TÉCNICAS DE PRÉ-CODIFICAÇÃO BASEADAS EM APRENDIZAGEM PROFUNDA PARA COMPRESSÃO DE VÍDEO DE PRÓXIMA GERAÇÃO

Aaron Chadha, Eirina Bourtsoulatze, Ilya Fadeev, Vasileios Giotsas,
Sergio Grce e Yiannis Andreopoulos
iSIZE Ltd, Londres, Reino Unido,www.isize.co ,info@isize.co

ABSTRATO

Vários grupos de pesquisa em todo o mundo estão atualmente investigando como o aprendizado profundo pode avançar no estado da arte em codificação de imagens e vídeos. Uma questão em aberto é como fazer redes neurais profundas funcionarem em conjunto com codecs de vídeo existentes (e futuros), como MPEG AVC/H.264, HEVC, VVC, Google VP9 e AOMedia AV1, bem como formatos de contêiner e transporte existentes. Tal compatibilidade é um aspecto crucial, uma vez que se espera que a indústria de conteúdos de vídeo e os fabricantes de hardware continuem empenhados em apoiar estas normas num futuro próximo.

Propomos redes neurais profundas como componentes de pré-codificação para ecossistemas de codecs atuais e futuros. Em nossas implantações atuais para streaming adaptativo DASH/HLS, isso inclui a redução da escala de redes neurais. A pré-codificação por meio de aprendizagem profunda permite total compatibilidade com codecs e padrões de transporte atuais e futuros, proporcionando economias significativas. Nossos resultados com conteúdo HD mostram que ocorre uma redução de taxa de 23% a 43% em uma variedade de implementações de codec de vídeo de última geração. O uso da pré-codificação também pode levar a uma redução significativa da complexidade da codificação, o que é essencial para a implantação na nuvem de codificadores complexos como AV1 e MPEG VVC. Portanto, além da economia de taxa de bits, a pré-codificação baseada em aprendizagem profunda pode reduzir os recursos de nuvem necessários para transcodificação de vídeo e tornar as soluções baseadas em nuvem competitivas ou superiores às implantações cativas de última geração.

INTRODUÇÃO

Em apenas alguns anos, a tecnologia reformulou completamente a forma como consumimos televisão, longas-metragens e outros conteúdos de primeira linha. Por exemplo, o Ofcom informou em julho de 2018 que há agora mais assinaturas no Reino Unido para Netflix, Amazon e NOW TV do que para serviços "tradicionais" de TV paga.[1]. A proliferação de conteúdo de streaming "over-the-top" (OTT) foi acompanhada por um apetite por conteúdo de alta resolução. Por exemplo, 50% dos lares dos EUA terão TVs UHD/4K até 2020. Ao mesmo tempo, os custos do equipamento de câmara 4K têm vindo a cair rapidamente. Olhando para o futuro, as TVs 8K foram apresentadas na CES 2018 por vários grandes fabricantes e várias emissoras anunciaram que iniciarão transmissões de 8K a tempo para os Jogos Olímpicos de 2020 no Japão. Infelizmente, para a maioria dos países, mesmo a entrega de conteúdo HD/UHD ainda é afetada por problemas de infraestrutura de banda larga.

[1] https://www.ofcom.org.uk/about-ofcom/latest/media/media-releases/2018/streaming-overtakes-pay-tv

Streaming adaptável HTTP

Para contornar esse problema, os provedores de conteúdo OTT agora transmitem escalas de resolução e taxa de bits, padronizadas como DASH e HLS (transmissão dinâmica adaptativa sobre HTTP e transmissão ao vivo HTTP), que permitem que o dispositivo cliente mude para uma faixa de resoluções e taxas de bits mais baixas quando a velocidade da conexão não é suficiente para o fluxo de bits de vídeo de alta qualidade/resolução total. No entanto, o impacto na qualidade da redução bicúbica amplamente utilizada pode ser bastante grave. Em princípio, isso poderia ser remediado por soluções de upscaling de vídeo aprendíveis pós-decodificação [1], que já demonstraram melhorar a qualidade da imagem em comparação com filtros de upscaling linear. No entanto, a sua implementação requer alterações muito substanciais no dispositivo cliente de descodificação, que são geralmente demasiado incómodas e complexas para serem realizadas na prática. Por exemplo, upscalers baseados em redes neurais convolucionais (CNN) com dezenas de milhões de parâmetros não podem ser suportados pelos principais navegadores da Web baseados em CPU que suportam reprodução de vídeo DASH e HLS.

Apresentando o conceito de pré-codificação para comunicações de vídeo

A pré-codificação foi inicialmente proposta para comunicações sem fio MIMO como um meio de pré-processar os símbolos transmitidos e realizar a diversidade de transmissão [2]. A pré-codificação é semelhante à equalização de canal, mas a principal diferença é que é necessário otimizar o pré-codificador com a operação do decodificador utilizado. Enquanto a equalização de canal visa minimizar os erros do canal, um pré-codificador visa minimizar o erro na saída do receptor.

Neste artigo, apresentamos o conceito de pré-codificação para entrega de vídeo. Nosso foco atual está no streaming de vídeo adaptável HTTP, que está no centro da entrega de vídeo OTT. A pré-codificação para transporte de vídeo por HTTP é feita pré-processando o sinal de vídeo antes da codificação padrão, permitindo ao mesmo tempo que um codificador de vídeo padrão e um player compatível com DASH/HLS o decodifique sem quaisquer modificações. Conforme ilustrado na Figura 1, o princípio fundamental é otimizar a operação do pré-codificador para minimizar a distorção na saída de um reprodutor habilitado para DASH/HLS. Isto é conseguido ajustando de forma ideal a resolução utilizada de acordo com: a taxa de bits, o codec de vídeo utilizado e sua especificação, e o filtro de upscaling usado pelo player compatível com DASH/HLS. Em relação a este último, os navegadores tendem a usar o filtro bilinear[2]para garantir uma reprodução suave em thin clients e dispositivos móveis.

