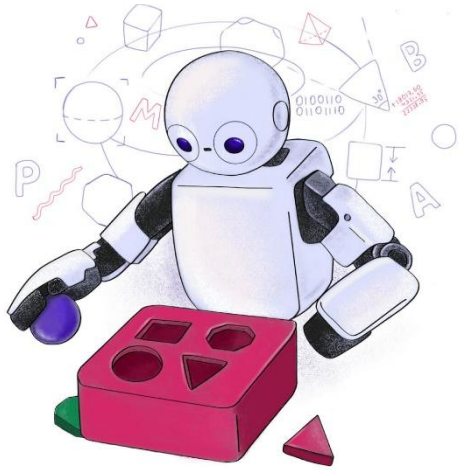
TP558 - Tópicos avançados em Machine Learning:

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



Inatel

Jonas Vilasboas Moreira
jonasmoreira@dtel.inatel.br

Introdução

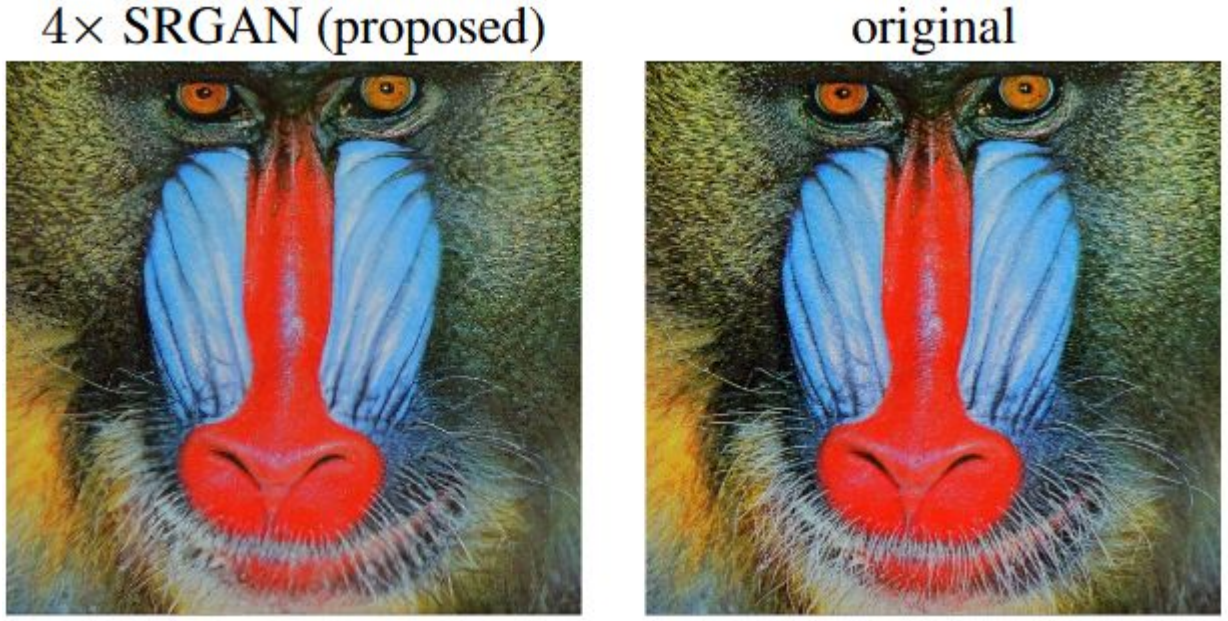
O SRGAN é um tipo de GAN que **pega uma imagem de baixa qualidade e aumenta sua resolução**, deixando-a **mais nítida e realista**.

Em vez de apenas "esticar" a imagem como faz o zoom comum (que deixa tudo borrado), ele **aprende como deveriam ser os detalhes que se perderam**, como texturas, fios de cabelo, folhas, ou traços finos, e os **recria de forma convincente**, fazendo a imagem parecer uma foto verdadeira e não algo artificial.

