

## AVALIAÇÃO DO MÉTODO SPATIO-TEMPORAL DEFORMABLE CONVOLUTION PARA REDUÇÃO DE ARTEFATOS DE COMPRESSÃO EM VÍDEOS DIGITAIS

GILBERTO KREISLER<sup>1</sup>; GARIBALDI DA SILVEIRA<sup>2</sup>; GUILHERME CORRÊA<sup>3</sup>

<sup>1</sup>Universidade Federal de Pelotas – ViTech – gkfneto@inf.ufpel.edu.br

<sup>2</sup>Universidade Federal de Pelotas – ViTech – garibaldi.dsj@inf.ufpel.edu.br

<sup>3</sup>Universidade Federal de Pelotas - ViTech – gcorrea@inf.ufpel.edu.br

### 1. INTRODUÇÃO

Os avanços em tecnologias de dispositivos de vídeo e o aumento da velocidade da internet pavimentaram o caminho para a popularização de vídeos digitais pela internet. Um estudo do CISCO (2020) aponta que o fluxo de vídeos na internet irá ultrapassar 90% até 2023. Um relatório de abril da STATISTA (2020) indica que a transmissão de vídeos pela internet cresceu 32.6% no primeiro mês da pandemia de COVID-19. Porém, a capacidade de transmissão de dados é limitada. Colocando em números, considerando um vídeo de resolução 1920x1080 (HDTV) em tempo-real com 30 quadros por segundo seriam necessários uma taxa de transmissão de 1.5Gb por segundo para ser transmitido. Quanto ao armazenamento, para armazenar um vídeo de 10 minutos de duração na resolução HDTV seriam necessários 112GB.

Diversas técnicas de compressão (CT) de vídeo foram criadas com o propósito de reduzir drasticamente a quantidade de informação de vídeo necessária para armazenar ou transmitir algum vídeo, dando origem aos padrões/formatos AV1, VVC e HEVC. Entretanto, durante o processo de codificação algumas informações são perdidas. Essa perda introduz artefatos de compressão, que degradam a qualidade da experiência do usuário, como é possível observar na Figura 1.



Figura 1: Exemplos de artefatos de compressão (DENG et al., 2020)

Para melhorar a qualidade do quadro de um vídeo existem duas estratégias: filtros *in-loop* e filtros de pós-processamento. Os filtros *in-loop* são parte do codificador/decodificador e estão diretamente relacionadas com as CT. Os filtros de pós-processamento são aplicados depois de todo o processo de decodificação, não tendo a necessidade de ter relação com a CT. Porém, técnicas de pós processamento clássicas ainda são baseadas nos coeficientes transformados de CS específicos, tornando-os difíceis de serem estendidos para outros padrões (DENG et al. 2020). Nos anos mais recentes, diversos estudos têm proposto filtros de pós-processamento baseado em modelos de aprendizagem profunda. Um desses estudos é o DENG et al. (2020) que é o estado-da-arte.

DENG et al. (2020) propõe que a solução poderia ser estendida para múltiplos padrões de compressão. Porém, na fase de teste do modelo proposto, ele foi treinado e avaliado apenas sobre o padrão HEVC, o que dificulta validar se o modelo pode ser realmente estendido para outros esquemas de compressão.

## 2. MÉTODO SPATIO-TEMPORAL DEFORMABLE CONVOLUTION

O trabalho *Spatio-Temporal Deformable Convolution for Compressed Video Quality Enhancement* (STDF) (DENG et al. 2020) utiliza informações de quadros temporalmente vizinhos dentro de um raio, que é um parâmetro da rede. Dessa forma, os quadros alvo e vizinhos são usados para alimentar a rede. Inicialmente, para cada quadro alvo a ser processado, vetores de deslocamento são calculados para cada posição do *kernel* de compensação de movimento (MC), que serão usados no módulo de convolução deformável (DCM).

O DCM funde os quadros enquanto implicitamente aplica a MC. Após, o módulo de aumento de qualidade (QEM) gera um mapa residual, que será somado ao componente de luminância do quadro, aprimorando sua qualidade.

