadas para avaliação quantitativa, enquanto visualizações qualitativas destacaram diferenças em nitidez e texturas. Os resultados indicam que o SRGAN tende a produzir imagens suavizadas, enquanto o ESRGAN e o Real-ESRGAN geram texturas mais nítidas, com o Real-ESRGAN apresentando melhor qualidade visual devido ao seu discriminador U-Net e normalização espectral.

Palavras Chave

Super-resolução, SRGAN, ESRGAN, Real-ESRGAN, GANs, DIV2K, PSNR, SSIM

1 Introdução

A super-resolução de imagens (SR) é uma tarefa de visão computacional que busca reconstruir imagens de alta resolução (HR) a partir de imagens de baixa resolução (LR). Redes Generativas Adversariais (GANs) têm se destacado nessa área, com arquiteturas como SRGAN [7], ESRGAN [13] e Real-ESRGAN [12] oferecendo avanços significativos em qualidade visual. O SRGAN introduziu a combinação de perdas adversariais e perceptivas, enquanto o ESRGAN aprimorou a arquitetura com blocos densos residuais (RRDB) e perda perceptual dupla. O Real-ESRGAN incorpora um discriminador U-Net e normalização espectral para melhorar a robustez em cenários complexos e com degradações desconhecidas.

Este trabalho compara essas três arquiteturas usando o dataset DIV2K, com um subconjunto de 100 imagens de treino e 20 de validação, imagens de 128x128 pixels e 5 épocas de treinamento, devido às **limitações computacionais** inerentes ao ambiente de execução (GPU, RAM e tempo). O estudo visa demonstrar as diferenças arquiteturais e de desempenho dessas abordagens, utilizando métricas como PSNR e SSIM, além de análises qualitativas.

2 Trabalhos Relacionados

A super-resolução baseada em aprendizado profundo evoluiu significativamente com o uso de GANs, conforme uma análise de Tian et al. [11] e Feng [2]. O **SRGAN** [7] foi pioneiro, introduzido por Ledig et al., ao ser a primeira arquitetura capaz de inferir imagens foto-realísticas com fatores de upscaling de 4× [7]. Este método divergiu do foco tradicional em minimizar o erro quadrático médio (MSE), que geralmente resultava em imagens com falta de detalhes de alta frequência e perceptualmente insatisfatórias [7]. Para isso, o SRGAN propôs uma função de perda perceptual, combinando uma perda adversarial e uma perda de conteúdo motivada pela similaridade perceptual, em vez da similaridade no espaço de pixels [7].

Subsequentemente, o **ESRGAN** [13], proposto por Wang et al., surgiu como uma evolução para aprimorar ainda mais a qualidade visual dos resultados. As melhorias do ESRGAN incluem a introdução do **Residual-in-Residual Dense Block (RR Thunder)** como unidade básica de construção da rede, e a **remoção das camadas de Batch Normalization (BN)** para evitar artefatos e melhorar a capacidade de generalização da rede [13, 13]. Além disso, o ESRGAN aprimorou a perda adversarial utilizando o conceito de GANs relativísticas (RaGAN), onde o discriminador prevê a "realidade relativa" de uma imagem em vez de um valor absoluto de real/falso [13]. Outra contribuição chave foi o **aprimoramento da perda perceptual, utilizando features das camadas VGG antes da ativação** (em vez de após a ativação, como no SRGAN), o que resultou em bordas mais nítidas e melhor consistência de brilho [13, 10]. Song et al. [10] sugerem que a combinação de features VGG e ResNet (Dual Perceptual Loss) pode trazer propriedades complementares para a reconstrução de texturas. O ESRGAN obteve a primeira colocação no desafio PIRM2018-SR, destacando-se em qualidade perceptual [13].