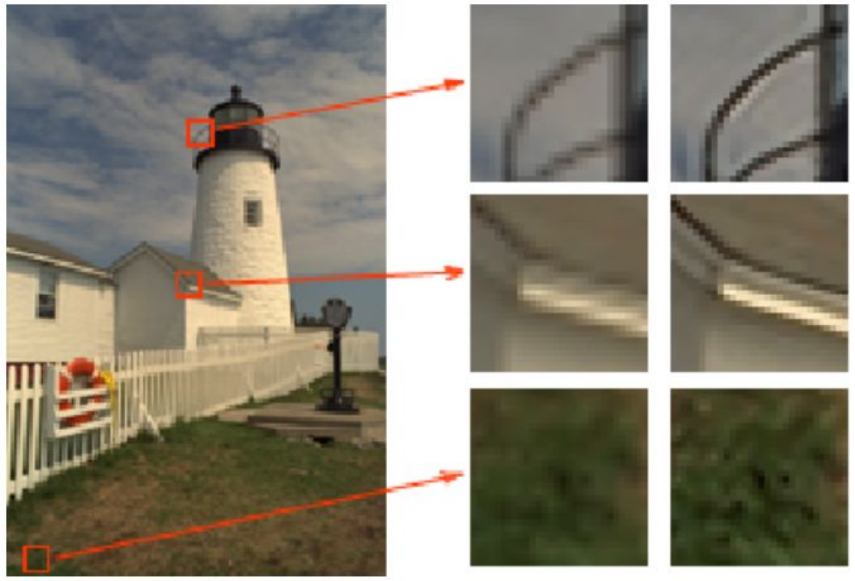
Fundamentação teórica

Super Resolution (SR)

Reconstruir uma imagem de **alta resolução (HR)** a partir de uma **imagem de baixa resolução (LR)**.



Fundamentação teórica



Métodos Clássicos de SR

- **Métodos de Interpolação**
  - **Exemplos:** Bilinear, Bicubic, Lanczos

  - **Ideia principal:**

    Esses métodos não “inventam” novos detalhes.

    Eles **usam apenas os pixels existentes** para calcular valores intermediários e preencher a imagem ampliada. Por exemplo, se você tem um pixel vermelho e um azul, o método gera um roxo entre eles — uma média simples.

Fundamentação teórica

Métodos Clássicos de SR

- **Bilinear, Bicubic, Lanczos**
  - **Vantagens:**
    - São **muito rápidos** e fáceis de implementar.
    - Funcionam bem para **pequenos aumentos** (2× ou menos).
  - **Desvantagens:**
    - Não conseguem recriar **texturas finas** (cabelos, folhas, padrões).
    - As imagens ficam **borradas, com bordas suaves** e aparência artificial.
    - Falham totalmente em grandes ampliações (4×, 8×).

Fundamentação teórica

Métodos Clássicos de SR

- **Métodos Baseados em Exemplos (ou Patches)**
  - **Ideia principal:**

    Esses métodos usam **bancos de dados de imagens** contendo pares de exemplos:
    - Um patch (pequeno pedaço) de baixa resolução (LR)
    - Seu correspondente de alta resolução (HR)

Fundamentação teórica

Métodos Clássicos de SR

- **Métodos Baseados em Exemplos (ou Patches)**
- **Durante a reconstrução:**
  - Para cada região da imagem LR, o algoritmo **procura patches semelhantes** no banco.
  - Substitui essa região por um patch HR compatível.
  - No final, **combina** todos os patches para formar a imagem ampliada.

(a) Input LR image (64×32)

(b) SR result (1024×512) using DDNM, by patch

Fundamentação teórica

SRCNN - CNN para SR

O que ela fazia

- Pegava uma imagem **já ampliada** por interpolação bicúbica.
- Passava essa imagem por **três camadas convolucionais**.
- Cada camada aprendia a **melhorar gradualmente os detalhes**:
  1. Extrair características da imagem (texturas, bordas);
  2. Fazer mapeamento para um espaço de alta resolução;
  3. Reconstituir a imagem final (mais nítida).