Para treinar o modelo, foram selecionados 130 vídeos não comprimidos de diversas resoluções e conteúdos, onde 106 deles foram separados para treino e o restante para validação. Para o teste foram usados 18 vídeos não comprimidos contendo 450 quadros cada. Todos os vídeos foram comprimidos usando o software de referência do HEVC, HM16.5, com a configuração *Low Delay P* (LDP) com quatro Parâmetros de Quantização (QP): 22, 27, 32 e 37.

Para mensurar a performance do modelo, em relação a qualidade do vídeo restaurado, foram usadas as métricas objetivas *Peak Signal-to-Noise Ratio* (PSNR) e *Structural Similarity Index Measure* (SSIM). A Tabela 1 mostra as variações de PSNR e SSIM entre o vídeo original e o vídeo descomprimido, o qual contém os artefatos de compressão.

Este trabalho investiga se o modelo pode ser usado com outros padrões de compressão, como o AV1, mantendo os pesos do treino feito pelo autor ou não.

Tabela 1: Resultados Quantitativos de  $\Delta$ PSNR (dB) /  $\Delta$ SSIM ( $\times 10^{-2}$ ) médio nos Vídeos de Teste em 4 Diferentes QPs. (DENG et al., 2020)



QP	Métodos de QE de Imagens			Métodos de QE de Vídeo				
	Dong et al. AR-CNN	Zhang et al. DnCNN	Zhang et al. RNAN <sup>‡</sup>	Yang et al. MFQE 1.0	Guan et al. MFQE 2.0	STDF-R1	STDF-R3	STDF-R3L
37	0.25 / 0.50	0.36 / 0.69	0.44 / 0.95	0.46 / 0.88	0.56 / 1.09	0.65 / 1.18	0.75 / 1.32	0.83 / 1.51
32	0.19 / 0.17	0.33 / 0.41	0.41 / 0.62	0.43 / 0.58	0.52 / 0.68	0.64 / 0.77	0.73 / 0.87	0.86 / 1.04
27	0.16 / 0.09	0.33 / 0.26	- / -	0.40 / 0.34	0.49 / 0.42	0.59 / 0.47	0.67 / 0.53	0.72 / 0.57
22	0.13 / 0.04	0.27 / 0.14	- / -	0.31 / 0.19	0.46 / 0.27	0.51 / 0.27	0.57 / 0.30	0.63 / 0.34

### 3. METODOLOGIA

Devido a limitações de *hardware* e tempo foram comprimidos apenas 32 quadros de cada vídeo do conjunto de dados utilizado e foram escolhidos apenas dois CQs, ao invés de quatro. Para o treino e teste foi utilizado uma NVIDIA GeForce RTX 3070.

A metodologia de treino utilizada é a que está descrita no repositório do projeto em XING; DENG (2020), que é a mesma descrita no artigo com algumas pequenas modificações. O raio de quadros utilizado na rede foi fixado em 3.

Em posse dos vídeos originais, cada um foi comprimido pelo padrão AV1 utilizando o *software* de referência LibAOM v3.3.0 com uma Restrição de Qualidade (CQ) de 43 e 55. Esses parâmetros foram escolhidos para tentar se aproximar ao máximo dos parâmetros de codificação utilizados no STDF.

Inicialmente estes vídeos foram alimentados na rede utilizando os pesos fornecidos pelos autores, ou seja, os pesos do treinamento sobre os vídeos comprimidos pelo padrão HEVC. Num segundo momento, a rede foi treinada sobre os vídeos comprimidos pelo padrão AV1 e após, foram executados os testes.

Devido algumas limitações, não foi possível gerar os resultados de SSIM, sendo apresentados apenas o PSNR.

