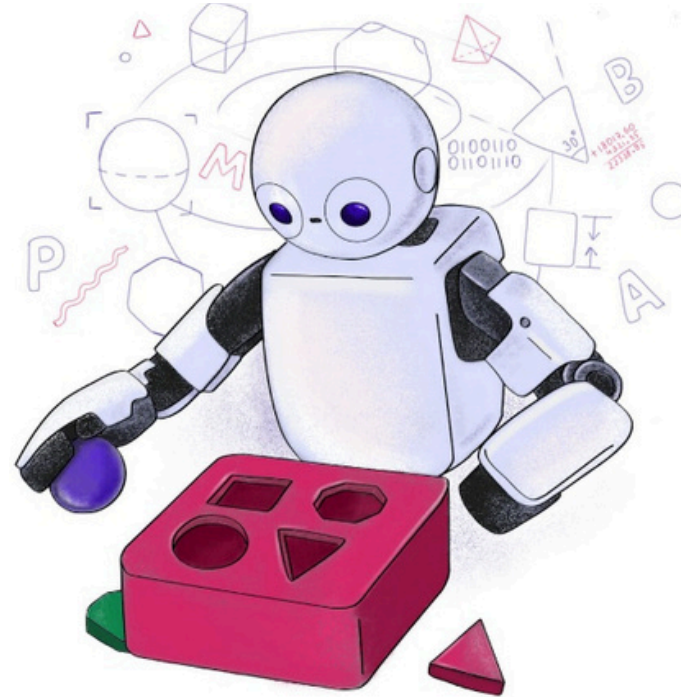


# TP558 - Tópicos avançados em Machine Learning: ***Enhanced Deep Residual Networks for Single Image Super-Resolution(EDSR)***



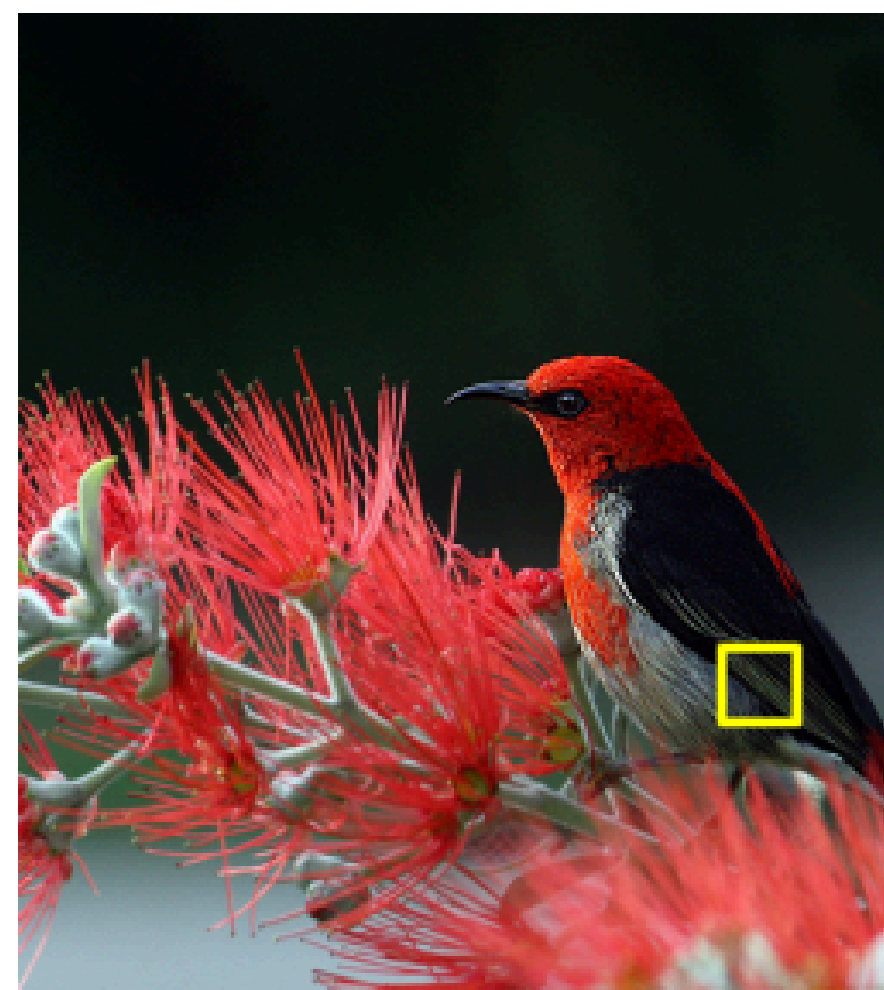
# Introdução

- O problema tratado é o de **Super-Resolução de Imagens (SR)**, em especial a **Super-Resolução de Imagem Única (SISR)**.
- **Objetivo:** transformar uma imagem de baixa resolução (LR) em uma imagem de alta resolução (HR), recuperando detalhes perdidos.
- **Desafio:** reconstruir texturas e bordas realistas a partir de pouca informação, múltiplas imagens podem corresponder a mesma versão degradada

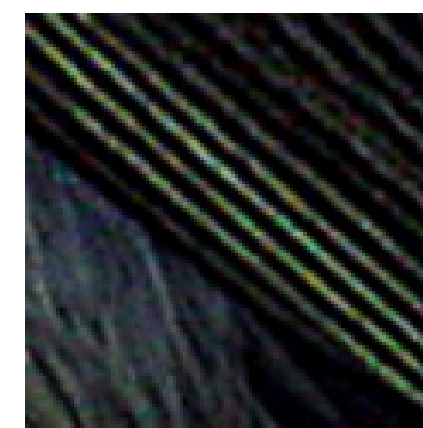
# Introdução

- Solução predominante foi métodos de interpolação, como o bicubic, mas geram imagens borradas.
- Avanço das CNNs → SRCNN, VDSR e SRResNet
- O **EDSR (Enhanced Deep Residual Networks for Single Image Super-Resolution)** é uma evolução direta dessas ideias.