Ao utilizar o aprendizado profundo como uma ferramenta para uma pré-codificação eficiente, uma questão central do nosso trabalho é:*Podemos reduzir a distorção da reprodução de vídeo DASH/HLS padrão por meio de pré-codificação com redes neurais convolucionais (CNNs)?*Seguindo o esquema da Figura 1, mostramos que isso é realmente possível por grupo de imagens (GOP) sem qualquer modificação no cliente (decodificador e reprodutor de vídeo). Nossos experimentos com conteúdo de teste padrão do repositório XIPH e as implementações x264/x265/libvpx-vp9 dos codificadores H.264/AVC, HEVC e VP9 mostram que, em comparação ao uso de filtros de redução de escala linear, a taxa de bits é de 14% a 55%. a redução é alcançada para codificação HD e UHD em taxas de bits e configurações padrão usadas em implantações comerciais. Um efeito importante da pré-codificação é que a codificação e decodificação de vídeo podem ser aceleradas, uma vez que muitos GOPs tendem a ser reduzidos pelo pré-codificador para apenas 6% -64% do seu tamanho original, dependendo da redução de escala selecionada.

---

[2]Apesar da abundância de filtros de upscaling, para permanecer eficiente em uma infinidade de dispositivos clientes, a maioria dos navegadores da Web suporta apenas o filtro bilinear para upscaling de imagem e vídeo no espaço de cores YUV, por exemplo, consulte o Código-fonte do cromo que usa libyuv.

fator. Isso é benéfico ao considerar implantações de tais codificadores na nuvem, especialmente tendo em vista os padrões futuros, como AV1 e MPEG VVC, que aumentaram substancialmente a complexidade de codificação. Alternativamente, essa aceleração pode ser trocada por predefinições de codificação mais complexas para entradas de resolução mais baixa, o que permite maior economia na taxa de bits.

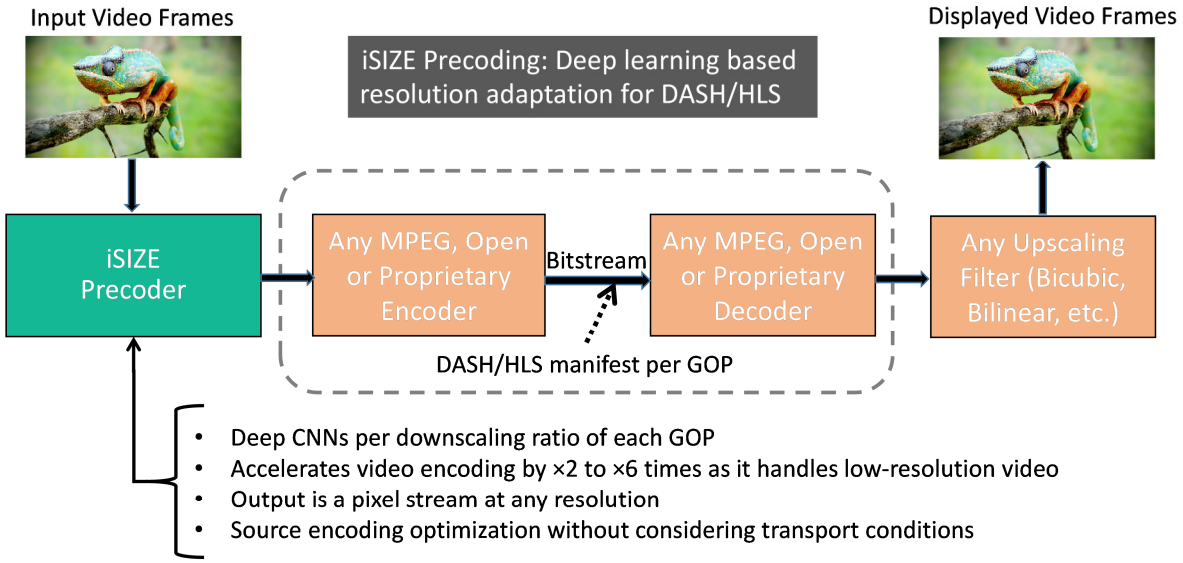

Figura 1. Proposta de pré-codificação baseada em aprendizagem profunda para streaming de vídeo DASH/HLS. A adaptação com redes neurais convolucionais (CNNs) ocorre por grupo de imagens (GOP). O lado do decodificador pode opcionalmente ser ajustado para obter maior melhoria de qualidade.

É importante enfatizar*a principal diferença entre a pré-codificação e a otimização ladder DASH/HLS convencional* : A pré-codificação é uma abordagem de otimização de codificação de origem e é realizada no lado do servidor antes da transmissão, a fim de selecionar e derivar de maneira ideal a melhor representação reduzida por segmento GOP de entrada, taxa de bits e especificação de codec, *sem considerar as condições de transporte*. Ou seja, uma vez que a escada de taxas de bits DASH/HLS é produzida por nossa abordagem (com a resolução GOP e redução de escala CNN por taxa de bits selecionada automaticamente por nossa abordagem), streaming de taxa de bits adaptativo, cache de fluxo e mecanismos de comutação de fluxo que lidam com taxas de bits e flutuações de canal pode funcionar normalmente. A principal diferença é que eles recebem, decodificam (e aprimoram) representações personalizadas criadas pelos pré-codificadores CNNs. Conseqüentemente, nossa proposta para pré-codificação baseada em aprendizado profundo é reduzir a distorção do player de vídeo DASH/HLS por taxa de bits, explorando o fato de que o upscaling é suportado pelo player e a pré-codificação permanece independente das condições do canal durante cada entrega individual de fluxo de bits de vídeo.

