
UNIVERSIDADE FEDERAL DO AMAZONAS – UFAM
INSTITUTO DE COMPUTAÇÃO – ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA – PPGI

JOÃO DA MATA LIBÓRIO FILHO

# DISTRIBUIÇÃO DE VÍDEO NA INTERNET APRIMORADA POR SUPER-RESOLUÇÃO BASEADA EM REDES NEURAIS ADVERSÁRIAS GENERATIVAS

MANAUS-AM

Maio/2023

JOÃO DA MATA LIBÓRIO FILHO

# DISTRIBUIÇÃO DE VÍDEO NA INTERNET APRIMORADA POR SUPER-RESOLUÇÃO BASEADA EM REDES NEURAIS ADVERSÁRIAS GENERATIVAS

Tese apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas como requisito para a obtenção do título de Doutor em Informática.

Orientador: Profº. Drº. César Augusto Viana Melo

MANAUS-AM

Maio/2023

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Libório Filho, João da Mata
L696d Distribuição de vídeo na internet aprimorada por super-resolução baseada em redes neurais adversárias generativas / João da Mata Libório Filho . 2023
158 f.: il. color; 31 cm.

Orientador: César Augusto Viana Melo
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. Super-resolução de vídeo. 2. Redes neurais adversária generativa. 3. Cdn. 4. Distribuição de vídeo. 5. Streaming de vídeo adaptativo. I. Melo, César Augusto Viana. II. Universidade Federal do Amazonas III. Título

Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Informática

**FOLHA DE APROVAÇÃO**

# "DISTRIBUIÇÃO DE VÍDEO NA INTERNET APRIMORADA POR SUPER-RESOLUÇÃO BASEADA EM REDES NEURAIS ADVERSÁRIAS GENERATIVAS"

## JOÃO DA MATA LIBÓRIO FILHO

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Dr. César Augusto Viana Melo - PRESIDENTE

Prof. Dr. Nelson Luís Saldanha da Fonseca - MEMBRO EXTERNO

Prof. Dr. Fábio Luciano Verdi - MEMBRO EXTERNO

Prof. Dr. Marco Antônio Pinheiro de Cristo - MEMBRO EXTERNO

Prof. Dr. José Reginaldo Hughes Carvalho - MEMBRO INTERNO

Manaus, 10 de Maio de 2023

Documento assinado eletronicamente por **Nelson Luis Saldanha da Fonseca**, **Usuário Externo**, em 10/05/2023, às 14:29, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **César Augusto Viana Melo**, **Professor do Magistério Superior**, em 10/05/2023, às 16:10, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **José Reginaldo Hughes Carvalho**, **Professor do Magistério Superior**, em 11/05/2023, às 11:05, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Marco Antônio Pinheiro de Cristo**, **Professor do Magistério Superior**, em 11/05/2023, às 14:57, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Fábio Luciano Verdi**, **Usuário Externo**, em 12/05/2023, às 11:04, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1456945** e o código CRC **DFD7C0E5**.

Avenida General Rodrigo Octávio, 6200 - Bairro Coroado I Campus Universitário Senador Arthur Virgílio Filho, Setor Norte - Telefone: (92) 3305-1181 / Ramal 1193
CEP 69080-900, Manaus/AM, coordenadorppgi@icomp.ufam.edu.br

*Dedico esta tese à minha amada esposa, Romana Góes da Mata,*
*pelo amor, apoio e paciência incansáveis que ela me proporcionou durante*
*todo o percurso deste doutorado. Sem ela, este trabalho não seria possível.*

*Também dedico aos meus filhos, Valentina Catherine, Juan Victor e*
*Cristine Victória, pela alegria e inspiração que eles me trazem diariamente.*
*Espero que este trabalho possa ser um exemplo de perseverança e*
*dedicação para que eles possam perseguir seus próprios sonhos e objetivos.*

*Por fim, dedico esta tese em memória do meu pai, João Libório,*
*que sempre me encorajou a buscar conhecimento.*
*"Estude meu filho! para um dia ser um doutor!", palavras dele.*
*Pai! nunca esquecerei seus conselhos!*

*Esta conquista é dedicada a vocês,*
*que foram e são fundamentais em minha vida!*

# Agradecimentos

Gostaria de expressar meus sinceros agradecimentos a todos aqueles que contribuíram para a realização desta tese. Primeiramente, gostaria de agradecer ao meu orientador pelo apoio e orientação ao longo de todo o processo. Seu comprometimento, *insights* e *feedbacks* foram inestimáveis para a conclusão deste trabalho.

Meus agradecimentos à Universidade do Estado do Amazonas e à Universidade Federal do Amazonas por terem me concedido uma licença remunerada para que eu pudesse cursar o doutorado. Além disso, gostaria de agradecer à Fundação de Amparo à Pesquisa do Estado do Amazonas pelo apoio financeiro concedido através do programa PROINT. Sem o apoio dessas renomadas instituições, eu jamais teria conseguido me manter no doutorado.

Agradeço à minha esposa Romana da Mata e aos meus filhos pela compreensão e apoio nestes anos de doutoramento. Seu amor e suporte foram essenciais para superar os desafios enfrentados durante este processo.

Por fim, agradeço aos meus queridos pais, que sempre me apoiaram em todos os momentos da vida. Ao meu pai, João Libório, que infelizmente já não está mais conosco, agradeço por ter sido um exemplo de força e determinação, e por ter me ensinado a importância do conhecimento e da educação. À minha mãe, Maria do Carmo, agradeço por todo amor, carinho e incentivo que sempre me proporcionou.

Agradeço imensamente a todos que contribuíram para tornar possível a realização desse importante passo na minha carreira acadêmica. Meu muito obrigado!

*"Our intelligence is what makes us human,*
*and AI is an extension of that quality."*
(Yann LeCun)

## RESUMO

A distribuição de vídeos através da Internet tem se tornado crescente nos últimos anos, estima-se que conteúdo de vídeos represente 82% de todo o tráfego da Internet. Os grandes provedores de conteúdos de vídeo como Netflix, Prime Vídeo, Youtube, utilizam redes de distribuição de conteúdos para replicar seus conteúdos em locais mais próximo da audiência, com o intuito de melhorar a latência e evitar *rebuffering*. Além disso, empregam a tecnologia de taxa de bits adaptável que permite a codificação de vídeos em diferentes resoluções e taxas de bits para diferentes dispositivos, sendo que essa codificação geralmente ocorre nos *data centers* desses provedores de conteúdo. Os algoritmos de posicionamento de conteúdo encontram o melhor ajuste entre esses arquivos e o público-alvo, levando em consideração restrições de custos. No entanto, essa técnica requer múltiplas representações do mesmo vídeo, o que resulta em dados redundantes trafegando nas infraestruturas de distribuição e pode sobrecarregar as redes de distribuição de conteúdo. Por outro lado, nos últimos anos, as redes neurais profundas, em especial as redes adversárias generativas, têm se destacado na literatura para métodos de super-resolução de imagem e vídeo. Esses métodos são capazes de restaurar imagens de baixa para alta resolução com qualidade imperceptível à visão humana. Nesta pesquisa, foi avaliada a aplicação de super-resolução de vídeo usando redes adversárias generativas em duas abordagens principais: *i*) reduzir o volume de dados de vídeos enviados pelas infraestruturas de nuvem, replicando vídeos em versões de baixa resolução entre *data centers* e servidores substitutos geograficamente distribuídos, e restaurando-os para alta resolução nesses servidores usando um modelo de super-resolução; *ii*) otimizar a qualidade de experiência da audiência dos aplicativos de *streaming* de vídeo ao vivo, melhorando a qualidade perceptiva por meio da super-resolução. O estudo foi conduzido em um ambiente experimental configurado para corresponder aos cenários reais. Os resultados apresentam dois *frameworks* que incorporam modelos de super-resolução, um para o serviço de replicação de vídeo em nuvem e outro para o serviço de distribuição de vídeo ao vivo, apoiado por computação de borda. As avaliações experimentais mostraram uma redução no tráfego relacionado a vídeo nas infraestruturas de até 88,37%, enquanto ao mesmo tempo melhoravam os padrões de qualidade de experiência na sessão, medidos durante o *streaming* de vídeo ao vivo.



**Palavras-chaves**: Super-resolução de vídeo, Redes neurais adversária generativa, CDN, Distribuição de vídeo, *Streaming* de vídeo adaptativo.

## ABSTRACT

Over the years, video content distribution over the internet has increased dramatically, with video content predicted to represent 82% of all internet traffic. Major video content providers, such as Netflix, Prime Video, and YouTube, use content delivery networks (CDNs) to replicate their content in locations closer to their users, improving latency and avoiding rebuffering. Content providers use adaptive video streaming to enable viewers to watch videos with adequate image quality based on their end-to-end connection with the provider. However, this technology requires multiple representations of the same video with varying resolutions and bitrates, increasing the volume of redundant data flowing through distribution infrastructures and overburdening CDN infrastructure. Recent literature has highlighted deep neural networks, particularly generative adversarial networks, for image and video super-resolution methods. These methods can restore low-resolution images and videos to high resolution with unnoticeable quality loss to human vision. In this study, the application of video super-resolution using a generative adversarial network was evaluated in two ways: i) To reduce video traffic in cloud infrastructures, videos were replicated in low-resolution versions between data centers and geographically distributed surrogate servers. These videos were then restored to high resolution on these servers using a super-resolution model; ii) To optimize the quality of experience for viewers of live video streaming applications, enhanced perceptual quality was achieved through super-resolution. The research was conducted in an experimental setting designed to simulate real-world scenarios. The findings demonstrate two frameworks that use a super-resolution model, one for a cloud video replication service and the other for a live video distribution service supported by edge computing. The experiment results revealed a reduction in video-related traffic in infrastructures of up to 88.37%. Additionally, the quality of the session experience during live video streaming was improved, as measured by perceptual quality. Overall, the study suggests that using super-resolution techniques for video content delivery can reduce network traffic and improve the quality of experience for viewers. These findings could have implications for the future of video content delivery, especially as video content continues to grow in popularity and demand.



**Key-words**: Video super-resolution, Generative adversarial networks, CDN, Video delivery, Adaptive video streaming.

# LISTA DE ILUSTRAÇÕES

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| $I^{HR}$ | imagem de alta resolução |
| $I^{LR}$ | imagem de baixa resolução |
| $I^{SR}$ | imagem por super-resolução |
| A3C | *asynchronous advantage actor-critic* |
| ABR | *adaptive bitrate streaming* |
| AMC | *adaptive motion compensation* |
| BM | *basic modules* |
| BN | *batch normalization* |
| C-RAN | *centralized/cloud radio access network* |
| CA | *content-aware* |
| CDN | *content delivery network* |
| CISRDCNN | *super-resolution of compressed images using deep convolutional neural networks* |
| CNN | *convolutional neural network* |
| ConvNet | *convolutional network* |
| CPU | *central processing unit* |
| CRF | *constant rate factor* |
| CU | *centralized unit* |
| DASH | *dynamic adaptive streaming over HTTP* |
| DenseNet | *dense convolutional network* |
| DFF | *dense feature fusion* |
| DNN | *deep neural network* |
| DRDVSR | *detail-revealing deep video super-resolution* |
| DU | *distributed unit* |
| EDSR | *enhanced deep super-resolution network* |
| EDVR | *video restoration with enhanced deformable convolutional networks* |
| ELiveSR | *on-edge enhanced live streaming with super-resolution* |
| ESPCN | *efficient sub-pixel convolutional neural networks* |
| ESRGAN | *enhanced super-resolution generative adversarial networks* |
| EVSRNet | *efficient video super-resolution network* |
| FDA | função de distribuição acumulada |

| | |
|---|---|
| FPS | *frames per second* |
| FR | *full-reference* |
| GAN | *generative adversarial network* |
| GPU | *graphics processing unit* |
| HAS | *HTTP-based adaptive streaming* |
| HD | *high definition* |
| HR | *high resolution* |
| HVS | *human visual system* |
| IaaS | *infrastructure as a service* |
| IMDB | *information multi-distillation block* |
| IMDN | *information multi-distillation network* |
| IoT | *internet of things* |
| ISP | *internet service provider* |
| IT | *information technology* |
| JND | *just-noticeable-difference* |
| KPI | *key performance indicator* |
| LeakyReLU | *leaky rectified linear unit* |
| LFF | *local feature fusion* |
| LoL | *low-on-latency* |
| LPIPS | *learned perceptual image patch similarity* |
| LR | *low-resolution* |
| LSTM | *long short-term memory* |
| MAE | *mean absolute error* |
| MEC | *multi-access edge computing* |
| MFSR | *multi-frame super-resolution* |
| ML | *machine learning* |
| MSE | *mean squared error* |
| NR | *no-reference* |
| OCO | *online convex optimization* |

| | |
|---|---|
| PCD | *pyramid, cascading and deformable convolutions* |
| PI | *perception index* |
| PoP | *point of presence* |
| PReLU | *parametric rectified linear unit* |
| PSNR | *peak signal-to-noise ratio* |
| | |
| QF | *quality factor* |
| QoE | *quality of experience* |
| QP | *quantization parameter* |
| | |
| RaD | *relativistic average discriminator* |
| RaGAN | *relativistic average GAN* |
| RDB | *residual dense block* |
| RDN | *residual dense network* |
| ReLU | *rectified linear unit* |
| RGB | *red, green, and blue* |
| RL | *reinforcement learning* |
| RMSE | *root-mean-square error* |
| RR | *reduced-reference* |
| RRDB | *residual-in-residual dense block* |
| RRU | *remote radio unit* |
| RTMP | *real time messaging protocol* |
| RTSRGAN | *real-time super-resolution generative adversarial networks* |
| RTVSRGAN | *real-time video super-resolution generative adversarial networks* |
| | |
| SD | *standard-definition* |
| SFENet | *shallow feature extraction net* |
| SISR | *sigle image super-resolution* |
| SPMC | *sub-pixel motion compensation* |
| SR | *super-resolution* |
| SRCNN | *super-resolution convolutional neural networks* |
| SRDenseNet | *super-resolution dense convolutional network* |
| SRGAN | *super-resolution generative adversarial networks* |
| SRResNet | *super-resolution residual network* |
| SSIM | *structural similarity* |
| SVM | *support vector machine* |
| | |
| Tanh | *hyperbolic tangent* |
| TCP | *transmission control protocol* |

| | |
|---|---|
| TDAN | *temporally deformable alignment network* |
| TSA | *temporal and spatial attention* |
| | |
| UHD | *ultra high definition* |
| UPNet | *up-sampling net* |
| | |
| VESPCN | *video efficient sub-pixel convolutional neural network* |
| VGG | *very deep convolutional networks* |
| VMAF | *video multi-method assessment fusion* |
| VoD | *video on demand* |
| VQA | *video quality assessment* |
| VR | *virtual reality* |
| VSR | *video super-resolution* |
| VSR-DUF | *video super-resolution using dynamic upsampling filters* |
| VSRGAN+ | *improved video super-resolution with GAN* |
| VSRnet | *video super-resolution network* |
| | |
| YCbCr | *Y: luminance; Cb: chrominance-blue; and Cr: chrominance-red* |

SUMÁRIO

# CAPÍTULO 1

# Introdução

Os vídeos são, atualmente, o meio digital mais popular para entretenimento, comunicação e educação *online*. O lançamento de serviços de vídeo, o avanço da tecnologia de redes computacionais e o uso generalizado de dispositivos móveis foram momentos decisivos que prepararam o caminho para a popularidade dos vídeos *online*. Além disso, a capacidade de criar e compartilhar conteúdo de vídeo a baixo custo aumentou o apelo pelos aplicativos de *streaming* de vídeos, tornando-os dominantes na Internet. Os relatórios de tráfego da Internet revelam que uma quantidade considerável de largura de banda é consumida por tais aplicativos e dispositivos capazes de reproduzir conteúdo de alta definição (HD, do inglês *high definition*) e ultra alta definição (UHD, do inglês *ultra high definition*), criando cenários em que essa demanda continuará nos próximos anos. De acordo com a análise anual de tráfego de Internet da Cisco [1], o vídeo representa 82% do tráfego global da Internet, sendo que 22% desse tráfego é composto por aplicativos de vídeo sob demanda (VoD, do inglês *video on demand*), que servem conteúdo em HD (57%) e UHD (22%).

Os principais serviços de *streaming* de vídeo são implantados usando uma solução de tecnologia multicamada [2], camada de conteúdo, de transporte e computacional, que oferece uma experiência semelhante à da televisão em qualquer lugar e a qualquer hora. Na camada de conteúdo esses serviços utilizam a tecnologia de taxa de bits de vídeo adaptável para fornecer vídeos que correspondam aos recursos de reprodução no lado cliente. Na camada de transporte, a redução do *playtime* [3], ou seja, o tempo entre apertar o botão *play* e a triagem do conteúdo, acontece por meio do uso de redes de distribuição de conteúdo (CDNs, do inglês *content delivery network*s), que entregam conteúdo de vídeo

com taxa de bits adaptável do ponto mais próximo à audiência. Na camada computacional, o gerenciamento dos padrões de acesso a conteúdo estabelecidos por grandes públicos é feito pela computação em nuvem.

Com a adoção de tecnologias de taxa de bits adaptáveis, os vídeos são codificados em um conjunto de resoluções de tela e taxas de bits de reprodução definidas pelos dispositivos da audiência. Esse processo de codificação geralmente ocorre nos *data centers* dos provedores de conteúdo. Em seguida, o resultado desse pré-processamento, que é um conjunto de arquivos de diferentes tamanhos e qualidades, é transferido usando a infraestrutura de transporte contratada. Os algoritmos de posicionamento de conteúdo encontram o melhor ajuste entre esses arquivos e o público-alvo, levando em consideração restrições de custos.

Todos esses esforços são limitados pelas práticas estabelecidas pelos processos de publicação de vídeo, ou seja, receitas de codificação que definem a resolução de destino e as taxas de bits de *streaming* no final da fase de codificação. Com base nessas receitas, os vídeos são codificados, agrupados e, em seguida, movimentados e armazenados em servidores substitutos. Além disso, apesar de todos os avanços recentes nas técnicas de compactação de vídeo, mover e armazenar vídeos codificados são operações que exigem naturalmente recursos. Por exemplo, uma sequência de vídeo de cinco segundos pode exigir até 97 MBytes de armazenamento e 156 Mbps para manter sua taxa de bits de codificação durante uma sessão de *streaming* direcionada a dispositivos HD.

O estado da arte de *streaming* de vídeo implementa a codificação por título usando técnicas de aprendizagem de máquina (ML, do inglês *machine learning*). Nessa abordagem, os modelos de ML encontram um conjunto de taxas de bits distintas que têm significado perceptivo para sua audiência. Em outras palavras, durante uma sessão, mudar a taxa de bits para um valor maior melhorará a qualidade percebida em uma sessão de vídeo. As receitas de codificação estática não podem fornecer tal garantia devido à sua generalização. Em Bitmovin Inc[4], a eficácia por título foi avaliada e mostrou uma economia impressionante de 84% devido a menos bits por segmento e menos alterações de qualidade por parte do cliente. No entanto, em tal dinâmica de produção, há um conjunto de vídeos codificados que requerem movimentação e armazenamento visando a redução do tempo de reprodução.

A otimização da entrega de vídeo na Internet tem sido discutida em muitos trabalhos [2, 5, 6, 7, 8], que incluem CDNs, *streaming* adaptativo baseado em HTTP (HAS, do inglês *HTTP-based adaptive streaming*), CDN-P2P, CDN assistido por *fog computing*, super-resolução de vídeo (VSR, do inglês *video super-resolution*), entre outros. No entanto, a entrega de vídeo ainda tem muitos desafios a serem resolvidos [2, 9]. Por exemplo, a quantidade de tráfego de vídeo nas redes vem aumentando anualmente, caminhando para um gargalo da Internet [1, 10].

Além disso, o tráfego de dados internacional fornecido pelos provedores de serviços de internet (ISPs, do inglês *internet service provider*s) de longa distância, conhecidos como *tier-1*, é mais caro do que o tráfego regional e local fornecido pelos provedores *tier-2* e *tier-3* [11]. Como resultado, reduzir o grande volume de dados que trafega pelas infraestruturas dos provedores *tier-1* pode ser uma alternativa viável para diminuir os custos de transferência de dados.

Por outro lado, houve um aumento no poder computacional disponível nas nuvens de borda, o que também resultou em um maior volume de recursos de computação ociosos nos servidores de *back-end* [12]. Além disso, muitas tarefas de processamento de vídeo podem ser executadas em unidades de processamento gráfico (GPUs, do inglês *graphics processing unit*s), que ganharam maior poder computacional nos últimos anos. Essa tendência no desempenho das GPUs tem levado à redução dos custos de processamento dos serviços em nuvem [13, 14]. Por exemplo, no início de 2020, o Google Cloud reduziu seus preços de processamento em GPU em mais de 60% [15].

A utilização desses recursos para processar vídeos nas bordas, como transcodificação, codificação e reconstrução de qualidade, apresenta uma abordagem promissora para reduzir os custos associados ao tráfego global da internet. Além disso, as empresas de entrega de vídeo e computação em nuvem podem estabelecer uma relação comercial recíproca, como por exemplo, reduzindo os custos de movimentação de conteúdo de vídeo entre servidores originais e substitutos e, ao mesmo tempo, consumindo mais o poder de processamento oferecido pelas empresas de computação em nuvem.

A aprendizagem de máquina experimentou um *boom* sem precedentes em aplicações que permitem a automação em diversas áreas [16, 17]. Esse avanço foi impulsionado pela crescente disponibilidade de dados, pelo aprimoramento das abordagens de ML e pelo progresso nas capacidades computacionais. Recentemente, descobertas em redes neurais profundas (DNNs, do inglês *deep neural network*s) forneceram novas possibilidades para melhorias na distribuição de vídeos na Internet [18, 19, 20, 21, 22, 23, 24, 25]. Essas descobertas mostraram que dois modelos neurais de aprendizado profundo, as redes neurais convolucionais (CNNs, do inglês *convolutional neural network*s) e as redes adversárias generativas (GANs, do inglês *generative adversarial network*s), podem receber imagens de baixa resolução e reconstruí-las em alta resolução [26]. Essa técnica é conhecida como *super-resolution* (SR) e, segundo esses estudos, as imagens reconstruídas têm uma qualidade perceptiva muito semelhante à das imagens originais de alta resolução.

Esses estudos motivaram a realização desta pesquisa, na qual avaliou-se a aplicação de *super-resolution* (SR) de vídeo com uso de *generative adversarial network* (GAN) com os seguintes objetivos:

*i*) Reduzir o tráfego de vídeos nas infraestruturas de nuvens computacionais. Nesse contexto, os vídeos são replicados entre *data center* (servidor original) e os servidores de

borda (servidores substitutos) em versões de baixa resolução e, nesses servidores, com uso de modelo de SR baseado em GAN, são reconstruídos para alta resolução;

*ii*) Melhorar a qualidade de experiência (QoE, do inglês *quality of experience*) durante sessões de vídeo *streamming* ao vivo. Nessas sessões, os vídeos são transmitidos em baixa resolução do servidor em nuvem à computação de borda de acesso múltiplo (MEC, do inglês *multi-access edge computing*), e uma GAN na MEC se encarregará de reconstruir os vídeos em HR com uso de SR. A possibilidade de acessar o conteúdo em HD, em um tempo mais reduzido de carregamento, potencializa a qualidade do serviço oferecido as audiências.

## 1.1 DEFINIÇÃO DO PROBLEMA

O termo SR não é recente e nas últimas duas décadas muitas pesquisas foram publicadas abordando esse problema em várias áreas, entre elas, processamento de imagens de satélites e aéreas, aprimoramento de imagens médicas, sintetização de imagens faciais, melhorias de imagens de textos, compressão de imagem e vídeos, entre outras [27]. Mas, foi a partir de 2015 que SR voltou a ser um tema ativo em pesquisas científicas, graças a aplicação de métodos baseados em *deep neural network*s (DNNs), as quais têm superado o estado da arte para SR [22, 28].

Em 2014 Goodfellow et al.[29] propôs um novo modelo de DNN para sintetização de imagens, conhecido por GAN. Este tipo de DNN é formada por duas redes, uma geradora e outra discriminadora. A discriminadora aprende a distinguir imagens falsas das reais, a geradora aprende a criar imagens falsas, tentando sempre enganar a discriminadora. Assim, ambas as redes são treinadas de forma competitiva, adversárias, tendendo a um equilíbrio como na teoria dos jogos [30]. Ao final, na proposta original, o objetivo é que a rede geradora aprenda a gerar imagens plausíveis.

As GANs têm alcançado ótimos resultados em uma variedade de aplicações de geração de imagens, inclusive para reconstrução de imagens, como é o caso da tarefa de SR. No entanto, as GANs ainda necessitam de investigação para serem utilizadas em reconstrução de vídeos em aplicação de *streaming* de vídeo [9], de maneira a reduzir o tráfego nas infraestruturas de rede e também para elevar a *quality of experience* (QoE) da audiência dessas aplicações.

Considerando as restrições temporais de uma aplicação de distribuição de vídeo em escala global, e a redução do tráfego associado a essa distribuição, neste trabalho investiga-se as seguintes problemáticas:

*i*) As GANs podem ser aplicadas para reconstruir vídeos transmitidos em baixa resolução e aumentar a sua resolução original com boa qualidade perceptiva, contribuindo assim para reduzir o tráfego nas infraestruturas de rede?

*ii*) As GANs podem melhorar a QoE da audiência de aplicações de vídeo *streaming*, onde os vídeos são recebidos em uma resolução baixa e, uma GAN seja aplicada para elevar a resolução do vídeo, permitindo à audiência assistir aos vídeo em uma melhor qualidade do que a originalmente recebida?

A hipótese é que essas indagações sejam viáveis, tendo como fundamento recentes pesquisas disponíveis na literatura que mostram que GANs podem ser aplicadas para SR de imagens e vídeos [21, 31, 32, 33, 34, 20] e para compressão de imagens [21, 35, 36] obtendo resultados que superaram o estado da arte.

## 1.2 OBJETIVOS

O objetivo geral desta pesquisa é verificar se o uso de GANs para reconstrução de vídeos em alta resolução a partir de vídeos em baixa resolução pode contribuir para aprimorar aplicações de vídeo *streaming*, reduzindo o volume de dados transmitidos e elevando a QoE nas sessões, impactando em dois dos fatores mais importantes na medição da QoE: as interrupções e a qualidade das imagens.

Para alcançar o objetivo geral desta pesquisa delineou-se os seguintes objetivos específicos:

- Definir um *framework* de serviço de replicação de vídeo em nuvem com uso de modelo de SR baseado em GAN para redução de volume de dados replicados através das infraestruturas de redes de conexões internacionais;
- Adaptar modelo de SR baseado em GAN para restauração de vídeos de baixa para alta resolução, aprimorando a qualidade perceptiva das imagens, em serviço de replicação de vídeo em nuvem;
- Avaliar a qualidade pixel a pixel (*pixel-wise*) e perceptiva dos vídeos restaurados por modelos de SR;
- Analisar a redução de dados replicados através das infraestruturas de rede ao transmitir as matrizes dos vídeos em baixa resolução e comparar com o volume, quando transmitido em alta resolução;
- Definir um *framework* para distribuição de vídeos ao vivo com aprimoramento de qualidade perceptiva por super-resolução provida por computação de borda;
- Propor modelo de SR baseado em GAN para reconstrução de vídeos em tempo real de baixa para alta resolução, aprimorando a qualidade perceptiva das imagens, em serviço de vídeo *streaming* ao vivo;
- Avaliar a QoE em serviços de *streaming* de vídeo ao vivo com uso de SR provida por computação de borda.

## 1.3 MÉTODO DE PESQUISA

Nesta pesquisa utilizou-se abordagem quantitativa, com uso de técnicas estatísticas para análise e interpretação dos dados. Nas análises estatísticas foram utilizadas ferramentas do ecossistema SciPy[1] baseado em Python, entre elas, Pandas[2], Matplotlib[3] e Seaborn[4].

Nas atividades que envolveram processamento de imagens e vídeos foram utilizadas as ferramentas OpenCV[5] e FFmpeg[6]. Na implementação e treinamento dos modelos de DNN utilizou-se a plataforma Tensorflow[7] com a biblioteca Keras[8] como *back-end*. Os modelos foram treinados com uso de unidade de processamento gráfico (GPU, do inglês *graphics processing unit*) GeForce GTX 1080Ti e testados em GPU GeForce GTX 1070Ti. Os procedimentos técnicos para a coleta de informação da pesquisa seguem uma pesquisa experimental, com testes realizados em ambiente emulado e com controle das variáveis observadas.

A execução do projeto ocorreu em duas etapas: *i*) serviço de replicação de vídeo em nuvem com modelo de restauração por SR baseado em GAN e, *ii*) serviço para distribuição de vídeos ao vivo com aprimoramento de qualidade perceptiva por super-resolução provida por computação de borda.

Cada etapa foi subdividida em artefatos, que são resultados parciais e estão diretamente relacionados aos objetivos específicos.

### 1.3.1 ETAPA 1 – SERVIÇO DE REPLICAÇÃO DE VÍDEO EM NUVEM COM MODELO DE RESTAURAÇÃO POR SR BASEADO EM GAN

Nesta etapa foram desenvolvidos os artefatos que envolvem o serviço de replicação de vídeo em nuvem para redução do volume de tráfego nas infraestruturas de rede que interligam os servidores originais e substitutos. Os resultados desta etapa encontram-se nos capítulos 3, 4 e 5. Estão incluídos nesta etapa os seguintes artefatos:

- Artefato 1 - Definição da arquitetura de serviço de replicação de vídeo em nuvem: compreendeu a definição da arquitetura de um serviço de replicação de vídeo em baixa resolução entre servidores originais e substitutos. Nos servidores substitutos o *framework* faz uso de modelo de SR baseado em GAN para restaurar vídeos em alta resolução, reduzindo o volume de dados replicados através das infraestruturas de rede de conexões internacionais;

[1] https://www.scipy.org/
[2] https://pandas.pydata.org/
[3] https://matplotlib.org/
[4] https://seaborn.pydata.org/
[5] https://opencv.org/
[6] https://ffmpeg.org/
[7] https://www.tensorflow.org/
[8] https://keras.io

- Artefato 2 - Adaptação de modelo de SR baseado em GAN para restauração de vídeos de baixa para alta resolução: neste artefato foi realizado um mapeamento sistemático da literatura sobre SR com redes neurais, os resultados dessa revisão encontram-se nos capítulos 3 e 4. A revisão teve como objetivo construir uma fundamentação teórica sobre modelos de SR e posterior adaptar um modelo de SR apropriado para ser aplicado no *framework* definido no artefato 1;

- Artefato 3 - Avaliação da qualidade quantitativa e perceptiva dos vídeos restaurados por modelos de SR: neste artefato foram utilizadas métricas de avaliação de qualidade de vídeos para avaliar a qualidade pixel a pixel e perceptiva dos vídeos restaurados por modelos de SR;

- Artefato 4 - Análise da redução de dados replicados através das infraestruturas de rede: neste artefato é analisada a redução na quantidade de dados trafegados nas infraestruturas de redes que interligam servidores originais e substitutos quando são transmitidos os vídeos em baixa resolução ao invés de transmitidos em alta resolução. Os resultados dos artefatos 3 e 4 encontram-se no capítulo 5.

### 1.3.2 ETAPA 2 – SERVIÇO PARA DISTRIBUIÇÃO DE VÍDEOS AO VIVO COM APRIMORAMENTO DE QUALIDADE PERCEPTIVA POR SUPER-RESOLUÇÃO APOIADA POR COMPUTAÇÃO DE BORDA

Nesta etapa foram desenvolvidos os artefatos que envolvem a incorporação de SR em aplicação de vídeo ao vivo apoiada por computação de borda. A etapa envolve a definição de um *framework* para incorporar modelo de SR adequado para ser executado em tempo real, considerando um serviço de transmissão de vídeo ao vivo. Os resultados desta etapa encontram-se no capítulo 6. Estão incluídos nesta etapa os seguintes artefatos:

- Artefato 5 - Definição de um *framework* para distribuição de vídeos ao vivo com aprimoramento de qualidade perceptiva por super-resolução apoiada por computação de borda: este artefato compreende a definição de um *framework* para distribuição de vídeo ao vivo com uso de SR em servidores localizados na *multi-access edge computing* (MEC) para restaurar os vídeos em alta resolução antes de entregar à audiência, contribuindo assim, para elevar a QoE.

- Artefato 6 - Proposição de um modelo de SR de tempo real baseado em GAN: envolve o desenvolvimento de modelo de SR baseado em GAN para reconstrução de vídeos em tempo real de baixa para alta resolução, a ser aplicado em serviço de vídeo *streaming* ao vivo apoiado por computação de borda.

- Artefato 7 - Avaliação de QoE em serviço de *streaming* de vídeo ao vivo apoiado por SR provida por computação de borda: compreende a análise, por meio de

experimentação com uso do *framework* definido no artefato 5, da QoE em aplicações de vídeo *streaming* ao vivo. Para a análise, vários cenários foram considerados, por exemplo, com e sem o uso de SR, além de diferentes configurações de largura de banda e variados algoritmos de fluxo de taxa de bits adaptável (ABR, do inglês *adaptive bitrate streaming*).

## 1.4 CONTRIBUIÇÕES DESTA TESE

Os capítulos 3 e 4 fornecem uma revisão da literatura sobre métodos de SR que empregam redes neurais. No capítulo 3, são abordados os métodos que realizam SR utilizando apenas uma imagem como entrada. Já no capítulo 4, são apresentados os métodos que utilizam múltiplas imagens como entrada para gerar SR.

No capítulo 5, utilizou-se um método de SR por redes neurais para propor um serviço de replicação de vídeo em nuvem. Este capítulo apresenta as seguintes contribuições:

- Foi proposto um *framework* baseado em nuvem para o posicionamento de conteúdo, que reduz significativamente o tráfego de vídeo em infraestruturas de longa distância. Nesse *framework*, vídeos de baixa resolução são transferidos entre o servidor original e servidores substitutos distribuídos geograficamente. Um modelo eficiente baseado em GAN de SR reconstrói os vídeos em alta resolução.
- Foi desenvolvido um modelo de SR de vídeo como uma solução prática para ser utilizada em um sistema de entrega de vídeo sob demanda. Esse modelo aumenta a resolução dos vídeos em um fator de escala de 2×, mantendo uma qualidade perceptual indistinguível em comparação com os vídeos de referência.
- Foi apresentado um método para mapear a qualidade perceptual de vídeos reconstruídos em relação à representação do mesmo vídeo em diferentes níveis de parâmetro de quantização (QP, do inglês *quantization parameter*). Esse método é essencial para comparar a qualidade de um vídeo reconstruído por SR com a representação do mesmo vídeo em diferentes níveis de compressão.
- Por fim, foi avaliada a contribuição da SR na redução de dados e comparada com a redução por compressão. Além disso, foram analisadas as vantagens das duas abordagens combinadas. Os experimentos demonstraram que é possível reduzir a quantidade de tráfego na infraestrutura em nuvem em até 98,42% em comparação com a distribuição de vídeo com compressão sem perda de dados.

No capítulo 6, é proposto um serviço para distribuição de vídeo ao vivo com aprimoramento de qualidade por meio do SR com suporte de computação de borda. Neste capítulo, são apresentadas as seguintes contribuições:

- Foi proposto um *framework* que utiliza computação de borda para realizar a escalabilidade de vídeos em tempo real, por meio de um modelo de SR baseado em DNN, visando aprimorar a QoE em sessões de *streaming* ao vivo.
- Foram avaliados modelos de SR de tempo real com treinamento sensível ao conteúdo de vídeos similares e com aplicação de redes GANs para aprimorar perceptivamente a qualidade dos vídeos.
- Foi proposto um novo modelo de SR de tempo real que utiliza GAN e destilação de conhecimento, com o objetivo de aprimorar perceptivamente a qualidade dos vídeos.

Os resultados desta tese foram publicados em forma de artigo nos seguintes veículos:

- LIBORIO FILHO, J. M.; MELO, A. C.; OLIVEIRA J. A. . **Super-resolution with perceptual quality for improved live streaming delivery on edge computing**. Computer Networks. 2023.
- LIBORIO FILHO, J. M.; MELO, A. C. ; SILVA, M. P. . **Internet Video Delivery Improved by Super-Resolution with GAN**. Future Internet, v. 14, p. 1-24, 2022. Disponível em <https://www.mdpi.com/1999-5903/14/12/364>.
- LIBORIO FILHO, J. M.; DE SOUZA COELHO, MAIARA ; MELO, CESAR A. V. . **Super-resolution on Edge Computing for Improved Adaptive HTTP Live Streaming Delivery**. In: 2021 IEEE 10th International Conference on Cloud Networking (CloudNet), 2021, Cookeville. 2021 IEEE 10th International Conference on Cloud Networking (CloudNet), 2021. p. 104. Disponível em <https://ieeexplore.ieee.org/document/9657150>.
- LIBORIO FILHO, J. M.; MELO, A. C. . **A GAN to Fight Video-related Traffic Flooding: Super-resolution**. In: 11th Latin-American Conference on Communications, 2019, Salvador, BA, Brazil. IEEE LATINCOM 2019 Conference Proceedings, 2019. Disponível em <https://ieeexplore.ieee.org/document/8937966>.

## 1.5 ESTRUTURA DA TESE

Além deste capítulo de introdução, esta tese contém mais seis capítulos, conforme descritos a seguir:

O capítulo 2 apresenta a fundamentação teórica com os principais conceitos abordados na pesquisa, além de apresentar os trabalhos relacionados. No capítulo 3 apresenta-se o resultado da revisão de literatura sobre as principais abordagens para SR de única imagem baseadas em redes neurais profundas. No capítulo 4 encontra-se a revisão de literatura sobre as abordagens de SR baseadas em múltiplas imagens. Os resultados sobre aplicação

de SR para reduzir o tráfego gerado por replicação de vídeo em sistema de distribuição de conteúdo de vídeo na nuvem são apresentados no capítulo 5. No capítulo 6 apresenta-se o serviço para distribuição de vídeos ao vivo com aprimoramento de qualidade perceptiva por SR apoiada por computação de borda. Finalmente, no capítulo 7, são apresentadas as conclusões e trabalhos futuros.

# CAPÍTULO 2

# Fundamentação Teórica

Nesta seção, serão apresentados os conceitos fundamentais relacionados a esta pesquisa, que são essenciais para compreender a hipótese formulada. Além disso, será apresentado um conjunto de trabalhos relacionados à temática da pesquisa.

## 2.1 INFRAESTRUTURA DE TRANSPORTE DAS REDES MÓVEIS 5G

A infraestrutura de transporte 5G é composta por três blocos principais [37]: as redes de *fronthaul*, *midhaul* e *backhaul*, criadas com o surgimento da arquitetura de rede centralizada de acesso por rádio (C-RAN, do inglês *centralized/cloud radio access network*). O *fronthaul* é a parte da rede que conecta a unidade de rede remota (RRU, do inglês *remote radio unit*) diretamente a um ponto de agregação intermediário, conhecido como unidade distribuída (DU, do inglês *distributed unit*). Esse ponto de agregação, por sua vez, deve estar conectado à unidade centralizada (CU, do inglês *centralized unit*), que representa a rede de *midhaul*. Neste trabalho, utiliza-se o termo *fronthaul* para nos referir a uma rede que inclui o *midhaul*.

A rede *backhaul* conecta a CU à rede central e é composta por fibra dedicada, cobre, micro-ondas e, ocasionalmente, *links* de satélite. As CUs também desempenham o papel de ponto de agregação de *backhaul*, concentrando as conexões de *backhaul* de todas as estações de rádio dentro do alcance do núcleo da rede. Como resultado, todo o tráfego gerado passa pela rede de *backhaul* e se torna um gargalo de tráfego na infraestrutura de transporte 5G [38].

A otimização desses recursos de transmissão tem sido objeto de estudo e pesquisa.

A MEC aborda esse problema trazendo aplicações e conteúdos mais próximos aos usuários finais. Ela oferece recursos de computação em nuvem, em um ambiente de serviço de tecnologia da informação (IT, do inglês *information technology*) na borda da rede, para desenvolvedores de aplicativos e provedores de conteúdo [39]. Em resumo, ao implantar esse serviço próximo aos usuários, é possível atender a requisitos rígidos de latência e aproveitar a largura de banda disponível na última milha.

## 2.2 VÍDEO STREAMING

O *streaming* de vídeo é uma aplicação que se beneficia dos conceitos associados à MECs devido à sua sensibilidade a atrasos e suas demandas de armazenamento e computação. Essas características são consequência da abordagem das aplicações de *streaming* de vídeo em um cenário de fragmentação da audiência e necessidade de escala econômica. Nesse contexto, o vídeo é processado para ser exibido em taxa de bits e formato adequados, por meio do processo de transcodificação, que utiliza algoritmos de codificação diferentes do vídeo original. Um vídeo transcodificado pode ter várias representações, combinando taxa de bits e resoluções [40].

A sensibilidade ao atraso representa um desafio adicional para as aplicações de *streaming* de vídeo em grande escala. Para lidar com esse desafio, foi desenvolvida uma técnica chamada de *streaming* de taxa de bits adaptável (ABR, do inglês *adaptive bitrate streaming*), que é amplamente utilizada pelas principais plataformas de vídeo. Essa técnica utiliza os resultados da transcodificação para ajustar a qualidade visual do vídeo transmitido de acordo com as condições de transporte durante a sessão de *streaming* de vídeo. Durante a transcodificação, o vídeo é dividido em pequenos segmentos, geralmente de 2 a 4 segundos, e é codificado em diferentes taxas de bits. Durante a sessão, os *players* de vídeo podem solicitar segmentos com taxas de bits correspondentes aos recursos de transmissão e processamento disponíveis, visando aprimorar a experiência dos usuários.

Assim como qualquer outra aplicação, as aplicações de *streaming* de vídeo buscam engajar sua audiência. Esse engajamento está diretamente relacionado à experiência percebida durante as sessões de vídeo. Existem diversos estudos dedicados a analisar e quantificar a qualidade de experiência (QoE, do inglês *quality of experience*) dos usuários de aplicativos, que é definida como o grau de satisfação proporcionado a eles durante as interações [41, 42]. Neste trabalho, calculamos a QoE das sessões de vídeo utilizando um modelo proposto por Yin et al.[43], que leva em consideração eventos como interrupções, mudanças frequentes na qualidade visual e longos períodos de espera para iniciar a reprodução do conteúdo, os quais podem afetar o engajamento das sessões de vídeo a longo prazo.

## 2.3 SUPER-RESOLUÇÃO

Imagens de alta definição são um dos principais elementos para fornecer sessões de vídeo com alta qualidade de experiência (QoE). A entrega dessas imagens apresenta um desafio de decisão com várias camadas, e o uso de Super-Resolução (SR) tem mostrado resultados promissores como uma ferramenta adicional nesse processo de decisão. Resumidamente, a SR é uma tarefa de visão computacional que visa reconstruir uma imagem de alta resolução (HR) com base em imagens adquiridas da mesma cena, denominadas imagens de baixa resolução (LR, do inglês *low-resolution*) [44].

A SR pode ser aplicada a uma única imagem, chamada de Super-Resolução de Imagem Única (SISR, do inglês *sigle image super-resolution*), em que a entrada e a saída são definidas por uma única imagem. Também pode ser aplicada à super-resolução multiquadros, em que várias imagens semelhantes são usadas como entrada para gerar uma imagem de saída. Essa abordagem é chamada de Super-Resolução de Vídeo (VSR, do inglês *video super-resolution*) [44]. Tanto a SISR quanto a VSR podem ser aplicadas para restaurar vídeos, mas diferem na quantidade de quadros de entrada para o modelo. Na SISR, apenas informações espaciais disponíveis em um único quadro são exploradas, enquanto na VSR, informações espaciais e temporais são exploradas, ou seja, além do quadro de referência, os quadros vizinhos também são utilizados para restaurar o quadro de referência.

A abordagem mais recente para resolver a tarefa de SR leva em consideração o funcionamento do sistema de visão humana. Essa abordagem é chamada de Super-Resolução Perceptiva, devido à metodologia de treinamento do modelo de SR, em que a função de perda aplicada correlaciona a qualidade da imagem super-resolvida com a forma como o sistema visual humano percebe as imagens. Modelos de SR baseados em funções de perda pixel a pixel (pixel-wise) geram imagens suavizadas e sem detalhes de alta frequência devido ao problema de média considerado na abordagem pixel a pixel [21]. Estudos têm mostrado que funções de perda baseadas em características de alto nível extraídas por modelos de DNN e treinamento com GANs geram imagens mais realistas [22, 45, 46].

Nos últimos anos, a tarefa de SR tem recebido atenção da comunidade de pesquisa, que tem abordado o problema utilizando redes neurais profundas [47, 48, 49]. Os resultados mais significativos têm utilizado *convolutional neural network*s (CNNs) e GANs, sendo que esta última introduz o treinamento adversarial e a função de erro perceptiva. No capítulo 3, é apresentada uma revisão da literatura dos métodos de SR de imagem única baseados em DNN, e no capítulo 4, são apresentados métodos de SR multiquadros também baseados em DNN.

## 2.4 MÉTRICAS DE AVALIAÇÃO DE QUALIDADE DE VÍDEO

Na literatura, existem numerosas métricas e métodos de avaliação de qualidade de vídeo (VQA, do inglês *video quality assessment*) que utilizam diversos recursos, incluindo características visuais, recursos de percepção e recursos psicológicos [50]. Dessa forma, as métricas VQA existentes podem ser classificadas de diferentes maneiras.

Alguns métodos de avaliação são baseados em um vídeo de referência, sendo chamados de métricas de referência completa (FR, do inglês *full-reference*), enquanto outros métodos não requerem um vídeo de referência e são chamados de métricas sem referência (NR, do inglês *no-reference*). Há também métodos denominados métricas de referência reduzida (RR, do inglês *reduced-reference*), que utilizam apenas alguns recursos do vídeo de referência, como quantidade de movimento ou detalhes espaciais, e fazem uma comparação entre a referência e o vídeo de teste com base nesses recursos.

Os métodos de avaliação da qualidade também podem ser categorizados como objetivos ou subjetivos [51]. Os métodos objetivos permitem prever automaticamente a qualidade do vídeo. Já os métodos subjetivos avaliam a qualidade do vídeo percebida por um observador humano. No entanto, esse tipo de método torna-se impraticável para a maioria das aplicações devido ao envolvimento humano no processo. No entanto, os estudos subjetivos fornecem informações valiosas que ajudam a avaliar o desempenho dos métodos objetivos, além de fornecerem meios para alcançar o objetivo final de combinar a percepção humana.

Nesta pesquisa, foram utilizadas cinco métricas objetivas, sendo quatro do tipo FR e uma do tipo NR, para avaliar a qualidade dos vídeos super-resolvidos. As duas primeiras métricas são baseadas em comparação pixel a pixel, ou seja, levam em consideração os pixels para determinar a qualidade. As outras três métricas são perceptivas, ou seja, utilizam meios que se assemelham ao sistema visual humano na percepção da qualidade.

### 2.4.1 MÉTRICAS DE QUALIDADE PIXEL A PIXEL

São métricas objetivas de avaliação da qualidade de vídeo que se baseiam apenas em medidas de distorção comuns e utilizam a comparação pixel a pixel. Neste trabalho foram utilizadas duas métricas pixel a pixel.

#### 2.4.1.1 MÉTRICA PSNR

A relação sinal-ruído de pico (PSNR, do inglês *peak signal-to-noise ratio*) avalia a qualidade dos vídeos calculando a relação entre o valor máximo de um sinal e a potência do ruído de distorção em decibéis. O valor mais alto de PSNR indica a melhor qualidade

de vídeo. Seu valor é calculado utilizando a Equação (2.1),

$$PSNR = \frac{1}{N}\sum_{i=1}^{N} 20\log_{10}\frac{\max(f_i^{HR})}{\sqrt{MSE(f_i^{HR}, f_i^{SR})}}, \tag{2.1}$$

onde $N$ é o número total de quadros. A $MSE$ é calculado por,

$$MSE = \frac{1}{WH}\sum_{x=1}^{H}\sum_{y=1}^{W}\left(f_{x,y}^{HR} - f_{x,y}^{SR}\right)^2, \tag{2.2}$$

onde $W$ e $H$ são as dimensões dos quadros.

Embora o PSNR não seja uma métrica apropriado para refletir a percepção visual humana [21, 45], tem sido aplicado na avaliação da qualidade do vídeo para revelar a distorção entre os sinais.

#### 2.4.1.2 MÉTRICA RMSE

O erro quadrático médio ou o desvio quadrático médio (RMSE, do inglês *root-mean-square error*) é uma das métricas clássicas mais usadas para avaliar a qualidade das previsões. Ela mostra até que ponto as previsões diferenciam dos valores verdadeiros medidos usando a distância euclidiana.

Para calcular o RMSE, calcula-se o residual (diferença entre a previsão e a referência) para cada pixel dos quadros do vídeo, calcula a norma do residual para cada pixel, calcula a média dos resíduos e obtêm-se a raiz quadrada dessa média. O RMSE pode ser expressa como,

$$RMSE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{WH}\sum_{x=1}^{H}\sum_{y=1}^{W}\left(f_{x,y}^{HR} - f_{x,y}^{SR}\right)^2}, \tag{2.3}$$

onde $W$ e $H$ são as dimensões dos quadros, e $N$ é o número total de quadros.

### 2.4.2 MÉTRICAS DE QUALIDADE PERCEPTIVAS

Avaliar a qualidade de vídeos por meio da visão humana é desejável, porém se torna inviável em aplicação de grande escala. Vários trabalhos de avaliação objetiva têm sido desenvolvidos com base no sistema visual humano (HVS, do inglês *human visual system*). A seguir, apresentam-se três métricas perceptivas que são utilizadas neste trabalho.

#### 2.4.2.1 MÉTRICA LPIPS

A métrica *Learned perceptual image patch similarity* (LPIPS) [52] avalia a distância perceptiva entre os vídeos distorcidos e o vídeo de referência. Distância zero significa

que dois vídeos apresentam notória equivalência. Esta métrica mede a distância usando um espaço estabelecido por características de alto nível extraídas dos quadros. Para a construção desse espaço, redes neurais profundas, criadas para classificação de imagens, são utilizadas para extrair essas características, a saber: VGG [53], SqueezeNet [54], e AlexNet [55].

Neste trabalho, construímos o espaço de características LPIPS usando a rede SqueezeNet [54] devido ao seu custo computacional reduzido e resultados semelhantes em comparação com VGG e AlexNet como mostrado em Zhang et al.[52]. No mesmo trabalho, os autores mostraram que essas três redes aprenderam representações de mundo relacionadas a julgamentos perceptivos. Portanto, o LPIPS se correlaciona fortemente com métricas de percepção humana, como a *just-noticeable-difference* (JND). Além disso, o LPIPS generaliza diferentes distorções, inclusive aquelas geradas pelos algoritmos SR.

#### 2.4.2.2 MÉTRICA VMAF

A métrica *video multi-method assessment fusion* (VMAF) [56, 57] correlaciona avaliações subjetivas, refletindo, indiretamente, a qualidade percebida pela visão humana. Essa métrica estima a qualidade de um vídeo distorcido em relação ao seu vídeo de referência. Primeiro, computando avaliações com métricas elementares, em seguida, gerando a pontuação final, com o regressor *support vector machine* (SVM), para fundir os valores alcançados pelas métricas elementares. A pontuação final, gerada pela VMAF, fica no intervalo de 0 a 100, onde 100 representa que o vídeo distorcido é igual ao de referência.

A VMAF tem sido utilizada pela indústria para avaliação de qualidade de vídeos, *e.g.*, pela Netflix [1]. A justificativa é que a VMAF tem forte correlação com avaliações subjetivas realizadas por seres humanos, podendo ser utilizada através de sistemas automatizados, assim, reduzindo custos e agilizando a tarefa de avaliação de qualidade.

#### 2.4.2.3 MÉTRICA PI

A métrica *Perception index* (PI) foi proposta em Blau et al.[46] para avaliação de qualidade perceptivas de imagens na competição PIRM 2018. Essa métrica combina as medidas de qualidade de outras duas métricas conforme definido na Equação 2.4, Ma et al.[58] e NIQE [59], a seguir:

$$PI = \frac{1}{2}\left((10 - Ma) + NIQE\right), \tag{2.4}$$

onde $Ma$ é o valor computado pela métrica proposta em Ma et al.[58] e NIQE é o valor computador pela métrica NIQE [59]. Destaca-se que ambas são métricas *no-reference* (NR).

[1] <https://medium.com/netflix-techblog/vmaf-the-journey-continues-44b51ee9ed12>, acessado em 19 de abril de 2022

## 2.5 TRABALHOS RELACIONADOS

Nesta seção, serão apresentados os trabalhos relacionados a esta pesquisa. Inicialmente, serão abordados estudos que aplicam modelos de SR para aprimorar serviços de distribuição de vídeo na Internet. Em seguida, serão apresentados trabalhos relacionados à aplicação de computação de borda para melhorar a transmissão de *streaming* de vídeos. Por fim, serão discutidos trabalhos que utilizam o método JND para avaliar a qualidade de vídeos.