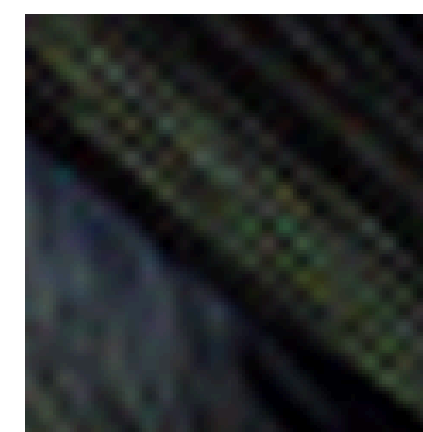
# Introdução



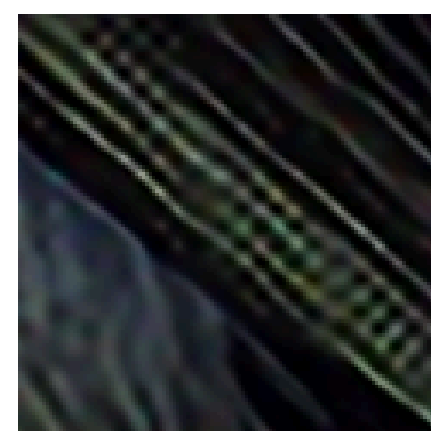
0853 from DIV2K [26]



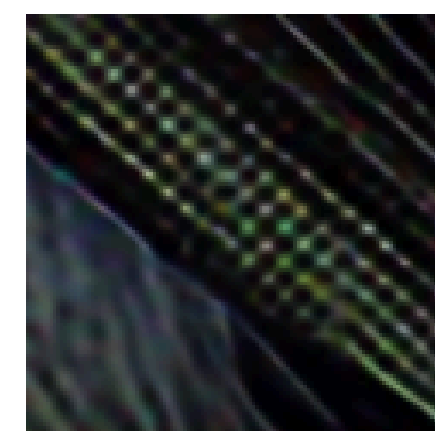
HR  
(PSNR / SSIM)



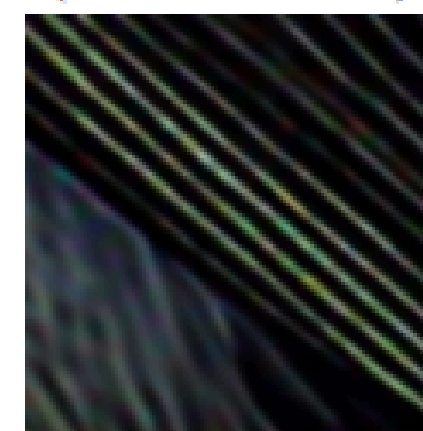
Bicubic  
(30.80 dB / 0.9537)



VDSR [11]  
(32.82 dB / 0.9623)



SRResNet [14]  
(34.00 dB / 0.9679)



**EDSR+ (Ours)**  
**(34.78 dB / 0.9708)**

## Fundamentação teórica

- **Super-resolução:** problema inverso, onde queremos recuperar informações perdidas após *downsampling*.
- Métodos clássicos (bilinear, bicubic): estimam os pixels a partir de combinações ponderadas de pixels vizinhos.
- **Redes neurais convolucionais (CNNs):** Surgimento em 2014, usadas para aprender a relação entre imagens de baixa e alta resolução. Treinamento complicado pela instabilidade do gradiente à medida que a rede cresce.

## Fundamentação teórica

- **ResNet (redes residuais):** utiliza conexões de atalho (skip connections) que facilitam o treinamento de redes mais profundas, permitindo que o gradiente flua diretamente entre camadas distantes da rede.
- Aplicar ResNet diretamente à SR não era ideal. Alguns componentes do ResNet, como batch normalization, normalizam a saída das convoluções para acelerar o treinamento, mas em SR restringem a flexibilidade do modelo.
- O EDSR parte da ideia de **ResNet + otimizações específicas para SR**, removendo componentes desnecessários (como Batch Normalization) e ajustando a rede para melhor estabilidade e desempenho.

