



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Explorando a Eficiência de Compressão e Qualidade de Vídeo: Um Estudo sobre Codecs de Vídeo Baseados em Deep Learning

Dissertação de Mestrado

Mairon Felipe Barreto Nascimento



São Cristóvão – Sergipe

2025

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Mairon Felipe Barreto Nascimento

**Explorando a Eficiência de Compressão e Qualidade de Vídeo:
Um Estudo sobre Codecs de Vídeo Baseados em Deep Learning**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Admilson de Ribamar Lima Ribeiro

São Cristóvão – Sergipe

2025

Agradecimentos

Primeiramente a Deus que permitiu que tudo isso acontecesse ao longo de minha vida, que me deu saúde e forças para superar todos os obstáculos nessa caminhada.

Aos meus pais José Renivaldo e Deuza Maria Nascimento, por sempre fomentar meus estudos.

A minha esposa Larissa Rozendo por estar comigo nas horas mais árduas e me ajudar sempre que preciso.

A minha irmã Marina Nascimento por frequentemente está me ajudando nas pequenas coisas.

A Denis Assaí por ser minha influencia na área da Tecnologia da Informação.

A meu Primo Pedro Nascimento e meu amigo Emanuel Melo por me fornecerem um refúgio na capital de Sergipe.

Aos meus amigos: as irmãs Maiara e Maiane Castro, Jeferson Andrade, Mariano Batista, Waldson Cardoso e Wesley Gomes, que desde o início, me apoiaram sem medir esforços.

Agradeço ao meu orientador Admilson Ribeiro, por ter desempenhado tal função com dedicação e amizade.

Enfim, a todos os funcionários do DCOMP/UFS especialmente a secretária Elaine por terem contribuído nesta conquista pessoal e profissional.

Resumo

O avanço na tecnologia de streaming de vídeo tem revolucionado a maneira como o conteúdo multimídia é consumido, tornando-se um componente essencial na distribuição de informações e entretenimento. A eficiência de transmissão e a qualidade de vídeo são aspectos críticos, especialmente em um cenário onde a demanda por conteúdos de alta resolução está em constante crescimento. Para atender a essas necessidades, codificadores/decodificadores (Codecs) de vídeo e técnicas de compressão eficientes são indispensáveis. Tradicionalmente, Codecs como H.264 e H.265 têm desempenhado um papel fundamental na compressão de vídeo, mas o aprendizado profundo trouxe novas possibilidades e desafios para este âmbito. Este trabalho tem como objetivo investigar e comparar soluções inovadoras que possam ser aplicadas para melhorar a eficiência de compressão e a qualidade de vídeo em cenários de streaming em tempo real, utilizando o potencial das técnicas de aprendizado profundo. O estudo inclui a análise de publicações científicas para identificar as contribuições propostas e suas limitações no âmbito do paradigma Codec baseado em *Deep Learning*, em relação a abordagens tradicionais para otimização de CODECs com o objetivo de reduzir a quantidade de banda utilizada.

Palavras-chave: Codec. Deep Learning. Compressão de Vídeo. Streaming.

Abstract

Advancements in video streaming technology have revolutionized the way content Multimedia is consumed, becoming an essential component in the distribution of information and entertainment. Transmission efficiency and video quality are only critical aspects, especially in a scenario where the demand for high-resolution content is constant growth To meet these needs, video encoders/decoders (Codecs) and Efficient compression techniques are essential. Traditionally, Codecs like H.264 and H.265 played a key role in video compression, but it was also learned profound brought new possibilities and challenges to this field. This work aimed to investigate and compare innovative solutions that can be applied to improve efficiency compression and video quality in real-time streaming scenarios using or potential of deep learning techniques. The study included analysis of publications identify the proposed contributions and their limitations within the paradigm Codec based on Deep Learning, in relation to traditional approaches to optimization CODECs with the aim of reducing the amount of bandwidth used.

Keywords: Codec. Deep Learning. Video Compression. Streaming.

Lista de ilustrações

Figura 1 – Plataformas de <i>Streaming de vídeo</i> .	19
Figura 2 – Codec	21
Figura 3 – Princípio da codificação escalável	23
Figura 4 – Cálculo do índice PSNR.	37
Figura 5 – Cálculo do índice MSSIM entre as imagens X e Y.	38
Figura 6 – Imagem do software PSNR 1.2	39
Figura 7 – Imagem do software VMAF-GUI	39
Figura 8 – Captura de tela do Vídeo Utilizado para Referência	41
Figura 9 – Tamanho das imagens	41
Figura 10 – Reação entre Bit Rate e PSNR	43
Figura 11 – Comparação de Qualidade de Vídeo (VMAF)	45
Figura 12 – Processo de Encoder e Decoder do AIVC	46
Figura 13 – Comparativo Bit Rate e MS-SSIM	47

Lista de tabelas

Tabela 1 – Palavras-chave e seus sinônimos em inglês	27
Tabela 2 – <i>String</i> utilizada para realizar as buscas nas bases	27
Tabela 3 – Hardware	36
Tabela 4 – Comparação de codecs, resoluções, bit rates, Tamannhos e PSNR	43
Tabela 5 – Porcentagem do bitrate, tamanho e PSNR em relação à referência (VP09 Original 4K para bitrate/tamanho e AV1 720p para PSNR)	43
Tabela 6 – Comparação de codecs, resoluções e VMAF	45

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
AV1	Codificador de vídeo AV1
AVC	Advanced Video Coding
CE	Critério de Exclusão
CI	Critério de Inclusão
CBR	Taxa de bits constante
CBR	Taxa de bits constante
CNN	Rede neural Convolucional
CODEC	Codificador e Decodificador
DCNN	Deep Convolutional Neural Network
DCOMP	Departamento de Computação
DLSS	Deep Learning Super Sampling
DPCM	Differential pulse-code modulation
GAN	Rede Adversária generativa
HEVC	High Efficiency Video Coding
H264	Outro nome dado para o AVC
H265	Outro nome dado para o HEVC
H.266	Outro nome dado para o VVC
MSI	Mapeamento sistemático da literatura
MS-SSIM	Multiscale Structural Similarity Index Measure
PSNR	Relação sinal-ruído de pico
QOE	Qualidade de Experiência
QPP	Questão de Pesquisa Primária
QPS	Questão de Pesquisa Secundária

RSI	Revisão Sistemática da literatura
UFS	Universidade Federal de Sergipe
VBR	Taxa de bits variável
VP9	Codificador de vídeo VP9
VVC	Versatile Video Coding
VMAF	Video Multimethod Assessment Fusion

Sumário

1	Introdução	11
1.1	Apresentação Geral	11
1.2	Motivação	12
1.3	Justificativa	13
1.4	Problemática e Hipótese	14
1.5	Objetivo geral	14
1.5.1	Objetivos Específicos	14
1.6	Estrutura do Documento	15
2	Fundamentação Teórica	16
2.1	História e Evolução dos CODECs de vídeo	16
2.2	Compressão de Vídeo	17
2.3	Compressão com perdas e sem perdas	17
2.4	Streaming de Vídeo	18
2.5	Deep Learning	19
2.6	CODECs	20
2.6.1	Tipos de CODECs	22
2.6.2	Principais Codecs	24
3	Estado da Arte	25
3.1	Mapeamento Sistemático	26
3.1.1	Questões de Pesquisa	26
3.1.2	Estratégia de busca	26
3.1.3	Resultados	28
3.2	Revisão Sistemática	29
3.3	Trabalhos de Pesquisas Relacionados	30
3.4	Análise dos Resultados do Mapeamento Sistemático	33
4	Materiais e Métodos	34
4.1	CODECs de Vídeo	34
4.2	Hardware	36
4.3	Metodologia	36
4.3.1	Métricas	36
4.3.2	Execução	38
5	Resultados	40

5.1	Convertendo os Arquivos de Vídeo	40
5.2	Obtendo o PSNR	41
5.3	Obtendo o VMAF	44
5.4	AIVC: Artificial Intelligence-base Video Coding	46
5.5	Análise dos Resultados	48
6	Considerações Finais	50
6.1	Limitações do Trabalho	51
6.2	Trabalhos Futuros	51
	Referências	53
	Anexos	56
	ANEXO A Algoritmo VMAF em Python	57

1

Introdução

Este capítulo apresenta de forma contextualizada uma abordagem sobre *Streaming de Vídeo*, CODEC e compressão de Vídeo. Em sua estrutura consta também, a motivação da pesquisa, objetivos, a justificativa e a organização do documento.

1.1 Apresentação Geral

De acordo com (RD; UDUPA, 2020a), a invenção da câmera ou de tecnologias de imagem forneceu uma nova maneira de registrar, armazenar e transmitir informações, sendo que uma imagem vale mais do que mil palavras. As tecnologias de imagem produzem diferentes tipos de imagens, como imagens de cena, imagens de documentos, esboços, imagens médicas, entre outras. Além disso, a invenção das imagens em movimento, como os vídeos, proporcionou utilidades ainda mais significativas, especialmente em áreas como aulas on-line baseadas em vídeo, videoconferências, bate-papos por vídeo e operações remotas guiadas por robótica.

Nos últimos anos, o *streaming de vídeo* tornou-se uma das formas mais populares de consumo de conteúdo online. A capacidade de transmitir vídeos de alta qualidade pela Internet, em tempo real, revolucionou a maneira como as pessoas assistem a filmes, programas de TV, transmissões esportivas e outras formas de mídia audiovisual. No cerne dessa revolução tecnológica está o CODEC, uma tecnologia essencial para a compressão e a transmissão eficiente de dados de vídeo.

Codec é uma abreviação de *coder/decoder* (codificador/decodificador) e refere-se a um conjunto de algoritmos e padrões que comprimem o conteúdo de vídeo para transmissão e armazenamento e, posteriormente, o decodificam para exibição aos espectadores. Segundo (RD; UDUPA, 2020a) A função de um CODEC é reduzir o tamanho dos arquivos de vídeo sem comprometer significativamente a qualidade da imagem. Dessa forma, é possível transmitir vídeos pela Internet sem exigir uma largura de banda excessiva.

Existem vários *Codecs* disponíveis no mercado, cada um com suas próprias características e recursos. Alguns dos *Codecs* mais populares são o H.264 (também conhecido como AVC-Advanced Video Coding), o H.265 (HEVC), o VP9 e o AV1. Esses são amplamente utilizados por serviços de *streaming de vídeo*, como Netflix, YouTube e Amazon Prime Video, para fornecer uma experiência de visualização de alta qualidade, mesmo em conexões de Internet mais lentas.

Além da compressão eficiente, os *CODECs* também podem oferecer recursos avançados, como suporte para vídeos em resoluções ultra HD (4K e 8K), alcance dinâmico expandido (HDR) e taxas de quadros mais altas. Esses recursos contribuem para uma experiência de visualização mais imersiva e de alta fidelidade.

O estudo de ([ARRIVUKANNAMMA; SATHIASEELAN, 2015](#)) revelou que a eficiência de um Codec pode ser avaliada não apenas por sua capacidade de compressão, mas também por sua capacidade de preservar a qualidade visual. A eficiência de compressão refere-se à capacidade de um codec de reduzir o tamanho do arquivo de um vídeo ou imagem sem comprometer excessivamente sua qualidade visual. Isso é medido pela taxa de compressão, ou seja, a relação entre o tamanho original do arquivo e o tamanho do arquivo comprimido. Quanto maior a eficiência de compressão, menor será o arquivo resultante para uma determinada qualidade visual. Ela é fundamental para transmissões e armazenamento de mídia, pois um codec eficiente pode minimizar o consumo de largura de banda e espaço de armazenamento, mantendo uma qualidade perceptível adequada para o usuário. Ao considerar a escolha de um CODEC, é essencial ponderar o equilíbrio entre taxa de bits, eficiência de compressão e qualidade perceptual, a fim de garantir uma experiência de visualização satisfatória para os usuários.

Em resumo, o *streaming de vídeo* e os *Codecs* desempenham papéis fundamentais na transmissão eficiente e na entrega de conteúdo audiovisual pela Internet. Com o constante avanço da tecnologia, podemos esperar que novos *Codecs* sejam desenvolvidos no futuro, proporcionando uma qualidade ainda melhor e maior eficiência na transmissão de vídeo online.

1.2 Motivação

Um problema fundamental com a codificação de vídeo é o fato de que a redução drástica na taxa de dados é baseada em métodos de codificação com perdas, ou seja, conforme a taxa de compressão aumenta, a qualidade da imagem se deteriora e leva a artefatos e aberrações visíveis.

Devido aos métodos de codificação aplicados, isso significa que atingir uma taxa de bits constante (CBR) resulta em qualidade de vídeo variável e, ao buscar uma qualidade de vídeo constante, a taxa de bits resultante não pode ser mantida constante (taxa de bits variável, VBR). Normalmente, o CBR é preferido para canais de transmissão com uma capacidade de canal específica, enquanto o VBR é frequentemente usado em casos em que é necessária uma qualidade de vídeo consistente, por exemplo, Blu-ray ou outra mídia de armazenamento.

Usando a codificação de vídeo em cenários de aplicativos clássicos, como streaming de vídeo, a qualidade da codificação será julgada pelo visualizador que, quando esta em uma rede limitada por exemplo, está disposto a aceitar artefatos devido a baixa qualidade de imagem em favor do uso de canais de transmissão de baixa taxa de bits.

Em contraste com esses cenários, as aplicações técnicas normalmente aplicam algoritmos de processamento de imagem nos fluxos de vídeo decodificados. Nesse caso, a qualidade do vídeo resultante deve ser mantida constante para obter os melhores resultados. Uma medida frequentemente usada para indicar a qualidade da codificação de vídeo é o PSNR (relação sinal-ruído de pico), em que um valor mais alto significa melhor qualidade de vídeo. Com o advento do amplo uso de DCNN (*deep convolutional neural network*) como algoritmos de processamento de imagem para várias aplicações, experimento no Capítulo 4 mostra que quanto maior o PSNR, maior é a confiança resultante da classificação, que obviamente se deteriora ao usar erro -canais de baixa taxa de bits propensos.([STABERNACK; STEINERT, 2021](#))

1.3 Justificativa

A transmissão de vídeo pela Internet enfrenta desafios, como a largura de banda limitada e as flutuações na qualidade da rede. A pesquisa em CODECs de vídeo busca soluções para otimizar a transmissão de vídeo em diferentes velocidades de conexão e condições de rede variáveis, garantindo uma experiência de visualização suave e contínua para os espectadores.

O estudo de ([MIAO et al., 2013](#)) destaca a importância da pesquisa de Codecs, afirmando que o uso de codificação de vídeo em camadas – uma técnica onde o vídeo é codificado em diferentes níveis ou camadas de qualidade e resolução – aproveitando Codecs de *hardware*, tem se mostrado promissor para melhorar tanto a eficiência quanto à qualidade do vídeo em telas multicamadas. A colocação em camadas permite adaptar a transmissão ou exibição de vídeo conforme a capacidade do dispositivo ou da rede, utilizando uma camada básica de qualidade menor e camadas adicionais que aumentam a qualidade visual conforme os recursos disponíveis. Esse método é especialmente eficiente em ambientes com múltiplas camadas de conteúdo, pois otimiza a otimização e reduz a necessidade de banda, ao mesmo tempo que preserva a qualidade visual ambientes com múltiplas camadas de conteúdo.