Fundamentação teórica

SRCNN - CNN para SR

Resultados

- Obteve **PSNR muito maior** do que os métodos clássicos.
- Mostrou que **redes neurais podiam aprender padrões de textura e borda** de forma automática.
- **Limitação:**
  - Rede **muito rasa (só 3 camadas)** → enxerga apenas **pouco contexto** da imagem. Isso faz com que detalhes finos **ainda fiquem suavizados**.
  - O modelo melhora os números (PSNR), mas **ainda não gera imagens realistas**.

Fundamentação teórica

PSNR (Peak Signal-to-Noise Ratio, ou **Relação Pico Sinal-Ruído**)

O PSNR é calculado a partir do MSE (erro médio) entre as duas imagens (original e SR).

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(I_{HR}(i) - I_{SR}(i))^2$$

E o PSNR é dado por:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right)$$

MAX é o valor máximo possível de intensidade de pixel (ex. 255 em imagens de 8 bits).

Fundamentação teórica

Avanços com Redes Mais Profundas

Modelos seguintes:

- **VDSR (2015):** rede mais profunda (20 camadas).
- **DRCN (2016):** convoluções recursivas → mais contexto.
- **ESPCN (2016):** usa *sub-pixel convolution* para upscaling eficiente.
- **Resultados:**
  - Maior PSNR
  - Texturas ainda artificiais

Fundamentação teórica

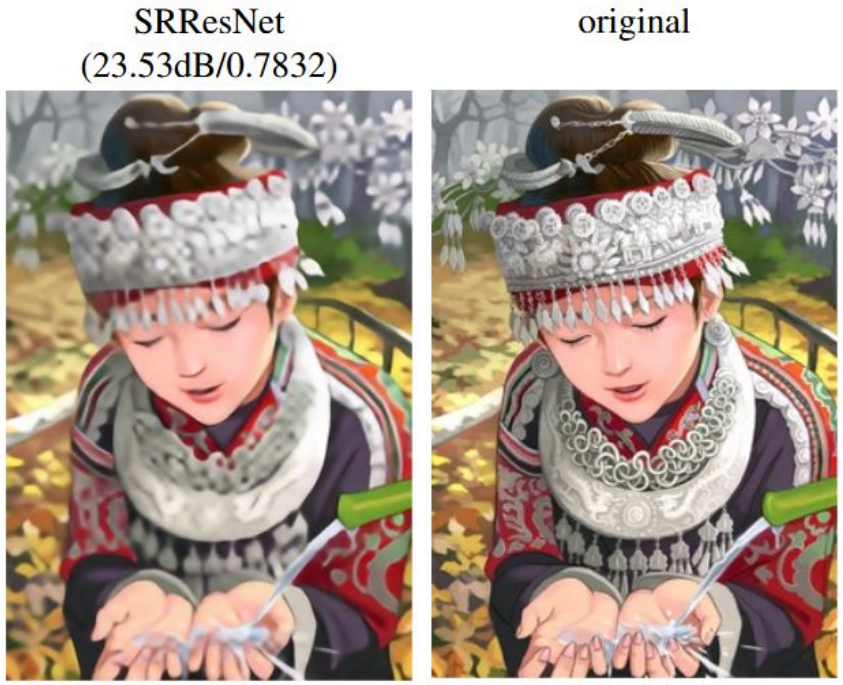
O Problema do MSE

Perda comum:

$$\text{MSE} = \frac{1}{N}\sum(I_{HR} - I_{SR})^2$$

Consequência:

- O modelo “faz a média” de várias soluções possíveis.
- Resultado: imagens **suaves e borradas**, sem texturas finas.



Fundamentação teórica

Métricas Tradicionais vs. Percepção Humana

PSNR e **SSIM** (Structural Similarity Index Measure — ou, Índice de Similaridade Estrutural)

- Avaliam similaridade pixel a pixel.
- **Não refletem** o que o olho humano considera “realista”.

Alta PSNR ≠ Imagem bonita.

Fundamentação teórica

Perdas Perceptuais (VGG Loss)

- Proposta de usar *feature maps* da **VGG19** para medir similaridade.
- Em vez de comparar pixels, comparam **formas, bordas e texturas internas**.
- **Benefício:**
  - Imagens mais próximas do que o cérebro percebe como “real”.