# Fundamentação teórica

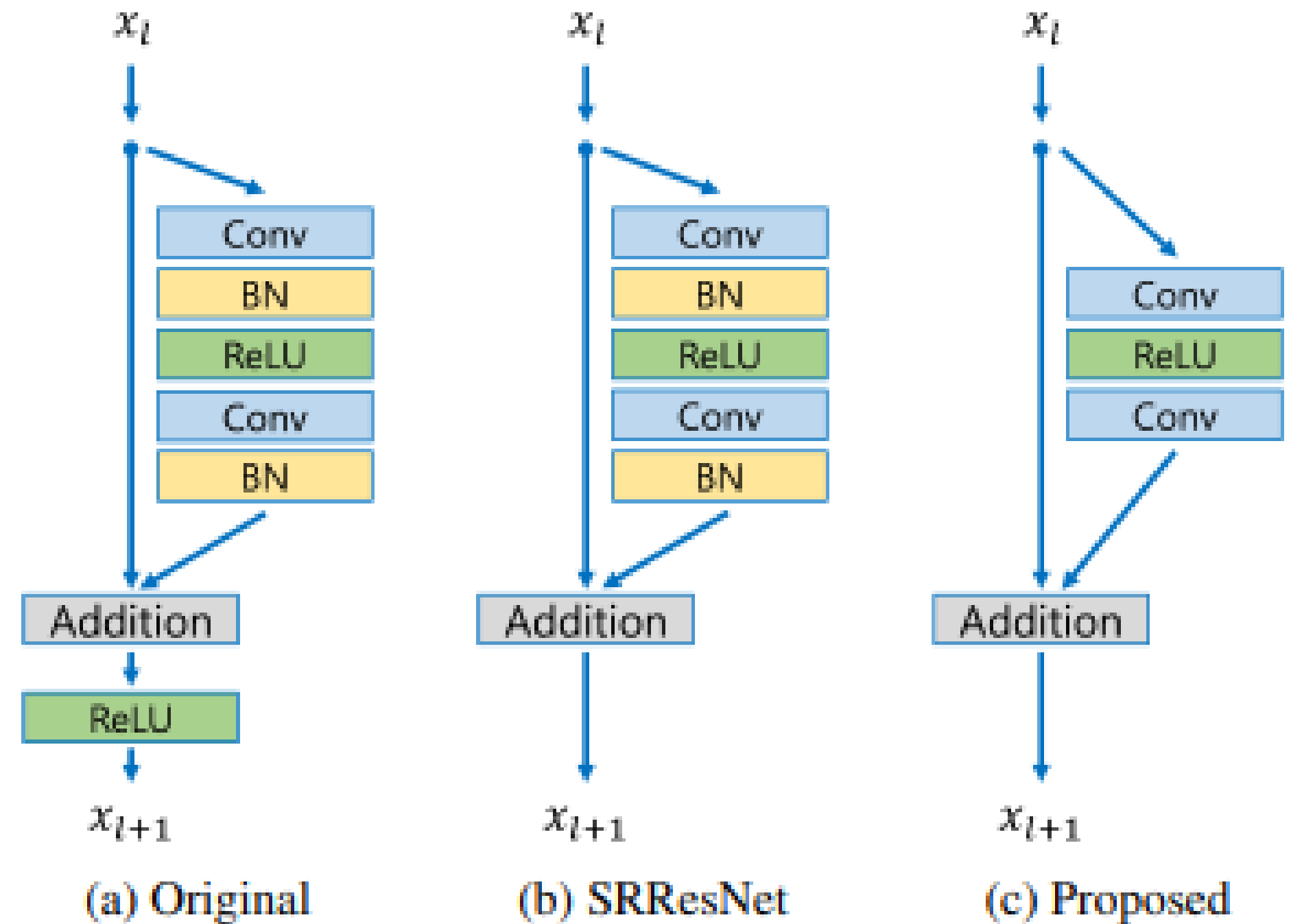


Figure 2: Comparison of residual blocks in original ResNet, SRResNet, and ours.

# Arquitetura e funcionamento

- O EDSR é composto por:
- **Blocos residuais modificados:** retiram a camada de *batch normalization* → isso economiza memória e melhora a performance. Consiste de duas camadas convolucionais com função de ativação ReLU, com conexão de atalho que soma a entrada do bloco à sua saída.
- **Residual scaling:** técnica que estabiliza o treinamento em redes muito grandes sem comprometer a convergência aplicando um fator de escala (0.1) dentro dos blocos residuais.



# Arquitetura e funcionamento

- O EDSR é composto por:
  - **EDSR (single-scale)**: rede projetada para um único fator específico de ampliação (ex: x2, x3, x4). Começa o treinamento com a versão x2 e reutiliza os pesos para treinar x3 e x4.
  - **MDSR (multi-scale)**: rede única que lida com múltiplos fatores de ampliação, compartilhando a maior parte dos parâmetros e apenas diferenciando nos módulos de entrada/saída, utilizando menos parâmetros que EDSR.
  - **Upsampling**: realizada no final da rede.

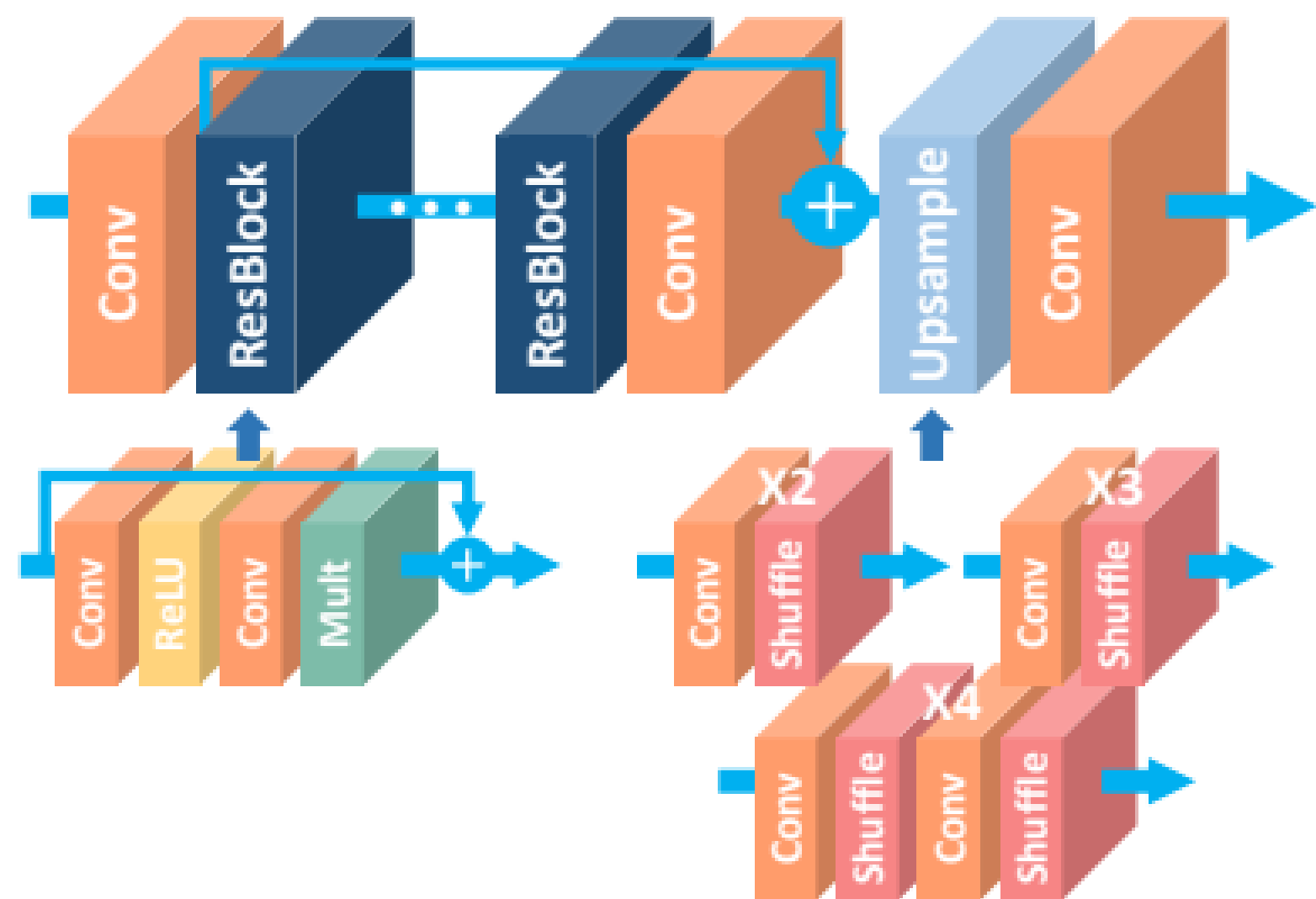


Figure 3: The architecture of the proposed single-scale SR network (EDSR).