Os Codecs de vídeo tradicionais têm alcançado níveis altos de eficiência de compressão, mas ainda existem oportunidades para melhorias. Com o uso de técnicas de *Deep Learning*, é possível explorar padrões complexos e não-lineares nos dados de vídeo, permitindo uma compressão mais eficiente e, conseqüentemente, a redução do tamanho dos arquivos de vídeo sem comprometer significativamente a qualidade.

1.4 Problemática e Hipótese

Compressão de vídeo eficiente é crucial para a transmissão em tempo real, mas muitas vezes compromete a qualidade do vídeo, especialmente em condições de largura de banda variável e recursos de computação limitados. Ao mesmo tempo, melhorar a qualidade do vídeo utilizando técnicas de *deep learning* apresenta desafios em termos de latência e capacidade de processamento.

Para nortear o desenvolvimento desta pesquisa formulou-se a seguinte problemática: Como os Codecs de vídeo baseados em *deep learning* podem otimizar a eficiência de compressão e manter a qualidade do vídeo em aplicações de streaming em tempo real, e quais tecnologias de melhoria de imagem utilizando *deep learning* podem ser integradas para superar os desafios de latência, variação de largura de banda e recursos de computação limitados?

Como hipótese temos que a utilização de Codecs de vídeo baseados em *Deep Learning* pode otimizar a eficiência de compressão e manter a qualidade do vídeo, mesmo em aplicações de *streaming* em tempo real. Além disso, tecnologias de melhoria de imagem baseadas em *deep learning* podem ser integradas a esses Codecs para superar as limitações tradicionais e fornecer uma experiência de vídeo superior, adaptando-se dinamicamente às condições da rede e aos recursos do dispositivo.

Com essa hipótese, o trabalho busca investigar e comparar soluções inovadoras que possam ser aplicadas para melhorar a eficiência de compressão e a qualidade de vídeo em cenários de *streaming* em tempo real, utilizando o potencial das técnicas de *Deep Learning*.

1.5 Objetivo geral

Esta pesquisa tem como objetivo analisar publicações científicas com o propósito de identificar quais contribuições foram propostas e suas limitações no âmbito do paradigma Codec baseado em Deep Learning e aprimorar CODECs de compressão de dados, visando a otimização da eficiência de compressão e qualidade de transmissão de vídeo em diversas aplicações tecnológicas. Além disso, será realizado um experimento para medir a qualidade do vídeo, permitindo uma avaliação objetiva do desempenho dos CODECs estudados. Espera-se que essa abordagem proporcione uma experiência de usuário aprimorada, reduza a demanda por largura de banda e melhore a eficiência de armazenamento, contribuindo assim para o avanço das tecnologias de multimídia e comunicação.

1.5.1 Objetivos Específicos

Para alcançar o objetivo principal, foram definidos os seguintes objetivos específicos:

- Realizar um levantamento sistemático e uma análise dos trabalhos relacionados a codecs

de vídeo baseados em Deep Learning, identificando tendências, desafios e oportunidades de pesquisa na área;

- Investigar o desempenho de CODECs de vídeo em diferentes cenários de aplicação, como streaming ao vivo, transmissão de vídeos em dispositivos móveis e armazenamento em nuvem;
- Aprimorar técnicas de otimização e melhorias em CODECs de vídeo existentes, visando reduzir ainda mais a taxa de bits e melhorar a qualidade visual sem aumentar o tamanho do arquivo;

1.6 Estrutura do Documento

Com o objetivo de garantir clareza, coesão e uma navegação facilitada, este trabalho foi organizado em capítulos e seções, conforme descrito a seguir:

- Capítulo 1 - Introdução: apresenta conceitos fundamentais extraídos da literatura, a problemática central, as principais argumentações e hipóteses relacionadas ao tema, além dos objetivos do estudo;
- Capítulo 2 - Fundamentação teórica: contextualiza o tema a partir de uma revisão da literatura pertinente, fornecendo a base conceitual para a pesquisa;
- Capítulo 3 - Estado da arte: aborda estudos correlatos, por meio de um mapeamento sistemático e de uma revisão de literatura estruturada, destacando os principais resultados obtidos e a síntese do processo de análise dos trabalhos investigados;
- Capítulo 4 - Materiais e Métodos: detalha os métodos, materiais e procedimentos empregados no desenvolvimento da pesquisa;
- Capítulo 5 - Resultados: apresenta uma análise sobre os resultados obtidos;
- Capítulo 6 - Considerações finais: discute as conclusões alcançadas com a pesquisa, bem como sugestões para trabalhos futuros.

2

Fundamentação Teórica

Neste capítulo é apresentada, a Fundamentação Teórica cuja abordagem abrange: Streaming de Vídeo; Compressão com perdas e sem perdas; Deep learning e Compressão de Vídeo; Codecs; e História e Evolução dos CODECs de vídeo.

2.1 História e Evolução dos CODECs de vídeo

A evolução da tecnologia de compressão de vídeo tem sido caracterizada por um esforço contínuo para melhorar a eficiência da codificação, mantendo ou melhorando a qualidade do sinal de vídeo. Desde os primeiros dias da modulação diferencial simples por código de pulso (DPCM) até o altamente sofisticado H.264/AVC, cada geração de CODECs introduziu técnicas inovadoras para reduzir a taxa de bits necessária para transmissão e armazenamento de vídeo de alta qualidade (SULLIVAN; WIEGAND, 2005).

H.120 foi o primeiro padrão de codificação de vídeo na década de 80 e H.261 e H.263 foram introduzidos na década de 90. Os primeiros padrões ajudaram a desenvolver técnicas avançadas que foram posteriormente adotadas em CODECs (RICHARDSON, 2002).

Ozer (2011) explica a correlação entre a evolução do CODEC de vídeo e a necessidade de uma compactação mais eficiente para atender à crescente demanda por maior qualidade de vídeo e taxas de bits mais baixas. Ele diz que os CODECs atuais são projetados para oferecer melhor compactação, resultando em uma redução acentuada na quantidade de dados necessários para representar vídeo de alta qualidade.

Os CODECs de vídeo evoluíram ao longo do tempo e foram caracterizados por uma série de avanços que visam melhorar a qualidade e a compactação do vídeo. Desde os padrões iniciais, como H.120 e H.261, até CODECs mais avançados, como MPEG-2 e H.264 /AVC, cada geração introduziu novas técnicas em compensação de movimento, codificação de transformação e codificação de entropia, ampliando continuamente os limites da tecnologia de compressão de

vídeo (JACK, 2004).

Codecs mais recentes têm algoritmos de compressão melhores, como codificação de entropia e previsão de movimento, como apontou (OZER, 2011). Ele afirma que essas melhorias não só aumentam a eficiência da compressão, mas também ajudam a preservar ou melhorar a qualidade visual do vídeo.

Para (SCHMIDT et al., 2019) Os padrões de codificação de vídeo evoluíram significativamente nas últimas décadas, desde os primeiros H.261 e MPEG-1 até o mais recente H.264 /AVC e H.265/HEVC Cada novo padrão tem como objetivo fornecer melhor eficiência de compressão e qualidade de vídeo.

Cochrane e Shkiperov (2020) discutem as mudanças ao longo do tempo em relação aos CODECs, para eles a evolução dos CODECs de vídeo é caracterizada por uma progressão constante em direção a sistemas mais eficientes e flexíveis, refletindo a crescente demanda por aplicações de vídeo de alta definição e em tempo real.

2.2 Compressão de Vídeo

De acordo com Le et al. (2022), a compressão de vídeo é um processo de redução do tamanho do arquivo de vídeo original, que contém uma sequência de imagens (quadros ou frames) em movimento, sem comprometer muito a qualidade visual do vídeo. O objetivo da compressão de vídeo é reduzir a quantidade de dados necessários para armazenar e transmitir o vídeo, tornando-o mais fácil de ser gerenciado e transmitido através de redes com limitações de largura de banda.

A compressão de vídeo é essencial para a transmissão eficiente de vídeos pela Internet ou por outras redes, especialmente em dispositivos móveis, onde a largura de banda pode ser limitada. Através desse processo, os vídeos podem ser enviados e recebidos mais rapidamente, economizando espaço de armazenamento e recursos de transmissão.

2.3 Compressão com perdas e sem perdas

A compressão de vídeo faz parte de uma categoria mais ampla de tecnologias de compressão de dados, que pode ser classificada em compressão sem perdas e com perdas. Na compressão sem perdas, o sinal reconstruído é bit a bit idêntico ao original após a compressão/descompressão, enquanto na compressão com perdas, há degradação do sinal, conhecido como distorção (SULLIVAN; WIEGAND, 2005).

Para (RICHARDSON, 2002) as técnicas de compressão são sem perdas ou com perdas. Algoritmos de compactação sem perdas reduzem o tamanho dos dados sem comprometer os dados originais, resultando em uma reconstrução perfeita dos dados originais. Técnicas de

compressão com perdas sacrificam algumas informações para obter taxas de compactação mais altas, introduzindo distorção perceptiva em troca de tamanhos de arquivo reduzidos.

Na compressão sem perdas, geralmente se consegue reduzir o tamanho dos dados em cerca de duas a três vezes. Por outro lado, na compressão com perdas, a taxa de variação alcançada depende do nível de margem aceitável.

2.4 Streaming de Vídeo

O *streaming de vídeo* é uma tecnologia que permite transmitir e reproduzir conteúdo de vídeo em tempo real através da Internet. Em vez de ser necessário baixar um arquivo de vídeo inteiro para assistir, o *streaming de vídeo* permite que o conteúdo seja reproduzido instantaneamente enquanto é transmitido.

O funcionamento básico do *streaming de vídeo* envolve a divisão do arquivo de vídeo em pequenos pacotes chamados de "chunks". Esses chunks são enviados ao dispositivo do usuário em tempo real, e o player de vídeo começa a reproduzir o conteúdo assim que recebe os primeiros pacotes, mesmo antes de ter baixado todo o arquivo.

De acordo com [Ghosh, Singhal e Wayal \(2022\)](#) o termo "*streaming de vídeo*" refere-se à transmissão contínua de dados de vídeo pela Internet, permitindo que os usuários assistam a conteúdo em tempo real sem a necessidade de baixar o arquivo de vídeo completo antes da reprodução. O autor destaca que o objetivo do *streaming de vídeo* é fornecer uma experiência de visualização suave e contínua, adaptando-se às condições de rede e às capacidades do dispositivo do usuário.

[Lin et al. \(2022\)](#) destaca que o *streaming de vídeo* refere-se à transmissão contínua de conteúdo de vídeo pela Internet ou outra rede de comunicação. Nesse contexto, o *streaming de vídeo* envolve o envio de dados de vídeo em tempo real, em vez de baixar o arquivo completo antes de reproduzi-lo. Portanto, o *streaming de vídeo* é uma tecnologia essencial para a entrega de conteúdo de vídeo em tempo real, como transmissões ao vivo, vídeos sob demanda e outros serviços de *streaming de vídeo*, tornando possível assistir a vídeos sem a necessidade de baixá-los completamente antes da reprodução.

A Figura 1 apresenta uma coleção das principais plataformas de *streaming de vídeo*, ilustrando a diversidade e a popularidade desses serviços na atualidade. Cada logotipo representa uma plataforma distinta, refletindo a ampla gama de opções disponíveis para os consumidores em termos de conteúdo e experiência de usuário.

Figura 1 – Plataformas de *Streaming de vídeo*.

Fonte: Próprio Autor

2.5 Deep Learning

Para [Tekalp et al. \(2021\)](#) o Deep Learning refere-se a uma abordagem de aprendizado de máquina que utiliza redes neurais artificiais com várias camadas para aprender representações complexas dos dados. Nesse contexto específico, o Deep Learning é aplicado para problemas de restauração e compressão de imagem e vídeo.

No Deep Learning, as redes neurais são projetadas com várias camadas de unidades de processamento, onde cada camada aprende a representação dos dados em um nível de abstração diferente. Essas redes podem ser treinadas em grandes conjuntos de dados para aprender a mapear entradas por exemplo, imagens degradadas ou vídeos comprimidos para saídas desejada, imagens restauradas ou vídeos comprimidos com alta qualidade.

O Deep Learning tem se destacado em diversas tarefas relacionadas a imagem e vídeo, como restauração de imagens borradas ou danificadas, redução de ruído, remoção de artefatos visuais em vídeos, e também em melhorias na eficiência de compressão, proporcionando codificações de vídeo com maior qualidade e taxas de bits mais baixas.

Assim, o Deep Learning é uma abordagem poderosa e em rápido desenvolvimento na área de processamento de imagem e vídeo, permitindo soluções avançadas e eficazes para problemas de restauração e compressão.

De acordo com [Le et al. \(2022\)](#), a compressão de vídeo é um processo de redução do tamanho do arquivo de vídeo original, que contém uma sequência de imagens (quadros ou

frames) em movimento, sem comprometer muito a qualidade visual do vídeo. O objetivo da compressão de vídeo é reduzir a quantidade de dados necessários para armazenar e transmitir o vídeo, tornando-o mais fácil de ser gerenciado e transmitido através de redes com limitações de largura de banda.

A compressão de vídeo é essencial para a transmissão eficiente de vídeos pela Internet ou por outras redes, especialmente em dispositivos móveis, onde a largura de banda pode ser limitada. Através desse processo, os vídeos podem ser enviados e recebidos mais rapidamente, economizando espaço de armazenamento e recursos de transmissão.

2.6 CODECs

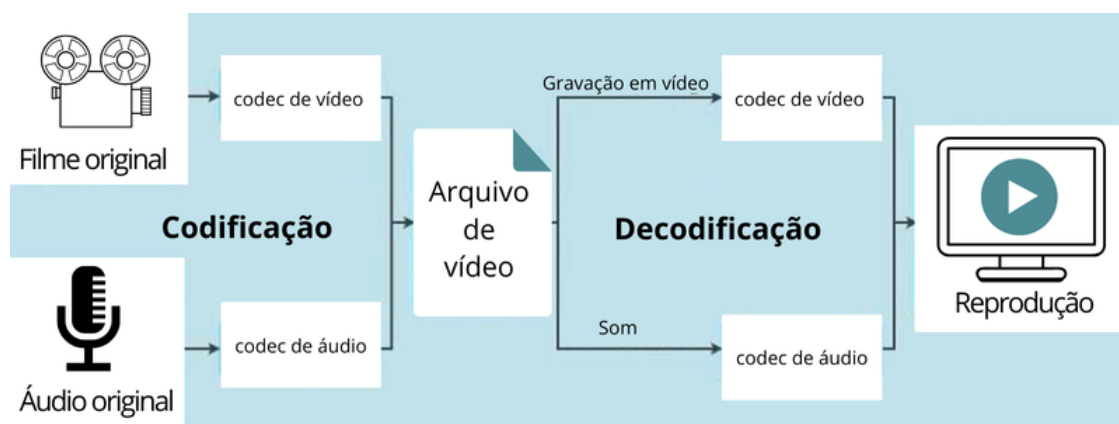
O termo CODEC refere-se à combinação das palavras codificação e decodificação. de acordo com ([RAMACHANDRAN et al., 2020](#)), um CODEC é um conjunto de algoritmos e técnicas utilizados para comprimir e descomprimir dados de áudio e vídeo. Esses algoritmos são responsáveis por reduzir o tamanho dos arquivos de mídia, sem comprometer significativamente a qualidade perceptível pelo usuário final. Além disso, o uso de CODECs permite a transmissão eficiente de conteúdo multimídia em diferentes canais de comunicação, como a Internet, por meio de técnicas de codificação adaptativa que se ajustam às condições de largura de banda disponíveis. A adoção de CODECs, conforme mencionado no artigo, oferece a vantagem de adaptar a codificação de vídeo em tempo real de acordo com as características do conteúdo transmitido, garantindo uma experiência de visualização aprimorada para os usuários, independentemente do dispositivo ou do meio de transmissão utilizado.