### 3. RESULTADOS E DISCUSSÃO

A Tabela 2 mostra os resultados de  $\Delta$ PSNR do modelo sobre vídeos comprimidos pelo padrão AV1. A primeira macro coluna apresenta os resultados utilizando os pesos fornecidos pelos autores, ou seja, com o treinamento feito sobre vídeos comprimidos pelo padrão HEVC. A segunda macro coluna apresenta os resultados utilizando os pesos do treinamento com vídeos comprimidos no padrão AV1.

Tabela 2: Resultados Quantitativos de  $\Delta$ PSNR (dB) médio nos Vídeos de Teste Comprimidos no Padrão AV1 da rede com Pesos de Treinamento com HEVC e AV1

Treinamento HEVC		Treinamento AV1	
CQ = 43	CQ = 55	CQ = 43	CQ = 55
-1,016	-0,22	0,228	0,208

Como mencionado anteriormente, o modo de teste da rede não retorna o vídeo aprimorado, apenas o  $\Delta$ PSNR, tornando assim impossível de calcular o  $\Delta$ SSIM e fazer uma comparação mais justa. Para isso, seriam necessárias modificações no código que por questões de tempo não foi possível.

Assumindo equivalência, os resultados sob QP 32 e CQ 43, e QP 37 e CQ 55 podem ser diretamente comparados com os resultados sob a coluna STDF-R3.

Como é possível observar, o modelo treinado pelos autores não apresentou um bom desempenho com vídeos comprimidos no padrão AV1, gerando valores negativos, ou seja, a qualidade do vídeo piorou após passar pela rede.

A drástica diferença dos valores de melhoria de qualidade em relação ao artigo do STDF pode ser atribuída ao reduzido número de quadros nos vídeos comprimidos pelo AV1 para o treinamento da rede.

Observando a Tabela 1, é perceptível que conforme aumenta o QP maior é a média, em  $\Delta$ PSNR, da melhoria. É possível atribuir isto ao fato que vídeos codificados com maiores QPs são os vídeos mais degradados e, consequentemente, possuem mais espaço para melhoria. Portanto, este era o comportamento esperado da rede treinada com vídeos comprimidos no padrão AV1. Porém, não foi o que aconteceu. Olhando direto para a média, é possível observar que os vídeos comprimidos com menor CQ obtiveram uma melhoria maior. Mais análises são necessárias para saber o motivo disso.

#### 4. CONCLUSÕES

É possível concluir que o modelo teve uma baixa performance com vídeos comprimidos pelo padrão AV1, não servindo como filtro de pós-processamento para múltiplos padrões da forma que foi treinado. Porém, treinando o modelo especificamente para o padrão AV1 foi possível obter um resultado satisfatório, próximo ao apresentado pelo artigo do STDF com vídeos comprimidos pelo padrão HEVC. Ainda é preciso investigar o motivo da rede não ter se comportado da mesma forma que os vídeos em HEVC com vídeos em AV1.

Como trabalho futuro será feita uma avaliação da complexidade do modelo STDF, analisar a viabilidade de redução de complexidade e sua aplicabilidade em tempo-real. Também serão investigadas outras soluções que independem de padrões/formatos de compressão para a melhoria da qualidade de um vídeo.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

DENG, J.; WANG, L.; PU, S.; ZHUO, C. Spatio-Temporal Deformable Convolution for Compressed Video Quality Enhancement. In: **Conference on Artificial Intelligence**, 7., Nova York, 2020. Proceedings of the AAAI Conference on Artificial Intelligence, Nova York: AAAI, 2020. v.34. p.10696-10703

CISCO. **Cisco Annual Internet Report (2018–2023) White Paper**. Março, 2020. Acessado em 17 ago. 2022. Online. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>

STATISTA. **Semiconductor market size worldwide from 1987 to 2020**. Abril, 2020. Acessado em 17 ago. 2022. Online. Disponível em: <https://www.statista.com/statistics/266973/global-semiconductor-sales-since-1988/>

XING, Q.; DENG, J. **PyTorch implementation of STDF**. v.1.0.0. set. 2020. Acessado em 18 mar. 2022. Online. Disponível em: <https://github.com/ryanxingql/stdf-pytorch>