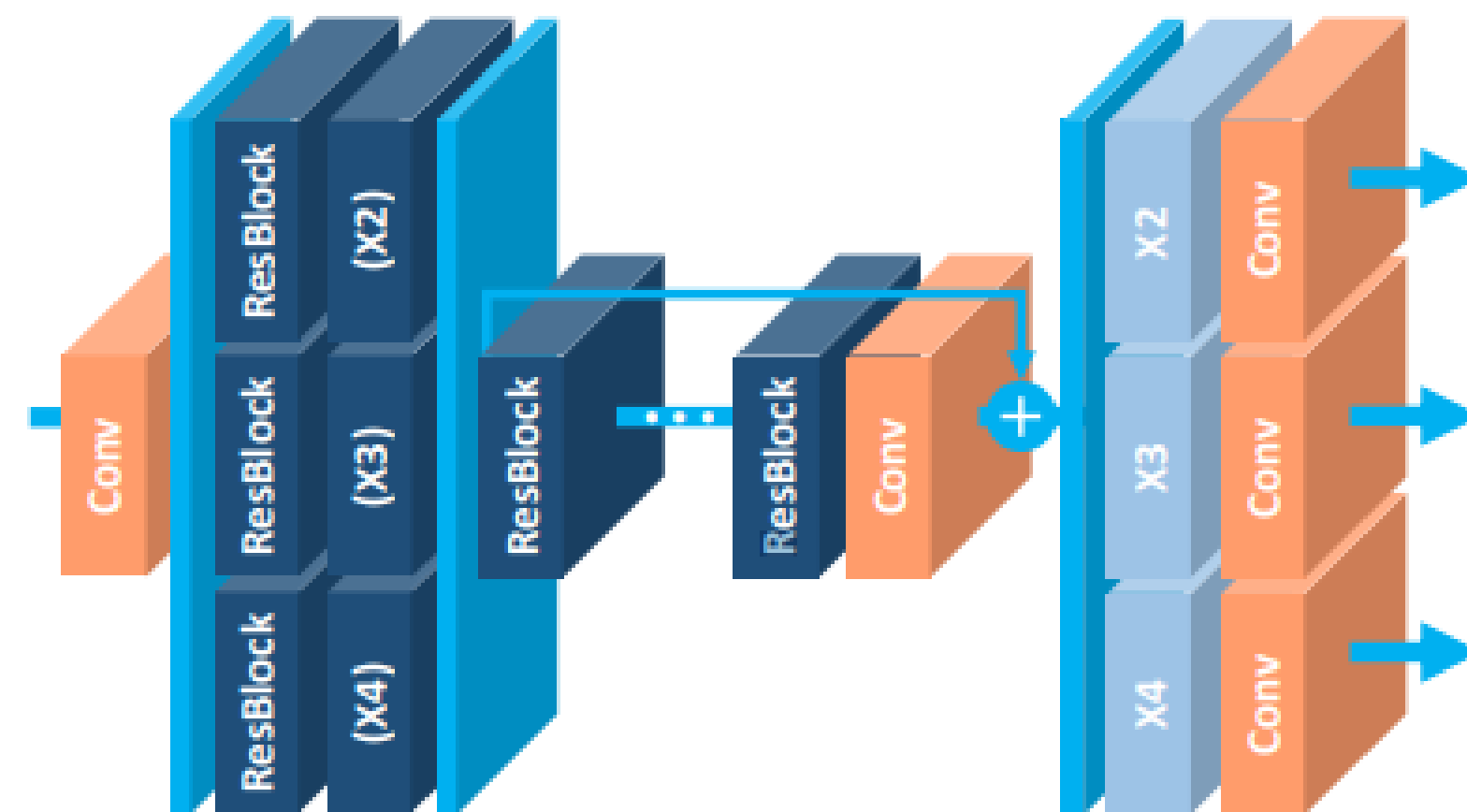


Figure 5: The architecture of the proposed multi-scale SR network (MDSR).

# Treinamento e otimização

- **Dados:** imagens do dataset **DIV2K**, com 800 para treino, 100 para validação e 100 para teste. Também testado em outros *benchmarks* clássicos (Set5, Set14, B100, Urban100).
- **Pré-processamento:** *patches* de 48x48 pixels, normalização pela média RGB e data augmentation (rotações e *flips*).
- **Inicialização progressiva:** treina-se primeiro o modelo x2, depois reutiliza-se esse modelo para treinar x3 e x4 (isso acelera a convergência).

# Treinamento e otimização

- **Função de perda:** em vez do usual **L2 (MSE)**, utilizam **L1(MAE)**, que empiricamente mostrou convergir melhor e gerar imagens mais nítidas.
- **Otimizador:** Adam, bastante usado em redes profundas por ajustar dinamicamente a taxa de aprendizado para cada parâmetro.
- **Self-ensemble:** na interferência, aplica rotações e flips na imagem de baixa resolução, processa todas as versões e depois combina os resultados na maior qualidade.