De acordo com ([ARRIVUKANNAMMA; SATHIASEELAN, 2015](#)), um CODEC pode ser definido como um conjunto de algoritmos e técnicas utilizados para comprimir e descomprimir dados de vídeo. O autor destaca a importância das métricas de qualidade do CODEC na avaliação da eficiência de compressão e da fidelidade do vídeo resultante, como por exemplo o Peak Signal-to-Noise Ratio (PSNR).

O estudo mencionado por ([ARRIVUKANNAMMA; SATHIASEELAN, 2015](#)) se concentra nas métricas de qualidade do CODEC, que são medidas objetivas ou subjetivas usadas para avaliar a qualidade visual do vídeo após a compressão e a descompressão. Essas métricas podem incluir análise de distorção, preservação de detalhes, taxa de bits, artefatos de compressão, entre outros fatores que afetam a qualidade percebida do vídeo.

A Figura 2 ilustra o processo de codificação e decodificação de vídeo e áudio. No lado esquerdo, temos os dados originais de vídeo e áudio sendo capturados. Esses dados passam por CODECs de vídeo e áudio durante o processo de codificação, resultando em um arquivo de vídeo combinado. No lado direito, o arquivo de vídeo é decodificado usando os CODECs correspondentes para reproduzir o conteúdo final.

Figura 2 – Codec



Fonte: Próprio Autor

2.6.1 Tipos de CODECs

Há dois tipos de CODECs disponíveis no mercado são eles os escaláveis e não escaláveis. Um codificador não escalável é um tipo de CODEC que não permite ajustar a taxa de bits em um mesmo arquivo, o que limita a flexibilidade para lidar com diferentes resoluções. Por exemplo, se um CODEC não escalável for projetado para a resolução 1920x1080, ao tentar transmitir uma resolução menor, a qualidade da reprodução será significativamente afetada ou ocorrerá um erro de compatibilidade.

Anteriormente, o uso de tais codificadores não escaláveis era bastante comum, principalmente devido às limitações de programação e licenciamento. No entanto, atualmente é muito mais comum utilizar CODECs que possuem escalabilidade intrínseca, permitindo a adaptação para diversas resoluções e taxas de *bits*.

Os codificadores não escaláveis também são conhecidos como codificadores de camada única, pois não permitem a criação de vários fluxos de bits com diferentes resoluções e qualidades. Essa limitação pode ser problemática quando há a necessidade de oferecer suporte a dispositivos com capacidades distintas de reprodução de vídeo, no *streaming* por exemplo.

Os esquemas escaláveis de codificação de vídeo são projetados para codificar o sinal apenas uma vez na resolução mais alta possível. No entanto, eles permitem a decodificação de fluxos parciais, de acordo com as taxas e resoluções específicas exigidas por um determinado aplicativo. Essa abordagem proporciona uma solução simples e flexível para a transmissão em redes heterogêneas, ao mesmo tempo que oferece adaptabilidade para variações de largura de banda e condições de erro.

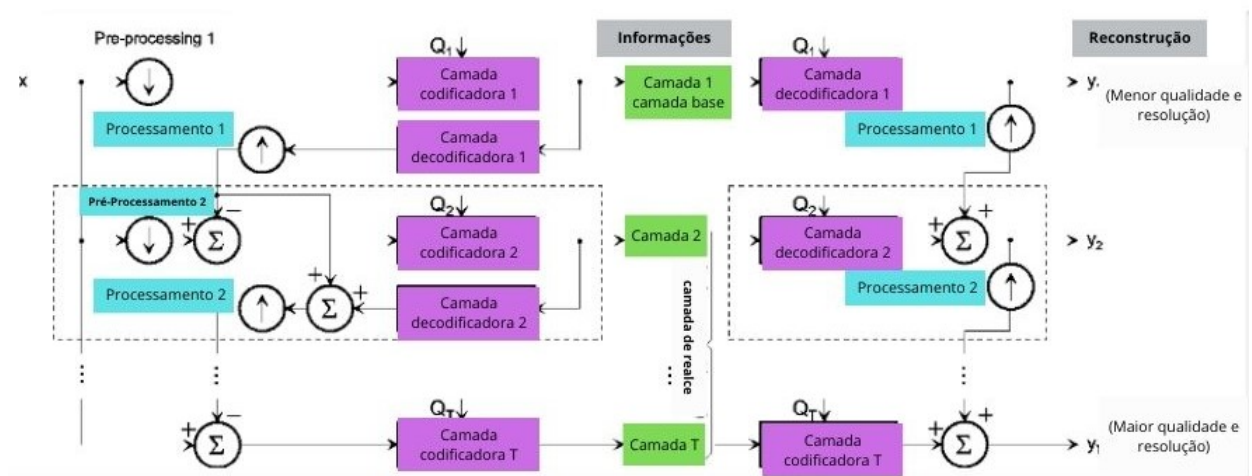
Um codificador escalável é um CODEC que possibilita que um vídeo de alta qualidade tenha um ou mais subconjuntos de fluxos de bits. Cada subconjunto pode representar uma resolução espacial mais baixa, resolução temporal reduzida ou uma qualidade de vídeo inferior (individualmente ou em combinação) em comparação com o fluxo de bits do qual são derivados. Resolução espacial mais baixa significa menos detalhes em uma imagem, pois cada pixel cobre uma área maior, tornando-a mais borrada. Já a resolução temporal reduzida indica que os dados são coletados com menor frequência ao longo do tempo, dificultando o acompanhamento de mudanças rápidas. (RICHARDSON, 2004).

Essa abordagem de codificação escalável é particularmente vantajosa em cenários em que a largura de banda pode variar, como em redes de Internet ou sistemas de transmissão sem fio. A capacidade de adaptar a taxa de bits e a resolução permite que o vídeo seja reproduzido de forma otimizada, garantindo uma experiência visual satisfatória, mesmo em condições adversas. Além disso, essa flexibilidade é útil para garantir que os usuários com diferentes capacidades de reprodução possam acessar o conteúdo com qualidade adequada às suas condições específicas, sem comprometer a experiência geral.

Um princípio bastante abrangente de codificação escalável (em camadas) e decodificação

é ilustrado na Figura 3. Nela observa-se que o sinal ao ser processado, pode passar por várias camadas de codificação e decodificação, possibilitando a produção de diferentes tipos de resolução e qualidade intermediária. A resolução espaço-temporal do sinal a ser representada pela camada de base é inicialmente gerada por dizimação (pré-processamento). No estágio de codificação subsequente, uma configuração adequada do quantizador resultará em um certo nível de qualidade geral da informação de base (OHM, 2005)

Figura 3 – Princípio da codificação escalável



Fonte: (OHM, 2005), Adaptado pelo Autor

2.6.2 Principais Codecs

- H.264 - conhecido como AVC (Advanced Video Coding), é um padrão de compressão de vídeo amplamente utilizado, desenvolvido em conjunto pelo Video Coding Experts Group e pelo Moving Picture Experts Group. Ele foi criado para fornecer boa qualidade de vídeo em taxas de bits mais baixas e é o CODEC mais comumente utilizado na gravação, compressão e distribuição de conteúdo de vídeo. Este CODEC é utilizado em várias plataformas, incluindo discos Blu-ray, fontes de streaming na Internet como Netflix, Vimeo, YouTube, e sistemas de cabo e satélite ([LOPEZ, 2021](#)).
- H.265 - conhecido como High Efficiency Video Coding (HEVC), oferece uma eficiência de compressão significativamente maior em comparação ao seu antecessor H.264, permitindo a transmissão de vídeos de alta qualidade com metade da taxa de bits. Esta eficiência é alcançada através de melhorias como flexibilidade de particionamento aumentada, melhores filtros de interpolação e desbloqueio, sinalização aprimorada de modos e vetores de movimento, e recursos eficientes de processamento paralelo ([GUMLET, 2023](#)).
- VP9 - um CODEC de vídeo de código aberto desenvolvido pelo Google, oferece eficiência de compressão entre 10% a 30% maior se comparado ao H.265, e é amplamente utilizado em plataformas como YouTube e Google Stadia. Ele é conhecido por seu suporte a vídeos de alta resolução e pela capacidade de economizar largura de banda sem comprometer a qualidade do vídeo ([BITMOVIN, 2023](#)).
- AV1 - desenvolvido pela Alliance for Open Media, oferece melhorias significativas na eficiência de compressão em comparação com CODECs anteriores, como VP9 e HEVC, ao mesmo tempo em que proporciona uma maior flexibilidade em termos de resolução e taxa de bits. Este CODEC é projetado para ser altamente eficiente, chegando em alguns casos a ser 30% mais eficiente se comparado com o VP9, permitindo uma transmissão de vídeo de alta qualidade com uma largura de banda reduzida, o que o torna ideal para aplicações de streaming e outras formas de distribuição de vídeo digital ([TOPIWALA; KRISHNAN; DAI, 2018](#)).

3

Estado da Arte

Nesta seção, são apresentados estudos relacionados ao tema, complementando a Revisão Sistemática da literatura. A partir dos artigos coletados na fase inicial da pesquisa, correspondente ao Mapeamento Sistemático, e posteriormente analisados na Revisão Sistemática, foram incorporados outros trabalhos relevantes. Para essa complementação, utilizou-se publicações disponíveis no Google Scholar, uma plataforma que reúne artigos de conferências e periódicos científicos.

Tradicionalmente, os CODECs de vídeo, como H.264/AVC, H.265/HEVC e VP9, utilizam técnicas baseadas em transformações, previsões e quantizações. No entanto, esses métodos enfrentam limitações em termos de eficiência e flexibilidade para lidar com a diversidade de conteúdos e resoluções atuais. Com o advento do *Deep Learning*, novas abordagens têm sido propostas, explorando redes neurais profundas para otimizar tanto a compressão quanto a qualidade dos vídeos. Estas abordagens baseadas em *Deep Learning* prometem superar as limitações dos métodos convencionais, oferecendo uma compressão mais eficiente e uma melhor preservação da qualidade visual.

Os CODECs de vídeo baseados em *Deep Learning* utilizam modelos treinados em grandes conjuntos de dados de vídeo, permitindo que a rede neural aprenda padrões complexos e redundâncias presentes nos vídeos. Assim, essas técnicas são capazes de realizar compressão de maneira mais inteligente, ajustando-se dinamicamente às características específicas de cada vídeo. Além disso, a utilização de redes neurais convolucionais (CNNs), redes adversárias generativas (GANs) e outras arquiteturas avançadas tem mostrado resultados promissores na melhoria da relação entre compressão e qualidade de vídeo.

A seção a seguir tem como objetivo identificar e analisar trabalhos que abordam temas relacionados à eficiência de compressão e qualidade de vídeo utilizando CODECs baseados em *Deep Learning*. Serão exploradas as principais metodologias, inovações e desafios enfrentados por esses estudos, proporcionando uma visão abrangente do estado da arte e das tendências

futuras nesse campo.

3.1 Mapeamento Sistemático

Diversas áreas, como a compressão e a otimização do vídeo, foram transformadas pelo avanço das tecnologias de *Deep Learning* nos últimos anos. O paradigma de CODEC baseado em *Deep Learning* tem fornecido novas maneiras de reduzir a quantidade de banda necessária para streaming. Essa é uma área que está se tornando cada vez mais importante devido à demanda por transmissões de alta qualidade e à eficiência de dados. O objetivo deste mapeamento sistemático é examinar publicações científicas recentemente publicadas com o objetivo de identificar as contribuições significativas sugeridas no campo dos CODECs baseados em *Deep Learning*. O objetivo é entender como esses CODECs são usados para melhorar a compressão de vídeo, o que significa que a quantidade de banda necessária para o streaming é reduzida. Além disso, o estudo visa identificar as limitações dos métodos atuais e fornecer uma visão crítica das áreas que ainda precisam de melhorias e desenvolvimento.

O mapeamento inclui uma revisão aprofundada de artigos acadêmicos, conferências e outras fontes científicas que abordam algoritmos, técnicas e implementações de CODECs baseados em *Deep Learning*. A análise sistemática ajudará a entender melhor as contribuições atuais, as metodologias usadas e as lacunas no conhecimento que podem ser exploradas em pesquisas futuras.

3.1.1 Questões de Pesquisa

O mapeamento tem como objetivo responder à seguinte Questão de Pesquisa Primária (QPP): Como os CODECs baseados em *Deep Learning* se comparam com os CODECs tradicionais em termos de eficiência de compressão e qualidade de vídeo? Para abordar essa questão, foram incluídas Questões de Pesquisa Secundárias (QPS). Ao final da pesquisa, buscou-se responder à maioria das questões norteadoras (primária e secundárias) e coletar os dados propostos para este mapeamento. A seguir, são apresentadas as questões de pesquisa.

1. Quais são as principais técnicas de compressão utilizadas pelos CODECs de vídeo?
2. Quais são os desafios específicos enfrentados pelos CODECs de vídeo em aplicações de streaming de vídeo em tempo real e como eles podem ser superados?

3.1.2 Estratégia de busca

A pesquisa foi realizada utilizando cinco bases de dados internacionais, a saber: ACM Digital Library, IEEE Xplore Digital Library, ScienceDirect, Scopus e ISI Web of Science.

- Scopus <<<http://www.scopus.com>>>;

- IEEE Xplore Digital Library <<<http://ieeexplore.ieee.org>>>;
- IsiWeb of Science <<<https://www.webofknowledge.com/>>>;
- Science Direct <<<http://www.sciencedirect.com>>>;
- ACM Digital Library <<<http://portal.acm.org>>>;

A tabela 1 apresenta as palavras-chave utilizadas na construção da string de busca.

Tabela 1 – Palavras-chave e seus sinônimos em inglês

Palavra-Chave	Sinônimo em Inglês
Codec de vídeo	Video Codec
Compressão de vídeo	Video Compression

Fonte: Próprio autor

Na Tabela 2 é apresentada a string utilizada para as buscas nas bases:

Tabela 2 – *String* utilizada para realizar as buscas nas bases

**(“video CODEC” OR “video compression”) AND
 (“bandwidth reduction” OR “video quality”)**

Fonte: Próprio Autor

Para filtrar os artigos relevantes foram estabelecidos Critérios de Inclusão (CI) e Exclusão (CE) indicados abaixo. Assim foram selecionados a partir somente da leitura do título e resumo de cada artigo resultando em 21 artigos oriundos de 602 artigos encontrados pela string. Após esta etapa vem o filtro por meio da leitura completa dos artigos resultantes da primeira seleção, analisando-os a partir de Questões de Qualidade.