Essa ideia de “perda perceptual” inspirou diretamente o **SRGAN**.

Fundamentação teórica

GAN (Generative Adversarial Network)

Duas redes que aprendem juntas:

- **Gerador (G):** tenta criar imagens realistas.
- **Discriminador (D):** tenta detectar se a imagem é real ou gerada.

Processo:

G melhora ao enganar D → D melhora ao detectar G.

Resultado:

O gerador aprende a **criar imagens que parecem reais**.

Arquitetura e funcionamento

SRGAN - Introdução

- Primeira GAN aplicada à super-resolução perceptual.

- Substitui MSE por uma **perda perceptual** baseada em **mapas de características**.
  - Imagens mais naturais e realistas.

- **Avaliação Experimental**
  - Teste de opinião média (**MOS**) com três datasets públicos.
    - **SRGAN supera todos os métodos anteriores** em realismo perceptual, mesmo com menor PSNR.

Arquitetura e funcionamento

Método

- Na super-resolução de imagem, o objetivo é estimar uma imagem de alta resolução (**ISR**) a partir de uma imagem de baixa resolução (**ILR**).

- No **SRGAN**, **ILR** é a versão de baixa resolução da sua correspondente **IHR** (imagem original de alta resolução).

Arquitetura e funcionamento

Método

- As **IHR** só estão disponíveis durante o **treinamento**.

- Durante o treinamento, **ILR** é obtida aplicando-se um **filtro Gaussiano** à **IHR**, seguido de uma **operação de reamostragem (downsampling)** com um **fator de redução r**.

- Para uma imagem com **C canais de cor**, **ILR** é representada por um **tensor de tamanho W × H × C**, enquanto **IHR** e **ISR** têm dimensão **rW × rH × C**.

Arquitetura e funcionamento

Método

- O objetivo final é **treinar uma função geradora G** que, para uma imagem LR de entrada, **estime sua contraparte HR correspondente**.

- Para isso, treina-se uma rede neural convolucional (**CNN**) feed-forward chamada **gerador**, parametrizada pelos pesos e vieses de uma rede profunda com **L camadas**.

- Os parâmetros dessa rede são obtidos **otimizando uma função de perda específica de super-resolução (ISR)**.

Arquitetura e funcionamento

Método

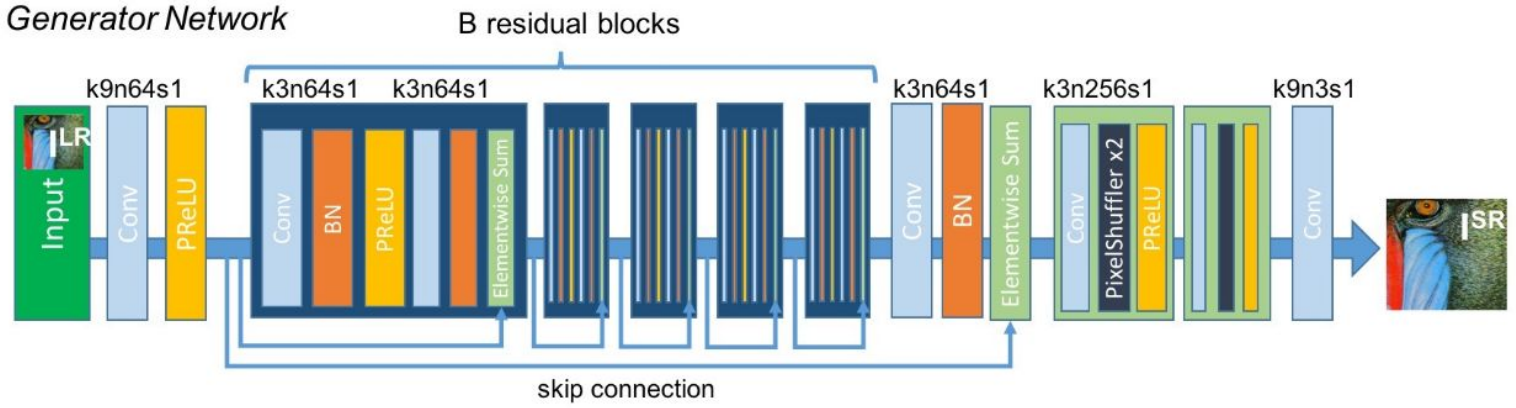
- Neste trabalho, os autores projetam uma **função de perda perceptual (lSR)** composta por uma **combinação ponderada de diferentes componentes de perda**, cada um modelando **características desejáveis distintas** da imagem super-resolvida reconstruída.

