UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação

Dissertação

MLISP: Esquema de Decisão ISP baseado em Aprendizado de Máquina para Codificadores VVC

Larissa de Ávila Araújo

Pelotas, 2025

Larissa de Ávila Araújo

MLISP: Esquema de Decisão ISP baseado em Aprendizado de Máquina para Codificadores VVC

Dissertação apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Daniel Palomino
Coorientadores: Prof. Dr. Bruno Zatt
Prof. Dr. Guilherme Correa

Pelotas, 2025

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação da Publicação

A656m Araújo, Larissa de Ávila

MLISP [recurso eletrônico] : esquema de decisão ISP baseado em aprendizado de máquina para codificadores VVC / Larissa de Ávila Araújo ; Daniel Palomino, orientador ; Bruno Zatt, Guilherme Corrêa, coorientadores. — Pelotas, 2025.
80 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2025.

1. VVC. 2. Predição intra-quado. 3. Aprendizado de máquina. 4. ISP. I. Palomino, Daniel, orient. II. Zatt, Bruno, coorient. III. Corrêa, Guilherme, coorient. IV. Título.

CDD 005

Elaborada por Simone Godinho Maisonave CRB: 10/1733

Larissa de Ávila Araújo

MLISP: Esquema de Decisão ISP baseado em Aprendizado de Máquina para Codificadores VVC

Dissertação aprovada, como requisito parcial, para obtenção do grau de Mestre em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 8 de maio de 2025

Banca Examinadora:

Prof. Dr. Daniel Palomino (orientador)
Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul

Prof. Dr. Luciano Volcan Agostini
Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul

Prof. Dr. Felipe Martin Sampaio
Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul

Dr. Iago Coelho Storch
Doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul

RESUMO

ARAÚJO, Larissa de Ávila. **MLISP: Esquema de Decisão ISP baseado em Aprendizado de Máquina para Codificadores VVC**. Orientador: Daniel Palomino. 2025. 80 f. Dissertação (Mestrado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2025.

O padrão Versatile Video Coding (VVC) introduz novas ferramentas de codificação para melhorar a eficiência de codificação, incluindo a predição por subpartições intra (*Intra Subpartition* – ISP). Apesar dos ganhos obtidos, o ISP aumenta o esforço computacional, pois adiciona novos modos a serem avaliados na decisão de modo intra-quadro. Diante desse desafio, esta dissertação propõe o MLISP, um esquema baseado em aprendizado de máquina para acelerar essa decisão. O MLISP é composto por duas soluções complementares. A primeira solução utiliza uma árvore de decisão treinada com características da imagem para prever se a avaliação do ISP é necessária em um determinado bloco. Já a segunda solução emprega uma árvore de decisão treinada com características do processo de codificação para determinar, caso o ISP seja avaliado, quais classes de modos intra devem ser consideradas. Os experimentos demonstram que a abordagem proposta reduz o tempo total de codificação em 10,97%, com uma perda de eficiência de codificação de apenas 0,32% segundo a métrica BD-BR. Esses resultados evidenciam a eficácia do MLISP na aceleração da codificação, mantendo um impacto mínimo na qualidade da compressão.

Palavras-chave: vvc; predição intra; isp; aprendizado de máquina.

ABSTRACT

ARAÚJO, Larissa de Ávila. **MLISP: Machine-Learning-based ISP Decision Scheme for VVC Encoders**. Advisor: Daniel Palomino. 2025. 80 f. Dissertation (Masters in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2025.

The Versatile Video Coding (VVC) standard introduces new coding tools to improve compression efficiency, including the Intra Subpartition Prediction (ISP). Despite its benefits, ISP increases computational effort, as it adds new modes to be evaluated during the intra mode decision. To address this challenge, we propose MLISP, a machine learning-based scheme to accelerate this decision. MLISP consists of two complementary solutions. The first solution employs a decision tree trained with image features to predict whether the ISP evaluation is necessary for a given block. The second solution utilizes a decision tree trained with encoding features to determine which intra mode class should be considered if ISP evaluation is required. The experimental results demonstrate that the proposed approach reduces total encoding time by 10.97%, with a coding efficiency loss of only 0.32%. These results highlight the effectiveness of MLISP in accelerating encoding while maintaining a minimal impact on compression quality.

Keywords: vvc; intra prediction; isp; machine learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AIP	Angular Intra Prediction
AI	All Intra
BT	Binary Tree
BD-BR	Bjøntegaard Delta Rate
CTC	Common Test Condition
CTU	Coding Tree Units
CU	Coding Units
HDR	High Dynamic Range
IBC	Intra Block Copy
ISP	Intra Subpartition Prediction
LFNST	Low-Frequency Non-Separable Transform
MIP	Matrix-based Intra Prediction
ML	Machine Learning
MPM	Most Probable Mode
MRL	Multiple Reference Line
MTT	Multi Type Tree
QT	Quadtree
QP	Quantization Parameter
RA	Random Access
RDO	Rate-Distortion Optimization
RFECV	Recursive Feature Elimination with Cross-Validation
RMD	Rough Mode Decision
SAD	Sum of Absolute Differences
SATD	Sum of Absolute Transformed Differences
TT	Ternary Tree
TS	Time Saving

VTM	VVC Test Model
VVC	Versatile Video Coding

SUMÁRIO

1 INTRODUÇÃO

A era digital transformou drasticamente a maneira como conteúdos são consumidos, e os vídeos emergiram como uma forma dominante de comunicação e entretenimento. A grande quantidade de usuários e o aumento exponencial de horas de vídeo transmitidas vem como um destaque da atualidade. Um estudo revela que, durante o terceiro trimestre de 2022, transmissões ao vivo com conteúdo relacionado a jogos acumularam aproximadamente 7,2 bilhões de horas de conteúdo assistido em plataformas líderes (Ceci, 2023). No entanto, esse crescimento não vem sem desafios, especialmente no que diz respeito ao armazenamento e transmissão de vídeos sem compressão. Este dilema impulsiona a necessidade de codificadores de vídeo eficientes, como o *Versatile Video Coding* (VVC), que desempenham um papel crucial na otimização desses processos.

O VVC é um dos padrões de compressão de vídeo mais recentes e avançados, proporcionando uma redução significativa na taxa de bits sem comprometer a qualidade visual, quando comparado a seus predecessores, como o H.264/AVC e o H.265/HEVC (Siqueira; Correa; Grellert, 2020). Isso é alcançado principalmente devido às novas ferramentas de codificação adotadas na predição intra-quadro do VVC, como a *Intra Subpartition* (ISP). O ISP permite uma granularidade ainda maior na subdivisão de blocos de imagem, tratando regiões com características visuais diferentes de maneira mais precisa. No entanto, seu uso em implementações do VVC implica em um aumento considerável do esforço computacional envolvido na decisão do modo intra.

O processo de decisão do modo consiste em determinar o modo de predição intra-quadro a ser usado para cada bloco, avaliando várias combinações possíveis por meio do processo de *Rate-Distortion Optimization* (RDO) (Sullivan; Wiegand, 1998). Contudo, esse processo é computacionalmente exigente, pois há um grande número de modos intra a serem considerados, e para cada um deles, os custos taxa-distorção devem ser calculados após as etapas de codificação, incluindo predição, transformação e quantização. Para lidar com essa complexidade, o *VVC Test Model* (VTM), o software de referência para VVC, implementa o *Rough Mode Decision* (RMD) (Zhao et al., 2011). A principal ideia do RMD é calcular custos rápidos aproximados para

os modos intra e gerar apenas um subconjunto dos modos mais promissores a serem avaliados pelo RDO.

Entretanto, mesmo com o RMD, a introdução de novas ferramentas de predição no VVC ainda aumenta o esforço computacional no processo de decisão do modo intra (Saldanha et al., 2020). À medida que mais ferramentas são incorporadas, o número de combinações a serem avaliadas pelo codificador aumenta significativamente. Portanto, é importante desenvolver soluções capazes de reduzir ainda mais o tamanho do subconjunto de modos intra que são totalmente avaliados pelo processo de decisão do modo intra do VTM.