A seguir os Critérios de Inclusão:

1. Acessível via web de maneira gratuita;
2. Artigos apenas com idioma inglês e português;
3. Somente estudos primários;
4. Artigos publicados em conferências e periódicos;
5. Estudos a publicados a partir de 2015.

A seguir os Critérios de Exclusão:

1. Estudos Duplicados;
2. Falta de disponibilidade para download de maneira gratuita;
3. Trabalhos publicados como resumos ou resumo expandido ou prefácio de periódicos e eventos;
4. Trabalhos que não contemplam a temática.

O critério de inclusão e exclusão enquanto ferramenta de busca, possibilitou o acesso a estudos de diferentes regiões sobre Codecs de vídeo baseados em *Deep Learning*. Incluiu-se neste interim, pesquisas empreendidas através da publicação de artigos em conferências ou periódicos; selecionou-se mediante tal critério apenas artigos em inglês e português, também foram utilizados somente estudos publicados a partir de 2015. No que concerne à exclusão, não houve ênfase em relação a estudos duplicados, estudos secundários e trabalhos publicados como resumos ou resumo expandido ou prefácio de periódicos e eventos. Também foram excluídas informações incompletas ou aquelas sem disponibilidade para download.

3.1.3 Resultados

Os autores (GHOSH; SINGHAL; WAYAL, 2022) realizaram uma comparação entre CODECs de vídeo baseados em Deep Learning e CODECs tradicionais, com foco na eficiência de compressão e na qualidade do vídeo. Eles descobriram que os CODECs baseados em Deep Learning têm o potencial de melhorar tanto a eficiência de compressão quanto a qualidade do vídeo, uma vez que esses sistemas podem se adaptar às características específicas do conteúdo de vídeo e às condições da rede. Além disso, o estudo enfatizou o uso de uma combinação de várias características e modelos de aprendizado otimizados para avaliar a Qualidade da Experiência (QoE) em vídeos de streaming.

Salhi et al. (2018) abordam questões importantes relacionadas à adaptação de aplicações de codec de vídeo AVC para plataformas multicore existentes. Eles propõem técnicas que visam otimizar o agendamento e a execução dessas aplicações em ambientes de múltiplos núcleos, buscando melhorar o desempenho e a eficiência no processamento de vídeos. Quanto a análise dos CODECs baseados em *Deep Learning* em comparação com os CODECs convencionais não é abordada diretamente neste artigo. O objetivo principal é otimizar as aplicações de CODEC de vídeo usando algoritmos DAG e GGEN para melhorar o desempenho e a eficiência da codificação. E sobre os problemas específicos que os CODECs de vídeo enfrentam em aplicativos de *streaming* de vídeo em tempo real: incluem equilibrar a qualidade do vídeo com a eficiência de compressão e se ajustar às variações de largura de banda.

A variação dinâmica das condições da rede é um grande problema discutido no trabalho de (RAMACHANDRAN et al., 2020), que pode afetar a qualidade do streaming de vídeo em tempo real. O artigo apresenta métodos de codificação adaptativos que ajustam a taxa de bits

e a resolução do vídeo em tempo real com base nas condições da rede. Isso reduz o *buffering* e mantém a qualidade do vídeo. O artigo enfatiza uma variedade de métodos de compressão que os CODECs de vídeo de código aberto usam. A quantização, a codificação de entropia, a transformação de blocos e a codificação preditiva estão entre essas categorias. Estas estratégias são vitais para reduzir o tamanho dos arquivos de vídeo.

3.2 Revisão Sistemática

A Revisão Sistemática seguiu um protocolo detalhado, e foi dividida em duas etapas: execução da busca e seleção. Uma Revisão Sistemática da literatura de acordo com (KITCHENHAM, 2004) é o estudo que busca a identificação, a avaliação e a interpretação de toda a pesquisa disponível e relevante frente a determinado tema ou questão de pesquisa específica.

Para a etapa de execução da busca, foi utilizada a seguinte string: ("video CODEC") OR ("video compression") AND ("deep learning "OR "Deep Learning") . Para desenvolvimento desse levantamento bibliográfico utilizou-se as seguintes bases de dados, ACM Digital Library ¹, IEEE Digital Library ² e Science Direct ³.

A string é utilizada nas fontes selecionadas e os resultados alcançados são armazenados na ferramenta online Parsifal ⁴. A coleta de dados foi realizada no dia 19 de julho de 2023, não se definiu limite temporal, foram selecionados apenas artigos publicados em conferências ou periódicos.

Na base ACM Digital Library, foram levantados 352 artigos e aceitos 5; na base IEEE Digital Library foram levantados 297 artigos e aceitos 5; e, na base Science Direct foram levantados 401 artigos e aceitos 3. Isso resultou em um total de 13 artigos aceitos.

¹ <<https://dl.acm.org/>>

² <<https://ieeexplore.ieee.org/Xplore/home.jsp>>

³ <<https://www.sciencedirect.com/>>

⁴ <<https://parsif.al/about/>>

3.3 Trabalhos de Pesquisas Relacionados

[Salhi et al. \(2018\)](#) destacam que a abordagem baseada em DAG e no algoritmo GGEN para programação e otimização de aplicações de CODECs de vídeo demonstrou ser eficaz na melhoria do desempenho do processamento de vídeo. Essa trabalho ressalta a relevância da utilização de técnicas de programação e otimização baseadas em DAG e GGEN para aprimorar o processamento de vídeo, permitindo uma alocação eficiente de recursos e uma ordem de execução otimizada das tarefas de codificação e decodificação. Por meio dessa abordagem, é possível alcançar uma melhor eficiência de processamento, menor latência e maior qualidade de vídeo em diferentes aplicações de CODECs.

Para [Kwan, Shang e Tran \(2018\)](#) a compressão de vídeo perceptualmente sem perdas com ocultação de erros é uma abordagem eficaz para preservar a qualidade visual de um vídeo, mesmo em situações de perda de dados. Essa citação enfatiza a importância da compressão perceptualmente sem perdas, que busca manter a qualidade visual percebida do vídeo, mesmo quando ocorrem erros de transmissão ou armazenamento. Além disso, a ocultação de erros é mencionada como uma técnica adicional para minimizar os efeitos visuais das perdas de dados. Essa abordagem contribui para melhorar a experiência do usuário ao garantir uma qualidade de vídeo preservada mesmo em condições adversas.

De acordo com [Miao et al. \(2013\)](#) o uso de codificação de vídeo em camadas, aproveitando o CODEC de *hardware*, tem se mostrado uma abordagem promissora para melhorar a eficiência e a qualidade de vídeo em telas *multilayer*. A combinação dessas tecnologias permite uma distribuição eficiente dos recursos de codificação, possibilitando uma experiência visual aprimorada em ambientes com restrições de largura de banda e capacidade de processamento limitada.

No estudo de ([KAZANTSEV; ZVEZDAKOV; VATOLIN, 2020](#)) um método baseado em aprendizado de máquina para encontrar configurações ótimas de CODEC de vídeo usando características físicas do vídeo de entrada revelou-se uma abordagem eficaz para melhorar a qualidade e a eficiência da compressão de vídeo. Ao explorar as características intrínsecas do vídeo, como resolução, taxa de quadros e conteúdo visual, esse método permite a seleção de configurações de CODEC personalizadas, resultando em uma melhor correspondência entre a codificação e as características específicas do vídeo, proporcionando assim uma experiência de visualização otimizada.

O trabalho de ([MERCAT; VIITANEN; VANNE, 2020](#)) ressalta a relevância do conjunto de dados UVG para a avaliação de desempenho, otimização e validação de CODECs de vídeo, fornecendo uma base sólida para pesquisas e avanços na área de compressão de vídeo. O conjunto de dados UVG permite uma análise aprofundada e uma compreensão mais abrangente das capacidades e limitações dos CODECs, impulsionando o desenvolvimento de técnicas mais eficientes e de alta qualidade.

A pesquisa de [RD e Udupa \(2020b\)](#) investiga a evolução dos CODECs de vídeo e sua

direção futura de pesquisa. O estudo revela que, ao longo do tempo, os CODECs de vídeo passarão por avanços significativos, impulsionados pela demanda por maior eficiência de compressão, qualidade de vídeo aprimorada e suporte a novas funcionalidades. Além disso, o artigo identifica as principais áreas de pesquisa para o futuro, incluindo a otimização da compressão baseada em aprendizado de máquina, aprimoramento da eficiência em termos de taxa de bits e qualidade visual, bem como a adoção de CODECs de próxima geração para acompanhar as necessidades emergentes de aplicações de vídeo avançadas.

O trabalho de [Kang e Xu \(2021\)](#) demonstra a importância do uso de FPGA para acelerar o processamento de CODECs de vídeo complexos, como o JPEG2000, e facilitar a transmissão eficiente de vídeo comprimido em tempo real por meio de redes de comunicação. A implementação proposta no artigo contribui para aprimorar o desempenho e a escalabilidade de sistemas de compressão e transmissão de vídeo.

[Le et al. \(2022\)](#) apresentam uma abordagem inovadora para a compressão de vídeo interquadros (inter-frame) utilizando técnicas de aprendizado de máquina em dispositivos móveis, destacam que o MobileCodec proporcionou um método eficiente e de alta qualidade para a compressão de vídeo em tempo real em dispositivos móveis, por meio da utilização de redes neurais para a predição e codificação de quadros subsequentes. Esse trabalho aborda a relevância do uso de técnicas de aprendizado de máquina para otimizar a compressão de vídeo em dispositivos móveis, permitindo uma transmissão e armazenamento mais eficiente de conteúdo multimídia em aparelhos com recursos limitados. O MobileCodec representa um avanço significativo na área de compressão de vídeo em dispositivos móveis, proporcionando uma experiência aprimorada aos usuários em termos de qualidade visual e desempenho.

[Tekalp et al. \(2021\)](#) apresentam uma introdução ao tema do uso de Deep Learning para restauração e compressão de imagem/vídeo. Os autores destacam que a aplicação de técnicas de Deep Learning tem revolucionado o campo da restauração e compressão de imagem/vídeo, permitindo melhorias significativas na qualidade visual e eficiência dos algoritmos de processamento. Essa citação enfatiza a importância do Deep Learning como uma abordagem inovadora para enfrentar os desafios relacionados à restauração e compressão de imagem/vídeo, fornecendo soluções mais precisas, rápidas e de alta qualidade.

[Tung e Gündüz \(2022\)](#) abordam o uso de Deep Learning para auxiliar na transmissão de vídeo sem fio. Os autores destacam que a aplicação de técnicas de Deep Learning tem sido explorada para melhorar a eficiência, a qualidade e a confiabilidade da transmissão de vídeo sem fio. Enfatizam a importância do uso de Deep Learning para otimizar a transmissão de vídeo, lidando com desafios inerentes à comunicação sem fio, como perda de pacotes, interferência e variações de qualidade do canal. A abordagem proposta pelos autores representa um avanço significativo na área da transmissão de vídeo sem fio, oferecendo soluções aprimoradas por meio do uso de técnicas de Deep Learning, resultando em uma melhor experiência de visualização para os usuários.

Wang et al. (2021) descrevem uma arquitetura eficiente de acelerador de Deep Learning para análise de vídeo comprimido. Os autores destacam que a arquitetura proposta oferece uma solução eficiente e de alto desempenho para a análise de conteúdo de vídeo comprimido por meio do uso de técnicas avançadas de aceleração de Deep Learning". Essa citação destaca a relevância do uso de aceleradores de hardware especializados para aumentar a eficiência e a velocidade de processamento de redes neurais profundas na análise de vídeos comprimidos. A arquitetura proposta representa um avanço significativo na área da análise de vídeo comprimido, fornecendo uma solução escalável e otimizada para extrair informações valiosas dos dados de vídeo comprimido de forma eficiente.

Lin et al. (2022) aborda o uso de Polar Codes para realizar a decodificação conjunta de fonte e canal para streaming de vídeo baseado em HEVC (High Efficiency Video Coding), que é um padrão de compressão de vídeo avançado. O uso de Polar Codes visa melhorar a eficiência e a confiabilidade da transmissão de vídeo em tempo real, permitindo a correção de erros que possam ocorrer durante a transmissão.

O estudo de Adhuran, Kulupana e Fernando (2022) explora como as técnicas de Deep Learning e o fluxo óptico bidirecional podem ser aplicados para prever o viewport em vídeos em 360°, a fim de melhorar a codificação e a eficiência da transmissão desses vídeos. A utilização do viewport permite concentrar os recursos de codificação no que o usuário está vendo, reduzindo a quantidade de dados a serem transmitidos e melhorando a experiência de visualização em dispositivos com capacidades de visualização limitadas, como óculos de realidade virtual ou plataformas de transmissão com largura de banda limitada.

Os autores Li et al. (2022) tratam de um framework de compressão de vídeo baseado em *Deep Learning* com uma transformação espacial implícita, voltado para aplicações na Internet das Coisas (IoT) abordam como o Deep Learning é aplicado ao processo de compressão de vídeo com o objetivo de torná-lo mais eficiente, considerando as restrições de recursos comuns em dispositivos IoT. O objetivo é criar um framework que seja capaz de comprimir vídeos de forma mais eficiente, mantendo uma qualidade visual aceitável, e que seja especialmente adequado para dispositivos IoT, onde os recursos computacionais, a largura de banda e o armazenamento são limitados.

Zhu et al. (2023) descrevem a abordagem baseada em Deep Learning para a derivação de modos intra na codificação de vídeo usando o padrão de codificação de vídeo chamado Versatile Video Coding (VVC) um padrão de codificação de vídeo avançado que visa melhorar a eficiência de compressão em relação aos padrões anteriores, como o High Efficiency Video Coding (HEVC). No entanto, com maior complexidade vem a necessidade de técnicas de codificação mais sofisticadas. Os autores propõem o uso de *Deep Learning* para a derivação de modos intra no VVC. Isso significa que os modos intra para a codificação de cada bloco de pixels são determinados usando redes neurais profundas que aprenderam padrões e características relevantes de dados de treinamento. A abordagem de Deep Learning pode levar a uma melhor

seleção de modos intra, resultando em maior eficiência de compressão e, portanto, em uma melhor qualidade de vídeo ou menor taxa de bits necessária para a transmissão. Essa abordagem combina os avanços em *Deep Learning* com o padrão de codificação VVC para melhorar a qualidade e eficiência da compressão de vídeo, tornando o processo de codificação mais eficaz para aplicações de alta qualidade e largura de banda limitada.

Zhang, Kwong e Wang (2020) revisam as diferentes abordagens de aprendizado de máquina aplicadas à codificação de vídeo e analisa como essas técnicas podem melhorar a eficiência da compressão e a qualidade visual dos vídeos resultantes, exploram como o aprendizado de máquina é utilizado para aprimorar a codificação de vídeo, identificando as vantagens e desafios dessas abordagens e fornecendo insights sobre o estado atual das otimizações baseadas em machine learning no campo da codificação de vídeo.