Mais recentemente, o **Real-ESRGAN** [12], também desenvolvido por Wang et al., foi proposto para enfrentar o desafio da super-resolução em cenários do mundo real, onde as degradações são complexas e desconhecidas. Para isso, o Real-ESRGAN introduziu um **processo de modelagem de degradação de alta ordem** para simular as complexas degradações do mundo real, incluindo artefatos de *ringing* e *overshoot* através do uso de filtros *sinc* [12]. Além disso, a arquitetura do discriminador foi aprimorada para um **discriminador U-Net com normalização espectral (SN)**, o que aumenta a capacidade discriminativa e estabiliza a dinâmica de treinamento, ajudando a suprimir artefatos excessivamente nítidos [12]. Esses avanços são comparados neste trabalho, com foco em implementações práticas.

3 Metodologia

3.1 Configuração do Experimento

O experimento foi conduzido no ambiente Google Colab. Para gerenciar os recursos computacionais disponíveis (GPU, RAM e tempo de execução), utilizou-se um subconjunto do dataset DIV2K [1], com 100 imagens de treino e 20 imagens de validação. As imagens de alta resolução (HR) foram redimensionadas para 128x128 pixels, e as correspondentes imagens de baixa resolução (LR) para 32x32 pixels, resultando em um fator de escala de 4×. O tamanho do mini-batch foi definido como 4, uma escolha que otimiza o uso da memória da GPU para o treinamento de redes profundas. O treinamento foi limitado a 5 épocas, divididas em 3 épocas de pré-treinamento com perda pixel-wise e 2 épocas de treinamento adversarial.

3.2 Arquiteturas

As três arquiteturas GAN comparadas (SRGAN, ESRGAN e Real-ESRGAN) consistem de um gerador (G) e um discriminador (D) que são treinados de forma adversarial.

- **SRGAN** [7]:
  - **Gerador (G_SRGAN)**: Emprega uma **rede residual profunda (ResNet)** com conexões de salto (*skip-connections*) para mitigar o problema do gradiente desvanecente [7, 4]. Os blocos residuais são compostos por camadas convolucionais com kernels de 3×3 e 64 mapas de características, seguidas por camadas de **Batch Normalization (BN)** e funções de ativação **ParametricReLU (PReLU)** [7]. O aumento da resolução é realizado por camadas de convolução sub-pixel [7, 8].
  - **Discriminador (D_SRGAN)**: É uma rede em estilo **VGG** [9], composta por oito camadas convolucionais que aumentam progressivamente o número de filtros (de 64 para 512) com kernels de 3×3 [7]. Utiliza ativação **LeakyReLU** ($\alpha = 0.2$) e evita o uso de max-pooling, empregando convoluções strided para reduzir a resolução da imagem [7]. A camada de saída consiste em duas camadas densas e uma função de ativação Sigmoid final, que produz a probabilidade de a imagem ser real [7].
- **ESRGAN** [13]:
  - **Gerador (G_ESRGAN)**: Substitui os blocos residuais do SRGAN pelos **Residual-in-Residual Dense Blocks (RRDB)**, que combinam multi-níveis de rede residual com conexões densas para aumentar a capacidade da rede [13]. Uma modificação crucial é a **remoção de todas as camadas de Batch Normalization (BN)**. Esta decisão foi tomada porque as camadas BN podem introduzir artefatos e limitar a capacidade de generalização, especialmente em redes mais profundas e treinadas com o *framework* GAN [13]. Para facilitar o treinamento de redes muito profundas sem BN, técnicas como **escalonamento residual (residual scaling)** (multiplicando os residuais por uma constante como 0.2) e inicialização menor foram empregadas [13].
  - **Discriminador (D_ESRGAN)**: Adota o conceito de **Relativistic average GAN (RaGAN)** [5]. Diferente do discriminador padrão que julga se uma imagem é real ou falsa, o RaGAN prevê se uma imagem é **relativamente mais realista do que outra** [13]. Esta abordagem simétrica permite que o gerador se beneficie dos gradientes tanto dos dados gerados quanto dos dados reais, o que ajuda na recuperação de bordas mais nítidas e texturas mais detalhadas [13].
  - **Perda Perceptual (L_perceptual)**: O ESRGAN aprimora a perda perceptual utilizada no SRGAN. Em vez de usar features após as camadas de ativação da rede VGG pré-treinada, o ESRGAN utiliza **features antes da ativação** [13]. Isso é motivado pelo fato de que features ativadas podem ser muito esparsas, fornecendo supervisão fraca, e o uso de features pós-ativação pode levar a inconsistências de brilho. A utilização de features pré-ativação proporciona supervisão mais forte para consistência de brilho e recuperação de textura, resultando em bordas mais nítidas e resultados visualmente mais agradáveis [13, 10]. A ideia de uma **Dual Perceptual Loss**, combinando perdas baseadas em VGG e ResNet, é mencionada como uma forma de explorar a complementaridade das características extraídas por diferentes redes pré-treinadas, o que pode melhorar a capacidade de recuperação de detalhes de textura [10].
- **Real-ESRGAN** [12]:
  - **Gerador (G_RealESRGAN)**: A arquitetura do gerador do Real-ESRGAN é idêntica à do ESRGAN, utilizando os mesmos blocos RRDB profundos [12]. Para fatores de escala menores (como 2× ou 1×, embora o experimento focado em 4×), o modelo emprega uma operação de **pixel-unshuffle** na entrada. Essa operação reduz o tamanho espacial da imagem e rearranja a informação para a dimensão do canal, otimizando o consumo de memória da GPU e os recursos computacionais durante o processamento [12].
  - **Discriminador (D_RealESRGAN)**: O discriminador é significativamente aprimorado para um design **U-Net com conexões de salto (skip-connections)** [12]. Este design permite que o discriminador forneça **feedback detalhado por pixel** ao gerador, o que é crucial para lidar com um espaço de degradação mais amplo e complexo [12]. Além disso, inguinal **normalização espectral (Spectral Normalization - SN)** é aplicada para estabilizar a dinâmica de treinamento, que pode se tornar instável com a estrutura U-Net e degradações complexas, e