O aprendizado de máquina é uma das técnicas mais promissoras atualmente para resolver problemas de otimização como a decisão de modo intra no VVC. No processo de decisão do modo intra, existem vários problemas de otimização que podem ser abordados por meio de aprendizado de máquina (ML). Por exemplo, a decisão de modos de ISPs pode ser tratada como um problema de classificação, onde os modos ISP são agrupados em classes e características extraídas do processo de codificação ou da própria imagem são usadas para treinar um modelo de ML. Esse modelo aprende a prever a classe mais provável para um determinado bloco, e com base nessa predição, apenas os modos ISP contidos na classe são avaliados pelo RDO, reduzindo o esforço computacional do processo de decisão do modo intra do VTM.

1.1 Objetivos

O objetivo desta dissertação é desenvolver um esquema baseado em aprendizado de máquina para reduzir o esforço computacional no processo de decisão dos modos ISP dentro da predição intra-quadro no VTM, introduzindo perdas mínimas na eficiência de codificação. O esquema proposto consiste em duas soluções complementares, ambas utilizando árvores de decisão para otimizar a avaliação do ISP. A primeira solução emprega uma árvore de decisão treinada com características da imagem para prever se a avaliação do ISP é necessária em um determinado bloco. A segunda solução utiliza uma árvore de decisão baseada em características do processo de codificação para determinar, caso o ISP seja avaliado, quais classes de modos intra devem ser consideradas.

Para alcançar esse objetivo, esta dissertação estabelece os seguintes objetivos específicos:

- Projetar e implementar uma solução baseada em aprendizado de máquina para acelerar a decisão entre os modos ISP e Não-ISP, utilizando características extraídas da imagem;
- Projetar e implementar uma solução baseada em aprendizado de máquina para

acelerar a decisão dos modos ISP, utilizando características extraídas do processo de codificação;

- Integrar ambas as soluções para obter um método completo para a decisão dos modos ISP.

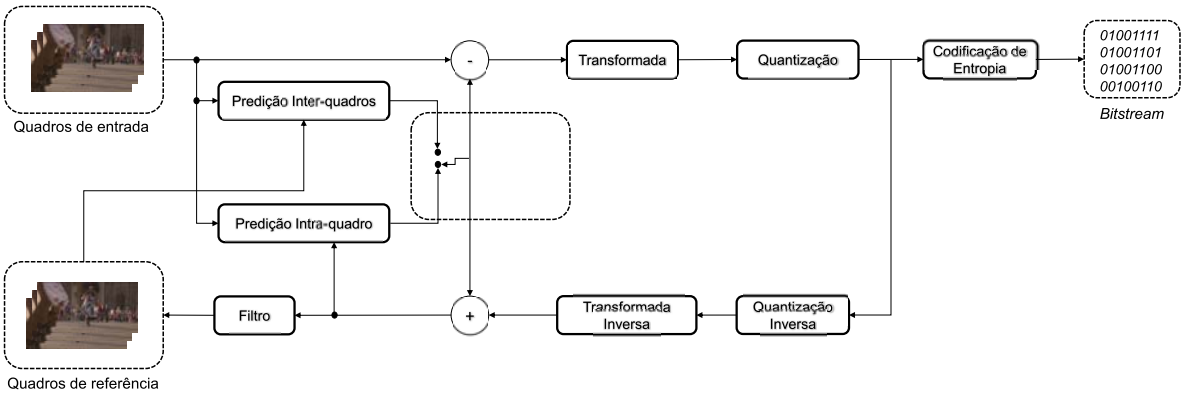
1.2 Considerações Finais

A estrutura deste trabalho está organizada de maneira a abordar progressivamente cada aspecto relevante e está organizada da seguinte forma. No Capítulo 2 será feita uma contextualização mais aprofundada sobre codificação de vídeo e o padrão de codificação VVC. No Capítulo 3 serão apresentados conceitos relevantes sobre Aprendizado de Máquina, utilizado neste trabalho. No Capítulo 4 serão apresentados trabalhos relacionados à pesquisa feita sobre trabalhos que desenvolveram métodos para reduzir o esforço computacional dos modos ISP. No Capítulo 5 é apresentada a metodologia utilizada neste trabalho. O Capítulo 6 descreve as análises realizadas sobre os dados obtidos durante o processo de codificação para alcançar os objetivos propostos. O Capítulo 7 detalha o passo a passo da solução proposta. Já o Capítulo 8 apresenta os resultados obtidos. Por fim, o Capítulo 9 destaca as conclusões sobre o trabalho desenvolvido.

2 CODIFICAÇÃO DE VÍDEO

A codificação de vídeo é um processo essencial nos dias de hoje, dedicado à otimização de entradas de vídeo para que possam ser armazenadas e transmitidas em uma variedade de dispositivos. Sem esta otimização, muitos arquivos seriam excessivamente grandes para armazenamento ou reprodução nos dispositivos dos usuários. Para ilustrar a problemática de manter os vídeos em seu tamanho original, sem passar por um processo de compressão, considere um vídeo de resolução 1920x1080 a 30 quadros por segundo, com três *bytes* por amostra e quinze minutos de duração, o mesmo possuirá um tamanho de aproximadamente 167,96GB. Devido a esta dimensão excessiva, é necessário desenvolver novas ferramentas que reduzam o tamanho dos arquivos de vídeos sem que perdas significativas na qualidade da imagem sejam inseridas.

Figura 1 – Fluxograma de um codificador híbrido



Fonte: Elaborada pelo autor.

Dada a dinâmica dos vídeos, onde diversas imagens são reproduzidas por segundo para dar a ideia de movimento, estes tendem a apresentar redundância espacial e temporal. Essas redundâncias podem ser observadas ao verificar quadros vizinhos e os blocos dentro de um quadro. A redundância espacial refere-se à similaridade entre regiões vizinhas dentro de um mesmo quadro. Essa semelhança permite

prever o conteúdo de um bloco com base nas amostras já reconstruídas ao seu redor. A redundância temporal, por sua vez, decorre da similaridade entre quadros vizinhos, permitindo a exploração de métodos para comunicar mudanças e apontamentos necessários para replicar informações de um quadro para outro, reduzindo a necessidade de duplicar informações. Esses métodos são aplicados na etapa inter-quadros, que procura blocos semelhantes em quadros anteriores, também conhecidos como quadros de referência.

A Figura 1 demostra o fluxograma de um codificador residual híbrido. As etapas se resumem a: **Particionamento**: etapa onde um quadro é dividido em unidades menores, conhecidas como blocos, cada bloco passa pelos processos seguintes com o objetivo de melhorar a eficiência de codificação e a qualidade da imagem. **Predição:** esta etapa tem como objetivo estimar o conteúdo de um bloco com base em informações já disponíveis, explorando redundâncias espaciais e temporais para reduzir a quantidade de dados a serem codificados. Na predição inter-quadro, busca-se blocos semelhantes em quadros anteriores já codificados, permitindo reutilizar informações de regiões parecidas. Já na predição intra-quadro, o conteúdo do bloco é estimado a partir de amostras vizinhas já reconstruídas dentro do mesmo quadro, utilizando técnicas como interpolação ou extrapolação direcional. Dentro dessa etapa, o modo de decisão é responsável por selecionar, entre todos os modos de predição intra e inter disponíveis, aquele que resulta na melhor estimativa para o bloco em questão. A diferença entre o bloco original e o bloco predito é então calculada, originando o resíduo que será processado nas etapas seguintes da codificação. **Transformada:** etapa que transforma os resíduos do domínio espacial para o domínio das frequências, com o objetivo de, na próxima etapa, atenuar informações visuais que são pouco perceptíveis ao sistema visual humano. Para isto, parte-se da premissa que, dentro do domínio das frequências, os coeficientes de alta frequência são pouco perceptíveis podendo assim, serem descartados ou atenuados. **Quantização:** etapa que reduz significativamente o tamanho do arquivo e a precisão da representação. É na quantização que são, de fato, eliminados os detalhes menos perceptíveis ao sistema visual humano e, também, onde existe perda de informação, uma vez que o cálculo feito sobre os dados são divisões inteiras baseadas no parâmetro de quantização (QP) que podem gerar resultados iguais a zero e estes resultados não podem ser revertidos durante a decodificação. **Codificação de Entropia:** é a última etapa antes da transmissão ou armazenamento, que realizará o processo de compressão. Até então todas as etapas serviram como preparação para realizar a compressão do vídeo. Esta etapa se concentra na representação eficiente dos dados, aproveitando padrões de probabilidade e redundâncias estatísticas nos dados para reduzir seu tamanho. Ainda, destaca-se as etapas de **quantização inversa** e **transformada inversa** utilizadas para reconstruir os quadros do vídeo original considerando as perdas de informação do processo para