3.4 Análise dos Resultados do Mapeamento Sistemático

Diante dos estudos selecionados observou-se que a maioria dos trabalhos tratam do estudo de técnicas utilizando *Machine Learning*, as quais oferecem várias vantagens para CODECs de vídeo, permitindo melhorias significativas na eficiência de compressão, qualidade de vídeo e adaptação a diferentes cenários. No entanto esse trabalho busca suprir a lacuna existente no que diz respeito a quantidade de estudos acerca do CODEC de vídeo baseado em *Deep Learning* uma vez que o mesmo assim como a linguagem de máquina também possui inúmeras vantagens nesse âmbito.

Este estudo busca fazer uma análise comparativa dos diferentes métodos de *Deep Learning* aplicados aos CODECs de vídeo. Os CODECs de vídeo são fundamentais para a eficiente e transmissão de conteúdo audiovisual. O advento do *Deep Learning* trouxe novas oportunidades para melhorar o desempenho dos CODECs de vídeo, através da herança de recursos mais robustos, modelagem de redundâncias espaciais e temporais, e otimização de taxas de bits. Neste estudo, serão investigados e comparados diversos modelos de *Deep Learning* aplicados aos CODECs de vídeo, com o objetivo de identificar as abordagens mais promissoras e eficazes para a abordagem e transmissão de vídeos.

A aplicação de *Deep Learning* em CODECs de vídeo tem uma importância significativa devido ao potencial de melhorar a eficiência, qualidade e flexibilidade da gestão de vídeo. Os CODECs de vídeo tradicionais, como H.264 e H.265 (HEVC), são limitados em termos de eficiência. O uso de técnicas de *Deep Learning* permite explorar padrões complexos e hierárquicos nos dados de vídeo, gerados em um melhor flexível e, portanto, taxas de bits mais baixas para a mesma qualidade de vídeo.

4

Materiais e Métodos

Este Capítulo descreve os CODECs de Vídeo, Hardware, metodologia, e execução do experimento para a análise de desempenho, qualidade e eficiência do AVC, HEVC, AV1 e AIVC.

4.1 CODECs de Vídeo

A compressão de vídeo é essencial para a transmissão e armazenamento de conteúdo digital em alta qualidade, especialmente à medida que as resoluções continuam a aumentar e as plataformas de streaming se tornam mais populares. Diversos CODECs de vídeo têm sido desenvolvidos ao longo dos anos para atender a essa demanda, cada um com suas próprias vantagens e desvantagens em termos de eficiência, qualidade e licenciamento. Esta seção apresenta uma visão geral dos principais CODECs de vídeo, como VP09, AVC, HEVC, AV1 e AIVC, destacando suas características, aplicações e a evolução das técnicas de compressão para otimizar o uso da largura de banda sem comprometer a qualidade visual.

- CODEC de vídeo VP09

O VP09, também conhecido como VP9, é um CODEC de vídeo de vídeo desenvolvido pelo Google como sucessor do VP8, projetado para oferecer compressão eficiente de vídeo com qualidade elevada, principalmente em plataformas de streaming como YouTube. Ele é um CODEC de vídeo de código aberto e gratuito, sendo uma alternativa ao H.264 e ao HEVC (H.265), sem a necessidade de licenciamento. O VP9 suporta compressão de vídeo em alta definição, incluindo 4K e HDR, com uma melhor relação entre qualidade e taxa de bits em comparação com CODEC de vídeos anteriores. Seu objetivo é reduzir o tamanho dos arquivos de vídeo enquanto mantém a qualidade visual, tornando-o ideal para streaming em redes com largura de banda limitada.

- CODEC de vídeo AVC

O CODEC de vídeo AVC, também conhecido como H.264, é um padrão de compressão de vídeo amplamente utilizado em diversas aplicações, desde streaming de vídeo online até transmissão de televisão e videoconferências. Desenvolvido pela ITU-T (International Telecommunication Union Telecommunication Standardization Sector) e pelo ISO (International Organization for Standardization)/IEC (International Electrotechnical Commission), o H.264 oferece uma excelente relação entre qualidade de vídeo e taxa de bits, permitindo compressão eficiente sem perda significativa de qualidade. Ele é conhecido por sua flexibilidade e eficiência, suportando uma ampla gama de resoluções, de baixa definição até 4K e além. O H.264 tornou-se um dos CODEC de vídeos mais populares devido à sua alta compatibilidade com dispositivos e plataformas, além de ser o CODEC de vídeo padrão para a maioria dos vídeos na internet. Contudo, o uso do H.264 está sujeito a licenciamento e royalties.

- CODEC de vídeo HEVC

O CODEC de vídeo HEVC, também conhecido como H.265, é a evolução do H.264, projetado para oferecer compressão de vídeo ainda mais eficiente, especialmente em resoluções mais altas como 4K e 8K. Desenvolvido pelo ITU-T e ISO/IEC, o HEVC reduz significativamente a taxa de bits necessária para entregar vídeos da mesma qualidade, ou permite qualidade superior com a mesma taxa de bits, em comparação ao H.264. Isso é alcançado por meio de técnicas avançadas de compressão, como a previsão intra-quadro aprimorada e a codificação baseada em blocos maiores. O HEVC é amplamente utilizado em streaming de vídeo, transmissão de televisão e armazenamento de mídia, especialmente em conteúdos de alta definição e HDR. No entanto, seu uso está sujeito a patentes e licenciamento, o que pode limitar sua adoção em algumas aplicações.

- CODEC de vídeo AV1

O CODEC de vídeo AV1 é um padrão de compressão de vídeo de próxima geração desenvolvido pelo Alliance for Open Media (AOMedia), com o objetivo de oferecer uma alternativa mais eficiente e livre de royalties em comparação com o H.264 e o HEVC. Projetado para melhorar a qualidade de vídeo e reduzir a largura de banda necessária para streaming e armazenamento, o AV1 utiliza técnicas avançadas de compressão para atingir uma eficiência superior, especialmente em resoluções altas como 4K e 8K. Com sua natureza aberta e gratuita, o AV1 tem o potencial de se tornar uma escolha popular para uma ampla gama de aplicações, desde plataformas de streaming até dispositivos móveis e redes de distribuição de conteúdo.

- CODEC de vídeo AIVC

O CODEC de vídeo AIVC, ou Artificial Intelligence-based Video Coding, é um sistema de compressão de vídeo que utiliza técnicas de inteligência artificial para melhorar a eficiência da codificação e a qualidade do vídeo. Diferente dos CODEC de vídeos tradicionais, o AIVC

aplica algoritmos de aprendizado de máquina para otimizar a compressão, adaptando-se de maneira dinâmica ao conteúdo do vídeo e melhorando a preservação de detalhes visuais com menos dados. Isso permite uma redução significativa no tamanho dos arquivos e na largura de banda necessária para transmissão, ao mesmo tempo que mantém uma qualidade visual elevada. AIVC é particularmente promissor para aplicações de alta definição e streaming, onde a eficiência e a qualidade são essenciais.

4.2 Hardware

O hardware utilizado para gerar os arquivos de vídeo, bem como para o processamento dos resultados, foi um computador de mesa tipo PC, equipado com um monitor Dell P2422h conectado através de uma porta DisplayPort. A configuração pode ser observada na Tabela 3.

Processador	12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz
Memória RAM	32,0 GB
Placa Mãe	ASUS Prime H610M-E D4
Placa de Vídeo	PNY Geforce RTX 3050 8GB
Armazenamento	Kingston SSD NV2 1TB NVMe M2.
Resolução de Vídeo	1920x1080p

Tabela 3 – Hardware

4.3 Metodologia

Este item descreve a metodologia utilizada no experimento.

4.3.1 Métricas

Nesta seção, abordamos três métricas fundamentais utilizadas para avaliar a qualidade de vídeos: PSNR, VMAF e MS-SSIM. O PSNR (Relação Sinal-Ruído de Pico) mede a diferença entre o vídeo original e o comprimido, fornecendo uma indicação quantitativa da qualidade. O VMAF, desenvolvido pela Netflix, oferece uma avaliação mais precisa da qualidade de vídeo, combinando múltiplos métodos de análise para refletir melhor a percepção humana. Por fim, o MS-SSIM expande a métrica SSIM, avaliando a similaridade estrutural em múltiplas escalas, o que permite uma análise mais detalhada e abrangente da qualidade visual.

- PSNR

O algoritmo do PSNR (Relação Sinal-Ruído de Pico) mede a qualidade de uma imagem ou vídeo comprimido comparando-o com a versão original. Primeiro, calcula-se o Erro Quadrático Médio (MSE), que é a média dos quadrados das diferenças entre os pixels correspondentes das duas imagens. Em seguida, o PSNR é calculado usando a fórmula logarítmica que relaciona o valor máximo possível de um pixel (geralmente 255 para imagens de 8 bits) com o MSE. Um valor de PSNR mais alto indica que a imagem comprimida está mais próxima da original, ou seja, tem uma melhor qualidade.

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE). \end{aligned}$$

Figura 4 – Cálculo do índice PSNR.

- VMAF

O VMAF (Video Multi-method Assessment Fusion) é uma métrica desenvolvida pela Netflix para avaliar a qualidade de vídeos com base em como os humanos percebem essas imagens. Ele combina vários métodos de avaliação, incluindo análises subjetivas e objetivas, para criar uma pontuação que reflete melhor a experiência visual real do espectador. O VMAF leva em consideração fatores como a nitidez, o contraste e a fidelidade do vídeo, e é projetado para oferecer uma avaliação mais precisa e correlacionada com a percepção humana, o que o torna útil para otimização de qualidade em serviços de streaming e outras aplicações de vídeo. Podemos ver seu algoritmo na listagem [A.1](#).

- MS-SSIM

O MS-SSIM (Multi-Scale Structural Similarity) é uma métrica de avaliação de qualidade de imagem que estende o SSIM (Structural Similarity Index) ao considerar múltiplas escalas de resolução. Enquanto o SSIM compara a similaridade estrutural entre uma imagem original e uma imagem de referência em uma única escala, o MS-SSIM avalia essas imagens em várias resoluções, oferecendo uma avaliação mais completa e robusta da percepção visual da qualidade da imagem. Isso é particularmente útil para capturar diferenças de qualidade em diferentes níveis de detalhamento e tamanhos de exibição, quanto maior o valor do MS-SSIM, melhor será a qualidade percebida da imagem.

$$\text{MSSIM}(X, Y) = \frac{1}{M} \sum_{j=1}^M \text{SSIM}(x_j, y_j)$$

Figura 5 – Cálculo do índice MSSIM entre as imagens X e Y.

4.3.2 Execução

Para executar o experimento, foram usados 4 passos:

- 1. Converter o vídeo original para os determinados CODEC de vídeo

Foi utilizado o site *Convertio* com as para converter o CODEC de vídeo original, sempre utilizando a qualidade mais alta de cada codec, que era VP09, para AVC, HEVC e AV1, além de realizar a conversão da resolução original de 4k (3840x2160 pixels) para 720p (1280x720 pixels) e 360p (640x360 pixels).

- 2. Obter o PSNR

Para obter o valor do PSNR dos vídeos, foi necessário abrir os mesmos utilizando o visualizador de vídeo *VLC Media Player*. A tela foi capturada de todos os arquivos utilizando o capturador padrão do próprio *Windows*, mantendo o mesmo formato em todas as capturas o PNG (Portable Network Graphics), resolução 1920x1080 pixels (resolução padrão do monitor da máquina utilizada) e sempre utilizando o mesmo quadro. A precisão foi garantida através de uma ferramenta do próprio *VLC*, que mostra quantos quadros foram descartados devido à pré-visualização. Isso foi necessário porque o software PSNR 1.2 (Figura 6), utilizado para obter o valor do PSNR, só funciona se as imagens estiverem no mesmo formato e possuírem a mesma resolução.

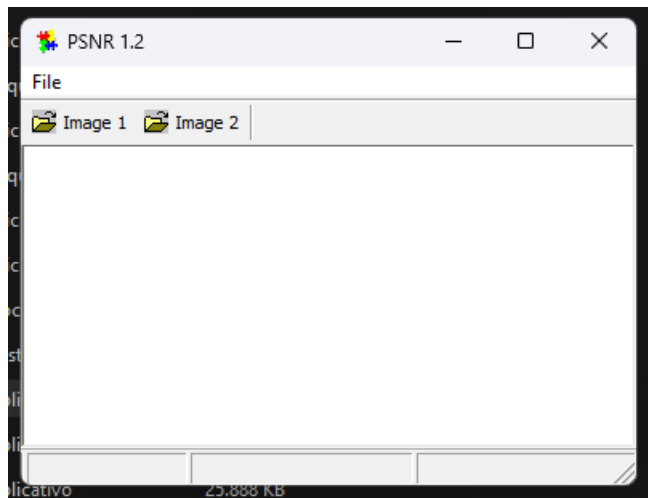
- 3. Obter o VMAF

Obter o valor do VMAF foi mais fácil. Utilizaram-se apenas os arquivos de vídeo que já haviam sido convertidos para os respectivos CODECs, juntamente com o software VMAF-GUI (Figura 7). Diferentemente do PSNR, neste caso, foi possível calcular o valor para todo o vídeo e não apenas para um quadro.

- 4. Comparar resultados

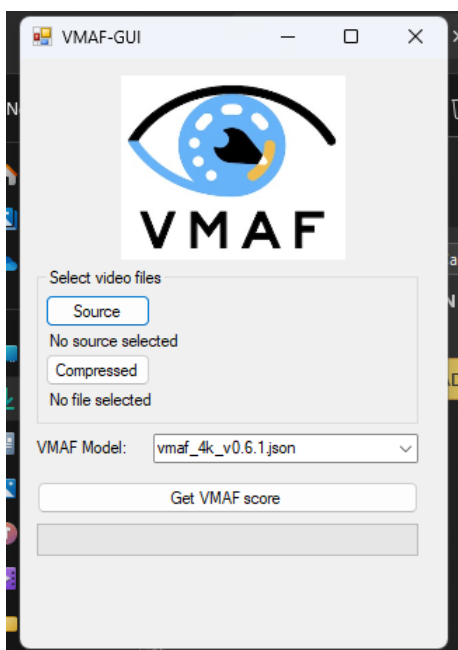
Após obter os valores de PSNR e VMAF, agora comparamos o tamanho do arquivo e bit rate utilizando os dados obtidos no *VLC media player*, também comparamos os dados obtidos no PSNR e VMAF.

Figura 6 – Imagem do software PSNR 1.2



Fonte: Próprio autor

Figura 7 – Imagem do software VMAF-GUI



Fonte: Próprio autor

5

Resultados

Este capítulo apresenta uma análise dos resultados obtidos, incluindo a conversão de vídeos em diferentes codecs e resoluções, a avaliação de qualidade com métricas como PSNR e VMAF, e a exploração do codec emergente AIVC, baseado em inteligência artificial. Os resultados destacam a superioridade dos codecs mais recentes e demonstram avanços significativos na compressão e qualidade de vídeos, sublinhando a importância da inovação contínua na área.