### 2.5.1 SUPER-RESOLUÇÃO POR REDES NEURAIS PROFUNDAS APLICADAS EM DISTRIBUIÇÃO DE VÍDEOS NA INTERNET

Em Yeo, Do e Han[9] e Yeo et al.[6], os autores destacam como as DNNs podem mudar a distribuição de vídeos pela Internet devido à sua performance em tarefas de restauração de imagens e vídeos. Eles enfatizam a evolução do poder computacional dos dispositivos utilizados pelas audiências, prevendo que em um futuro próximo essas demandas de processamento das DNNs possam ser realizadas nesses dispositivos. Com base nessas projeções, os autores propõem um *framework* com uma arquitetura de CDN sensível ao conteúdo, que agrupa vídeos similares e gera uma versão alternativa em baixa resolução para distribuição aos clientes. Nos agrupamentos, é gerado um modelo de SR baseado em DNNs, que é entregue ao cliente juntamente com os vídeos.

Em Yeo et al.[6], o *framework* utiliza *streaming* adaptativo dinâmico sobre HTTP (DASH, do inglês *dynamic adaptive streaming over HTTP*) e incorpora um algoritmo de taxa de bits adaptativa (ABR) baseado em aprendizado por reforço (RL, do inglês *reinforcement learning*) [60] para lidar com a heterogeneidade dos recursos de rede e dos dispositivos dos clientes. No lado do cliente, o vídeo é restaurado para uma resolução maior utilizando o modelo de SR, resultando em vídeos com melhor qualidade e menor consumo de largura de banda. O modelo de SR é treinado *offline* usando a função de perda pixel a pixel, realizando aumento de escala de 2×, 3× e 4×.

No estudo de Hu, Misra e Katti[61], a SR é utilizada para melhorar a qualidade da imagem em videoconferências. Um modelo treinado *offline* é empregado para aumentar a resolução dos quadros de vídeo no lado do cliente. O conjunto de treinamento é gerado a partir de sessões anteriores de videoconferências com quadros de baixa resolução (*low-resolution* (LR)) e alta resolução (HR). O modelo de SR desenvolvido é baseado em DNNs treinadas com a função de perda pixel a pixel. Os autores relatam o uso de um fator de escala de 2× com o objetivo de melhorar a qualidade do vídeo, sem considerar a integração do modelo de SR em um aplicativo de videoconferência ou medir a QoE do usuário.

Em Kim et al.[62], é proposto um *framework* de ingestão de vídeo ao vivo que adapta a qualidade do *streaming* na origem. O serviço utiliza os recursos do servidor

de ingestão para aplicar a SR quando o fluxo chega ao servidor, minimizando assim a sobrecarga nos clientes de ingestão. Os autores afirmam que esse serviço produz alta qualidade nas transmissões ao vivo para a audiência, mesmo quando a rede de ingestão está congestionada, pois desacopla a qualidade da transmissão ao vivo da largura de banda do cliente de ingestão. A função de perda utilizada para treinar o modelo de SR é pixel a pixel, e o treinamento ocorre *online*, utilizando pequenos fragmentos dos quadros de alta resolução enviados pelo cliente de ingestão. Os vídeos são transportados do cliente de ingestão para o servidor de ingestão usando o protocolo WebRTC[2].

Em Chen et al.[63], é proposto um *framework* de ingestão de vídeo que considera uma largura de banda limitada no *uplink*. A arquitetura do sistema inclui um módulo de compressão de quadros-chave baseado em super-resolução, treinamento *online* de modelo de super-resolução e um módulo de adaptação de taxa de bits ao vivo. O módulo de compressão aplica uma taxa de compressão mais alta aos quadros-chave, assumindo a disponibilidade de uma técnica de super-resolução durante a decodificação do fluxo de ingestão. No lado do servidor, a SR é aplicado para restaurar os quadros-chave. O modelo utiliza treinamento *online* e um fator de escala de aumento de 2× e 3×.

Em Khani, Sivaraman e Alizadeh[64], é proposta uma abordagem de compressão para a entrega de vídeos na Internet, que combina SR com algoritmos de compressão existentes. A técnica divide o vídeo de entrada em dois fluxos de bits - um fluxo de conteúdo e um fluxo de modelo - cada um com uma taxa de bits exclusiva que pode ser ajustada independentemente do outro. O fluxo de conteúdo utiliza um *codec* padrão para enviar quadros de baixa resolução com baixa taxa de bits. Ao decodificar o fluxo de conteúdo, os quadros de baixa resolução são passados pelo modelo de super-resolução para reconstruir os quadros de alta resolução. O modelo é treinado *offline* utilizando a função de perda pixel a pixel e um fator de escala de aumento de 2×.

Na maioria das pesquisas relacionadas mencionadas [9, 6, 61, 62, 63, 64], foram utilizados modelos de SR baseados em DNNs. No entanto, todos eles treinaram esses modelos utilizando funções de perda pixel a pixel. Em contraste, nesta pesquisa, foram utilizadas funções de perda perceptiva e GAN durante o treinamento dos modelos. Além disso, as aplicações específicas abordadas neste trabalho, voltadas para a distribuição de vídeos, apresentam duas propostas distintas de *frameworks* para esse fim na Internet.

A primeira proposta, descrita no capítulo 5, trata da replicação de vídeos para aplicações sob demanda entre servidores de uma CDN, visando minimizar os custos de movimentação de conteúdo de vídeo entre servidores conectados por *links* de longa distância. A segunda proposta, descrita no capítulo 6, aborda os desafios da entrega de vídeos adaptativos em redes móveis de banda larga, como as 5G, que frequentemente apresentam gargalos nos *links* de *backhaul*.

[2] https://webrtc.org/

Tabela 1 – Comparativo com os trabalhos relacionados.

| Trabalhos | Função de Perda | | Forma de Treinamento | | Execução da SR | | | Streaming ao vivo | | Fator Escala | ABR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Perceptiva | pixel a pixel | Online | Offline | Na nuvem | Na MEC | No cliente | Ingestão | Entrega | | |
| [9] | | √ | | √ | | | √ | | | 2×, 3×, 4× | |
| [6] | | √ | | √ | | | √ | | | 2×, 3×, 4× | √ |
| [61] | | √ | | √ | | | √ | | | 2× | |
| [62] | | √ | √ | | √ | | | √ | | 3× | |
| [63] | | √ | √ | | √ | | | √ | | 2×, 3× | |
| [64] | | √ | | √ | | | √ | | | 2× | |
| Cap. 5 | √ | | | √ | √ | | | | | 2× | |
| Cap. 6 | √ | | √ | √ | | √ | | | √ | 2×, 3×, 4× | √ |

Fonte: De autoria própria a partir de dados dos autores.

A Tabela 1 lista os trabalhos relacionados e seus principais pontos de forma resumida.

### 2.5.2 STREAMING DE TAXA DE BITS ADAPTÁVEIS ASSISTIDOS POR COMPUTAÇÃO DE BORDA

Em Wang et al.[65] e Wang et al.[7], uma estrutura de *crowd cast* assistida por borda foi proposta para otimizar a QoE de serviços de *crowdsourcing* de transmissão ao vivo usando *reinforcement learning* (RL). Um servidor de borda recebe conteúdo de alta taxa de *bits* de uma CDN, codifica e o entrega aos visualizadores. A aprendizagem por reforço é combinada com rede neural profunda para executar a política de agendamento com base em informações de rede em tempo real, para personalizar a QoE. Assim, o modelo aprende estratégias para escalonar e transcodificar usando um modelo de rede ator crítico de vantagem assíncrono (A3C, do inglês *asynchronous advantage actor-critic*).

Dogga et al.[66] apresentam medidas para motivar serviços de vídeo ao vivo revisarem suas infraestruturas. Avalia-se o uso da computação de borda para o apoio à transcodificação e transmissão, com o propósito de mitigar a sobrecarga ocasionada por conteúdos de vídeo gerados pelos usuários. Portanto, a maior carga de computação seria mantida na borda e apenas uma pequena fração passaria pela infraestrutura.

Em Wang et al.[67], os autores propõem um *framework edge-to-cloud* para aplicações de análise de vídeos em tempo real, como detecção de objetos e navegação autônoma. Os vídeos são enviados para uma nuvem em baixa resolução e são reconstruídos para uma super-resolução antes que a tarefa de análise do vídeo seja realizada pela aplicação. A motivação dessa abordagem é o equilíbrio entre a precisão da aplicação de análise de vídeos e a redução da demanda por largura de banda, da borda para a nuvem.

Em nosso estudo, ao contrário das abordagens anteriores que se concentraram na avaliação do potencial da SR em dispositivos, explora a viabilidade de implementar a SR em servidores na nuvem e na borda da rede. No capítulo 6, apresenta-se um *framework*

chamado ***On-edge enhanced live streaming with super-resolution*** **(ELiveSR)**, que utiliza um modelo de SR e computação de borda para fornecer *streaming* de vídeo ao vivo usando redes móveis 5G.

Inicialmente, projetou-se um modelo leve de SR baseado em DNN para atender ao requisito de processamento em tempo real na MEC. Em seguida, implementou-se um procedimento de aprendizado *online*, visto que modelos de SR sensíveis ao conteúdo são mais apropriados aos catálogos das plataformas de vídeo. Posteriormente, implantou-se um algoritmo para detectar o melhor fator de *upscaling* para o *streaming* de vídeo *LR* entrante na MEC. Finalmente, utilizou-se o modelo de SR para construir um conjunto de representações do vídeo demandadas em sessões de *streaming* com taxa *bits* adaptável.

Essas melhorias permitiram a automação da seleção do melhor modelo de SR para cada conteúdo de vídeo, o equilíbrio entre alta qualidade e os requisitos de tempo para oferecer sessões de transmissão ao vivo de alta QoE e a incorporação da tecnologia *dynamic adaptive streaming over HTTP* (DASH), que tem sido usada para tratar o problema da heterogeneidade de recursos no lado do cliente.

### 2.5.3 AVALIAÇÃO SUBJETIVA PARA QUALIDADE DE VÍDEO

O JND é a diferença entre dois sinais no limiar de detectabilidade [68] e é aplicado para entender a sensibilidade do sistema visual humano. No contexto de imagem e vídeo, Watson[69] propôs uma maneira de utilizar JND para avaliar a qualidade de vídeos, posteriormente, Lin et al.[70] propuseram um método iterativo para avaliar a qualidade de vídeo utilizando o JND.

Em Wang et al.[71] foi constituída uma base de dados de vídeos denominada VideoSet que possui avaliações de qualidade perceptivas baseadas na metodologia JND [70]. A base contém 220 vídeos em quatro resoluções distintas, em que cada vídeo foi codificado em 52 versões variando os níveis de distorção alterando-se o parâmetro de quantização (QP, do inglês *quantization parameter*). A qualidade de cada vídeo foi avaliada subjetivamente por 30 pessoas, em média, seguindo a metodologia JND. A partir de testes subjetivos, foi possível definir para cada vídeo, três pontos JND identificados por valor do QP.

Zhang et al.[52] propuseram a métrica LPIPS que mede a distância perceptiva entre duas imagens baseadas em características extraídas por DNN de classificação de imagens. Os resultados da métrica LPIPS foram comparados com avaliações JND, apresentando forte correlação, superando a correlação de outras métricas para avaliação indiretamente de similaridade perceptiva, o que evidenciou que os julgamento JND tem forte relação com representações visuais das características extraídas por DNN.

No capítulo 5 deste trabalho, foram realizados estudos experimentais utilizando o

conjunto de dados VideoSet [71]. Adicionalmente, apresenta-se um novo método que se baseia na métrica LPIPS para mapear pontos JND de vídeos super-resolvidos.

# CAPÍTULO 3

# Super-Resolução de Única Imagem Baseada em Redes Neurais Profundas

Neste capítulo, será apresentada uma revisão do estado da arte em relação ao problema de super-resolução de imagem. Serão descritas as principais abordagens que empregam redes neurais profundas, as quais têm se destacado como o estado da arte para a super-resolução nos últimos anos.

As abordagens estão organizadas em duas categorias: super-resolução de única imagem (SISR, do inglês *sigle image super-resolution*) - que será abordada neste capítulo - e super-resolução multiquadro (MFSR, do inglês *multi-frame super-resolution*), também conhecida como super-resolução de vídeo (VSR, do inglês *video super-resolution*), que será discutida no capítulo 4.

A super-resolução de única imagem é um problema de interesse no campo do processamento de imagem digital. Seu objetivo é recuperar uma imagem de alta resolução (HR) a partir de uma imagem de baixa resolução (LR). O SISR tem aplicações em várias áreas onde imagens de alta qualidade são desejáveis, como em imagens médicas [72], reconhecimento de faces [73], imagens por satélite [74, 75] e vigilância [76].

A SISR é definido como a tarefa de obter uma imagem super-resolvida (SR), representada por imagem por super-resolução ($I^{SR}$), a partir de uma imagem de baixa resolução (LR), representada por imagem de baixa resolução ($I^{LR}$), que é obtida ao reduzir por uma escala $r$ a imagem original correspondente à imagem de alta resolução (HR), representada por imagem de alta resolução ($I^{HR}$). No processo de redução de escala, é comum utilizar um filtro gaussiano seguido de interpolação bicúbica. Esses métodos

são populares e estão implementados em importantes ferramentas de processamento de imagem, como MATLAB[1], PILL[2] e OpenCV[3].

As imagens em LR e HR são representadas por tensores de tamanho $H$x$W$x$C$ e $rH$x$rW$x$C$, respectivamente, onde $H$ representa a altura, $W$ a largura e $C$ os canais de cores. Os espaços de cores mais utilizados para SR são o RGB (*red, green, and blue*) para imagens coloridas com três canais e o YCbCr (*Y: luminance; Cb: chrominance-blue; and Cr: chrominance-red*) quando se utiliza apenas um canal de cor, o de luminância $Y$, por ser mais sensível à percepção humana.

Nos últimos anos, os métodos baseados em redes neurais para tarefas de SISR têm alcançado um melhor desempenho e se tornaram o estado da arte. Como resultado, diversos modelos baseados em redes neurais foram propostos. A seguir, são apresentadas as principais abordagens utilizadas, juntamente com um modelo representativo descrito em cada uma delas.

## 3.1 ABORDAGEM COM REDES NEURAIS CONVOLUCIONAIS

As redes neurais convolucionais (CNNs, do inglês *convolutional neural network*s), também conhecidas como *convolutional network* (ConvNet), foram inspiradas na estrutura do córtex cerebral. Nos últimos anos, com a implementação de operações convolucionais utilizando o poder computacional das *graphics processing unit* (GPU) [55], as CNNs têm alcançado resultados promissores em diversas áreas, incluindo o reconhecimento e classificação de imagens, superando o desempenho da visão humana [77].

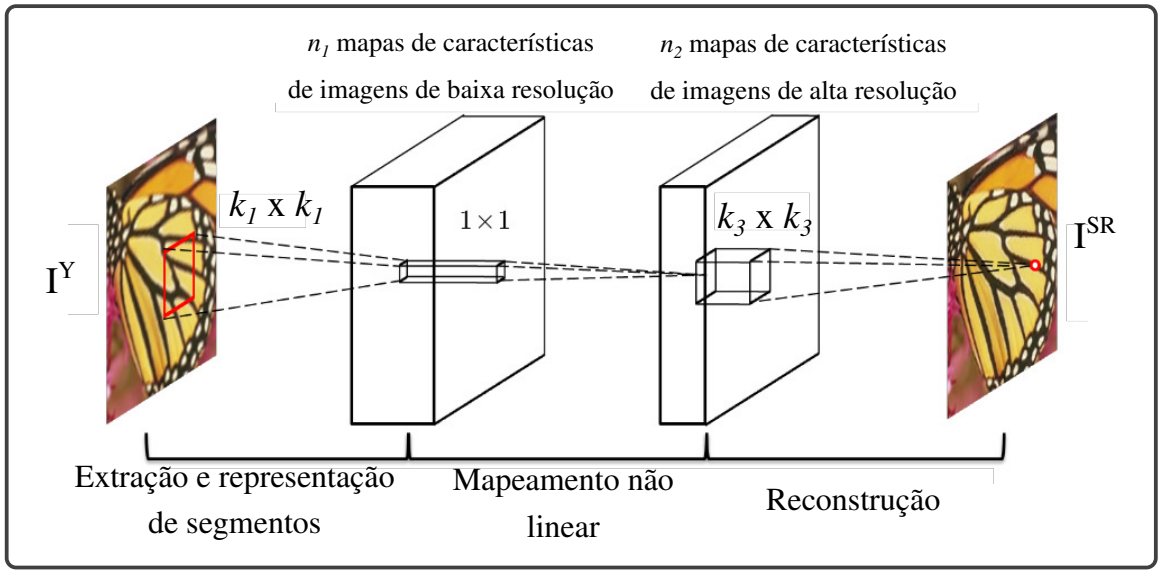
Na área de SR, as CNNs foram inicialmente utilizadas por Dong et al.[78] [79], que propuseram o modelo denominado *Super-resolution convolutional neural networks* (SRCNN). Nesse modelo, a super-resolução é realizada primeiramente redimensionando a imagem $I^{LR}$ por meio de uma interpolação bicúbica, resultando na imagem $I^{Y}$. O objetivo final do modelo é recuperar a imagem $I^{SR}$, que seja similar à imagem original $I^{HR}$, a partir da imagem $I^{Y}$. Apesar de a imagem $I^{Y}$ ter a mesma dimensão que $I^{HR}$, os autores consideram-na de baixa resolução.

A Figura 1 ilustra a arquitetura do modelo SRCNN, que recebe como entrada a imagem $I^{Y}$. A primeira camada convolucional extrai um conjunto de mapas de características. A segunda camada mapeia esses mapas de forma não linear para representações de segmentos de alta resolução. A última camada combina as previsões dentro de uma vizinhança espacial para produzir a imagem final de alta resolução $I^{SR}$.

O modelo SRCNN representado pela função $F(I^{Y})$ consiste em três operações:

[1] https://www.mathworks.com
[2] https://pillow.readthedocs.io
[3] https://opencv.org

Figura 1 – Arquitetura do modelo SRCNN.




FONTE: Adaptado de Dong et al.[78].

1. **Extração e representação de fragmentos:** essa operação é realizada pela primeira camada de convolução do modelo, que extrai $n_1$ mapas de características da imagem $I^Y$. Formalmente, essa operação é definida pela Equação (3.1),

$$F_1(I^Y) = \phi(W_1 * I^Y + B_1), \tag{3.1}$$

onde $W_1$ e $B_1$ representam, respectivamente, pesos e viés, $\phi$ é a função de ativação *rectified linear unit* (ReLU) [80]. O peso $W_1$ é um tensor de tamanho $c \times k_1 \times k_1 \times n_1$, sendo $c$ o número de canais da imagem, $k_1$ é a dimensão do filtro e $n_1$ é o número de filtros utilizados.

2. **Mapeamento não linear:** os mapas extraídos em $F_1$ são a entrada para a camada de mapeamento não linear $F_2$. Ou seja, os $n_1$ mapas de características de $F_1$ são mapeados para $n_2$ novos mapas de características, que são a saída de $F_2$, definida pela Equação (3.2),

$$F_2(I^Y) = \phi(W_2 * F_1(I^Y) + B_2), \tag{3.2}$$

sendo $W_2$ o tensor de pesos com dimensão $n_1 \times 1 \times 1 \times n_2$ e $B_2$ é o viés que tem dimensão $n_2$. Cada um dos mapas de características $n_2$ representa um segmento de alta resolução que é utilizado na reconstrução da imagem $I^{SR}$.

3. **Reconstrução:** essa operação reconstrói a imagem a partir dos segmentos de alta resolução provindos de $F_2$. A reconstrução é definida pela Equação (3.3),

$$F_3(I^Y) = W_3 * F_2(I^Y) + B_3, \tag{3.3}$$

aqui $W_3$ tem dimensão $n_2 \times k_3 \times k_3 \times c$ e $B_3$ é um tensor de dimensão $c$. Como saída de $F_3$ espera-se a imagem reconstruída $I^{SR}$ semelhante à imagem original $I^{HR}$.

Uma configuração típica para os valores relativos aos filtros é $k_1 = 9$, $k_2 = 1$, $k_3 = 5$, $n_1 = 64$, $n_2 = 32$ e $n_3 = c$; outras configurações também foram testadas em Dong et al.[78]. Para treinar o modelo deve-se otimizar os parâmetros $\theta = \{W1, W2, W3, B1, B2, B3\}$ de maneira a minimizar o erro entre a imagem reconstruída $I^{SR}$ e a imagem original $I^{HR}$. Para isso, utilizou-se o otimizador estocástico gradiente descendente [81] com a função de erro $L(\theta)$ sendo a *mean squared error* (MSE), definida na Equação (3.4).

$$L(\theta) = \frac{1}{r^2WH} \sum_{x=1}^{rH} \sum_{y=1}^{rW} (I_{x,y}^{HR} - F_\theta(I_{x,y}^{BI}))^2 \tag{3.4}$$

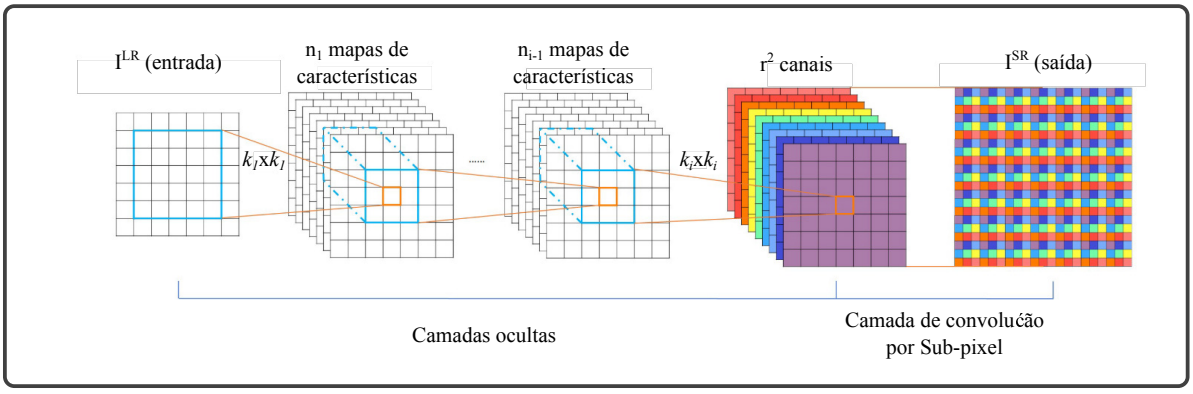
O modelo SRCNN demonstrou superar os métodos clássicos de SR não baseados em redes neurais, como os baseados em representação esparsa [82], o [83], incorporação de vizinhos [84, 85] e regressão de vizinhança ancorada [86]. No entanto, uma desvantagem desse método é o alto tempo de execução, devido ao processo de convolução ser realizado na imagem $I^Y$, que tem o mesmo tamanho que a imagem $I^{HR}$. Isso limita a utilização prática de um maior número de parâmetros treináveis no modelo, como mais filtros em cada camada de convolução ou tornar o modelo mais profundo, adicionando mais camadas. Apesar disso, o modelo SRCNN serviu de inspiração para o desenvolvimento de novos modelos de SR baseados em DNNs.

## 3.2 ABORDAGEM COM CONVOLUÇÃO POR SUBPIXEL

O método SRCNN, apresentado na seção anterior, realiza o redimensionamento da imagem antes da primeira camada de convolução, ou seja, a imagem de entrada da rede já possui dimensões de HR. Isso apresenta principalmente duas desvantagens. Primeiro, antes de prosseguir com a etapa de aprimoramento da imagem, é necessário aumentar o tamanho de $I^{LR}$, o que resulta em operações de convolução mais custosas computacionalmente. Isso pode ser problemático para uma rede convolucional, pois a velocidade de processamento está diretamente relacionada ao tamanho da imagem de entrada. Em segundo lugar, utiliza-se o métodos de interpolação bicúbica para redimensionar a imagem. No entanto, esse método não adiciona informações relevantes para melhorar a qualidade da imagem [87].

Shi et al.[87] propuseram o método *Efficient sub-pixel convolutional neural networks* (ESPCN), que realiza o redimensionamento da imagem apenas no final da rede de convolução. A imagem $I^{SR}$ é obtida a partir dos mapas de características extraídos da imagem $I^{LR}$. Em outras palavras, a tarefa de redimensionar a imagem é realizada por uma camada eficiente de convolução por subpixel, que compõe a última camada da rede, como ilustrado na Figura 33a. O aprimoramento da qualidade da imagem é aprendido pelos filtros das camadas de convolução anteriores à camada de subpixel. Para uma rede composta por $L$

Figura 2 – Arquitetura do modelo ESPCN.




FONTE: Adaptado de Shi et al.[87].

camadas, o modelo ESPCN pode ser descrito pela função $F$ a seguir:

$$\begin{aligned} F^1(I^{LR}; W_1, b_1) &= \phi(W_1 \times I^{LR} + b_1) \\ F^i(I^{LR}; W_{1:i}, b_{1:i}) &= \phi(W_i \times F^{i-1}(I^{LR}) + b_i) \\ F^L(I^{LR}; W_{1:L}, b_{1:L}) &= PS(W_L \times F^i(I^{LR}) + b_L), \end{aligned} \tag{3.5}$$

onde $W_i$, $b_i$ com $i \in (1, L-1)$ são pesos e viés, respectivamente, que são os hiperparâmetros treináveis da rede. $W_i$ é um tensor de tamanho $n_{i-1} \times n_i \times k_i \times k_i$, onde $n_i$ é o número de filtros da camada de convolução $i$, $n_0 = C$, $C$ é o número de canais, $k_i$ é o tamanho do filtro da camada $i$. O tensor $b_i$ tem tamanho $n_i$, $\phi$ é a função de ativação e $PS$ é a função que realiza a operação de subpixel nos mapas de características de $I^{LR}$ a partir da camada $F^{L-1}$ e reescala em $I^{SR}$.

A função $PS$ reescala $I^{LR}$ aplicando convolução com passo fracionado $\frac{1}{r}$ no espaço de LR com filtro $W_s$ de tamanho $k_s$. Os pesos que ficam entre os pixels não são ativados, eliminando assim a necessidade de cálculo. O número de padrões de ativação é exatamente $r^2$. Cada padrão de ativação, de acordo com sua localização, tem no máximo $\left\lceil \frac{k_s}{r} \right\rceil^2$ pesos ativados. Esses padrões são ativados pelos filtros de convolução aplicados na imagem e dependem da localização do subpixel, $mod(x, r)$, $mod(y, r)$ onde $x, y$ são coordenadas do pixel de saída no espaço *high resolution* (HR).

A equação 3.6 define a operação realizada pela função $PS$, que é aplicável somente se $mod(k_s, r) = 0$.

$$PS(T)_{x,y,c} = T\left\lfloor x/r \right\rfloor, \left\lfloor y/r \right\rfloor, C.r.mod(x, r) + C.mod(y, r) + c. \tag{3.6}$$

$PS$ rearranja os elementos do tensor $H \times W \times C \times r^2$ para um tensor da forma $rH \times rW \times C$. A convolução $W_L$ da função 3.5 tem a forma $n_{L-1} \times r^2C \times k_L \times k_L$, com $k_L = \frac{k_s}{r}$, $mod(k_s, r) = 0$ e é equivalente a convolução por subpixel no espaço $LR$ com o filtro $W_s$.

A função de erro utilizada está definida na equação 3.7,

$$L(W_{1:L}, b_{1:L}) = \frac{1}{r^2WH} \sum_{x=1}^{rH} \sum_{y=1}^{rW} (I_{x,y}^{HR} - F_{x,y}^{L}(I_{x,y}^{LR}))^2 \tag{3.7}$$

Em Shi et al.[87] o método ESPCN foi implementado utilizando $L = 4$, $(k_1, n_1) = (5, 64)$, $(k_2, n_2) = (3, 32)$ e $(k_3, n_3) = (3, r^2 * C)$. Utilizou-se ainda a função de ativação ReLU nas camadas 1 e 2 e a *hyperbolic tangent* (Tanh) na última camada. Para o treino foram utilizadas fragmentos de imagem de dimensão $17r \times 17r$ para melhorar a performance do treinamento.

Demonstrou-se que apesar de cada filtro ser independente no espaço LR, os filtros são suavizados no espaço de HR após a Equação (3.6). Em comparação com os últimos filtros de camada do SRCNN, os filtros de camada final do ESPCN têm padrões complexos para diferentes mapas de características, além de representações mais ricas e significativas. O método ESPCN superou o método SRCNN em qualidade da imagem com a métrica quantitativa *peak signal-to-noise ratio* (PSNR) nos *benchmarks* testados, além disso, superou em tempo de execução, podendo restaurar imagens de vídeos em tempo real.

## 3.3 ABORDAGEM COM REDES NEURAIS RESIDUAIS

Redes neurais profundas são difíceis de treinar por causa do problema da dissipação ou explosão do gradiente [88, 89]. Esse problema ocorre quando os gradientes se tornam muito pequenos ou muito grandes à medida que são propagados pelas camadas da rede durante o processo de retropropagação. Para contornar esse problema, técnicas de normalização dos dados têm sido amplamente utilizadas [89, 90].

No entanto, à medida que as redes neurais profundas convergem, surge outro problema conhecido como degradação. Esse problema ocorre devido ao aumento da profundidade da rede, onde a precisão do modelo atinge um ponto de saturação e, em seguida, começa a se degradar rapidamente [91].

Essa degradação não é causada por *overfitting* (sobreajuste), pois adicionar mais camadas ao modelo profundo resulta em um maior erro de treinamento, conforme relatado por He e Sun[92] e Srivastava, Greff e Schmidhuber[93]. Esse fenômeno contradiz a intuição de que uma rede neural mais profunda deveria ser capaz de aprender uma representação mais rica e, portanto, obter um desempenho melhor.

Uma solução proposta para o problema de degradação é baseada no conceito de aprendizado residual. Essa abordagem foi introduzida por He et al.[91] e aplicada com sucesso em tarefas de reconhecimento de imagem. O bloco residual, ilustrado na Figura 3a, é uma construção fundamental nessa abordagem. Ele utiliza conexões de atalho para pular uma ou mais camadas convolucionais, permitindo que a rede aprenda a diferença entre as representações de entrada e de saída de cada bloco.

Essas conexões de atalho permitem que as informações originais sejam preservadas e facilitam o fluxo do gradiente durante o treinamento. Isso ajuda a mitigar o problema de degradação, permitindo que redes neurais mais profundas sejam treinadas com sucesso, alcançando melhor desempenho em várias tarefas de visão computacional.

A aplicação de aprendizado residual para tarefas de SR foi inicialmente explorada por Kim, Lee e Lee[94] e Kim, Lee e Lee[95], que aplicaram conexões residuais semelhantes às utilizadas em tarefas de reconhecimento de imagem [91]. Embora a super-resolução seja uma tarefa de visão computacional de baixo nível, essa abordagem mostrou-se eficaz na melhoria dos resultados.

Baseado nos trabalhos anteriores [91, 94, 95], Ledig et al.[21] propuseram o modelo *Super-resolution residual network* (SRResNet), que utiliza blocos residuais e conexões residuais. Os blocos residuais nesse modelo diferem da proposta original (Figura 3a) devido à substituição da função de ativação ReLU pela *parametric rectified linear unit* (PReLU) [77] e à remoção da função de ativação após a adição da conexão residual, conforme ilustrado na Figura 3b.

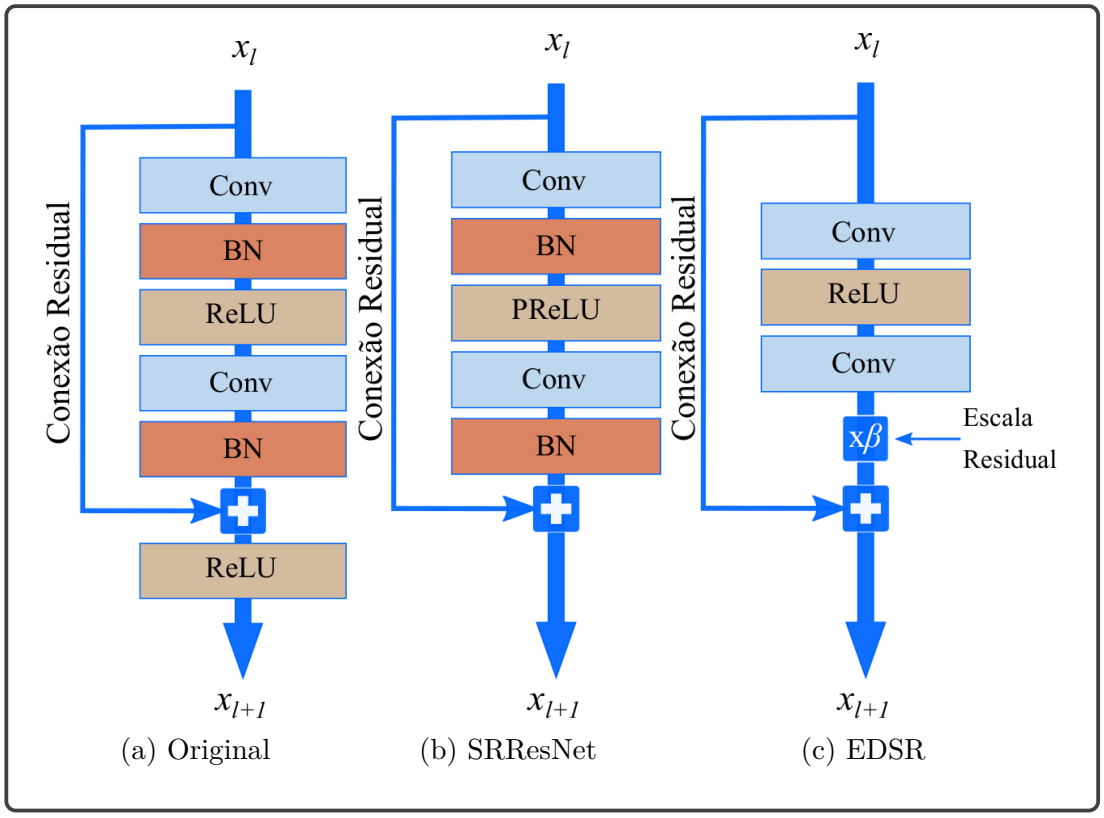
Posteriormente, com base na modificação do bloco residual proposta por Nah, Kim e Lee[96], Lim et al.[97] propuseram o modelo *Enhanced deep super-resolution network* (EDSR). Nesse modelo, os blocos residuais não utilizam a camada de normalização por lotes (*batch normalization* (BN)) [90], o que torna o modelo mais compacto. Além disso, o EDSR utiliza escala residual, ou seja, multiplica os pesos por uma constante $\beta$ entre zero e um, o que ajuda a estabilizar o processo de treinamento. Essa técnica de escala residual foi uma melhoria proposta anteriormente para tarefas de reconhecimento de imagem [98].

O aprimoramento introduzido no bloco residual do modelo EDSR em comparação com o SRResNet resultou em melhorias na performance do modelo. Especificamente, a redução do uso de memória da GPU permitiu adicionar mais blocos residuais à rede, tornando-a mais profunda e, consequentemente, aumentando sua acurácia.

O uso de redes convolucionais densas, conhecidas como *Dense convolutional network* (DenseNet) [99], foi introduzido em tarefas de SR por Tong et al.[100]. Eles propuseram o modelo *Super-resolution dense convolutional network* (SRDenseNet), que utiliza conexões residuais densas para explorar as características hierárquicas da imagem de baixa resolução. Outro modelo proposto é o *Residual dense network* (RDN) [101], que utiliza blocos residuais densos (RDB) para extrair características locais por meio de camadas de convolução densamente conectadas. O RDN é considerado um representante exemplar das redes residuais, pois incorpora os conceitos fundamentais que compõem o estado da arte dessa arquitetura.

O modelo RDN, conforme ilustrado na Figura 4, é composto por quatro partes principais: a rede rasa de extração de características (SFENet, do inglês *shallow feature*

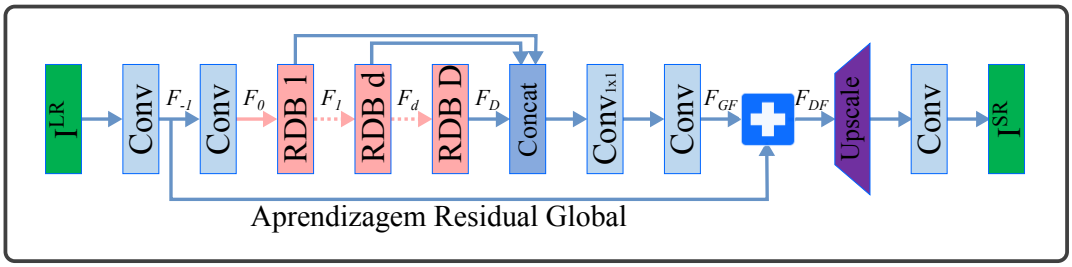
Figura 3 – Comparação de blocos residuais ResNet original, SRResNet e EDSR.




FONTE: Adaptado de Lim et al.[97].

*extraction net*), os blocos residuais densos (RDB), a fusão densa de características (*dense feature fusion* (DFF)) e a rede de aumento de resolução (DFF, do inglês *dense feature fusion*) e *up-sampling net* (UPNet).

Figura 4 – Arquitetura da rede residual densa (RDN).




FONTE: Adaptado de Zhang et al.[101].

1. **SFENet:** é composta por duas camadas de convolução que antecedem os RDBs. A primeira camada de convolução, representada na Equação (3.8), extrai características $F_{-1}$ a partir de $I^{LR}$,

$$F_{-1} = H_{SFE1}(I^{LR}), \tag{3.8}$$

onde $H_{SFE1}(.)$ denota operação de convolução. $F_{-1}$ possui uma conexão residual para DFF, nomeada aprendizagem residual global. A segunda camada de convolução (Equação (3.9)) recebe $F_{-1}$,

$$F_0 = H_{SFE2}(F_{-1}), \tag{3.9}$$

aqui $H_{SFE2}(.)$ representa a operação de convolução que antecede o primeiro bloco residual.

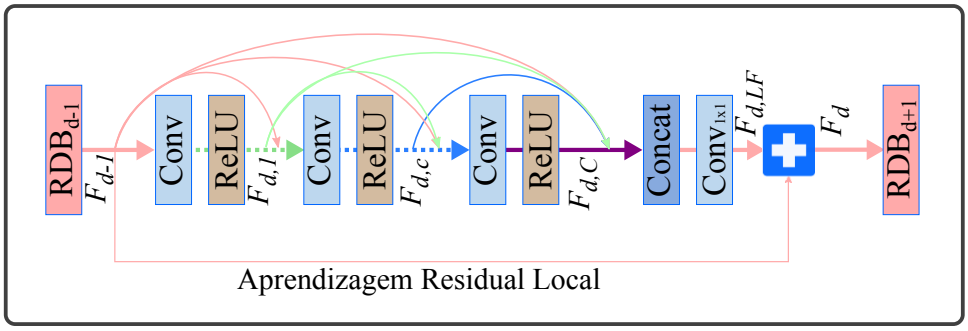
2. **RDB:** suponha uma rede com $D$ blocos residuais densos, a saída $F_d$ do $d$-ésimo RDB pode ser obtida por,

$$\begin{aligned} F_d &= H_{RDB,d}(F_{d-1}) \\ &= H_{RDB,d}(H_{RDB,d-1}(\dots(H_{RDB,1}(F_0))\dots)), \end{aligned} \tag{3.10}$$

onde $H_{RDB,d}$ denota a operação do $d$-ésimo RDB.

A arquitetura de um RDB está ilustrada na Figura 5. Esse bloco possui camadas densamente conectadas, fusão de características locais – *local feature fusion* (LFF)) – e utiliza aprendizagem residual local. Além disso, o RDB lida com um mecanismo de memória contígua.

Figura 5 – Arquitetura do bloco residual denso (RDB).




FONTE: Adaptado de Zhang et al.[101].

O **mecanismo de memória contígua** é realizado passando o estado do RDB anterior a cada camada do RDB atual. Seja $F_{d-1}$ e $F_d$ a entrada e a saída do $d$-ésimo RDB, respectivamente, ambos com $n_0$ mapas de características. A saída da $c$-ésima camada de convolução do $d$-ésimo RDB pode ser formulada como,

$$F_{d,c} = \sigma(W_{d,c}\,[F_{d-1}, F_{d,1}, \dots, F_{d,c-1}] + b_{d,c}\,[F_{d-1}, F_{d,1}, \dots, F_{d,c-1}]), \tag{3.11}$$

onde $\sigma$ representa a função de ativação ReLU. $W_{d,c}$ e $b_{d,c}$ são os tensores de pesos e viés da $c$-ésima camada de convolução, respectivamente. Assumindo que $F_{d,c}$ consiste de $n$ mapas de características; $[F_{d-1}, F_{d,1}, \dots, F_{d,c-1}]$ representa a concatenação dos mapas de características do $(d-1)$-ésimo RDB, camada convolucional $1, \dots, (c-1)$

no $d$-ésimo RDB, resultando em $n_0 + (c-1) \times n$ mapas de características. A saídas dos RDBs precedentes tem conexões residuais para todos os RDBs subsequentes.

A **fusão de características locais** é realizada pela camada de convolução com filtro $n = 1$ que controla o tamanho da saída da informação. Esta operação pode ser formulada como,

$$F_{d,LF} = H^d_{LFF}([F_{d-1}, F_{d,1}, \ldots, F_{d,c}, F_{d,C}]), \tag{3.12}$$

onde $H^d_{LFF}$ representa a função de camada de convolução $1 \times 1$ do $d$-ésimo RDB.

A **aprendizagem residual local** é introduzida no RDB para melhorar o fluxo de informação, uma vez que há várias camadas de convolução em um RDB. A saída do $d$-ésimo RDB pode ser obtida por,

$$F_d = F_{d-1} + F_{d,LF}. \tag{3.13}$$

3. **DFF:** explora características hierárquica de maneira global e é composta de fusão de características globais e aprendizagem residual global.

   A **fusão de características globais** extrai as características globais $F_{GF}$ usando características de todos os RDBs. Pode ser formulada como,

   $$F_{GF} = F_{GFF}([F_1, \ldots, F_D]), \tag{3.14}$$

   onde $[F_1, \ldots, F_D]$ representa a concatenação de mapas de características produzido pelos blocos residuais densos $1, \ldots, D$. $H_{GFF}$ é a função composta de convolução $1 \times 1$ e $3 \times 3$. A camada convolucional $1 \times 1$ é usada para fundir de forma adaptativa uma série de características com diferentes níveis. A camada convolucional $3 \times 3$ seguinte é introduzida para extrair características adicionais a aprendizagem residual global.

   A **aprendizagem residual global** é utilizada para obter os mapas de características antes de reescalar,

   $$F_{DF} = F_{-1} + F_{GF}, \tag{3.15}$$

   aqui $F_{-1}$ denota os mapas de características de baixo nível. Todas as outras camadas antes da fusão de característica global são totalmente utilizadas com os RDBs. Os RDBs produzem características densas locais multinível que são adaptadas para formar o $F_{GF}$. Após o aprendizado residual global, se obtém as características densas $F_{DF}$.

4. **UPNet:** a camada de *up-sampling* é utilizada para reescalar a imagem, nela, é utiliza camada de subpixel que foi descrita na seção 3.2.

O modelo RDN pode ser implementado com parâmetros variáveis. A configuração que apresentou melhor performance em experimentos realizados pelos autores encontra-se a seguir: $D = 20$ representa o número de RDBs, $C = 6$ representa o número de camadas de convolução por RDB e $n = 32$ é o número de filtros. Todas as camadas de convolução têm $k = 3 \times 3$, exceto nas camadas de fusão local e global nas quais $k = 1 \times 1$. As camadas da *shallow feature extraction net* (SFENet) e da fusão de características local e global têm $n_0 = 64$ filtros. As outras camadas em cada um dos RDBs possuem $n = 32$ filtros, seguidas da função de ativação ReLU. A última camada tem $n = 3$ ou $n = 1$ filtros que correspondem aos canais de cores da imagem.

O otimizador utilizado no modelo foi o Adam [102], com uma taxa de aprendizagem inicial de $10^{-4}$ e um decrescimento de $5 \times 10^{-1}$ a cada 200 épocas. A função de erro utilizada foi a $L_1$, conforme definido na Equação (3.16).

$$L_1(W_{1:L}, b_{1:L}) = \frac{1}{r^2WH} \sum_{x=1}^{rH} \sum_{y=1}^{rW} \left| I_{x,y}^{HR} - F_{x,y}^{L}(I_{x,y}^{LR}) \right| \tag{3.16}$$

O modelo foi treinado utilizando mini-lotes de tamanho 16 com imagens no espaço de cores *red, green, and blue* (RGB). Fragmentos de imagem LR de tamanho $32 \times 32$ pixels foram extraídos das imagens. Foi aplicado um aumento de dados randômico (*data augmentation*) nos segmentos, incluindo giro horizontal, giro vertical e rotação de 90°.

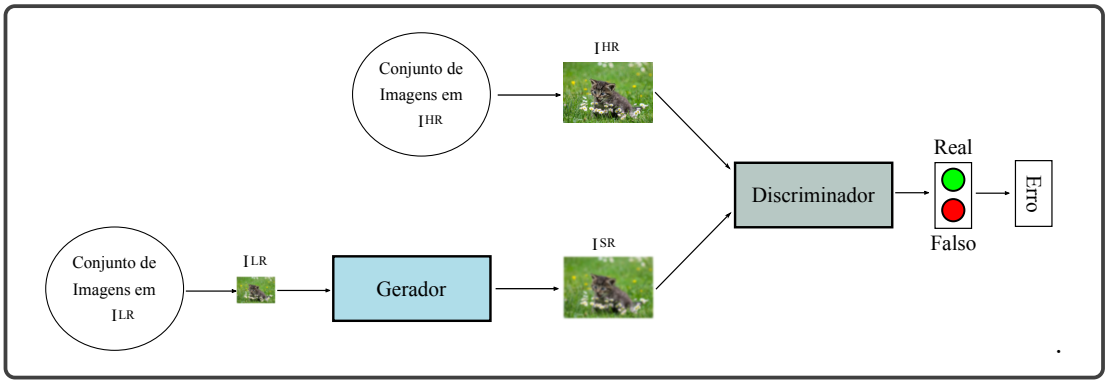
## 3.4 ABORDAGEM COM REDES ADVERSÁRIAS GENERATIVAS E FUNÇÕES DE ERRO PERCEPTIVA

Nos modelos anteriores, foram utilizadas funções de erro que operam no espaço dos pixels, como a $L_1$, *Mean absolute error* (MAE), e a $L_2$, *Mean squared error* (MSE). Com essas funções, os modelos buscam minimizar o erro com base na média dos pixels, o que acaba não sendo capaz de recuperar detalhes de alta frequência perdidos, como texturas, resultando em imagens suavizadas e de baixa qualidade perceptiva [103, 104, 105, 106].

Para contornar esse problema, Ledig et al.[21] propuseram o uso de uma função de erro perceptual para treinar redes adversárias generativas (GANs), que produzem imagens mais realistas.

A arquitetura das GANs consiste em uma rede geradora (G) e uma rede discriminadora (D) competindo entre si. A rede geradora aprende a mapear imagens de baixa resolução $I^{LR}$ para imagens de alta resolução $I^{SR}$ que se assemelham a imagens reais $I^{HR}$, tentando enganar a rede discriminadora. Por outro lado, a rede discriminadora aprende a distinguir entre imagens restauradas pela rede geradora e imagens reais. Essa competição entre as redes busca alcançar um equilíbrio, onde nenhuma das redes prevalece completamente. A Figura 6 ilustra a arquitetura conceitual das redes adversárias

Figura 6 – Arquitetura conceitual de uma rede adversária generativa (GAN).




Fonte: De autoria própria.

generativas.

$$\min_{\theta_G} \max_{\theta_D} V(D_{\theta_D}, G_{\theta_G}) = \mathbb{E}_{I^{HR} \sim p_{train}(I^{HR})} \left[ \log D_{\theta_D}(I^{HR}) \right] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})} \left[ \log \left( 1 - D_{\theta_D}(G_{\theta_G}(I^{LR})) \right) \right] \tag{3.17}$$

O criador das GANs, Goodfellow et al.[107], define-as como um problema de minimax, como mostrado na Equação (3.17) adaptada, onde o discriminador é treinado para maximizar a probabilidade de atribuir a classificação correta tanto para imagens reais como para imagens restauradas. Por outro lado, o gerador aprende a gerar imagens cada vez mais realistas ajustando seus parâmetros $\theta_G$ para minimizar o termo $\log \left( 1 - D_{\theta_D}(G_{\theta_G}(I^{LR})) \right)$. Aqui, $\theta_G$ representa os parâmetros do gerador, incluindo pesos ($W$) e bias ($b$), que são otimizados por meio de uma função de erro durante o treinamento.

### 3.4.1 ARQUITETURA DO MODELO SRGAN

O modelo *Super-resolution generative adversarial networks* (SRGAN) proposto por Ledig et al.[21] utiliza uma abordagem baseada em GAN em conjunto com uma função de erro perceptual para melhorar a qualidade das imagens restauradas. A parte geradora do SRGAN, denotada como G, quando treinada com a função de erro MSE, resulta no modelo conhecido como SRResNet, mencionado na seção seção 3.3.

A arquitetura da parte geradora do SRGAN é composta por cinco principais partes, como ilustrado na Figura 7. Essas partes incluem:

1. Parte pré-residual: formada por uma convolução de 64 filtros de tamanho nove e *stride* um (k9n64s1), seguida da função de ativação PReLU.

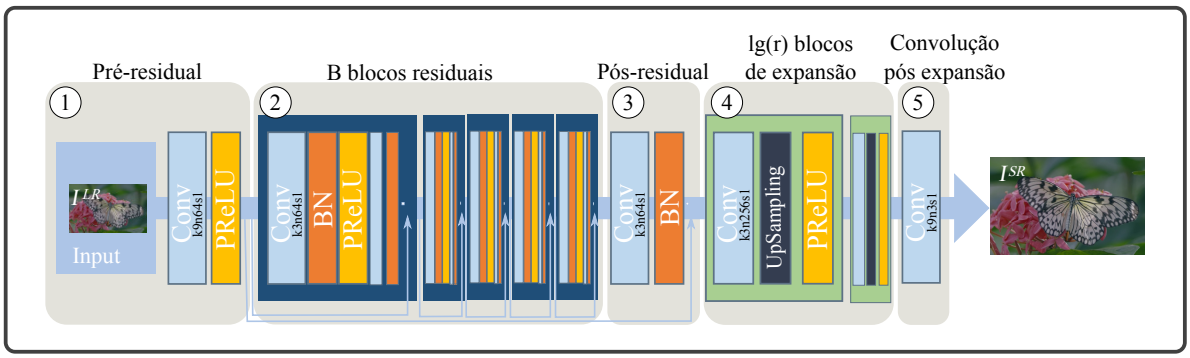
2. Blocos residuais: possui $B$ blocos residuais idênticos; a arquitetura dos blocos residuais encontra-se na Figura 3b.

3. Parte pós-residual: formada por uma camada de convolução (k3n64s1) seguida de uma camada de BN e conexão residual provinda da parte pré-residual.

4. Blocos de expansão: esta é a parte em que a imagem é efetivamente expandida, ou seja, ganha maior resolução a partir dos $n$ mapas de características extraídos pelos blocos residuais. Cada bloco é composto por uma camada de convolução (k3n256s1), seguida por uma camada de subpixel e pela função de ativação PReLU. Cada bloco de expansão aumenta a escala da imagem em duas vezes. Portanto, se o fator de escala for $r \in \{2, 4, 8, 16\}$, serão necessários $\log(r)$ blocos de expansão no modelo.

5. Convolução pós expansão: ao final dos blocos de expansão há uma camada de convolução (k9n3s1) que define a imagem final com três canais de cores.

Figura 7 – Arquitetura de rede geradora do modelo SRGAN.



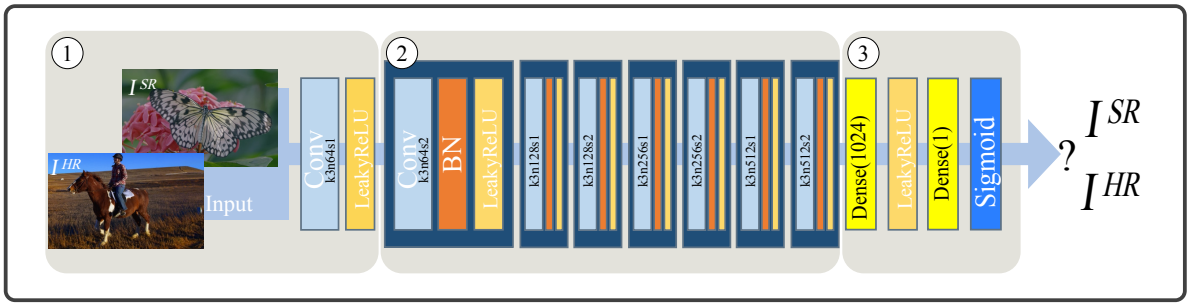

FONTE: Adaptado de Ledig et al.[21].

A parte discriminadora $D$ do modelo SRGAN tem a função de diferenciar imagens reais e sintetizadas pelo gerador. O discriminador é utilizado apenas na fase de treinamento do modelo SRGAN com abordagem adversária, o que ajuda a aprimorar o gerador a sintetizar imagens mais realistas.

A parte discriminadora D do modelo SRGAN tem a função de distinguir entre imagens reais e imagens restauradas pelo gerador G. O discriminador é utilizado apenas durante a fase de treinamento do modelo SRGAN com a abordagem adversarial. Essa abordagem permite que o gerador aprenda a sintetizar imagens mais realistas, pois o discriminador fornece um *feedback* indicando o quão bem as imagens geradas se aproximam das reais.