que possam ser utilizadas como quadros de referência para a codificação dos próximos quadros, isto se faz necessário para que o codificador e o decodificador utilizem exatamente as mesmas referências. Estes quadros passam por uma etapa de **filtro** para atenuar artefatos na imagem que resultam do processo de compressão (VIDEO CODING CONCEPTS, 2010).

2.1 Versatile Video Coding (VVC)

O VVC (Bross et al., 2021) é um dos mais recentes e avançados padrões de compressão de vídeo, lançado em 2020. Ele proporciona uma maior redução na taxa de bits, sem comprometer a qualidade visual, quando comparado a seus predecessores, como o H.264/AVC e o H.265/HEVC (Siqueira; Correa; Grellert, 2020). O VVC é fruto dos esforços do grupo *Joint Video Experts Team* (JVET), formado pela colaboração de outros dois grupos: *Video Coding Experts Group* (ITU-T VCEG) e *Moving Picture Experts Group* (ISO/IEC MPEG).

Este novo padrão incorpora diversas técnicas avançadas, incluindo os novos modos de predição, ferramentas de particionamento aprimoradas, transformadas avançadas, e outras melhorias. O objetivo é otimizar a compressão de vídeo em uma ampla gama de cenários, como vídeos 360°, conteúdo de tela, nuvens de pontos, vídeos em resoluções 4K e 8K, e *High Dynamic Range* (HDR) (Bross et al., 2021). O VVC, portanto, se destaca como uma solução abrangente e sofisticada para atender às demandas crescentes e diversificadas no campo da compressão de vídeo.

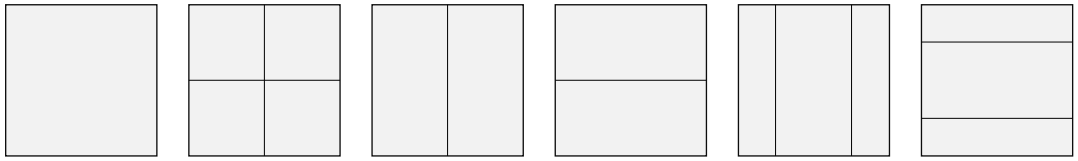
Dentre as diversas inovações trazidas pelo VVC, destaca-se o **modo de particionamento** *Multi Type Tree* (MTT). Como discutido anteriormente, o particionamento desempenha um papel crucial na codificação de vídeo, visando dividir um quadro em blocos para aprimorar a eficiência da codificação. No VVC, cada quadro é subdividido em estruturas denominadas *Coding Tree Units* (CTU), com um tamanho típico de 128×128. Essas CTUs, por sua vez, podem ser subdivididas recursivamente em blocos menores conhecidos como *Coding Units* (CU).

As CUs podem ser divididas utilizando o método *Quadtree* (QT), já presente no padrão HEVC, que efetua a divisão em quatro CUs quadradas menores de tamanho igual. No VVC, foi introduzido o MTT, permitindo a divisão das CUs em formato quadrado ou retangular. Além disso, em conjunto com o QT, é possível utilizar os métodos *Binary Tree* (BT) e *Ternary Tree* (TT), permitindo a divisão tanto de forma vertical quanto horizontal.

Conforme ilustrado na Figura 2, a divisão pode não ocorrer, resultando no processo de codificação utilizando a CU em seu tamanho original. Por outro lado, a divisão pode ocorrer nos formatos QT, BT ou TT. Observa-se que os métodos QT e BT, tanto na horizontal quanto na vertical, dividem as CUs em tamanhos iguais. Já o método TT divide

a CU em três CUs menores, podendo ocorrer tanto horizontal quanto verticalmente. Nesse caso, a CU central adquire 50% do tamanho original, enquanto as duas CUs periféricas obtêm 25% cada da CU original.

Figura 2 – Particionamentos possíveis no VVC considerando blocos quadrados



Fonte: Elaborada pelo autor.

Outras das inovações do VVC estão presentes dentro da **predição intra-quadro**, foco deste trabalho. Esta é uma etapa crucial onde as amostras de um quadro são preditos com base em informações contidas no próprio quadro. Mais especificamente, essa predição utiliza amostras de referência, os quais são as amostras localizadas à esquerda e acima do bloco que está sendo predito. Essa abordagem desempenha um papel fundamental na redução da redundância espacial dentro de um quadro.

Enquanto os modos não-direcionais, Planar e DC, permanecem inalterados desde o HEVC, os modos direcionais, também conhecidos como angulares, foram significativamente expandidos no VVC, totalizando agora 65 modos em comparação com os 33 do padrão HEVC. O modo Planar realiza interpolação entre as amostras de referência, sendo eficaz em áreas da imagem com variação gradual, onde as amostras podem ser aproximados por uma superfície plana.

Por outro lado, o modo DC busca calcular a média entre as amostras de referência, mostrando-se eficaz em áreas da imagem com transições suaves, onde os valores das amostras mantêm uma relação próxima com a média local. Finalmente, os modos angulares desempenham um papel importante em regiões da imagem caracterizadas por bordas ou texturas com orientação direcional. Ao oferecer uma gama mais ampla de ângulos, o VVC consegue representar de forma mais precisa contornos inclinados e estruturas lineares que não são perfeitamente horizontais ou verticais, o que contribui para uma codificação intra-quadro mais eficiente nesses casos.

Além destas ferramentas, outras três ditas não convencionais foram adicionadas na predição intra-quadro do codificador VVC, sendo estas a *Intra Subpartition Prediction* (ISP), *Matrix-based Intra Prediction* (MIP) e *Multiple Reference Line* (MRL). O ISP, foco deste trabalho, é uma ferramenta que trata de dividir um bloco de forma horizontal ou vertical em duas subpartições para blocos de tamanho 4×8 ou 8×4 e quatro subpartições para demais tamanho de bloco exceto 4×4 que não pode ser subdividido.