5.1 Convertendo os Arquivos de Vídeo

Antes do início do experimento, utilizamos um vídeo (imagem 8) de 16 segundos, baixado através do Youtube¹, livre de direitos autorais, foi realizado uma conversão dos arquivos de vídeo utilizando o site Convertio², o arquivo original estava codificado em VP09 na resolução de 4k (3840x2160 pixels) e foi convertido para 3 tipos de CODECs diferentes, sendo eles o AVC, HEVC e AV1, cada um em duas resoluções diferentes a 720p (1280x720 pixels) e 360p (640x360 pixels), assim gerando 8 arquivos de vídeo como mostrado na figura 9.

¹ <https://www.youtube.com/watch?v=sI8diqCAFw>

² <https://convertio.co/>

Figura 8 – Captura de tela do Vídeo Utilizado para Referência



Fonte: Próprio autor

Figura 9 – Tamanho das imagens

	VP09 Original 4k.we...		03/07/2024 14:48	Arquivo WEBM	22.840 KB	00:00:16
	AVC 720p.mp4		13/08/2024 21:31	Arquivo MP4	2.802 KB	00:00:16
	HEVC 720p.mp4		03/07/2024 16:14	Arquivo MP4	1.712 KB	00:00:16
	AV1 720p.mp4		22/07/2024 13:57	Arquivo MP4	1.075 KB	00:00:16
	AVC 360p.mp4		13/08/2024 21:35	Arquivo MP4	767 KB	00:00:16
	HEVC 360p.mp4		16/07/2024 21:20	Arquivo MP4	535 KB	00:00:16
	AV1 360p.mp4		30/07/2024 22:45	Arquivo MP4	369 KB	00:00:16

Fonte: Próprio autor

A figura 9 ilustra como diferentes codecs (VP9, AVC, HEVC, e AV1) e resoluções (4K, 720p, 360p) afetam o tamanho dos arquivos de vídeo para a mesma duração. Em geral, codecs mais recentes e eficientes como HEVC e AV1 produzem arquivos menores em comparação com codecs mais antigos como AVC, especialmente em resoluções mais baixas.

5.2 Obtendo o PSNR

Nesse estudo investigativo em diferentes cenários de aplicação, utilizamos 8 amostras do mesmo vídeo, por causa da limitação do software PSNR 1, sendo uma em 4k, três em HD e três

em 360p, a primeira VP09 Original 4k, é o arquivo original, codificado em VP09, na resolução 4k (3840x2160), o bit rate é de 11250kbps, com o tamanho em 22.840 KB, com ele é o arquivo referência não há valor de PSNR.

O próximo é o AVC 720p, codificado em HEVC, com a resolução em HD (1280x720), com o bit rate de 1420kbps, pesando cerca de 2.802 KB, com o PSNR de 34.16dB. Na sequencia temos o HEVC 720, também temos a resolução em HD, o bit rate em 864kbps, com o tamanho em 1.712 KB, o PSNR é de 34,85dB.

AV1 720p é a ultima amostra na resolução HD, possui o bit rate mais baixo nessa resolução de 542kbps, também o mais leve na sequência com 1.075 e possui o maior PSNR na resolução HD de 35,05dB.

Abrindo a resolução de 360p (640x360) temos o AVC 360, com o bit rate de 383kbps, pesando 767 KB e com o pior PSNR de todas as amostras com 29,85dB, o HEVC 360 é o intermediário da resolução, utilizando o codec HEVC, na resolução 360p, com o bit rate de 264kbps, com 535 KB e possuindo um PSNR de 30,15dB, e fechando essas amostra temos o AV1 360p, codificado em AVC, com a resolução 360, com o menor bit rate da sequencia com 180kbps, sendo o arquivo mais leve 369 KB, por outro lado possui um bom PSNR de 30,16dB.

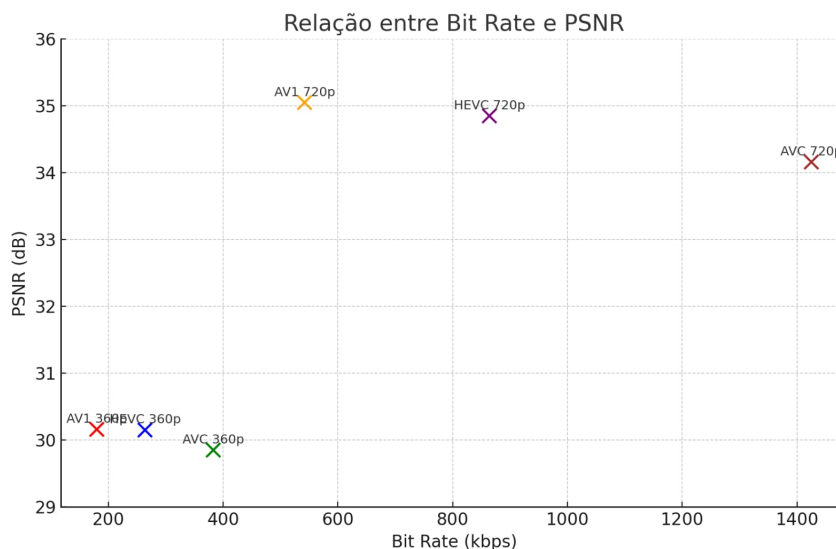
Nome	Codec	Resolução	Bit rate	Tamanho	PSNR
VP09 Original 4k	VP09	3840x2160	11250kbps	22.840 KB	Referência
AVC 720p	AVC	1280x720	1425kbps	2.802 KB	34.16dB
HEVC 720p	HEVC	1280x720	864kbps	1.712 KB	34.85dB
AV1 720p	AV1	1280x720	542kbps	1.075 KB	35.05dB
AVC 360p	AVC	640x360	383kbps	767 KB	29,85dB
HEVC 360p	HEVC	640x360	264kbps	535 KB	30,15dB
AV1 360p	AV1	640x360	180kbps	369 KB	30,16dB

Tabela 4 – Comparação de codecs, resoluções, bit rates, Tamannhos e PSNR

Nome	Bitrate (%)	Tamanho (%)	PSNR (%)
VP09 Original 4K	100%	100%	Referência
AVC 720p	12,67%	12,27%	97,46%
HEVC 720p	7,68%	7,50%	99,43%
AV1 720p	4,82%	4,71%	100,00%
AVC 360p	3,40%	3,36%	85,17%
HEVC 360p	2,35%	2,34%	86,02%
AV1 360p	1,60%	1,62%	86,05%

Tabela 5 – Porcentagem do bitrate, tamanho e PSNR em relação à referência (VP09 Original 4K para bitrate/tamanho e AV1 720p para PSNR)

Figura 10 – Reação entre Bit Rate e PSNR



Fonte: Próprio autor

A Figura 10 e a Tabela 4 mostram a relação entre a taxa de bits (bit rate) e a qualidade do vídeo (PSNR) para diferentes codecs (AVC, HEVC, AV1) em resoluções 720p e 360p. O codec AV1 se destaca por oferecer uma melhor qualidade (maior PSNR) com uma taxa de bits menor em comparação com AVC e HEVC, especialmente na resolução 720p, o que indica uma compressão mais eficiente. Isso significa que o AV1 pode produzir vídeos de qualidade

semelhante ou superior utilizando menos dados, resultando em tamanhos de arquivo menores, sendo o ideal para ser utilizado em streaming de vídeo e armazenamento em nuvem.

A Tabela 5 apresenta a relação percentual do bitrate, do tamanho do arquivo e da métrica PSNR (Peak Signal-to-Noise Ratio) em comparação com as referências escolhidas. O bitrate e o tamanho do arquivo são normalizados em relação ao VP09 Original 4K (100%), enquanto o PSNR é normalizado em relação ao maior valor da amostra, que pertence ao AV1 720p (35,05 dB), sendo esse considerado 100%.

Os valores mostram que os codecs mais modernos, como AV1 e HEVC, oferecem uma compressão significativamente maior sem comprometer a qualidade do vídeo. Por exemplo, o AV1 720p reduz o bitrate para apenas 4,82% do original, enquanto ainda mantém 100% da qualidade de PSNR relativa. Já codecs mais antigos, como AVC, apresentam uma relação menos eficiente entre redução de tamanho e manutenção da qualidade.

Além disso, os vídeos em 360p possuem um bitrate e tamanho extremamente reduzidos, porém, como esperado, apresentam perdas significativas na qualidade medida pelo PSNR. O AV1 360p, por exemplo, tem apenas 1,60% do bitrate do VP09 Original 4K, enquanto mantém 86,05% da qualidade PSNR relativa.

5.3 Obtendo o VMAF

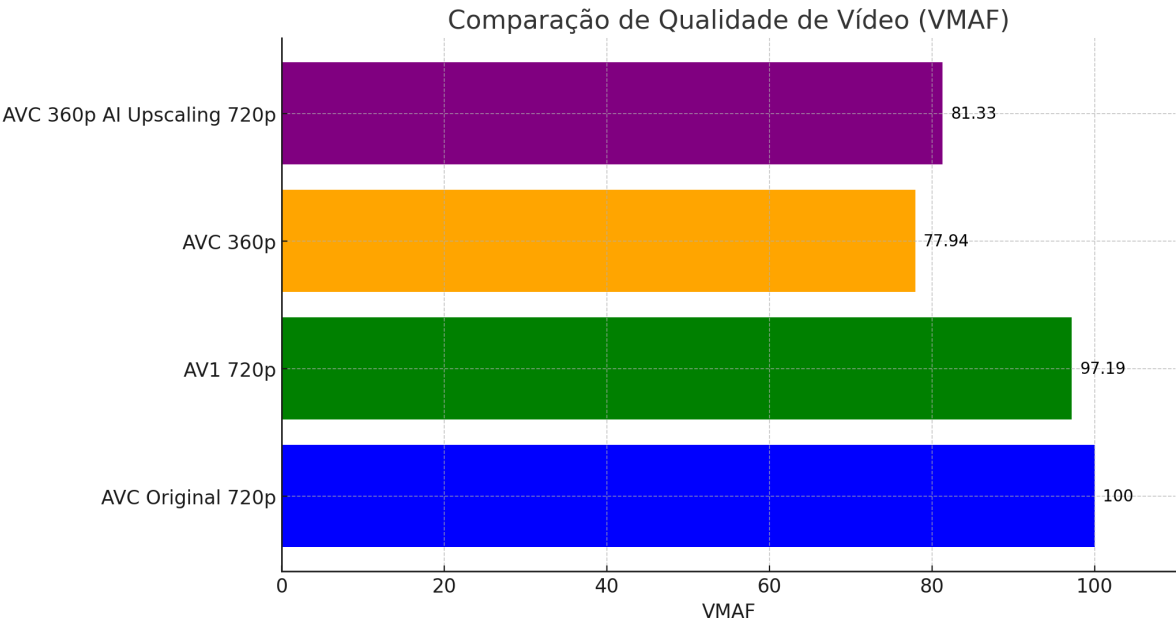
Nesse cenário, exploramos técnicas de otimização e melhoria em codecs, com o objetivo de calcular o VMAF para avaliar a qualidade dos CODECs de vídeo existentes, visando reduzir a taxa de bit rate e melhorar a qualidade visual sem aumentar o tamanho do arquivo, tivemos cinco amostras, sendo três delas codificada em AVC e uma em AV1, nessa circunstâncias ignoramos o bit rate e tamanho do arquivo, já que iremos utilizar técnicas de pós-processamento e substituímos o PSNR pelo VMAF como parâmetro de qualidade, pois o PSNR se mostrou impreciso quando comparamos com arquivos que passaram pelo o processo de pós-processamento de upscaling utilizando Deep Learning, utilizamos como referência o arquivo de vídeo AVC Original 720p, codificado em AVC, com a resolução em HD (1280x720), como é o arquivo original utilizamos ele como orientação para o valor de VMAF. Na sequencia temos o arquivo já apresentado anteriormente o AV1 720p, ele tem um valor de VMAF de 97,19.

Depois temos o arquivo de vídeo em AVC 320p com a resolução em 640x360, o VMAF dele foi de 77,94, esse arquivo foi utilizado como base para gerar o próximo. O ultimo é o arquivo nomeado de AVC 360p AI Upscaling 720p, foi codificado em AVC na resolução HD e tem o VMAF de 81,33, esse em especifico foi utilizado técnicas de pós-processamento utilizando o software Winxvideo AI.

Nome	Codec	Resolução	VMAF	Parâmetro
AVC Original 720p	AVC	1280x720	Referência	100%
AV1 720p	AV1	1280x720	97,19	97,19%
AVC 360p	AVC	640x360	77,94	77,94%
AVC 360p AI Upscaling 720p	AVC	1280x720	81,33	81,33%

Tabela 6 – Comparação de codecs, resoluções e VMAF

Figura 11 – Comparação de Qualidade de Vídeo (VMAF)



Fonte: Próprio autor

A Tabela 6 apresenta uma comparação entre diferentes arquivos de vídeo, considerando o codec utilizado, a resolução e o valor de VMAF (Video Multimethod Assessment Fusion), que é uma métrica de qualidade de vídeo. Os testes incluem vídeos codificados nos formatos AVC e AV1, com diferentes resoluções e técnicas de upscaling aplicadas.

A primeira linha da tabela mostra o arquivo de vídeo original em 720p, codificado com o codec AVC, que serve como referência para a comparação, apresentando um valor de VMAF de 100%. Na segunda linha, temos um arquivo de vídeo codificado com o codec AV1 em 720p, que obteve um VMAF de 97,19%, ligeiramente inferior ao valor de referência, indicando uma leve perda de qualidade em relação ao arquivo original.

A terceira linha apresenta um arquivo de vídeo em 360p, utilizando o codec AVC, com um VMAF de 77,94%, que indica uma redução significativa na qualidade de vídeo devido à menor resolução. Por fim, a última linha exibe o arquivo AVC 360p com a técnica de upscaling por inteligência artificial (AI) para 720p. Esse arquivo obteve um valor de VMAF de 81,33%, superando o VMAF do arquivo original em 360p, mas ainda abaixo do arquivo original em 720p.

Os resultados gerados foram implementados na Tabela 6 e na Figura 11, onde observamos

que o arquivo que utilizou a técnica de Deep Learning (AVC 360p AI Upscaling 720p) teve uma pontuação VMAF maior que seu arquivo original (AVC 360p).