Conforme ilustrado na Figura 8, a primeira parte do discriminador é composta por uma camada de convolução com um núcleo de tamanho 3 e 64 filtros, seguida de

Figura 8 – Arquitetura de rede discriminadora do modelo SRGAN.




FONTE: Adaptado de Ledig et al.[21].

uma ativação *leaky rectified linear unit* (LeakyReLU) [108] com um fator de inclinação de $\alpha = 2 \times 10^{-1}$. Essa camada possui um *stride* de 1.

A segunda parte do discriminador, que é o núcleo do modelo, consiste em sete blocos, cada um composto por uma camada de convolução, uma camada de BN e uma ativação LeakyReLU com $\alpha = 2 \times 10^{-1}$. A primeira camada de convolução desse núcleo possui 64 filtros de tamanho 3 e um *stride* de 2. Nas demais camadas, o número de filtros é dobrado a cada dois blocos, começando com 64 filtros e chegando a 512 filtros. O *stride* varia entre 1 e 2 para cada par de blocos.

A parte final do discriminador é composta por duas camadas densas. A primeira camada possui 1024 neurônios e é seguida por uma ativação LeakyReLU com $\alpha = 2 \times 10^{-1}$. A segunda camada é uma camada densa com apenas um neurônio. Por fim, a função de ativação sigmoide é aplicada para obter a probabilidade de classificação da imagem como falsa ou verdadeira.

Essa arquitetura do discriminador é utilizada no modelo SRGAN para distinguir entre imagens reais e imagens restauradas pelo gerador, contribuindo para o treinamento adversarial e o aprimoramento da qualidade das imagens super-resolvidas.

#### 3.4.1.1 FUNÇÃO DE ERRO PERCEPTIVA

As funções clássicas de erro baseadas em média, como a $L_1$ e $L_2$ (referenciadas em [109]), têm a tendência de produzir imagens com bons resultados quando avaliadas por métricas "*pixel-wise*" como a PSNR e a similaridade estrutural (SSIM, do inglês *structural similarity*), mas podem resultar em imagens suavizadas que são facilmente perceptíveis pela visão humana. Esse problema tem motivado pesquisas em busca de funções de erro que levem a resultados perceptivos melhores [103, 106, 104, 21, 22, 46, 45].

Essas abordagens visam superar as limitações das funções de erro baseadas em média, buscando capturar melhor as características perceptivas das imagens, como texturas e detalhes de alta frequência. Para isso, são utilizadas funções de erro que levam em

consideração medidas perceptivas, como a diferença de conteúdo e a diferença adversarial, que são avaliadas com base na comparação entre as imagens geradas e as imagens reais de alta qualidade.

Essas técnicas têm demonstrado resultados promissores, gerando imagens super-resolvidas de alta qualidade e com maior fidelidade perceptiva em relação aos métodos tradicionais.

Para o modelo SRGAN os autores Ledig et al.[21] propuseram uma função de erro $l^{SR}$ que melhorou a avaliação perceptiva. Esta função foi baseada nos trabalhos de Gatys, Ecker e Bethge[110], Johnson, Alahi e Li[104] e Bruna, Sprechmann e LeCun[106]. A função de erro $l^{SR}$ é composta pela soma de uma função de erro de conteúdo $l_{VGG_{i,j}}^{SR}$ e uma função de erro adversário $l_{Gen}^{SR}$, conforme definida na Equação (5.4),

$$l^{SR} = \lambda \times l_{VGG_{i,j}}^{SR} + \eta \times l_{Gen}^{SR} \tag{3.18}$$

onde $\lambda$ e $\eta$ são constantes utilizadas para balancear os pesos entre as duas funções e foram definidas com os valores $6 \times 10^{-3}$ e $10^{-3}$, respectivamente.

A função de erro de conteúdo $l_{VGG_{i,j}}^{SR}$ é calculada pela MSE aplicada em mapas de características extraídas das imagens $I^{SR}$ e sua correspondente $I^{HR}$. A extração de características é realizada pela rede neural pré-treinada $VGG19$ [53] na $j$-ésima convolução após a ativação e antes da $i$-ésima camada de *maxpooling*. Na Equação (3.19) encontra-se definida a função de erro de conteúdo,

$$\begin{aligned} l_{VGG_{i,j}}^{SR} = \frac{1}{W.H} \sum_{x=1}^{W} \sum_{y=1}^{H} \\ \left( VGG_{i,j}(I^{HR})_{x,y} - VGG_{i,j}\left(I^{SR}\right)_{x,y} \right)^2, \end{aligned} \tag{3.19}$$

onde $W$ e $H$ são as dimensões dos mapas de características $VGG_{i,j}$.

Na Equação (6.4) encontra-se a função do erro adversário $l_{Gen}^{SR}$ que é calculado baseado no erro de entropia cruzada e é adicionado ao erro de conteúdo conforme apresentado na equação 5.4. Essa função avalia o erro da probabilidade de $D(.)$ identificar se uma imagem sintetizada é real ou não.

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G(I^{LR})) \tag{3.20}$$

#### 3.4.1.2 DETALHES DA CONFIGURAÇÃO DO TREINAMENTO

Originalmente, o SRGAN foi treinado utilizando uma amostra aleatória de 350 mil imagens da base de dados ImageNet [111]. As imagens de baixa resolução (LR) foram obtidas reduzindo a escala das imagens originais em um fator de 4 usando interpolação

bicúbica. Os mini-lotes de treinamento tinham tamanho 16 e foram criados cortando as imagens com segmentos de tamanho $96 \times 96$ pixels.

Os valores dos pixels das imagens LR foram normalizados para o intervalo [0, 1], enquanto as imagens de alta resolução (HR) foram normalizadas para o intervalo [-1, 1]. O otimizador utilizado foi o Adam [102], com um valor de $\beta_1$ de $9 \times 10^{-1}$. A rede geradora foi projetada com 16 blocos residuais idênticos (B=16). Durante os testes, a normalização por lotes (BN) foi desativada para que cada imagem de saída dependesse deterministicamente apenas da imagem de entrada.

O treinamento do modelo foi dividido em duas fases. Na primeira fase, apenas a parte geradora foi treinada, resultando no modelo chamado SRResNet. Nessa fase, uma taxa de aprendizagem de $10^{-4}$ foi utilizada, juntamente com a função de erro MSE e um total de $10^6$ iterações de treinamento.

Na segunda fase, o modelo foi treinado usando uma abordagem adversarial, envolvendo tanto o gerador quanto o discriminador. O gerador foi inicializado com os pesos obtidos na primeira fase de treinamento, o que permitiu que o gerador começasse a gerar imagens que fizessem sentido, ou seja, imagens que se assemelhassem às imagens reais, evitando que a GAN ficasse presa em mínimos locais indesejados. Nessa segunda fase, o modelo adversarial foi treinado utilizando a função de erro perceptivo definida na Equação (5.4) por $10^5$ iterações, com uma taxa de aprendizagem de $10^{-4}$ . Em seguida, foram realizadas mais $10^5$ iterações com uma taxa de aprendizagem de $10^{-5}$.

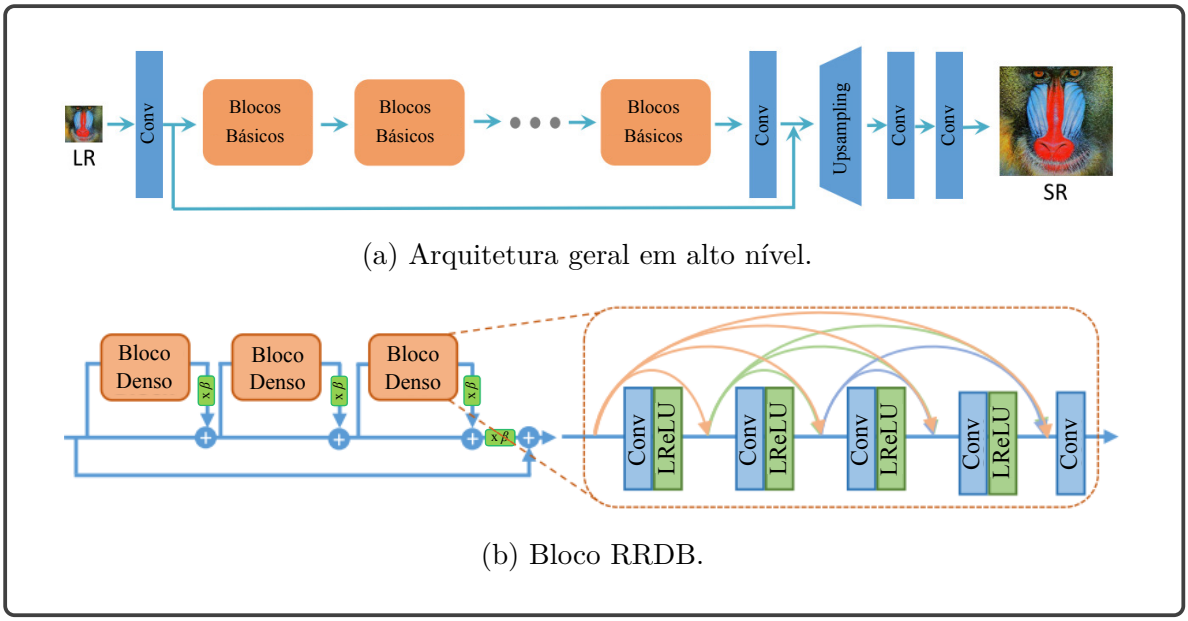
### 3.4.2 SRGAN APRIMORADA

Wang et al.[22] propuseram um modelo aprimorado do SRGAN, chamado *Enhanced super-resolution generative adversarial networks* (ESRGAN). A principal modificação ocorreu no núcleo da rede, mais especificamente nos blocos residuais utilizados na parte geradora.

Primeiramente, uma modificação foi introduzida na estrutura dos blocos residuais, tornando-os mais capazes e fáceis de treinar. Esses novos blocos foram denominados de *residual-in-residual dense block* (RRDB). Neles, as camadas de BN foram removidas, seguindo a abordagem proposta por Lim et al.[97]. Além disso, foi introduzida a escala residual, inspirada em trabalhos anteriores, como Lim et al.[97] e Szegedy, Ioffe e Vanhoucke[98]. Também foi aplicada uma inicialização menor para os pesos aleatórios, facilitando o treinamento de uma rede profunda.

Em segundo lugar, o discriminador foi aprimorado, com base no trabalho de Jolicoeur-Martineau[112], que propõe o uso da GAN com média relativística (*relativistic average GAN* (RaGAN)). A RaGAN aprende a julgar se uma imagem é mais realista do que a outra, em vez de simplesmente distinguir entre imagens reais e falsas, como nas

Figura 9 – Arquitetura do gerador da ESRGAN.




(a) Arquitetura geral em alto nível.

(b) Bloco RRDB.

FONTE: Adaptado de Wang et al.[22].

GANs tradicionais.

Por fim, a terceira alteração ocorreu na função de erro perceptiva. Entre as mudanças, utilizaram-se características extraídas pela rede VGG-19 antes da ativação, em vez de após a ativação, como no SRGAN. Essa nova função de erro aprimorou o modelo, resultando em imagens mais nítidas e com resultados visualmente mais agradáveis.

#### 3.4.2.1 ARQUITETURA DA ESRGAN

Na Figura 9a encontra-se ilustrada a arquitetura geral do modelo ESRGAN. Os blocos básicos são compostos por blocos RRDB que estão detalhados na Figura 9b. Observou-se em modelos anteriores que aumentar a profundidade da rede e utilizar conexões residuais aumentam o desempenho [101, 97, 113]. Assim, o RRDB emprega uma estrutura mais profunda e complexa do que o bloco residual original no SRGAN. Especificamente, o RRDB possui uma estrutura em que dentro de um bloco básico, há blocos residuais densos, onde o aprendizado residual é usado em diferentes níveis. Essa ideia foi inspirada em trabalhos anteriores, como o de Zhang et al.[114], que também aplicaram uma rede residual em vários níveis. O diferencial do RRDB é o uso de blocos densos, semelhante ao bloco do modelo RDN apresentado anteriormente na seção 3.3.

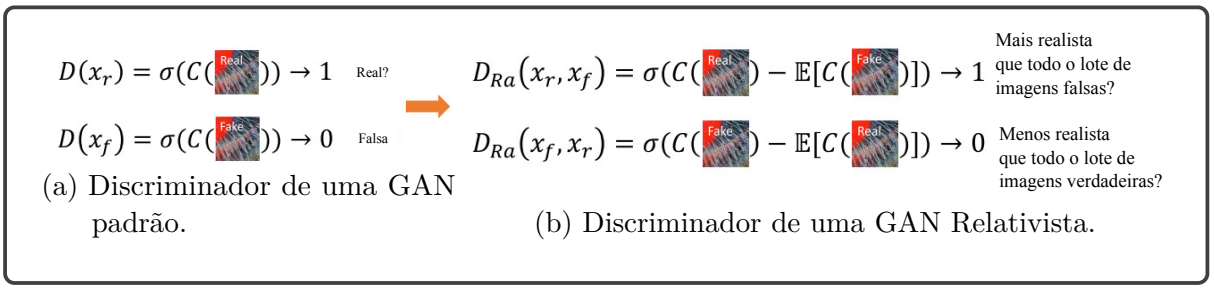
Além da arquitetura aprimorada, também foram exploradas duas técnicas para facilitar o treinamento de uma rede muito profunda: (*i*) o uso de escala residual [98, 97], ou seja, diminuindo os resíduos multiplicando uma constante entre zero e um antes de adicioná-los à rede principal, para prevenir a instabilidade; (*ii*) inicialização menor dos

pesos, pois empiricamente observou-se que a arquitetura residual se torna mais fácil de treinar quando a variação inicial dos parâmetros é menor.

#### 3.4.2.2 O DISCRIMINADOR RELATIVISTA

No modelo SRGAN, o discriminador padrão $D$ estima a probabilidade de uma imagem ser real $x_r$ ou falsa $x_f$ (Figura 10a), por outro lado, um discriminador relativista $D_{Ra}$ tenta prever a probabilidade de uma imagem real $x_r$ ser, relativamente, mais realista que uma falsa $x_f$, como ilustrado na Figura 10b.

Figura 10 – Diferença entre um discriminador padrão e um discriminador relativista.




(a) Discriminador de uma GAN padrão.

(b) Discriminador de uma GAN Relativista.

FONTE: Adaptado de Wang et al.[22].

No modelo ESRGAN, o discriminador padrão $D$ foi substituído pelo discriminador relativista $D_{Ra}$ proposto em Jolicoeur-Martineau[112]. No modelo SRGAN, o discriminador padrão era expresso por $D(x) = \sigma(C(x))$, onde $\sigma$ representa a função sigmoide e $C(x)$ é a saída do discriminador antes da função sigmoide, já no modelo ESRGAN o discriminador relativista é formulado como $D_{Ra}(x_r, x_f) = \sigma(C(x_r) - \mathbb{E}_{x_f}[C(x_f)]$, onde $\mathbb{E}_{x_f}[.]$ representa a operação de calcular a média de todos as imagens falsas no mini-lote. Assim, a função de erro do discriminador é definida pela Equação (3.21),

$$L_D^{Ra} = -\mathbb{E}_{x_f}[\log(D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(1 - D_{Ra}(x_r, x_f))]. \tag{3.21}$$

A função de erro adversário para o gerador é de forma simétrica, definida na Equação (3.22),

$$L_G^{Ra} = -\mathbb{E}_{x_r}[\log(1 - D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(D_{Ra}(x_r, x_f))], \tag{3.22}$$

onde $x_f = G(x_i)$ e $x_i$ representa a imagem LR de entrada. Observa-se que o erro adversário para o gerador contém $x_r$ e $x_f$. Portanto, o gerador se beneficia dos gradientes de imagens geradas e reais no treinamento adversário, enquanto no SRGAN, apenas imagens geradas eram utilizadas. O discriminador relativista ajudou a rede a aprender bordas mais nítidas e texturas mais detalhadas.

#### 3.4.2.3 FUNÇÃO DE ERRO PERCEPTIVA

A função de erro perceptiva apresentada na seção 3.4.1.1 utiliza características extraídas pela rede neural $VGG19$ [53] na $j$-ésima convolução após a ativação e antes da $i$-ésima camada de *maxpooling*, onde a distância entre duas características ativadas é minimizada.

Diferentemente da abordagem utilizada no modelo SRGAN, no modelo ESRGAN foram utilizadas características antes da camada de ativação, o que superou duas desvantagens da versão anterior. Primeiro, as características após ativação são muito esparsas, especialmente em uma rede muito profunda. A ativação esparsa fornece supervisão fraca e, portanto, leva a um desempenho inferior. Segundo, o uso de características após a ativação também gera um brilho reconstruído inconsistente em comparação com a imagem real. Assim, a nova função de erro perceptiva foi definida, conforme Equação (3.23),

$$L_G = L_{percep} + \lambda L_G^{Ra} + \eta L_1, \tag{3.23}$$

onde $L_1 = \mathbb{E}_I \left\| G(I^{LR}) - I^{HR} \right\|_1$ é o erro de conteúdo mensurado pela distância absoluta a partir da imagem restaurada $G(I^{LR})$ e a imagem em alta resolução $I^{HR}$; $L_G^{Ra}$ está definido na Equação (3.22); $\lambda$ e $\eta$ são coeficientes utilizados para balancear os diferentes termos da função; e,

$$L_{percep} = \frac{1}{W.H} \sum_{x=1}^{W} \sum_{y=1}^{H} \left( VGG_{i,j}(I^{HR})_{x,y} - VGG_{i,j} \left( I^{SR} \right)_{x,y} \right)^2, \tag{3.24}$$

sendo $W$ e $H$ as dimensões dos mapas de características $VGG_{i,j}$.

#### 3.4.2.4 DETALHES DA CONFIGURAÇÃO DO TREINAMENTO

Para treinar o modelo ESRGAN, foram definidos 23 blocos RRDB. As imagens LR foram obtidas a partir das imagens HR, reduzindo a escala em 4× por meio de interpolação bicúbica utilizando a ferramenta MATLAB. Mini-lotes de tamanho 16 foram utilizados, com as imagens HR sendo cortadas em pequenos segmentos de imagem de tamanho 128x128 pixels.

Observou-se que quanto maiores os segmentos de imagem HR, maior o benefício para o modelo, pois fragmentos maiores ajudam a capturar mais informações semânticas. No entanto, o tamanho maior dos segmentos também resulta em um tempo de treinamento mais longo e no consumo de mais recursos computacionais.

O treinamento foi realizado em duas etapas: ($i$) Na primeira etapa, apenas a parte geradora do modelo foi treinada utilizando a função de erro $L_1$, o otimizador Adam com os parâmetros $\beta_1 = 9 \times 10^{-1}$, $\beta_2 = 9.99 \times 10^{-1}$ e taxa de aprendizagem inicial de $2 \times 10^{-4}$ com decaimento de 2 a cada $2 \times 10^5$ iterações. ($ii$) Na segunda etapa, o treinamento adversarial foi realizado, com o gerador sendo iniciado com os pesos do treinamento da primeira etapa.

A abordagem adversária utilizou a função de erro da Equação (3.23) com os valores de $\lambda$ e $\eta$ definidos como $5 \times 10^{-3}$ e $10^{-2}$, respectivamente. A taxa de aprendizagem foi definida como $10^{-4}$, com decaimento pela metade nas iterações $[5 \times 10^4, 1 \times 10^5, 2 \times 10^5$ e $3 \times 10^5]$.

O pré-treinamento realizado na primeira fase ajudou a GAN a obter resultados visualmente mais nítidos. Isso ocorreu porque iniciar o treinamento adversarial com pesos pré-treinados evita que o gerador fique preso em mínimos locais. Além disso, as imagens geradas durante o pré-treinamento são mais próximas das reais, em vez de serem completamente pretas ou cheias de ruído, o que ajuda o discriminador a se concentrar mais nas texturas.

## 3.5 COMPARATIVO DE DESEMPENHO DOS MODELOS

Nesta seção, apresenta-se o desempenho dos modelos de *sigle image super-resolution* (SISR) abordados neste capítulo nas principais bases de dados de referência para imagens. Inicialmente, encontra-se uma breve descrição dessas bases de dados e, em seguida, apresenta-se os valores comparativos de desempenho utilizando as métricas *pixel-wise* PSNR e SSIM. Esses valores foram coletados por meio de pesquisa na literatura.

### 3.5.1 CONJUNTOS DE DADOS PARA SUPER-RESOLUÇÃO DE ÚNICA IMAGEM

Diversas bases de dados têm sido utilizadas para tarefas de SISR, algumas delas possuem conjuntos de treino, validação e teste, enquanto outras possuem apenas conjunto de teste. A seguir, apresenta-se uma breve descrição das principais bases de dados utilizadas nos trabalhos que abordam tarefas de SR.

1. **Set5**[4] **[84]:** é um conjunto de dados composto por cinco imagens de diferentes resoluções, sendo a imagem com maior resolução de $512 \times 512$ pixels. O conjunto Set5 tem sido amplamente utilizado como *benchmark* para modelos de SISR.

2. **Set14**[4] **[83]:** é um conjunto de dados composto por quatorze imagens de diferentes resoluções, sendo a imagem com maior resolução de $720 \times 576$ pixels. O conjunto Set14 também tem sido amplamente utilizado como *benchmark* para modelos de SISR.

3. **BSDS100**[5]**:** é o subconjunto de teste da base de dados BSDS500 [115, 116]. Originalmente esta base foi proposta para tarefas de segmentação de imagem, possuindo um total de 500 imagens com subconjuntos para treino, validação e teste.

[4] https://www.kaggle.com/ll01dm/set-5-14-super-resolution-dataset
[5] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html

4. **Manga109**[6] **[117]:** é composto por 109 imagens de mangás desenhados por artistas profissionais de mangá no Japão. O conjunto Mangá109 tem sido amplamente utilizado como base de teste em tarefas de SISR.

5. **DIV2K**[7] **[118]:** é um conjunto de dados com 1000 imagens de alta resolução (2K) com uma grande diversidade de conteúdos. Foi originalmente proposto para ser utilizada no NTIRE 2017[8] para o desafio na super-resolução de imagem e desde então tem sido utilizado principalmente para treinar modelos de SISR. A base DIV2K está dividida em conjunto de treino com 800 imagens, conjunto de validação com 100 imagens e conjunto de teste com 100 imagens.

6. **ImageNet**[9] **[111]:** é um conjunto de dados de imagens organizadas de acordo com a hierarquia léxica da WordNet [119]. Atualmente, em cada nó da hierarquia tem em média mais de quinhentas imagens.

### 3.5.2 DESEMPENHO DOS MODELOS

Nesta seção, apresenta-se o desempenho dos modelos de SISR abordados neste capítulo para as métricas PSNR e SSIM em três bases de *benchmark*: Set5, Set14 e BSDS100.

Na Tabela 2, é apresentada a avaliação do desempenho na base Set5. Para a escala de 4×, o modelo ESRGAN alcançou o melhor desempenho em ambas as métricas. Para as escalas de 2× e 3×, o ESRGAN não foi avaliado, e o melhor resultado foi obtido pelo modelo RDN.

Tabela 2 – Valores médios das métricas PSNR e SSIM para a base de dados Set5.

| **Escala** | **Métricas** | **Métodos de SISR** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Bicubic** | **SRCNN** | **ESPCN** | **SRResNet** | **SRGAN** | **EDSR** | **RDN** | **ESRGAN** |
| 2x | PSNR | 33,66 | 36,66 | - | - | - | 38,11 | **38,24** | - |
| | SSIM | 0,9299 | 0,9542 | - | - | - | 0,9601 | **0,9614** | - |
| 3x | PSNR | 30,39 | 32,75 | 33,13 | - | - | 34,65 | **34,71** | - |
| | SSIM | 0,8682 | 0,909 | - | - | - | 0,9282 | **0,9296** | - |
| 4x | PSNR | 28,42 | 30,49 | 30,90 | 32,05 | 29,4 | 32,46 | 32,47 | **32,73** |
| | SSIM | 0,8104 | 0,8628 | 0,8784 | 0,9019 | 0,8472 | 0,8968 | 0,899 | **0,9011** |

Fonte: De autoria própria a partir de dados dos autores Dong et al.; Shi et al.; Lim et al.; Zhang et al.; Ledig et al.; Wang et al.[79, 87, 97, 101, 21, 22]

Na Tabela 3, é apresentado o desempenho dos modelos na base Set14. Observa-se que o modelo ESRGAN também apresenta o melhor desempenho na escala de 4×, enquanto o modelo RDN possui o melhor desempenho nas escalas de 2× e 3×, uma vez que o ESRGAN não foi avaliado para essas escalas.

[6] http://www.manga109.org/en/
[7] https://data.vision.ee.ethz.ch/cvl/DIV2K/
[8] http://www.vision.ee.ethz.ch/ntire17/
[9] http://image-net.org/

Na Tabela 4, está o desempenho dos modelos avaliados na base BSDS100. Observa-se que o desempenho segue uma tendência semelhante às avaliações das duas bases anteriores, com exceção dos valores da métrica *structural similarity* (SSIM) na escala de 3×, em que houve um empate entre os modelos EDSR e RDN.

Nos três *benchmarks*, observa-se que o modelo SRGAN não alcança um bom desempenho em relação às métricas apresentadas nas tabelas, com valores próximos ao modelo de referência *Bicubic*. Isso ocorre porque o modelo SRGAN é treinado com uma função de erro perceptivo, conforme a Equação (5.4), que tem como objetivo otimizar a qualidade perceptiva em vez do desempenho pixel a pixel, como é avaliado nas Tabelas 2, 3 e 4. No entanto, a qualidade perceptiva pode ser avaliada por métricas específicas para esse propósito.

Tabela 3 – Valores médios das métricas PSNR e SSIM para a base de dados Set14.

| **Escala** | **Métricas** | **Métodos de SISR** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Bicubic** | **SRCNN** | **ESPCN** | **SRResNet** | **SRGAN** | **EDSR** | **RDN** | **ESRGAN** |
| 2x | PSNR | 30,23 | 32,45 | - | - | - | 33,92 | **34,01** | - |
| | SSIM | 0,8687 | 0,9067 | - | - | - | 0,9195 | **0,9212** | - |
| 3x | PSNR | 27,54 | 29,3 | 29,49 | - | - | 30,52 | **30,57** | - |
| | SSIM | 0,7736 | 0,8215 | - | - | - | 0,8462 | **0,8468** | - |
| 4x | PSNR | 26,00 | 27,5 | 27,73 | 28,49 | 26,02 | 28,8 | 28,81 | **28,99** |
| | SSIM | 0,7019 | 0,7513 | 0,8004 | 0,8184 | 0,7397 | 0,7876 | 0,7871 | **0,7917** |

Fonte: De autoria própria a partir de dados dos autores Dong et al.; Shi et al.; Lim et al.; Zhang et al.; Ledig et al.; Wang et al.[79, 87, 97, 101, 21, 22]

Tabela 4 – Valores médios das métricas PSNR e SSIM para a base de dados BSDS100.

| **Escala** | **Métricas** | **Métodos de SISR** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Bicubic** | **SRCNN** | **ESPCN** | **SRResNet** | **SRGAN** | **EDSR** | **RDN** | **ESRGAN** |
| 2x | PSNR | 29,56 | 31,36 | - | - | - | 32,32 | **32,34** | - |
| | SSIM | 0,8431 | 0,8879 | - | - | - | 0,9013 | **0,9017** | - |
| 3x | PSNR | 27,21 | 28,41 | - | - | - | 29,25 | **29,26** | - |
| | SSIM | 0,7385 | 0,7863 | - | - | - | **0,8093** | **0,8093** | - |
| 4x | PSNR | 25,96 | 26,68 | 27,02 | 27,58 | 25,16 | 27,71 | 27,72 | **27,85** |
| | SSIM | 0,6675 | 0,7291 | 0,7442 | 0,762 | 0,6688 | 0,742 | 0,7419 | **0,7455** |

Fonte: De autoria própria a partir de dados dos autores Dong et al.; Shi et al.; Lim et al.; Zhang et al.; Ledig et al.; Wang et al.[79, 87, 97, 101, 21, 22]

## 3.6 CONSIDERAÇÕES

Neste capítulo, foram apresentados os principais modelos que representam o estado da arte para tarefas de SISR. Esses modelos são baseados em *Deep Learning*, mais especificamente em Redes Neurais Convolutivas. Foram abordadas as principais técnicas e arquiteturas utilizadas, sendo destacado um modelo representativo para cada abordagem.

Além disso, foram apresentadas as principais bases de dados utilizadas para treinamento e avaliação dos modelos de SISR. Ao final do capítulo, foi fornecida uma análise de desempenho dos modelos mais relevantes, utilizando métricas como PSNR e SSIM, com

base em três conjuntos de dados de referência (*benchmarks*). Os resultados apresentados foram obtidos a partir das publicações dos respectivos autores.

Vale ressaltar que os modelos apresentados neste capítulo têm como objetivo melhorar o estado da arte do SISR, sem se preocupar com aplicações específicas. No entanto, este trabalho difere dessas abordagens, pois se concentra na aplicação de SR em tarefas de vídeo *streaming*, visando melhorar a Qualidade de Experiência e reduzir o tráfego de conteúdo de vídeo nas infraestruturas de rede.

# CAPÍTULO 4

# Super-Resolução Multiquadro Baseada em Redes Neurais Profundas

Os métodos de Super-Resolução de Imagem Única (SISR), nos quais apenas uma imagem de baixa resolução (LR) é fornecida como entrada, exploram a redundância dos dados nas imagens por meio de correlações espaciais para recuperar detalhes de alta frequência, como nitidez e textura.

Por outro lado, na super-resolução multiquadro ou super-resolução de vídeo (VSR), assume-se que diferentes observações da mesma cena estão disponíveis, ou seja, uma sequência de imagens. Essa redundância nas imagens de vídeo implica em uma dimensão adicional de dados, a dimensão temporal, que possui um alto grau de correlação e pode ser explorada para melhorar o desempenho em termos de precisão e eficiência [120].

O objetivo da VSR é estimar um quadro de alta qualidade $F_t^{SR}$, que seja semelhante ao quadro real de alta resolução $F_t^{HR}$, a partir dos quadros reais e seus vizinhos em baixa resolução $F_{t+i,i\in[-N:+N]}^{LR}$. Ao contrário do SISR, no qual as imagens restauradas $I^{SR}$ são obtidas a partir de uma única imagem de baixa resolução $I^{LR}$, explorando a informação espacial, na VSR um quadro $F^{SR}$ é restaurado a partir de vários quadros em baixa resolução, daí o termo "super-resolução multiquadro", com o objetivo de aproveitar informações temporais redundantes, ou seja, repetições da mesma cena nos quadros vizinhos ao quadro que se pretende restaurar.

Para obter alta qualidade nas restaurações, um desafio nos modelos de super-resolução multiquadro é o alinhamento dos quadros ou compensação de movimento, para então realizar a fusão e, posteriormente, a restauração [28]. Os modelos propostos aplicam

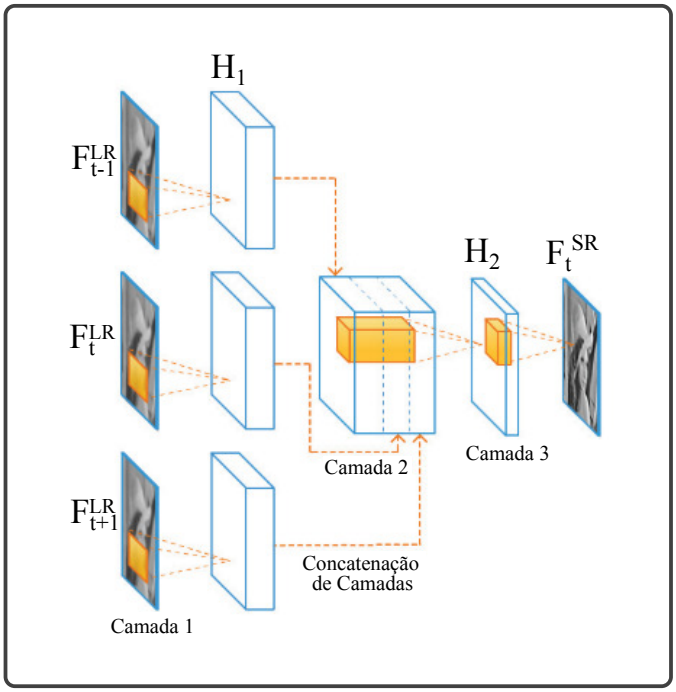
várias técnicas para resolver esse problema, alguns com técnicas de alinhamento explícito e outros com alinhamentos implícitos. Nas seções seguintes, serão apresentados os métodos mais recentes e principais de VSR que utilizam redes neurais profundas.

## 4.1 SUPER-RESOLUÇÃO DE VÍDEO COM REDES NEURAIS CONVOLUCIONAIS

O modelo *Video super-resolution network* (VSRnet), proposto por Kappeler et al.[18], é um modelo de super-resolução multiquadro baseado no trabalho anterior de Dong et al.[78]. Os autores adaptaram o modelo original de três camadas de convolução para restaurar vídeos utilizando a abordagem multiquadro.

Na Figura 11, é possível observar a arquitetura do VSRnet. O modelo recebe três quadros como entrada: $F_{t-1}^{LR}$ (quadro anterior), $F_t^{LR}$ (quadro atual) e $F_{t+1}^{LR}$ (quadro seguinte). Cada quadro passa por uma camada de convolução e, em seguida, as saídas das camadas de convolução são concatenadas e alimentadas em uma nova camada de convolução. Por fim, o resultado é passado por uma terceira camada de convolução, que gera o quadro atual restaurado $F_t^{SR}$.

Figura 11 – Arquitetura do modelo VSRnet.




FONTE: Adaptado de Kappeler et al.[18].

A saída da primeira camada, denotada por $H_1$, possui dimensões $M \times N \times C$, onde $C$ é o número de filtros utilizados. Os elementos de $H_1$ são calculados utilizando a Equação (4.1),

$$h_1(i,j,c) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{t'=t-1}^{t+1} w_v(m,n,t') \times y_{t'}(i-m, j-n, t') + b_v(c), \qquad (4.1)$$

onde $w_v(.)$ e $b_v(.)$ são pesos e viés; $c$ é o índice do filtro e $y_{t'}$ é o quadro no tempo $t'$.

A compensação de movimento dos quadros $F_{t-1}^{LR}$ e $F_{t+1}^{LR}$ seria ideal se fossem idênticos ao quadro corrente $F_t^{LR}$, resultando em três quadros de entrada idênticos. No entanto, devido a erros inerentes ao algoritmo de fluxo óptico e à presença de objetos deformáveis, a compensação de movimento pode não ser precisa.

Para lidar com esse desafio, os autores aplicaram uma abordagem adaptativa de compensação de movimento (AMC, do inglês *adaptive motion compensation*) para reduzir a influência de estruturas vizinhas na reconstrução em caso de registro incorreto. Isso é especialmente útil quando ocorrem grandes movimentos ou desfoques de movimento no vídeo, evitando artefatos indesejados e contornos visíveis no quadro reconstruído $F^{SR}$.

No modelo apresentado na Figura 11, os pesos da camada convolucional 1 são compartilhados pelos pares $(t-1,\, t+1)$, o que significa que a mesma filtragem é aplicada em posições correspondentes nos quadros vizinhos. Essa abordagem explora a natureza convolucional da rede na dimensão temporal, permitindo reduzir o tempo de treinamento em quase 20%.

Em resumo, o modelo proposto utiliza a compensação de movimento adaptável (AMC) e compartilhamento de pesos na camada convolucional 1 para lidar com erros de compensação de movimento e reduzir a influência de estruturas vizinhas. Essas técnicas contribuem para melhorar o desempenho na super-resolução multiquadro.

A compensação de movimento foi realizada utilizando a seguinte equação:

$$y_{t-T}^{amc}(i,j) = (1 - r(i,j))y_t(i,j)y_{t-T}^{mc}(i,j), \tag{4.2}$$

onde $r(i,j)$ controla a combinação convexa entre o quadro atual e seu vizinho em cada posição do pixel $(i,j)$. O quadro central é representado por $y_t$, o quadro vizinho com compensação de movimento é representado por $y_{t-T}^{mc}$ e o quadro vizinho após a compensação de movimento é representado por $y_{t-T}^{amc}$.

A equação utilizada para calcular o valor de $r(i,j)$ é a seguinte:

$$r(i,j) = \exp(-k \cdot e(i,j)), \tag{4.3}$$

onde $k$ é uma constante e $e(i,j)$ representa a compensação de movimento. Erros significativos na compensação de movimento podem ocorrer devido a movimentos intensos, oclusões, desfoque do objeto ou quando a posição $(i,j)$ está próxima de um limite de movimento.

De acordo com as Equações 4.2 e 4.3, quando o erro de compensação de movimento $e(i,j)$ é alto no local $(i,j)$, o peso correspondente $r(i,j)$ é baixo. Isso implica que o pixel compensado de movimento adaptativo é apenas o pixel no quadro atual $y_t$, indicando que as informações dos quadros vizinhos não são utilizadas devido à falta de confiabilidade.

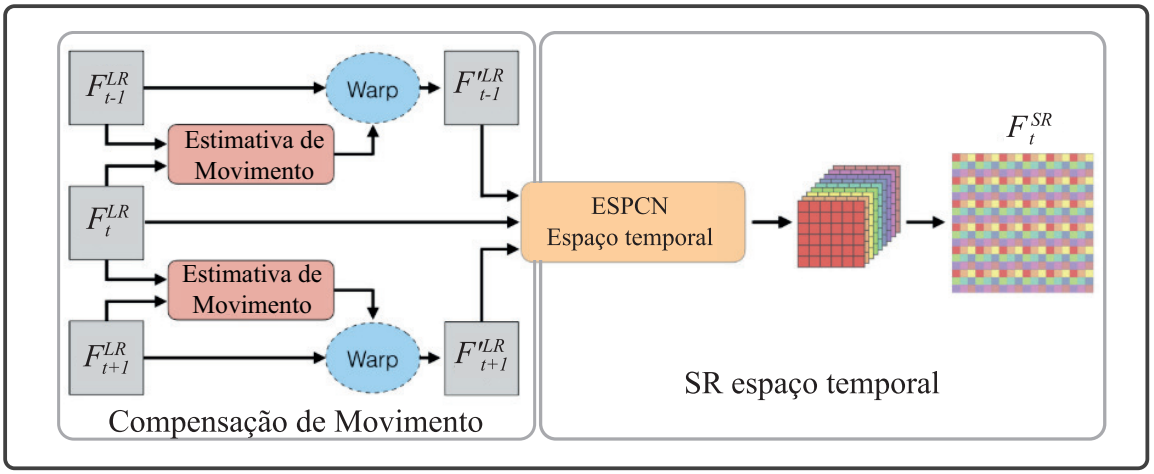
Uma limitação do modelo VSRnet é que ele não realiza a reescala dos quadros dos vídeos, apenas melhora sua qualidade. A reescala é realizada de forma igual ao modelo

SRCNN, por meio de um processo de interpolação bicúbica. Como resultado, todas as operações de convolução nas três camadas são executadas em imagens de alta resolução (HR), o que aumenta consideravelmente a complexidade computacional do modelo.

## 4.2 SUPER-RESOLUÇÃO DE VÍDEO COM REDES ESPAÇO-TEMPORAIS E COMPENSAÇÃO DE MOVIMENTO

Um modelo chamado *Video efficient sub-pixel convolutional neural network* (VESPCN) foi proposto por Caballero et al.[120] para abordar a tarefa de *Video super-resolution* (VSR). Esse modelo é uma extensão do ESPCN, apresentado na seção 3.2, e incorpora redes espaço-temporais e compensação de movimento. Os autores combinaram a técnica de convolução por subpixel com essas abordagens para desenvolver um algoritmo eficaz de VSR.

Figura 12 – Arquitetura do modelo VESPCN.




FONTE: Adaptado de Caballero et al.[120].

Conforme ilustrado na Figura 12, o modelo VESPCN é composto por dois módulos principais. O primeiro módulo é responsável por estimar a compensação de movimento dos quadros. Essa etapa identifica os deslocamentos entre os quadros e calcula os vetores de movimento correspondentes. Em seguida, os quadros de entrada são encaminhados para o segundo módulo, chamado de módulo de SR espaço-temporal.

No módulo de SR espaço-temporal, um número ímpar de quadros consecutivos é processado para estimar o quadro intermediário SR. Essa abordagem leva em consideração a informação temporal dos quadros para melhorar a qualidade do resultado final. O modelo utiliza redes neurais profundas para realizar a reconstrução de alta resolução, levando em conta tanto as informações espaciais quanto temporais dos quadros de entrada.

Essa combinação de estimativa de compensação de movimento e processamento espaço-temporal permite ao modelo VESPCN gerar quadros de alta resolução e qualidade

a partir de uma sequência de quadros de baixa resolução.

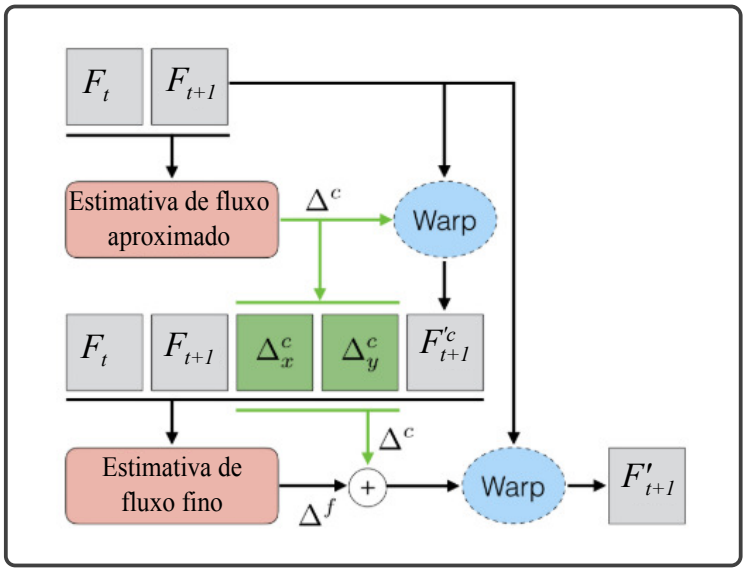
### 4.2.1 COMPENSAÇÃO DE MOVIMENTO COM TRANSFORMADOR ESPACIAL

Na Figura 13, é apresentado um esquema do módulo de compensação de movimento do modelo VESPCN. Essa ilustração simplificada mostra o processo de compensação de movimento entre dois quadros, $F_t$ e $F_{t+1}$.

O objetivo do módulo é encontrar a melhor representação do fluxo óptico, que relaciona o novo quadro $F_{t+1}$ com um quadro de referência atual, $F_t$. O fluxo óptico é considerado em termos de pixels, permitindo o deslocamento de cada pixel para uma nova posição. Os pixels resultantes são organizados em uma grade regular, exigindo a interpolação de volta para preencher os espaços vazios. Nesse caso, foi utilizada a interpolação bilinear $\mathfrak{T}\{.\}$.

O processo de compensação de movimento busca determinar as melhores posições para os pixels no quadro $F_{t+1}$, de forma a minimizar as diferenças entre os quadros e obter uma estimativa precisa do movimento. Isso permite alinhar corretamente os quadros e preparar o terreno para o subsequente processo de reconstrução de alta resolução.

Figura 13 – Arquitetura do módulo de compensação de movimento com transformador espacial.




FONTE: Adaptado de Caballero et al.[120].

O fluxo óptico é uma função dos parâmetros $\theta_{\Delta,t+1}$ e é representado por dois mapas de características $\Delta_{t+1} = (\Delta_{t+1}x, \Delta_{t+1}y; \theta_{\Delta,t+1})$, que representam o deslocamento nas dimensões $x$ e $y$. Dessa forma, uma imagem compensada pode ser expressa como $F'_{t+1}(x, y) = \mathfrak{T}\{F_{t+1}(x + \Delta_{t+1}x, y + \Delta_{t+1}y)\}$, ou de forma abreviada como $F'_{t+1} = \mathfrak{T}\{F_{t+1}(\Delta_{t+1})\}$. Isso implica que a imagem compensada $F'_{t+1}$ é obtida aplicando o deslocamento representado por $\Delta_{t+1}$ aos pixels de $F_{t+1}$ usando a interpolação bilinear $\mathfrak{T}\{.\}$.

Na Tabela 5, são apresentados detalhes da arquitetura das duas redes neurais convolucionais que compõem o módulo de compensação de movimento. A primeira rede é responsável por estimar o fluxo aproximado, enquanto a segunda rede realiza a estimativa de fluxo fino.

Na primeira rede, uma estimativa de fluxo com uma escala de aumento de 4× é obtida através da fusão dos dois quadros de entrada e da redução das dimensões espaciais usando um *stride* igual a dois. O fluxo estimado é então aumentado usando uma convolução por subpixel, resultando no mapa de fluxo aproximado $\Delta^c_{t+1}$. Esse mapa de fluxo é aplicado para distorcer a produção do quadro de destino ${F'_{t+1}}^c$. Em seguida, a imagem distorcida ${F'_{t+1}}^c$ é processada em conjunto com o fluxo estimado $\Delta^c_{t+1}$ e os quadros originais por meio de uma segunda rede neural, que realiza a estimativa de fluxo fino. Essa rede neural refinada produz um mapa de fluxo mais preciso, representado por $\Delta^f$.

Por fim, o quadro final compensado pelo movimento, $F'_{t+1}$, é obtido distorcendo-se o quadro alvo $F_{t+1}$ com o fluxo total $\Delta^c_{t+1}+\Delta^f_{t+1}$ aplicado usando a interpolação bilinear $\mathfrak{T}\{.\}$. Dessa forma, o quadro de destino $F'_{t+1}$ é obtido aplicando-se o deslocamento representado pelo mapa de fluxo total $\Delta^c_{t+1} + \Delta^f_{t+1}$ aos pixels do quadro alvo $F_{t+1}$.

Tabela 5 – Detalhes da arquitetura do módulo de compensação de movimento.

| **Camadas** | **Fluxo estimado** | **Fluxo fino** |
|---|---|---|
| 1 | Conv k5-n24-s2 / ReLU | Conv k5-n24-s2 / ReLU |
| 2 | Conv k3-n24-s1 / ReLU | Conv k3-n24-s1 / ReLU |
| 3 | Conv k5-n24-s2 / ReLU | Conv k3-n24-s1 / ReLU |
| 4 | Conv k3-n24-s1 / ReLU | Conv k3-n24-s1 / ReLU |
| 5 | Conv k3-n32-s1 / tanh | Conv k3-n8-s1 / tanh |
| 6 | Subpixel upscale ×4 | Subpixel upscale ×2 |

Fonte: Adaptado de Caballero et al.[120]

Para treinar o módulo de compensação de movimento, foi utilizada a função de erro MSE, a fim de otimizar os parâmetros $\theta_{\Delta,t+1}$ e minimizar a discrepância entre o quadro transformado e o quadro de referência. Além disso, foi empregada a função de erro Huber [121], definida na Equação (4.4), para penalizar os gradientes no mapa de fluxo, resultando em um comportamento mais suave no espaço.

A função de erro Huber é uma alternativa robusta ao erro quadrático tradicional, que é menos sensível a valores discrepantes. Ela é definida como:

$$\theta^*_{\Delta,t+1} = \underset{\theta_{\Delta,t+1}}{\arg\min} \, ||F_t - F'_{t+1}||_2^2 + \lambda H(\partial_{x,y}\Delta_{t+1}), \tag{4.4}$$

em que $\partial_{x,y}\Delta_{t+1}$ representa os gradientes espaciais do mapa de fluxo $\Delta_{t+1}$ e $\lambda$ é um

parâmetro de ajuste. A função Huber $H(\partial_{x,y}\Delta_{t+1})$ é definida como:

$$H(\partial_{x,y}\Delta) = \sqrt{\epsilon + \sum_{i=x,y} (\partial_x \Delta i^2 + \partial_y \Delta i^2)}, \tag{4.5}$$

em que $\epsilon$ é uma pequena constante adicional (por exemplo, $\epsilon = 10^{-2}$). Essa função apresenta um comportamento suave próximo à origem (como o erro quadrático) e uma dispersão gradual à medida que os gradientes aumentam, proporcionando uma penalização menos sensível a *outliers*.

Portanto, durante o treinamento, os parâmetros $\theta_{\Delta,t+1}$ são otimizados usando uma combinação do erro quadrático entre o quadro transformado $F'_{t+1}$ e o quadro de referência $F_t$, juntamente com a função Huber aplicada aos gradientes do mapa de fluxo $\Delta_{t+1}$, a fim de obter um melhor desempenho de compensação de movimento.

O módulo do transformador espacial apresenta vantagens em relação a outros mecanismos de compensação de movimento, pois pode ser combinado para treinar a compensação de movimento em conjunto com a rede de SR espaço-temporal. Ambos os módulos, o transformador espacial e o módulo de SR, são diferenciáveis, o que significa que suas operações podem ser representadas por funções contínuas e deriváveis, permitindo que sejam treinados de forma fim a fim.

Essa característica de diferenciabilidade é fundamental, pois permite otimizar os parâmetros de ambos os módulos de forma conjunta, visando minimizar a perda de maneira composta e melhorar a qualidade da super-resolução, incorporando compensação de movimento eficaz.

### 4.2.2 SUPER-RESOLUÇÃO ESPAÇO-TEMPORAL

O componente espaço-temporal do modelo recebe uma sequência de quadros consecutivos como entrada, que já foram processados pelo módulo de compensação de movimento. Para representar essa sequência na rede neural, introduz-se uma dimensão adicional chamada profundidade temporal ($D_l$), sendo $D_0$ o número ímpar de quadros consecutivos. O raio temporal de um bloco espaço-temporal é definido como $R = \frac{D_0-1}{2}$. Os quadros de entrada centrados no tempo $t$ são denotados por $F^{LR}_{[t-R:t+R]} \in \mathbb{R}^{H\times W\times D_0}$.

$$f_l(F^{LR}_{[t-R:t+R]}; \theta_l) = \phi\left(W_l \times f_{l-1}(F^{LR}; \theta_{l-1}) + b_l\right), \forall l, \tag{4.6}$$

Formalmente, o módulo espaço-temporal pode ser representado pela Equação (4.6), em que $f_l(\cdot; \theta_l)$ é a função que mapeia a entrada de baixa resolução ($LR$) para alta resolução ($HR$). O modelo é composto por $L$ camadas, em que $l \in [0, L-1]$, e cada camada $l$ possui um conjunto de parâmetros $\theta_l$ com pesos $W_l$ e viés $b_l$. A camada de entrada é representada por $f_l(F^{LR}_{[t-R:t+R]}; \theta_0) = F^{LR}$, e a dimensão dos filtros é definida

por $d_l \times n_{l-1} \times n_l \times k_l \times k_l$, onde $n_l$, em que $n_l$ e $k_l$ indicam o número e o tamanho dos filtros da camada $l$, respectivamente. A última camada do modelo é a função de subpixel, conforme apresentada na Equação (3.6).

$$\theta^* = \arg\min_{\theta} ||F^{HR} - f(F^{LR}; \theta)||_2^2. \tag{4.7}$$

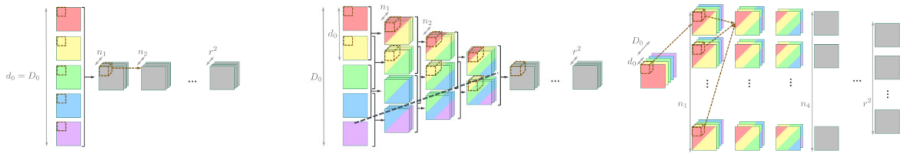
Os parâmetros do módulo espaço-temporal são otimizados minimizando a função de erro MSE dada pela Equação (4.7), em que $\theta^*$ representa os parâmetros ótimos que minimizam o erro entre a saída de alta resolução desejada ($F^{HR}$) e a saída da rede ($f(F^{LR}; \theta)$). A otimização conjunta dos módulos de compensação de movimento e espaço-temporal é realizada combinando as Equações (4.5) e (4.7) na Equação (4.8). Nessa equação, $\theta^*$ e $\theta^*_\Delta$ são os conjuntos ótimos de parâmetros que minimizam a função de erro composta, que inclui a diferença entre o quadro de alta resolução desejado e o quadro obtido através do fluxo de compensação de movimento, juntamente com as penalidades para a diferença entre os quadros de entrada e as penalidades dos gradientes do mapa de fluxo.

$$\begin{aligned}(\theta^*, \theta^*_\Delta) = &\arg\min_{\theta, \theta^*_\Delta} ||F_t^{HR} - f(F'^{LR}_{t-1:t+1}; \theta)||_2^2 \\ &+ \sum_{i=\pm 1} \left[\beta ||F'^{LR}_{t+i} - F_t^{LR}||_2^2 + \lambda H(\partial_{x,y} \Delta_{t+i})\right].\end{aligned} \tag{4.8}$$

Foram propostos três diferentes modos de fusão de quadros: fusão precoce, fusão lenta e convolução 3D. No modo de **fusão precoce**, a profundidade temporal da camada de entrada é combinada com o número de quadros ($d_0 = D_0$), capturando todas as informações temporais na primeira camada, e as operações subsequentes são semelhantes às de uma rede de super-resolução de imagem única (SISR), em que $d_l = 1$ para $l \geq 1$. A Figura 14a ilustra a arquitetura de fusão precoce para o caso em que $D_0 = 5$, sendo que as cores representam a dimensão temporal e o mapeamento de saída para o espaço 2D é omitido.