para atenuar artefatos de super-nitidez e indesejados induzidos pelo treinamento GAN [12].
– **Modelagem de Degradação**: Um aspecto fundamental do Real-ESRGAN é sua capacidade de lidar com imagens degradadas do mundo real. Isso é conseguido através de um **processo de modelagem de degradação de alta ordem**, que simula uma combinação mais complexa de procedimentos de degradação (como múltiplos estágios de desfoque, ruído e compressão JPEG) em comparação com o modelo de degradação clássico de primeira ordem [12]. O uso de **filtros *sinc*** também é incorporado para modelar artefatos comuns de *ringing* e *overshoot*, que frequentemente aparecem em imagens reais devido à perda de frequências altas [12].

3.3 Treinamento

Os modelos foram treinados utilizando o otimizador **Adam**, com os hiperparâmetros $\beta_1 = 0.9$, $\beta_2 = 0.999$, e uma taxa de aprendizado (lr) de 0.0001 [6]. As funções de perda para o gerador foram definidas como uma combinação de perda de conteúdo, perda adversarial e perda perceptual, conforme a equação:

$$\text{Loss}_G = \lambda_{\text{content}} \cdot \text{L1} \ \lambda_{\text{adversarial}} \cdot \text{BCE} \ \lambda_{\text{perceptual}} \cdot \text{VGG},$$

onde L1 refere-se à perda Mean Absolute Error (MAE) para o conteúdo, BCE (Binary Cross-Entropy) é utilizada para a perda adversarial, e VGG representa a perda perceptual baseada em características extraídas da rede VGG pré-treinada. Os pesos aplicados a cada componente da perda foram: $\lambda_{\text{content}} = 1.0$, $\lambda_{\text{adversarial}} = 0.01$, e $\lambda_{\text{perceptual}} = 0.1$. A estratégia de treinamento incluiu um estágio de **pré-treinamento** do gerador com a perda L1 para convergir para resultados visualmente agradáveis e evitar ótimos locais indesejados na fase adversarial [12]. Posteriormente, o **treinamento adversarial** foi conduzido, com atualizações alternadas entre o gerador e o discriminador para promover a competição entre eles [3]. Checkpoints dos modelos foram salvos periodicamente para permitir a retomada do treinamento e a avaliação do desempenho em diferentes estágios.