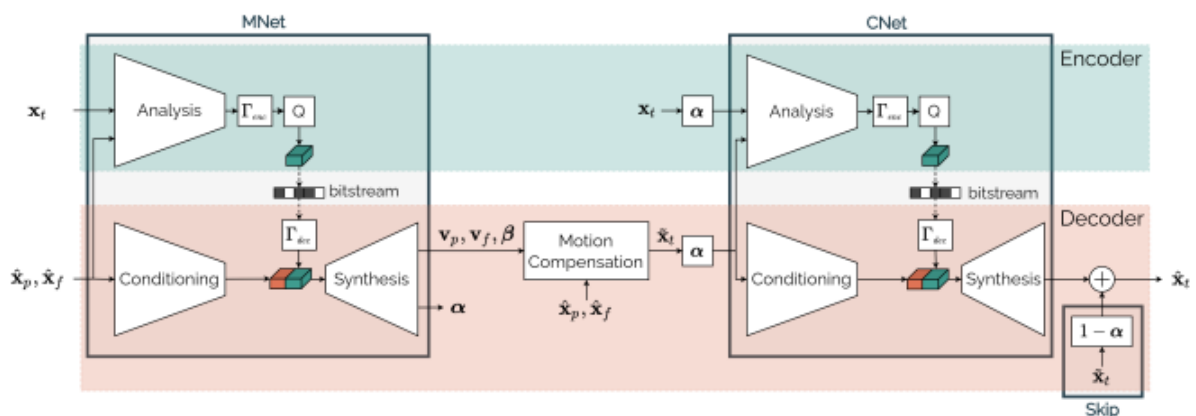
5.4 AIVC: Artificial Intelligence-base Video Coding

Um codec baseado em IA e Deep Learning utiliza redes neurais para compressão e descompressão de dados, oferecendo maior eficiência e melhor qualidade em comparação aos codecs tradicionais. Esses modelos são treinados para representar dados de forma compacta, resultando em menos artefatos e preservando mais detalhes, mas exigem maior poder computacional e podem enfrentar desafios como latência em aplicações em tempo real (LADUNE; PHILIPPE, 2022).

(LADUNE; PHILIPPE, 2022) destaca que O AIVC foi projetado para ser um codec versátil, capaz de suportar qualquer configuração de codificação desejada. Ele é avaliado em condições de teste que buscam equilibrar os conhecimentos adquiridos com as práticas da comunidade de codificação de vídeo convencional.

Codecs de vídeo utilizando AI e Deep Learning já estão sendo realidade, ainda apesar que estão em fase iniciais de desenvolvimento já temos algum resultado, um deles é o AIVC: Artificial Intelligence-base Video Coding, desenvolvido pela a dupla Théo Ladune e Pierrick Philippe, utilizando majoritariamente a linguagem Python. Apesar do projeto está disponível no Github, não foi possível instalar ele no computador, porém o autor disponibilizou alguns testes que o mesmo realizou.

Figura 12 – Processo de Encoder e Decoder do AIVC

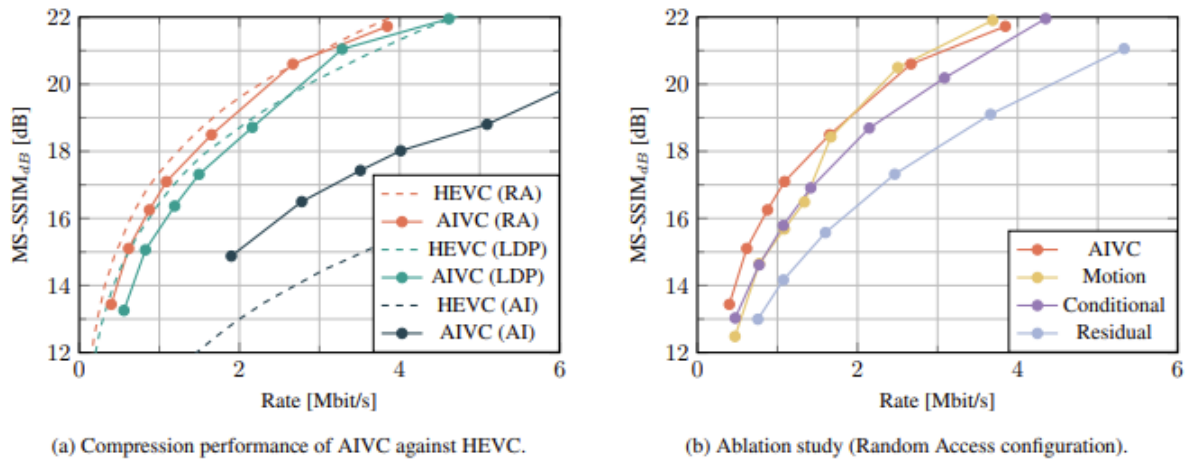


Fonte: (LADUNE; PHILIPPE, 2022)

A Figura 12 mostra a arquitetura de um sistema de compressão de vídeo baseado em redes neurais, composto por duas redes principais: MNet e CNet. A MNet processa a imagem atual, realizando análise, quantização e síntese para gerar uma versão comprimida, enquanto utiliza

parâmetros de movimento para prever quadros futuros. A CNet refina essa imagem comprimida e a combina com a original através de uma conexão residual ponderada, melhorando a qualidade da imagem reconstruída final. O processo utiliza compensação de movimento e uma combinação ponderada para otimizar a compressão e reconstrução do vídeo.

Figura 13 – Comparativo Bit Rate e MS-SSIM



Fonte: (LADUNE; PHILIPPE, 2022)

Podemos observar que a unidade de medida utilizada por (LADUNE; PHILIPPE, 2022) é o MS-SSIM (Multi-Scale Structural Similarity Index Measure), uma métrica que avalia a qualidade de imagem ou vídeo em comparação com uma referência, medindo a similaridade perceptual entre ambos. Diferente de métricas como o PSNR (Peak Signal-to-Noise Ratio), que se baseiam principalmente em diferenças absolutas de pixel, o MS-SSIM enfoca características estruturais que são mais relevantes para a percepção humana, como luminância, contraste e estrutura. Além disso, (ARAÚJO; MONTEIRO, 2013) também aborda aspectos semelhantes no contexto da análise de qualidade visual.

A figura 13 mostra dois gráficos que comparam como diferentes métodos de compressão de vídeo se saem. No gráfico à esquerda, vemos como a compressão feita por um codec baseado em inteligência artificial (AIVC) se compara ao HEVC, que é um padrão tradicional de compressão. O AIVC se sai melhor em todas as configurações testadas, alcançando uma qualidade de imagem mais alta (medida em MS-SSIM/dB) para a mesma quantidade de dados transmitidos.

No gráfico à direita, temos um estudo que analisa cada parte do AIVC para entender como elas contribuem para o resultado final. Ele mostra que o desempenho completo do AIVC, quando todas as partes estão funcionando juntas, é o melhor, mas também revela como cada componente (Movimento, Condicional, Residual) afeta a qualidade da compressão.

5.5 Análise dos Resultados

A conversão dos arquivos de vídeo revelou insights significativos sobre o impacto dos codecs e das resoluções na eficiência de compressão. Observou-se que codecs mais recentes, como HEVC e AV1, oferecem vantagens consideráveis sobre codecs mais antigos, como AVC. Em termos de tamanho de arquivo, tanto o HEVC quanto o AV1 conseguiram reduzir o tamanho significativamente em comparação ao AVC, especialmente em resoluções menores. Por exemplo, na resolução 720p, o codec AV1 gerou o menor tamanho de arquivo (1.075 KB), com a melhor qualidade de imagem (PSNR de 35.05 dB), demonstrando sua superioridade em eficiência de compressão. O AV1 apresentou uma redução de 61,63% no tamanho do arquivo e uma melhoria de 2,61% no PSNR em relação ao AVC 720p.

Na análise dos resultados, o codec AV1 se destacou, especialmente nas resoluções de 720p e 360p. O AV1 não só reduziu o tamanho do arquivo de forma eficaz, mas também alcançou o maior PSNR em comparação com AVC e HEVC. Isso indica que o AV1 oferece a melhor relação entre qualidade e compactação, proporcionando uma qualidade visual superior com menor bit rate.

A diferença de PSNR entre codecs na resolução 720p, por exemplo, revela que o AV1 (35.05dB) supera o HEVC (34.85dB) e o AVC (34.16dB) em termos de qualidade visual, mesmo com uma taxa de bits significativamente mais baixa. Na resolução 360p, o AV1 também demonstrou um leve aumento no PSNR em comparação com HEVC e AVC, confirmando sua eficácia em diferentes configurações de resolução e bit rate.

O VMAF, que considera a percepção humana de qualidade de vídeo, forneceu uma visão mais detalhada sobre a qualidade percebida. Os resultados mostraram que o vídeo codificado com AV1 teve o maior VMAF (97,19), indicando uma excelente qualidade visual. Isso é consistente com os resultados de PSNR, reforçando a conclusão de que o AV1 proporciona uma compressão eficiente sem comprometer a qualidade.

Por outro lado, o vídeo com técnicas de upscaling baseadas em inteligência artificial (AVC 360p AI Upscaling 720p) também apresentou uma melhoria significativa no VMAF (81,33) em comparação com o AVC 360p (77,94). Esta melhoria confirma a eficácia das técnicas de upscaling em melhorar a qualidade do vídeo, mesmo quando a resolução original é menor.

O AIVC, um codec emergente baseado em inteligência artificial, demonstrou uma capacidade superior em comparação com o HEVC em todas as configurações testadas. A análise revelou que o AIVC consegue proporcionar uma qualidade de imagem superior (medida em MS-SSIM) com menor bit rate, evidenciando a eficácia das redes neurais na compressão de vídeo. O AIVC superou o HEVC em termos de qualidade de imagem e eficiência de compressão, indicando um avanço significativo na tecnologia de codecs de vídeo.

A figura 13 mostram que o AIVC não só oferece uma melhor qualidade perceptual, mas também mostra uma capacidade promissora para otimizar a compressão e melhorar a qualidade

visual, apesar de ainda estar em fase inicial de desenvolvimento.

A análise dos resultados demonstra que codecs mais recentes, como AV1 e AIVC, são altamente eficientes em termos de compressão e qualidade de vídeo. O AV1 se destaca por oferecer a melhor qualidade com menor bit rate, e o AIVC apresenta um avanço significativo na compressão de vídeo, mostrando um potencial promissor para futuras aplicações. As técnicas de upscaling baseadas em inteligência artificial também se mostram eficazes na melhoria da qualidade visual, evidenciando a importância da inovação contínua na codificação de vídeo.

6

Considerações Finais

Este capítulo descreve as considerações finais acerca dos estudos empreendidos na elaboração desta dissertação.

O objetivo macro desta pesquisa foi analisar publicações científicas com o propósito de identificar quais contribuições foram propostas e suas limitações no âmbito do paradigma Codec baseado em Deep Learning em relação a abordagens para otimização de Codec com objetivo de reduzir a quantidade de banda em streaming.

Para o alcance deste objetivo, a primeira etapa deste trabalho consiste em um levantamento e análise dos trabalhos relacionados ao tema.

Para o desenvolvimento da primeira etapa, elaborou-se uma questão norteadora capaz de definir o que estaria sendo procurado e quais resultados se pretendia alcançar, para assim, selecionar as bibliografias pertinentes. A questão norteadora possibilitou traçar um panorama da pesquisa e construir uma string de busca que definiu critérios de exclusão e inclusão de publicações dada a relevância acerca do tema.

Já na segunda etapa, a de revisão bibliográfica, buscou-se concentrar um maior número de trabalhos que permitiu buscar, selecionar, identificar, avaliar e interpretar toda literatura disponível e relevante frente ao tema proposto ou questão de pesquisa.

No desenvolvimento dessa revisão bibliográfica, utilizou-se as bases de dados Springer Link 1, ACM Digital Library 2, IEEE Digital Library 3, ISI Web Of Science 4, Science Direct 5 e Scopus.

Esta etapa consolidou a revisão sistemática a partir da qual se construiu um corpo de conhecimento com o intuito de compreender os avanços recentes em codecs de vídeo baseado em *deep learning* e diretrizes valiosas para pesquisadores e engenheiros interessados em melhorar a eficiência e a qualidade da cobertura de vídeo.

A terceira etapa consiste na comparação de diferentes codecs em imagens afim de aferir

a eficiência na performance e qualidade gráfica, com isso foi possível concluir que o estudo de codecs de vídeo baseado em *deep learning* é de grande importância na área de codificação de vídeo e processamento de mídia, o mesmo, tem se mostrado eficiente no que diz respeito a resolução de problemas complexos em várias áreas, incluindo visão computacional e processamento de linguagem natural. Quando aplicado ao desenvolvimento de codecs de vídeo, o *deep learning* traz diversas vantagens e oportunidades como, melhorar a eficiência de compressão dos codecs de vídeo, permitindo uma maior taxa de compressão sem comprometer a qualidade visual. Isso resulta em arquivos de vídeo menores, economia de espaço de armazenamento e redução do consumo de largura de banda para transmissão em redes. Podem aprender a representar melhor as características visuais e padrões presentes em diferentes cenas de vídeo, o que traz melhoria significativa na qualidade do vídeo codificado, reduzindo artefatos indesejados, como blocagem e borrrões. Podem ser mais adaptáveis a diferentes tipos de conteúdo de vídeo, ajustando-se automaticamente para oferecer a melhor qualidade de compressão para cada tipo de cena, seja uma paisagem, uma pessoa ou uma sequência de ação rápida. Alguns algoritmos de *deep learning* podem simplificar a estrutura dos codecs, tornando sua implementação mais eficiente e menos complexa, o que é benéfico para a otimização do desempenho em *hardware* ou *software*. No entanto, é importante ressaltar que o desenvolvimento de codecs de vídeo baseado em *deep learning* também traz desafios, como a complexidade computacional e a necessidade de grandes quantidades de dados de treinamento. Além disso, a interoperabilidade com os padrões existentes e a questão da licenciabilidade também devem ser consideradas.

No geral, o estudo de codecs de vídeo baseado em *deep learning* tem o potencial de revolucionar a forma como o vídeo é comprimido, transmitido e exibido, abrindo caminho para avanços significativos na área de processamento de mídia e aplicações práticas em vários setores, incluindo transmissão, *streaming*, realidade virtual, jogos e muito mais.

6.1 Limitações do Trabalho

Mediante observações cabíveis, após análise dos dados obtidos, convém conceituar a AIVC como base de estudo relevante no que concerne ao Codec utilizando Inteligência Artificial, além disso não foi possível gerar resultados, devido a indisponibilidade de acesso aos arquivos do Codec, porém os resultados bem como o artigo que utiliza ainda estão disponíveis e foram utilizados no trabalho.

6.2 Trabalhos Futuros

Nesta seção serão apresentadas algumas contribuições para trabalhos futuros. São elas:

- Desenvolver um Codec funcional utilizando Inteligência Artificial;

- Realizar novos testes.