Arquitetura e funcionamento

Generator Network

Baseada em uma **ResNet profunda** com **B blocos residuais idênticos**. Cada bloco contém:

a. **Duas convoluções 3×3** com **64 feature maps**
b. **Batch Normalization** em cada camada
c. **Ativação ParametricReLU (PReLU)**



Arquitetura e funcionamento

Generator Network

- **Upscaling (aumento da resolução)**
  a. Feito por **duas camadas de convolução subpixel treináveis**

- Substituem a interpolação bicúbica, tornando o processo **aprendido** e mais **eficiente**.

Arquitetura e funcionamento

Discriminator Network

Tem como objetivo avaliar se uma imagem **SR** é **real (HR verdadeira)** ou **gerada pelo modelo (falsa)**.

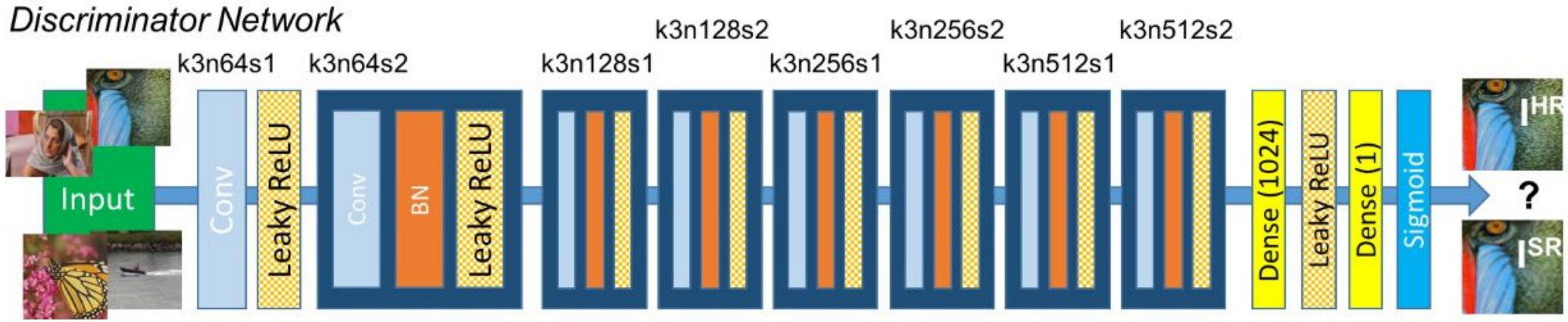
Arquitetura e funcionamento

Discriminator Network

- **8 camadas convolucionais 3×3**, com número de filtros crescendo:
  64 → **128** → **256** → **512** (como na **VGG19**).
- A cada aumento de filtros, usa-se **convoluções com stride**
  para reduzir a resolução da imagem.
- Os **512 feature maps finais** alimentam:
  - **2 camadas densas**, seguidas de
  - **1 função sigmoide** → produz **probabilidade** de “imagem real”.

Arquitetura e funcionamento

Discriminator Network



Arquitetura e funcionamento

Discriminator Network - Resumo

- D aprende o que torna uma imagem *realista*.
- G tenta enganar D $\rightarrow$ gera texturas e padrões cada vez mais naturais.
- O equilíbrio entre os dois cria o **realismo perceptual** do SRGAN.

Arquitetura e funcionamento

Função de Perda Perceptual (Perceptual Loss)

A chave do sucesso do SRGAN:

- Ele não tenta reduzir o erro numérico — ele tenta *enganar nossos olhos*.
- A função de perda perceptual combina duas ideias:
  a. a perda de conteúdo, baseada nas representações da VGG, que garante coerência visual;
  b. e a perda adversarial, que empurra o modelo para o espaço das imagens realistas.
- Esse equilíbrio produz o realismo foto-perceptual que nenhum método anterior alcançava.