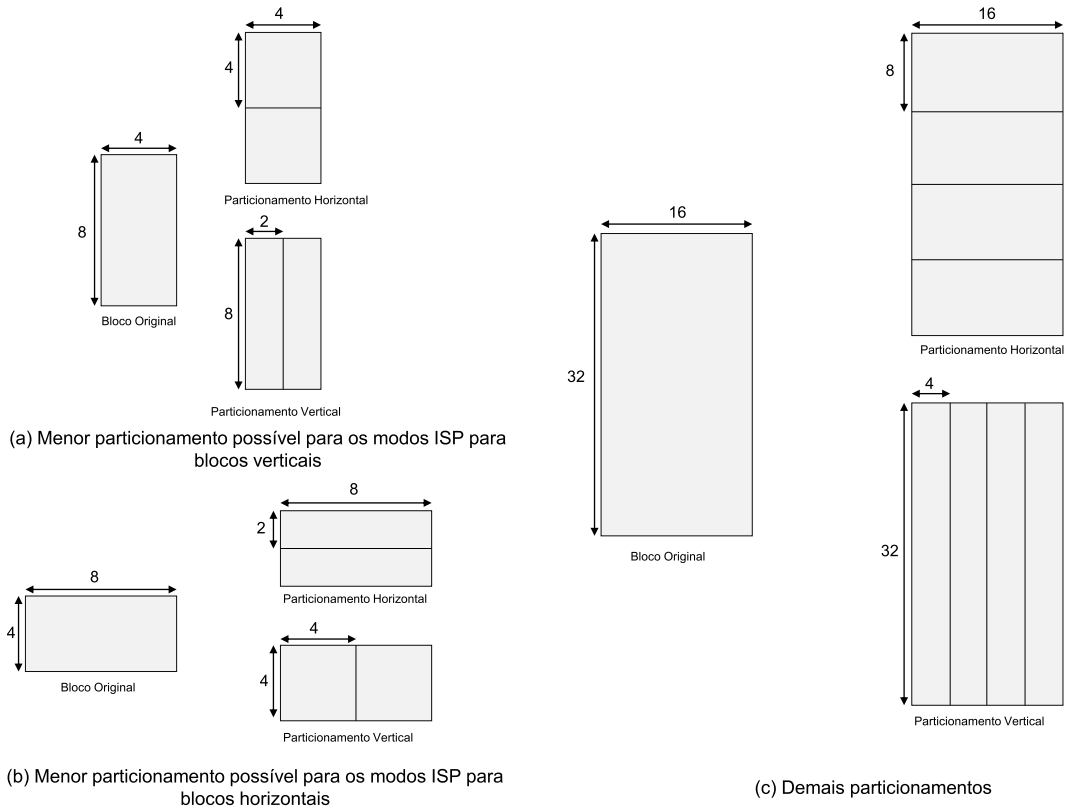
Após, cada subpartição passa pelo processo de predição intra-quadro e após ser reconstruída suas amostras são utilizadas como referência para a predição intra-quadro da próxima subpartição. Ao fim, todas as subpartições devem possuir o mesmo modo de predição intra, que se limitam aos modos convencionais DC, Planar ou Angulares. Esta ferramenta foi proposta por (De-luxán-hernández et al., 2019), onde os autores registraram um ganho na eficiência de codificação de 0,57% e 0,29% nas configuração *All Intra* (AI) e *Random Access* (RA), respectivamente, considerando a métrica BD-BR (*Bjøntegaard Delta Rate*) (Bjontegaard, 2001).

Já o MIP é uma ferramenta que possui uma adequação para blocos aos quais possuem mais de uma direção (Saldanha et al., 2020). Esta ferramenta foi concebida ao treinar redes neurais em uma extensa base de dados, gerando três conjuntos de matrizes, com o objetivo de distinguir-se da linearidade dos modos convencionais. O processo de predição executado pelo MIP, é destacado em três fases distintas: a Etapa de Média, a Etapa de Multiplicação Vetor-Matriz e a Etapa de Interpolação. A Etapa de Média gera um vetor para a linha que contém as amostras de referência do bloco a ser predito e outro vetor para a coluna com as amostras de referência sendo que ambos os vetores são adquiridos através da média das amostras adjacentes. Estes dois vetores são concatenados, condensando em um único vetor que será passado para a próxima etapa. Na Etapa de Multiplicação Vetor-Matriz, uma matriz e um vetor de *offset* são selecionados com base no tamanho do bloco e no modo MIP. O vetor da etapa anterior é multiplicado pela matriz selecionada, e o resultado é somado ao vetor de *offset* correspondente. O resultado desta etapa é um bloco predito sub-amostrado, que será passado para a etapa posterior. Finalmente, na Etapa de Interpolação, as amostras ausentes são preditas por meio de interpolação bilinear entre as amostras de referência e as amostras obtidas após a segunda etapa.

Por fim, o MRL foi proposto por (Chang et al., 2019) e trata-se de uma ferramenta que expande o local onde as amostras de referências podem ser encontradas para gerar a predição das amostras. Onde anteriormente eram buscadas amostras de referência na linha e coluna imediatamente adjacente ao bloco a ser predito, com esta ferramenta é possível fazer busca em uma ou três linhas e colunas anteriores à coluna imediatamente adjacente ao bloco. De acordo com os testes desenvolvidos pelos autores, o MRL tende a aumentar a eficiência de codificação média em 0,46% para a configuração AI e 0,2% para a configuração RA. Além disso, foi observado um aumento na eficiência de codificação de até 1,46% em vídeos com conteúdo de gravação de tela.

Dado que o foco deste trabalho está na ferramenta ISP para predição intra-quadro, nas seguintes Seções serão aprofundados os conceitos e as informações do VVC relacionadas ao tópico.

Figura 3 – Exemplos de subparticionamentos dos ISPs



(a) Menor particionamento possível para os modos ISP para blocos verticais

(b) Menor particionamento possível para os modos ISP para blocos horizontais

(c) Demais particionamentos

Fonte: Elaborada pelo autor.

2.1.1 Intra Subpartition Prediction (ISP)

O ISP é uma ferramenta originada pela evolução da *Line-Based Intra Prediction* (LIP) (De-luxán-hernández et al., 2018), desenvolvida pelos mesmos autores. Seu propósito é dividir blocos de luminância nas direções horizontal ou vertical, contanto que as subpartições resultantes contenham no mínimo 16 amostras. Quando um bloco é dividido horizontalmente, a altura é dividida por K, e quando é dividido verticalmente, a largura é dividida por K, onde K representa o número de subpartições.

O valor de K é inicialmente definido como 4 por padrão. No entanto, para blocos de tamanho 4×8 e 8×4, e apenas para estes tamanhos de bloco, o valor é ajustado para 2. Blocos de tamanho 4×4 não podem ser subparticionados devido à exigência mínima de 16 amostras por subpartição, conforme mencionado anteriormente. A Figura 3 ilustra o processo de subparticionamento.

Cada subpartição é processada de maneira semelhante a um bloco de predição intra-quadro, gerando predições e resíduos. Após a codificação de cada subpartição, as amostras reconstruídas podem ser utilizadas para a subpartição seguinte. Todas as subpartições devem ser codificadas com o mesmo modo de predição, sendo sinalizado apenas uma vez para o bloco. É importante notar que apenas os modos de

predição intra-quadro, como Planar, DC e Angulares, podem ser combinados com os modos ISP.

Utilizando a métrica BD-BR, os resultados relatados pelos autores indicam que o ganho médio em eficiência de codificação é de 0,57% para a configuração AI e 0,27% para a configuração RA. Estes resultados surgiram desconsiderando os vídeos da classe D e F da CTC do VVC (Bossen et al., 2020), onde os vídeos da classe D são aqueles de resolução 416x240 e os vídeos da classe F são *screen content*. Por outro lado, o tempo de codificação é 112% maior para a configuração AI e 102% maior para a configuração RA. Os testes foram desenvolvidos no VTM na versão 3.0.

Uma das principais vantagens da utilização dos modos ISP em relação à fragmentação da árvore de partição está na redução do *overhead* de codificação. O termo *overhead* refere-se à quantidade adicional de dados que precisa ser incluída no *bitstream* apenas para descrever as decisões de codificação, como, por exemplo, a forma como um bloco foi particionado. Ao optar por dividir a árvore, o codificador precisa sinalizar explicitamente cada nova subdivisão, o que pode aumentar significativamente o tamanho do *bitstream*. Já os modos ISP realizam uma subdivisão interna do bloco exclusivamente para fins de predição, sem modificar a estrutura da árvore de partição, evitando assim a sinalização de novas divisões e reduzindo o *overhead*. Além disso, os modos ISP se mostram especialmente eficazes em blocos que apresentam variação direcional acentuada, como bordas ou padrões inclinados, permitindo aplicar predições mais adequadas em subáreas do bloco, o que melhora a qualidade da predição sem a necessidade de dividir o bloco por completo.