TRABALHO RELATADO

A codificação adaptativa de conteúdo surgiu como uma solução popular para ganhos de qualidade ou taxa de bits na codificação de vídeo baseada em padrões. A maioria dos fornecedores comerciais já demonstrou soluções de codificação adaptáveis ao conteúdo, normalmente na forma de adaptação da taxa de bits baseada na combinação de métricas perceptivas, ou seja, reduzindo a taxa de bits da codificação para cenas que são

considerado simples o suficiente para um codificador padrão processar com 50% (ou menos) de bits. Tais soluções também podem ser estendidas a outros parâmetros do codificador, e sua essência está no acoplamento de um perfil de qualidade visual a uma receita de ajuste específica do codificador pré-preparada.

A adaptação da resolução na codificação de imagens e vídeos tem sido explorada por diversos autores. Recentemente, Katsavounidis*e outros.*[3][4] propuseram a noção de otimizador dinâmico na codificação de vídeo: cada cena é reduzida para uma faixa de resoluções e subsequentemente compactada para uma faixa de taxas de bits. Após o upscaling para resolução total, o casco convexo de taxas de bits/qualidades é produzido para selecionar a melhor resolução operacional e configurações de codificação para cada grupo de imagens no vídeo. A qualidade pode ser medida com uma ampla gama de métricas, desde uma simples relação sinal-ruído de pico (PSNR) até métricas complexas baseadas em fusão, como VMAF [5]. Embora ganhos de 30% na taxa BD [6] tenham sido demonstrados em experimentos para H.264/AVC e VP9, o otimizador dinâmico é extremamente caro para implantar em um cenário operacional e ainda usa filtros de redução de escala não aprendíveis.

No geral, embora tais métodos tenham mostrado a possibilidade de economia de taxa por meio do downscaling de imagem e vídeo, eles não conseguiram superar o downscaling bicúbico clássico no contexto da codificação prática. Isso levou a maioria dos pesquisadores a concluir que o downscaling com filtros bicúbicos ou Lanczos é a melhor abordagem e, em vez disso, o foco mudou para soluções de upscaling no lado do cliente (ou seja, decodificador) que aprendem a recuperar detalhes da imagem assumindo tais operadores de downscaling. Para este fim, arquiteturas de redes neurais convolucionais profundas (CNN) estabeleceram o que há de mais moderno, com CNNs profundas recentes como VDSR [1] e EDSR [7] alcançando PSNR vários dB mais alto no canal de luminância de benchmarks de imagem padrão para upscaling de imagem sem perdas.

Inspirado por esses sucessos e pelo sucesso dos autoencoders para compactação de imagens, a Wave One propôs recentemente a codificação de vídeo com redes neurais profundas [8] e demonstrou ganhos de qualidade em relação a um codificador de vídeo convencional sem quadros B, e focando na codificação de taxa de bits muito alta (20mbps ou maior para HD). Embora esta seja uma conquista importante, tais soluções ainda não mostraram desempenho superior em relação aos codificadores de vídeo modernos quando estes utilizam todos os seus recursos (por exemplo, codificação VBV e suas configurações mais avançadas, como predefinição “mais lenta” AVC/H.264 libx264). Além disso, eles exigem recursos avançados de GPU tanto no lado do cliente quanto no lado do servidor e não oferecem suporte a padrões para transporte e decodificação de vídeo. Portanto, apesar dos avanços que podem ser oferecidos por todos esses métodos no futuro, eles não são compatíveis com os padrões de codificação de vídeo e não consideram as rigorosas restrições de complexidade impostas ao streaming de vídeo em thin clients compatíveis com DASH ou HLS, como tablets e dispositivos móveis. . Nosso trabalho preenche essa lacuna, oferecendo uma solução baseada em aprendizagem profunda que pode operar integralmente no lado do servidor e não requer nenhuma alteração no transporte, decodificação e exibição do vídeo.

REDES DE PRÉ-CODIFICAÇÃO

A rede neural de pré-codificação multiescala proposta compreende uma série de blocos de pré-codificação, que reduzem progressivamente os quadros de alta resolução (HR) em múltiplos fatores de escala correspondentes aos de qualquer escada DASH/HLS designada e operam inteiramente no lado do codificador antes da transmissão. Os quadros de baixa resolução (LR) resultantes podem então ser codificados pelo codec com menor complexidade e maior eficiência de codificação, porque projetamos as CNNs de pré-codificação para compactar informações de maneira que o padrão linear

o upscaler do lado do jogador será capaz de se recuperar com mais eficiência do que o caso de um downscaler linear. Em relação à economia de complexidade de codificação/decodificação, para redução de escala por fator -, dado - -1,o número de pixels a codificar é um fator - menor que o quadro original. O vídeo reduzido pode ser opcionalmente transcodificado e transmitido a uma taxa de bits substancialmente mais baixa para um thin client (por exemplo, um dispositivo móvel). Do lado do cliente, o vídeo é decodificado e aumentado para a resolução original (HR), com um upscaler linear simples, sendo o caso mais comum o filtro bilinear. Isso é contrário às arquiteturas recentes de aumento de escala de imagem que assumem um simples downscaling bicúbico e uma arquitetura CNN de super-resolução extremamente complexa no lado do player de vídeo. Por exemplo, o upscaling EDSR [7] compreende mais de 40 milhões de parâmetros e seria, portanto, impraticável no lado do cliente para vídeo HD/UHD de 30-60 quadros por segundo (fps).

Nas seções a seguir, resumimos a metodologia de projeto das redes de pré-codificação multiescala propostas, incluindo a arquitetura da rede e a seleção on-line do melhor modo de pré-codificação para cada GOP.