Arquitetura e funcionamento

Função de Perda Perceptual (Perceptual Loss)

$$l_{SR} = l_{SR}^{X} + 10^{-3} l_{SR}^{Gen}$$

onde:

- l_{SR}^{X} → **Content Loss** (*perda de conteúdo*)
- l_{SR}^{Gen} → **Adversarial Loss** (*perda adversarial*)

- **Content Loss:** compara a imagem gerada com a real no **espaço de features da VGG**, não nos pixels. Garante fidelidade visual, sem distorções.
- **Adversarial Loss:** força o gerador a **produzir imagens plausíveis** para o discriminador. Intentiva realismo perceptual (texturas naturais).
- **Perceptual Loss = equilíbrio** entre **fidelidade** e **realismo**.

Arquitetura e funcionamento

Perda de Conteúdo (Content Loss)

- O MSE mede apenas diferenças ponto a ponto, o que leva a imagens nítidas mas sem textura.
- O SRGAN substitui isso por uma perda baseada nas ativações da VGG19 — assim, o modelo aprende o que é visualmente parecido com a imagem real, não o que é numericamente idêntico.
- Essa foi a virada que permitiu gerar texturas e detalhes perceptualmente realistas.”

Arquitetura e funcionamento

Perda VGG

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

- Baseada nos **mapas de ativação (feature maps)** da rede **VGG19**.
- **$\varphi_{i,\square}$:** camada de ativação ReLU antes da *i-ésima* camada de *max pooling*.
- Mede **distância euclidiana no espaço de features**, não nos pixels.
- Captura **semelhanças perceptuais**: formas, bordas, texturas.

Arquitetura e funcionamento

Perda Adversarial (Adversarial Loss)

Além da perda de conteúdo (VGG), o SRGAN adiciona uma **componente generativa (Adversarial Loss)** à função de perda total.

- Isso **encoraja o gerador (G)** a criar imagens que **pareçam naturais**, situadas no **manifold das imagens reais**.
- O objetivo é **enganar o discriminador (D)**.

O **manifold** é o “espaço das imagens possíveis do mundo real”.

O **SRGAN** tenta gerar imagens **que fiquem sobre esse manifold**, isto é, que pareçam plausíveis e naturais, não artificiais.

Arquitetura e funcionamento

Perda Adversarial (Adversarial Loss)

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

- $D_{\theta_D}(G_{\theta_G}(I_{LR}))$:
  probabilidade de o discriminador acreditar que a imagem gerada é real.
- Minimizar $-\log D(G(I_{lr}))$ → melhora o gradiente e a estabilidade do treinamento.

Arquitetura e funcionamento

Perda Adversarial (Adversarial Loss)

- A perda adversarial é o que dá ‘vida’ às imagens do SRGAN.

- Em vez de só comparar pixels, o gerador tenta enganar o discriminador, aprendendo o que faz uma imagem parecer real para um observador — texturas, ruído, variações sutis de luz.

- Esse termo é o responsável pelo realismo foto-perceptual das imagens finais.

Treinamento e otimização

Bases de Dados Utilizadas

- **Set5** → pequeno, imagens clássicas (faces, paisagens)
- **Set14** → maior diversidade de cenas e texturas
- **BSD100** → subconjunto de teste do BSD300, com imagens naturais
- **Fator de ampliação:** 4× (redução de 16× em número de pixels)

Treinamento e otimização

Procedimentos de Avaliação

- Métricas: **PSNR [dB]** e **SSIM** (Índice de Similaridade Estrutural)
- Calculadas apenas no **canal Y (luminância)**
- **Recorte central** + **remoção de 4 pixels nas bordas**
- Implementação via **pacote Daala** (um codec experimental de vídeo (um tipo de codificador/decodificador), criado com o objetivo de explorar novas técnicas de compressão sem patentes.)