3.4 Avaliação

As métricas **PSNR (Peak Signal-to-Noise Ratio)** e **SSIM (Structural Similarity Index)** foram calculadas no conjunto de teste para avaliar quantitativamente a qualidade das imagens super-resolvidas. Enquanto o PSNR é uma métrica amplamente utilizada que mede a similaridade pixel-a-pixel, o SSIM busca quantificar a similaridade estrutural, brilho e contraste, muitas vezes correlacionando-se melhor com a percepção humana [14]. Para uma análise qualitativa, foram geradas visualizações de uma imagem de teste, apresentada como uma única figura contendo a comparação entre a imagem de baixa resolução (LR), as saídas super-resolvidas de cada modelo (SRGAN, ESRGAN, Real-ESRGAN) e a imagem de alta resolução (HR) original (ground truth). Esta comparação visual permitiu a observação direta de detalhes finos, texturas e a presença de artefatos em cada reconstrução.

4 Resultados

4.1 Resultados Quantitativos

A Tabela 1 apresenta as métricas PSNR e SSIM médias no conjunto de validação após 5 épocas de treinamento.

Table 1: Comparação Quantitativa das Arquiteturas

Modelo	PSNR (dB)	SSIM
SRGAN	18.69	0.4809
ESRGAN	19.14	0.5189
Real-ESRGAN	18.72	0.5511

Os resultados indicam que o Real-ESRGAN obteve o maior SSIM (0.5511), o que sugere uma melhor similaridade estrutural com as imagens originais de alta resolução, refletindo sua capacidade de restaurar texturas mais fidedignas. Por outro lado, o ESRGAN apresentou o maior PSNR (19.14 dB). O SRGAN teve o menor desempenho em ambas as métricas, o que é consistente com a tendência de métodos otimizados para MSE (como seu pré-treinamento) de produzir imagens mais suavizadas e com menos detalhes perceptuais.

4.2 Resultados Qualitativos

A Figura 1 mostra a comparação qualitativa de uma imagem de teste.