2.1.2 Modo de Decisão Intra

Considerando todas as possíveis maneiras de particionar um quadro em blocos e, para cada um desses blocos, as diversas opções de aplicação de modos de predição intra-quadro ou técnicas de predição inter-quadro, fica evidente que, embora um método de busca exaustiva possa oferecer resultados ótimos, ele também acarreta uma complexidade computacional imensa. Por essa razão, implementações do padrão VVC, como o codificador de referência VTM, adotam estratégias que buscam soluções sub-ótimas, avaliando apenas um subconjunto dessas possibilidades. No caso do VTM, por exemplo, é empregado o método RMD para reduzir o custo computacional do processo de seleção dos modos de predição.

No que diz respeito à predição intra-quadro, esses dois métodos se destacam como os principais na busca por determinar os custos dos modos de predição intra-quadro ao serem aplicados a um bloco específico. Esses custos são calculados por meio da Equação (1). Por um lado, o RDO é um processo de otimização que procura encontrar a melhor combinação entre a taxa de bits e a distorção ao calcular os custos para diferentes modos de predição dentro de um bloco e determinar qual é o melhor

entre todos eles. Estes custos só são avaliados no final do processo de codificação, considerando os cálculos após a codificação de entropia, uma vez que o RDO calcula o custo real para cada modo.

Por outro lado, o RMD tem uma abordagem mais simples. Esse processo busca determinar um custo aproximado para cada modo de predição dentro de um bloco específico. Ele utiliza a mesma equação do RDO, mas o cálculo da distorção e taxa de bits é obtido de forma aproximada utilizando apenas as informações residuais. Isso ocorre logo após a aplicação do modo de predição dentro do quadro. O objetivo do RMD é selecionar um conjunto menor de opções para o RDO calcular, já que o RMD é menos complexo e, portanto, mais rápido. Assim, o RDO precisará calcular os custos apenas de um subconjunto de modos.

Na Equação (1), o J se refere ao valor do custo taxa-distorção aplicado ao modo de predição intra-quadro (*mode*) e o QP, o objetivo é encontrar o menor valor de J e, consequentemente, o melhor modo de predição intra-quadro para o bloco que está sendo predito. O QP é responsável por definir a força da quantização, sendo assim, valores mais altos de QP resultam em uma maior perda de qualidade, mas também uma maior taxa de compressão, enquanto que valores menores resultam em uma menor perda de qualidade, porém com uma menor taxa de compressão. D se refere à medida de distorção, esta medida é obtida ao calcular a diferença entre o bloco predito e o bloco original utilizando as fórmulas SAD (*Sum of Absolute Differences*), vide Equação (2), ou SATD (*Sum of Absolute Transformed Differences*), vide Equação (3). O menor valor dentre estas duas fórmulas é o escolhido. O R indica o número de bits utilizado, este valor é relacionado à quantidade de informações que são necessárias para representar o bloco. Isso inclui informações sobre os modos de predição, transformações aplicadas, coeficientes de quantização, entre outros. Já o λ destaca-se por definir a prioridade entre a compressão e a qualidade dos dados. Assim, valores maiores indicam uma importância maior para a compressão enquanto que valores menores indicam uma importância maior para a qualidade dos dados.

$$J(mode|Qp) = D(mode|Qp) + \lambda * R(mode|Qp) \tag{1}$$

$$SAD = \sum_{i=1}^{altura} \sum_{j=1}^{largura} |O(i,j) - P(i,j)| \tag{2}$$

$$SAD = \sum_{i=1}^{altura} \sum_{j=1}^{largura} Hadamard(O(i,j) - P(i,j)) \tag{3}$$

Desta forma, o RMD possui quatro etapas distintas para gerar o subconjunto de modos a serem avaliados pelo RDO, este subconjunto é chamado de RD-List. A

etapa *Angular Intra Prediction* (AIP) busca gerar os custos para os modos Planar, DC e Angulares, adicionando os melhores cálculos na RD-List, ordenando os modos em ordem crescente baseado nos custos gerados nesta etapa. Na sequência, a etapa MRL gera os custos e atualiza a RD-List para cada combinação entre seis modos de predição intra-quadro e o local onde as amostras de referências podem ser obtidas, conhecidos por linhas de referência. Os seis modos de predição são encontrados em uma lista chamada *Most Probable Modes* (MPM) que trata-se de uma lista dos seis modos prováveis derivados a partir dos modos dos blocos vizinhos. Já as linhas de referências utilizadas pelo MRL são apenas duas extras. Por fim, a etapa MIP gera os custos de predição para seus próprios candidatos. Os melhores modos MIP, com base nesses custos, são então comparados com os candidatos já presentes na RD-List, e, caso apresentem melhor desempenho, podem ser adicionados ao conjunto final de modos a serem avaliados na etapa de RDO.

Ainda, os modos ISP são incorporados à RD-List, porém somente ao término do processo de RMD, uma vez que é necessário que as amostras tenham sido reconstruídas de um subparticionamento para o seguinte, o que apenas se torna possível através do processo de RDO. Para isso, uma lista é gerada, contendo os modos de predição mais promissores. Esses candidatos são selecionados dentre os oito melhores modos intra convencionais já presentes na RD-List, excluindo os modos MRL e MIP e adicionando a lista de MPMs. A lista resultante, que compreende até oito modos de predição intra-quadro para os modos ISP horizontais, é então duplicada para os modos ISP verticais, resultando em um acréscimo de até dezesseis novos modos de predição na RD-List relacionados aos modos ISP.

Embora a RD-List represente um subconjunto dos modos intra mais promissores, apenas um deles resultará na melhor eficiência de codificação. Mesmo quando se considera apenas o subconjunto de candidatos ISP presente na RD-List, o processo de RDO ainda precisa avaliar até 16 modos para um único bloco, a fim de determinar o melhor modo ISP. Nesse contexto, surge a necessidade de soluções que visem à redução da quantidade de modos avaliados pelo processo de RDO. Essas soluções devem ser capazes de prever com precisão os modos ISP mais promissores, de forma a diminuir o esforço computacional necessário para a decisão do modo ISP, com perda mínima na eficiência de codificação. Nesse sentido, o uso de técnicas baseadas em aprendizado de máquina se destaca como uma abordagem promissora, pois permite a seleção antecipada dos modos ISP mais prováveis de serem escolhidos, com base em características extraídas do bloco.

3 APRENDIZADO DE MÁQUINA

Aprendizado de máquina, também conhecido como *Machine Learning* em inglês, é um subcampo da inteligência artificial (IA) que se concentra no desenvolvimento de sistemas capazes de aprender e melhorar a partir de experiências passadas sem serem explicitamente programados. Em vez de depender de regras programadas manualmente, os sistemas de aprendizado de máquina usam algoritmos que analisam dados para identificar padrões e tomar decisões com o mínimo de intervenção humana.

O treinamento em aprendizado de máquina pode ser geralmente dividido em três categorias principais, dependendo da natureza dos dados e do tipo de aprendizado desejado. Essas categorias são o aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.

No aprendizado supervisionado os algoritmos são treinados em um conjunto de dados rotulados, ou seja, onde cada exemplo de treinamento consiste em entradas e suas saídas correspondentes, também chamados de rótulos. O algoritmo aprende a mapear as entradas para as saídas, permitindo prever a saída para novos dados. Já no aprendizado não supervisionado os algoritmos são alimentados com conjuntos de dados não rotulados e devem encontrar padrões ou estruturas neles. Isso é útil para descobrir *insights* em grandes volumes de dados, identificar padrões ocultos ou realizar tarefas como clusterização. Por fim, no aprendizado por reforço os algoritmos aprendem por meio de tentativa e erro, recebendo *feedback* na forma de recompensas ou penalidades com base em suas ações. Desta forma, o algoritmo aprende a alcançar um objetivo otimizando suas ações com base nesse *feedback* (Tan; Steinbach; Kumar, 2006).