Treinamento e otimização

Métodos de Comparação

- **Interpolação:** *Nearest Neighbor*, *Bicubic*
- **Redes:** *SRCNN*, *SelfExSR*, *DRCN*
- **Propostos:** *SRResNet* (MSE, VGG/2.2) e *SRGAN* (todas as variantes)

Treinamento e otimização

Configuração de Treinamento do SRGAN

Dados de Treinamento

- 350 mil imagens aleatórias do ImageNet
- Imagens diferentes das usadas para teste
- Downsampling bicúbico (r = 4) para gerar versões LR
- Formato: BGR, C = 3 canais (RGB)

Treinamento e otimização

Configuração de Treinamento do SRGAN

Pré-processamento

- Mini-batches: 16 subimagens 96×96 (HR) aleatórias
- Gerador totalmente convolucional ⟶ aceita qualquer tamanho de imagem
- Escala de valores:
  - LR ⟶ [0, 1]
  - HR ⟶ [-1, 1]
- Perda MSE calculada sobre [−1, 1]

Treinamento e otimização

Configuração de Treinamento do SRGAN

Ajustes Técnicos

- VGG feature maps reescalados por 1 / 12.75 → fator ≈ 0.006
  - garante que VGG Loss e MSE Loss fiquem em mesma escala

- Otimizador: Adam ($\beta_1 = 0.9$)

Treinamento e otimização

Configuração de Treinamento do SRGAN

Treinamento

- **SRResNet:**
  - LR = 10^{-4}, 10^{6} iterações
- SRGAN:
  - Inicializado com SRResNet
  - 10^{5} iterações @ 10^{-4}, depois 10^{5} @ 10^{-5}
  - Atualizações alternadas G ↔ D (k = 1)
- Gerador: 16 blocos residuais (B = 16)
- BatchNorm congelado no teste ⟶ saída determinística

Treinamento e otimização

Configuração de Treinamento do SRGAN

Implementação

- Frameworks: **Theano** e **Lasagne**
- Hardware: **GPU NVIDIA Tesla M40**

Treinamento e otimização

Avaliação Perceptual (MOS Test)

Objetivo - Avaliar o realismo perceptual das imagens geradas com base em opiniões humanas (Mean Opinion Score).

Procedimento Experimental

- 26 avaliadores humanos
- Atribuíram notas de 1 (ruim) a 5 (excelente)
- 12 versões de cada imagem nos conjuntos Set5, Set14 e BSD100

Treinamento e otimização

Métodos avaliados

NN, Bicubic, SRCNN, SelfExSR, DRCN, ESPCN, SRResNet (MSE, VGG22), SRGAN (MSE, VGG22, VGG54) e HR original

- Cada avaliador julgou 1.128 imagens (amostras aleatórias)
- Calibração:
  - NN = nota 1
  - HR = nota 5
- Teste piloto:
  - Repetição de 10 imagens (BSD100)
  - Resultados consistentes, sem diferença significativa

Treinamento e otimização

Resultados

- Avaliadores consistentes: NN sempre = 1, HR sempre = 5
- SRGAN (VGG54) obteve as maiores notas médias (MOS)
- Superou todos os métodos anteriores em percepção de realismo

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**	4.32
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**	4.32
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**	4.46

Treinamento e otimização

Comparação de Funções de Perda

- MSE → maior PSNR, mas texturas suaves e pouco realistas.
- VGG-based losses → menor PSNR, porém maior qualidade perceptual.
- SRGAN-VGG54 → melhores detalhes e texturas.
- Set14: SRGAN-VGG54 superou significativamente todas as variantes segundo MOS.

	SRResNet-		SRGAN-		
Set5	MSE	VGG22	MSE	VGG22	VGG54
PSNR	32.05	30.51	30.64	29.84	29.40
SSIM	0.9019	0.8803	0.8701	0.8468	0.8472
MOS	3.37	3.46	3.77	3.78	3.58
Set14					
PSNR	28.49	27.19	26.92	26.44	26.02
SSIM	0.8184	0.7807	0.7611	0.7518	0.7397
MOS	2.98	3.15*	3.43	3.57	3.72*

Vantagens

- **Realismo perceptual sem precedentes**
  → Imagens super-resolvidas com **texturas naturais** e **detalhes foto-realistas**.