Na abordagem de **fusão lenta**, as informações temporais são gradualmente incorporadas em uma estrutura hierárquica, mesclando-se lentamente à medida que progridem pela rede. Nesse caso, a profundidade temporal das camadas da rede é configurada para $1 \leq d_l < D_0$, de modo que algumas camadas possuem uma extensão temporal até que todas as informações sejam mescladas e a profundidade seja reduzida para um. A Figura 14b ilustra uma rede de fusão lenta com $D_0 = 5$ e uma taxa de fusão definida por $d_l = 2$ para $l \leq 3$, e $d_l = 1$ caso contrário. Isso significa que, em cada camada, apenas dois quadros consecutivos ou mapas de características são mesclados até que a profundidade temporal da rede seja reduzida para um.

Figura 14 – Modelos de fusão espaço-temporal.



(a) Fusão precoce. (b) Fusão lenta. (c) Convolução 3D.

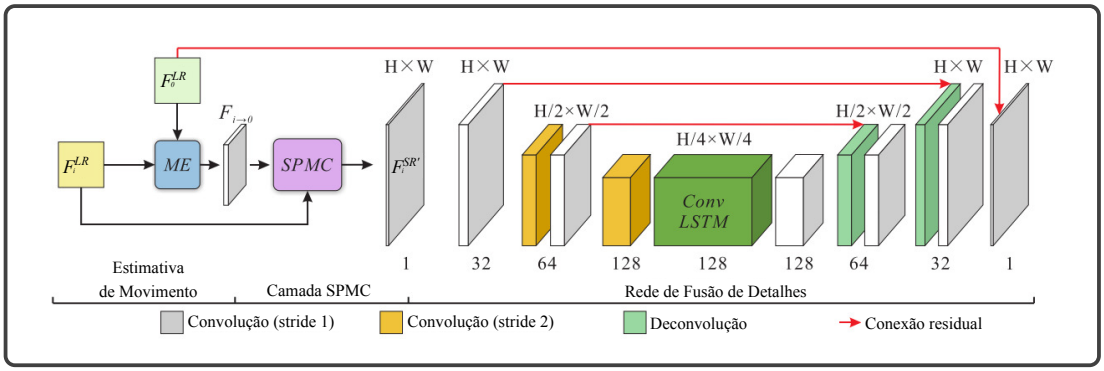
FONTE: Adaptado de Caballero et al.[120].

No modo de **convolução 3D**, ocorre o compartilhamento de pesos da camada em toda a dimensão temporal. Considerando o processamento *online* dos quadros, quando um novo quadro está disponível, o resultado de algumas camadas do quadro anterior pode ser reutilizado. Por exemplo, na Figura 14c, assumindo que o quadro inferior seja o último quadro recebido, todas as ativações acima da linha tracejada estão imediatamente disponíveis, pois foram necessárias para processar o quadro anterior.

## 4.3 SUPER-RESOLUÇÃO DE VÍDEO COM CNN E COMPENSAÇÃO DE MOVIMENTO POR SUBPIXEL

Em Tao et al.[122], foi proposto um método chamado *Detail-revealing deep video super-resolution* (DRDVSR) que consiste em três módulos principais: estimativa de movimento, compensação de movimento e fusão de detalhes. A Figura 15 apresenta a arquitetura geral do modelo, e nas seções seguintes, será fornecida uma descrição detalhada de cada um dos módulos.

Figura 15 – Arquitetura do modelo DRDVSR.




FONTE: Adaptado de Tao et al.[122].

### 4.3.1 ESTIMATIVA DE MOVIMENTO

Para estimar o fluxo de movimento, o método proposto em Caballero et al.[120] foi utilizado, conforme descrito na seção 4.2.1. No modelo DRDVSR, apenas dois quadros LR consecutivos são utilizados como entrada, e o modelo produz um campo de movimento LR definido pela Equação (4.9):

$$F_{i\to j} = \mathbf{Net}_{ME}(F_i^{LR}, F_j^{LR}; \theta_{ME}), \tag{4.9}$$

onde $F_{i\to j} = (u_{i\to j}, v_{i\to j})$ representa o campo de movimento dos quadros $F_i^{LR}$ para $F_j^{LR}$, e $\theta_{ME}$ é o conjunto de parâmetros do módulo de estimativa de movimento.

### 4.3.2 CAMADA SPMC

O alinhamento dos quadros é realizado por meio de uma camada que utiliza o método de compensação de movimento por subpixel, denominado *Sub-pixel motion compensation* (SPMC). O método é definido pela Equação (4.10):

$$F^{SR'} = \mathbf{Layer}_{SPMC}(F^{LR}, F; \alpha), \tag{4.10}$$

onde $F^{LR}$ representa o quadro de entrada, $F^{SR'}$ é o quadro de saída da camada SPMC, $F$ é o fluxo óptico utilizado para a distorção transposta, e $\alpha$ é o fator de escala. A camada SPMC é composta por dois submódulos.

A etapa do **Gerador de Amostragem de Grade** é onde as coordenadas transformadas são calculadas com base no fluxo estimado $F = (u, v)$, conforme mostrado na Equação (4.11):

$$\begin{pmatrix} x_p^s \\ y_p^s \end{pmatrix} = W_{F;\alpha} \begin{pmatrix} x_p \\ y_p \end{pmatrix} = \alpha \begin{pmatrix} x_p + u_p \\ y_p + v_p \end{pmatrix}, \tag{4.11}$$

onde $p$ indexa os pixels no espaço da imagem LR, $x_p$ e $y_p$ são as duas coordenadas de $p$, $u_p$ e $v_p$ são os vetores de fluxo estimados do estágio anterior. A transformação de coordenadas é definida como $W_{F;\alpha}$, que depende do fluxo óptico $F$ e do fator de escala $\alpha$. $x_p^s$ e $y_s^p$ são as coordenadas transformadas em um espaço de imagem ampliado.

O **Amostrador de Imagem Diferenciável** é responsável por construir a imagem de saída no espaço ampliado de acordo com $x_p^s$ e $y_p^s$, conforme definido na Equação (4.12):

$$F_q^{SR'} = \sum_{p=1} F_p^{LR} M(x_p^s - x_q) M(y_p^s - y_q), \tag{4.12}$$

onde $q$ indexa os pixels na imagem SR, $x_q$ e $y_q$ são as duas coordenadas do pixel $q$ na grade SR, e $M(\cdot)$ é o núcleo de amostragem que define os métodos de interpolação da imagem.

Essa camada é totalmente diferenciável, o que permite que o erro de retropropagação flua suavemente pelos campos. Por ser diferenciável, apresenta três vantagens:

- É capaz de obter simultaneamente a compensação de movimento e aprimoramento da resolução.
- É livre de parâmetros e totalmente diferenciável, o que permite que seja incorporada efetivamente em redes neurais com quase nenhum custo adicional.
- A lógica subjacente dessa camada é baseada em um modelo de imagem LR preciso, garantindo um bom desempenho.

### 4.3.3 FUSÃO DE DETALHES

A camada SPMC produz uma série de quadros compensados de movimento $\{F_i^{SR'}\}$ definidos como:

$$F_i^{SR'} = \mathbf{Layer}_{SPMC}(F_i^{LR}, F_{i\to 0}; \alpha), \tag{4.13}$$

A rede de fusão não é trivial devido a algumas considerações. Primeiro, os quadros $\{F_i^{SR'}\}$ têm o tamanho de HR, resultando em grandes mapas de características, o que torna o custo computacional um fator importante. Segundo, devido à propriedade de distorção direta e ampliação zero, $\{F_i^{SR'}\}$ é esparsa, com a maioria dos pixels tendo valor zero (cerca de 15/16 são zeros para um fator de escala de 4×). Isso requer que a rede tenha grandes campos receptivos para capturar padrões de imagem em $\{F_i^{SR'}\}$.

No entanto, usar uma simples interpolação para preencher essas áreas vazias não é uma solução adequada, pois os valores interpolados podem dominar durante o treinamento. Por fim, é importante que o quadro de referência sirva como orientação para o quadro SR, garantindo consistência em termos de estruturas de quadro. No entanto, enfatizar muito o quadro de referência pode levar a um efeito adverso, negligenciando as informações presentes nos outros quadros. O caso extremo seria o sistema se comportar como uma SISR.

Para contornar esses desafios, a arquitetura da rede de fusão de detalhes foi projetada como um codificador-decodificador [123] com conexões residuais, inspirada em trabalhos anteriores [123, 124, 125] que demonstraram eficácia em várias tarefas de regressão de imagem. Uma sub-rede de codificação reduz o tamanho do quadro de entrada $F^{SR'}$ em 1/4, o que ajuda a reduzir o custo computacional. Essa tarefa também torna os mapas de características menos esparsos, permitindo a efetiva agregação de informações sem a necessidade de redes muito profundas.

As conexões residuais são usadas em todas as etapas para acelerar o treinamento. Um módulo de convolução *Long short-term memory* (LSTM) foi inserido como uma opção natural para entrada sequencial no estágio intermediário. A estrutura da rede é definida pela Equação (4.14):

$$\begin{aligned} f_i &= \mathbf{Net}_E(F_i^{SR'}; \theta_E) \\ g_i, s_i &= \mathbf{ConvLSTM}(f_i, s_{i-1}; \theta_{LSTM}) \\ F_0^{(i)} &= \mathbf{Net}_D(g_i, S_i^E; \theta_D) + F_0^{LR\uparrow} \end{aligned} \tag{4.14}$$

Nessa equação, $\mathbf{Net}_E$ e $\mathbf{Net}_D$ são redes convolucionais de codificação e decodificação, respectivamente, com parâmetros $\theta_E$ e $\theta_D$. $f_i$ é a saída da rede de codificação; $g_i$ é a entrada da rede de decodificação; $s_i$ é o estado oculto da LSTM na $i$-ésima etapa; $S_i^E$ são mapas de características intermediárias de $\mathbf{Net}_E$ usados para as conexões residuais; $F_0^{LR\uparrow}$ é a imagem $F_0^{LR}$ restaurada por interpolação bicúbica, e $F_0^{(i)}$ é a saída da $i$-ésima etapa.

A primeira camada de $\mathbf{Net}_E$ e a última camada de $\mathbf{Net}_D$ têm um tamanho de filtro de $5 \times 5$. Todas as outras camadas de convolução usam um tamanho de filtro de $3 \times 3$, incluindo as da **ConvLSTM**. As camadas de deconvolução têm um tamanho de filtro de $4 \times 4$ e um *stride* de 2. A função de ativação ReLU é utilizada em todas as camadas de convolução e deconvolução. Para as conexões residuais, é utilizado o operador de soma entre as camadas conectadas. Outros parâmetros estão explicitados na Figura 15.

## 4.4 SUPER-RESOLUÇÃO DE VÍDEO COM REDE NEURAL PROFUNDA USANDO FILTROS DE UPSAMPLING DINÂMICOS SEM COMPENSAÇÃO EXPLÍCITA DE MOVIMENTO

Os métodos convencionais de VSR baseados em aprendizagem profunda apresentam uma abordagem de duas etapas, que consiste na estimativa de movimento e no procedimento de compensação, seguido pelo processo de *upsampling*. No entanto, essa abordagem enfrenta alguns desafios. Primeiramente, os resultados dependem de uma estimativa precisa de movimento, o que pode ser difícil de obter em certos cenários. Além disso, a geração do quadro de saída SR, por meio da mistura dos valores dos quadros LR de entrada compensados por múltiplos movimentos, pode resultar em um quadro SR desfocado.

Para contornar esses problemas, Jo et al.[126] propuseram um modelo chamado *Video super-resolution using dynamic upsampling filters* (VSR-DUF) que adota uma abordagem diferente dos métodos anteriores. Nesse modelo, em vez de calcular e compensar explicitamente o movimento entre os quadros de entrada, as informações de movimento são utilizadas de forma implícita para gerar filtros de *upsampling* dinâmicos. Com base nesses filtros gerados, o quadro SR é construído diretamente por meio de uma filtragem local no quadro central de entrada. Como resultado, o modelo produz vídeos com maior nitidez e consistência temporal.

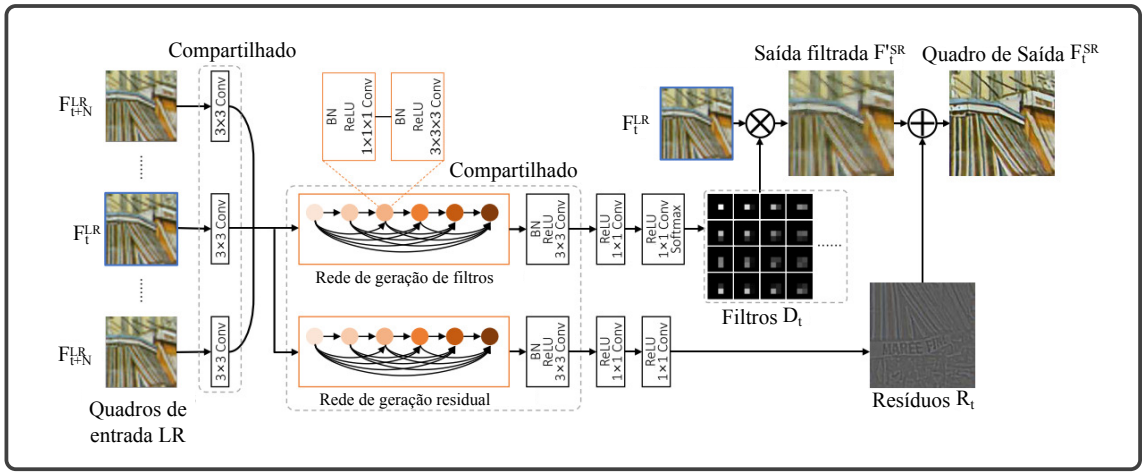
A abordagem do VSR-DUF tem a vantagem de evitar a necessidade de estimativa de movimento explícita, o que simplifica o processo e reduz a dependência de uma estimativa precisa de movimento. Além disso, ao filtrar localmente o quadro central de entrada, o modelo é capaz de preservar detalhes e evitar o desfoque que pode ocorrer na etapa de mistura de quadros. Essa abordagem resulta em vídeos SR com melhor qualidade e maior fidelidade em relação aos métodos convencionais.

A arquitetura do modelo VSR-DUF é ilustrada na Figura 16, e sua definição matemática é apresentada na Equação (4.15). Nessa equação, $F_t^{SR}$ representa o quadro em alta definição reconstruído no tempo $t$. O modelo é definido pela função $G_\theta$, que é uma rede neural com parâmetros $\theta$. O raio temporal $N$ é centralizado em torno do tempo $t$.

$$F_t^{SR} = G_\theta(F_{t-N:t+N}^{LR}), \tag{4.15}$$

A entrada para o modelo é um tensor de tamanho $T \times H \times W \times C$, onde $T = 2N+1$ é o número total de quadros de baixa resolução $F^{LR}$ considerados, $H$ e $W$ são as dimensões espaciais dos quadros de baixa resolução e $C$ é o número de canais de cores. A saída do modelo é um tensor de tamanho $1 \times rH \times rW \times C$, correspondendo ao quadro reconstruído em alta definição $F_t^{SR}$, onde $r$ é o fator de escala.

Figura 16 – Arquitetura do modelo VSR-DUF.




FONTE: Adaptado de Jo et al.[126].

Na saída, o modelo gera o quadro $F_t^{SR}$ usando os filtros dinâmicos de *upsampling* $D_t$ e os resíduos $R_t$, que são obtidos a partir de um conjunto de quadros de baixa resolução $\{F_{t-N:t+N}^{LR}\}$ fornecidos como entrada para o modelo. O quadro central de entrada $F_t^{LR}$ é inicialmente filtrado localmente pelos filtros dinâmicos de *upsampling* $D_t$, resultando na saída filtrada $F_t'^{SR}$. Em seguida, os resíduos $R_t$ são adicionados a $F_t'^{SR}$ para gerar o quadro de saída $F_t^{SR}$. Os filtros dinâmicos de *upsampling* $D_t$ e os resíduos $R_t$ são gerados por sub-redes descritas nas seções 4.4.1 e 4.4.2.

As duas redes, de filtros dinâmicos e de geração de resíduos, compartilham a maioria dos pesos, o que reduz a sobrecarga computacional. Elas são compostas por blocos densos que incluem camadas de convolução 3D, o que é mais adequado para explorar informações espaço-temporais dos vídeos [127]. Cada bloco residual é composto por BN, ReLU, convolução 3D $1 \times 1 \times 1$, BN e convolução 3D $3 \times 3 \times 3$, nesta ordem. Os quadros de entrada são processados inicialmente por uma camada de convolução 2D $3 \times 3 \times 3$ compartilhada e concatenados ao longo do eixo temporal. Em seguida, os mapas de características espaço-temporais resultantes passam pelos blocos densos 3D e são processados em ramificações separadas, que consistem em várias camadas convolucionais 2D para gerar as duas saídas. Para obter a saída final $F_t^{SR}$, a saída filtrada $D_t$ é adicionada aos resíduos $R_t$.

$$H(F_t^{SR}, F_t^{HR}) = \begin{cases} \frac{1}{2}||F_t^{SR} - F_t^{HR}||_2^2 & \text{se } ||F_t^{SR} - F_t^{HR}||_1 \leq \delta, \\ \delta||F_t^{SR} - F_t^{HR}||_1 - \frac{1}{2}\delta^2 & \text{caso contrário,} \end{cases} \tag{4.16}$$

Durante o treinamento do modelo, foi utilizada a função de erro de Huber, conforme definida na Equação (4.16). Essa função de erro é empregada para medir a diferença entre o quadro de saída $F_t^{SR}$ e o quadro de alta resolução correspondente $F_t^{HR}$. O parâmetro $\delta$ é um limiar definido com o valor $10^{-2}$. O otimizador Adam [102] foi utilizado, com uma taxa de aprendizagem inicial de $10^{-3}$, multiplicada por $10^{-1}$ a cada dez épocas.

### 4.4.1 FILTROS DE UPSAMPLING DINÂMICOS

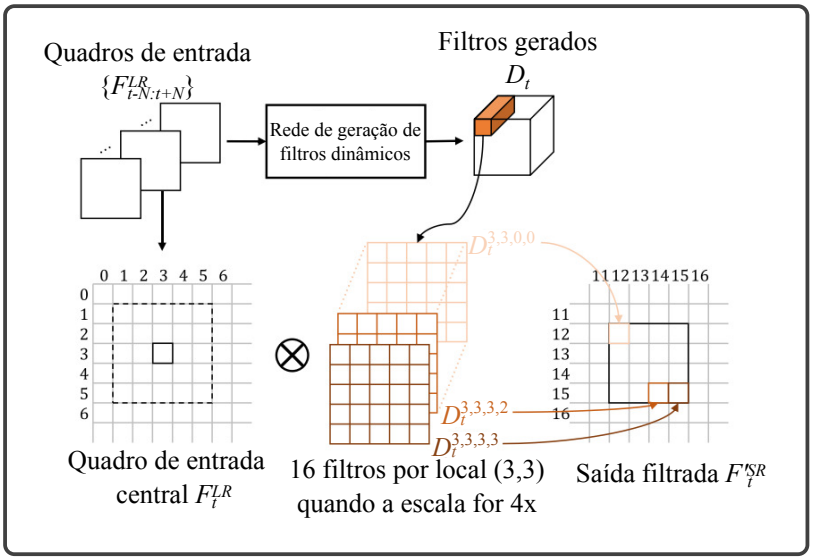
Esse tipo de filtro foi inspirado na rede de filtros dinâmicos proposta por Brabandere et al.[128]. Os filtros de *upsampling* são gerados dinamicamente, levando em consideração a vizinhança espaço-temporal de cada pixel nos quadros $F^{LR}$. A visão do procedimento de *upsampling* dinâmico é ilustrada na Figura 17.

No início do processo, um conjunto de quadros de entrada $F_{t-N:t+N}^{LR}$ é fornecido à rede de geração de filtros dinâmicos. Nos experimentos realizados pelos autores, foi utilizado um conjunto de 7 quadros, definindo $N$ como 3. Após o treinamento da rede, é gerado um conjunto de filtros de *upsampling* $D_t$, com dimensões $r^2HW$, que são utilizados para criar novos pixels no quadro filtrado $F_t'^{SR}$.

Em seguida, para cada valor de pixel em $F_t'^{SR}$, é realizada uma filtragem local no quadro central de entrada $F_t^{LR}$ usando o filtro correspondente $D_t^{y,x,v,u}$. Em outras palavras, cada pixel em $F_t'^{SR}$ é obtido aplicando-se o filtro $D_t^{y,x,v,u}$ ao pixel correspondente em $F_t^{LR}$, como mostrado na Equação (4.17).

$$F_t'^{SR}(yr + v, xr + u) = \sum_{j=-2}^{2} \sum_{i=-2}^{2} D_t^{y,x,v,u}(j+2, i+2) F_t^{LR}(y + j, x + i), \tag{4.17}$$

Figura 17 – Detalhes dos filtros dinâmicos.




FONTE: Adaptado de Jo et al.[126].

onde $y$ e $x$ são as coordenadas no quadro $LR$, e $v$ e $u$ são as coordenadas em cada bloco de saída $r \times r$ $(0 \leq v, u \leq r-1)$. É importante observar que essa operação é semelhante à deconvolução ou convolução transposta, permitindo que a rede seja treinada de ponta a ponta, usando *backpropagation*.

Essa abordagem difere dos métodos de VSR descritos anteriormente, onde redes neurais profundas aprendem a reconstruir quadros de SR através de convoluções no espaço de características. Em vez disso, esse método utiliza uma rede neural profunda para aprender os melhores filtros de *upsampling*, que são usados para reconstruir diretamente os quadros de SR a partir de quadros LR específicos. Esses filtros dinâmicos são criados com base nos movimentos dos pixels, observando as vizinhanças espaço-temporais dos pixels e evitando a compensação explícita de movimento.

### 4.4.2 APRENDIZAGEM RESIDUAL

Após a aplicação dos filtros dinâmicos de *upsampling*, a saída filtrada $F_t'^{SR}$ pode não ser uma imagem tão nítida, podendo haver detalhes que não puderam ser recuperados completamente.

Para contornar esse problema, uma imagem residual $R_t$ foi introduzida. Essa imagem residual é gerada a partir dos vários quadros de entrada e é adicionada à saída filtrada $F_t'^{SR}$ como um complemento, visando aumentar os detalhes de alta frequência. Ao combinar esses componentes complementares, foi possível obter uma melhor nitidez espacial e consistência temporal no quadro de saída $F_t^{SR}$.

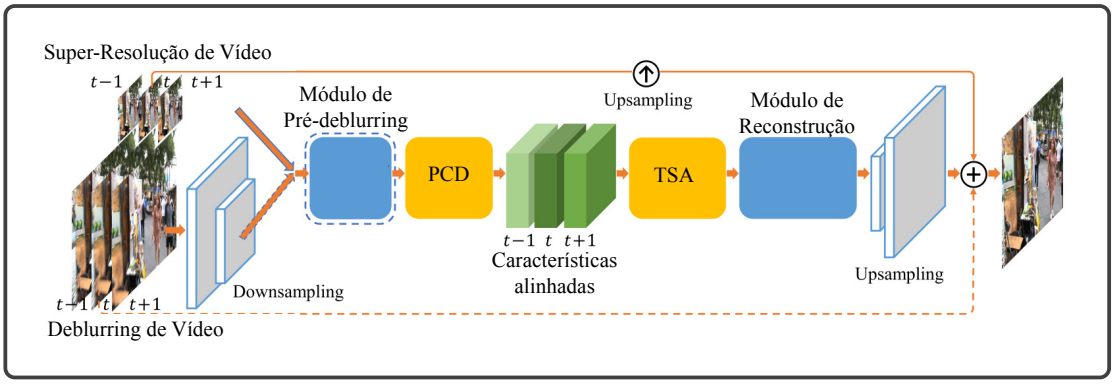
## 4.5 RESTAURAÇÃO DE VÍDEO COM REDES CONVOLUCIONAIS DEFORMÁVEIS APRIMORADAS

Em Wang et al.[28], foi proposto um método chamado *Video restoration with enhanced deformable convolutional networks* (EDVR), que é aplicável a várias tarefas de restauração de vídeo, incluindo super-resolução e *deblurring*.

O núcleo do método EDVR é composto por dois módulos principais: um módulo de alinhamento chamado *Pyramid, cascading and deformable convolutions* (PCD) e um módulo de fusão chamado *Temporal and spatial attention* (TSA) (Agregação Temporal e Espacial).

A Figura 18 mostra a arquitetura geral do modelo EDVR. Essa arquitetura é projetada de forma genérica, sendo adequada para várias tarefas de restauração de vídeo, como super-resolução, *deblurring*, eliminação de ruído, desbloqueio, entre outras. No entanto, neste contexto, o método será descrito apenas em relação às tarefas de super-resolução.

Figura 18 – Visão geral do framework EDVR.




FONTE: Adaptado de Wang et al.[28].

O modelo EDVR adota uma abordagem onde $2N + 1$ quadros de baixa resolução $F^{LR}$ são utilizados como entrada e a saída gerada é um único quadro de alta resolução $F^{SR}$. Esses quadros de baixa resolução são alinhados em relação ao quadro central por meio do módulo de alinhamento PCD, que opera no nível de características dos quadros. O módulo de fusão TSA é responsável por fundir as informações dos diferentes quadros, permitindo a geração de um quadro de alta resolução mais refinado.

Os recursos fundidos resultantes passam por um módulo de reconstrução, que é composto por uma cascata de blocos residuais. É importante destacar que esse módulo pode ser substituído por outros modelos avançados de SISR caso necessário. No final da rede, é realizada a operação de *upsampling* para aumentar o tamanho espacial da imagem.

Por fim, o quadro de alta resolução $F^{SR}$ é obtido adicionando-se o quadro residual

gerado ao quadro ampliado diretamente por meio de interpolação. Essa etapa de adição permite a correção de detalhes finos que podem não ter sido completamente recuperados pelo processo de *upsampling*. Nas seções seguintes, estão descritos os módulos de alinhamento (PCD) e de fusão (TSA).

### 4.5.1 MÓDULO PCD

O módulo PCD do EDVR foi inspirado no modelo *Temporally deformable alignment network* (TDAN) [129], que utiliza convoluções deformáveis para alinhar os quadros vizinhos em relação ao quadro de referência no nível de características. Para lidar com movimentos grandes e complexos, o alinhamento é realizado em duas etapas: um alinhamento inicial grosseiro seguido por um alinhamento refinado.

O alinhamento inicial é realizado por meio de uma estrutura de pirâmide, onde as características dos quadros de escalas inferiores são alinhadas usando estimativas iniciais mais grosseiras. Em seguida, as compensações e as características alinhadas nas escalas superiores são propagadas para auxiliar no alinhamento preciso do movimento. Essa abordagem é semelhante ao conceito utilizado na estimativa de fluxo óptico [130, 131].

Além disso, foi adicionada uma cascata de convolução deformável adicional após a etapa de alinhamento piramidal para melhorar ainda mais a robustez do alinhamento. Essa cascata de convolução deformável permite capturar informações mais refinadas sobre o movimento e a deformação dos quadros, contribuindo para um alinhamento mais preciso.

O alinhamento deformável é aplicado às características de cada quadro vizinho $F_{t+i}^{LR}$, onde $i \in [-N : +N]$, utilizando um módulo deformável modulado, similar ao apresentado em Zhu et al.[132]. Esse módulo utiliza um núcleo de deformação convolutiva com locais de amostragem denotados por $K$. Os pesos $w_k$ e as compensações pré-especificadas $p_k$ para cada local $k$ são utilizados. Por exemplo, um kernel $3 \times 3$ é definido com $K = 9$ e $p_k \in \{(-1,-1),(-1,0),\ldots,(1,1)$.

As características alinhadas $F_{t+i}^{a}$ em uma determinada posição $p_0$ são obtidas pela Equação (4.18),

$$F_{t+i}^{a}(p_0) = \sum_{k=1}^{K} w_k \cdot F_{t+i}^{LR}(p_0 + p_k + \Delta p_k) \cdot \Delta m_k. \tag{4.18}$$

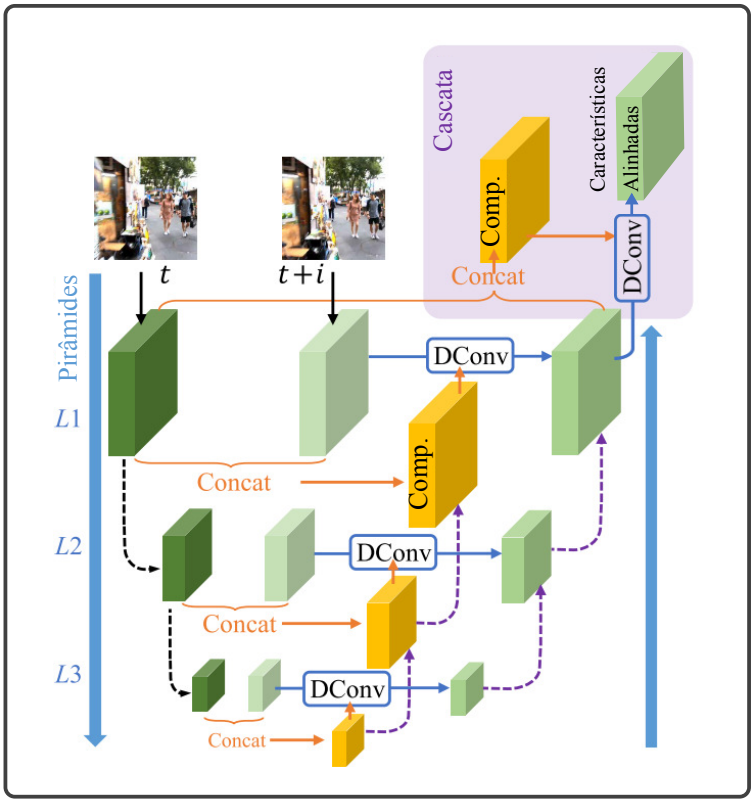
Aqui, $\Delta p_k$ é a compensação aprendida e $\Delta m_k$ é o escalar de modulação, que são gerados a partir da concatenação de características de um quadro vizinho e o quadro de referência, conforme descrito na Equação (4.19),

$$\Delta P_{t+i} = f([F_{t+i}^{LR}, F_{t}^{LR}]),\ i \in [-N : +N], \tag{4.19}$$

onde $\Delta P = \{\Delta p\}$, $f$ é uma função geral que consiste em várias camadas de convolução e $[\cdot,\cdot]$ representa a operação de concatenação.

Para simplificar, considerou-se apenas os desvios aprendidos $\Delta p_k$ e ignorou-se a modulação $\Delta m_k$ nas descrições e figuras. Como $p_0 + p_k + \Delta p_k$ é uma coordenada fracionária, a interpolação bilinear é aplicada, conforme visto em Dai et al.[133].

Figura 19 – Visão geral do módulo PCD.




FONTE: Adaptado de Wang et al.[28].

Foi proposto o módulo PCD baseado em princípios estabelecidos no fluxo óptico, utilizando o processamento piramidal [134, 135] e o refinamento em cascata [131, 130, 136], a fim de lidar com movimentos complexos e problemas de paralaxe maiores no alinhamento.

Como destacado com linhas tracejadas pretas na Figura 19, para gerar características $F^l_{t+i}$ no $l$-ésimo nível, aplicam-se filtros de convolução com *stride* para reduzir a amostragem das características em um fator de 2 em relação ao nível $(l-1)$ da pirâmide, resultando em $L$ níveis de pirâmide de representação de características. No $l$-ésimo nível, as compensações e as características alinhadas são geradas com um aumento de escala de $2\times$ a partir do nível superior $(l+1)$-ésimo, respectivamente (linhas roxas na Figura 19), conforme as equações (4.20) e (4.21).

$$\Delta P^l_{t+i} = f([F^{LR}_{t+i}, F^{LR}_t]), (\Delta P^{l+1}_{t+i})^{\uparrow 2}, \tag{4.20}$$

$$(F^a_{t+i})^l = g(\mathbf{DConv}(F^l_{t+i}, \Delta P^l_{t+i}), ((F^a_{t+i})^{l+1})^{\uparrow 2}), \tag{4.21}$$

A equação (4.20) representa a geração das compensações $\Delta P^l_{t+i}$ para o $l$-ésimo nível a partir da concatenação das características dos quadros vizinhos $F^{LR}_{t+i}$ e o quadro de referência $F^{LR}_t$, bem como o aumento de escala em 2× das compensações do nível $(l+1)$-ésimo. Já a equação (4.21) descreve a geração das características alinhadas $(F^a_{t+i})^l$ para o $l$-ésimo nível utilizando a convolução deformável **DConv** em conjunto com as compensações $\Delta P^l_{t+i}$ e o aumento de escala em 2× das características alinhadas do nível $(l+1)$-ésimo. A amostragem em escala é implementada utilizando a interpolação bilinear. No módulo PCD, foi utilizado uma pirâmide de três níveis, ou seja, $L = 3$. Para reduzir o custo computacional, o número de canais não foi aumentado à medida que os tamanhos espaciais diminuíam.

Após a estrutura da pirâmide, é aplicado um subsequente alinhamento deformável em cascata para refinar ainda mais as características alinhadas de maneira mais precisa (parte com fundo lilás na Figura 19). O módulo PCD melhora o alinhamento com precisão de subpixel. Além disso, o módulo de alinhamento do PCD é aprendido em conjunto com toda a estrutura, sem necessidade de supervisão adicional ou pré-treinamento em outras tarefas, como o fluxo óptico.

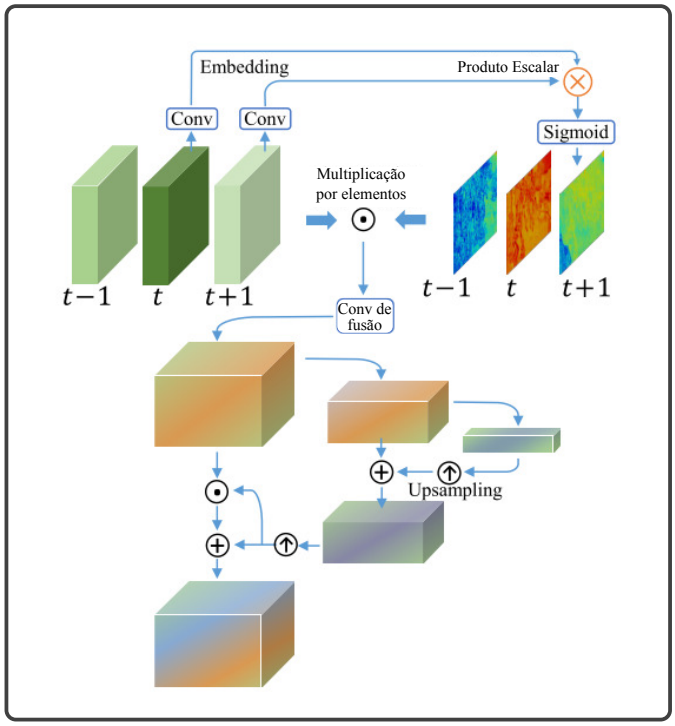
### 4.5.2 MÓDULO TSA

A relação temporal entre quadros consecutivos e a relação espacial dentro de cada quadro são elementos críticos a serem considerados, pois: 1) diferentes quadros vizinhos podem conter informações desiguais devido à oclusão, regiões desfocadas e problemas de paralaxe; 2) o desalinhamento resultante do estágio anterior de alinhamento pode afetar negativamente o desempenho subsequente da reconstrução. Portanto, é indispensável realizar uma agregação dinâmica dos quadros vizinhos no nível de pixel para obter uma fusão eficaz e eficiente. Para resolver esses problemas, foi proposto o módulo de fusão TSA, que atribui pesos de agregação a cada pixel em cada quadro, levando em consideração atenções temporais e espaciais durante o processo de fusão, como ilustrado na Figura 20.

O objetivo da ponderação temporal é calcular a similaridade entre os quadros em um espaço de incorporação (*embedding*). Intuitivamente, no espaço de incorporação, um quadro vizinho que seja mais semelhante ao quadro de referência deve receber um peso maior. Para cada quadro $i \in [-N : +N]$, a distância de similaridade $h$ pode ser calculada pela Equação (4.22), onde $\theta(F^a_{t+i})$ e $\phi(F^a_t)$ são dois *embeddings*, que podem ser obtidos usando filtros de convolução simples. A função de ativação sigmoid é usada para restringir as saídas no intervalo [0, 1], estabilizando a propagação reversa do gradiente. É importante notar que a ponderação temporal é específica para cada localização espacial, ou seja, o tamanho espacial de $h(F^a_{t+i}, F^a_t)$ é o mesmo que o de $F^a_{t+i}$.

$$h(F^a_{t+i}, F^a_t) = \text{sigmoid}(\theta(F^a_{t+i})^T \phi(F^a_t)), \tag{4.22}$$

Figura 20 – Visão geral do módulo TSA.




FONTE: Adaptado de Wang et al.[28].

Os mapas de ponderação temporal são então multiplicados ponto a ponto para as características alinhadas originais $F^a_{t+i}$. Uma camada adicional de convolução de fusão é adotada para agregar essas características ponderados moduladas $\widetilde{F}^a_{t+i}$, conforme as Equações (4.23) e (4.24), onde $\odot$ e $[\cdot,\cdot,\cdot]$ denotam a multiplicação e concatenação ponto a ponto, respectivamente.

$$\widetilde{F}^a_{t+i} = F^a_{t+i} \odot h(F^a_{t+i}, F^a_t), \tag{4.23}$$

$$F_{fusão} = \text{Conv}([\widetilde{F}^a_{t-N}, \ldots, \widetilde{F}^a_t, \ldots, \widetilde{F}^a_{t+N}]), \tag{4.24}$$

Em seguida, as máscaras de ponderação espacial são calculadas a partir das características fundidas. Um projeto de pirâmide é utilizado para aumentar o campo receptivo da ponderação. Posteriormente, as características fundidas são modulados pelas máscaras por meio de multiplicação e adição ponto a ponto.

## 4.6 COMPARATIVO DE DESEMPENHO DOS MODELOS

Nesta seção, são apresentadas as avaliações dos modelos discutidos neste capítulo, utilizando métricas de nível de pixel, como o PSNR e SSIM, com base nos experimentos realizados pelos respectivos autores de cada modelo. Na Tabela 6, alguns valores podem

estar ausentes devido ao fato de o modelo não ter sido avaliado nessa escala específica. Antes de apresentar as avaliações, faz-se uma breve descrição das três principais bases de dados utilizadas para tarefas de super-resolução em vídeos. Em seguida, são apresentadas as avaliações na base de dados Vid4 para os métodos descritos neste capítulo.

### 4.6.1 BASE DE DADOS PARA SUPER-RESOLUÇÃO DE VÍDEO

Para tarefas de super-resolução de vídeos, existem três bases de dados amplamente utilizadas. A mais antiga é a Vid4 [137], que consiste em quatro clipes que são comumente usados como referência para avaliação de desempenho de métodos de super-resolução de vídeos.

Outras duas bases de dados mais recentes são a Vimeo90K [138] e a REDS [139]. Ambas são bases de dados em larga escala que fornecem conjuntos de treinamento e teste para aprimorar algoritmos de super-resolução de vídeos. No entanto, é importante notar que essas bases de dados contêm vídeos com quadros sequenciais de uma mesma cena, o que difere das aplicações de *streaming* propostas neste trabalho. Portanto, a análise de desempenho de modelos de SR em bases de dados como a Vimeo90K e a REDS pode não refletir totalmente as demandas e desafios enfrentados em aplicações de *streaming*. A seguir, apresenta-se uma breve descrição de cada uma dessas bases.

1. **Vid4 [137]:** É uma base de dados de teste amplamente utilizada em super-resolução de vídeo, composta por quatro clipes. As imagens de baixa resolução são obtidas a partir da redução de escala utilizando um kernel bicúbico. A Vid4 é comumente utilizada como um conjunto de referência para avaliar o desempenho de algoritmos de super-resolução de vídeo.

2. **Vimeo90K [138]:** É uma base de dados de vídeos em larga escala e alta qualidade. Consiste em 89.800 clipes de vídeo baixados do Vimeo[1], abrangendo uma ampla variedade de cenas e ações em vídeos distintos. Essa base de dados foi criada para tarefas de processamento de vídeo, como interpolação temporal, *denoising*, desbloqueio e super-resolução de vídeo. Ela fornece conjuntos de treinamento e teste para avaliação de algoritmos.

3. **REDS [139]:** É uma base de dados de alta resolução (720p) voltada para tarefas de super-resolução e *deblurring* (remoção de desfoque). Ela é composta por 300 sequências de vídeo, cada uma contendo 100 quadros de resolução $720 \times 1280$. O conjunto de dados é dividido em 240 clipes para treinamento, 30 clipes para validação e 30 clipes para teste. O REDS oferece quatro subconjuntos de dados, cada um contendo diferentes degradações aplicadas às imagens de alta resolução,

[1] https://vimeo.com

como *downsampling* bicúbico, *downsampling* bicúbico com desfoque de movimento, desfoque de movimento e desfoque de movimento com artefatos de compactação. Essa base de dados abrange uma ampla diversidade de conteúdos, incluindo pessoas, objetos artesanais e ambientes urbanos.

### 4.6.2 DESEMPENHO DOS MODELOS

Na Tabela 6, são apresentados os valores de avaliação dos métodos para as métricas PSNR e SSIM, extraídos de publicações na literatura, conforme fornecidos pelos respectivos autores. A avaliação foi realizada na base de dados Vid4, pois é a base comum em que os métodos foram avaliados na literatura. Os campos vazios indicam a ausência de avaliações para a base de dados Vid4 nos estudos mencionados.

Tabela 6 – Valores médios das métricas PSNR e SSIM para a base de dados Vid4.

| **Escala** | **Métricas** | **Métodos de VSR** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **Bicubic** | **SRCNN** | **VSRnet** | **VESPCN** | **DRDVSR** | **VSR-DUF** | **EDVR** |
| 2× | PSNR | 28,43 | 30,70 | 31,30 | - | - | **33,73** | - |
| | SSIM | 0,8676 | 0,9172 | 0,9278 | - | - | **0,9554** | - |
| 3× | PSNR | 25,28 | 26,51 | 26,79 | 27,25 | 27,49 | **28,9** | - |
| | SSIM | 0,7329 | 0,7933 | 0,8098 | 0,8447 | 0,84 | **0,8898** | - |
| 4× | PSNR | 23,79 | 24,69 | 24,84 | 25,35 | 25,52 | 27,34 | **27,35** |
| | SSIM | 0,6332 | 0,6918 | 0,7049 | 0,7557 | 0,76 | **0,8327** | 0,8264 |

Fonte: De autoria própria a partir de dados dos autores Kappeler et al.; Caballero et al.; Tao et al.; Jo et al.; Wang et al.[18, 120, 122, 126, 28].

Como pode ser observado na Tabela 6, destacado em negrito, os modelos VSR-DUF e EDVR apresentaram melhor desempenho, sendo considerados o estado da arte na literatura para tarefas de super-resolução de vídeo[2]. O método EDVR recebe maior destaque por ter vencido a NTIRE19, uma competição de relevância na comunidade de visão computacional [140]. O EDVR superou o segundo lugar por uma grande margem em todas as quatro faixas de desafios de restauração e aprimoramento de vídeo, incluindo super-resolução. A base de dados utilizada na competição foi a REDS [139].

## 4.7 CONSIDERAÇÕES

Neste capítulo, foi apresentado o estado da arte da super-resolução de vídeo, também conhecida como super-resolução multiquadros. Os principais métodos destacados na literatura foram apresentados em ordem cronológica, fornecendo uma síntese da evolução dos métodos que utilizam redes neurais profundas.

No final, foi realizado um comparativo do desempenho desses métodos usando as métricas PSNR e SSIM, com base em informações obtidas da literatura. Esses métodos

[2] https://paperswithcode.com/sota/video-super-resolution-on-vid4-4x-upscaling

foram propostos pela comunidade de visão computacional com o objetivo de melhorar a qualidade das imagens restauradas, sem uma aplicação específica em um domínio particular.

Esses métodos exploram a profundidade das redes neurais, sem se preocupar tanto com a complexidade dos recursos computacionais necessários para sua utilização prática. Neste estudo, a aplicação da super-resolução foi realizada em aplicações de *streaming* de vídeo sob demanda e ao vivo.

Os métodos escolhidos para este estudo foram selecionados de forma a explorar o estado da arte em super-resolução, buscando tanto a qualidade das imagens quanto a viabilidade de sua utilização em aplicações reais de *streaming* de vídeo.

# CAPÍTULO 5

# Uma Abordagem para Reduzir o Tráfego de Streaming de Vídeo na Nuvem

Os desafios relacionados a escala global da audiência das plataformas de video têm sido abordados com um conjunto grande de tecnologias. Nesse conjunto está a Rede de distribuição de conteúdo, com seus algoritmos de posicionamento e replicação de conteúdo; a Computação em nuvem, para tratar da flutuação das demandas por processamento, transmissão e armazenamento; e as Técnicas de codificação do vídeo, para a produção de conteúdo adequado a fragmentação da audiência por diversas plataformas de software e hardware.

Uma audiência global requer que provedores de conteúdos em vídeos repliquem esses conteúdos, a partir de seus *data centers*, para servidores localizados próximos de sua audiência [141, 142]. Tais servidores interconectados compõem redes de distribuição de conteúdos (CDNs, do inglês *content delivery network*s) com servidores substitutos localizados em vários países. A forma mais eficiente de implementar essas redes é com o uso de técnicas de computação em nuvem e na borda. Essas técnicas permitem e facilitam o gerenciamento de recursos de processamento, transmissão e armazenamento, racionalizando a disponibilização e o uso desses recursos.

As principais plataformas de vídeos sob demanda, como Netflix, Youtube, Prime Video, entre outras, adotam o *streaming* adaptativo baseado em HTTP (HAS, do inglês *HTTP-based adaptive streaming*) para aprimorar a experiência de visualização de seus usuários. Essa estratégia consiste em fornecer conteúdo codificado com diferentes taxas de bits de forma dinâmica durante a reprodução, permitindo lidar com o dilema entre

a demanda de recursos de transmissão, especialmente em redes de acesso, e a qualidade das imagens exibidas. Esse enfoque evita problemas como interrupções na reprodução, conhecidos como *rebuffering*, causados por descompasso entre o tempo de chegada do segmento de vídeo e o tempo de reprodução.

No contexto do HAS, cada vídeo é transcodificado em múltiplas taxas de bits, resultando em várias representações do mesmo conteúdo. Essas representações são replicadas para servidores distribuídos em uma CDN, o que gera um tráfego significativo capaz de sobrecarregar as infraestruturas da Internet. Esse aumento na demanda de tráfego pode impactar negativamente a qualidade de serviço oferecida aos usuários [141, 142] e representar um desafio para as operadoras de rede [1, 10].

Além disso, é importante ressaltar que os custos de tráfego de dados internacional, fornecido por provedores de serviços de internet de longa distância, conhecidos como *internet service provider*s (ISPs) *tier-1* , tendem a ser mais elevados em comparação com o tráfego regional e local, fornecido por ISPs de níveis inferiores, como ISPs *tier-2* e *tier-3* [11, 143]. Portanto, reduzir o volume de dados transmitidos nas infraestruturas dos provedores *tier-1* representa uma alternativa viável para lidar com os custos associados às operações em escala global.

Neste capítulo, apresenta-se o resultado de um estudo que corresponde à primeira etapa desta pesquisa. A proposta consistiu na criação e avaliação de um serviço de replicação de vídeo em nuvem utilizando um modelo de restauração por super-resolução baseado em GAN. Esse serviço é implementado por meio de um *framework* de distribuição de vídeo em nuvem, no qual apenas uma versão em baixa resolução de cada vídeo é replicada. Nos servidores substitutos, utiliza-se um modelo de SR baseado em redes neurais para restaurar o vídeo para alta definição. Esse processo tem como objetivo reduzir o volume de dados transmitidos e, consequentemente, os custos associados à movimentação de dados entre servidores.

O *framework* proposto tem como objetivo principal reduzir o tráfego de dados de vídeo nas infraestruturas de rede dos ISPs *tier-1*. No entanto, uma contrapartida desse método é o aumento da demanda por processamento nos servidores em nuvem, especialmente em tarefas que podem ser paralelizadas utilizando GPUs. Uma das motivações para essa abordagem é que o poder computacional nas nuvens de borda tem aumentado, resultando em um maior volume de recursos de computação ociosos nesses servidores [12]. Além disso, nos últimos anos, o desempenho das GPUs para servidores tem crescido significativamente [144], o que tem levado a uma redução nos custos de processamento de serviços em nuvem [13, 14]. Por exemplo, no início de 2020, o Google Cloud reduziu os preços de GPUs em mais de 60% [15]. Esses fatores indicam a viabilidade da abordagem proposta, considerando a demanda computacional envolvida na solução.

## 5.1 O FRAMEWORK DE REPLICAÇÃO DE CONTEÚDO DE VÍDEO EM NUVEM

Serviços de *streaming* de vídeo, *e.g.*, Netflix, Youtube e Prime Video, usam CDNs para aproximar os conteúdos da audiência e, assim, melhorar a QoE das sessões de vídeo, reduzindo o atraso e aumentando a taxa de transferência. As CDNs são implementadas usando uma das seguintes técnicas de posicionamento de servidores: entrar profundo (*enter deep*) e trazer para casa (*bring home*) [145]. A primeira técnica penetra nas redes de acesso dos ISPs, implantando grupos de servidores de distribuição de conteúdo nos pontos de presença (PoPs, do inglês *point of presence*s) dos ISPs. A segunda técnica aproxima os ISPs aos agrupamentos de servidores da CDN. Nesse caso, cria-se grandes centros de distribuição de conteúdo, em algumas áreas estratégicas, conectando-os aos ISPs por meio de conexões privadas de alta velocidade.

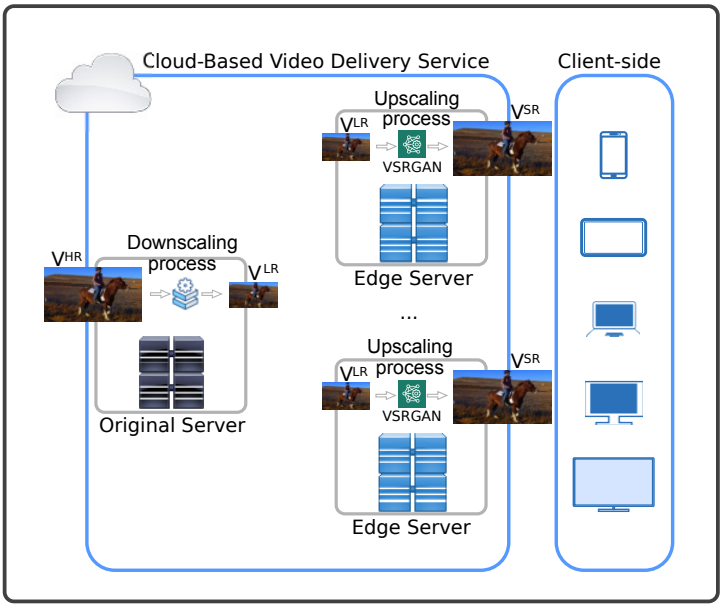
Devido à segmentação de audiência dos serviços de *streaming* de vídeo, as políticas de replicação de conteúdo nas CDNs exigem a movimentação de grandes quantidades de dados dos *data centers* dos provedores de conteúdos para os servidores substitutos, que são os pontos de acesso mais próximos à audiência. Infraestrutura como serviço (IaaS, do inglês *infrastructure as a service*), um dos modelos de serviço de computação em nuvem, tem sido usado para lidar com as flutuações naturais nas demandas por recursos de processamento e de transmissão. Apesar das vantagens do modelo IaaS oferecer recursos sob demanda e pagamento conforme o uso, existem o custos associados à movimentação dos grandes volumes de conteúdos pela Internet, por exemplo, o congestionamento dos links privados e o alto custo para atualização desse tipo de link, principalmente quando exige-se instalação de cabos de fibra óptica de longas distância [10, 11, 143].

Neste capítulo, foi proposto um *framework* de replicação de conteúdo com o objetivo de reduzir a quantidade de dados transferidos entre os *data centers* dos provedores de conteúdo e os servidores substitutos nas CDNs. Esse *framework* é capaz de oferecer suporte aos serviços de *streaming* de vídeo sob demanda (VoD) usando tanto arquiteturas CDN de tipo *enter-deep* quanto *bring-home*. A Figura 21 ilustra o *framework* de replicação de conteúdo proposto.

Neste *framework*, o procedimento de super-resolução começa depois que a política de replicação de conteúdo estabelece qual vídeo $V$ deve ser armazenado no servidor substituto $S$. A versão de alta definição $V^{HR}$ do vídeo $V$ selecionado, é armazenada no servidor original e codificada para produzir sua representação de baixa definição $V^{LR}$. Esse processo de *downscaling* reduz a resolução do vídeo original em um fator de $r$.