Referências

- ADHURAN, J.; KULUPANA, G.; FERNANDO, A. Deep learning and bidirectional optical flow based viewport predictions for 360° video coding. *IEEE Access*, IEEE, v. 10, p. 118380–118396, 2022. Citado na página 32.
- ARAÚJO, J. N.; MONTEIRO, C. de C. Análise de métricas de similaridade estrutural. In: SBC. *Anais do I Encontro Nacional de Computação dos Institutos Federais*. [S.l.], 2013. p. 2102–2105. Citado na página 47.
- ARRIVUKANNAMMA, M.; SATHIASEELAN, J. A study on codec quality metric in video compression techniques. In: IEEE. *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. [S.l.], 2015. p. 1–5. Citado 2 vezes nas páginas 12 e 20.
- BITMOVIN. Vp9 codec: Complete guide to google’s open source codec. *Bitmovin*, 2023. Disponível em: <<https://bitmovin.com/vp9-codec>>. Citado na página 24.
- COCHRANE, A. B.; SHKIPEROV, A. I. A comparative analysis of modern video codecs. *Journal of Video Technology*, Video Tech Publishers, v. 15, n. 3, p. 245–260, 2020. Citado na página 17.
- GHOSH, M.; SINGHAL, D. C.; WAYAL, R. Desvq: Deep learning based streaming video qoe estimation. In: *Proceedings of the 23rd International Conference on Distributed Computing and Networking*. [S.l.: s.n.], 2022. p. 19–25. Citado 2 vezes nas páginas 18 e 28.
- GUMLET. H.265 - high efficiency video coding - a complete guide. 2023. Accessed: 2024-07-19. Disponível em: <<https://www.gumlet.com/h265-high-efficiency-video-coding-guide/>>. Citado na página 24.
- JACK, K. *Video Demystified: A Handbook for the Digital Engineer*. 5th. ed. Burlington, MA: Newnes, 2004. Citado na página 17.
- KANG, Y.; XU, X. A system and its implementation based on fpga for video jpeg2000 codec and network transmission. In: *2021 5th International Conference on Digital Signal Processing*. [S.l.: s.n.], 2021. p. 260–265. Citado na página 31.
- KAZANTSEV, R.; ZVEZDAKOV, S.; VATOLIN, D. S. Machine-learning-based method for finding optimal video-codec configurations using physical input-video features. In: *DCC*. [S.l.: s.n.], 2020. p. 374. Citado na página 30.
- KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. 2004, p. 1–26, 2004. Citado na página 29.
- KWAN, C.; SHANG, E.; TRAN, T. D. Perceptually lossless video compression with error concealment. In: *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*. [S.l.: s.n.], 2018. p. 1–5. Citado na página 30.
- LADUNE, T.; PHILIPPE, P. Aivc: Artificial intelligence based video codec. In: IEEE. *2022 IEEE International Conference on Image Processing (ICIP)*. [S.l.], 2022. p. 316–320. Citado 2 vezes nas páginas 46 e 47.

- LE, H. et al. Mobilecodec: neural inter-frame video compression on mobile devices. In: *Proceedings of the 13th ACM Multimedia Systems Conference*. [S.l.: s.n.], 2022. p. 324–330. Citado 3 vezes nas páginas 17, 19 e 31.
- LI, Q. et al. Learning-based video compression framework with implicit spatial transform for applications in the internet of things. *IEEE Transactions on Industrial Informatics*, IEEE, 2022. Citado na página 32.
- LIN, J. et al. Joint source-channel decoding of polar codes for hevc-based video streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM New York, NY, v. 18, n. 4, p. 1–23, 2022. Citado 2 vezes nas páginas 18 e 32.
- LOPEZ, L. M. Why h.264 is the most popular video compression standard. *Videomaker*, 2021. Accessed: 2024-07-19. Disponível em: <<https://www.videomaker.com/>>. Citado na página 24.
- MERCAT, A.; VIITANEN, M.; VANNE, J. Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In: *Proceedings of the 11th ACM Multimedia Systems Conference*. [S.l.: s.n.], 2020. p. 297–302. Citado na página 30.
- MIAO, D. et al. Layered screen video coding leveraging hardware video codec. In: IEEE. *2013 IEEE International Conference on Multimedia and Expo (ICME)*. [S.l.], 2013. p. 1–6. Citado 2 vezes nas páginas 13 e 30.
- OHM, J.-R. Advances in scalable video coding. *Proceedings of the IEEE*, IEEE, v. 93, n. 1, p. 42–56, 2005. Citado na página 23.
- OZER, J. L. *Video compression for flash, apple devices and HTML5*. [S.l.]: Doceo Publishing, 2011. Citado 2 vezes nas páginas 16 e 17.
- RAMACHANDRAN, P. et al. Content adaptive live encoding with open source codecs. In: *Proceedings of the 11th ACM Multimedia Systems Conference*. [S.l.: s.n.], 2020. p. 345–348. Citado 2 vezes nas páginas 20 e 28.
- RD, A. K.; UDUPA, N. A study of the evolution of video codec and its future research direction. In: IEEE. *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAIECC)*. [S.l.], 2020. p. 1–13. Citado na página 11.
- RD, A. K.; UDUPA, N. A study of the evolution of video codec and its future research direction. In: IEEE. *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAIECC)*. [S.l.], 2020. p. 1–13. Citado na página 30.
- RICHARDSON, I. E. *Video codec design: developing image and video compression systems*. [S.l.]: John Wiley & Sons, 2002. Citado 2 vezes nas páginas 16 e 17.
- RICHARDSON, I. E. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. [S.l.]: John Wiley & Sons, 2004. Citado na página 22.
- SALHI, A. et al. Video codec applications scheduling and optimization based on dag and ggen algorithm. In: IEEE. *2018 2nd European Conference on Electrical Engineering and Computer Science (EECS)*. [S.l.], 2018. p. 236–241. Citado 2 vezes nas páginas 28 e 30.
- SCHMIDT, R. d. O. et al. Towards the next generation of video codecs: The av1 performance and complexity analysis. *IEEE Transactions on Multimedia*, IEEE, v. 21, n. 11, p. 2841–2856, 2019. Citado na página 17.

- STABERNACK, B.; STEINERT, F. Architecture of a low latency h. 264/avc video codec for robust ml based image classification. In: *Workshop on Design and Architectures for Signal and Image Processing (14th edition)*. [S.l.: s.n.], 2021. p. 1–9. Citado na página 13.
- SULLIVAN, G. J.; WIEGAND, T. Video compression—from concepts to the h.264/avc standard. *Proceedings of the IEEE*, IEEE, v. 93, n. 1, p. 18–31, 2005. Citado 2 vezes nas páginas 16 e 17.
- TEKALP, A. M. et al. Introduction to the issue on deep learning for image/video restoration and compression. *IEEE Journal of Selected Topics in Signal Processing*, IEEE, v. 15, n. 2, p. 157–161, 2021. Citado 2 vezes nas páginas 19 e 31.
- TOPIWALA, P.; KRISHNAN, M.; DAI, W. Performance comparison of vvc, av1, and hevc on 8-bit and 10-bit content. *SPIE proceedings: applications of digital image processing XLI*, Society of Photo-Optical Instrumentation Engineers (SPIE), v. 10396, p. 103960L, 2018. Citado na página 24.
- TUNG, T.-Y.; GÜNDÜZ, D. Deep-learning-aided wireless video transmission. In: IEEE. 2022 *IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*. [S.l.], 2022. p. 1–5. Citado na página 31.
- WANG, Y. et al. An efficient deep learning accelerator architecture for compressed video analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, IEEE, v. 41, n. 9, p. 2808–2820, 2021. Citado na página 32.
- ZHANG, Y.; KWONG, S.; WANG, S. Machine learning based video coding optimizations: A survey. *Information Sciences*, Elsevier, v. 506, p. 395–423, 2020. Citado na página 33.
- ZHU, L. et al. Deep learning-based intra mode derivation for versatile video coding. *ACM Transactions on Multimedia Computing, Communications and Applications*, ACM New York, NY, v. 19, n. 2s, p. 1–20, 2023. Citado na página 32.

Anexos

ANEXO A – Algoritmo VMAF em Python

```

1 import os
2 import subprocess
3 import logging
4
5 _copyright_ = "Copyright 2016-2020, Netflix, Inc."
6 _license_ = "BSD+Patent"
7 _version_ = "3.0.0"
8
9 logging.basicConfig()
10 logger = logging.getLogger(os.path.splitext(os.path.basename(_file_))[0])
11 logger.setLevel("INFO")
12
13 try:
14     from matplotlib import pyplot as plt
15 except BaseException:
16     plt = None
17
18 from . import config
19
20 VMAF_PYTHON_ROOT = os.path.dirname(os.path.abspath(_file_))
21 VMAF_ROOT = os.path.abspath(os.path.join(VMAF_PYTHON_ROOT, '..', '..'))
22
23 class ProcessRunner(object):
24     def run(self, cmd, kwargs):
25         try:
26             logger.info(cmd)
27             subprocess.check_output(cmd, stderr=subprocess.STDOUT,
28                                     **kwargs)
29         except subprocess.CalledProcessError as e:
30             raise AssertionError(f'Process returned {e.returncode}, cmd:
31                                   {cmd}, kwargs: {kwargs}, msg: {str(e.output)}')
32
33 def run_process(cmd, **kwargs):
34     process_runner = ProcessRunner()
35     process_runner.run(cmd, kwargs)
36     return 0
37
38 def project_path(relative_path):
39     path = os.path.join(VMAF_ROOT, relative_path)
40     return path
41
42 def required(path):
43     if not os.path.exists(path):

```

```

42     raise AssertionError(f"{path} does not exist, did you build?")
43     return path
44
45 def convert_pixel_format_ffmpeg2vmafexec(ffmpeg_pix_fmt):
46     assert ffmpeg_pix_fmt in ['yuv420p', 'yuv422p', 'yuv444p',
47                               'yuv420p10le', 'yuv422p10le', 'yuv444p10le',
48                               'yuv420p12le', 'yuv422p12le', 'yuv444p12le',
49                               'yuv420p16le', 'yuv422p16le', 'yuv444p16le']
50
51     if ffmpeg_pix_fmt in ['yuv420p', 'yuv420p10le', 'yuv420p12le',
52                           'yuv420p16le']:
53         pixel_format = '420'
54     elif ffmpeg_pix_fmt in ['yuv422p', 'yuv422p10le', 'yuv422p12le',
55                             'yuv422p16le']:
56         pixel_format = '422'
57     elif ffmpeg_pix_fmt in ['yuv444p', 'yuv444p10le', 'yuv444p12le',
58                             'yuv444p16le']:
59         pixel_format = '444'
60     else:
61         assert False
62
63     if ffmpeg_pix_fmt in ['yuv420p', 'yuv422p', 'yuv444p']:
64         bitdepth = 8
65     elif ffmpeg_pix_fmt in ['yuv420p10le', 'yuv422p10le', 'yuv444p10le']:
66         bitdepth = 10
67     elif ffmpeg_pix_fmt in ['yuv420p12le', 'yuv422p12le', 'yuv444p12le']:
68         bitdepth = 12
69     elif ffmpeg_pix_fmt in ['yuv420p16le', 'yuv422p16le', 'yuv444p16le']:
70         bitdepth = 16
71     else:
72         assert False
73
74     return pixel_format, bitdepth
75
76 class ExternalProgram(object):
77     try:
78         from . import externals
79         external_vmaf_feature = config.VmafExternalConfig.vmaf_path()
80         external_vmafexec = config.VmafExternalConfig.vmafexec_path()
81     except ImportError:
82         external_vmaf_feature = None
83         external_vmafexec = None
84
85     vmaf_feature = project_path(os.path.join("libvmaf", "build", "tools",
86                                               "vmaf_feature")) if external_vmaf_feature is None else
87         external_vmaf_feature
88     vmafexec = project_path(os.path.join("libvmaf", "build", "tools",
89                                          "vmaf")) if external_vmafexec is None else external_vmafexec

```

```

83
84 class ExternalProgramCaller(object):
85     @staticmethod
86     def call_vmafexec_single_feature(feature, yuv_type, ref_path,
87                                     dis_path, w, h, log_file_path, logger=None, options=None):
88         options2 = {feature: options.copy() if options is not None else
89                     None}
89         if options2[feature] is not None and 'disable_avx' in
90             options2[feature]:
91             options2['disable_avx'] = options2[feature]['disable_avx']
92             del options2[feature]['disable_avx']
93         if options2[feature] is not None and 'n_threads' in
94             options2[feature]:
95             options2['n_threads'] = options2[feature]['n_threads']
96             del options2[feature]['n_threads']
97         if options2[feature] is not None and '_open_workfile_method' in
98             options2[feature]:
99             options2['_open_workfile_method'] =
100                 options2[feature]['_open_workfile_method']
101             del options2[feature]['_open_workfile_method']
102         if options2[feature] is not None and '_close_workfile_method' in
103             options2[feature]:
104             options2['_close_workfile_method'] =
105                 options2[feature]['_close_workfile_method']
106             del options2[feature]['_close_workfile_method']
107         return ExternalProgramCaller.call_vmafexec_multi_features(
108             [feature], yuv_type, ref_path, dis_path, w, h, log_file_path,
109             logger=logger, options=options2)
110
111     @staticmethod
112     def call_vmafexec_multi_features(features, yuv_type, ref_path,
113                                     dis_path, w, h, log_file_path, logger=None, options=None):
114
115         pixel_format, bitdepth =
116             convert_pixel_format_ffmpeg2vmafexec(yuv_type)
117
118         cmd = [
119             required(ExternalProgram.vmafexec),
120             '--reference', ref_path,
121             '--distorted', dis_path,
122             '--width', str(w),
123             '--height', str(h),
124             '--pixel_format', pixel_format,
125             '--bitdepth', str(bitdepth),
126             '--output', log_file_path,
127             '--xml',
128             '--no_prediction',

```

```

119     ]
120
121     if options is not None and 'disable_avx' in options:
122         assert isinstance(options['disable_avx'], bool)
123         if options['disable_avx'] is True:
124             cmd += ['--cpumask', '-1']
125
126     if options is not None and 'n_threads' in options:
127         assert isinstance(options['n_threads'], int) and
128             options['n_threads'] >= 1
129         cmd += ['--threads', str(options['n_threads'])]
130
131     for feature in features:
132         if options is None:
133             feature_str = feature
134         else:
135             assert isinstance(options, dict)
136             if feature in options and options[feature] is not None and
137                 len(options[feature]) > 0:
138                 assert isinstance(options[feature], dict)
139                 options_lst = []
140                 for k, v in options[feature].items():
141                     if isinstance(v, bool):
142                         v = str(v).lower()
143                     options_lst.append(f'{k}={v}')
144                 options_str = ':'.join(options_lst)
145                 feature_str = '='.join([feature, options_str])
146             else:
147                 feature_str = feature
148         cmd += ['--feature', feature_str]
149
150     if logger:
151         logger.info(' '.join(cmd))
152         run_process(' '.join(cmd), shell=True)
153
154 @staticmethod
155 def call_vifdiff_feature(yuv_type, ref_path, dis_path, w, h,
156     log_file_path, logger=None):
157
158     vifdiff_feature_cmd = "{vmaf} vifdiff {yuv_type} {ref_path}
159         {dis_path} {w} {h} >> {log_file_path}" \
160         .format(
161             vmaf=required(ExternalProgram.vmaf_feature),
162             yuv_type=yuv_type,
163             ref_path=ref_path,
164             dis_path=dis_path,
165             w=w,

```

```
162         h=h,
163         log_file_path=log_file_path,
164     )
165     if logger:
166         logger.info(vifdiff_feature_cmd)
167     run_process(vifdiff_feature_cmd, shell=True)
168
169     @staticmethod
170     def call_vmafexec(reference, distorted, width, height, pixel_format,
171                     bitdepth,
172                     float_psnr, psnr, float_ssim, ssim, float_ms_ssim,
173                     ms_ssim, float_moment,
174                     no_prediction, models, subsample, n_threads,
175                     disable_avx, output, exe, logger,
176                     vif_enhn_gain_limit=None, adm_enhn_gain_limit=None,
177                     motion_force_zero=False):
178
179         if exe is None:
180             exe = required(ExternalProgram.vmafexec)
```

Listing A.1 – Algoritmo VMAF em Python