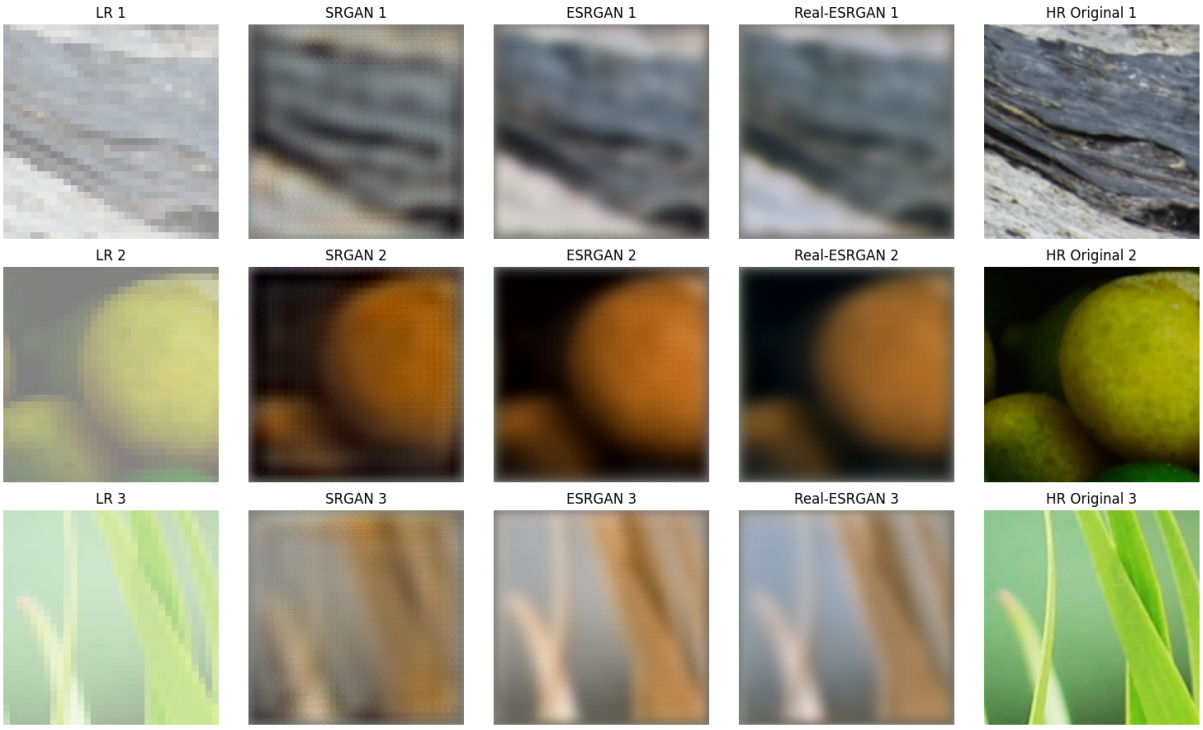

Figure 1: Comparação qualitativa de uma imagem de teste, contendo a imagem de baixa resolução (LR), saídas super-resolvidas (SRGAN, ESRGAN, Real-ESRGAN) e a imagem de alta resolução (HR).

5 Conclusão

Este trabalho comparou três arquiteturas de Redes Generativas Adversariais (GANs) – SRGAN, ESRGAN e Real-ESRGAN – na tarefa de super-resolução de imagens. A avaliação foi realizada utilizando um subconjunto do dataset DIV2K em um ambiente com **limitações computacionais** (Google Colab), o que restringiu o tamanho do dataset e o número de épocas de treinamento. Quantitativamente, o Real-ESRGAN superou os outros modelos em SSIM (0.5511), indicando melhor similaridade estrutural, enquanto o ESRGAN alcançou o maior PSNR (19.14 dB). Qualitativamente, tanto o ESRGAN quanto o Real-ESRGAN geraram imagens mais nítidas e com maior riqueza textural em comparação com o SRGAN. O Real-ESRGAN, em particular, destacou-se na recuperação de texturas detalhadas e realistas, uma vantagem atribuída ao seu discriminador U-Net e à aplicação de normalização espectral. Os resultados obtidos, apesar das restrições de recursos, demonstraram de forma prática as diferenças arquiteturais e os avanços de desempenho entre as abordagens. Trabalhos futuros podem explorar a métrica LPIPS (Learned Perceptual Image Patch Similarity) para uma avaliação perceptual mais robusta e investir na otimização de hiperparâmetros utilizando ferramentas como Weights & Biases para refinar ainda mais o desempenho dos modelos.

References

[1] Eirikur Agustsson and Radu Timofte. 2017. Ntire 2017 challenge on single image super-resolution: dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 126–135.

[2] Huining Feng. 2024. Review of gan-based image super-resolution techniques. *Proceedings of the Quantum Machine Learning: Bridging Quantum Physics and Computational Simulations - CONFMPCS 2024*, 134.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

[5] Alexia Jolicoeur-Martineau. 2018. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*.

[6] Diederik P Kingma and Jimmy Ba. 2015. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*.

[7] Christian Ledig et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681–4690.

[8] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1874–1883.

[9] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[10] Jie Song, Huawei Yi, Wenqian Xu, Xiaohui Li, Bo Li, and Yuanyuan Liu. 2024. Dual perceptual loss for single image super-resolution using esrgan. *Neural Computing and Applications*, 1–22.

[11] Chunwei Tian, Xuanyu Zhang, Qi Zhu, Bob Zhang, and Jerry Chun-Wei Lin. 2024. Generative adversarial networks for image super-resolution: a survey. *ACM Computing Surveys (CSUR)*, 1, 1, 1–31.

[12] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. 2021. Real-esrgan: training real-world blind super-resolution with pure synthetic data. *arXiv preprint arXiv:2107.03055*.

[13] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. 2018. Esrgan: enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, 0–0.

[14] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13, 4, 600–612.

Received 16 July 2025