A representação de baixa resolução é replicada para os servidores substitutos, em contraste com a abordagem de replicar todas as versões. Uma DNN executa um *upscaling* de fator $r$ para aumentar essa representação nos servidores substitutos. A saída desse procedimento é fornecida como sessões de vídeo de taxa de bits única ou está envolvida no

Figura 21 – Arquitetura do framework de replicação de conteúdo de vídeo em nuvem




Fonte: De autoria própria.

fluxo de trabalho de publicação de taxa de bits múltiplas.

Desenhamos este *framework* para serviços de *streaming* de VoD, e assim as tarefas de SR são executadas *offline* e com base em um escalonamento pré-estabelecido. Além disso, assumimos que o servidor substituto possui os recursos de computação para executar as tarefas de SR usando um modelo de processamento paralelo em GPUs.

## 5.2 O PROBLEMA DE OTIMIZAÇÃO DO TAMANHO DO VÍDEO

O engajamento da audiência é fundamental para viabilizar os serviços de vídeo *streamming* em escala global. Nesse contexto, tais serviços precisam lidar com a fragmentação de suas audiências. As estratégias adotadas para fomentar o engajamento pressupõem mover e armazenar uma grande quantidade de dados em servidores substitutos próximos às suas audiências, com objetivo produzir sessões de vídeo com elevada QoE, ou seja, atraso reduzido e fluxos compatíveis com os *devices* das audiências. Nesse cenário, o custo de mover o conteúdo em vídeo é fortemente afetado pelo tamanho dos vídeos codificados. Assim, o desafio está em resolver o impasse entre diminuir o tamanho da mídia a ser movida, consequentemente, diminuindo os custos operacionais com a movimentação, e manter a qualidade visual em alto padrão.

A seguir, apresenta-se a formulação do problema, que consiste em minimizar o tamanho do vídeo. Esse problema envolve o *quantization parameter* (QP), que varia de 0 a 51, e a resolução do vídeo, sujeitos a um limite de qualidade estabelecido, conforme definido na Equação (5.1).

$$\begin{aligned} &\min_{q,p} \sum_{i=1}^{K} Size_{V_i^{LR}}(q,p) \\ &\text{s.t. } VQ\left(DNN(V^{LR})\right) - VQ\left(V^{HR}\right) \leq VQ_T \\ &\forall i, q \in \{0, \cdots, 51\} \\ &\forall i, p \leq V_T^{LR}, \end{aligned} \tag{5.1}$$

onde $Size_{V_i^{LR}}$ é o tamanho do vídeo de baixa resolução $i$ em bytes, $K$ é o número total de vídeos envolvidos na operação. A qualidade dos vídeos super resolvidos baseados em DNN é representada por $VQ(DNN(V^{LR}))$, a qualidade dos vídeos originais é representada por $VQ(V^{HR})$, e o limite de degradação de qualidade desejado é representado por $VQ_T$. $q$ e $p$ são o QP e a resolução dos vídeos de baixa qualidade, respectivamente. $V_T^{LR}$ é o limite da resolução dos vídeos de baixa qualidade.

**Abordagem**: O modelo de SR baseado em DNN é treinado para restaurar $V^{LR}$ com um limite de degradação determinado pela compressão e resolução, estabelecendo uma restrição de qualidade para a saída $V^{SR}$ em comparação com o vídeo de referência $V^{HR}$. No entanto, encontrar a solução ótima é inviável, uma vez que alterações significativas nos níveis de compressão exigem o retratamento do modelo de DNN, um processo computacionalmente complexo. Para lidar com esse problema, adotamos a abordagem de utilizar o JND como limite para $VQ_T$, com base em estudos apresentados por Wang et al.[71] em um extenso conjunto de dados de vídeo.

## 5.3 A SUPER-RESOLUÇÃO DE VÍDEO COM USO DE GAN

Neste capítulo, foi avaliada a técnica de super-resolução de vídeo como solução para a entrega de conteúdo de alta qualidade em um serviço de vídeo baseado em nuvem. Para lidar com o congestionamento da rede de ponta a ponta, um modelo de super-resolução foi utilizado como alternativa aos métodos convencionais. Esse modelo, é uma rede adversária generativa de super-resolução, que é um tipo de aprendizado profundo capaz de reconstruir vídeos em alta definição a partir de suas versões em baixa definição, preservando a nitidez e a realidade das imagens [19, 22].

O modelo proposto foi chamado de Super-Resolução de Vídeo Aprimorada com GAN (VSRGAN+, do inglês *improved video super-resolution with GAN*), e é apresentado nas subseções a seguir.

### 5.3.1 ARQUITETURA DO VSRGAN+

A arquitetura da rede adversária inclui uma rede geradora $G(f^{LR})$ e uma rede discriminadora $D(G(f^{LR}))$, que competem entre si durante o treinamento. $G(.)$ aprende como gerar quadros $f^{SR}$, buscando ficar indistinguível do quadros reais $f^{HR}$, esperando passar despercebidos por $D(.)$. $D(.)$ aprende como distinguir os quadros gerados dos

quadros reais. Em outras palavras, o treinamento adversário trabalha para equilibrar essas duas dinâmicas.

Em Goodfellow et al.[107], os autores classificaram as redes adversárias como um problema de mínimo-máximo. Neste trabalho, utilizou-se essa classificação para definir uma rede adversária para super-resolução de vídeo da seguinte forma:

$$\min_{\theta_G} \max_{\theta_D} V(D_{\theta_D}, G_{\theta_G}) = \begin{aligned} & f^{HR} \sim p_{train}(f^{HR}) \left[\log D_{\theta_D}(f^{HR})\right] + \\ & f^{LR} \sim p_G(f^{LR}) \left[\log\left(1 - D_{\theta_D}(G_{\theta_G}(f^{LR}))\right)\right], \end{aligned} \tag{5.2}$$

onde treinamos a rede discriminadora $D_{\theta_D}$ para maximizar a probabilidade de seus resultados classificando corretamente os dois quadros: o real e o super-resolvido. A rede geradora $G_{\theta_G}$ aprende como gerar quadros mais realistas $f^{SR}$ ajustando os parâmetros $\theta_G$ para minimizar $\log\left(1 - D_{\theta_D}(G_{\theta_G}(f^{LR}))\right)$, onde $\theta_G = \{W_{1:L}; b_{1:L}\}$. $W$ são os pesos e $b$ é o viés da rede neural da camada $L$ que é otimizada por uma função de perda $L_G$ durante o treinamento.

Durante as sessões de treinamento da rede geradora, utilizou-se um conjunto de quadros, $f_i^{LR}$, codificados em baixa taxa de bits, e suas contrapartes $f_i^{HR}$, sujeitos à minimização da seguinte função de perda.

$$\theta_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{i=1}^{N} L_G\left(G_{\theta_G}(f_i^{LR}), f_i^{HR}\right) \tag{5.3}$$

onde $L_G$ é a função genérica que calcula a perda entre o quadro super-resolvido e o quadro real; na seção 5.3.2, esta função de perda é apresentada em detalhes.

A rede geradora é composta por diferentes blocos que desempenham funções específicas. A parte 1 consiste em um bloco pré-residual, que contém uma camada de convolução com 64 filtros de tamanho $9 \times 9$ e uma função de ativação PReLU [77].

A parte 2 é o núcleo da rede geradora, que contém vários blocos residuais densos em cascata (RRDB) com conexões densas de salto. Cada bloco RRDB é composto por cinco camadas de convolução, utilizando a configuração $k3n64s1$, seguida pela ativação LeakyReLU [108]. Após cada bloco RRDB, é adicionada uma escala residual $\beta$ [98]. Além disso, há uma conexão de salto do bloco pré-residual e uma camada de convolução com a configuração $k3n64s1$ antes dos blocos RRDB.

Para aumentar a resolução, a parte 3 de $G(.)$ possui $\log_2(r)$ blocos, onde $r$ é o fator de escala, que é um múltiplo de 2. Cada bloco consiste em uma camada de convolução com a configuração $k3n256s1$, seguida por uma camada de subpixel e ativação PReLU [87].

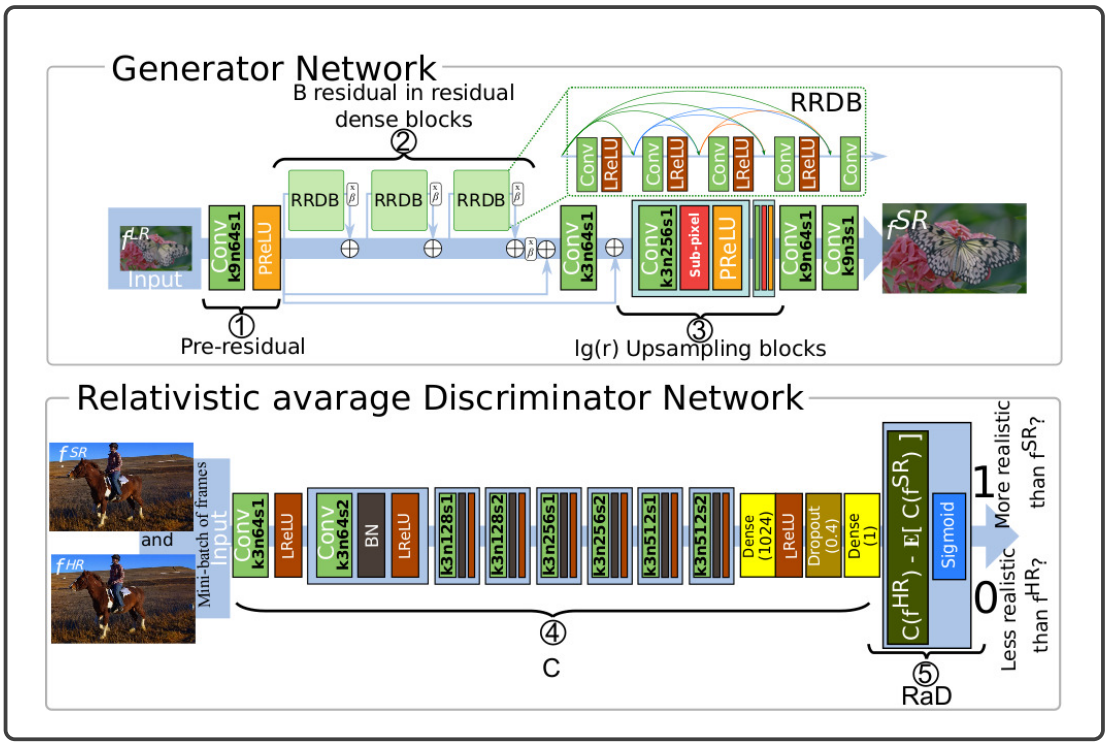
Finalmente, a última parte de $G(.)$ é composta por duas camadas de convolução: uma com a configuração $k9n64s1$ e outra com a configuração $k9n3s1$. Essas camadas

ajudam a refinar ainda mais a saída da rede geradora antes de produzir a imagem super-resolvida.

O discriminador tem duas partes, $C$ (parte 4) e o discriminador médio relativístico (RaD, do inglês *relativistic average discriminator*) (parte 5). $C$ consiste na convolução $k3n64s1$, seguida por uma ativação LeakyReLU com $\alpha = 0.2$. O núcleo de $C$ possui sete blocos compostos por uma camada de convolução, uma camada BN e ativação LeakyReLU com $\alpha = 0,2$. A camada de convolução do primeiro bloco consiste em $k3n64s2$. Os outros blocos conduzem a convolução com filtros de três tamanhos; o número de filtros varia para cada par de blocos em 128, 256 e 512, com o primeiro par tendo um *step* e o segundo tendo dois *steps*. A parte final de $C$ inclui uma camada densa com 1024 neurônios, ativação LeakyReLU, *dropout* de 40% e uma camada densa com apenas um neurônio.

Geralmente, em abordagens convencionais de super-resolução, é comum utilizar um discriminador padrão, denotado por $D$, para calcular a probabilidade de um quadro ser realista. Esse discriminador é definido como $D(f^{SR}) = \sigma(C(f^{SR}))$ para todos os quadros super-resolvidos $f^{SR}$, e $D(f^{HR}) = \sigma(C(f^{HR}))$ para todos os quadros em alta resolução $f^{HR}$, onde $C$ representa a saída do discriminador antes da ativação $\sigma$.

Figura 22 – Arquitetura das redes geradora e discriminador relativista baseado em média.




Fonte: De autoria própria.

No entanto, neste trabalho, adotou-se uma abordagem diferente ao utilizar um discriminador médio relativístico (RaD, do inglês *relativistic average discriminator*) [112]. Essa abordagem propõe uma modificação no discriminador convencional, introduzindo

uma camada adicional para medir a diferença entre as probabilidades atribuídas a dados reais e dados gerados pelo modelo. Essa diferença é chamada de pontuação relativística.

O discriminador médio relativístico considera a relação relativa entre os dados reais e gerados, o que leva a uma melhor estimação da qualidade e realismo dos quadros super-resolvidos. Essa abordagem ajuda a melhorar a performance do modelo e aprimorar a qualidade visual dos resultados finais.

Portanto, neste trabalho, optou-se por utilizar o discriminador médio relativístico (RaD) em vez do discriminador convencional para aprimorar a avaliação da realidade dos quadros super-resolvidos e obter melhores resultados no processo de super-resolução.

A Figura 22 (parte 5) ilustra a aplicação do discriminador médio relativístico. A função RaD utiliza as informações do componente $C$, ou seja, $RaD(f^{HR}, f^{SR}) = \sigma(C(f^{HR}) - \mathbb{E}[C(f^{SR})])$ para todos os quadros em alta resolução $f^{HR}$, e $RaD(f^{SR}, f^{HR}) = \sigma(C(f^{SR}) - \mathbb{E}[C(f^{HR})])$ para todos os quadros super-resolvidos $f^{SR}$. Aqui, $\mathbb{E}[.]$ representa a média da saída de $C$ para todos os quadros em um mini-lote.

Em termos simples, o RaD calcula a probabilidade de que o quadro em alta resolução $f^{HR}$ seja relativamente mais realista, em média, em comparação com uma amostra aleatória de quadros super-resolvidos $f^{SR}$, e vice-versa. Essa abordagem permite que a rede geradora aprenda a super-resolver imagens de forma mais nítida, pois considera a relação relativa entre os quadros em alta resolução e os quadros super-resolvidos.

A utilização do RaD auxilia no treinamento da rede geradora, incentivando-a a produzir resultados super-resolvidos que sejam mais realistas e de maior qualidade visual. Essa técnica contribui para melhorar a nitidez e a percepção de detalhes nas imagens super-resolvidas, conforme mencionado em [22].

### 5.3.2 FUNÇÃO DE ERRO PERCEPTIVA

Em Zhao et al.[109], os autores propuseram funções de erro baseadas em média, essas funções geram imagens de boa qualidade quando avaliadas por métricas pixel a pixel, como PSNR e SSIM. No entanto, o olho humano detecta rapidamente os artefatos de suavização dessas imagens. Em outras palavras, as funções de erro baseadas na média são insuficientes para capturar a percepção visual humana, o que inspirou um novo corpo de trabalhos sobre funções de perda orientadas à percepção; veja [103, 104, 106, 21, 22, 23, 45, 46], no entanto, nenhum deles aplicados a modelos de SR para aplicações de *streaming* de vídeo.

Em Wang et al.[22], os autores propuseram uma função de erro perceptiva que possui três componentes: (i) o componente perceptivo, (ii) o componente adversário e (iii) o componente de conteúdo; veja a Equação (5.4).

$$L_G = L_{percept} + \lambda L_G^{RaD} + \eta L_1 \tag{5.4}$$

O componente de erro perceptivo $L_{percept}$ é calculado pela função *mean squared error* (MSE) no espaço de características de quadro $F^{SR}$ e seu correspondente $F^{HR}$, extraídos pela rede VGG19 [53] pré-treinada para classificação de imagens, ao invés de ser calculada pixel a pixel de $F^{SR}$ e $F^{HR}$ como nas abordagens clássicas com as funções MSE ou MAE aplicadas diretamente no espaço de pixels.

A extração de características é realizada no $5^o$ bloco da rede VGG19, antes da função de ativação da $4^a$ camada de convolução, daí chamada característica $VGG_{54}$. Os coeficientes $\lambda$ e $\eta$ são utilizados como pesos para balancear a influência dos componente da função. Na Equação (5.5) encontra-se o componente de erro perceptivo,

$$\begin{aligned} L_{percept} =& \frac{1}{W.H} \sum_{x=1}^{W} \sum_{y=1}^{H} \\ & \left( VGG_{54}(F^{HR})_{x,y} - VGG_{54}\left(F^{SR}\right)_{x,y} \right)^2, \end{aligned} \tag{5.5}$$

onde $W$ e $H$ são as dimensões das características conhecidas como $VGG_{54}$ e representa características e similaridades de alto nível. A camada $VGG_{54}$ foi utilizada por apresentar melhores resultados perceptuais em avaliações anteriores [21, 22].

Na Equação (6.4) encontra-se o componente de erro adversário $L_G^{RaD}$ que é calculada baseado na função de entropia cruzada,

$$\begin{aligned} L_G^{RaD} =& - \mathbb{E}_{x_r} \left[ \log \left( 1 - D_{RaD}(x_r, x_f) \right) \right] \\ & - \mathbb{E}_{x_f} \left[ \log \left( D_{Ra}(x_f, x_r) \right) \right], \end{aligned} \tag{5.6}$$

onde $x_r = F^{HR}$ e $x_f = G(F^{LR})$. A Equação (5.7) apresenta o componente $L_1$,

$$L_1 = \frac{1}{WH} \sum_{x=1}^{H} \sum_{y=1}^{W} \left| F_{x,y}^{HR} - G(F^{LR})_{x,y} \right|, \tag{5.7}$$

sendo $W$ e $H$ dimensões de $F^{HR}$.

## 5.4 BASES DE DADOS UTILIZADAS

Dois conjuntos de dados foram utilizados para treinamento e teste dos modelos avaliados neste trabalho. O primeiro é um conjunto de dados de imagem de alta definição usado para treinar os modelos. O segundo é um conjunto de dados de vídeo usado para testar os modelos. Devido à diversidade de cenários, o primeiro conjunto de dados demonstrou excelente ajuste para o treinamento dos modelos. Ele Inclui 1,8 milhão de imagens de 365 categorias no conjunto de treinamento; e 36.500 imagens, 100 por categoria, no conjunto de validação. Em Zhou et al.[146], os autores apresentaram e nomearam este conjunto de dados como Places365-Standard.

O conjunto de dados de vídeos tem 220 videoclipes de cinco segundos, cada um com quatro resoluções: 1080p, 720p, 540p e 360p. Esta gama de resoluções de vídeo visa a

prevalência de 1080p e 720p em aplicativos de *streaming* de vídeo para telas largas, por exemplo, *smart* TVs e *laptops*, e a prevalência de 540p e 360p em aplicativos de *streaming* de vídeo para telas pequenas, por exemplo, *smartphones* e *tablets*. Este conjunto de dados foi apresentado em Wang et al.[71] e é chamado de VideoSet.

Todos os vídeos foram codificados com o codec H.264/AVC no espaço de cores YCbCr4:2:0 e $QP = \{x \in \mathbb{N} \mid 0 \leq x \leq 51\}$. $QP = 0$ indica que os vídeos são codificados sem perdas e $QP = 51$ significa que os vídeos têm a maior taxa de compressão, que mostra a maior (menor) taxa de bits e melhor (pior) qualidade de imagem por quadro, respectivamente.

Tabela 7 – Resumo dos cenas que compõem a base VideoSet.

| **Vídeo de Origem** | **Quantidade de Cenas** | **FPS** |
|---|---|---|
| El Fuente | 31 | 30 |
| Chimera | 59 | 30 |
| Ancient Thought | 11 | 24 |
| Eldorado | 14 | 24 |
| Indoor Soccer | 5 | 24 |
| Life Untouched | 15 | 30 |
| Lifting Off | 13 | 24 |
| Moment of Intensity | 10 | 30 |
| Skateboarding | 9 | 24 |
| Unspoken Friend | 13 | 24 |
| Tears of Steel | 40 | 24 |

Fonte: De autoria própria a partir de dados dos autores.

Em Wang et al.[71], 30 indivíduos, em média, avaliaram a qualidade de todos os videoclipes da base de dados VideoSet. Cada indivíduo assistiu aos vídeos codificados e identificou três pontos JND, dividindo os vídeos codificados em quatro conjuntos de qualidade, $Q_1$ apresentando a melhor qualidade percebida e $Q_4$ a pior qualidade. Em $Q_i$ estão definidos os intervalos que classificam as qualidades com relação aos pontos JND. Os vídeos avaliados variam de $QP = \{x \in \mathbb{N} \mid 7 \leq x \leq 47\}$; pois, vídeos codificados com $QP = \{x \in \mathbb{N} \mid 0 \leq x \leq 6\}$ têm alterações de qualidade não percebida na visão humana, e aqueles com $QP = \{x \in \mathbb{N} \mid 48 \leq x \leq 51\}$ têm qualidade inaceitável.

$$Q_i = \begin{cases} Q_1 & \text{se } QP_{V_i} < QP_{1^o JND_{V_i}} \\ Q_2 & \text{se } QP_{1^o JND_{V_i}} \leq QP_{V_i} < QP_{2^o JND_{V_i}} \\ Q_3 & \text{se } QP_{2^o JND_{V_i}} \leq QP_{V_i} < QP_{3^o JND_{V_i}} \\ Q_4 & \text{se } QP_{V_i} \geq QP_{3^o JND_{V_i}} \end{cases}$$

A base de dados VideoSet compreende clipes de cinco segundos amostrados de

vídeos de vários assuntos. A Tabela 7 mostra os títulos dos vídeos, o número de amostras e o número de *frames per second* (FPS) de cada amostra. Todos os vídeos originais foram provindos de base de dados públicas [147, 148].

A conclusão geral é que o conteúdo e as qualidades do VideoSet são amostras representativas do conteúdo disponível nos serviços de vídeo atuais. Por exemplo, o título *Tears of Steel* é um filme de ação semelhante a um desenho animado, *El Fuente* é uma série dramática semelhante à TV e *Unspoken Friend* é um filme de 90 minutos. Todo esse conteúdo é codificado para segmentar audiências multitelas.

## 5.5 RESULTADOS EXPERIMENTAIS

Esta seção apresenta os resultados de experimentos numéricos que avaliaram a efetividade do *framework* proposto. A seção 5.5.1 mostra os parâmetros e detalhes de treinamento dos modelos de SR. Depois disso, avaliou-se a qualidade do vídeo usando métricas pixel a pixel e perceptivas. Na seção 5.5.3, são apresentadas evidências de que o sistema visual humano é capaz de perceber apenas distorções significativas quando há introdução de ruídos em uma imagem. Ademais, foi constatado que o sistema visual humano percebe os vídeos super-resolvidos e suas versões originais em alta definição como semelhantes.

Na seção 5.5.4, foram analisados os compromissos entre o tempo de processamento e a qualidade do vídeo. Por fim, examinamos como a super-resolução de vídeo pode melhorar as políticas de replicação de conteúdo implementadas por serviços de vídeo baseados em nuvem, que são influenciadas pela configuração da taxa e resolução de codificação de vídeo.

### 5.5.1 DETALHES DO TREINAMENTO E PARÂMETROS DOS MODELOS DE SR

Todos os modelos foram treinados em máquinas equipadas com GPU NVIDIA GeForce GTX 1080Ti-11GB, CPU i7-7700 com *clock* de 3.60GHz e 62GB de RAM, utilizando a base de dados Places365-Standard [146], que é descrita na seção 5.4. Já nos testes, utilizou-se uma máquina com GPU NVIDIA GeForce GTX 1070Ti-8GB, CPU i7-8700 com *clock* de 3.20GHz e 32GB de RAM.

Os modelos foram desenvolvidos utilizando a plataforma TensorFlow[1], com a biblioteca Keras[2] como *backend*. Os *baselines* foram treinados seguindo a metodologia definida por seus respectivos autores. Na Tabela 8, são apresentados de forma resumida os principais parâmetros definidos em cada modelo.

O modelo de SR proposto, denominado *improved video super-resolution with GAN* (VSRGAN+), foi treinado utilizando lotes de imagens com tamanho 16 e fator de escala de

[1] https://www.tensorflow.org
[2] https://keras.io

Tabela 8 – Configuração dos modelos

| **Modelos** | **Configuração** |
|---|---|
| SRCNN | Filtros = 64, 32, 3 em cada camada, respectivamente<br>Tamanho de filtro = 9, 1, 5 em cada camada, respectivamente<br>Otimizador: SGD c/ taxa de aprendizagem $10^{-4}$<br>Tamanho de lotes: 128<br>HR subimagens: $33 \times 33$ pixels<br>Função de erro: $L_2$<br>Número de iterações = $8 \times 10^7$ |
| ESPCN | Filtros = 64, 32, $r^2 \times 3$ em cada camada, respectivamente<br>Tamanho de filtro = 5, 3, 3 em cada camada, respectivamente<br>Otimizador: Adam c/ taxa de aprendizagem $10^{-4}$<br>Tamanho de lotes: 128<br>HR subimagens: $34 \times 34$ pixels<br>Função de erro: $L_2$<br>Número de iterações = $8 \times 10^7$ |
| CISRDCNN | Bloco DBCNN: $K_1 - 1$ CNN com 64 filtros<br>de tamanho $3 \times 3$ +BN+ReLU, $K_1$-th camada usa 3 filtros<br>de tamanho $3 \times 3$, e aprendizagem residual<br>Bloco USCNN: $K_2 - 1$ CNN com 64 filtros<br>de tamanho $3 \times 3$ +BN+ReLU, $K_2$-th é uma camada<br>deconvolutional com 3 filtros de tamanho $9 \times 9$<br>Bloco QECNN é similar ao DBCNN<br>Função de erro: $L_2$<br>$K_1 = 20$, $K_2 = 10$, $K_3 = 10$, e $QF = 20$ |
| SRResNet | Blocos Residuais: B=16<br>Otimizador: Adam c/ taxa de aprendizagem $10^{-4}$<br>Tamanho de lotes: 16<br>HR subimagens: $96 \times 96$ pixels<br>Função de erro: $L_2$<br>Número de iterações = $10^6$ |
| SRGAN | Blocos Residuais: B=16<br>Otimizador: Adam c/ taxa de aprendizagem $10^{-4}$ /<br>taxa de aprendizagem $10^{-5}$<br>Tamanho de lotes: 16<br>HR subimagens: $96 \times 96$ pixels<br>Função de erro: Erro perceptiva + Erro adversário<br>Número de iterações = $10^5$ / $10^5$ |
| VSRGAN+ | Blocos residuais em residuais: B=3<br>Otimizador: Adam c/ taxa de aprendizagem $2 \times 10^{-4}$/<br>taxa de aprendizagem $10^{-4}$<br>Tamanho de lotes: 16<br>HR subimagens: $128 \times 128$ pixels<br>Função de erro: $L_1$/ $L_G + L_G^{RaD}$<br>$\beta = 0,2$<br>$\lambda = 5 \times 10^{-3}$<br>$\eta = 10^{-2}$<br>Número de iterações = $10^6$ / $5 \times 10^5$ |

Fonte: De autoria própria a partir de dados dos autores.

2× entre imagens LR e imagens HR. De cada imagem HR, foram aleatoriamente extraídas sub-imagens de tamanho $128 \times 128$. Em seguida, a redução de 2× foi aplicada a cada sub-imagem utilizando interpolação bicúbica e *GaussianBlur* com a biblioteca OpenCV[3].

O processo de treinamento ocorreu em duas etapas: *i*) treinou-se por $10^6$ iterações somente a parte geradora com a função de erro $L_1$ (ver Equação (5.7)), taxa de aprendizagem inicial de $2 \times 10^{-4}$ e decaimento com fator de $5 \times 10^{-1}$ a cada $2 \times 10^5$ iterações ou se ocorresse 50 iterações sem redução no erro de validação; *ii*) treinou-se por $5 \times 10^5$ iterações a GAN, parte geradora e discriminadora, com a função de erro $L_G$ (ver Equação (5.4)), com $\lambda = 5 \times 10^{-3}$ e $\eta = 10^{-2}$, taxa de aprendizagem inicial de $10^{-4}$ e decaimento de um

[3] https://pypi.org/project/opencv-python/

fator de $5 \times 10^{-1}$ a cada $5 \times 10^4$, $1 \times 10^5$, $2 \times 10^5$, $3 \times 10^5$ iterações. A parte geradora foi inicializada com os pesos do modelo treinado na primeira etapa.

### 5.5.2 AVALIAÇÃO DE QUALIDADE DOS VÍDEOS

Após o treinamento dos modelos, realizou-se testes de super-resolução de vídeos aplicando o modelo proposto (VSRGAN+) e os seguintes modelos *baselines*: SRCNN [79], ESPCN [87], SRResNet [21] e SRGAN [21].

Primeiramente, utilizou-se como entrada para cada modelo os 220 vídeos da base VideoSet [71], com resolução de 360p e sem compressão, ou seja, sem perda. Como resultado, foram obtidos vídeos com resolução de 720p, uma vez que os modelos foram treinados para uma escala de 2×. Em seguida, a mesma atividade foi realizada utilizando como entrada vídeos com resolução de 540p e sem compressão, obtendo como saída vídeos de 1080p.

Para avaliar a qualidade dos vídeos gerados pelos modelos, foram empregadas as métricas PSNR, LPIPS e VMAF. Cada vídeo gerado foi comparado aos vídeos de referência da base VideoSet [71] nas resoluções correspondentes de 720p e 1080p, com compressão sem perdas. Os resultados numéricos estão apresentados na Tabela 9, e os resultados gráficos podem ser observados na Figura 23. Utilizou-se um intervalo de confiança de 95% baseado na distribuição normal.

Tabela 9 – Valores da avaliação de qualidade dos vídeos

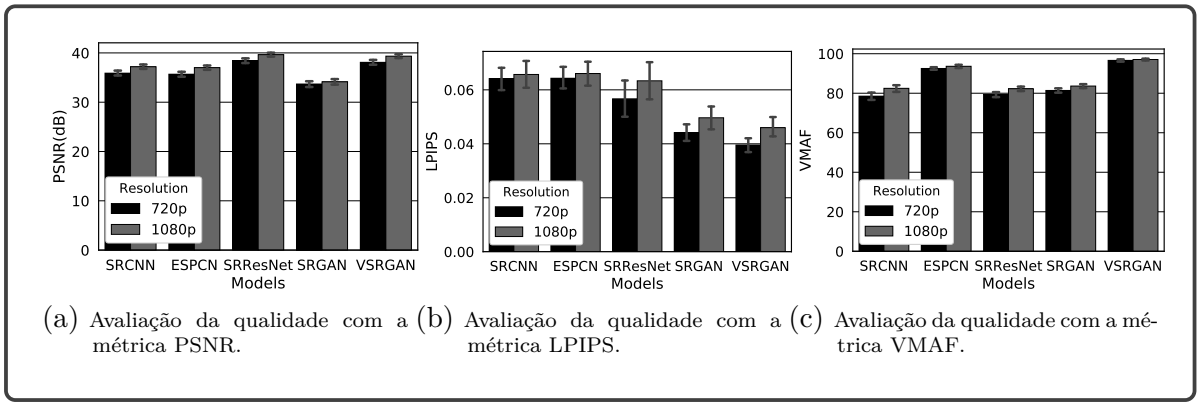
| **Métricas** | **Resolução** | **SRCNN** | **ESPCN** | **SRResNet** | **SRGAN** | **VSRGAN** |
|---|---|---|---|---|---|---|
| PSNR | 720p | $35,89(\pm 0,51)$ | $35,68(\pm 0,50)$ | $\mathbf{38,44}(\pm 0,47)$ | $33,69(\pm 0,56)$ | $\mathbf{38,09}(\pm 0,49)$ |
| | 1080p | $37,19(\pm 0,43)$ | $37,01(\pm 0,42)$ | $\mathbf{39,65}(\pm 0,37)$ | $34,14(\pm 0,56)$ | $\mathbf{39,34}(\pm 0,39)$ |
| LPIPS | 720p | $0,064(\pm 0,004)$ | $0,064(\pm 0,004)$ | $0,057(\pm 0,007)$ | $\mathbf{0,044}(\pm 0,003)$ | $\mathbf{0,039}(\pm 0,003)$ |
| | 1080p | $0,066(\pm 0,005)$ | $0,066(\pm 0,005)$ | $0,063(\pm 0,007)$ | $\mathbf{0,050}(\pm 0,004)$ | $\mathbf{0,046}(\pm 0,004)$ |
| VMAF | 720p | $78,52(\pm 1,80)$ | $92,53(\pm 0,66)$ | $79,37(\pm 1,27)$ | $81,41(\pm 1,12)$ | $\mathbf{96,62}(\pm 0,55)$ |
| | 1080p | $82,48(\pm 1,72)$ | $93,64(\pm 0,80)$ | $82,30(\pm 1,13)$ | $83,63(\pm 1,01)$ | $\mathbf{97,08}(\pm 0,47)$ |

Fonte: De autoria própria.

A avaliação da qualidade dos vídeos super-resolvidos por meio da métrica PSNR (ver Tabela 9) demonstrou que os modelos SRResNet e VSRGAN+ obtiveram os melhores resultados, com 38,44 dB (±0,47) e 38,09 dB (±0,49) na resolução 720p, e 39,65dB(±0,37) e 39,34 dB (±0,39) na resolução 1080p, respectivamente.

No mesmo cenário, o modelo SRGAN apresentou o pior resultado, registrando 33,69 dB (±0,56) e 34,14 dB (±0,56) para as resoluções 720p e 1080p, respectivamente. Os modelos SRCNN e ESPCN obtiveram resultados intermediários de 35,89 dB (±0,51) e 35,68 dB (±0,50) na resolução de vídeo 720p, e 37,19 dB (±0,43) e 37,01 dB (±0,42) na resolução 1080p, respectivamente. É importante ressaltar que o resultado inferior do SRGAN se deve à natureza perceptual de sua função de erro, que não reflete bons resultados quando avaliada por métricas pixel a pixel, como é o caso do PSNR.

Figura 23 – Avaliação da qualidade dos vídeos restaurados com escala de 2× aplicando as métricas PSNR, LPIPS e VMAF.




(a) Avaliação da qualidade com a métrica PSNR. (b) Avaliação da qualidade com a métrica LPIPS. (c) Avaliação da qualidade com a métrica VMAF.

Fonte: De autoria própria.

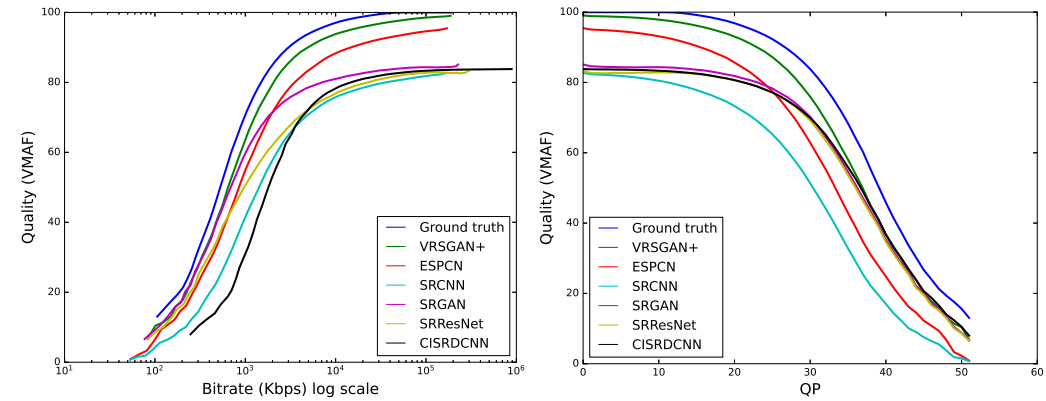
A avaliação da qualidade dos vídeos utilizando a métrica LPIPS revelou que os modelos VSRGAN+ e SRGAN obtiveram os melhores resultados, ou seja, as menores distâncias LPIPS, entre os modelos avaliados. Os vídeos super-resolvidos com VSRGAN+ nas resoluções 720p e 1080p apresentaram valores de LPIPS iguais a $0,039$ ($\pm 0,003$) e $0,046$ ($\pm 0,004$), respectivamente. Os valores do LPIPS do SRGAN foram $0,044$ ($\pm 0,003$) e $0,050$ ($\pm 0,004$). Já os outros três modelos apresentaram as maiores distâncias LPIPS, com $0,057$ ($\pm 0,007$) e $0,063$ ($\pm 0,007$) para o SRResNet, $0,064$ ($\pm 0,004$) e $0,066$ ($\pm 0,005$) para o SRCNN e $0,064$ ($\pm 0,004$) e $0,066$ ($\pm 0,005$) para o ESPCN, nas resoluções 720p e 1080p, respectivamente.

As avaliações de qualidade dos vídeos utilizando a métrica VMAF indicaram que o modelo VSRGAN+ obteve os melhores resultados entre os modelos avaliados, com valores de VMAF de $96,62$ ($\pm 0,55$) e $97,08$ ($\pm 0,47$) para as resoluções de vídeo 720p e 1080p, respectivamente. O modelo ESPCN apresentou valores de VMAF de $92,53$ ($\pm 0,66$) e $93,64$ ($\pm 0,80$) para as resoluções de vídeo de 720p e 1080p. Por outro lado, os modelos SRGAN, SRResNet e SRCNN obtiveram os menores valores de VMAF.

Em seguida, como os modelos anteriores são generalistas e tratam diferentes tipos de artefatos, introduziu-se o modelo CISRDCNN [149], especializado em artefatos de compressão, a fim de avaliar com mais detalhes a efetividade do modelo VSRGAN+. Nesse cenário, foi aplicada a escala 2× (ou seja, 540p para 1080p) nos vídeos e a qualidade perceptiva foi avaliada usando a métrica VMAF, variando a taxa de bits no intervalo $QP = x \in \mathbb{N} \mid 0 \leq x \leq 51$.

O treinamento do modelo *Super-resolution of compressed images using deep convolutional neural networks* (CISRDCNN) utilizou imagens com artefatos de compressão, definidos pelo fator de qualidade (QF, do inglês *quality factor*) 20 JPEG, aplicado para o conjunto de dados Places365-Standard [146].

Figura 24 – Avaliação da qualidade perceptiva dos vídeos redimensionados em 2× por métodos SR.




(a) Valores VMAF para vídeos codificados em diferentes taxas de bits.

(b) Valores VMAF para vídeos codificados com diferentes níveis QP.

Fonte: De autoria própria.

A Figura 24 apresenta os resultados do estudo que avaliou a qualidade perceptiva do vídeo para os diferentes modelos de SR avaliados. Na Figura 24a, a qualidade é considerada em relação aos níveis de compressão em taxa de bits, enquanto na Figura 24b, os níveis de compressão são apresentados de acordo com o QP. Os resultados indicam que o modelo VSRGAN+ apresentou melhor qualidade do que os demais modelos para níveis de compressão até 42 QP; somente para níveis de compressão acima de 42 QP, o modelo CISRDCNN superou o VSRGAN+.

Embora o modelo CISRDCNN seja treinado especificamente para restaurar imagens com artefatos de compressão, seus melhores resultados só foram alcançados em altas taxas de compressão (acima de QP 42), com pouca diferença em relação ao modelo VSRGAN+. No entanto, é importante destacar que vídeos com tais níveis de compressão apresentam baixa qualidade e não são práticos para aplicações de *streaming* de vídeo. Isso ocorre porque o método CISRDCNN não é adaptado para a qualidade perceptiva, o que limita seu desempenho nesse aspecto.

Em conclusão, o modelo VSRGAN+ apresentou os melhores resultados entre os modelos avaliados, segundo as três métricas utilizadas: VMAF, PSNR e LPIPS. Essa superioridade pode ser atribuída à função de erro do modelo (Equação (5.4)), a qual leva em consideração tanto recursos perceptivos quanto pixel a pixel. Isso mostra a versatilidade do modelo proposto, que aprende a pontuar valores altos em relação a essas métricas de qualidade.

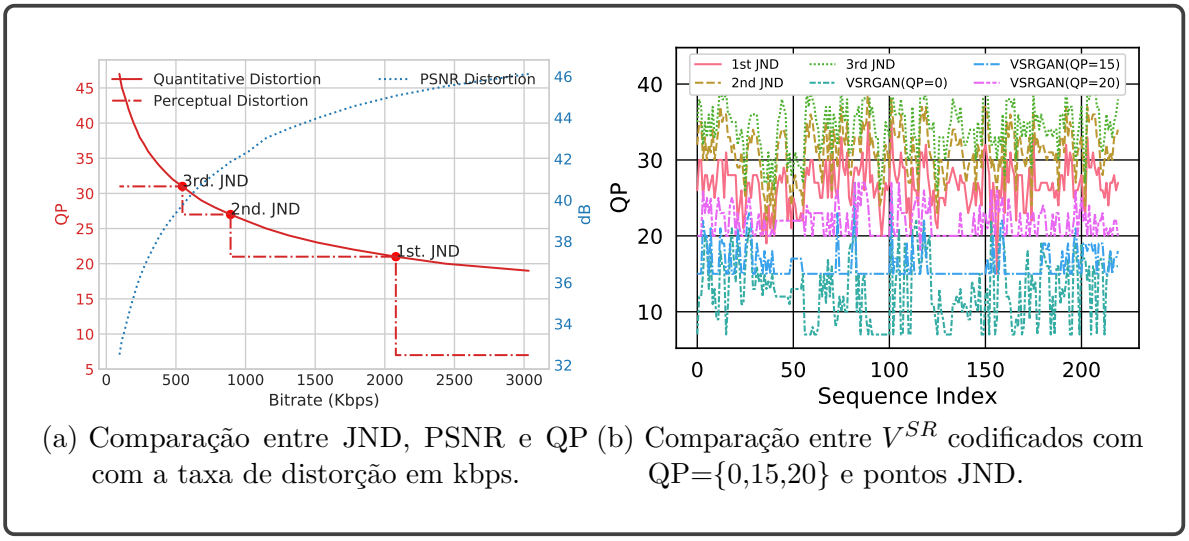
### 5.5.3 QUALIDADE PERCEPTIVA E MAPEAMENTO PARA JND

Analisou-se a influência da distorção nos vídeos da base de dados ao variar o parâmetro QP em relação à taxa de bits e a taxa de bits em relação ao PSNR. Com base nos pontos JND mapeados por Wang et al.[71], determinou-se a partir de qual valor de QP a distorção dos vídeos torna-se perceptível aos olhos humanos.

A Figura 25a apresenta o resultado da análise para o vídeo de índice 112 e resolução 1080p. Quanto menor o valor do parâmetro QP, maior será a taxa de bits utilizada para codificar (comprimir) o vídeo. Os demais vídeos da base de dados também foram avaliados e apresentaram comportamento semelhante, com pequenas variações nos pontos JND.

Para o vídeo selecionado (índice 112), a taxa de bits varia de 94 kbps quando QP é igual a 47 a 264.088 kbps quando QP é igual a 7. Observa-se que, à medida que a distorção diminui com a redução de QP, há um crescimento exponencial na taxa de bits do vídeo.

Figura 25 – Análise da percepção visual da distorção em vídeos.




(a) Comparação entre JND, PSNR e QP com a taxa de distorção em kbps.

(b) Comparação entre $V^{SR}$ codificados com QP={0,15,20} e pontos JND.

Fonte: De autoria própria.

Quanto à percepção visual avaliada pela métrica JND, observou-se que o primeiro ponto JND ocorre em QP=21, o segundo em QP=27 e o terceiro em QP=31 para o vídeo de índice #112, conforme mostrado na Figura 25a. Esses pontos de inflexão indicam os intervalos de distorção QP que não são perceptíveis à visão humana, como evidenciado pelo efeito de escada na linha tracejada do JND. Em outras palavras, para o vídeo #112, distorções variando nos intervalos $Q_1 = x \in \mathbb{N} \mid 0 \leq x < 21$, $Q_2 = x \in \mathbb{N} \mid 21 \leq x < 27$, $Q_3 = x \in \mathbb{N} \mid 27 \leq x < 31$, $Q_4 = x \in \mathbb{N} \mid 31 \leq x \leq 51$ não são perceptíveis para a visão humana.

Por outro lado, a métrica objetiva PSNR é sensível a cada mudança na taxa de bits, como observado na variação da taxa de bits para o vídeo #112, que varia de 94 kbps a 264.088 kbps à medida que o QP decai de 47 a 7. Assim, é possível concluir que a

visão humana não percebe algumas variações na qualidade dos vídeos, enquanto métricas objetivas, como o PSNR, são sensíveis a cada variação. Portanto, há níveis de distorção que podem ser aplicados aos vídeos sem que a qualidade perceptiva à visão humana seja afetada.

Com base na observação mencionada, foram analisados os vídeos restaurados pelo modelo VSRGAN+ aplicando-se a Equação (5.8), que utiliza a distância perceptiva LPIPS para identificar o QP que apresenta a menor distância perceptiva em relação aos vídeos restaurados $V^{SR}$ e seus correspondentes em alta resolução $V^{HR}$. Dessa forma, é possível mapear, de forma indireta e sem a necessidade de avaliação pela visão humana, em qual intervalo definido pelo JND ($Q_1,\ Q_2,\ Q_3,\ Q_4$) os vídeos restaurados se encontram.

$$QP_{V_{i,k}^{SR}} = j, \text{com } j \text{ sendo } \min\left(D\left(V_{i,k}^{SR}, V_{i,j}^{HR}\right)\right), \tag{5.8}$$

onde, $D$ representa a distância perceptiva LPIPS, enquanto $k = \{0, 15, 20\}$ se refere ao QP do vídeo que foi utilizado para a restauração. O índice do vídeo é denotado por $i = \{1, \ldots, 220\}$, enquanto $j = \{7, \ldots, 47\}$ é usado para identificar a distorção em QP que foi empregada para codificar os vídeos em alta resolução $V^{HR}$. Com base nesses parâmetros, o vídeo restaurado $V_{i,k}^{SR}$ é mapeado de acordo com os seguintes conjuntos:

$$V_{i,k}^{SR} \in \begin{cases} Q_1, \text{ se } QP_{V_{i,k}^{SR}} < QP_{1^oJND,i} \\ Q_2, \text{ se } QP_{1^oJND,i} \leq QP_{V_{i,k}^{SR}} < QP_{2^oJND,i} \\ Q_3, \text{ se } QP_{2^oJND,i} \leq QP_{V_{i,k}^{SR}} < QP_{3^oJND,i} \\ Q_4, \text{ se } QP_{V_{i,k}^{SR}} \geq QP_{3^oJND,i}, \end{cases}$$

onde, $QP_{1^oJND,i},\ QP_{2^oJND,i}$ e $QP_{3^oJND,i}\ \in\ QP_{V_i^{HR}}$.

Na Figura 25b, é possível observar os pontos JND dos 220 vídeos originais em resolução 1080p, bem como os QPs mapeados pela Equação (5.8) dos vídeos restaurados pelo modelo VSRGAN+, a partir das versões originais de 540p com QP=$\{$0,15,20$\}$. Os vídeos são ordenados no eixo $x$ pelos seus índices $i = \{1, \ldots, 220\}$.

Pode-se verificar que os vídeos restaurados a partir dos QPs 0 e 15 possuem qualidade mapeada em $Q_1$ em 100% dos casos, enquanto aqueles restaurados a partir do QP 20 encontram-se mapeados em $Q_1$ em $91,4\%$ dos casos e em $Q_2$ em $8,6\%$ dos casos. Isso evidencia que, mesmo quando o vídeo passa por um processo de super-resolução, se gerado a partir de matrizes com baixa taxa de compressão (QP 0 e 15), os ruídos introduzidos não são percebidos pela visão humana. Quando restaurados a partir de matrizes com nível de compressão dado por QP=20, apenas $8,6\%$ apresentam diferenças perceptíveis na qualidade visual.

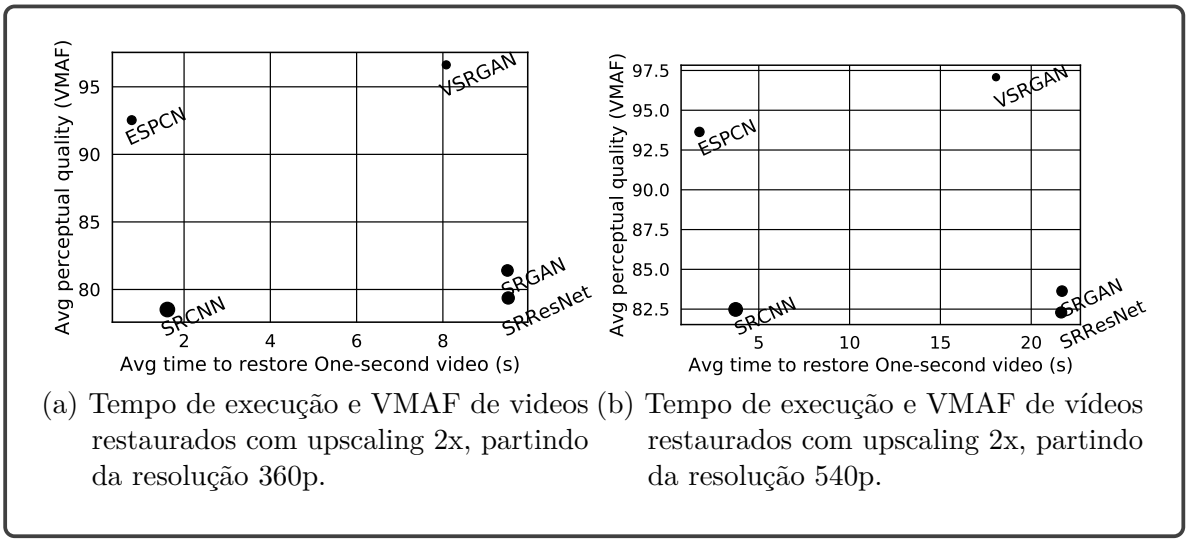
Dessa forma, conclui-se que o modelo VSRGAN+ foi capaz de restaurar vídeos com qualidade perceptiva indistinguível dos originais pela visão humana, mesmo após

terem sido submetidos a um processo de compressão.

### 5.5.4 ANÁLISE DO TEMPO DE EXECUÇÃO

Na Figura 26, apresenta-se o tempo médio de execução dos modelos para restaurar um segundos de vídeo de 360p e 540p para 720p e 1080p, respectivamente. A qualidade dessas sequências super-resolvidas foi avaliada pelo métrica VMAF. O modelo proposto, ou seja, VSRGAN+, teve um custo computacional intermediário. Foi superior ao custo dos modelos SRCNN e ESPCN, mas inferior ao dos modelos SRResNet e SRGAN. Esse desempenho intermediário foi compensatório, pois permitiu que o VSRGAN+ alcançasse a mais alta qualidade de percepção avaliada pela métrica VMAF.

Figura 26 – Tempo médio para restaurar um segundo de vídeo vs. qualidade quantitativa e perceptiva.




(a) Tempo de execução e VMAF de videos restaurados com upscaling 2x, partindo da resolução 360p.

(b) Tempo de execução e VMAF de vídeos restaurados com upscaling 2x, partindo da resolução 540p.

Fonte: De autoria própria.

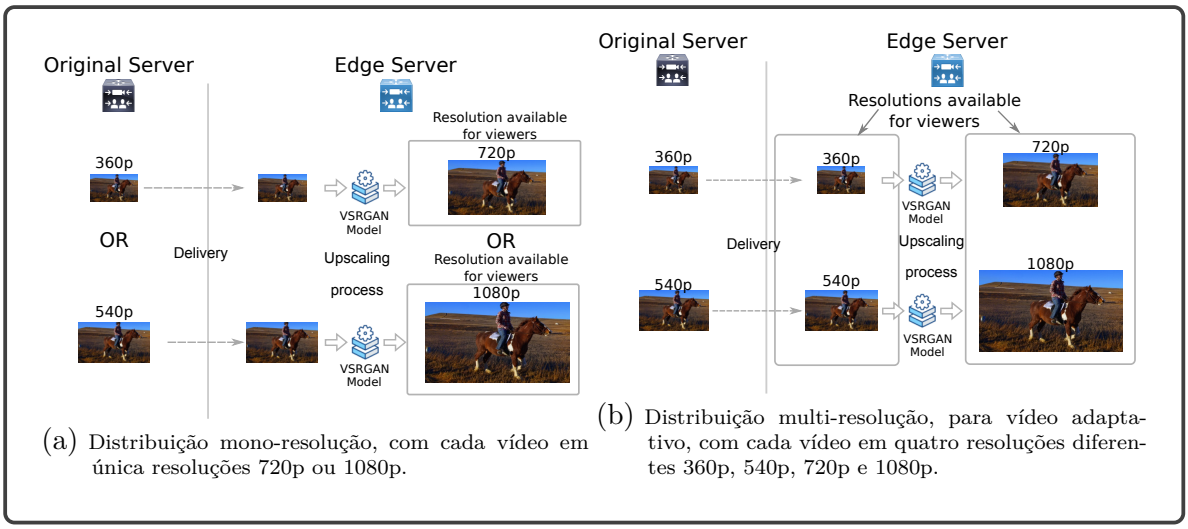
A arquitetura de distribuição de vídeo proposta é especialmente projetada para a modalidade *video on demand* (VoD), como descrito na seção 5.1. Nesse cenário, qualquer editor que se preocupe com altos padrões de qualidade perceptiva concordaria em adiar o lançamento do conteúdo para obter a melhor saída na etapa de publicação. Esse raciocínio reforça a importância de equilibrar o custo computacional e a qualidade perceptiva dos vídeos super-resolvidos, como é o caso do modelo VSRGAN+.