# Treinamento e otimização

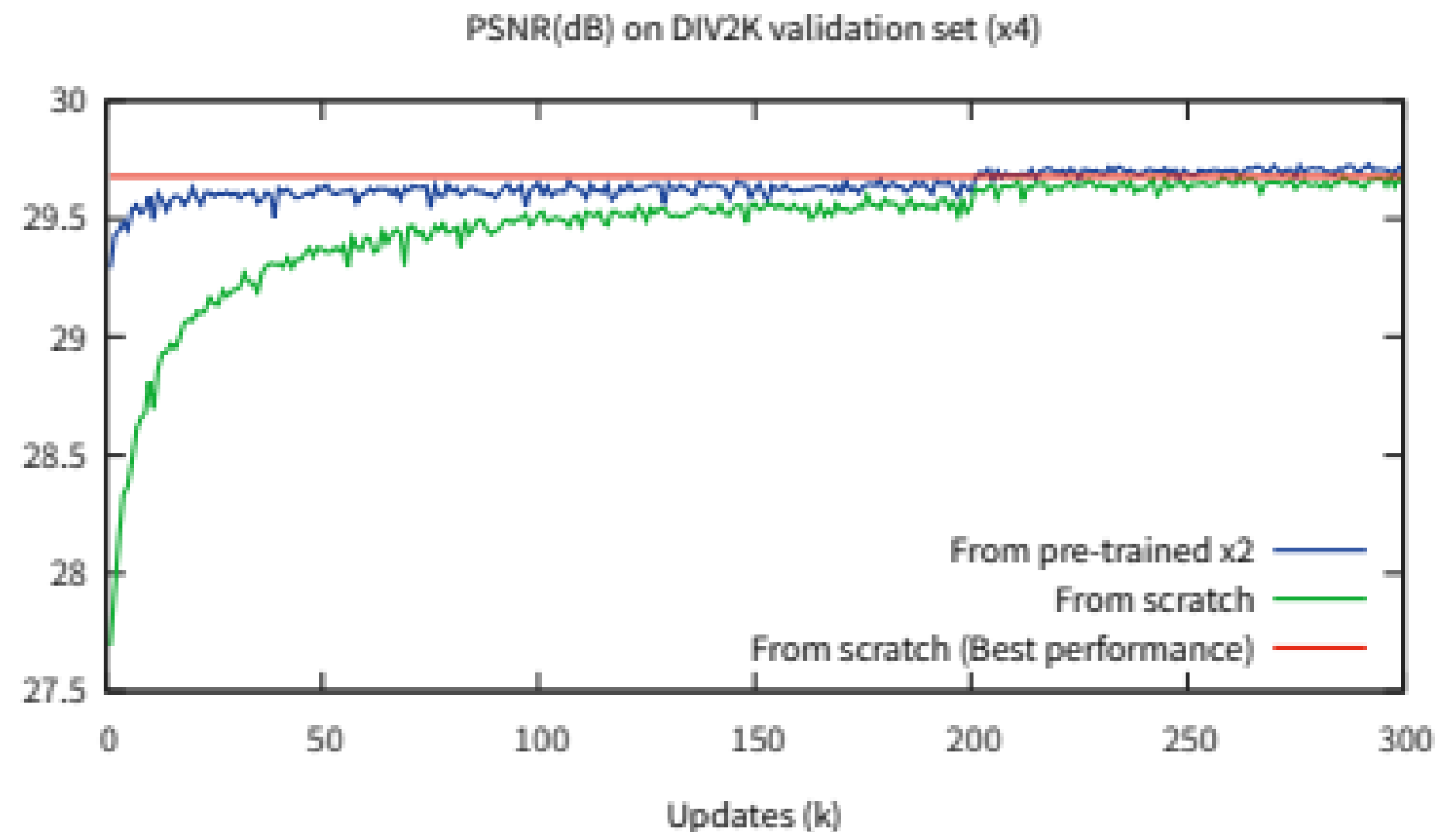


Figure 4: Effect of using pre-trained  $\times 2$  network for  $\times 4$  model (EDSR). The red line indicates the best performance of green line. 10 images are used for validation during training.

# Vantagens

## - Vantagens:

- Arquitetura simplificada (sem BN) → menor uso de memória e melhor qualidade da reconstrução.
- Estratégias de treinamento tornam o aprendizado mais estável, principalmente o residual scaling.
- Supera métodos anteriores em PSNR e SSIM, que medem a qualidade da reconstrução.
- MDSR reduz a necessidade de treinar vários modelos separados, reduzindo também o número de parâmetros utilizados.