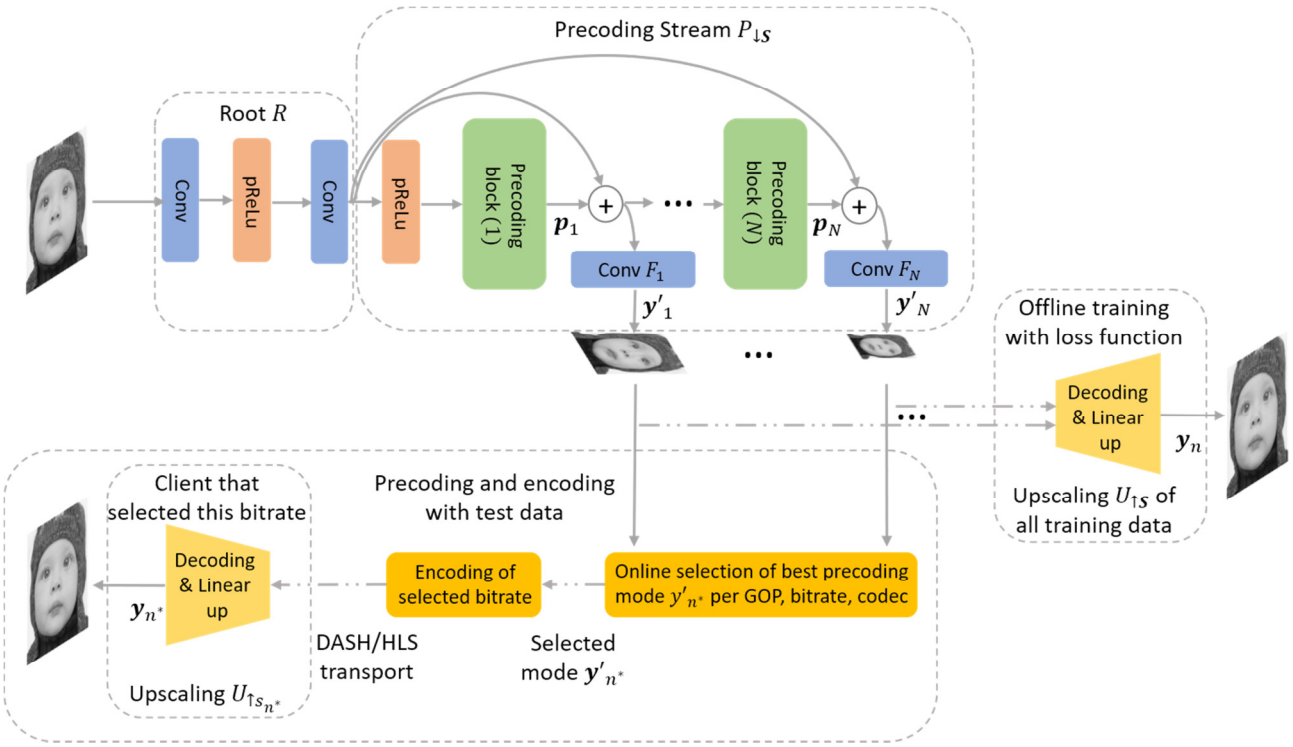

Figura 2. Rede de pré-codificação multiescala para downscaling de vídeo antes da codificação, compreendendo um mapeamento de raiz, fluxo de pré-codificação único e upsampling linear. Cada quadro de entrada de um vídeo é reduzido pela rede de pré-codificação no servidor por meio de blocos de pré-codificação, correspondentes aos pontos de adaptação DASH/HLS do cliente, que são usados durante o treinamento da rede. Na implantação de teste, os quadros reduzidos do modo selecionado são passados para o codec para codificação em qualquer taxa de bits escolhida. Durante a transmissão ao vivo, o dispositivo cliente seleciona uma taxa de bits por GOP e decodifica e aumenta a amostragem do vídeo reduzido com aumento de escala linear simples, por exemplo, o filtro bilinear.

Arquitetura de rede

Visualizamos a rede de pré-codificação na Figura 2 para redução progressiva de quadros de luminância (Y) individuais∈ (onde e são a altura e a largura respectivamente) em múltiplas escalas. Dado que os telespectadores são mais sensíveis às informações de luma, processamos intencionalmente apenas o canal Y e não os canais de crominância (Cb, Cr), a fim de evitar cálculos desnecessários. Dong*e outros.*[9] apoiam esta afirmação empiricamente e, adicionalmente, descobrem que treinar uma rede em todos os três canais pode, na verdade, piorar o desempenho devido à queda da rede em um mínimo local ruim. Observamos também que isso permite a subamostragem de croma (por exemplo, YUV420), pois os canais de crominância podem ser reduzidos e processados de forma independente.

Primeiro realizamos a extração de características (sem redução de escala) usando um mapeamento "raiz", compreendendo duas camadas convolucionais, a fim de extrair um conjunto de mapas de características de alta dimensão da entrada. Os mapas de recursos extraídos são então passados para o fluxo de pré-codificação. Um fluxo de pré-codificação↓(ou para abreviar) compreende uma sequência de blocos de pré-codificação, que reduzem progressivamente a entrada em um conjunto de fatores de escala, - , - , … , - ,onde - -… - e↓indica redução de escala. Para pré-codificação blocos e fatores de escala , esperamos que o fluxo de incorporação produza um conjunto de ' representações reduzidas , , … , da entrada. As ativações de saída em ' são então cortados (redimensionados) entre as intensidades de pixel mínima e máxima e cada representação pode, portanto, ser usada pelo codec de vídeo como um quadro de baixa resolução (LR) reduzido. Esses quadros podem então ser aumentados para a resolução original usando upscaling linear ↑"do lado do cliente, como bilinear, lanczos ou bicúbico. Referimo-nos ao #ésimo quadro ampliado gerado como$.

Aprendizagem Residual Global

Conforme ilustrado na Figura 2, nosso fluxo de pré-codificação utiliza uma estratégia de aprendizagem residual global, onde usamos uma conexão de salto e realizamos uma soma pixel a pixel entre os mapas de recursos raiz (função de pré-ativação e após redução linear para a resolução correta) e o saídas de cada bloco de pré-codificação. Implementações semelhantes de aprendizagem residual global também foram adotadas por modelos SR [1] [7]. No nosso caso, nosso fluxo de pré-codificação segue efetivamente uma configuração de pré-ativação sem normalização em lote. Descobrimos empiricamente que a convergência durante o treinamento é geralmente mais rápida com o aprendizado residual global, já que os blocos de pré-codificação só precisam aprender o mapa residual para remover a distorção introduzida pelas operações de redução de escala.