O aprendizado de máquina é amplamente utilizado em uma variedade de aplicações, como reconhecimento de padrões, reconhecimento de fala, detecção de fraudes, recomendação de produtos, diagnóstico médico, veículos autônomos e muito mais. Ele tem o potencial de automatizar tarefas complexas e melhorar a eficiência em muitas áreas (Tan; Steinbach; Kumar, 2006).

Neste estudo, o foco será no aprendizado de máquina supervisionado, uma vez

que esta categoria é a mais adequada para abordar o problema da tomada de decisões nos modos do VVC.

3.1 Aprendizado de Máquina Supervisionado

O aprendizado de máquina supervisionado é um paradigma no qual um algoritmo é treinado em um conjunto de dados rotulado. Esses dados rotulados consistem em pares de entrada e saída esperada. O objetivo do aprendizado supervisionado é fazer com que o algoritmo aprenda a relação entre as entradas e as saídas, de modo que, quando apresentado a novas entradas não vistas durante o treinamento, ele seja capaz de fazer previsões ou classificações precisas.

Este paradigma segue uma sequência de etapas, sendo estas: **I. Conjunto de Dados Rotulado**: Durante a fase de treinamento, o algoritmo é alimentado com um conjunto de dados que contém exemplos de entradas, juntamente com as saídas correspondentes. **II. Treinamento do Modelo**: O algoritmo usa esses dados rotulados para aprender a mapear as entradas para as saídas. O objetivo é otimizar os parâmetros internos do modelo para que ele possa fazer previsões precisas. **III. Validação e Teste**: Após o treinamento, o modelo é frequentemente validado usando um conjunto de dados separado para garantir que ele generalizou bem e não está apenas memorizando os exemplos de treinamento. Se necessário, os parâmetros do modelo podem ser ajustados para melhorar o desempenho. **IV. Teste com Dados Novos**: Finalmente, o modelo treinado é testado com um conjunto de dados que não foi usado durante o treinamento ou validação. Isso ajuda a avaliar a capacidade do modelo de fazer previsões precisas em novos dados.

O aprendizado supervisionado é frequentemente subdividido em dois tipos principais de problemas: classificação e regressão. Essas duas categorias representam maneiras distintas de abordar diferentes tipos de tarefas analíticas e preditivas.

A classificação é um tipo de problema de aprendizado supervisionado em que o objetivo é separar dados de teste em categorias específicas. O algoritmo é treinado em um conjunto de dados rotulado, onde cada exemplo possui uma entrada e uma saída correspondente. Durante o treinamento, o modelo aprende a relação entre as características das entradas e as categorias às quais pertencem. Posteriormente, ao lidar com novos dados não vistos, o modelo faz previsões sobre a categoria a que esses dados pertencem.

A regressão é usada quando o objetivo é entender a relação entre variáveis dependentes e independentes. Diferentemente da classificação, onde o foco está na atribuição de rótulos de categoria, a regressão busca prever valores numéricos ou contínuos. Ela é comumente usada para fazer projeções e estimativas. Alguns dos algoritmos de regressão incluem Regressão Linear, Regressão Logística e Regressão

Polinomial.

3.2 Classificação

O problema da classificação é uma tarefa comum em aprendizado de máquina, onde o objetivo é atribuir uma categoria ou classe a uma instância de dados com base em suas características. Em outras palavras, o modelo de classificação aprende a mapear as entradas para rótulos predefinidos, permitindo prever a classe de novos dados não vistos. Este é um dos tipos mais fundamentais de problemas em aprendizado de máquina supervisionado.

Em problemas de classificação, as classes e características desempenham papéis essenciais. As classes representam as categorias às quais o modelo atribui instâncias de dados, enquanto as características são informações individuais que descrevem cada instância. Por exemplo, em uma tarefa de classificação de imagens de animais, as classes podem ser "gato", "cachorro"e "pássaro", enquanto as características podem incluir informações como a presença de pelos, orelhas, etc, na imagem.

A seleção apropriada de características é crítica para o desempenho do modelo. Identificar quais variáveis são mais informativas e realizar engenharia de características são práticas comuns. Durante o treinamento, o modelo aprende a relação entre as características e as classes no conjunto de dados de treinamento, buscando generalizar para novos dados durante a fase de teste.

As classes e características precisam ser representadas numericamente para facilitar o treinamento do modelo. Geralmente, as classes recebem representações numéricas distintas, enquanto técnicas como codificação de variáveis categóricas e normalização são aplicadas às características.

O desempenho do modelo é avaliado com base em sua capacidade de atribuir corretamente as instâncias às classes. Métricas como precisão, *recall* e *F1-score* (Tan; Steinbach; Kumar, 2006) são comumente utilizadas para avaliação. A relação dinâmica entre classes e características destaca a importância de uma abordagem flexível, adaptando a seleção de características de acordo com a natureza específica da tarefa de classificação.

O balanceamento de conjuntos de dados é um ponto crítico ao lidar com problemas de aprendizado de máquina, especialmente em tarefas de classificação. Em muitos casos, os conjuntos de dados podem ter classes desproporcionalmente representadas, o que pode levar a um viés do modelo em direção à classe majoritária. Estratégias como *oversampling* que busca aumentar a quantidade de exemplos da classe minoritária ou *undersampling* que busca reduzir a quantidade de exemplos da classe majoritária podem ser empregadas para equilibrar as classes, melhorando assim a capacidade do modelo de generalizar para ambas as classes.

Para que a classificação possa ocorrer é essencial que o conjunto de dados passe por um treinamento com um algoritmo de aprendizado de máquina, bem como validações e testes. Desta forma, a divisão adequada dos dados é crucial para avaliar e otimizar o desempenho do modelo. O conjunto de treinamento é utilizado para treinar o modelo, enquanto o conjunto de validação é empregado para ajustar hiperparâmetros e evitar *overfitting* — fenômeno no qual o modelo aprende excessivamente os detalhes e ruídos dos dados de treinamento, resultando em desempenho insatisfatório em dados novos. O conjunto de teste, por sua vez, é reservado para avaliar o desempenho do modelo final em dados não usados durante o treinamento e validação.

O *Random Search* e o *Grid Search* são técnicas de busca de hiperparâmetros usadas para otimizar o desempenho do modelo (Pedregosa et al., 2011). O *Grid Search* explora todas as combinações possíveis de hiperparâmetros em uma grade predefinida, sendo uma abordagem mais exaustiva. Por outro lado, o *Random Search* seleciona aleatoriamente conjuntos de hiperparâmetros, oferecendo uma abordagem mais eficiente em termos computacionais. Hiperparâmetros são parâmetros externos ao modelo que controlam seu comportamento durante o treinamento, como a profundidade de uma árvore ou a taxa de aprendizado, e precisam ser definidos antes do processo de aprendizado. A escolha entre essas técnicas depende da disponibilidade de recursos computacionais e da complexidade do espaço de hiperparâmetros. Além disso é possível utilizar estas técnicas em combinação de forma a encontrar um conjunto sub-ótimo de hiperparâmetros para o modelo em questão.