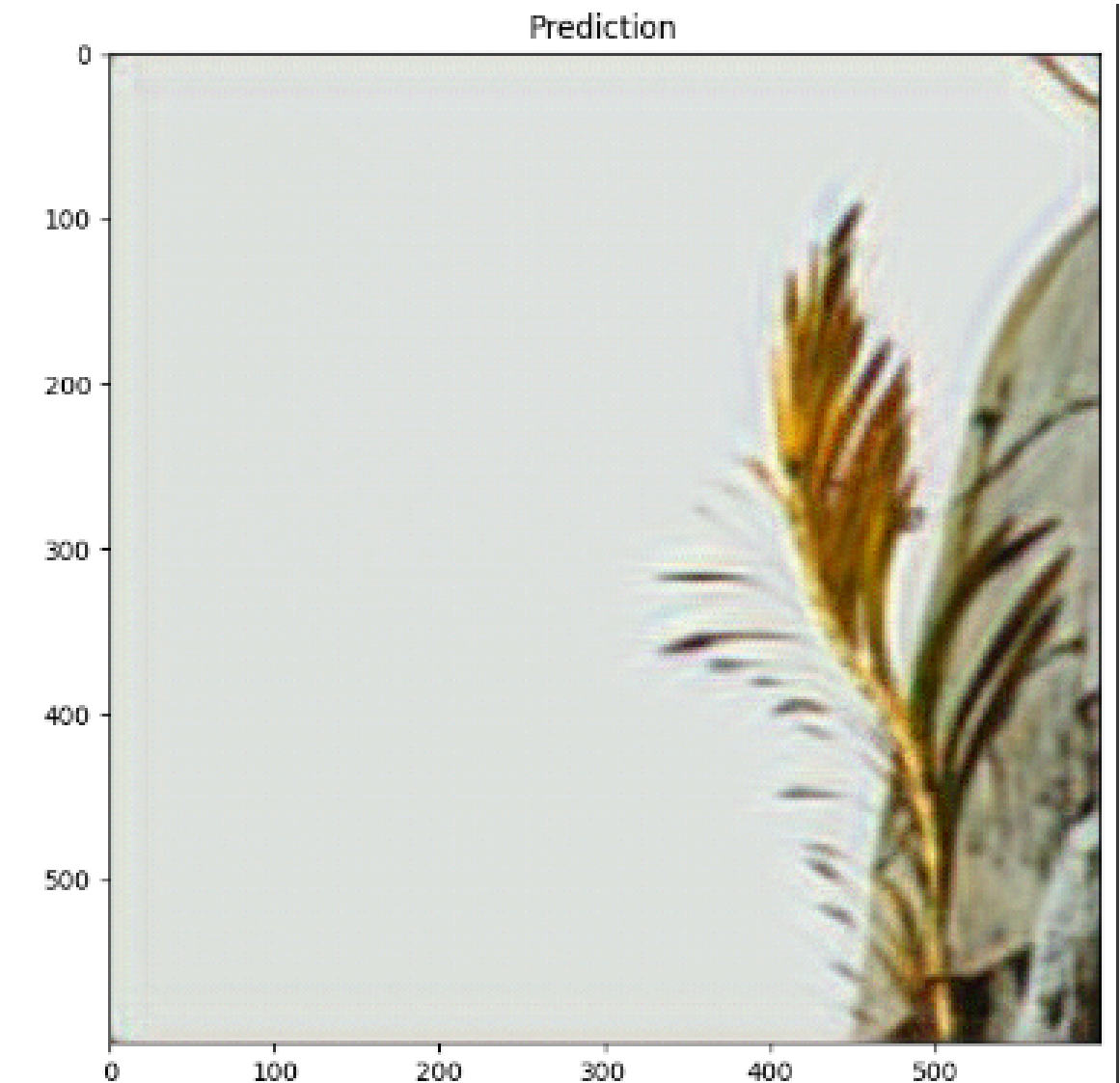
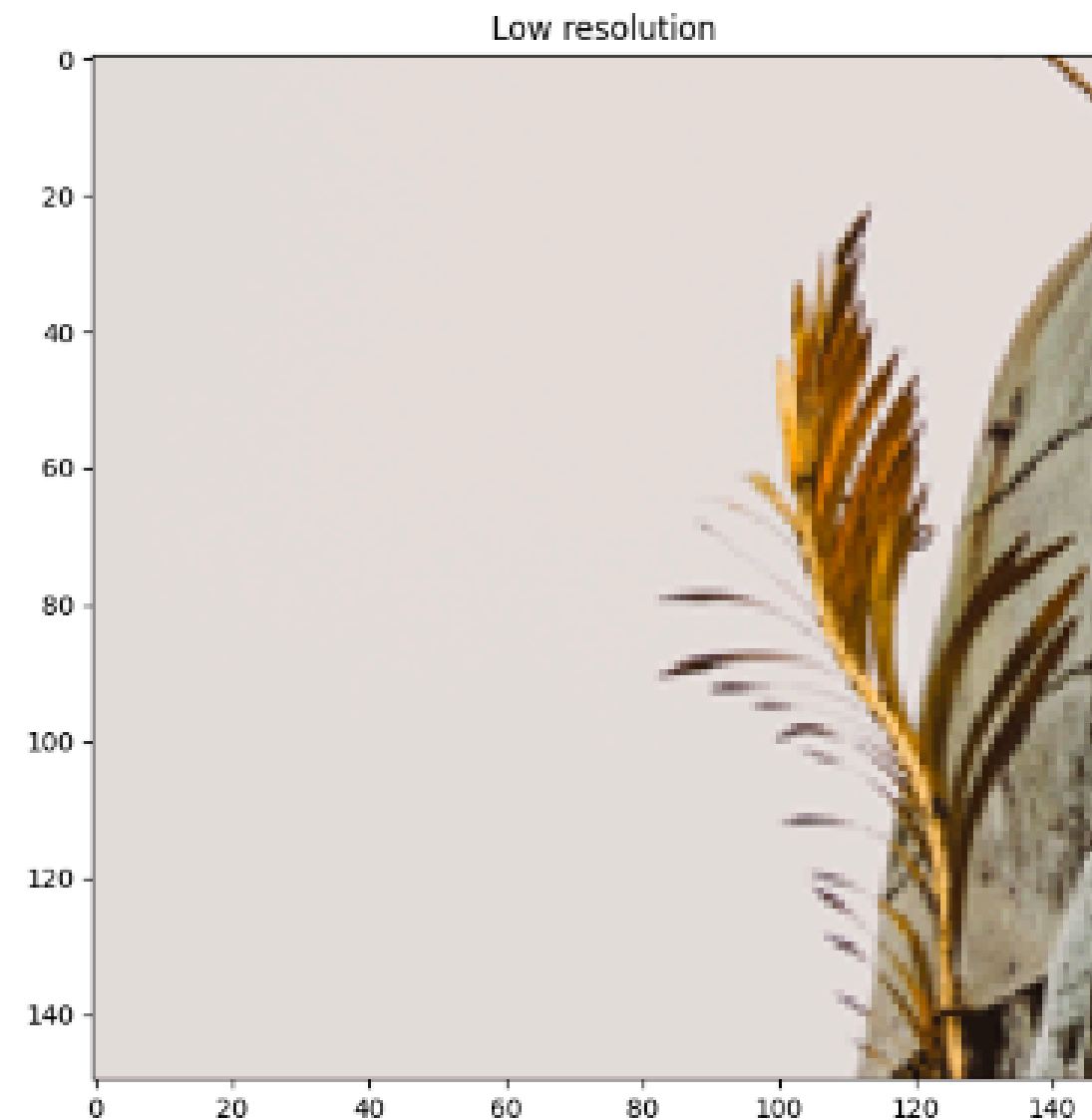
# Desvantagens

- **Desvantagens:**
  - Modelo ainda muito grande (EDSR chega a 43 milhões de parâmetros, exigindo grande poder computacional).
  - Embora melhore métricas (PSNR/SSIM), pode não capturar realismo à olho humano como métodos baseados em GAN (ex: SRGAN).
  - Treinamento é custoso (dias em GPUs de ponta), o que limita o acesso de grupos de pesquisa menores ou empresas com menos recursos.

# Exemplo(s) de aplicação

Link da aplicação:

<https://colab.research.google.com/drive/1MabSrQH8ugdy60cUeIcHQmad3yUR48D#scrollTo=jXloVyWKGRve>





## Exemplo(s) de aplicação

- **Restauro de imagens antigas:** aumentar qualidade de fotos ou vídeos antigos.
- **Aplicações médicas:** melhorar a resolução de exames (ex: ressonâncias).
- **Monitoramento por satélite:** reconstruir imagens de satélite com mais detalhes.
- **Streaming de vídeo:** transmitir em baixa resolução e ampliar no dispositivo do usuário.