Bloco de pré-codificação

Nossos blocos de pré-codificação, que constituem os componentes primários de nossa rede, são ilustrados na Figura 3. O bloco de pré-codificação consiste em alternar3 3e1 1 camadas convolucionais, onde cada camada é seguida por uma função de ativação paramétrica ReLu (pReLu). O1 1a convolução é usada como um meio eficiente para redução de canal, a fim de reduzir o número geral de acumulações múltiplas (MACs) para computação.

O #º bloco de pré-codificação é efetivamente responsável pela redução de escala por fator -$⁄-$&.Para manter a baixa complexidade, é importante reduzir a escala o mais cedo possível. Portanto, agrupamos todas as operações de redução da resolução com a primeira camada convolucional em cada

bloco de pré-codificação. Se -$⁄-$&é divisível por 2, simplesmente usamos um passo de 2 na primeira camada convolucional. Se -$⁄-$&não é divisível por 2, usamos um passo de 1 na primeira camada convolucional e precedemos isso com uma redução linear bilinear/bicúbica para o fator de escala correto.

O bloco de pré-codificação também deve ser capaz de remover artefatos de alias gerados pela redução da resolução com passos largos: considerando que as operações de aumento de escala são apenas lineares e, portanto, fortemente restritas, a rede é assimétrica e a rede de pré-codificação não pode simplesmente aprender a inverter o aumento de escala. Um filtro anti-aliasing linear tradicional é um filtro passa-baixo que remove os componentes de alta frequência da imagem, de modo que a imagem possa ser resolvida adequadamente. No entanto, ao remover as altas frequências da imagem, isso leva ao desfoque. A desfocagem é um problema inverso, pois normalmente há informações limitadas na imagem desfocada para determinar exclusivamente uma entrada viável. Como tal, podemos modelar respectivamente os processos de anti-aliasing e desblurring como uma composição funcional de mapeamentos lineares e não lineares. Conforme mostrado na Figura 3, isso é feito com uma conexão de salto entre caminhos lineares e não lineares abaixo, novamente seguindo a estrutura de pré-ativação. Para garantir que a saída do caminho não linear tenha alcance completo ()∞, +∞,,podemos inicializar o pReLu final antes da conexão de salto de forma que se aproxime de uma função de identidade.

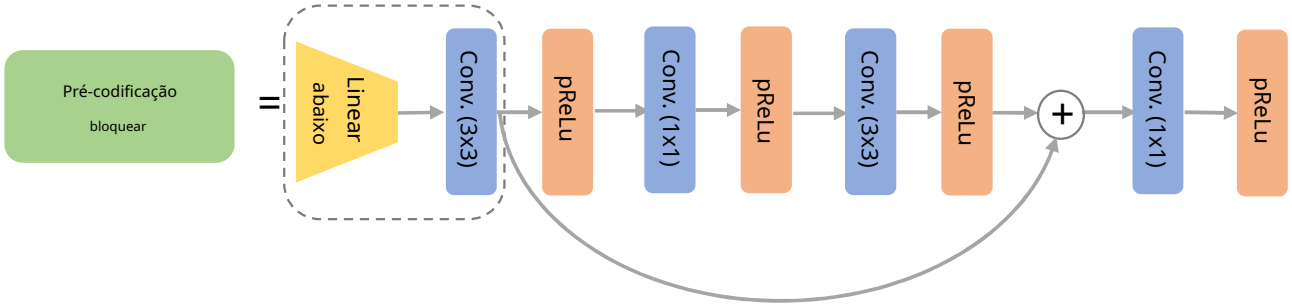

Figura 3. Design de bloco para pré-codificação eficiente, composto por uma série de3 3e1 1 convoluções. A primeira camada é agrupada com uma operação de redução de escala via passo na camada convolucional ou com uma redução de escala bicúbica/bilinear anterior. O mapeamento linear aprendido na primeira camada (função de pré-ativação) é passado para a saída da segunda 3 3camada conv (função pós-ativação) com uma conexão de salto e soma de pixels.

Seleção online do melhor modo de pré-codificação durante a codificação GOP

Depois que a arquitetura de pré-codificação multiescala da Figura 2 tiver sido treinada no conteúdo representativo, cada segmento GOP do vídeo será pré-codificado em todas as escalas possíveis e um processo de seleção on-line determinará o melhor modo de pré-codificação a ser usado. A ideia principal é percorrer tantos modos de pré-codificação quanto possível e avaliar uma função de distorção de taxa que determine as características de distorção de taxa do codificador de vídeo se ele usar esse modo. A avaliação das características de distorção de taxa determina o melhor modo por GOP, que é então usado para a codificação real. O decodificador simplesmente decodifica o modo fornecido e aumenta para a resolução total usando o filtro de aumento de escala integrado do player. Dado que a adaptação de resolução já é suportada por segmento de vídeo DASH/HLS, nenhuma modificação é necessária em todo o fluxo de bits, transporte, DASH/HLS e lado do cliente para apoiar nossa abordagem.

Múltiplas funções de distorção de taxa podem ser selecionadas para a avaliação do melhor modo de pré-codificação. Por exemplo, pode-se utilizar modelos operacionais de distorção de taxa para H.264/AVC ou

HEVC [10]. Em vez disso, realizamos a codificação seletiva de alguns quadros em cada GOP com um processo que chamamos de “footprinting”. Uma vez que a taxa-MSE[3]características dessas codificações rápidas são extraídas do codec utilizado, eliminamos todos os pontos que não fornecem uma curva convexa de distorção de taxa (RD). Dos pontos restantes, selecionamos o ponto que corresponde à distorção mais baixa quando todos os pontos são recodificados para sua taxa de bits média com codificação de taxa de bits constante. Essa função de pegada, remoção e remapeamento para selecionar o melhor ponto é rápida de calcular se apenas alguns quadros forem usados e ainda permanece independente do codec, pois utiliza o codec para obter os pontos RD operacionais necessários.