- **Primeiro uso bem-sucedido de GANs** em super-resolução
  → Abriu uma nova linha de pesquisa em **perceptual SR**.

- **Perda perceptual (VGG)** capta **similaridades visuais reais**, superando métricas tradicionais como PSNR e SSIM.

- **Arquitetura residual (ResNet)** melhora estabilidade e profundidade do treinamento.

- **Resultados validados por humanos (MOS)** → qualidade percebida comprovada.

Desvantagens

- **Treinamento complexo e instável**
  → Necessita ajuste fino entre **gerador e discriminador** (equilíbrio GAN).

- **Custo computacional elevado**
  → Rede profunda + adversarial → **alto tempo de treino e inferência**.

- **Dificuldade de controle**
  → Pode gerar **artefatos de alta frequência** ou **detalhes "alucinados"**.

- **Não otimizado para aplicações em tempo real**
  → Ineficiente para vídeo ou dispositivos embarcados.

- **Avaliação subjetiva (MOS)** → depende de observadores humanos.

Exemplo(s) de aplicação

Melhoria e Restauração de Imagens

- Aumento de resolução de **fotos antigas**, **filmes** e **vídeos**
- **Remasterização** de conteúdo visual para **4K / 8K**

Ciência e Tecnologia

- **Sensoriamento remoto** e **astronomia** → mais detalhes em imagens de satélite e telescópios
- **Microscopia e exames médicos** → ampliação de imagens com preservação de textura

Entretenimento e Segurança

- **Jogos e streaming** com imagens aprimoradas em tempo real
- **Reconstrução de imagens** em câmeras de segurança ou fotos borradas

Comparação com outros algoritmos

- **Métodos Clássicos:** Nearest Neighbor, Bicubic
  → Rápidos, mas geram **texturas borradas**.

- **CNNs Iniciais:** SRCNN, DRCN, ESPCN
  → Melhoram **PSNR**, mas produzem **imagens suaves**.

- **SRResNet (autores):** Alta **fidelidade numérica**, porém **baixa qualidade perceptual**.

- **SRGAN (proposto):** Texturas **foto-realistas**, **melhor avaliação humana (MOS)**, mesmo com PSNR menor.

Perguntas?

Referências

- Repositório com exemplo prático:
  - https://github.com/jonasvm/seminario-srgan/blob/main/srgan_demo.ipynb
- Quiz
  - https://forms.gle/KDmF9JZoDWGvViuc8

Referências

C. Ledig et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 105-114, doi: 10.1109/CVPR.2017.19.
keywords: {Image resolution;Signal resolution;Gallium nitride;Image reconstruction;Manifolds;Training;Network architecture}
Abstract: Despite the breakthroughs in accuracy and speed of single image super-resolution using faster and deeper convolutional neural networks, one central problem remains largely unsolved: how do we recover the finer texture details when we super-resolve at large upscaling factors? The behavior of optimization-based super-resolution methods is principally driven by the choice of the objective function. Recent work has largely focused on minimizing the mean squared reconstruction error. The resulting estimates have high peak signal-to-noise ratios, but they are often lacking high-frequency details and are perceptually unsatisfying in the sense that they fail to match the fidelity expected at the higher resolution. In this paper, we present SRGAN, a generative adversarial network (GAN) for image super-resolution (SR). To our knowledge, it is the first framework capable of inferring photo-realistic natural images for 4x upscaling factors. To achieve this, we propose a perceptual loss function which consists of an adversarial loss and a content loss. The adversarial loss pushes our solution to the natural image manifold using a discriminator network that is trained to differentiate between the super-resolved images and original photo-realistic images. In addition, we use a content loss motivated by perceptual similarity instead of similarity in pixel space. Our deep residual network is able to recover photo-realistic textures from heavily downsampled images on public benchmarks. An extensive mean-opinion-score (MOS) test shows hugely significant gains in perceptual quality using SRGAN. The MOS scores obtained with SRGAN are closer to those of the original high-resolution images than to those obtained with any state-of-the-art method.
url: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.19

Obrigado!