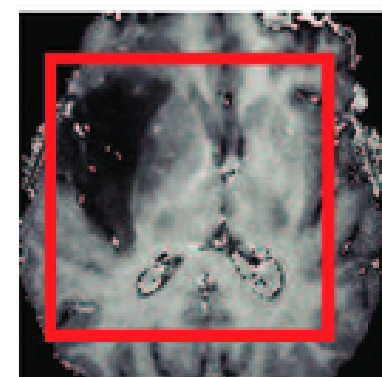
# Comparação com outros algoritmos

- **Bicubic**: rápido, mas perde detalhes (PSNR baixo).
- **A+/Sparse coding**: melhor que interpolação, mas limitado, dependendo de dicionários fixos.
- **SRCNN**: primeira CNN para SR, mas limitado pela baixa profundidade.
- **VDSR**: rede mais profunda, usa múltiplas escalas, mas exige interpolação bicúbica antes, o que reduz sua eficiência.

# Comparação com outros algoritmos

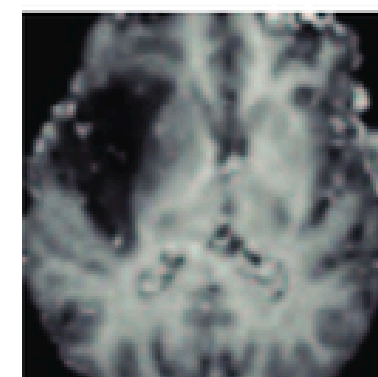
- **SRResNet**: melhora eficiência, mas arquitetura não é otimizada para SR.
- **EDSR**: remove módulos desnecessários, aumenta filtros, usa residual scaling → melhor resultado em benchmarks.
- **SRGAN**: gera imagens mais realistas para o olho humano, mas com métricas objetivas piores.
- **MDSR**: mantém desempenho alto, mas com menos parâmetros do que treinar vários EDSRs separados.