Detalhes de implementação e treinamento

Em nossa rede de pré-codificação multiescala proposta, todos os kernels são inicialmente configurados usando a inicialização Xavier [11]. Usamos ReLu paramétrico (pReLu) como função de ativação, conforme indicado na Figura 2 e Figura 3, e preenchimento de zero para garantir que todas as camadas sejam do mesmo tamanho. A redução da escala só é controlada por operações de redução da resolução, como uma passada. O mapeamento raiz compreende um único3 3e1 1camada convolucional. Definimos o número de canais em todos1 1e3 3 camadas convolucionais para 4 e 8 respectivamente (excluindo camadas convolucionais de canal único -). Dessa forma, por um1920 1080 4-mapa dimensional de recursos (assumindo que não há redução de escala), um único bloco de pré-codificação requer aproximadamente apenas 1,33 G MACs para redução de escala e 640 parâmetros. Nossa implementação final requer apenas MACs de 3,38 G e parâmetros de 5,5 K em todas as escalas para um1920 1080quadro de entrada (incluindo mapeamento raiz e todos os fluxos de pré-codificação).

Figura 4. Comparação subjetiva e objetiva em exemplos do conjunto de validação DIV2K.

Treinamos o módulo raiz e todos os fluxos de pré-codificação de ponta a ponta com upscaling linear (sem qualquer codificação) em imagens do conjunto de treinamento DIV2K [12] usando um erro médio-absoluto composto (MAE) e uma função de perda baseada em gradiente. Isto significa que o nosso

[3]Embora pudéssemos usar outras medidas de distorção, o erro quadrático médio (MSE) é rápido de calcular e é fornecido automaticamente por todos os codificadores.

O objetivo do treinamento não é apenas minimizar o MAE de maneira dependente dos dados, mas também levar em consideração os aspectos estruturais das imagens de treinamento. Todos os modelos foram treinados com o otimizador Adam com tamanho de lote de 32 para 200 mil iterações. A taxa de aprendizado inicial é definida como 0,001 e reduzida por um fator de 0,1 em 100 mil iterações. Usamos aumento de dados durante o treinamento, invertendo aleatoriamente as imagens e treinando com um120 120 colheitas aleatórias extraídas das imagens DIV2K. Todos os experimentos foram conduzidos em Tensorflow em GPUs NVIDIA K80. É importante ressaltar que, para garantir que nossa implementação corresponda ao upscaling linear padrão (por exemplo, FFmpeg ou OpenCV), escrevemos todas as funções de upscaling linear do zero e não usamos as funções integradas do Tensorflow. Resultados indicativos para partes de duas imagens de validação do DIV2K são apresentados na Figura 4.

AVALIAÇÃO DE MODOS DE PRÉ-CODIFICAÇÃO EM SEQUÊNCIAS DE VÍDEO HD E UHD

Para avaliar nossa proposta de cenários com o maior impacto prático, avaliamos fatores de escala padrão para codificação de vídeo HD e UHD dentro de regimes típicos de taxa de bits e nos concentramos nos dois padrões MPEG em uso atualmente, ou seja, H.264/AVC e HEVC sob suas implementações FFmpeg libx264 e libx265 e conteúdo de teste padrão[4]. Após o upscaling bilinear padrão com o libyuv que é suportado por todos os players e navegadores da web, a distorção é medida via PSNR e VMAF [5], sendo o último usado pela comunidade de streaming de vídeo como uma métrica de qualidade visual autointerpretável de 0-100 que funde vários índices de qualidade individuais.

	AVC/H.264				HEVC			
	bicúbico		Lanços		bicúbico		Lanços	
Fator	Taxa BD	BD-PSNR	Taxa BD	BD-PSNR	Taxa BD	BD-PSNR	Taxa BD	BD-PSNR
5/2	- 24,70%	0,61dB	- 19,21%	0,45dB	- 25,17%	0,55dB	- 18,84%	0,39dB
2	- 18,85%	0,56dB	- 14,71%	0,42dB	- 19,25%	0,52dB	- 14,46%	0,37dB
3/2	- 17,11%	0,45dB	- 11,75%	0,31dB	- 13,18%	0,32dB	- 8,26%	0,20dB

Tabela 1. Economia na taxa BD de nossa pré-codificação para sequências HD para PSNR.

	AVC/H.264				HEVC			
	bicúbico		Lanços		bicúbico		Lanços	
Fator	Taxa BD	BD-VMAF	Taxa BD	BD-VMAF	Taxa BD	BD-VMAF	Taxa BD	BD-VMAF
5/2	- 39,74%	7,86	- 34,30%	6,49	- 39,73%	7.03	- 33,75%	5,74
2	- 30,32%	5,81	- 27,57%	5.18	- 30,20%	5.12	- 27,41%	4,57
3/2	- 23,21%	3,43	- 21,73%	3.18	- 18,66%	2,61	- 17,67%	2,46

Tabela 2. Economia na taxa BD de nossa pré-codificação para sequências HD para VMAF.

[4]Configuração: predefinição “média” libx264/libx265, GOP=30 quadros e faixa de taxa de bits de 0,5-10mbps para HD, GOP=50 e faixa de 3-17mbps para UHD, controle de taxa de duas passagens:https://trac.ffmpeg.org/wiki/Encode/H.264#twopass ; o conteúdo do teste dehttps://media.xiph.org/video/derf/ o site era: Aspen, Blue Sky, Controlled Burn, Rush Field Cuts, Sunflower, Rush Hour, Old Town Cross, Crowd Run, Tractor, Touchdown, Riverbed, Red Kayak, West Wind Easy, Área de pedestres, Ducks Take Off, Park Joy. As sequências UHD usadas foram (apenas os primeiros 240 quadros, cortando para3840 2160da parte central e conversão sem perdas para o formato YUV420 antes da codificação): Barscene, Boat, Crosswalk, Dancers, Dinnerscene, DrivingPOV, Foodmarket, Foodmarket2, Narrator.