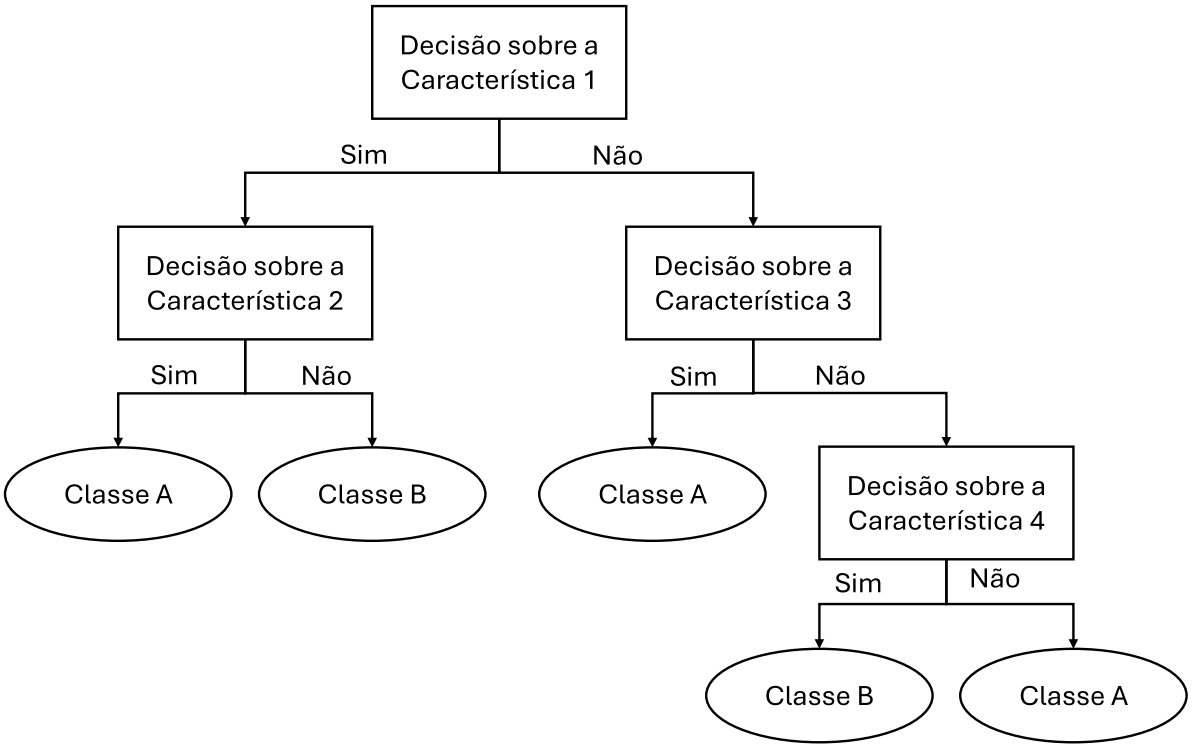
A combinação eficaz desses conceitos é crucial para desenvolver modelos de aprendizado de máquina robustos. O balanceamento adequado das classes pode melhorar a capacidade do modelo de generalizar para ambas as classes. Uma divisão apropriada entre conjuntos de treinamento, validação e teste é essencial para avaliar e ajustar o modelo de maneira adequada. A busca eficiente de hiperparâmetros, seja por meio de *Random Search* ou *Grid Search*, permite encontrar configurações que otimizam o desempenho do modelo. Em última análise, um processo bem planejado que leve em consideração esses fatores contribui para o desenvolvimento de modelos mais precisos e generalizáveis.

3.2.1 Árvores de Decisão

Árvores de decisão são modelos de aprendizado de máquina que utilizam uma estrutura em forma de árvore para representar regras de decisão e possíveis resultados. Essas árvores são construídas durante o processo de treinamento com base em dados rotulados e são usadas para tomar decisões ou fazer previsões (Russell; Norvig, 2009). As árvores de decisão pertencem a uma família de algoritmos e são utilizados como base de algoritmos como *Random Forest* e *Light Gradient Boosting Machine*.

A Figura 4 ilustra a estrutura de uma árvore de decisão. Essa estrutura é composta

Figura 4 – Exemplo de uma Árvore de Decisão



Fonte: Elaborada pelo autor.

em nós e arestas onde os nós podem ser tanto internos quanto folhas. Os nós internos, também chamados de nós de decisão, representam uma decisão com base em alguma característica específica presente dentro do conjunto de dados, enquanto os nós folhas, também chamados de nós de saída, representam os resultados ou classificações finais. Cada folha está associada a uma classe ou valor de saída. Já as arestas conectam os nós e representam as possíveis respostas à decisão tomada em um nó anterior.

O processo de construção de uma árvore de decisão envolve dividir recursivamente os dados com base nas características que melhor separam as classes ou valores de saída. Essa divisão é feita de maneira a maximizar a pureza dos grupos resultantes. Em outras palavras, o objetivo é criar ramos da árvore onde os exemplos dentro de cada ramo sejam mais homogêneos em relação à classe de destino.

Ao fazer uma predição ou tomar uma decisão com uma árvore de decisão, os dados percorrem a árvore, seguindo os ramos de acordo com as características relevantes, até atingirem uma folha que fornece a predição ou classificação final.

As árvores de decisão são atraentes porque são interpretáveis e podem lidar com dados categóricos e numéricos. No entanto, é importante considerar a possibilidade de *overfitting* durante o treinamento, o que pode ser mitigado usando técnicas como a

poda da árvore. Além disso, em alguns casos, uma única árvore de decisão pode não ser suficiente, levando à utilização de técnicas mais avançadas, como *Random Forest* ou *Gradient Boosting*, que combinam várias árvores para obter um desempenho mais robusto.

A construção de uma árvore de decisão pode ser ajustada por meio de diversos hiperparâmetros, que influenciam diretamente sua profundidade, a forma como os nós são divididos e a regularização do modelo. A escolha adequada desses hiperparâmetros impacta a capacidade da árvore de aprender padrões nos dados sem incorrer em *overfitting* — quando o modelo aprende excessivamente os detalhes e ruídos do conjunto de treinamento, perdendo capacidade de generalização — ou *underfitting* — quando o modelo é simples demais para capturar os padrões relevantes nos dados.

Um dos principais hiperparâmetros é o *criterion*, que define a métrica utilizada para avaliar a qualidade das divisões nos nós internos. Entre as opções mais comuns, destaca-se o índice de Gini, que mede a impureza dos grupos resultantes, favorecendo divisões que maximizam a homogeneidade das classes. Outra alternativa amplamente utilizada é a entropia, que baseia a divisão na teoria da informação, buscando minimizar a quantidade de incerteza nas ramificações. Em geral, o índice de Gini é mais eficiente computacionalmente, enquanto a entropia pode produzir divisões ligeiramente mais precisas, embora a diferença prática entre ambas seja frequentemente pequena.

O hiperparâmetro *max depth* é outro fator essencial e indica a profundidade da árvore. Árvores muito profundas podem memorizar os dados de treinamento, levando ao *overfitting*, enquanto árvores excessivamente rasas podem não capturar relações relevantes entre as variáveis, resultando em *underfitting*. O valor ideal varia conforme o conjunto de dados, sendo comum que esse hiperparâmetro seja ajustado por meio de validação cruzada. Quando não especificado, a árvore cresce até que todas as folhas sejam puras ou contenham um número mínimo de amostras.

Relacionado à profundidade, o hiperparâmetro *min samples split* controla a menor quantidade de amostras necessária para que um nó possa ser dividido. Um valor muito baixo permite que a árvore se aprofunde excessivamente, aumentando o risco de ajuste excessivo aos dados. Já valores mais altos limitam a fragmentação dos dados, forçando a árvore a encontrar padrões mais gerais. O valor padrão geralmente é 2, mas aumentar esse número pode ser benéfico para conjuntos de dados ruidosos.

Outro hiperparâmetro relevante é o *min samples leaf*, que define o menor número de amostras permitido em um nó final. Esse parâmetro atua como uma forma de regularização, prevenindo divisões que resultem em folhas com pouquíssimos exemplos. Valores mais altos levam a árvores mais simples e generalizáveis, enquanto valores baixos permitem maior granularidade nas predições.