# Comparação com outros algoritmos



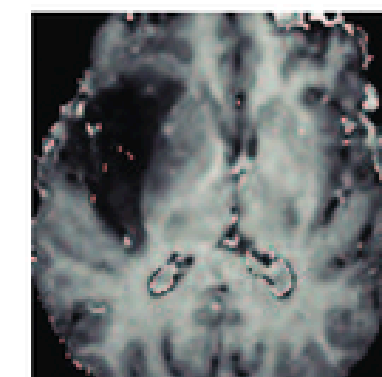
**Ground Truth**

PSNR (dB) / SSIM



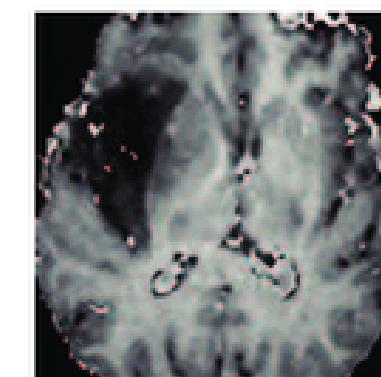
**Bicubic**

20.58 dB / 0.7410



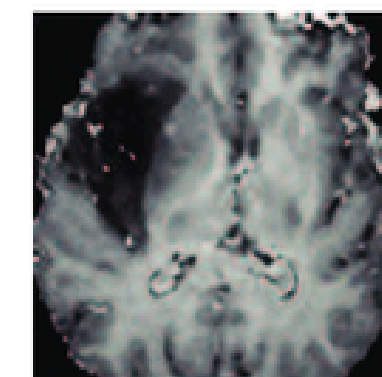
**EDSR**

22.86 dB / 0.8697



**SRGAN**

22.07 dB / 0.8626



**Meta-SRGAN**

22.09 dB / 0.8372

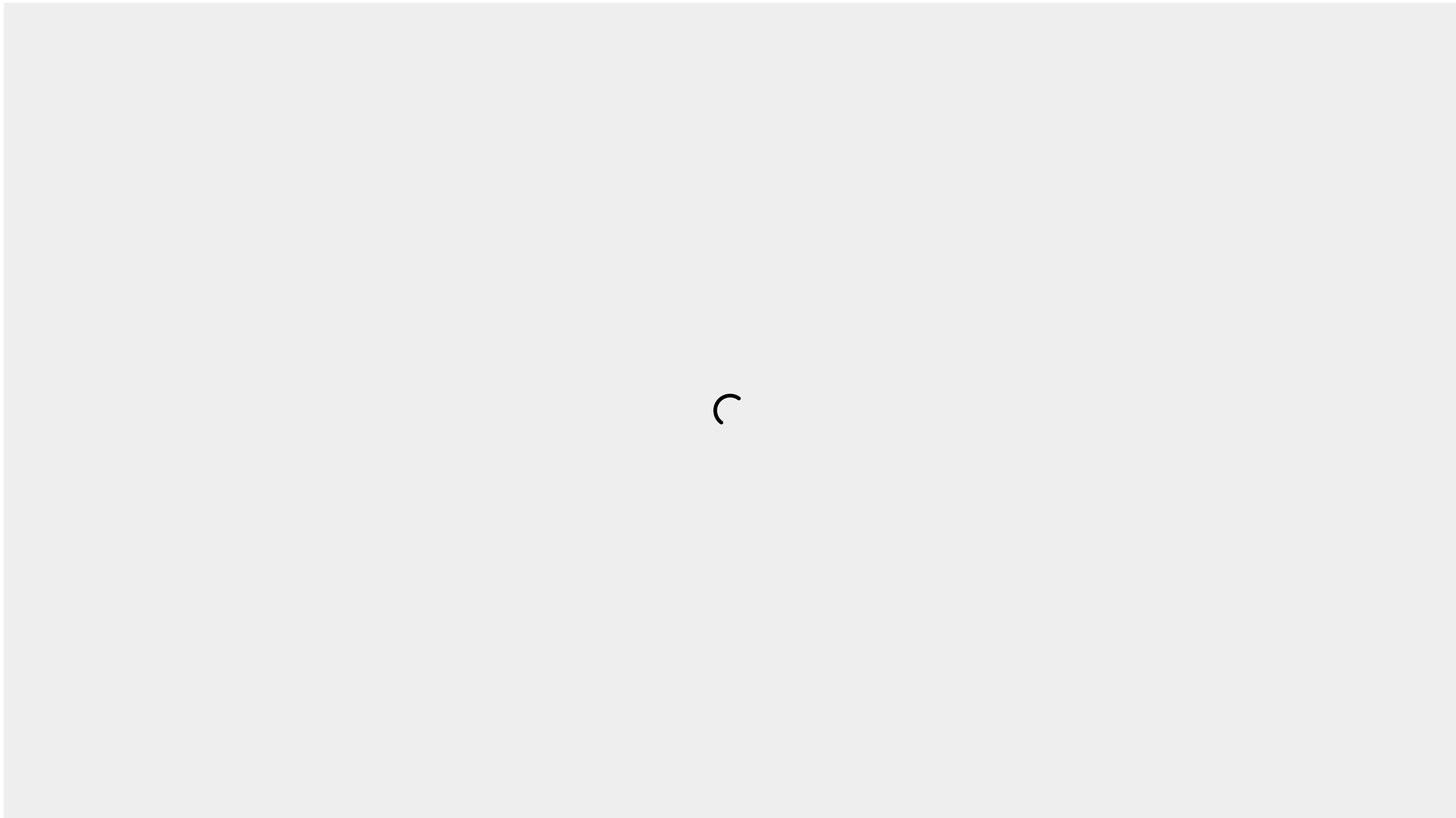
# Comparação com outros algoritmos

Options	SRResNet [14] (reproduced)	Baseline (Single / Multi)	EDSR	MDSR
# Residual blocks	16	16	32	80
# Filters	64	64	256	64
# Parameters	1.5M	1.5M / 3.2M	43M	8.0M
Residual scaling	-	-	0.1	-
Use BN	Yes	No	No	No
Loss function	L2	L1	L1	L1

Table 1: Model specifications.

Dataset	Scale	Bicubic	A+ [27]	SRCNN [4]	VDSR [11]	SRResNet [14]	EDSR (Ours)	MDSR (Ours)	EDSR+ (Ours)	MDSR+ (Ours)
Set5	$\times 2$	33.66 / 0.9299	36.54 / 0.9544	36.66 / 0.9542	37.53 / 0.9587	- / -	38.11 / 0.9601	38.11 / 0.9602	38.20 / 0.9606	38.17 / 0.9605
	$\times 3$	30.39 / 0.8682	32.58 / 0.9088	32.75 / 0.9090	33.66 / 0.9213	- / -	34.65 / 0.9282	34.66 / 0.9280	34.76 / 0.9290	34.77 / 0.9288
	$\times 4$	28.42 / 0.8104	30.28 / 0.8603	30.48 / 0.8628	31.35 / 0.8838	32.05 / 0.8910	32.46 / 0.8968	32.50 / 0.8973	32.62 / 0.8984	32.60 / 0.8982
Set14	$\times 2$	30.24 / 0.8688	32.28 / 0.9056	32.42 / 0.9063	33.03 / 0.9124	- / -	33.92 / 0.9195	33.85 / 0.9198	34.02 / 0.9204	33.92 / 0.9203
	$\times 3$	27.55 / 0.7742	29.13 / 0.8188	29.28 / 0.8209	29.77 / 0.8314	- / -	30.52 / 0.8462	30.44 / 0.8452	30.66 / 0.8481	30.53 / 0.8465
	$\times 4$	26.00 / 0.7027	27.32 / 0.7491	27.49 / 0.7503	28.01 / 0.7674	28.53 / 0.7804	28.80 / 0.7876	28.72 / 0.7857	28.94 / 0.7901	28.82 / 0.7876
B100	$\times 2$	29.56 / 0.8431	31.21 / 0.8863	31.36 / 0.8879	31.90 / 0.8960	- / -	32.32 / 0.9013	32.29 / 0.9007	32.37 / 0.9018	32.34 / 0.9014
	$\times 3$	27.21 / 0.7385	28.29 / 0.7835	28.41 / 0.7863	28.82 / 0.7976	- / -	29.25 / 0.8093	29.25 / 0.8091	29.32 / 0.8104	29.30 / 0.8101
	$\times 4$	25.96 / 0.6675	26.82 / 0.7087	26.90 / 0.7101	27.29 / 0.7251	27.57 / 0.7354	27.71 / 0.7420	27.72 / 0.7418	27.79 / 0.7437	27.78 / 0.7425
Urban100	$\times 2$	26.88 / 0.8403	29.20 / 0.8938	29.50 / 0.8946	30.76 / 0.9140	- / -	32.93 / 0.9351	32.84 / 0.9347	33.10 / 0.9363	33.03 / 0.9362
	$\times 3$	24.46 / 0.7349	26.03 / 0.7973	26.24 / 0.7989	27.14 / 0.8279	- / -	28.80 / 0.8653	28.79 / 0.8655	29.02 / 0.8685	28.99 / 0.8683
	$\times 4$	23.14 / 0.6577	24.32 / 0.7183	24.52 / 0.7221	25.18 / 0.7524	26.07 / 0.7839	26.64 / 0.8033	26.67 / 0.8041	26.86 / 0.8080	26.86 / 0.8082
DIV2K validation	$\times 2$	31.01 / 0.9393	32.89 / 0.9570	33.05 / 0.9581	33.66 / 0.9625	- / -	35.03 / 0.9695	34.96 / 0.9692	35.12 / 0.9699	35.05 / 0.9696
	$\times 3$	28.22 / 0.8906	29.50 / 0.9116	29.64 / 0.9138	30.09 / 0.9208	- / -	31.26 / 0.9340	31.25 / 0.9338	31.39 / 0.9351	31.36 / 0.9346
	$\times 4$	26.66 / 0.8521	27.70 / 0.8736	27.78 / 0.8753	28.17 / 0.8841	- / -	29.25 / 0.9017	29.26 / 0.9016	29.38 / 0.9032	29.36 / 0.9029

Table 3: Public benchmark test results and DIV2K validation results (PSNR(dB) / SSIM). Red indicates the best performance and blue indicates the second best. Note that DIV2K validation results are acquired from published demo codes.



Perguntas?



Quiz

Escanear o QRCode para acessar



Quiz

# Referências

- Lim, B., Son, S., Kim, H., Nah, S., & Mu Lee, K. (2017). **Enhanced Deep Residual Networks for Single Image Super-Resolution**. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 136–144.
- Tan, Chuan & Zhu, Jin & Lio, Pietro. (2020). Arbitrary Scale Super-Resolution for Brain MRI Images. 10.48550/arXiv.2004.02086.
- Keras. Enhanced Deep Residual Networks for single-image super-resolution. Disponível em: <https://keras.io/examples/vision/edsr/>. Acesso: 21/08/2025.

Obrigado!