Os ganhos da taxa de distorção de Bjontegaard [6] são mostrados na Tabela 1-Tabela 4. Nossa pré-codificação mostra-se particularmente eficaz para o caso de VMAF, onde a redução da taxa de 18% -40% e 35% -55% é obtida para HD sequências e sequências UHD, respectivamente. Para o caso de sequências UHD, os dois modos mostrados foram suficientes para a cobertura de toda a região de taxa de bits UHD (2,5-10mbps), portanto o fator de redução de escala do modo 3/2 é omitido.

	AVC/H.264				HEVC			
	bicúbico		Lanços		bicúbico		Lanços	
Fator	Taxa BD	BD-PSNR	Taxa BD	BD-PSNR	Taxa BD	BD-PSNR	Taxa BD	BD-PSNR
5/2	- 19,66%	0,26dB	- 12,60%	0,17dB	- 18,93%	0,25dB	- 11,71%	0,15dB
2	- 6,45%	0,11dB	- 2,99%	0,05dB	- 8,30%	0,12dB	- 4,39%	0,07dB

Tabela 3. Economia na taxa BD de nossa pré-codificação para sequências UHD para PSNR.

	AVC/H.264				HEVC			
	bicúbico		Lanços		bicúbico		Lanços	
Fator	Taxa BD	BD-VMAF	Taxa BD	BD-VMAF	Taxa BD	BD-VMAF	Taxa BD	BD-VMAF
5/2	- 55,15%	5,87	- 48,05%	4,98	- 52,19%	5.41	- 45,26%	4,57
2	- 36,99%	4,34	- 34,57%	4.02	- 39,69%	3,93	- 37,60%	3,69

Tabela 4. Economia na taxa BD de nossa pré-codificação para sequências UHD para VMAF.

AVALIAÇÃO INDICATIVA DE PRÉ-CODIFICAÇÃO ADAPTATIVA VERSUS RESULTADOS DE CODEC

Como a pré-codificação pode ser aplicada a qualquer codec e qualquer resolução de entrada de vídeo, há uma gama virtualmente ilimitada de testes que podem ser realizados para avaliar seu desempenho em diversos cenários de interesse. Apresentamos aqui dois casos representativos como exemplos ilustrativos do efeito impulsionador que a pré-codificação baseada em aprendizagem profunda pode ter em codecs e ecossistemas de transporte existentes sem quebrar a conformidade com os padrões.

O primeiro, apresentado na parte esquerda da Figura 5, refere-se à melhoria do desempenho da codificação VP9 para conteúdo HD em toda a gama de taxas de bits de qualidade de interesse comercial. Para este fim, habilitamos vários modos de pré-codificação (fatores de redução de escala variando de -5⁄4 para -4)e utilizou as 16 sequências HD usadas em nossos testes anteriores em conjunto com libvpx-vp9 do FFmpeg[5]. Para ilustrar a competitividade da configuração de codificação utilizada, também fornecemos os resultados correspondentes do AWS Elastic Transcoder. As configurações dos trabalhos do Elastic Transcoder foram baseadas nas predefinições integradas[6], que personalizamos para corresponder ao codec de vídeo de saída desejado, resolução, taxa de bits e tamanho do GOP, e definimos a taxa de quadros de acordo com a taxa de quadros do vídeo de entrada. Tal customização é necessária porque os presets integrados não seguem os parâmetros de entrada do vídeo e servem

[5]Configuração: codificação de qualidade constante com taxa min-max (veja https://trac.ffmpeg.org/wiki/Encode/VP9), GOP=90 frames, maxrate=1,45×minrate, speed=0 para codificação de resolução mais baixa e speed=2 para codificação de resolução total, já que codificamos versões reduzidas com 6% a 64% dos pixels de vídeo da resolução original . Observe que a redução adicional da taxa de bits pode ser alcançada utilizando a codificação VBV em libvpx-vp9, mas optamos por não usar a codificação VBV para equilibrar nossa comparação com a implementação VP9 fornecida pelo AWS Elastic Transcoder.

[6]https://docs.aws.amazon.com/elastictranscoder/latest/developerguide/preset-settings.html

principalmente como clichês. Nossa proposta oferece redução de taxa de 23% e 30% em relação ao libvpx-vp9 e ao Elastic Transcoder, respectivamente.

O segundo cenário refere-se a melhorar o desempenho de um codificador H.264/AVC já altamente otimizado para conteúdo HD e UHD e evitar a mudança para um codec mais complexo como HEVC ou VP9. A parte direita da Figura 5 mostra os resultados obtidos com H.264/AVC[7] e a pré-codificação proposta. Para ilustrar que nossos ganhos são alcançados em uma configuração comercialmente competitiva, incluímos resultados com o AWS MediaConvert usando o perfil MULTI_PASS_HQ AVC/H.264, que personalizamos para definir a resolução e a taxa de bits do vídeo de saída. Para o modo QVBR do MediaConvert, usamos o valor padrão do nível de qualidade 7. Para codificação HD, nossa proposta oferece redução de taxa de 28% e 43% em relação a libx264 e MediaConvert, respectivamente.