### 5.5.5 ANÁLISE DA REDUÇÃO DE DADOS

Esta subseção apresenta a avaliação para medir a redução na quantidade de dados de vídeos que flui pela infraestrutura de rede. Essa infraestrutura conecta a fonte de vídeo com os servidores substitutos que distribuem sequências de vídeo em modalidades de mono ou múltipla resolução.

No primeiro cenário, mostra-se a diminuição na transferência de dados para uma modalidade de mono-resolução. Nesta modalidade, os vídeos de baixa resolução $V^{LR}$ codificados em 360p ou 540p são super-resolvidos usando um fator de escala de 2× e a audiência engajada acessa esse conteúdo em resolução de 720p ou 1080p (consulte Figura 27a).

Figura 27 – Distribuição de vídeo em mono-resolução e multi-resolução com modelo de SR de vídeo.




(a) Distribuição mono-resolução, com cada vídeo em única resoluções 720p ou 1080p.

(b) Distribuição multi-resolução, para vídeo adaptativo, com cada vídeo em quatro resoluções diferentes 360p, 540p, 720p e 1080p.

Fonte: De autoria própria.

Na Figura 28a é apresentada a função de distribuição acumulada (FDA) para a redução de tráfego gerada pelo uso do *framework* proposto. As curvas mostram a probabilidade da redução atingir certos valores quando usa-se a SR para distribuir vídeos nas resoluções 720p e 1080p. Indiretamente, mediu-se o quanto de dados deixou de se transmitir na abordagem com distribuição em mono-resolução.

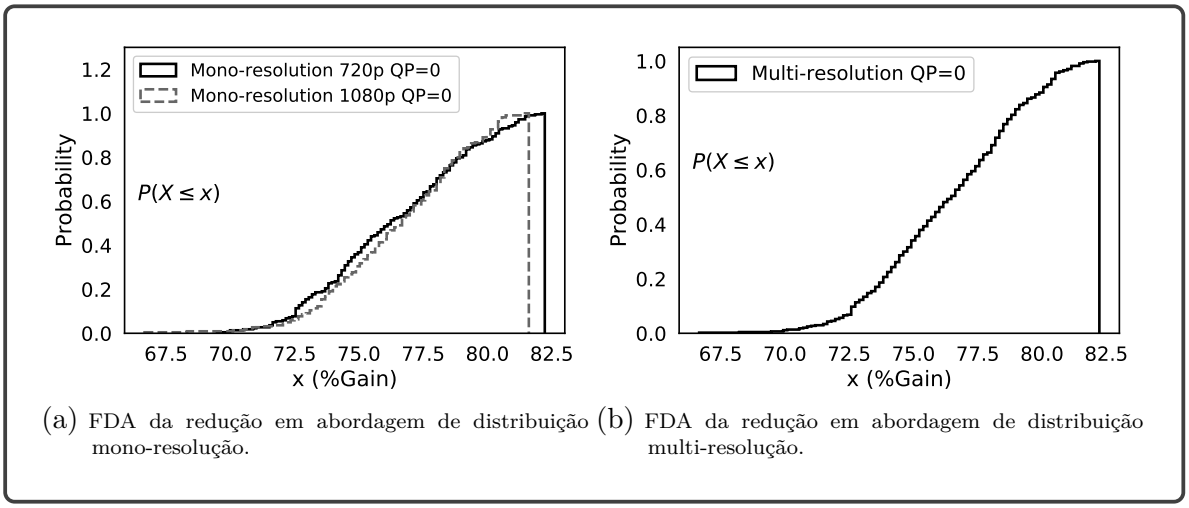
Cada FDA foi calculada a partir de 220 amostras dos vídeos transmitidos em 360p e 540p, ambos com $QP = 0$, e restaurados em 720p e 1080p, respectivamente. O cálculo da redução para cada amostra $V_i$ se deu pela Equação (5.9).

$$R_{mono}(V_i) = 1 - \frac{size(V_i^{LR})}{size(V_i^{HR})} \tag{5.9}$$

Para as transmissões em 360p e restauração em 720p a probabilidade de redução varia de aproximadamente $69,48\%$ a $82,23\%$. Para os vídeos transmitidos na resolução 540p e restaurados em 1080p a redução fica de $66,67\%$ a $81,61\%$.

Em valores absolutos para a base inteira dos 220 vídeos de 360p (2,6GB) e 720p (11,04GB) houve uma redução de $76,45\%$, que corresponde a aproximadamente 8,44GB de dados. A redução absoluta na versão de 540p (6GB) para 1080p (25,9GB) foi de $76,8\%$, que corresponde a aproximadamente 19,9GB.

Figura 28 – FDA da redução em volume de dados quando utilizada SR de 2×, com abordagem de distribuição mono e multi-resolução.




(a) FDA da redução em abordagem de distribuição mono-resolução.

(b) FDA da redução em abordagem de distribuição multi-resolução.

Fonte: De autoria própria.

Na Figura 28b a FDA foi calculada considerando que para cada vídeo são transmitidas duas versões em baixa resolução, 360p e 540p, as quais são restauradas no servidor de borda para as resoluções 720p e 1080p, respectivamente. A diferença em relação a anterior é que ao final, quatro versões do vídeo ficam disponíveis para serem acessadas pela audiência em uma abordagem multi-resolução, cenário ilustrado na Figura 27b. Deste modo, a redução para uma amostra $V_i$ foi calculado pela Equação (5.10).

$$R_{multi}(V_i) = 1 - \frac{size(V_{i360p}^{LR} + V_{i540p}^{LR})}{size(V_{i360p}^{LR} + V_{i540p}^{LR} + V_{i720p}^{HR} + V_{i1080p}^{HR})} \tag{5.10}$$

Na Figura 28b, apresenta-se a FDA da redução para a abordagem multi-resolução, cenário ilustrado na Figura 27b. Neste cenário a redução varia de $75,67\%$ a $84,59\%$. Em valores absolutos o volume de dados dos vídeos em baixa resolução (360p e 540p) que são transmitidos correspondem a aproximadamente 8,63GB, já o volume de dados dos vídeos disponíveis à audiência após o processo de restauração (360p, 540p, 720p e 1080p) é de aproximadamente 45,57GB, apresentando uma redução absoluta de 36,94GB. Isso resulta em $81,06\%$ menos dados indo para a infraestrutura de distribuição para atender ao público dos servidores substitutos.

### 5.5.6 REDUÇÃO DE DADOS POR SUPER-RESOLUÇÃO E A COMPRESSÃO

Realizou-se também, análise comparativa do tamanho dos vídeos, dada a variação da resolução e os níveis de compressão em termos do parâmetro QP.

Na Tabela 10, são apresentados os tamanhos médios, com intervalo de confiança de 95%, das 220 amostras de vídeos para as resoluções de 360p, 540p, 720p, 1080p e, também, dada a variação de níveis de compressão em função do QP=$\{0,10,15,20,25\}$. Observa-se

pelos intervalos de confiança que a variabilidade nos tamanhos dos 220 vídeos é pequena quando analisados mesma resolução e mesmo QP na compressão. Analisando o intervalo de confiança, em porcentagem em ralação a média, percebe-se que varia entre 4% a 18%, sendo maior a porcentagem relativa nos conjunto de vídeos com maior compressão, *i.e.*, com QP=25.

Tabela 10 – Tamanho médio dos vídeos com variação de QP={0, 10, 15, 20, 25}.

| QP | 360p | 540p | 720p | 1080p |
|---|---|---|---|---|
| 0 | 11,80Mb (±0,54) | 27,43Mb (±1,20) | 50,18Mb (±2,16) | 117,71Mb (±5,07) |
| 10 | 4,74Mb (±0,44) | 11,20Mb (±0,97) | 21,42Mb (±1,75) | 53,76Mb (±4,13) |
| 15 | 2,38Mb (±0,29) | 5,01Mb (±0,62) | 9,00Mb (±1,09) | 22,81Mb (±2,54) |
| 20 | 1,24Mb (±0,18) | 2,35Mb (±0,35) | 3,80Mb (±0,58) | 8,18Mb (±1,28) |
| 25 | 0,65Mb (±0,10) | 1,18Mb (±0,20) | 1,80Mb (±0,32) | 3,38Mb (±0,59) |

Fonte: De autoria própria.

Na Figura 29a e Figura 29b verifica-se que ocorre um decaimento exponencial no tamanho médio dos vídeos, tanto reduzindo a resolução quanto reduzindo a compressão em níveis do QP, indicando que é possível reduzir o volume de dados a transmitir aplicando a compressão e também reduzindo a resolução. A fim de comprovar essa evidência, analisou-se a redução de dados a transmitir tendo como referência os vídeos com o QP=0.

Na Tabela 11, encontram-se os valores médios, com intervalo de confiança de 95%, das reduções para o tipo de redução de dados distribuídos em sistema mono (720p e 1080p) e multi-resolução, como ilustrado na Figura 27.

Tabela 11 – Redução na distribuição de vídeos mono e multi-resolução com super-resolução e compressão.

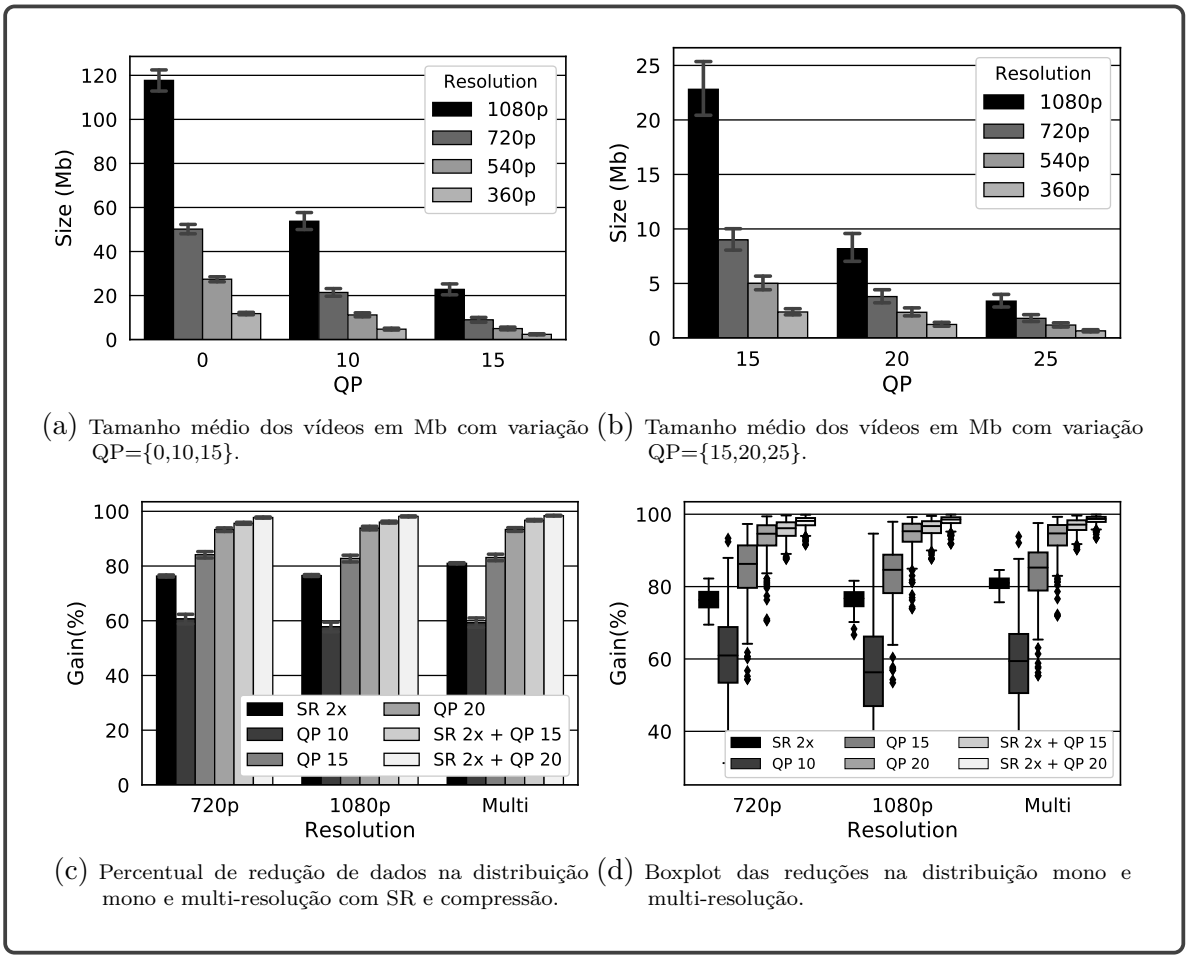
| Tipo de Redução | Mono-resolução 720p | Mono-resolução 1080p | Multi Resolução |
|---|---|---|---|
| SR 2× | 76,35%(±0,38) | 76,52%(±0,35) | 80,99%(±0,23) |
| QP10 | 60,67%(±1,73) | 57,83%(±1,78) | 59,28%(±1,69) |
| QP15 | 84,13%(±1,23) | 82,80%(±1,22) | 83,12%(±1,18) |
| QP20 | 93,34%(±0,69) | 93,91%(±0,65) | 93,37%(±0,68) |
| SR 2×+QP15 | 95,62%(±0,36) | 96,07%(±0,34) | 96,74%(±0,27) |
| SR 2×+QP20 | **97,74**%(±0,22) | **98,14**%(±0,20) | **98,42**%(±0,16) |

Fonte: De autoria própria.

Para melhor comparação, os valores das reduções também estão em forma de gráfico na Figura 29c. Observa-se que a redução por super-resolução em 2× (SR 2×) é melhor que a redução por compressão com QP=10 tanto em mono quanto e multi-resolução. A combinação de SR 2× com QP=15 tem mais redução que somente o QP=15 ou QP=20 e, a melhor redução é alcançado quando se combina SR 2× + QP 20, que chega a 97, 4%, 98, 14% e 98, 42%, respectivamente para distribuição mono-resolução 720p, 1080p e multi-resolução.

Também foi analisada a amplitude da redução nas 220 amostras, como apresentado na Figura 29d. Pode-se observar que as reduções por compressão têm uma amplitude

Figura 29 – Tamanho médio dos vídeos codificados em diferentes resoluções e níveis de compressão e a redução de tráfego quando os vídeos são distribuídos em sistema mono e multi-resolução.




(a) Tamanho médio dos vídeos em Mb com variação QP={0,10,15}.

(b) Tamanho médio dos vídeos em Mb com variação QP={15,20,25}.

(c) Percentual de redução de dados na distribuição mono e multi-resolução com SR e compressão.

(d) Boxplot das reduções na distribuição mono e multi-resolução.

Fonte: De autoria própria.

maior e estão mais dispersas em comparação com as reduções por SR, o que também é refletido na Tabela 11 com intervalos de confiança maiores para a compressão. Essa dispersão ocorre mais na compressão devido à sua relação intrínseca com a forma como a compressão explora os pixels e quadros dos vídeos. A compressão elimina informações redundantes ou irrelevantes da imagem original para diminuir o tamanho do arquivo, portanto, dependendo da repetição de cores e quadros nos vídeos, a compressão pode ser mais efetiva em alguns casos.

Por outro lado, na SR, a redução é mais uniforme, uma vez que envolve uma redução no tamanho da resolução do vídeo. Isso resulta em uma menor dispersão nas reduções, como pode ser observado pela menor amplitude dos *boxplots* de SR na Figura 29d.

## 5.6 CONSIDERAÇÕES

Neste capítulo, utilizou-se o modelo de rede neural VSRGAN+ para restaurar vídeos em uma arquitetura de replicação de conteúdo de *streaming* na nuvem. Na abordagem adotada, os vídeos foram transmitidos em baixa resolução entre o servidor original e os servidores de borda, reduzindo, assim, o tráfego de dados nas infraestruturas do serviço de distribuição de vídeo baseado em nuvem. O modelo VSRGAN+ foi empregado nos servidores substitutos para restaurar os vídeos em escala de 2×. Os resultados revelaram uma redução de até $98,42\%$ em relação à distribuição com compressão sem perda.

Foram utilizadas três métricas de avaliação de qualidade para analisar a qualidade dos vídeos restaurados, incluindo uma métrica de comparação pixel a pixel (PSNR) e duas métricas perceptivas (LPIPS e VMAF). Os resultados mostraram que a SR de vídeo, utilizando o modelo VSRGAN+ com escala de 2×, manteve uma qualidade perceptiva indistinguível da visão humana entre os vídeos restaurados e suas referências.

No capítulo 6, esta pesquisa vai além e explora o poder computacional das MECs e o crescimento da largura de banda disponível nas redes móveis, especialmente as redes 5G e além, para aproximar o serviço de escala de vídeo por SR do público que consome conteúdo de vídeo. O objetivo é disponibilizar conteúdos de vídeos de alta definição em servidores mais próximos da audiência, reduzindo, assim, o tráfego de conteúdos de vídeo nas infraestruturas que conectam as redes móveis ao núcleo da Internet.

# CAPÍTULO 6

# Streaming de Vídeos ao Vivo com Aprimoramento de Qualidade Perceptiva por Super-resolução Provida por Computação de Borda

As redes 5G e 6G contemplam em suas especificações aplicações que demandam altas taxas de transmissão, o que sugere um crescimento das taxas de transmissão na última milha ao longo da próxima década [150]. De acordo com o anuário da Internet, produzindo pela Cisco [151], mais de 70% da população mundial terá conectividade móvel até 2023, deste quantitativo, mais de 10% serão por conectividade 5G. A previsão é que a velocidade média nos dispositivos móveis mais que triplicarão, indo de 13,2 Mbps em 2018 para 43,9 Mbps em 2023 e, a velocidade média de conexão 5G chegará a 575 Mbps, sendo 13 vezes maior que a média da conexão móvel.

No mesmo anuário, constata-se o grande número e a diversidade de dispositivos que estarão conectados à rede. Estima-se que até 2023 o número de TVs conectadas no mundo será de 3,2 bilhões, desse total de TVs instaladas, 66% serão 4K, *ultra high definition* (UHD). Os aparelhos de *smartphones* passarão de 4,9 bilhões em 2018 para 6,7 bilhões até 2023.

Esses dois fatores combinados, i.e., crescimento da velocidade das redes de acesso e o aumento no número de dispositivos conectados com capacidades de consumo e geração de vídeos UHD, tem alterado o padrão de tráfego da Internet. A tecnologia de vídeo UHD possui, em média, uma taxa de bits de 15 a 18 Mbps, sendo mais que o dobro da taxa de bits de vídeo *high definition* (HD) e nove vezes mais do que a taxa de bits de vídeo *standard-definition* (SD).

Segundo a Cisco [1], o tráfego global de vídeo na Internet representa 82% e o tráfego de vídeo ao vivo aumentou em 15 vezes de 2016 até 2021. A tendência é que aplicações de vídeos do futuro elevem ainda mais o volume de tráfego dos conteúdos de *streaming* na Internet devido à popularização de aplicações de realidade virtual em HD e UHD [151].

A despeito da evolução verificada nas redes móveis, as redes de *backhaul*, que interligam as redes de acesso ao núcleo da Internet, não têm acompanhado essa evolução e passaram a ser um ponto de gargalo de dados provindos/enviados ao núcleo da Internet [38, 152, 153, 154, 150]. Superar essa restrição tem sido um desafio para as aplicações que demandam maior largura de banda e menor jitter, como é o caso das diferentes formas de streaming de vídeo.

As redes de *backhaul* geralmente alcançam longas distâncias e, em sua maioria, utilizam o meio óptico e apresentam um maior custo de atualização [153]. Nesse contexto, descompassos entre eventos de atualização dessas redes e a evolução das redes de acesso podem impactar a percepção que a melhoria dessas últimas poderia proporcionar [38, 152, 153, 154]. Além disso, as redes backhaul são compartilhadas por vários ISPs, o que naturalmente aumenta a intensidade de tráfego, contribuindo para a formação de gargalos em todo o sistema. Nesse arranjo, torna-se um desafio a operacionalização de aplicações distribuídas que têm fortes restrições temporal e de taxa de transmissão.

Uma tendência para enfrentamento desse desafio tem sido trazer para as redes de acesso os serviços que geram maior tráfego. Neste sentido, diversas pesquisas [7, 65, 155, 156] têm proposto a instanciação de técnicas e tecnologias de computação em nuvem, em escala reduzida de processamento, armazenamento e transmissão, a partir da borda da rede. Esse movimento é observado em diversos serviços, por exemplo, *internet of things* (IoT), *virtual reality* (VR) e *streaming* de vídeo.

Nos últimos anos, ocorreram importantes avanços no uso de DNNs, visando a solução de problemas complexos. Os problemas formulados na área de visão computacional foram particularmente beneficiados com tais avanços, em particular, pelo contexto deste trabalho, a SR de imagens e vídeos [47]. A SR é a tarefa de escalar imagem e vídeo de uma resolução menor para uma maior minimizando os impactos que essa operação causa na qualidade visual da imagem e vídeo. As DNNs para SR têm atingido resultados sem precedentes entre os métodos anteriormente existentes e têm sido apontadas como oportunas para sintetizar vídeos de alta definição e melhorar a forma de distribuição de vídeos na Internet [157, 158].

Neste capítulo é apresentado e avaliado o *framework On-edge enhanced live streaming with super-resolution* (ELiveSR) para distribuição de vídeo adaptativo ao vivo assistido por computação de borda, que tira proveito da melhoria de velocidade das redes de acesso e faz uso de SR em tempo real baseado em DNN para reduzir o volume de tráfego de vídeo na porção *backhaul* da rede, além disso, aprimora a entrega de vídeos em

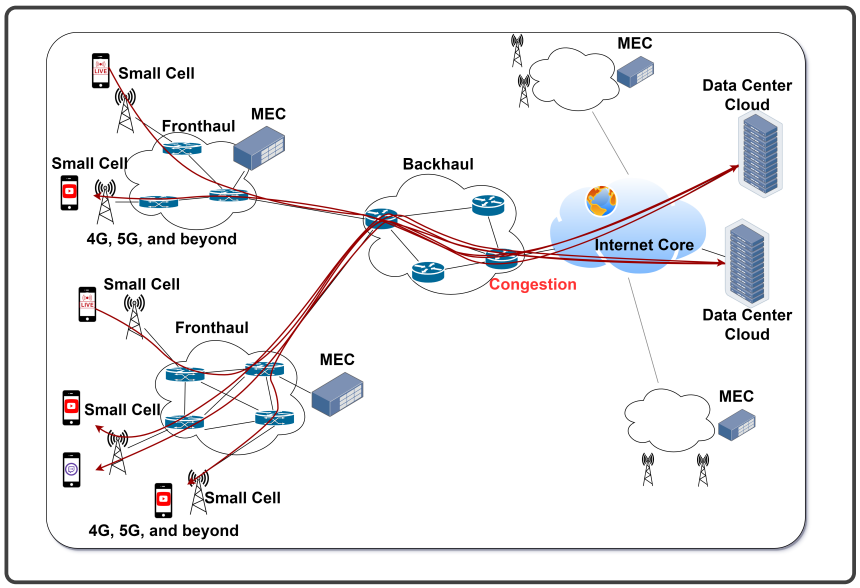
alta definição e melhora a QoE das audiências.

## 6.1 VISÃO GERAL DO FRAMEWORK

O *framework* proposto neste capítulo tem como objetivo melhorar a QoE de sessões de vídeos ao vivo em cenários em que as redes de *backhaul* estão congestionadas. As aplicações de *streaming* ao vivo exigem largura de banda e apresentam um ritmo de crescimento consistente [151], sugerindo que a otimização de recursos dedicados a transmissão dessa modalidade de streaming propiciará melhoria em seus indicadores de QoE.

A Figura 30 ilustra um exemplo deste cenário. A rede móvel e a porção *fronthaul* são caracterizadas com redes de banda larga. O tráfego das aplicações de vídeos concorre com o tráfego das demais aplicações na rede de *backhaul*, contribuindo para o seu congestionamento.

Figura 30 – Cenário de rede, *backhaul* e *fronthaul* e seu papel em sessões de transmissão ao vivo.



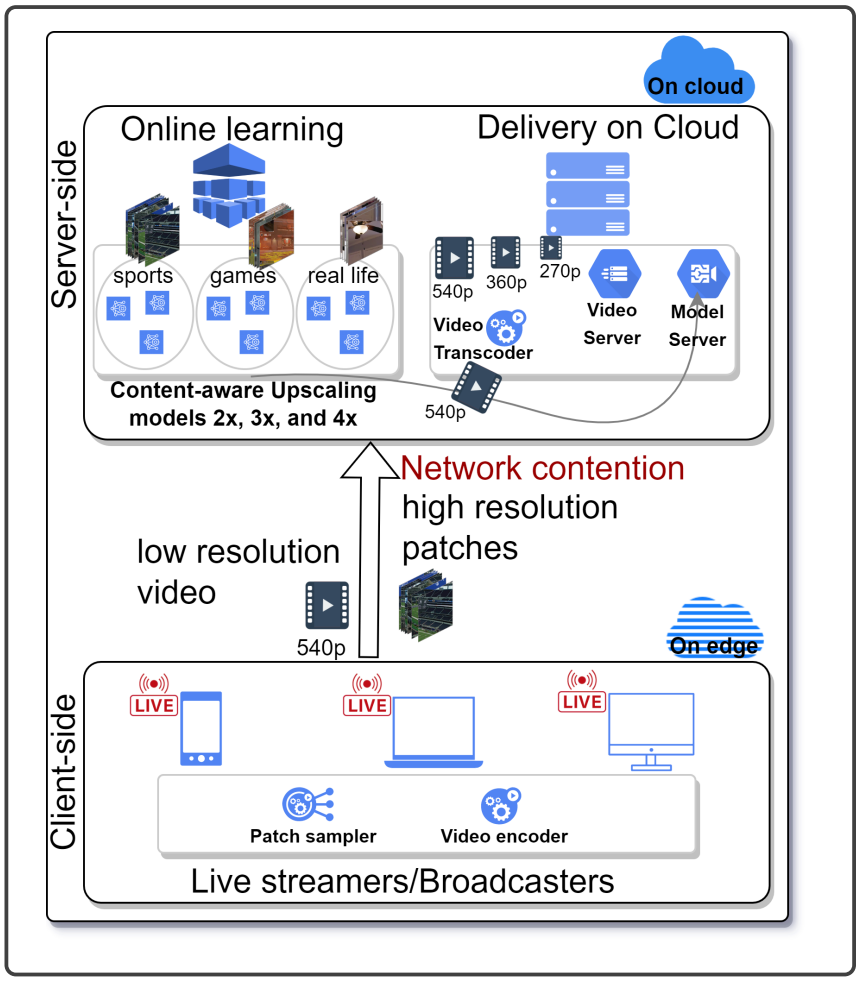

Fonte: De autoria própria.

No *framework* ELiveSR, O fluxo de vídeo é gerado em duas fases. A primeira, ilustrado na Figura 31, é a fase de ingestão de conteúdo no sistema e envolve o cliente de transmissão, que encontra-se em uma rede de borda, e o servidor de ingestão, que encontra-se em uma infraestrutura de nuvem. A segunda, ilustrada na Figura 32, é responsável pela entrega do *streaming* ao vivo aos servidores descentralizados, localizados nas MECs. Ainda nesta segunda fase encontra-se a entrega do conteúdo às audiências, usando tecnologia de adaptação de taxa de codificação. A seguir descrevemos com detalhes cada uma dessas fases e seus componentes internos.

### 6.1.1 A FASE DE INGESTÃO DE CONTEÚDO

O vídeo é transmitido em baixa resolução da fonte (client-side) para um servidor de ingestão localizado em nuvem (server-side). A partir do servidor de ingestão o vídeo fica disponível para ser distribuído aos servidores de borda. Conforme ilustrado na Figura 31, a parte de ingestão possui módulos específicos descritos a seguir.

Figura 31 – O estágio de ingestão de conteúdo




Fonte: De autoria própria.

#### 6.1.1.1 O CLIENTE DE TRANSMISSÃO AO VIVO

Similar ao sistema de ingestão proposto em [62], no lado cliente de transmissão (fonte), o vídeo é capturado em alta resolução. O módulo chamado "*patch sampler*" extrai amostras de dimensões reduzidas, como por exemplo $144 \times 144$ pixels, para o treinamento *online* do modelo de SR. Em seguida, o vídeo é codificado para baixa resolução e, juntamente com as amostras, é transmitido para o servidor em nuvem. A transmissão em baixa resolução requer menor largura de banda de *uplink* e permite o *streaming* em uma conexão de baixa taxa de transferência de ponta a ponta. Por exemplo, um vídeo com resolução de 540p e

taxa de bits de 850 Kbps pode ser transmitido através de um *uplink* com uma taxa de transferência de 1 Mbps.

#### 6.1.1.2 O MODELO DE SUPER-RESOLUÇÃO SENSÍVEL AO CONTEÚDO

Uma plataforma de *live streaming* possui múltiplos canais, com conteúdos diversificados, e.g., Youtube e Twitch. Nesse contexto, os vídeos produzindo e transmitidos pela plataforma serão agrupados com base na similaridade de seus conteúdos. Diversas abordagens podem ser aplicadas para realizar esse agrupamento, por exemplo, agrupamentos baseado em metadados, incluindo-se as categorias, ou ainda o agrupamentos baseado em modelos de inteligência artificial. O estudo do melhor método de agrupamento para conteúdo gerado naquelas plataformas extrapola o escopo deste trabalho. Dessa forma, assume-se que vídeos que chegam à plataforma, como fluxos a serem distribuídos, foram previamente classificados e agrupados. Para cada agrupamento, modelos de SR com fator de escala de 2×, 3× e 4× são treinados e ficam disponíveis nos servidores de entrega.

#### 6.1.1.3 O TREINAMENTO ONLINE

Os vídeos apresentam uma quantidade significativa de redundância de dados, não apenas entre quadros adjacentes, mas também entre quadros temporalmente distantes. Por exemplo, em uma transmissão de uma partida de futebol, as cenas do gramado se repetem ao longo da transmissão. O treinamento *online* ocorre periodicamente a cada $x$ minutos e utiliza amostras de quadros da mesma transmissão para aproveitar essa redundância de dados temporalmente distantes, visando melhorar a precisão dos modelos de SR.

No lado do servidor de ingestão, as amostras recebidas da fonte passam por um pré-processamento para aprimorar o modelo, levando em consideração o conteúdo específico de cada agrupamento. Como esse módulo está localizado na nuvem, onde há maior poder de processamento, a aprendizagem *online* é realizada utilizando computação paralela em GPUs, sem interferir no fluxo de transmissão. Em outras palavras, esse treinamento é realizado por servidores separados. Nessa fase, é utilizado o treinamento *online* adversarial com o uso de uma função de erro perceptiva. Esse tipo de treinamento melhora a qualidade perceptiva dos vídeos [22, 45, 46]. O treinamento *online* ocorre periodicamente, e os modelos aprimorados são disponibilizados pelo servidor de modelos para serem obtidos pelos servidores das MECs.

#### 6.1.1.4 O SERVIÇO DE CODIFICAÇÃO DE VÍDEO

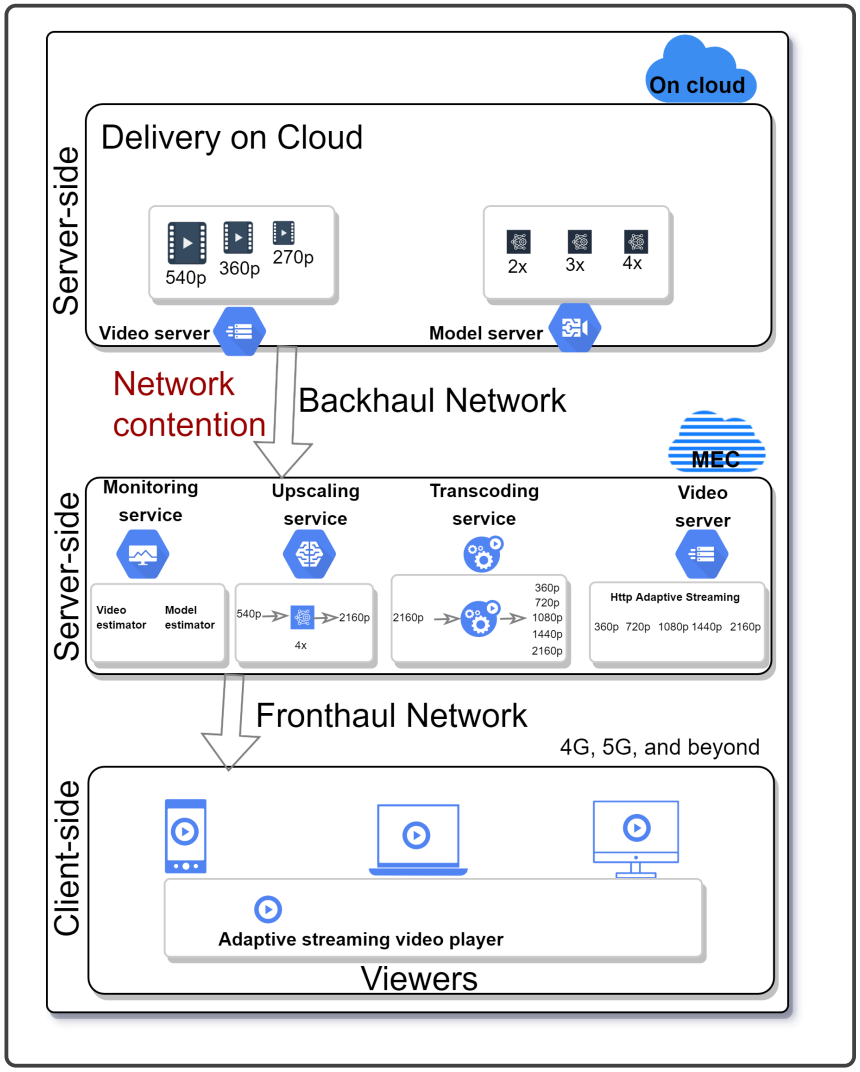
Antes de ser entregue aos servidores de distribuição nas MEC, o vídeo passa por um processo de codificação que o converte em $n$ representações com diferentes resoluções. Por exemplo, um vídeo recebido em 540p pode ser codificado para as resoluções 270p,

360p e 540p. Esta codificação tem como objetivo servir MECs com diferentes larguras de banda em seus enlaces na *backhaul*.

### 6.1.2 A DISTRIBUIÇÃO DE CONTEÚDO

No *live streaming*, a distribuição ocorre em duas etapas. A primeira etapa envolve o servidor de distribuição, localizado na nuvem, para os servidores localizados nas MECs. A segunda etapa envolve os servidores na MECs e os equipamentos da audiência, que devem ser habilitados a reproduzir conteúdo com taxa de bit variada. A Figura 32 ilustra os componentes do *framework* envolvidos na distribuição.

Figura 32 – Serviços executados no lado do cliente e no lado do servidor.




Fonte: De autoria própria.

#### 6.1.2.1 O SERVIÇO DE POSICIONAMENTO DE CONTEÚDO

Em nuvem, após o vídeo ser codificado, as $n$ representações ficam disponíveis para serem entregues aos servidores nas MECs. Uma vez que há mais de uma taxa de bits

disponível no servidor em nuvem, é possível que esta entrega seja de forma adaptativa com emprego de algoritmos de adaptação de taxa de bits. No entanto, nesse estudo não empregou-se abordagem adaptativa na entrega em nuvem.

#### 6.1.2.2 O SERVIÇO DE MONITORAMENTO DE CONTEÚDO

O módulo de monitoramento de serviço mantém uma lista dos canais com transmissão ao vivo e monitora a vazão da rede *backhaul* e da rede móvel. O evento de chegada de um espectador para uma sessão ao vivo é seguido por duas ações: 1) definição da taxa de bits do vídeo $v_{bitrate}$ que precisa ser puxada do servidor de distribuição localizado na nuvem; e, 2) definição de qual fator de escala $f\_sc$ do modelo de *upscaling* será empregado. O Algoritmo 1 apresenta a computação envolvida nessas ações.

**Algoritmo 1** Definir taxa de bits de vídeo e fator de upscaling do modelo de super-resolução

**Pseudocódigo** MONITORAMENTO($v_{bitrate}, f\_sc$)
$b\_thrpt \leftarrow get\_backhaul\_average\_throughput()$
$f\_thrpt_{max} \leftarrow get\_fronthaul\_max\_throughput()$
$v_{bitrate} \leftarrow \max v_i \mid i \in [1, ..., n] \wedge v_i < (1 - \tau) \times b\_thrpt$
$f\_sc \leftarrow \max k \mid k \in [1, 4] \wedge k \leq \lceil ((1 - \beta) \times f\_thrpt_{max}) / v_{bitrate} \rceil$

A variável $b\_thrpt$ é a vazão média da rede *backhaul*; a variável $f\_thrpt_{max}$ é a vazão máxima da rede *fronthaul*; a variável $v_{bitrate}$ é a taxa de bits do vídeo a ser puxada do servidor de nuvem; a variável $f\_sc$ é o fator de escala do modelo de *upscaling* a ser puxado da nuvem; e as variáveis $\tau$ e $\beta$ são fatores de ajustes, definidos no intervalo $\{x \in \mathbb{R};\ 0 \leq x < 1\}$, da vazão estimada das redes *backhaul* e *fronthaul*, respectivamente. Caso $f\_sc$ seja igual a 1 (um), não será realizado *upscaling* nos vídeos e a entrega através da MEC terá $v_{bitrate}$ como sendo a maior taxa de bits entre as representações disponíveis. Neste estudo considerou-se que os recursos computacionais disponíveis na MEC são suficientes para atender as tarefas de *upscaling* e codificação e, que, o custo associada a essa operação é justificado por uma possível perda de qualidade, afetando a aderência da audiência aos canais.

Ao iniciar uma transmissão, no lado da MEC, após o algoritmo decidir qual taxa de bits e qual fator de escala serão empregados, ambos, vídeo e modelo de *upscaling*, são obtidos do servidor da nuvem para a MEC.

#### 6.1.2.3 O SERVIÇO DE UPSCALING DE VÍDEO

O serviço de *upscaling* é realizado pelo modelo de SR que foi obtido da nuvem no início da transmissão e pode ser de fator de escala 2×, 3× ou 4×. Por exemplo, se a versão do vídeo em baixa resolução puxada da nuvem for de 540p e o modelo de *upscaling* for de fator de escala de 4×, a audiência será servida a partir da MEC com versões dos vídeos de

até 4K. Caso a versão de baixa resolução puxada da nuvem seja de 270p e o modelo for de 4×, a audiência terá resolução máxima de 1080p. Periodicamente os pesos do modelo de SR obtidos no treinamento *online* são replicados da nuvem para as MECs, de maneira que o modelo utilize da redundância de dados temporais de quadros não adjacentes para restaurar vídeos obtendo melhor qualidade perceptiva.

#### 6.1.2.4 O SERVIÇO DE CODIFICAÇÃO DE TAXA DE BITS ADAPTÁVEL

Após o vídeo ser submetido ao serviço de *upscaling*, a etapa seguinte é a codificação em $m$ taxas de bits, permitindo que a audiência experimente sessões adaptadas aos recursos disponíveis. A quantidade de taxas de bits a serem codificadas representa um *trade-off* que depende dos recursos computacionais disponíveis na MEC. Por exemplo, um grande número de taxas de bits demanda mais tempo de processamento, o que pode resultar em *rebuffering* e ter um impacto negativo na QoE da audiência. No entanto, ao oferecer um maior número de taxas de bits, a audiência experimenta transições mais suaves entre diferentes taxas, o que influencia positivamente a QoE.

#### 6.1.2.5 A ENTREGA DE CONTEÚDO COM TAXA DE BITS ADAPTÁVEL NAS MECS

Nesta etapa, ocorre a entrega do *streaming* ao vivo à audiência. As versões do vídeo codificadas na etapa anterior ficam disponíveis para serem consumidas pela audiência a partir de servidores localizados na rede móvel. Nessa fase, supõe-se que a qualidade da sessão - fluxo e imagens apresentadas - seja determinada pela maior largura de banda existente entre o servidor de distribuição e a audiência. Independentemente do tamanho da audiência, o conteúdo é entregue a partir de servidores nas MECs, o que evita o estabelecimento de conexões individuais com os servidores da nuvem para possibilitar a distribuição *multicasting*.

#### 6.1.2.6 VÍDEO PLAYER BASEADO EM TAXA DE ADAPTAÇÃO DE BITS

No lado cliente, o *player* de vídeo é capaz de reproduzir conteúdo com taxa de bit adaptável. Um *player* com essa capacidade é capaz alternar entre diferentes representações de um mesmo vídeo, em tempo de sessão, observando os recursos disponíveis, *e.g.*, largura de banda, capacidade de processamento e o volume de conteúdo já acessado mas não reproduzido. Sessões com taxa de bit adaptável têm o potencial de melhorar a QoE das audiências, uma vez que podem reduzir *rebuffering*, o tempo de inicialização das sessões e manter a qualidade visual compatível com os recursos de transmissão disponíveis.

## 6.2 MODELOS DE SUPER-RESOLUÇÃO APRIMORADO PARA QUALIDADE PERCEPTIVA

Para a etapa de super-resolução dos vídeos, descrita na seção 6.1.2.3, propôs-se um modelo de super-resolução de tempo real denominado *Real-time video super-resolution generative adversarial networks* (RTVSRGAN). Avaliou-se o modelo juntamente com outros quatro modelos de super-resolução que destacam-se na tarefa de *upscaling* de vídeo em tempo real.

Originalmente o problema de SR assume que as imagens em baixa resolução têm sua escala reduzida (*downscaling*) por modelo de interpolação linear bicúbica com artefato de *blur* introduzido por filtros artificiais. No entanto, em *streaming* de vídeo ao vivo não é comum o artefato de *blur*, por outro lado, artefatos de compressão são comuns, pois, os *codecs* de vídeo aplicam a compressão nos vídeos ao reduzir a taxa de bits, consequentemente, a quantidade de dados a serem transmitidos. Assim, para corresponder com maior fidelidade ao problema de *upscaling* de *streaming* de vídeo ao vivo, implementou-se os quatro modelos mencionados, tendo-se treinado os mesmos usando um conjunto de dados de vídeos com artefatos de compressão introduzidos pelo *codec H.264*. O codec foi configurado com nível de compressão 17 definido pelo parâmetro fator de taxa constante (CRF, do inglês *constant rate factor*).

Visando aprimorar a qualidade perceptiva dos vídeos restaurados, introduziu-se nos modelos o treinamento adversarial relativista [112], com destilação de conhecimento [159] e funções de erro perceptiva [104].

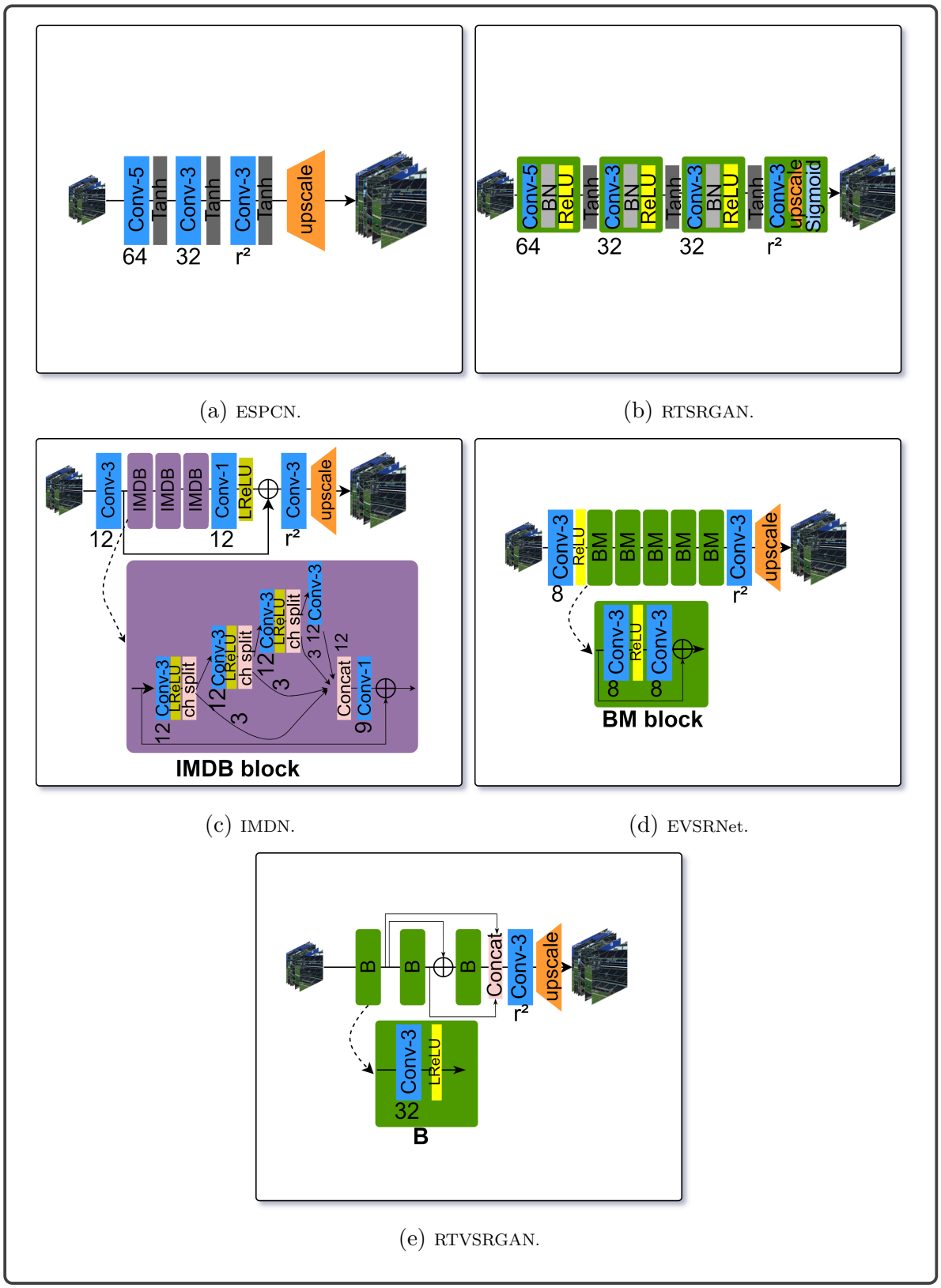
### 6.2.1 MODELOS DE SUPER-RESOLUÇÃO POR REDES NEURAIS PROFUNDAS DE TEMPO REAL

Nesta seção apresenta-se a arquitetura dos quatro modelos utilizados neste trabalho como *baselines* (Figura 33a, 33b, 33c e 33d), além da arquitetura do modelo proposto (RTVSRGAN, Figura 33e).

#### 6.2.1.1 ESPCN

O modelo *Efficient sub-pixel convolutional neural networks* (ESPCN) [87] é um *baseline* para modelos de SR e foi o primeiro a introduzir camada de subpixel para a tarefa de SR. A Figura 33a apresenta a arquitetura do modelo ESPCN que é composto por 3 camadas de convolução, com (64, 5), (32, 3) e ($r^2$, 3), onde o primeiro valor representa a quantidade de filtros e o segundo representa o tamanho do *kernel*, *i.e.*, (*filters, kernel size*). As três convoluções são seguindas de função de ativação Tanh, ao final, uma camada de subpixel realiza o *upscale*.

Figura 33 – A arquitetura dos modelos de super-resolução de tempo real avaliados nesta pesquisa.




(a) ESPCN. (b) RTSRGAN.

(c) IMDN. (d) EVSRNet.

(e) RTVSRGAN.

Fonte: De autoria própria a partir de dados dos autores.

#### 6.2.1.2 RTSRGAN

O modelo *Real-time super-resolution generative adversarial networks* (RTSRGAN) [160] foi inspirado nos modelos ESPCN [87] e ESRGAN [22]. Como ilustrado na Figura 33b, o RTSRGAN utiliza uma arquitetura similar ao ESPCN substituindo as camadas de convolução por blocos compostos por convolução, BN, ativação ReLU e entre os blocos utiliza-se a ativação Tanh. Ao final, o bloco de *upscale* é composto por uma camada de convolução, subpixel e ativação *sigmoid*.

#### 6.2.1.3 IMDN

O modelo *Information multi-distillation network* (IMDN) foi proposto no *Mobile AI 2021 Challenge* [161]. Este modelo utiliza *information multi-distillation block*s (IMDBs) em cascata e foi baseado em Hui et al.[162], sendo uma versão mais leve proposta para SR em tempo real. A arquitetura do modelo é apresentada na Figura 33c.

#### 6.2.1.4 EVSRNET

O modelo *Efficient video super-resolution network* (EVSRNet) [163] também foi apresentado no *Mobile AI 2021 Challenge* [161]. O modelo utiliza uma camada de convolução, ativação ReLU, seguidos por 5 blocos *basic modules* (BM) compostos por convolução, ativação ReLU e convolução. Ao final possui mais uma camada de convolução e uma camada de subpixel que realiza o *upscale*. A arquitetura do modelo é apresentada na Figura 33d.

#### 6.2.1.5 RTVSRGAN

O modelo proposto neste trabalho para o SR de vídeo em tempo real é o *Real-time video super-resolution generative adversarial networks* (RTVSRGAN). Ele é composto por três blocos B, que utilizam convoluções com *kernel* de tamanho (32, 3) e ativação LeakyReLU. A saída do segundo bloco B é adicionada à conexão residual proveniente do primeiro bloco B, e a saída do terceiro bloco B é concatenada com as conexões residuais dos blocos B anteriores. Após a concatenação, há uma convolução adicional e uma camada de subpixel responsável pelo *upscale*.

O modelo utiliza as conexões residuais, que abrangem desde características de baixo até alto nível, para melhorar a super-resolução. Além das conexões residuais intermediárias, o modelo reutiliza todas essas conexões realizando uma operação de concatenação antes da última convolução, que prepara as características para realizar a operação de *upscale*. O RTVSRGAN mostrou-se um modelo leve, com boa capacidade de restauração de vídeos em tempo real e com boa qualidade perceptiva. A arquitetura do modelo pode ser vista na Figura 33e.

### 6.2.2 A FUNÇÃO DE ERRO ORIENTADA A PERCEPÇÃO

Com base em modelos de SR que têm demonstrado bons resultados perceptivos [21, 46, 22, 164, 45], este trabalho propõe uma função de erro e um discriminador para treinar os modelos descritos na seção 6.2.1. A função de erro proposta é composta por quatro componentes, definidos empiricamente, que visam equilibrar a qualidade pixel a pixel e a qualidade perceptiva, a fim de produzir imagens mais realistas. Essa função de erro traz uma contribuição, introduzindo um novo componente em uma função de erro perceptiva: o erro por destilação. Esse componente tem como objetivo treinar um modelo de SR compacto, viável para ser utilizado em tempo real, com base em um modelo professor, mais profundo e com maior capacidade de aprendizado. Dessa forma, a função de erro proposta pode melhorar ainda mais o treinamento dos modelos de SR, tornando-os mais eficientes e práticos para uso em aplicações de vídeo *streaming* ao vivo.

#### 6.2.2.1 ERRO PIXEL A PIXEL

Calcula o erro médio no espaço dos pixels entre os quadros de referência e os quadros super-resolvidos. Para calcular o erro no espaço dos pixels utilizou-se a função MSE ao qual denominou-se como sendo o componente $L_{pix}$,

$$L_{pix} = \sum (G(f^{LR}) - f^{HR})^2, \tag{6.1}$$

onde $G(.)$ é a função geradora que faz a tarefa de SR; $f^{LR}$ e $f^{HR}$ são os quadros em baixa e alta resolução, respectivamente. O componente $L_{pix}$ minimiza o erro no espaço dos pixels elevando a qualidade das imagens quando avaliado por métricas objetivas baseadas em pixels.

#### 6.2.2.2 ERRO POR CARACTERÍSTICAS

O objetivo do erro por características é incentivar a rede a restaurar o conteúdo de alta frequência para melhorar a qualidade perceptiva. Estudos anteriores [21, 46, 22, 164, 45] mostraram que otimizar o erro no espaço das características de alto nível, extraídas por redes neurais, como a rede *Very deep convolutional networks* (VGG) [53], melhora a qualidade perceptiva das imagens super-resolvidas. Inspirados por esses estudos, utilizamos um componente perceptivo na função de erro que calcula o erro no espaço das características extraídas dos quadros $f^{SR}$ e $f^{HR}$ pelo modelo VGG. Esse componente de erro por características é definido por,

$$L_{fea} = \sum (VGG(f^{SR}) - VGG(f^{HR}))^2, \tag{6.2}$$

onde $VGG(.)$ é a rede neural de classificação proposta por Simonyan e Zisserman[53]; $f^{SR} = G(f^{LR})$. As características foram extraídas a partir das mesmas camadas VGG utilizadas em Wang et al.[22]. A racionalidade desse componente é restaurar informações de alta frequência para melhorar a satisfação perceptiva.

#### 6.2.2.3 ERRO POR DESTILAÇÃO

A destilação de conhecimento é um procedimento para compressão de modelos de DNN [159], no qual um modelo pequeno é treinado para corresponder a um modelo grande, pré-treinado. O conhecimento é destilado do modelo grande para o pequeno, minimizando uma função de erro. Utilizou-se um modelo mais profundo $T(.)$ treinado de maneira *offline* no agrupamento de vídeos com similaridade de conteúdo, descrito na seção 6.1.1.2, para servir de professor para o modelo mais compacto $G(.)$ e, realizou-se a destilação de conhecimento usando a seguinte função de erro,

$$L_{dis} = \sum |G(f^{LR}) - T(f^{LR})|, \tag{6.3}$$

onde $G(.)$ representa o modelo gerador e $T(.)$ o modelo professor, que destila o conhecimento para $G(.)$.

A razão para incluir esse componente é melhorar a qualidade das imagens restauradas usando o modelo $T(.)$ como referência. Em outras palavras, o objetivo é aprimorar a qualidade das imagens através de métricas baseadas em pixels, já que $T(.)$ foi treinado com a função de erro definida na Equação (6.1). Além disso, como $T(.)$ foi treinado em um conjunto de vídeo com conteúdo similar, ele sabe super-resolver vídeos daquele conjunto, e ensino o novo modelo, como se trouxesse informações relevantes para o modelo aprendiz, que são temporal e espacialmente distantes.

#### 6.2.2.4 ERRO ADVERSARIAL RELATIVISTA

O quarto componente é o erro adversarial relativista [112]. Esta função é usada no treinamento adversarial, onde quadros referência ($f^{HR}$) e quadros restaurados ($f^{SR}$) são avaliados pela rede discriminadora. A função estima a probabilidade de que os quadros de referência fornecidos sejam mais realistas do que quadros restaurados amostrados aleatoriamente. A seguir, defini-se a função de erro adversarial relativística,

$$\begin{aligned} L_{RaD} = &- \mathbb{E}_{f^{HR}} \left[ \log \left( 1 - D_{RaD} \left( f^{HR}, G(f^{LR}) \right) \right) \right] \\ &- \mathbb{E}_{G(f^{LR})} \left[ \log \left( D_{RaD} \left( G(f^{LR}), f^{HR} \right) \right) \right] \end{aligned} \tag{6.4}$$

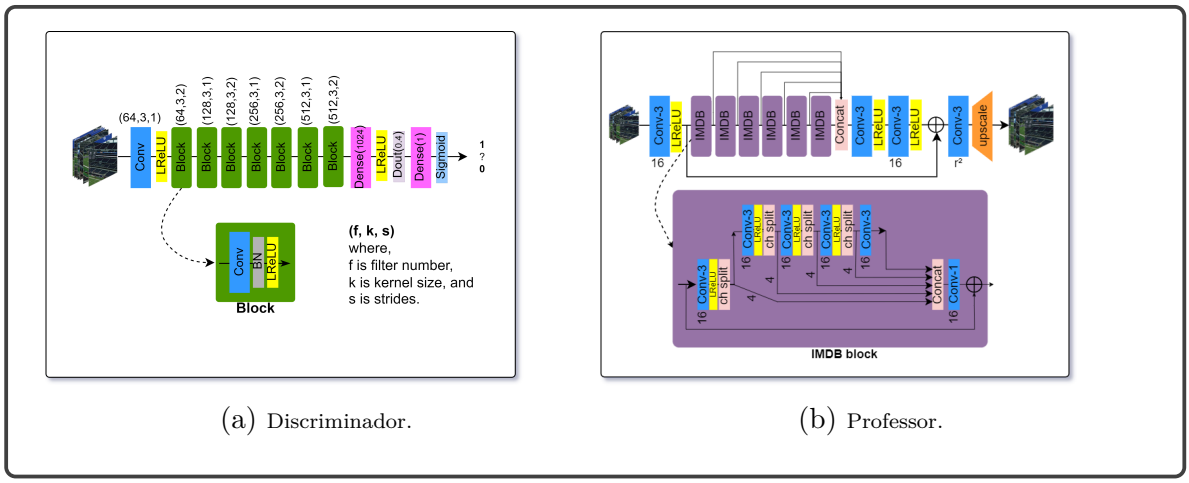
A racionalidade dessa função de erro adversária é incentivar o gerador a aprender a restaurar quadros mais realistas.

#### 6.2.2.5 ERRO GERAL

A função de erro geral é definida somando os componentes e atribuindo pesos para valorar a contribuição de cada componente na aprendizagem do modelo $G(.)$,

$$L_G = \alpha L_{pix} + \eta L_{fea} + \lambda L_{dis} + \mu L_{RaD} \tag{6.5}$$

Figura 34 – A arquitetura dos modelos discriminador $D(.)$ e professor $T(.)$.




(a) Discriminador. (b) Professor.

Fonte: De autoria própria.

onde $\alpha$, $\eta$, $\lambda$ e $\mu$ são pesos que equilibram a contribuição de cada um dos componentes, *e.g.*, aumentando o valor de $\alpha$ e $\lambda$ o modelo tende a melhorar a qualidade para métricas pixel a pixel, por outro lado, o aumento dos valores de $\eta$ e $\mu$ melhoram a qualidade para métricas perceptivas.

### 6.2.3 TREINAMENTO ADVERSARIAL COM DESTILAÇÃO DE CONHECIMENTO

Para treinar os modelos orientados a percepção visual utilizou-se três redes neurais: uma rede geradora $G(.)$, uma rede discriminadora $D(.)$ e, uma rede professora $T(.)$.

A rede geradora $G(.)$ é a parte que é utilizada para realizar a SR, *i.e.*, no serviço de SR dos vídeos somente esta parte da rede é utilizada. Por outro lado, na fase de treinamento adversarial dos modelos, além da rede $G(.)$, são utilizadas as outras duas redes, a rede discriminadora $D(.)$ e a rede professora $T(.)$, que servem para auxiliar no treinamento da rede $G(.)$.

A rede discriminadora $D(.)$ é utilizada no treinamento adversarial e contribui para que a rede geradora aprenda a restaurar quadros mais realistas, pois enquanto $D(.)$ quer se aperfeiçoar em discriminar os quadros restaurados dos quadros de referência, $G(.)$ é motivada a se aperfeiçoar a restaurar quadros cada vez mais realistas, de maneira que $D(.)$ seja confundido e passe a considerar os quadros restaurados como reais.

A rede professora $T(.)$, previamente treinada com função de erro pixel a pixel, também é utilizada na fase de treinamento adversarial. Tem o objetivo de equilibrar a rede $G(.)$ a manter boa acurácia para métricas baseadas em pixels, destilando seu conhecimento para a rede $G(.)$, uma vez que ela é um modelo mais profundo e tem mais parâmetros e maior poder de aprendizagem do que a rede $G(.)$.

Como rede geradora utilizou-se os modelos apresentados na seção 6.2.1. Uma

avaliação da qualidade das imagens restauradas pelos modelos encontra-se na seção 6.4.2.4. A arquitetura da rede discriminadora está ilustrada na Figura 34a e foi inspirada em trabalhos anteriores [21, 22, 8].

A arquitetura da rede professora encontra-se na Figura 34b e é uma variação da rede IMDN [162], que utiliza multi-destilação de informação e é destaque na literatura de SR [162, 165]. O modelo professor é uma versão mais profunda e com menos filtros de convolução nos blocos do que o modelo IMDN, isso faz com que a rede seja mais leve, exigindo menos memória e processamento na fase de treino.

### 6.2.4 BASE DE DADOS

Os modelos foram treinados usando quatro bases de dados para SR, nomeados de *generic*, *game*, *sport* e *podcast*. A base de dados *generic* foi gerada a partir de 16 vídeos 4K de 60 FPS obtidos em Harmonic[166]. Utilizou-se 14 vídeos para a base de treino e 2 para a base de teste. As demais base de dados foram geradas a partir de vídeos em 4K, como segue, *game* [167, 168, 168], *sport* [169, 170, 171] e *podcast* [172, 173, 173]. Cada uma destas bases foram geradas a partir de 3 vídeos, sendo dois para treino e um para teste.

Os vídeos LR foram definidos com as resoluções 270p, 360p e 540p. Cada versão LR foi codificada com o nível de compressão 17, definido pelo parâmetro *constant rate factor* (CRF) [174] do *codec H.264*. Para cada resolução foram codificadas, a partir dos vídeos de referência, as respectivas versões em HR de acordo com os fatores de escala 2×, 3× e 4×, *i.e.*, 270p= {540p, 810p, 1080p}, 380p= {720p, 1080p, 1440p} e 540p= {1080p, 1620p, 2160p}. As versões em HR também foram codificadas com CRF 17. Todos os vídeos codificados para a base de SR foram definidos com 30 FPS.

Para a etapa de treinamento dos modelos de SR, os quadros foram preparados com subimagens LR de tamanho 36 × 36 pixels e HR $r\times(36 \times 36)$ pixels, onde $r$ é o fator de escala de *upscaling*.

## 6.3 O PROBLEMA DE MAXIMIZAÇÃO DE QOE EM VÍDEO ADAPTATIVO

Há na literatura diversas modelagens para a QoE com vídeo *streaming* que variam de acordo com as preferências do usuário. Neste estudo utilizou-se uma modelagem que tem sido utilizada para avaliar a QoE com *HTTP-based adaptive streaming* (HAS) [43, 60, 6]. A escolha dessa modelagem se deu por envolver variáveis que estão presentes em um fluxo de transmissão de vídeo ao vivo com abordagem HAS e que possam influenciar na QoE, a fim de obter maior envolvimento do usuário a longo prazo.

Dado um trace de vazão da rede $\{C_t,\ t \in [t_1, t_{K+1}]\}$ e um conjunto de qualidades de taxa de bits discretas $\{R_k,\ k \in [1, \ldots, K]\}$, investigou-se as escolhas ideais de $T_s$, o

*startup delay*, e $R_k$, a *bitrate quality*, para maximizar a QoE do usuário para todos os segmentos no horizonte temporal. Isso pode ser expresso formalmente como:

$$\begin{aligned}\max_{R_1,\ldots,R_K,T_s} \quad QoE_1^K = &\sum_{k=1}^{K} q(R_k) - \lambda \sum_{k=1}^{K-1} |q(R_{k+1}) - q(R_k)| \\ &- \mu \sum_{k=1}^{K} \left( \frac{d_k(R_k)}{C_k} - B_k \right)_+ - \mu_s T_s\end{aligned} \tag{6.6}$$

$$\text{s.t.} \qquad t_{k+1} = t_k + \frac{d_k(R_k)}{C_k} + \Delta t_k, \tag{6.7}$$

$$C_k = \frac{1}{t_{k+1} - t_k - \Delta t_k} \int_{t_k}^{t_{k+1}-\Delta t_k} C_t \, dt, \tag{6.8}$$

$$B_{k+1} = \left( \left( B_k - \frac{d_k(R_k)}{C_k} \right) + L - \Delta t_k \right)_+, \tag{6.9}$$

$$B_k \in [0,\, B_{max}], \tag{6.10}$$

$$R_k \in \mathcal{R}, \forall\, k = 1, \ldots, K, \tag{6.11}$$

onde, $q(R_k)$ e $d_k(R_k)$ são, respectivamente, a função de qualidade e o tamanho do segmento $k$ com taxa de bit $R_k$; $|q(R_{k+1}) - q(R_k)|$ é a diferença absoluta de qualidade entre o segmento $k$ e $k+1$; $\left(\frac{d_k(R_k)}{C_k} - B_k\right)_+$ é o *rebuffering time* ao baixar o segmento $k$, caso ocorra; $C_k$ é o *throughput* durante o *download* do segmento $k$; $B_k$ é a ocupação do *buffer* quando inicia o *download* do segmento $k$; $L$ é o comprimento do segmento $k$; $T_s$ é o tempo de inicialização (*startup delay*); $\Delta t_k$ é o tempo entre o término do *download* do segmento $k$ e o início do *download* do segmento $k+1$; e, $\lambda$, $\mu$ e $\mu_s$ são pesos não negativos para os componentes, *quality variations*, *rebuffering time* e *startup delay*, respectivamente. Para manter todos os componentes da equação na mesma unidade, os valores de $\mu$ e $\mu_s$ devem ser definidos na mesma unidade de taxa de bits, enquanto $\lambda$ deve ser definido como um valor inteiro em segundos.

A diretriz geral de design de *players* baseados em *adaptive bitrate streaming* (ABR) é a seguinte: i) avaliar o *key performance indicator* (KPI) das sessões de vídeo em andamento e ii) definir a taxa de bits do próximo segmento. Existem três abordagens principais para projetar algoritmos ABRs: algoritmos baseados em *buffer*, baseados em taxa de transferência e híbridos. O último avalia os estados do *buffer* e da taxa de transferência para determinar a taxa de bits do próximo segmento. A maioria dos *players* permite que o mantenedor do sistema selecione um algoritmo para agir nas decisões de escolha adaptativa. Nesse contexto, a variável estocástica é a taxa de transferência alcançada $C_t$ porque, no momento $t_k$, a taxa de bits selecionada $R_k$ é baseada nas taxas de transferência anteriores $\{C_t,\, t \leq t_k\}$. Os *players* não sabem qual será a taxa de transferência no momento $t_k$.

Portanto, eles contam com algoritmos de previsão de taxa de transferência para escolher o $R_k$ mais apropriado.

O *framework* ELiveSR aborda o problema de otimização da QoE usando um paradigma de mudança de tráfego. O *framework* move o vídeo em baixa resolução para servidores localizados na MEC, que alimenta um modelo baseado em DNN para fazer o *upscaling* desse vídeo, em seguida, entrega *streaming* baseado em HAS com resolução de fator de escala de até 4× do vídeo de entrada.

O vídeo transmitido pelo *backhaul* por ser de baixa resolução, exige menos largura de banda, assim, o desempenho geral da rede de *backhaul* aumenta. A combinação de redes de acesso de banda larga móvel de alta velocidade e redes de *backhaul* com desempenho aprimorado também cria um cenário de rede em que uma taxa de transferência estável e alta pode ser alcançada. Esses fatores são essenciais para criar sessões de vídeo com taxas de bits mais altas, mantendo a estabilidade na flutuação de qualidade e diminuindo o *rebuffering*, o que influencia diretamente na QoE.

## 6.4 AVALIAÇÃO

Nesta seção, apresenta-se as configurações, tecnologias e os resultados dos experimentos que foram realizados para avaliar o *framework* ELiveSR. Na seção 6.4.1 encontra-se a descrição das tecnologias utilizadas na implementação do *framework*. A descrição de como os parâmetros foram configurados, como os modelos de SR foram treinados e a avaliação da qualidade dos vídeos restaurados encontra-se na seção 6.4.2. Na seção 6.4.3 apresentam-se os detalhes dos cenários para condução dos experimentos e a avaliação da QoE, latência e redução de tráfego no *backhaul*.

### 6.4.1 IMPLEMENTAÇÃO

O *framework* ELiveSR[1,2] foi implementado utilizando as seguintes tecnologias. O cliente de transmissão foi implementado em python e utiliza o FFmpeg [175]. Os modelos de SR e processamento dos quadros foram implementados em python utilizando o *framework* Tensorflow [176]. Os servidores de ingestão e distribuição foram implementados sobre o NGINX Open Source [177] e FFmpeg [175]. Utilizou-se o protocolo *real time messaging protocol* (RTMP) para *streaming* das mídias entre os servidores. Como *player*, no lado da audiência, utilizou-se o Dash.js [178] com implementação de um coletor de *logs* para métricas de QoE. Os canais de redes foram emulados utilizando a ferramenta Mahimahi [179].

Para treinar e testar os modelos de SR utilizou-se um computador *desktop* com processador Intel(R) Core(TM) i7-8700, 3.20GHz, 32GB de memória e GPU NVIDIA

[1] https://github.com/jlfilho/LiveSR
[2] https://github.com/jlfilho/sr-tf2

GeForce GTX 1070 Ti com 8GB de memória. Para realizar os experimentos com o *framework* ELiveSR utilizaram-se três computadores com as seguintes configurações, Processador Intel(R) Core(TM) i7-7700, 3.60GHz, 64GB de memória e GPU NVIDIA GeForce GTX 1080 Ti com 11GB de memória. Os computadores foram interligados utilizando rede gigabits.

### 6.4.2 AVALIAÇÃO DOS MODELOS DE SUPER-RESOLUÇÃO

Nesta seção apresenta-se como foi realizado o treinamento dos modelos de super-resolução e os resultados em relação à qualidade e tempo de execução para realização de SR dos vídeos.

#### 6.4.2.1 TREINAMENTO PIXEL A PIXEL

Primeiro treinaram-se os modelos, a parte geradora $G(.)$ e a parte professor $T(.)$, com a base de dados *generic* e função de erro MSE por 200 épocas, com *batch size* 32, otimizador Adam, com taxa de aprendizagem iniciado em $10^{-3}$, sendo reduzido pelo fator de $10^{-1}$ a cada 100 épocas.

#### 6.4.2.2 TREINAMENTO SENSÍVEL AO CONTEÚDO

Após o treinamento inicial píxel a píxel, realizou-se um segundo treinamento píxel a píxel com os mesmos parâmetros da fase anterior. No entanto, foram treinados modelos separados para cada agrupamento de vídeos, como por exemplo, "*game*", "*sport*" e "*podcast*". Esses modelos foram chamados de modelos sensíveis ao conteúdo (CA, do inglês *content-aware*), pois foram treinados em bases de dados agrupadas por similaridade de conteúdo. No início do treinamento de cada modelo, os pesos da etapa anterior foram carregados, permitindo que o modelo aprimorasse seus pesos com base nos agrupamentos de vídeos similares durante o treinamento.

#### 6.4.2.3 TREINAMENTO ORIENTADO A PERCEPÇÃO

Após as duas fases anteriores, treinaram-se os modelos de maneira adversarial. Nesta fase utilizou-se a função de erro orientada a percepção definida na Equação (6.5). Empiricamente adotou-se os seguintes valores dos pesos para as constantes da Equação (6.5), $\alpha = 2 \times 10^{-3}$, $\eta = 5 \times 10^{-5}$, $\lambda = 10^{-5}$ e $\mu = 7 \times 10^{-3}$. Antes do treinamento adversarial os pesos do treino CA foram carregados na rede geradora, assim como na rede professora. Isso contribuiu para que os modelos iniciassem com uma boa acurácia para as métricas pixel a pixel, e, ao longo do treinamento adversarial, aprimorasse a qualidade perceptiva. Nesta fase treinaram-se os modelos por 200 épocas com *batch size* 8, otimizador Adam com taxa de aprendizagem inicial de $10^{-3}$, sendo reduzida a cada 100 épocas pelo fator de $10^{-1}$.

#### 6.4.2.4 QUALIDADE DOS VÍDEOS

Para avaliar a qualidade dos vídeos restaurados pelos modelos descritos na seção 6.2.1 adotou-se 2 métricas de qualidade de imagens, sendo uma delas pixel a pixel (RMSE) e outra perceptiva (PI [46]).

Os resultados apresentados nas Figuras 35 e 36 são para as versões de vídeos LR de 270p e *upscaling* com fator de 4×, ou seja, os vídeos restaurados ficam com a resolução de 1080p. Para diferenciar os nomes dos modelos com e sem treino adversarial utilizou-se o prefixo P para a versão com treino adversarial, o prefixo G para modelos originalmente GAN, porém com o gerador treinado sem abordagem adversarial, e sem prefixo para modelos originalmente não GANs também com treino sem abordagem adversarial. O sufixo *ca* se refere a modelos com treinamento sensível a conteúdo. Na Tabela 12 está descrito o significado de cada uma das siglas.

Tabela 12 – Detalhamento da nomenclatura dos modelos.

| Sigla | Descrição |
|---|---|
| X_ca | Modelo ***c**ontent-**a**ware* (sensível a conteúdo) treinado em um *cluster* com similaridade de conteúdo. |
| P_X_ca | Modelo **p**erceptivo que teve treinamento adversário. |
| G_X_ca | É a parte **g**eneradora de uma rede GAN sem treinamento adversário. |

Fonte: De autoria própria.

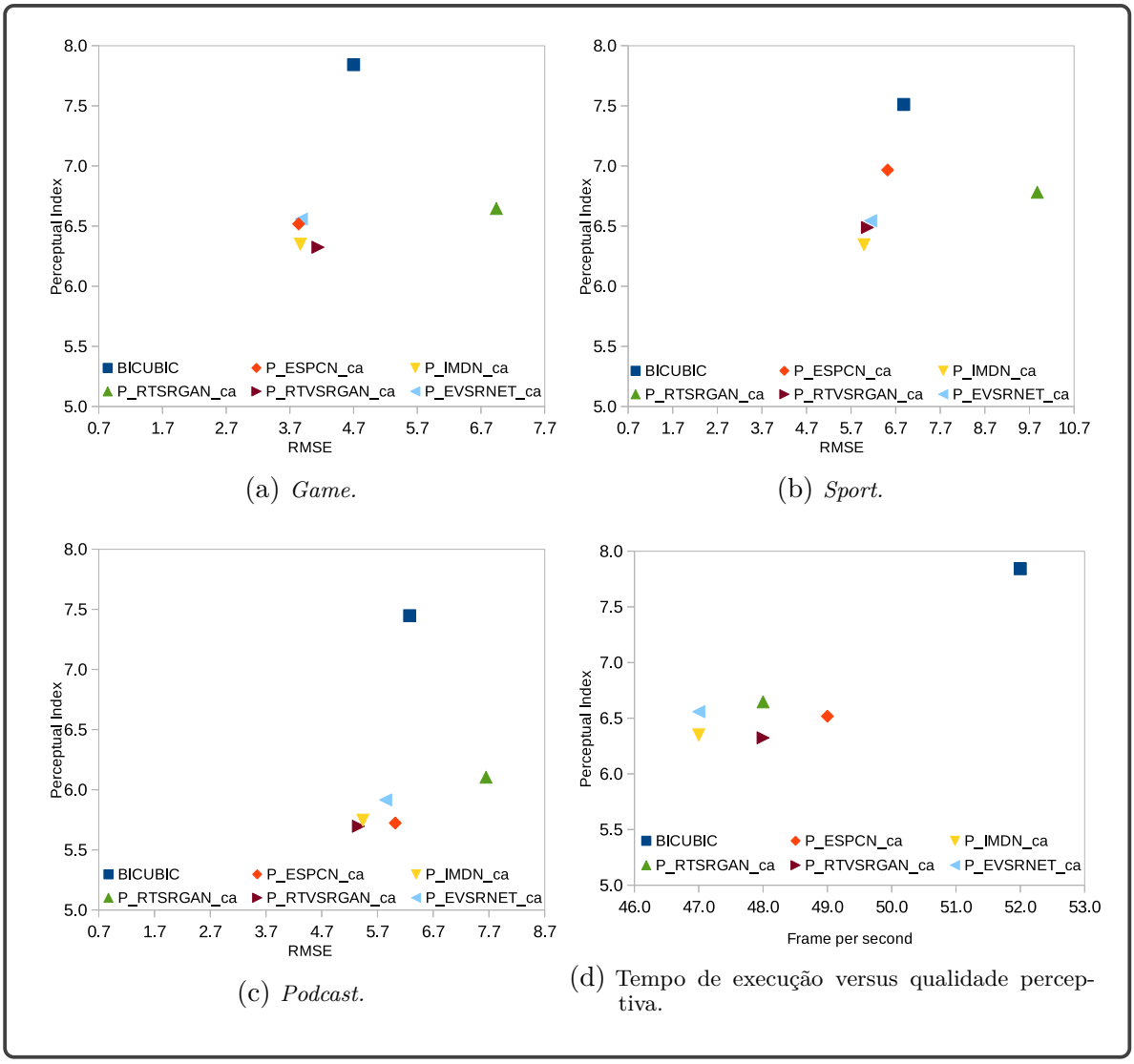
Foi adotado como *baseline* um modelo clássico da literatura de SR, o método de interpolação bicúbica, que não utiliza DNN. Esse modelo possui implementação nas principais ferramentas de visão computacional, como Matlab e OpenCV. Nas Figuras 35a, 35b e 35c, são apresentados os valores das métricas *root-mean-square error* (RMSE) (erro pixel a pixel) e PI (erro perceptivo). Para ambas as métricas, RMSE e PI, quanto mais próximo de zero, melhor é o resultado.

Analisando a qualidade perceptiva com a métrica PI, observa-se que na base de dados *game* (Figura 35a) os modelos P_RTVSRGAN_ca e P_IMDN_ca alcançaram melhores resultados. Na base de dados *sport* (Figura 35b) os melhores resultados ficaram para os modelos P_IMDN_ca, P_RTVSRGAN_ca e P_EVSRNET_ca, enquanto que, na base *podcast* (Figura 35c) os melhores resultados foram alcançados pelos modelos P_RTVSRGAN_ca, P_ESPCN_ca e P_IMDN_ca.

Avaliando a qualidade com a métrica pixel a pixel RMSE, na base de dados *game* (Figura 35a) os modelos P_IMDN_ca, P_ESPCN_ca e P_EVSRNET_ca apresentaram melhor resultado. Os modelos P_IMDN_ca, P_RTVSRGAN_ca e P_EVSRNET_ca aucançaram melhor resultado na base *sport* (Figura 35b) e na base *podcast* (Figura 35c) os modelos P_RTVSRGAN_ca e P_IMDN_ca obtiveram o melhor resultado.

Avaliou-se também a qualidade perceptiva vs o tempo de execução dos modelos.

Figura 35 – a) a c) RMSE vs. PI para os modelos de super-resolução usando as bases de dados de *games, sports e podcast.* d) O tempo de execução em quadros por segundo vs. índice perceptivo.



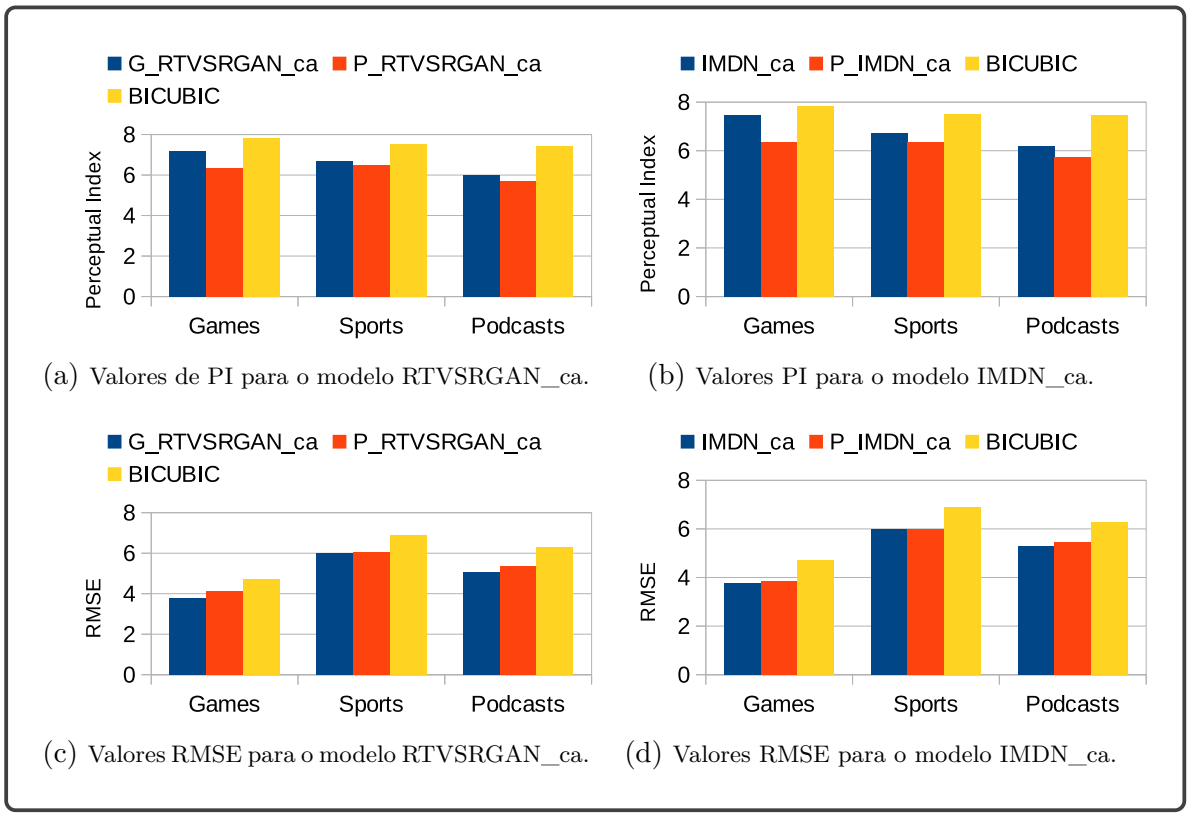

(a) *Game.* (b) *Sport.* (c) *Podcast.* (d) Tempo de execução versus qualidade perceptiva.

Fonte: De autoria própria.

Na Figura 35d plotou-se o valor de PI vs o tempo de execução em FPS. Observa-se que, os modelos baseados em DNN foram mais lentos do que o modelo de interpolação bicúbica (Bicubic), por outro lado, os modelos baseados em DNN restauraram vídeos com melhor qualidade perceptiva do que o modelo Bicubic. Os modelos mais lentos foram P_IMDN_ca e P_EVSRNET_ca com 47 FPS e o mais rápido foi o modelo Bicubic com 52 fps. Levando em consideração o fator qualidade perceptiva e o fator tempo de execução, o modelo P_RTVSRGAN_ca foi o que apresentou melhor desempenho, atingindo melhor qualidade e um tempo de execução de 48 FPS.

Para saber se o treinamento adversarial com destilação de conhecimento apresentou alguma melhoria perceptiva na qualidade dos vídeos restaurados, selecionou-se dois modelos

Figura 36 – Comparação dos valores das métricas PI e RMSE para treinamento por função de erro pixel a pixel e perceptiva (**P**_X_ca).




(a) Valores de PI para o modelo RTVSRGAN_ca.

(b) Valores PI para o modelo IMDN_ca.

(c) Valores RMSE para o modelo RTVSRGAN_ca.

(d) Valores RMSE para o modelo IMDN_ca.

Fonte: De autoria própria.

que obtiveram destaque nas avaliações, o RTVSRGAN e IMDN, para analisar a qualidade perceptiva (PI) e pixel a pixel (RMSE) antes e depois do treinamento adversarial com destilação de conhecimento.

Como pode ser observado nas Figuras 36a e 36b, os modelos treinados com função de erro perceptiva (**P**_X_ca) apresentaram melhorias na qualidade conforme medido pela métrica PI, em comparação com o treinamento baseado no erro píxel a píxel (representado em azul). No entanto, como pode ser observado nas Figuras 36c e 36d, os valores da métrica RMSE tiveram uma leve piora com o treinamento baseado em erro perceptivo, indicando que a melhoria na qualidade perceptiva resultou em uma perda na qualidade píxel a píxel. Esse *trade-off* entre métricas perceptivas e métricas de erro píxel a píxel também foi observado em trabalhos anteriores [21, 46, 45]. No entanto, no *framework* proposto, observou-se que as perdas na qualidade píxel a píxel não foram significativas, devido ao ajuste dos pesos dos componentes $\alpha$ e $\lambda$ na Equação (6.5), utilizados na fase de treinamento dos modelos.

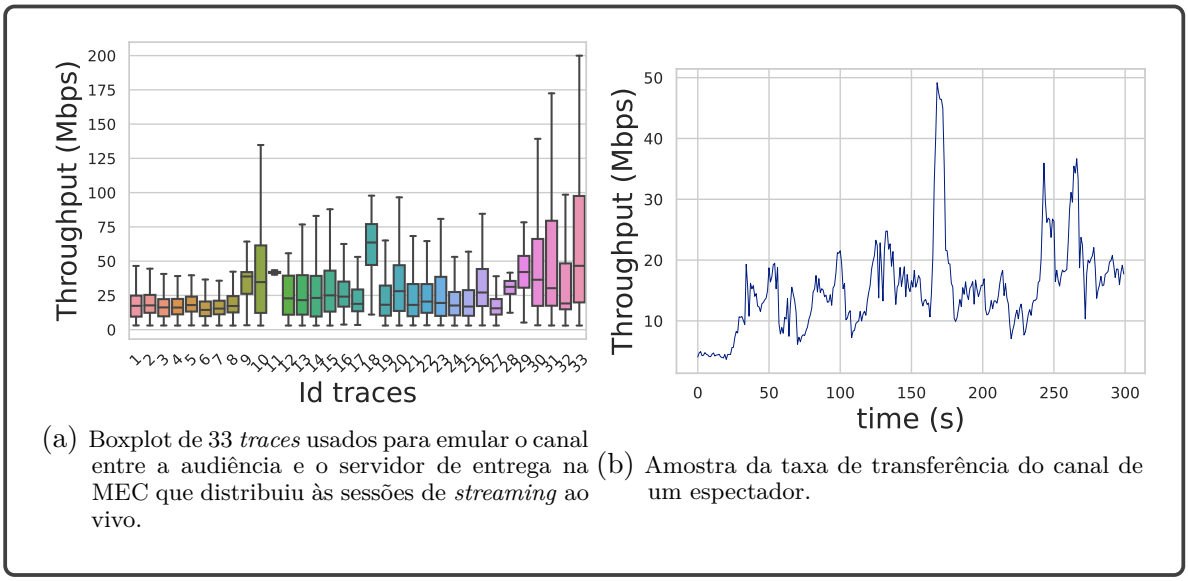
### 6.4.3 AVALIAÇÃO DA ENTREGA POR TRANSMISSÃO AO VIVO

Nesta seção apresentam-se as configurações, os cenários dos experimentos para transmissão do *streaming* de vídeo ao vivo, assim como os resultados e avaliação da QoE, latência e redução de tráfego no *backhaul* impactado pelo uso do *framework* ELiveSR.

#### 6.4.3.1 BASE DE DADOS DE VAZÃO DA REDE

Para realizar os experimentos utilizou-se uma base de dados [180] de registros de taxa de transferência (*throughputs traces*) coletados de rede móvel real, incluindo 5G. Os *traces* foram coletados a cada 1 segundo utilizando dois padrões de mobilidade (*static* e *car*) e três fontes diferentes (*file download*, *Netflix* e *Amazon Prime*). Isso permitiu que a base de dados apresentasse uma diversidade de taxa de transferência e permitisse emular um comportamento realista do canal.

Figura 37 – Traces das taxas de transferência utilizadas para emular o canal entre o servidor de entrega da MEC e a audiência que assistiu o streaming ao vivo.




(a) Boxplot de 33 *traces* usados para emular o canal entre a audiência e o servidor de entrega na MEC que distribuiu às sessões de *streaming* ao vivo.

(b) Amostra da taxa de transferência do canal de um espectador.

Fonte: De autoria própria.

Filtraram-se os *traces* com taxa de transferência entre 3Mbps e 200Mbps e mediana maior ou igual 13Mbps, para que se pudesse emular nos experimentos uma rede móvel de alta vazão, tal como uma rede 5G. Selecionou-se aleatoriamente uma amostragem com 33 *traces*. Na Figura 37a apresenta-se o *boxplot* das amostras nas quais se observa variabilidade entre os *traces*, com medianas no intervalo de 14,4 a 46,6 Mbps. A maioria dos *traces* apresenta taxa de transferência menor que 50Mbps, apenas 4 *traces* possuem pelo menos uma medida maior ou igual a 100Mbps. A Figura 37b apresenta 300 segundos do trace de *id* 8, no qual pode-se observar o comportamento da taxa de transferência de um canal ao longo do tempo. Observa-se o padrão dente de serra, que é característico do controle de congestionamento do protocolo *transmission control protocol* (TCP).

#### 6.4.3.2 CENÁRIOS DOS EXPERIMENTOS

Avaliou-se a qualidade de experiência do usuário, definida pela equação 6.6, considerando o seguinte. O serviço de *upscaling*, o serviço de codificação e o servidor de entrega adaptável estão localizados na MEC. Os vídeos transmitidos através da porção *backhaul* da rede são versões em baixa resolução, 270p, 360p e 540p. Na parte *fronthaul* da rede, a taxa de transmissão da audiência até o servidor de distribuição localizado na MEC foi emulada utilizando os 33 *traces* apresentados na seção 6.4.3.1. Nesta rede, os vídeos são entregues seguindo uma abordagem adaptativa com uso de DASH.

A porção *backhaul* da rede, que conecta o servidor de entrega ao núcleo da Internet, foi emulada utilizando 3 perfis de vazão: 1Mpbs, 1,5Mbps e 2,5Mbps. Com base nesses perfis, foram executados os experimentos seguindo 6 cenários diferentes, conforme apresentados na Tabela 13.

Os cenários 270p, 360p e 540p são *baselines*, nesses cenários os vídeos não são reescalados pelo modelo de SR, mas são codificados em 4 representações adaptativas, conforme apresentados na Tabela 13. Já nos cenários 270p 4×, 360p 3× e 540p 2× os vídeos são reescalados pelo modelo de SR antes de serem codificados para as versões adaptativas, isso permitiu avaliar o quanto o modelo de SR poderia impactar na melhoria da QoE à audiência.

Tabela 13 – Cenários dos experimentos.

| **Cenários** | **Canal Backhaul** | **Fator Escala** | **Vídeo LR Recebido da Nuvem** | **Vídeo HR após SR** | **Versões Adaptativas após codificação** |
|---|---|---|---|---|---|
| 270p | 1Mbps | - | 270p / 400Kbps | - | {144p, 180p, 240p, 270p} |
| 270p 4× | 1Mbps | 4× | 270p / 400Kbps | 1080p / 4300Kbps | {360p, 540p, 720p, 1080p} |
| 360p | 1,5Mbps | - | 360p / 700Kbps | - | {180p, 240p, 270p, 360p } |
| 360p 3× | 1,5Mbps | 3× | 360p / 700Kbps | 1080p / 4300Kbps | {360p, 540p, 720p, 1080p} |
| 540p | 2,5Mbps | - | 540p / 1600Kbps | - | {240p, 270p, 360p, 540p } |
| 540p 2× | 2,5Mbps | 2× | 540p / 1600Kbps | 1080p / 4300Kbps | {360p, 540p, 720p, 1080p} |

Fonte: De autoria própria.

Nos experimentos conduzidos neste estudo o Algoritmo 1 teve os valores de $\tau$ e $\beta$ definidos para $3 \times 10^{-1}$. Quando o link de *backhaul* era de 1Mbps, o $v_{bitrate}$ foi de 400Kbps que corresponde a resolução 270p, quando o link *backhaul* foi de 1,5Mbps o $v_{bitrate}$ foi de 700Kbps (360p) e quando o link de *backhaul* teve vazão de 2,5Mbps, o $v_{bitrate}$ foi de 1600Kbps (540p). Quanto ao fator de escala, a resolução 270p teve o fator de escala 4×, a resolução 360p teve fator de escala 3× e a resolução 540p teve fator de escala 2×. Após o vídeo ser reescalado para uma maior resolução, foi codificado em 4 representações com diferentes taxas de bits, como apresentado na Tabela 13.

#### 6.4.3.3 ALGORITMOS DE ADAPTAÇÃO DE TAXA DE BITS

Nos experimentos usaram-se três algoritmos ABR que acompanham o *player* dash.js para criar e gerenciar sessões de vídeo.

BOLA [181]: é um algoritmo baseado em *buffer* que usa a técnica de otimização *Lyapunov* para selecionar taxas de bits, sujeita a seguinte restrição: reduzir *rebuffering* sem afetar o QoE do usuário. Este algoritmo aborda o problema de adaptação de taxa de bits de vídeo como um problema de maximização de utilidade que incorpora métricas de QoE, por exemplo, tempo de *rebuffering*.

LoL+ [182]: visa reduzir a latência da transmissão de vídeo adaptativo ao vivo, basea-se em informações do histórico da sessão e estimativas futuras de taxa de transferência e de nível de *buffer*. Utiliza métodos heurísticos e de aprendizagem de máquina para realizar a adaptação da taxa de bits. Seu objetivo é maximizar a QoE do usuário, combinando fatores como taxa de bits, tempo de *rebuffering*, latência e velocidade do *player*.

Learn2Adapt-LowLatency (L2A-LL) [183]: foi inspirado no algoritmo *low-on-latency* (LoL), mas conta com otimização convexa *online* (OCO, do inglês *online convex optimization*) para adaptar a taxa de bits das sessões de vídeo ao vivo.

#### 6.4.3.4 PARÂMETROS DO VÍDEO UTILIZADO NO EXPERIMENTO

Na transmissão ao vivo foi utilizado um vídeo de 10 minutos de futebol americano da base de dados Harmonic [166]. A codificação do vídeo no cliente de transmissão foi com taxa de bits de 1600 Kbps e resolução 540p.

#### 6.4.3.5 AVALIAÇÃO DA QUALIDADE DE EXPERIÊNCIA

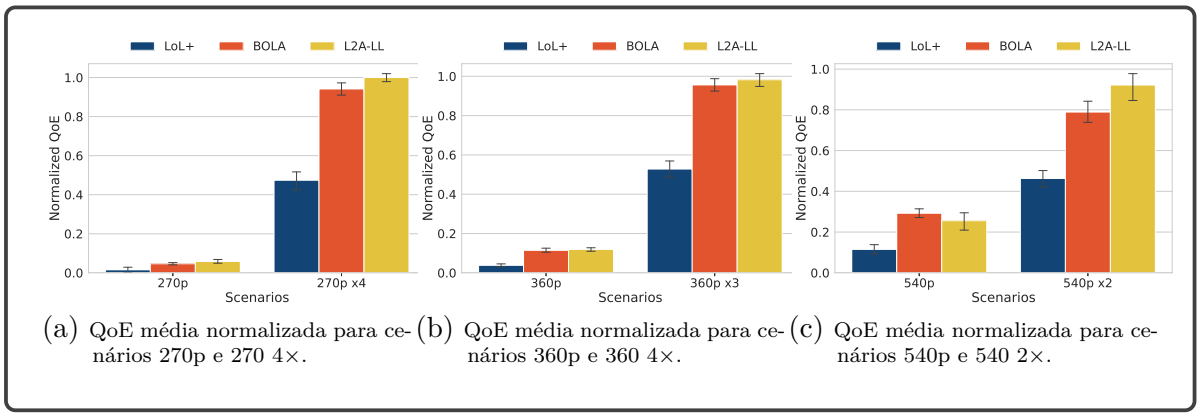
Avaliou-se a QoE utilizando a Equação (6.6). No serviço de *upscaling* de vídeo utilizou-se o modelo RTVSRGAN, descrito na seção 6.2.1.5, por ter se mostrado eficiente na restauração de vídeos em tempo real apresentando boa qualidade perceptiva.

Os experimentos mostraram que o *framework* ELiveSR apresentou melhoria na QoE ao comparar os cenários com e sem o uso de SR para os três algoritmos de adaptação avaliados.

Na Figura 38 são apresentados os 6 cenários com os valores médios da QoE normalizados pela equação $nQoE = \frac{QoE}{\max\left(\left[\overline{QoE_{BOLA}}, \overline{QoE_{LoL+}}, \overline{QoE_{L2A-LL}}\right]\right)}$. Na Figura 38a encontra-se a nQoE para o cenário 270p (sem SR) e 270p 4× (SR com fator de escala 4×). Observa-se que o cenário com SR apresentou melhoria com diferença de 0,44, 0,84 e 0,90 para os algorítimos LoL+, BOLA e L2A-LL, respectivamente.

Nos cenários 360p e 360p 3×, Figura 38b, observa-se o mesmo comportamento de melhoria, com diferença de 0,45, 0,78 e 0,80 para os algoritmos LoL+, BOLA e L2A-LL,

Figura 38 – QoE média normalizado com intervalo de confiança de 95% para os 6 cenários e ABR L2A-LL, BOLA e LOL+.




(a) QoE média normalizada para cenários 270p e 270 4×. (b) QoE média normalizada para cenários 360p e 360 4×. (c) QoE média normalizada para cenários 540p e 540 2×.

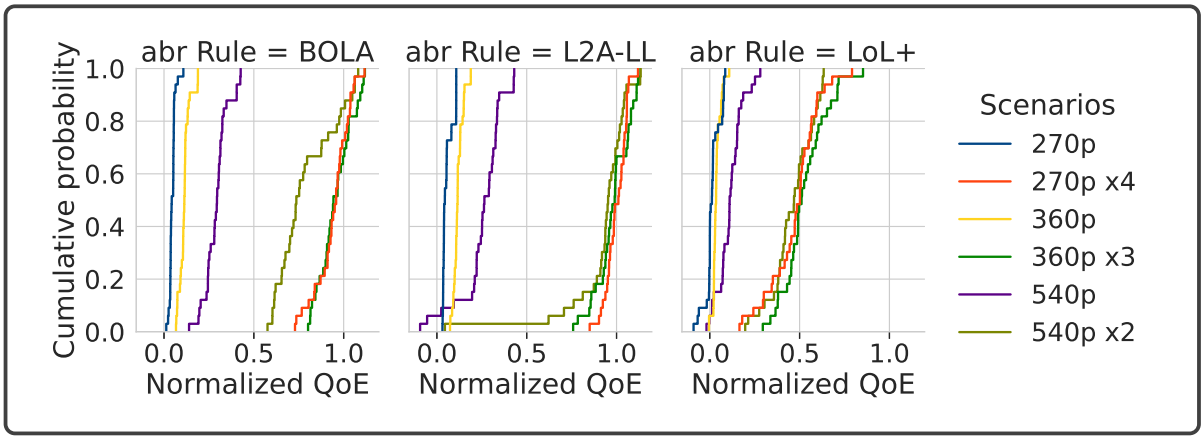
Fonte: De autoria própria.

respectivamente. Também nos cenários 540p e 540p 2×, embora menor que nos cenários anteriores, observou-se melhoria de 0,28, 0,43 e 0,58 para os algoritmos LoL+, BOLA e L2A-LL, respectivamente.

Ao analisar a QoE em relação aos algoritmos de adaptação, constatou-se que em todos os cenários com uso de SR (270p 4×, 360p 3× e 540p 2×) o algoritmo L2A-LL alcançou maior valor de nQoE, enquanto que o algoritmo LoL+ obteve os menores valores.

Nos cenários sem SR (270p, 360p e 540p) o algoritmo LoL+ também obteve o pior resultado com valores de nQoE 0,06, 0,09 e 0,21, respectivamente. Os algorítimos L2A-LL e BOLA ficaram empatados nos cenários 270p (0,09) e 360p (0,17), enquanto que no cenário 540p o algoritmo BOLA obteve o melhor resultado com 0,38.

Figura 39 – A Função de distribuição acumulada de QoE normalizada para os 6 cenários.




Fonte: De autoria própria.

Para analisar a distribuição de probabilidades dos valores de nQoE apresenta-se na Figura 39 a função de distribuição acumulada (FDA) para a nQoE dos 33 espectadores

agrupados por algoritmo de adaptação. Para cada algoritmo de adaptação observa-se dois agrupamentos de probabilidades de nQoE, a dos cenários sem SR e a dos cenários com SR.

Nos cenários em que a SR não é utilizada, observa-se que os valores da nQoE estão concentrados no intervalo com nQoE inferior a 0,5. Isso significa que esses cenários apresentam uma probabilidade de 100% de os valores de nQoE serem menores que 0,5. Essa situação ocorre devido às resoluções e taxas de bits mais baixas dos vídeos nessas condições, que afetam a QoE mesmo quando a taxa de transferência da rede de borda é de boa qualidade.

Por outro lado, nos cenários com uso de SR, os valores de nQoE são mais densos no intervalo de nQoE maior que 0,5, *e.g.*, o algoritmo BOLA e L2A-LL apresentam-se com probabilidade de 100% ter uma nQoE maior que 0,5, exceto para L2A-LL no cenário 540p 2× que apresentou probabilidade de 0,03 da nQoE ser menor do que 0,5. Mesmo o algoritmo LoL+, que apresentou o pior resultado, teve probabilidade de 100% ter uma nQoE maior do que 0,3. A melhoria na QoE ocorre porque nesses cenários os vídeos possuem versões com maior resolução e maior taxa de bits, com isso, os algoritmos de adaptação aproveitam a alta taxa de transferência da rede de borda para entregar segmentos de vídeos de maior taxa de bits, com isso, elevando a QoE.

#### 6.4.3.6 LATÊNCIA POR SEGMENTO DA REDE DE TRANSMISSÃO

Na Tabela 14 apresentam-se as medidas de latência por segmento de processamento ou transmissão. O primeiro segmento é do cliente de transmissão até a mídia ficar disponível para ser puxada do servidor de distribuição na nuvem, nesse seguimento a latência foi de 2 segundos.

O segundo segmento compreende a latência transcorrida desde a saída do servidor de distribuição na nuvem até a mídia ser reescalada pelo serviço de *upscaling*, essa latência foi de 3 segundos. Por último, apresenta-se a latência desde a saída do serviço de *upscaling* até a mídia ficar acessível no servidor de entrega na MEC, já codificada em 4 versões com diferentes taxas de bits. Nesse segmento a latência foi de 7 segundos.

Tabela 14 – Latência por segmento de transmissão.

| **Segmento de transmissão** | **Latência** |
|---|---|
| Do cliente de transmissão (fonte) até o servidor de distribuição na nuvem | 2s |
| Do servidor de distribuição na nuvem para a saída do serviço *upscaling* na MEC | 3s |
| Serviço de codificação | 7s |

Fonte: De autoria própria.

Esse estudo não teve como objetivo otimizar a latência, por isso, foi utilizada a computação paralela em GPU apenas no serviço de *upscaling* para resolver a SR. No entanto, pode-se observar que o serviço de codificação foi o que introduziu a maior latência

no fluxo (terceiro segmento), isso ocorreu porque foram codificadas 4 versões do vídeo em uma mesma máquina com uso apenas de CPU. Essa latência poderia ser otimizada se fosse empregado computação paralela tanto para o serviço de codificação como também para o serviço de *upscaling*, sendo possível serem utilizadas mais de uma GPU nesse serviço.

A latência do servidor de entrega na MEC até a audiência foge do escopo deste estudo e pode variar de acordo com a estratégia dos algoritmos de adaptação e a característica da rede de acesso do usuário final.

#### 6.4.3.7 REDUÇÃO DE DADOS DE VÍDEOS TRANSFERIDOS PELA REDE BACKHAUL

Na Tabela 15 apresenta-se o volume de dados transferidos através da rede *backhaul* e o volume de dados que é entregue à audiência, através da rede *fronthaul*. Caso os dados não estivessem sendo entregue à audiência a partir de servidores localizados na MEC, todo o quantitativo de dados da porção *fronthaul* estaria também passando na porção *backhaul*.

No entanto, o uso do *framework* ELiveSR reduz esse volume por três razões: 1) os vídeos transferidos na porção *backhaul* da rede estão em baixa resolução e, ao chegar na MEC são reescalados por SR em até 4×; 2) apenas uma cópia do vídeo é transferido através da rede *backhaul*, uma vez que, o servidor de entrega está localizado na MEC e, somente a partir dela, são estabelecidas as sessões com à audiência; e, 3) os dados gerados pelas sessões estabelecidas pelas audiências são transferidos apenas entre servidor de entrega e a audiência, ou seja, apenas da MEC até a estação do usuário.

Na Tabela 15 encontra-se também, o percentual de redução de tráfego calculado a partir dos experimentos realizados com 33 espectadores (*traces* apresentados na seção 6.4.3.1) assistindo uma sessão de vídeo com duração de 10 minutos. Os cálculos são apresentados por cenário e por algoritmo de adaptação. Para calcular a redução de tráfego foi utilizada a Equação $TR = \left(1 - \frac{Backhaul_{data}}{Fronthaul_{data}}\right) \times 100$.

O cenário 270p 4× foi o que apresentou maior percentual de redução, chegando a 88,37% para o algoritmo L2A-LL, 87,93% para BOLA e 83,01% para o LoL+. Essa melhoria ocorreu porque o fator de escala nesse cenário foi o maior (4×), logo, a resolução do vídeo entregue à audiência foi de até 4× maior do que a resolução transferida através da rede *backhaul*.

O cenário 360p 3× ficou como o segundo melhor na redução de tráfego com 79,74%, 79,28% e 72,28% para os algoritmos BOLA, L2A-LL e LoL+, respectivamente. O cenário 540p 2× ficou com o terceiro melhor resultado na redução de dados com 72,06%, 69,58% e 54,83% para os algoritmos BOLA, L2A-LL e LoL+, respectivamente.

Observou-se que os cenários sem SR ficaram com menor redução de dados trafegados na rede *backhaul* do que os com SR. Além disso, observou-se que quanto maior o fator de escala utilizado para os modelos de SR, maior é a redução de dados transferidos

Tabela 15 – Quantidade de dados transferidos em cada segmento da rede.

| **Cenários** | **ABR** | **Número de Espectadores** | **Fator de Escala** | **Dados Enviados pelo Backhaul** | **Dados Entregues à Audiência (Fronthaul)** | **Redução de Tráfego** |
|---|---|---|---|---|---|---|
| 270p | BOLA | 33 | - | 334M | 1017,8M | 67,18% |
| | L2A | 33 | - | 334M | 1020,8M | 67,28% |
| | LoL+ | 33 | - | 334M | 689,6M | 51,57% |
| 270p 4× | BOLA | 33 | 4 | 334M | 2767,4M | 87,93% |
| | L2A | 33 | 4 | 334M | 2870,9M | 88,37% |
| | LoL+ | 33 | 4 | 334M | 1966,4M | 83,01% |
| 360p | BOLA | 33 | - | 549M | 1775,5M | 69,08% |
| | L2A | 33 | - | 549M | 1777,6M | 69,12% |
| | LoL+ | 33 | - | 549M | 1020,1M | 46,18% |
| 360p 3× | BOLA | 33 | 3 | 549M | 2710,3M | 79,74% |
| | L2A | 33 | 3 | 549M | 2649,8M | 79,28% |
| | LoL+ | 33 | 3 | 549M | 1980,2M | 72,28% |
| 540p | BOLA | 33 | - | 1,1G | 3937,0M | 59,51% |
| | L2A | 33 | - | 1,1G | 3615,7M | 59,85% |
| | LoL+ | 33 | - | 1,1G | 2435,0M | 47,11% |
| 540p 2× | BOLA | 33 | 2 | 1,1G | 2716,4M | 72,06% |
| | L2A | 33 | 2 | 1,1G | 2739,7M | 69,58% |
| | LoL+ | 33 | 2 | 1,1G | 2079,9M | 54,83% |

Fonte: De autoria própria.

através do *backhaul*. Essa redução na transferência de dados pela porção *backhaul* da rede evidencia que o ELiveSR contribui para reduzir o congestionamento no *backhaul* causado por *streaming* ao vivo, ao mesmo tempo que, ao entregar vídeos codificados com maior taxa de bits e maior resolução, eleva a QoE da audiência, uma vez que a rede de acesso da audiência apresenta alta taxa de transferência.