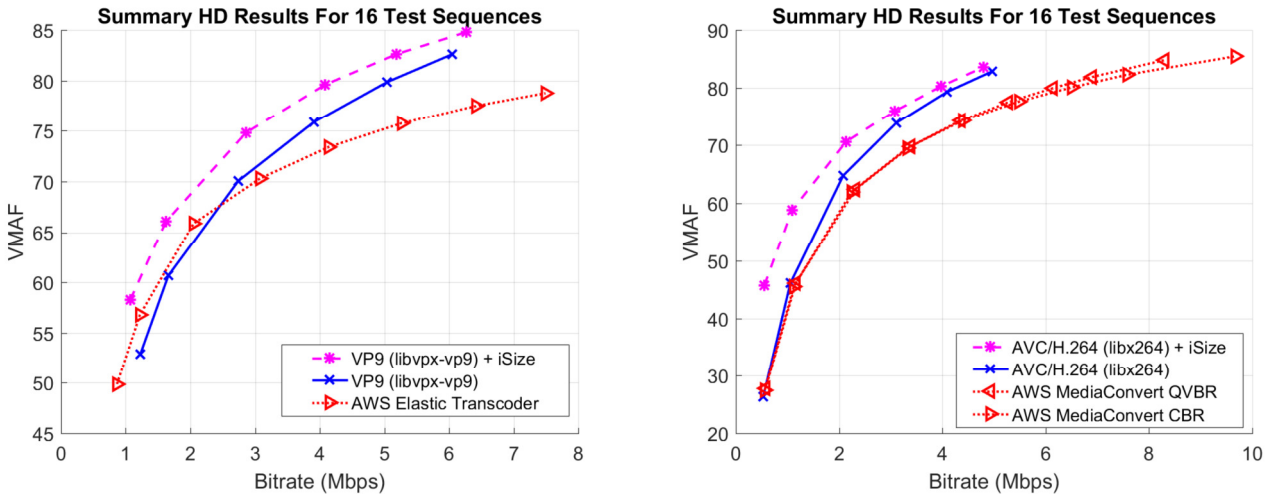

Figura 5.**Esquerda:**Codificação VP9 com pré-codificação habilitada versus VP9 independente e o codec AWS Elastic Transcoder VP9.**Certo:**Codificação AVC/H.264 com pré-codificação habilitada versus AVC/H.264 independente e o codec AWS MediaConvert H.264/AVC.

CONCLUSÃO

Propomos o conceito de pré-codificação adaptativa para vídeo baseado em redes neurais convolucionais profundas, com o foco atual na compactação baseada em downscaling sob adaptação DASH/HLS. Um aspecto fundamental da nossa abordagem é que ela permanece compatível com os sistemas existentes e não requer nenhuma alteração no decodificador e no display. Dado que a nossa abordagem não altera o processo de codificação, ela oferece uma dimensão de otimização adicional que vai além da codificação adaptável ao conteúdo e da otimização dos parâmetros do codec. Na verdade, os experimentos mostram que ele traz benefícios além dessas otimizações bem conhecidas: sob taxas de bits realistas de streaming de vídeo HD e UHD e configurações de codificador de última geração, nossa pré-codificação oferece redução de 14% a 55% na taxa de bits em relação a soluções de downscaling linear para AVC/H.264 e HEVC, com potencial substancial para melhorias adicionais. Além disso, para condições realistas de codificação AVC/H.264 e VP9,

[7]Configuração: predefinição “mais lenta”, GOP=90; como o AWS MediaConvert oferece suporte à codificação VBV por meio do modo QVBR, optamos pela codificação VBV para libx264:https://trac.ffmpeg.org/wiki/Encode/H.264#AdditionalInformationTips com crf=18 para todos -5 2e crf=23 para todos -

descobriu-se que a pré-codificação oferece economia de taxa de 23% a 43% em relação ao codificador equivalente. Seus recursos de compactação garantem não apenas a economia da taxa de bits, mas também a redução da complexidade da codificação de vídeo. Trabalhos futuros podem considerar como estender a noção de pré-codificação além dos sistemas DASH/HLS, aprendendo a pré-processar adaptativamente as entradas de vídeo, de modo que sejam recuperadas de maneira ideal pelos decodificadores atuais sob faixas de interesse de taxa de bits especificadas.

REFERÊNCIAS

[1] J.Kim, *e outros.*, "Superresolução de imagem precisa usando redes convolucionais muito profundas," *Processo. Conferência IEEE. Computação. Padrão de Visão Rec.*, pp.

[2] A. Wiesel, *e outros.*, "Pré-codificação linear via otimização cônica para receptores MIMO fixos," *IEEE Trans. Processo de sinal.*, vol. 54, não. 1, páginas 161-176, 2006.

[3] I. Katsavounidis, "Otimizador dinâmico – Uma estrutura de otimização de codificação de vídeo perceptual," *O blog de tecnologia da Netflix*, 2018.

[4] CG Bampis, *e outros.*, "Rumo ao streaming de vídeo adaptativo de ponta a ponta, perceptualmente otimizado", pré-impressão arXiv, *arXiv:1808.03898*, 2018.

[5] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy e M. Manohara, "Rumo a uma métrica prática de qualidade de vídeo perceptual", *O blog de tecnologia da Netflix*, 6, 2016.

[6] G. Bjontegaard, "Cálculo de av. Diferencial PSNR. entre. Curvas RD," *VCEG-M33*, 2001.

[7] B. Lim, *e outros.*, "Redes residuais profundas aprimoradas para super-resolução de imagem única," *Processo. Conferência IEEE. Computação. Padrão de Visão Rec.*Trabalho, pp. 136–144, 2017.

[8] O.Rippel, *e outros.*, "Compressão de vídeo aprendida", pré-impressão arXiv, *arXiv:1811.06981*(2018).

[9] C.Dong, *e outros.*, "Super-resolução de imagem usando redes convolucionais profundas," *IEEE Trans. Padrão Anal. e Máquina Intel.*, vol. 38, não. 2, pp. 295–307, fevereiro de 2016.

[10] S. Ma, *e outros.*"Análise de distorção de taxa para codificação de vídeo H. 264/AVC e sua aplicação para controle de taxa", *IEEE Trans. CSVT, vol.*15, no.12, pp. 1533-1544, dezembro de 2005.

[11] X. Glorot e Y. Bengio, "Compreendendo a dificuldade de treinar redes neurais feedforward profundas," *Processo. 13ºInternacional Conf. Artefato. Intel. e Estado.*, pp.

[12] E. Agustsson e R. Timofte, "Desafio NTIRE 2017 em super-resolução de imagem única: conjunto de dados e estudo," *Processo. Trabalho IEEE CVPR.*, pp.