## 6.5 CONSIDERAÇÕES

O *framework* apresentado neste capítulo teve como objetivo explorar os recursos de computação de borda para aprimorar a qualidade perceptiva de vídeo *streaming* ao vivo com uso de super-resolução por redes neurais adversárias de tempo real, contribuindo para melhoria na qualidade de experiência da audiência dessas aplicações. Além disso, o *framework* contribui para redução de dados gerados pelas transmissões de vídeo ao vivo nas redes *backhaul*, mitigando o problema de congestionamento de tráfego neste segmento da rede.

Os resultados mostraram que a introdução da SR em um sistema de *streaming* ao vivo impacta de forma positiva na melhoria da QoE, como mostrado na seção 6.4.3.5. Além disso, o envio dos vídeos em baixa resolução e a reescala para alta resolução em servidores localizados na MEC, reduz o tráfego de vídeo transmitido nas redes de *backhaul*, seção 6.4.3.7. Esses resultados evidenciam que o uso de SR de vídeo ao vivo em computação de borda, além de melhorar a QoE da audiência, pode contribuir para mitigar o problema de redes *backhaul* com congestionamento causado por *streaming* de vídeo ao vivo, nas redes de acesso móvel ou com fio, que apresentam alta vazão.

# CAPÍTULO 7

# Conclusão

Nesta tese, foi conduzida uma pesquisa com o objetivo de investigar se as GANs poderiam ser utilizadas em aplicações de *streaming* de vídeo para reconstruir vídeos transmitidos em baixa resolução e restaurá-los para a resolução original, mantendo uma boa qualidade perceptiva. Além disso, buscou-se responder se o uso das GANs poderia melhorar a QoE dos espectadores de aplicações de vídeo *streaming*, nos quais os vídeos são transmitidos em baixa resolução e, em seguida, uma GAN é aplicada para elevar a resolução do vídeo, permitindo que os espectadores assistam aos vídeos em uma qualidade melhor do que a originalmente transmitida.

O estudo teve como foco principal a redução do tráfego nas infraestruturas de rede ao utilizar técnicas de reconstrução de vídeo de baixa resolução para alta resolução, com o objetivo de melhorar a experiência dos espectadores. Por meio da aplicação das GANs, buscou-se explorar seu potencial para aprimorar a qualidade visual dos vídeos em tempo real, o que pode contribuir para uma experiência mais imersiva e satisfatória dos usuários.

Ao abordar essas problemáticas, a pesquisa teve como intuito fornecer *insights* sobre o uso das GANs em aplicações de *streaming* de vídeo e seu impacto na melhoria da QoE dos espectadores. Os resultados obtidos podem ser relevantes para o desenvolvimento de soluções que aperfeiçoem a qualidade do vídeo em tempo real, promovendo uma experiência mais envolvente e de maior qualidade para os usuários de serviços de *streaming*.

No capítulo 5, foi apresentado um *framework* de serviço de replicação de vídeo em nuvem, utilizando um modelo de SR baseado em GAN, com o objetivo de reduzir o volume de dados de vídeo sob demanda replicados por meio das infraestruturas de redes de conexões internacionais. O problema foi formulado como uma otimização do tamanho

do vídeo, abordando o desafio de encontrar um equilíbrio entre a redução do tamanho da mídia a ser movida entre servidores original e substituto geograficamente distribuídos, visando diminuir os custos operacionais com a movimentação, ao mesmo tempo em que se mantém a qualidade perceptiva das imagens em alto padrão. Os resultados obtidos demonstraram que a combinação de SR com compressão alcança níveis satisfatórios na redução dos dados transmitidos, ao mesmo tempo em que mantém a qualidade perceptiva dos vídeos em níveis semelhantes aos vídeos de referência.

Já no capítulo 6, o estudo foi aprimorado com a aplicação de SR em tempo real por meio de GAN em *streaming* adaptativo de vídeo ao vivo. O *framework* proposto nessa etapa da pesquisa tem como objetivo a distribuição de vídeos ao vivo com aprimoramento da qualidade perceptiva por meio da super-resolução provida pela computação de borda. O objetivo é buscar melhorias na QoE dos espectadores e mitigar o problema de gargalo nos *links* de *backhaul* de redes móveis causado pelos conteúdos das aplicações de vídeo ao vivo. A qualidade percebida pelos espectadores foi modelada como um problema de maximização da QoE em vídeo adaptativo. Os resultados obtidos demonstram que o *framework* apresentou melhorias na QoE em todos os cenários com SR em comparação com os cenários sem o uso de SR, indicando que a SR pode ser incorporada nas aplicações de *streaming* de vídeo ao vivo, contribuindo para a melhoria da QoE, ao mesmo tempo em que reduz a demanda por largura de banda nos *links* de *backhaul*.

## 7.1 TRABALHOS FUTUROS

Como trabalhos futuros, nesta mesma linha de pesquisa, podem ser consideradas as seguintes ideias como promissoras:

- Aplicar modelos de super-resolução em vídeos adaptativos, utilizando a técnica de SR em segmentos transmitidos a partir da nuvem. Essa abordagem aproveitaria o poder de processamento das MECs de maneira colaborativa e hierárquica, permitindo o uso eficiente dos recursos computacionais. Dessa forma, seria possível utilizar a super-resolução sem depender de dispositivos finais equipados com GPUs, tornando a técnica mais acessível aos clientes. Essa abordagem adaptativa de super-resolução em tempo real permitiria melhorar a qualidade visual dos vídeos transmitidos, mesmo em dispositivos com capacidades de processamento limitadas. Seria necessário desenvolver algoritmos eficientes de distribuição de tarefas e processamento colaborativo entre as MECs para otimizar o desempenho e garantir uma experiência de visualização contínua.

- Outra abordagem, seguindo a ideia mencionada anteriormente, seria a combinação dos recursos computacionais da MEC com os recursos computacionais dos clientes em uma abordagem híbrida MEC-P2P. Essa abordagem visa tornar o sistema mais

robusto e escalável, aproveitando os recursos disponíveis tanto nos dispositivos de borda quanto nos dispositivos dos próprios clientes. Nessa abordagem, os dispositivos de borda, como servidores MEC, podem atuar como nós de processamento centralizados, fornecendo recursos adicionais de computação e armazenamento. Por sua vez, os dispositivos dos clientes podem contribuir com seus próprios recursos computacionais, criando uma rede descentralizada de processamento distribuído. Essa abordagem híbrida MEC-P2P oferece benefícios como maior escalabilidade, resiliência a falhas e aproveitamento eficiente dos recursos disponíveis. Além disso, a descentralização do processamento pode reduzir a latência e melhorar a eficiência do sistema como um todo. No entanto, é importante considerar desafios como a gestão de recursos, a segurança e a coordenação entre os dispositivos. Esses aspectos devem ser cuidadosamente tratados para garantir um equilíbrio adequado entre o aproveitamento dos recursos e a qualidade do serviço oferecido aos clientes.

- Além disso, as tendências futuras relacionadas a conteúdos multimídia, conforme indicado pela Cisco [151], apontam para um aumento significativo no tráfego de vídeos volumétricos, como realidade virtual (RV) e vídeos em 360º. Essas aplicações oferecem uma experiência imersiva aos usuários finais e têm potencial para gerar um grande volume de tráfego nas redes. Portanto, explorar o uso de técnicas de SR em redes de borda para vídeos volumétricos é uma direção promissora para pesquisas futuras, visando aprimorar a qualidade visual dessas aplicações e atender às demandas crescentes por conteúdos imersivos nas redes.

- Na área de técnicas de IA generativa criativa, uma direção de pesquisa interessante é explorar métodos de SR para a criação de conteúdo com uma garantia mínima de informação. Por exemplo, considerando a transmissão ao vivo de uma partida de futebol, é crucial que o gol aconteça no momento correto, executado pelo jogador certo e da forma adequada, mas sem a necessidade de garantir detalhes específicos, como a textura da grama ou o rosto de um torcedor na arquibancada. Nesse contexto, as técnicas de SR podem ser utilizadas para aumentar a resolução e a qualidade visual do conteúdo transmitido ao vivo, mantendo a essência e as informações essenciais, mas sem se preocupar com detalhes irrelevantes para a compreensão do evento em questão. Isso permite uma transmissão mais eficiente em termos de largura de banda e recursos computacionais, sem comprometer a experiência do espectador. Ao aplicar métodos de SR com garantia mínima de informação em transmissões ao vivo, é possível melhorar a visualização de eventos em tempo real, mantendo a qualidade perceptiva necessária para a compreensão e a apreciação do conteúdo. Essa abordagem criativa de IA generativa permite uma otimização inteligente dos recursos, proporcionando uma experiência visual aprimorada aos espectadores sem comprometer a transmissão ou a qualidade do evento.

- Também com uso de técnicas de IA generativa criativa, uma linha de pesquisa promissora é a criação de conteúdo quase personalizado, em que o usuário especifica suas preferências e a IA é responsável por gerar o conteúdo de acordo com essas diretrizes. Por exemplo, o usuário pode solicitar um conteúdo voltado para crianças, com enredo contendo pontos essenciais específicos e um final escolhido a partir de uma seleção pré-determinada. Além disso, o estilo de animação e as características dos personagens também podem ser especificados. Nesse contexto, a IA pode ser treinada para entender as preferências do usuário e gerar conteúdo sob medida. Ela pode aproveitar algoritmos generativos e técnicas de aprendizado de máquina para criar narrativas e animações personalizadas, levando em consideração as restrições e os requisitos definidos pelo usuário. Dessa forma, o resultado final será um conteúdo único e adaptado aos interesses e preferências do usuário, garantindo uma experiência de visualização mais personalizada. Ao desenvolver sistemas de criação de conteúdo quase personalizado, é possível explorar a capacidade da IA em gerar histórias, personagens e animações de forma criativa e dinâmica, proporcionando uma experiência de entretenimento única para cada usuário. Essa abordagem permite uma maior interação e participação do usuário na criação do conteúdo, tornando-o mais envolvente e personalizado.

Essas ideias podem servir como pontos de partida para futuras investigações e contribuir para avanços significativos na área de super-resolução de vídeos.

# Referências

1 CISCO VNI. *Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper*. [S.l.], 2019. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>. Acesso em: 15 ago. 2022. Citado 4 vezes nas páginas 21, 22, 89 e 112.

2 ZOLFAGHARI, B. et al. Content delivery networks: State of the art, trends, and future roadmap. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 53, n. 2, abr. 2020. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3380613>. Acesso em: 15 ago. 2022. Citado 2 vezes nas páginas 21 e 22.

3 Li, Z. et al. Video delivery performance of a large-scale vod system and the implications on content delivery. *IEEE Transactions on Multimedia*, v. 17, n. 6, p. 880–892, 2015. Citado na página 21.

4 Bitmovin Inc. *Per-Title Encoding*. 2020. Disponível em: <https://bitmovin.com/demos/per-title-encoding>. Acesso em: 15 ago. 2022. Citado na página 22.

5 YAN, B. et al. Livejack: Integrating cdns and edge clouds for live content broadcasting. In: *Proceedings of the 25th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2017. (MM '17), p. 73–81. ISBN 9781450349062. Disponível em: <https://doi.org/10.1145/3123266.3123283>. Acesso em: 15 ago. 2022. Citado na página 22.

6 YEO, H. et al. Neural adaptive content-aware internet video delivery. In: *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. Carlsbad, CA: USENIX Association, 2018. p. 645–661. ISBN 978-1-931971-47-8. Disponível em: <https://www.usenix.org/conference/osdi18/presentation/yeo>. Acesso em: 15 ago. 2022. Citado 5 vezes nas páginas 22, 37, 38, 39 e 125.

7 Wang, F. et al. Deepcast: Towards personalized qoe for edge-assisted crowdcast with deep reinforcement learning. *IEEE/ACM Transactions on Networking*, v. 28, n. 3, p. 1255–1268, 2020. Citado 3 vezes nas páginas 22, 39 e 112.

8 LIBORIO, J. M.; Souza, C. M.; MELO, C. A. V. Super-resolution on edge computing for improved adaptive http live streaming delivery. In: *2021 IEEE 10th International Conference on Cloud Networking (CloudNet)*. [S.l.: s.n.], 2021. p. 104–110. Citado 2 vezes nas páginas 22 e 125.

9 YEO, H.; DO, S.; HAN, D. How will deep learning change internet video delivery? In: *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*. New York, NY, USA: ACM, 2017. (HotNets-XVI), p. 57–64. ISBN 978-1-4503-5569-8. Disponível em: <http://doi.acm.org/10.1145/3152434.3152440>. Acesso em: 15 ago. 2022. Citado 5 vezes nas páginas 22, 24, 37, 38 e 39.

10 HECHT, J. The bandwidth bottleneck that is throttling the internet. *Nat.*, v. 536, n. 7615, p. 139–142, 2016. Disponível em: <https://doi.org/10.1038/536139a>. Acesso em: 10 dez. 2022. Citado 3 vezes nas páginas 22, 89 e 90.

11 CHRISTIAN, P. *Int'l Bandwidth and Pricing Trends*. [S.l.], 2018. Disponível em: <https://docplayer.net/169802807-Int-l-bandwidth-and-pricing-trends.html>. Acesso em: 10 dez. 2022. Citado 3 vezes nas páginas 23, 89 e 90.

12 Wang, Z. et al. Joint online transcoding and geo-distributed delivery for dynamic adaptive streaming. In: *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. [S.l.: s.n.], 2014. p. 91–99. Citado 2 vezes nas páginas 23 e 89.

13 IMPACTS, A. *2019 recent trends in GPU price per FLOPS*. [S.l.], 2019. Disponível em: <https://aiimpacts.org/2019-recent-trends-in-gpu-price-per-flops/>. Acesso em: 10 dez. 2022. Citado 2 vezes nas páginas 23 e 89.

14 CORPORATION, N. *Accelerated Computing And The Democratization Of Supercomputing*. [S.l.], 2018. Disponível em: <https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-product-literature/sc18-tesla-democratization-tech-overview-r4-web.pdf>. Acesso em: 10 dez. 2022. Citado 2 vezes nas páginas 23 e 89.

15 CLOUD, G. *Cheaper Cloud AI deployments with NVIDIA T4 GPU price cut*. [S.l.], 2020. Disponível em: <https://cloud.google.com/blog/products/ai-machine-learning/cheaper-cloud-ai-deployments-with-nvidia-t4-gpu-price-cut>. Acesso em: 10 dez. 2022. Citado 2 vezes nas páginas 23 e 89.

16 PAPIDAS, A. G.; POLYZOS, G. C. Self-organizing networks for 5g and beyond: A view from the top. *Future Internet*, v. 14, n. 3, 2022. ISSN 1999-5903. Disponível em: <https://www.mdpi.com/1999-5903/14/3/95>. Acesso em: 15 ago. 2022. Citado na página 23.

17 DONG, J.; QIAN, Q. A density-based random forest for imbalanced data classification. *Future Internet*, v. 14, n. 3, 2022. ISSN 1999-5903. Disponível em: <https://www.mdpi.com/1999-5903/14/3/90>. Acesso em: 15 ago. 2022. Citado na página 23.

18 Kappeler, A. et al. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, v. 2, n. 2, p. 109–122, June 2016. ISSN 2333-9403. Citado 3 vezes nas páginas 23, 66 e 86.

19 PÉREZ-PELLITERO, E. et al. Photorealistic video super resolution. *CoRR*, abs/1807.07930, 2018. Disponível em: <http://arxiv.org/abs/1807.07930>. Acesso em: 10 dez. 2022. Citado 2 vezes nas páginas 23 e 92.

20 KARRAS, T. et al. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. Disponível em: <http://arxiv.org/abs/1710.10196>. Acesso em: 10 dez. 2022. Citado 2 vezes nas páginas 23 e 25.

21 LEDIG, C. et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. p. 105–114, 2016. ISSN 0018-5043. Disponível em: <http://arxiv.org/abs/1609.04802>. Acesso em: 15 ago. 2022. Citado 18 vezes nas páginas 23, 25, 33, 35, 48, 52, 53, 54, 55, 56, 62, 63, 95, 96, 100, 122, 125 e 131.

22 WANG, X. et al. Esrgan: Enhanced super-resolution generative adversarial networks. In: *The European Conference on Computer Vision Workshops (ECCVW)*. [S.l.: s.n.], 2018. Citado 16 vezes nas páginas 23, 24, 33, 55, 57, 58, 59, 62, 63, 92, 95, 96, 115, 121, 122 e 125.

23 LUCAS, A. et al. Generative adversarial networks and perceptual losses for video super-resolution. *CoRR*, abs/1806.05764, 2018. Disponível em: <http://arxiv.org/abs/1806.05764>. Acesso em: 10 dez. 2022. Citado 2 vezes nas páginas 23 e 95.

24 He, Z. et al. Mrfn: Multi-receptive-field network for fast and accurate single image super-resolution. *IEEE Transactions on Multimedia*, v. 22, n. 4, p. 1042–1054, 2020. Citado na página 23.

25 WANG, J.; TENG, G.; AN, P. Video super-resolution based on generative adversarial network and edge enhancement. *Electronics*, v. 10, n. 4, 2021. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/10/4/459>. Acesso em: 10 dez. 2022. Citado na página 23.

26 Yang, W. et al. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, v. 21, n. 12, p. 3106–3121, 2019. Citado na página 23.

27 NASROLLAHI, K.; MOESLUND, T. B. Super-resolution: a comprehensive survey. *Machine Vision and Applications*, v. 25, n. 6, p. 1423–1468, Aug 2014. ISSN 1432-1769. Disponível em: <https://doi.org/10.1007/s00138-014-0623-4>. Acesso em: 10 dez. 2022. Citado na página 24.

28 WANG, X. et al. EDVR: video restoration with enhanced deformable convolutional networks. *CoRR*, abs/1905.02716, 2019. Citado 6 vezes nas páginas 24, 65, 80, 82, 84 e 86.

29 GOODFELLOW, I. et al. Generative adversarial nets. In: GHAHRAMANI, Z. et al. (Ed.). *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014. p. 2672–2680. Disponível em: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>. Acesso em: 10 dez. 2022. Citado na página 24.

30 SALIMANS, T. et al. Improved techniques for training gans. In: LEE, D. D. et al. (Ed.). *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016. p. 2234–2242. Disponível em: <http://papers.nips.cc/paper/

6125-improved-techniques-for-training-gans.pdf>. Acesso em: 15 ago. 2022. Citado na página 24.

31 TIAN, B. et al. Super-resolution deblurring algorithm for generative adversarial networks. *Proceedings - 2017 2nd International Conference on Mechanical, Control and Computer Engineering, ICMCCE 2017*, v. 2018-January, p. 135–140, 2018. Citado na página 25.

32 DENG, X. Enhancing Image Quality via Style Transfer for Single Image Super-Resolution. *IEEE Signal Processing Letters*, v. 25, n. 4, p. 571–575, 2018. ISSN 10709908. Citado na página 25.

33 BOSCH, M.; GIFFORD, C. M.; RODRIGUEZ, P. A. Super-Resolution for Overhead Imagery Using DenseNets and Adversarial Learning. p. 1414–1422, 2017. Disponível em: <http://arxiv.org/abs/1711.10312>. Acesso em: 15 ago. 2022. Citado na página 25.

34 BAO, J. et al. CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training. *Proceedings of the IEEE International Conference on Computer Vision*, v. 2017-October, p. 2764–2773, 2017. ISSN 15505499. Citado na página 25.

35 AGUSTSSON, E. et al. Extreme Learned Image Compression with GANs. Citado na página 25.

36 LIU, H. et al. Deep Image Compression via End-to-End Learning. p. 2575–2578, 2018. Disponível em: <http://arxiv.org/abs/1806.01496>. Acesso em: 15 ago. 2022. Citado na página 25.

37 I, C.-L. et al. Ran revolution with ngfi (xhaul) for 5g. *Journal of Lightwave Technology*, v. 36, n. 2, p. 541–550, 2018. Citado na página 31.

38 JABER, M. et al. 5g backhaul challenges and emerging research directions: A survey. *IEEE Access*, v. 4, p. 1743–1766, 2016. Citado 2 vezes nas páginas 31 e 112.

39 DARIO, S. *Multi-access Edge Computing (MEC)*. [S.l.]: ETSI, 2022. Disponível em: <https://www.etsi.org/technologies/multi-access-edge-computing/mec>. Acesso em: 15 ago. 2022. Citado na página 32.

40 AHMAD, I. et al. Video transcoding: an overview of various techniques and research issues. *IEEE Transactions on Multimedia*, v. 7, n. 5, p. 793–804, 2005. Citado na página 32.

41 BRUNNSTRÖM, K. et al. Qualinet white paper on definitions of quality of experience. 2013. Citado na página 32.

42 BARAKABITZE, A. A.; WALSHE, R. Sdn and nfv for qoe-driven multimedia services delivery: The road towards 6g and beyond networks. *Computer Networks*, v. 214, p. 109133, 2022. ISSN 1389-1286. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128622002523>. Acesso em: 15 ago. 2022. Citado na página 32.

43 YIN, X. et al. A control-theoretic approach for dynamic adaptive video streaming over http. In: *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. New York, NY, USA: Association for Computing Machinery, 2015. (SIGCOMM '15), p. 325–338. ISBN 9781450335423. Disponível em:

<https://doi.org/10.1145/2785956.2787486>. Acesso em: 15 ago. 2022. Citado 2 vezes nas páginas 32 e 125.

44 TIAN, J.; MA, K.-K. A survey on super-resolution imaging. *Signal, Image and Video Processing*, v. 5, n. 3, p. 329–342, Sep 2011. ISSN 1863-1711. Disponível em: <https://doi.org/10.1007/s11760-010-0204-6>. Acesso em: 10 dez. 2022. Citado na página 33.

45 ZHANG, K.; GU, S.; TIMOFTE, R. Ntire 2020 challenge on perceptual extreme super-resolution: Methods and results. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2020. p. 2045–2057. Citado 7 vezes nas páginas 33, 35, 55, 95, 115, 122 e 131.

46 BLAU, Y. et al. 2018 PIRM challenge on perceptual image super-resolution. *CoRR*, abs/1809.07517, 2018. Disponível em: <http://arxiv.org/abs/1809.07517>. Acesso em: 10 dez. 2022. Citado 8 vezes nas páginas 33, 36, 55, 95, 115, 122, 129 e 131.

47 ANWAR, S.; KHAN, S.; BARNES, N. A deep journey into super-resolution: A survey. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 53, n. 3, maio 2020. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3390462>. Acesso em: 15 ago. 2022. Citado 2 vezes nas páginas 33 e 112.

48 YANG, W. et al. Deep learning for single image super-resolution: A brief review. *Trans. Multi.*, IEEE Press, v. 21, n. 12, p. 3106–3121, dec 2019. ISSN 1520-9210. Disponível em: <https://doi.org/10.1109/TMM.2019.2919431>. Acesso em: 15 ago. 2022. Citado na página 33.

49 LEE, R.; VENIERIS, S. I.; LANE, N. D. Deep neural network–based enhancement for image and video streaming systems: A survey and future directions. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 54, n. 8, oct 2021. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3469094>. Acesso em: 15 ago. 2022. Citado na página 33.

50 FAN, Q. et al. Metrics and methods of video quality assessment: a brief review. *Multimedia Tools and Applications*, v. 78, n. 22, p. 31019–31033, Nov 2019. ISSN 1573-7721. Disponível em: <https://doi.org/10.1007/s11042-017-4848-x>. Acesso em: 17 jul. 2022. Citado na página 34.

51 SESHADRINATHAN, K. et al. Study of subjective and objective quality assessment of video. *IEEE Transactions on Image Processing*, v. 19, n. 6, p. 1427–1441, 2010. Citado na página 34.

52 ZHANG, R. et al. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924, 2018. Disponível em: <http://arxiv.org/abs/1801.03924>. Acesso em: 17 jul. 2022. Citado 3 vezes nas páginas 35, 36 e 40.

53 SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. Disponível em: <http://arxiv.org/abs/1409.1556>. Acesso em: 13 jul. 2018. Citado 5 vezes nas páginas 36, 56, 60, 96 e 122.

54 IANDOLA, F. N. et al. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016. Disponível em: <http://arxiv.org/abs/1602.07360>. Acesso em: 17 jul. 2022. Citado na página 36.

55 KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. USA: Curran Associates Inc., 2012. (NIPS'12), p. 1097–1105. Disponível em: <http://dl.acm.org/citation.cfm?id=2999134.2999257>. Acesso em: 15 ago. 2022. Citado 2 vezes nas páginas 36 e 43.

56 LI, Z. et al. *VMAF: The Journey Continues*. https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12, 2018. Citado na página 36.

57 AARON et al. Challenges in cloud based ingest and encoding for high quality streaming media. In: *2015 IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2015. p. 1732–1736. Citado na página 36.

58 MA, C. et al. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, v. 158, p. 1–16, 2017. ISSN 1077-3142. Disponível em: <https://www.sciencedirect.com/science/article/pii/S107731421630203X>. Acesso em: 15 ago. 2022. Citado na página 36.

59 MITTAL, A.; SOUNDARARAJAN, R.; BOVIK, A. C. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, v. 20, n. 3, p. 209–212, 2013. Citado na página 36.

60 MAO, H.; NETRAVALI, R.; ALIZADEH, M. Neural adaptive video streaming with pensieve. In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. New York, NY, USA: ACM, 2017. (SIGCOMM '17), p. 197–210. ISBN 978-1-4503-4653-5. Disponível em: <http://doi.acm.org/10.1145/3098822.3098843>. Acesso em: 15 ago. 2022. Citado 2 vezes nas páginas 37 e 125.

61 HU, P.; MISRA, R.; KATTI, S. Dejavu: Enhancing videoconferencing with prior knowledge. In: . [S.l.]: HotMobile '19: Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications, 2019. p. 63–68. Citado 3 vezes nas páginas 37, 38 e 39.

62 KIM, J. et al. Neural-enhanced live streaming: Improving live video ingest via online learning. In: *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*. New York, NY, USA: Association for Computing Machinery, 2020. (SIGCOMM '20), p. 107–125. ISBN 9781450379557. Disponível em: <https://doi.org/10.1145/3387514.3405856>. Acesso em: 15 ago. 2022. Citado 4 vezes nas páginas 37, 38, 39 e 114.

63 CHEN, Y. et al. Higher quality live streaming under lower uplink bandwidth: an approach of super-resolution based video coding. In: *NOSSDAV '21: Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. [S.l.: s.n.], 2021. Citado 2 vezes nas páginas 38 e 39.

64 KHANI, M.; SIVARAMAN, V.; ALIZADEH, M. Efficient video compression via content-adaptive super-resolution. In: *Proceedings of the IEEE/CVF International*

*Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2021. p. 4521–4530. Citado 2 vezes nas páginas 38 e 39.

65 WANG, F. et al. Intelligent edge-assisted crowdcast with deep reinforcement learning for personalized qoe. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. [S.l.: s.n.], 2019. p. 910–918. Citado 2 vezes nas páginas 39 e 112.

66 DOGGA, P. et al. Edge-based transcoding for adaptive live video streaming. In: *HotEdge*. [S.l.: s.n.], 2019. Citado na página 39.

67 WANG, Y. et al. Bridging the Edge-Cloud barrier for real-time advanced vision analytics. In: *11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19)*. Renton, WA: USENIX Association, 2019. Disponível em: <https://www.usenix.org/conference/hotcloud19/presentation/\wang>. Acesso em: 15 ago. 2022. Citado na página 39.

68 LUBIN, J. A human vision system model for objective image fidelity and target detectability measurements. In: *9th European Signal Processing Conference (EUSIPCO 1998)*. [S.l.: s.n.], 1998. p. 1–4. Citado na página 40.

69 WATSON, A. B. Proposal : Measurement of a jnd scale for video quality. In: . [S.l.: s.n.], 2000. Citado na página 40.

70 LIN, J. Y.-c. et al. Experimental design and analysis of jnd test on coded image/video. In: . [S.l.: s.n.], 2015. p. 95990Z. Citado na página 40.

71 WANG, H. et al. Videoset: A large-scale compressed video quality dataset based on JND measurement. *CoRR*, abs/1701.01500, 2017. Disponível em: <http://arxiv.org/abs/1701.01500>. Acesso em: 17 jul. 2022. Citado 6 vezes nas páginas 40, 41, 92, 97, 100 e 103.

72 GREENSPAN, H. Super-Resolution in Medical Imaging. *The Computer Journal*, v. 52, n. 1, p. 43–63, 02 2008. ISSN 0010-4620. Disponível em: <https://doi.org/10.1093/comjnl/bxm075>. Acesso em: 15 ago. 2022. Citado na página 42.

73 Gunturk, B. K. et al. Eigenface-domain super-resolution for face recognition. *IEEE Transactions on Image Processing*, v. 12, n. 5, p. 597–606, May 2003. ISSN 1057-7149. Citado na página 42.

74 Demirel, H.; Anbarjafari, G. Discrete wavelet transform-based satellite image resolution enhancement. *IEEE Transactions on Geoscience and Remote Sensing*, v. 49, n. 6, p. 1997–2004, June 2011. ISSN 0196-2892. Citado na página 42.

75 QIFANG, X.; GUOQING, Y.; PIN, L. Super-resolution reconstruction of satellite video images based on interpolation method. *Procedia Computer Science*, v. 107, p. 454 – 459, 2017. ISSN 1877-0509. Advances in Information and Communication Technology: Proceedings of 7th International Congress of Information and Communication Technology (ICICT2017). Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050917303642>. Acesso em: 17 jul. 2022. Citado na página 42.

76 ZHANG, L. et al. A super-resolution reconstruction algorithm for surveillance images. *Signal Processing*, v. 90, n. 3, p. 848 – 859, 2010. ISSN 0165-1684. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0165168409003776>. Acesso em: 17 jul. 2022. Citado na página 42.

77 HE, K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. Disponível em: <http://arxiv.org/abs/1502.01852>. Acesso em: 17 jul. 2022. Citado 3 vezes nas páginas 43, 48 e 93.

78 DONG, C. et al. Learning a deep convolutional network for image super-resolution. In: FLEET, D. et al. (Ed.). *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014. p. 184–199. ISBN 978-3-319-10593-2. Citado 4 vezes nas páginas 43, 44, 45 e 66.

79 Dong, C. et al. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 38, n. 2, p. 295–307, Feb 2016. ISSN 0162-8828. Citado 4 vezes nas páginas 43, 62, 63 e 100.

80 NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. USA: Omnipress, 2010. (ICML'10), p. 807–814. ISBN 978-1-60558-907-7. Disponível em: <http://dl.acm.org/citation.cfm?id=3104322.3104425>. Acesso em: 15 ago. 2022. Citado na página 44.

81 Lecun, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 0018-9219. Citado na página 45.

82 Yang, J. et al. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, v. 19, n. 11, p. 2861–2873, Nov 2010. ISSN 1057-7149. Citado na página 45.

83 ZEYDE, R.; ELAD, M.; PROTTER, M. On single image scale-up using sparse-representations. In: *Proceedings of the 7th International Conference on Curves and Surfaces*. Berlin, Heidelberg: Springer-Verlag, 2012. p. 711–730. ISBN 978-3-642-27412-1. Disponível em: <https://doi.org/10.1007/978-3-642-27413-8_47>. Acesso em: 15 ago. 2022. Citado 2 vezes nas páginas 45 e 61.

84 BEVILACQUA, M. et al. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: *Proceedings of the British Machine Vision Conference*. [S.l.]: BMVA Press, 2012. p. 135.1–135.10. ISBN 1-901725-46-4. Citado 2 vezes nas páginas 45 e 61.

85 Hong Chang; Dit-Yan Yeung; Yimin Xiong. Super-resolution through neighbor embedding. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* [S.l.: s.n.], 2004. v. 1, p. I–I. ISSN 1063-6919. Citado na página 45.

86 Timofte, R.; De, V.; Gool, L. V. Anchored neighborhood regression for fast example-based super-resolution. In: *2013 IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 1920–1927. ISSN 1550-5499. Citado na página 45.

87 SHI, W. et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *CoRR*, abs/1609.05158, 2016. Disponível em: <http://arxiv.org/abs/1609.05158>. Acesso em: 17 jul. 2022. Citado 9 vezes nas páginas 45, 46, 47, 62, 63, 93, 100, 119 e 121.

88 Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, v. 5, n. 2, p. 157–166, March 1994. ISSN 1045-9227. Citado na página 47.

89 GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: TEH, Y. W.; TITTERINGTON, M. (Ed.). *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010. (Proceedings of Machine Learning Research, v. 9), p. 249–256. Disponível em: <http://proceedings.mlr.press/v9/glorot10a.html>. Acesso em: 17 jul. 2022. Citado na página 47.

90 IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. Disponível em: <http://arxiv.org/abs/1502.03167>. Acesso em: 17 jul. 2022. Citado 2 vezes nas páginas 47 e 48.

91 He, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778. ISSN 1063-6919. Citado 2 vezes nas páginas 47 e 48.

92 HE, K.; SUN, J. Convolutional neural networks at constrained time cost. *CoRR*, abs/1412.1710, 2014. Disponível em: <http://arxiv.org/abs/1412.1710>. Citado na página 47.

93 SRIVASTAVA, R. K.; GREFF, K.; SCHMIDHUBER, J. Highway networks. *CoRR*, abs/1505.00387, 2015. Disponível em: <http://arxiv.org/abs/1505.00387>. Citado na página 47.

94 KIM, J.; LEE, J. K.; LEE, K. M. Accurate image super-resolution using very deep convolutional networks. *CoRR*, abs/1511.04587, 2015. Disponível em: <http://arxiv.org/abs/1511.04587>. Citado na página 48.

95 KIM, J.; LEE, J. K.; LEE, K. M. Deeply-recursive convolutional network for image super-resolution. *CoRR*, abs/1511.04491, 2015. Disponível em: <http://arxiv.org/abs/1511.04491>. Citado na página 48.

96 Nah, S.; Kim, T. H.; Lee, K. M. Deep multi-scale convolutional neural network for dynamic scene deblurring. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 257–265. ISSN 1063-6919. Citado na página 48.

97 LIM, B. et al. Enhanced deep residual networks for single image super-resolution. *CoRR*, abs/1707.02921, 2017. Disponível em: <http://arxiv.org/abs/1707.02921>. Citado 6 vezes nas páginas 48, 49, 57, 58, 62 e 63.

98 SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. Disponível em: <http://arxiv.org/abs/1602.07261>. Citado 4 vezes nas páginas 48, 57, 58 e 93.

99 HUANG, G.; LIU, Z.; WEINBERGER, K. Q. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. Disponível em: <http://arxiv.org/abs/1608.06993>. Citado na página 48.

100 Tong, T. et al. Image super-resolution using dense skip connections. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2017. p. 4809–4817. ISSN 2380-7504. Citado na página 48.

101 ZHANG, Y. et al. Residual dense network for image super-resolution. *CoRR*, abs/1802.08797, 2018. Disponível em: <http://arxiv.org/abs/1802.08797>. Citado 6 vezes nas páginas 48, 49, 50, 58, 62 e 63.

102 KINGMA, D.; BA, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. Citado 3 vezes nas páginas 52, 57 e 78.

103 MATHIEU, M.; COUPRIE, C.; LECUN, Y. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015. Disponível em: <http://arxiv.org/abs/1511.05440>. Acesso em: 17 jul. 2022. Citado 3 vezes nas páginas 52, 55 e 95.

104 JOHNSON, J.; ALAHI, A.; LI, F. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. Disponível em: <http://arxiv.org/abs/1603.08155>. Acesso em: 17 jul. 2022. Citado 5 vezes nas páginas 52, 55, 56, 95 e 119.

105 DOSOVITSKIY, A.; BROX, T. Generating images with perceptual similarity metrics based on deep networks. *CoRR*, abs/1602.02644, 2016. Disponível em: <http://arxiv.org/abs/1602.02644>. Acesso em: 17 jul. 2022. Citado na página 52.

106 BRUNA, J.; SPRECHMANN, P.; LECUN, Y. Super-resolution with deep convolutional sufficient statistics. *CoRR*, abs/1511.05666, 2015. Disponível em: <http://arxiv.org/abs/1511.05666>. Citado 4 vezes nas páginas 52, 55, 56 e 95.

107 Goodfellow, I. J. et al. Generative Adversarial Networks. *arXiv e-prints*, p. arXiv:1406.2661, jun. 2014. Citado 2 vezes nas páginas 53 e 93.

108 MAAS, A. L. Rectifier nonlinearities improve neural network acoustic models. In: . [S.l.: s.n.], 2013. Citado 2 vezes nas páginas 55 e 93.

109 Zhao, H. et al. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, v. 3, n. 1, p. 47–57, March 2017. ISSN 2333-9403. Citado 2 vezes nas páginas 55 e 95.

110 GATYS, L. A.; ECKER, A. S.; BETHGE, M. Texture synthesis using convolutional neural networks. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. Cambridge, MA, USA: MIT Press, 2015. (NIPS'15), p. 262–270. Disponível em: <http://dl.acm.org/citation.cfm?id=2969239.2969269>. Acesso em: 15 ago. 2022. Citado na página 56.

111 RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, v. 115, n. 3, p. 211–252, 2015. Citado 2 vezes nas páginas 56 e 62.

112 JOLICOEUR-MARTINEAU, A. The relativistic discriminator: a key element missing from standard GAN. *CoRR*, abs/1807.00734, 2018. Disponível em: <http://arxiv.org/abs/1807.00734>. Citado 5 vezes nas páginas 57, 59, 94, 119 e 123.

113 ZHANG, Y. et al. Image super-resolution using very deep residual channel attention networks. *CoRR*, abs/1807.02758, 2018. Disponível em: <http://arxiv.org/abs/1807.02758>. Citado na página 58.

114 ZHANG, K. et al. Residual networks of residual networks: Multilevel residual networks. *CoRR*, abs/1608.02908, 2016. Disponível em: <http://arxiv.org/abs/1608.02908>. Citado na página 58.

115 MARTIN, D. et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proc. 8th Int'l Conf. Computer Vision*. [S.l.: s.n.], 2001. v. 2, p. 416–423. Citado na página 61.

116 ARBELAEZ, P. et al. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 33, n. 5, p. 898–916, maio 2011. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2010.161>. Acesso em: 15 ago. 2022. Citado na página 61.

117 MATSUI, Y. et al. Sketch-based manga retrieval using manga109 dataset. *CoRR*, abs/1510.04389, 2015. Disponível em: <http://arxiv.org/abs/1510.04389>. Citado na página 62.

118 AGUSTSSON, E.; TIMOFTE, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. [S.l.: s.n.], 2017. Citado na página 62.

119 MILLER, G. A. Wordnet: A lexical database for english. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 38, n. 11, p. 39–41, nov. 1995. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/219717.219748>. Acesso em: 15 ago. 2022. Citado na página 62.

120 CABALLERO, J. et al. Real-time video super-resolution with spatio-temporal networks and motion compensation. In: . [S.l.: s.n.], 2017. p. 2848–2857. Citado 7 vezes nas páginas 65, 68, 69, 70, 73, 74 e 86.

121 HUBER, P. J. Robust estimation of a location parameter. *Ann. Math. Statist.*, The Institute of Mathematical Statistics, v. 35, n. 1, p. 73–101, 03 1964. Disponível em: <https://doi.org/10.1214/aoms/1177703732>. Citado na página 70.

122 Tao, X. et al. Detail-revealing deep video super-resolution. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2017. p. 4482–4490. Citado 2 vezes nas páginas 73 e 86.

123 MAO, X.; SHEN, C.; YANG, Y. Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections. *CoRR*, abs/1603.09056, 2016. Disponível em: <http://arxiv.org/abs/1603.09056>. Citado na página 75.

124 LIU, Z. et al. Video frame synthesis using deep voxel flow. *CoRR*, abs/1702.02463, 2017. Disponível em: <http://arxiv.org/abs/1702.02463>. Citado na página 75.

125 SU, S. et al. Deep video deblurring. *CoRR*, abs/1611.08387, 2016. Disponível em: <http://arxiv.org/abs/1611.08387>. Citado na página 75.

126 JO, Y. et al. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 3224–3232. Citado 4 vezes nas páginas 76, 77, 79 e 86.

127 TRAN, D. et al. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014. Disponível em: <http://arxiv.org/abs/1412.0767>. Citado na página 78.

128 BRABANDERE, B. D. et al. Dynamic filter networks. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. USA: Curran Associates Inc., 2016. (NIPS'16), p. 667–675. ISBN 978-1-5108-3881-9. Disponível em: <http://dl.acm.org/citation.cfm?id=3157096.3157171>. Acesso em: 15 ago. 2022. Citado na página 78.

129 TIAN, Y. et al. TDAN: temporally deformable alignment network for video super-resolution. *CoRR*, abs/1812.02898, 2018. Citado na página 81.

130 HUI, T.; TANG, X.; LOY, C. C. Liteflownet: A lightweight convolutional neural network for optical flow estimation. *CoRR*, abs/1805.07036, 2018. Disponível em: <http://arxiv.org/abs/1805.07036>. Citado 2 vezes nas páginas 81 e 82.

131 ILG, E. et al. Flownet 2.0: Evolution of optical flow estimation with deep networks. *CoRR*, abs/1612.01925, 2016. Disponível em: <http://arxiv.org/abs/1612.01925>. Citado 2 vezes nas páginas 81 e 82.

132 ZHU, X. et al. Deformable convnets v2: More deformable, better results. *CoRR*, abs/1811.11168, 2018. Disponível em: <http://arxiv.org/abs/1811.11168>. Citado na página 81.

133 DAI, J. et al. Deformable convolutional networks. *CoRR*, abs/1703.06211, 2017. Disponível em: <http://arxiv.org/abs/1703.06211>. Citado na página 82.

134 RANJAN, A.; BLACK, M. J. Optical flow estimation using a spatial pyramid network. *CoRR*, abs/1611.00850, 2016. Disponível em: <http://arxiv.org/abs/1611.00850>. Citado na página 82.

135 SUN, D. et al. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *CoRR*, abs/1709.02371, 2017. Disponível em: <http://arxiv.org/abs/1709.02371>. Citado na página 82.

136 HUI, T.; TANG, X.; LOY, C. C. A lightweight optical flow CNN - revisiting data fidelity and regularization. *CoRR*, abs/1903.07414, 2019. Disponível em: <http://arxiv.org/abs/1903.07414>. Citado na página 82.

137 Liu, C.; Sun, D. On bayesian adaptive video super resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 36, n. 2, p. 346–360, Feb 2014. ISSN 1939-3539. Citado na página 85.

138 XUE, T. et al. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, Springer, v. 127, n. 8, p. 1106–1125, 2019. Citado na página 85.

139 NAH, S. et al. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: *CVPR Workshops*. [S.l.: s.n.], 2019. Citado 2 vezes nas páginas 85 e 86.

140 NAH, S. et al. Ntire 2019 challenge on video super-resolution: Methods and results. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 1985–1995, 2019. Citado na página 86.

141 Adhikari, V. K. et al. Measurement study of netflix, hulu, and a tale of three cdns. *IEEE/ACM Transactions on Networking*, v. 23, n. 6, p. 1984–1997, Dec 2015. ISSN 1558-2566. Citado 2 vezes nas páginas 88 e 89.

142 NAIR, M. *How Netflix works: the (hugely simplified) complex stuff that happens every time you hit Play*. 2017. Medium. Citado 2 vezes nas páginas 88 e 89.

143 TELEGEOGRAPHY, T. *Telegeography's Global Bandwidth Research Service*. 2022. Disponível em: <https://www2.telegeography.com/global-bandwidth-research-service-samples-download>. Citado 2 vezes nas páginas 89 e 90.

144 ESHELMAN, E. *NVIDIA Tesla V100 Price Analysis*. 2018. Microway. Disponível em: <https://www.microway.com/hpc-tech-tips/nvidia-tesla-v100-price-analysis/>. Citado na página 89.

145 HUANG, C. et al. Measuring and evaluating large-scale cdns. In: *In IMC*. [S.l.: s.n.], 2008. Citado na página 90.

146 ZHOU, B. et al. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, 2017. Citado 3 vezes nas páginas 96, 98 e 101.

147 CABLELABS. *4K video*. CableLabs. Disponível em: <https://www.cablelabs.com/4k>. Citado na página 98.

148 PINSON, M. *The consumer digital video library*. 2013. Disponível em: <http://www.cdvl.org/resources/index.php>. Citado na página 98.

149 CHEN, H. et al. Cisrdcnn: Super-resolution of compressed images using deep convolutional neural networks. *Neurocomputing*, v. 285, p. 204 – 219, 2018. ISSN 0925-2312. Citado na página 101.

150 TELEGEOGRAPHY. *Executive summary: Telegeography Global Internet Research Service*. 2022. Disponível em: <https://www2.telegeography.com/hubfs/assets/product-tear-sheets/product-page-content-samples/global-internet-geography/telegeography-global-internet-geography-executive-summary.pdf>. Citado 2 vezes nas páginas 111 e 112.

151 CISCO. *Cisco Annual Internet Report (2018–2023) White Paper*. [S.l.], 2020. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/executive\-perspectives/annual-internet-report/white-paper-c11-741490.html>,. Acesso em: 24 ago 2022. Citado 4 vezes nas páginas 111, 112, 113 e 141.

152 SHARMA, T.; CHEHRI, A.; FORTIER, P. Review of optical and wireless backhaul networks and emerging trends of next generation 5g and 6g technologies. *Trans. Emerg. Telecommun. Technol.*, John Wiley  Sons, Inc., USA, v. 32, n. 3, mar. 2021. ISSN 2161-3915. Disponível em: <https://doi.org/10.1002/ett.4155>. Acesso em: 15 ago. 2022. Citado na página 112.

153 AHAMED, M. M.; FARUQUE, S. 5g backhaul: Requirements, challenges, and emerging technologies. In: _____. [S.l.: s.n.], 2018. p. 43–58. ISBN 978-1-78923-743-6. Citado na página 112.

154 TEZERGIL, B.; ONUR, E. Wireless backhaul in 5g and beyond: Issues, challenges and opportunities. *IEEE Communications Surveys  Tutorials*, p. 1–1, 2022. Citado na página 112.

155 GE, C. et al. Qoe-assured 4k http live streaming via transient segment holding at mobile edge. *IEEE Journal on Selected Areas in Communications*, v. 36, n. 8, p. 1816–1830, 2018. Citado na página 112.

156 SACCO, A.; ESPOSITO, F.; MARCHETTO, G. Resource inference for task migration in challenged edge networks with ritmo. In: *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*. [S.l.: s.n.], 2020. p. 1–7. Citado na página 112.

157 YEO, H.; DO, S.; HAN, D. How will deep learning change internet video delivery? In: *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*. New York, NY, USA: Association for Computing Machinery, 2017. (HotNets-XVI), p. 57–64. ISBN 9781450355698. Disponível em: <https://doi.org/10.1145/3152434.3152440>. Acesso em: 15 ago. 2022. Citado na página 112.

158 Liborio, J. M.; Melo, C. A. V. A gan to fight video-related traffic flooding: Super-resolution. In: *2019 IEEE Latin-American Conference on Communications (LATINCOM)*. [S.l.: s.n.], 2019. p. 1–6. Citado na página 112.

159 HINTON, G.; VINYALS, O.; DEAN, J. *Distilling the Knowledge in a Neural Network*. 2015. Citado 2 vezes nas páginas 119 e 123.

160 HU, X. et al. Rtsrgan: Real-time super-resolution generative adversarial networks. In: *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*. [S.l.: s.n.], 2019. p. 321–326. Citado na página 121.

161 IGNATOV, A. et al. *Real-Time Video Super-Resolution on Smartphones with Deep Learning, Mobile AI 2021 Challenge: Report*. 2021. Citado na página 121.

162 HUI, Z. et al. Lightweight image super-resolution with information multi-distillation network. *Proceedings of the 27th ACM International Conference on Multimedia*, ACM, Oct 2019. Disponível em: <http://dx.doi.org/10.1145/3343031.3351084>. Citado 2 vezes nas páginas 121 e 125.

163 LIU, S. et al. Evsrnet: Efficient video super-resolution with neural architecture search. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2021. p. 2480–2485. Citado na página 121.

164 SHANG, T. et al. Perceptual extreme super-resolution network with receptive field block. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2020. p. 440–441. Citado na página 122.

165 LI, Y. et al. Ntire 2022 challenge on efficient super-resolution: Methods and results. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2022. Citado na página 125.

166 HARMONIC. *4K and Ultra High Definition Video Services Explained*. 2021. Disponível em: <https://www.harmonicinc.com/insights/blog/4k-in-context>. Acesso em: 15 ago. 2022. Citado 2 vezes nas páginas 125 e 134.

167 HEDGEHOG. 2021. Disponível em: <https://www.youtube.com/watch?v=jEBDNLNEUgU>. Acesso em: 15 ago. 2022. Citado na página 125.

168 HEDGEHOG. 2021. Disponível em: <https://www.youtube.com/watch?v=6EsTH7JQT_o>. Acesso em: 15 ago. 2022. Citado na página 125.

169 GIBRAN anto. 2020. Disponível em: <https://www.youtube.com/watch?v=peuIQ9Iztwo>. Acesso em: 15 ago. 2022. Citado na página 125.

170 TELLYDAN. 2020. Disponível em: <https://www.youtube.com/watch?v=xp763iNB_MA>. Acesso em: 15 ago. 2022. Citado na página 125.

171 JAVIERNATHANIEL. 2015. Disponível em: <https://www.youtube.com/watch?v=G4t6TqG5LM8>. Acesso em: 15 ago. 2022. Citado na página 125.

172 FLOW, R. no. 2020. Disponível em: <https://www.youtube.com/watch?v=enoAM1XNKBs>. Acesso em: 15 ago. 2022. Citado na página 125.

173 PODCAST, C. de. 2021. Disponível em: <https://www.youtube.com/watch?v=qlkNx3BrfbI>. Acesso em: 15 ago. 2022. Citado na página 125.

174 ROBITZA, W. *CRF Guide (Constant Rate Factor in x264, x265 and libvpx)*. 2017. Disponível em: <https://slhck.info/video/2017/02/24/crf-guide.html>. Acesso em: 15 ago. 2022. Citado na página 125.

175 FFMPEG. *A complete, cross-platform solution to record, convert and stream audio and video*. 2022. Disponível em: <https://www.ffmpeg.org>. Acesso em: 15 ago. 2022. Citado na página 127.

176 ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Disponível em: <https://www.tensorflow.org/>. Acesso em: 15 ago. 2022. Citado na página 127.

177 NGINX. *Repository NGINX Open Source*. 2022. Disponível em: <https://github.com/nginx/nginx>. Acesso em: 15 ago. 2022. Citado na página 127.

178 FORUM, D. I. *dash.js*. 2021. Disponível em: <https://github.com/Dash-Industry-Forum/dash.js>. Acesso em: 15 ago. 2022. Citado na página 127.

179 NETRAVALI, R. et al. Mahimahi: Accurate record-and-replay for http. In: *Proceedings of the 2015 USENIX Conference on Usenix Annual Technical Conference*. USA: USENIX Association, 2015. (USENIX ATC '15), p. 417–429. ISBN 9781931971225. Citado na página 127.

180 RACA, D. et al. Beyond throughput, the next generation: A 5g dataset with channel and context metrics. In: *Proceedings of the 11th ACM Multimedia Systems Conference*. New York, NY, USA: Association for Computing Machinery, 2020. (MMSys '20), p. 303–308. ISBN 9781450368452. Disponível em: <https://doi.org/10.1145/3339825.3394938>. Acesso em: 15 ago. 2022. Citado na página 132.

181 SPITERI, K.; URGAONKAR, R.; SITARAMAN, R. K. Bola: Near-optimal bitrate adaptation for online videos. *IEEE/ACM Transactions on Networking*, v. 28, n. 4, p. 1698–1711, 2020. Citado na página 134.

182 LIM, M. et al. When they go high, we go low: Low-latency live streaming in dash.js with lol. In: *Proceedings of the 11th ACM Multimedia Systems Conference*. New York, NY, USA: Association for Computing Machinery, 2020. (MMSys '20), p. 321–326. ISBN 9781450368452. Disponível em: <https://doi.org/10.1145/3339825.3397043>. Acesso em: 15 ago. 2022. Citado na página 134.

183 KARAGKIOULES, T. et al. Online learning for low-latency adaptive streaming. In: *Proceedings of the 11th ACM Multimedia Systems Conference*. New York, NY, USA: Association for Computing Machinery, 2020. (MMSys '20), p. 315–320. ISBN 9781450368452. Disponível em: <https://doi.org/10.1145/3339825.3397042>. Acesso em: 15 ago. 2022. Citado na página 134.