A seleção de variáveis utilizadas em cada divisão é regulada pelo hiperparâmetro

max features. Esse parâmetro determina quantas das variáveis disponíveis serão testadas ao definir o critério de separação em um nó. Quando todas as variáveis são consideradas, o modelo pode sofrer de *overfitting*. Alternativamente, restringir o número de variáveis testadas em cada divisão pode aumentar a diversidade das árvores em modelos como *Random Forest*, melhorando a generalização.

Outro fator importante é o *max leaf nodes*, que impõe um limite na quantidade de folhas na árvore. Esse parâmetro pode ser útil para impedir o crescimento excessivo da árvore, reduzindo a complexidade do modelo sem a necessidade de definir manualmente uma profundidade máxima.

A escolha dos hiperparâmetros ideais depende do contexto e do conjunto de dados analisado. Modelos mais profundos e com poucas restrições podem capturar padrões complexos, mas correm o risco de *overfitting*, enquanto modelos muito restritos podem ser incapazes de aprender relações significativas. Para encontrar a melhor configuração, técnicas como validação cruzada, *Grid Search* e *Random Search* são amplamente utilizadas.

Os hiperparâmetros abordados seguem a nomenclatura e os comportamentos padrão da biblioteca Scikit-learn, uma das mais utilizadas em aprendizado de máquina em Python (Pedregosa et al., 2011).

3.2.2 Métricas de Classificação

As métricas de classificação são utilizadas para apresentação dos resultados de predição e servem para exprimir a performance do modelo. Na Tabela 1 é mostrada a matriz de confusão binária que se usa na classificação de dados para visualizar o quão bom o modelo proposto está. Nesta Tabela identifica-se que os casos considerados Verdadeiros Positivos (VP) são aqueles dados que foram classificados corretamente como positivos, já os Verdadeiros Negativos (VN) se tratam dos dados classificados corretamente como negativos. Os Falsos Positivos (FP) são os dados classificados como positivos quando deveriam ser classificados como negativos e, por fim, os Falsos Negativos (FN) são aqueles dados classificados como negativo quando deveriam ser classificados como positivos.

Tabela 1 – Matriz de Confusão

	Classificado como Positivo	Classificado como Negativo
Rotulado como Positivo	Verdadeiro Positivo	Falso Negativo
Rotulado como Negativo	Falso Positivo	Verdadeiro Negativo

Fonte: Elaborada pelo autor.

- **Acurácia**: A acurácia é a métrica de avaliação mais intuitiva dentre todas. Nesta, é feito o cálculo da proporção das observações corretamente classificadas utilizando a Equação (4). Ainda que seja uma ótima métrica, ela só dará a acurácia correta caso o número de observações classificadas como falso positivo e falso negativo forem praticamente iguais.

$$acuracia = \frac{VP + VN}{VP + VN + FP + FN} \tag{4}$$

- **Precisão**: A precisão é a proporção das observações corretamente classificadas como positivas em relação ao total de observações classificadas como positivas. Esta métrica mede a corretude do modelo pelo quanto de observações classificadas como positivas são realmente positivas. O cálculo é feito de acordo com a Equação (5).

$$precisao = \frac{VP}{VP + FP} \tag{5}$$

- ***Recall***: Também conhecido como sensibilidade, esta métrica indica a proporção das observações corretamente classificadas como positivas em relação ao número de obervações pertencentes a classe positivo. Pode ser calculado através da fórmula em 6.

$$sensibilidade = \frac{VP}{VP + FN} \tag{6}$$

- ***F-Measure***: Esta métrica, também conhecida como F1 Score, é a soma dos pesos da precisão e sensibilidade. Ela leva em consideração tanto os falsos positivos quanto os falsos negativos e por isso é considerada melhor que a acurácia. Pode ser calculada utilizando a Equação (7).

$$F1\_Score = \frac{2 * precisao * sensibilidade}{precisao + sensibilidade} \tag{7}$$

3.3 Aprendizado de Máquina Supervisionado e o Problema da Decisão de Modo na Predição Intra-Quadro

O aprendizado de máquina supervisionado pode ser empregado para otimizar o processo de decisão na predição intra-quadro do VTM. Isso se deve ao fato de que, para cada bloco a ser codificado, o codificador precisa escolher um modo de predição entre um conjunto de possibilidades. Esse processo de seleção impacta diretamente a eficiência da compressão, pois afeta tanto a qualidade da reconstrução do vídeo quanto o bitrate e o custo computacional do processo de codificação.

Uma das abordagens possíveis é a formulação do problema como uma tarefa de classificação, na qual o modelo aprende, a partir de dados rotulados, a identificar quais modos de predição são mais prováveis para um determinado bloco. Essa estratégia pode ser aplicada, por exemplo, à decisão de utilização dos modos ISP, onde o objetivo é determinar se um bloco específico deve ser predito com ISP ou se o uso de outro modo seria mais adequado. Nesse contexto, cada bloco de vídeo pode ser representado por um conjunto de características extraídas durante o processo de codificação e, com base nessas informações, um classificador supervisionado pode ser treinado para prever se o modo ISP deve ser utilizado ou não.

A decisão tomada pelo classificador pode ser utilizada para reduzir a complexidade computacional da codificação, pois permite que o codificador evite testar determinados modos quando sua escolha não for vantajosa. Essa abordagem pode ser estendida a outras etapas do processo de codificação, contribuindo para um melhor equilíbrio entre eficiência computacional e eficiência de codificação.

Além disso, a aplicação de aprendizado de máquina na predição intra-quadro do VTM não se limita apenas à escolha entre ISP e não-ISP. Modelos preditivos também podem ser explorados em outros aspectos da codificação, como a seleção direta de um modo específico dentro do conjunto de modos intra disponíveis, ou até mesmo a definição de estratégias de refinamento para reduzir a necessidade de cálculos exaustivos. Essas estratégias refletem o potencial do aprendizado de máquina para aprimorar os métodos tradicionais de decisão no VTM, possibilitando ganhos significativos em eficiência sem comprometer a qualidade da reconstrução do vídeo.

4 TRABALHOS RELACIONADOS

Este Capítulo se dedica a explorar as contribuições passadas e o estado atual do conhecimento relacionado aos modos ISP no software de referência VTM. A pesquisa foi realizada na base de dados da IEEE para identificar trabalhos que tenham desenvolvido métodos para reduzir o esforço computacional no processo de decisão dos modos ISP na predição intra-quadro do VTM.

Para cada trabalho, foram detalhados seus objetivos, metodologia e resultados. Na Seção de metodologia, foram destacadas as análises estatísticas que orientaram a escolha dos dados relevantes para atender aos objetivos do estudo. Nos resultados finais, foram apresentadas as reduções alcançadas no tempo de codificação e as perdas na eficiência de codificação. Foi utilizada a métrica TS (*Time Saving*) para medir a redução do tempo de codificação. Essa métrica compara o tempo de codificação antes e depois da implementação da solução proposta no software de referência VTM (Bossen; Suehring; Li, 2018).

Para quantificar a eficiência de codificação, empregamos a métrica BD-BR. Essa métrica calcula a variação na taxa de bits para uma mesma qualidade entre o codificador de referência e o codificador modificado com a solução proposta. Valores positivos indicam um aumento na taxa de bits para uma qualidade visual equivalente, sugerindo uma perda de eficiência na codificação com a implementação da solução proposta.

4.1 Descrição dos Trabalhos

- **(Park; Kim; Jeon, 2020):** os autores deste estudo exploram métodos para reduzir a complexidade dos modos ISP no VTM, propondo o ISP-PPC, uma abordagem que agiliza a busca e simplifica a otimização do RDO. Essa técnica compõe uma lista restrita para a RD-List, eliminando previamente os candidatos menos promissores e, consequentemente, diminuindo o esforço computacional.

  Partindo da hipótese de que a similaridade entre a direção da predição e a orientação da subpartição resulta em resíduos parecidos, foi definida uma faixa pré-determinada para os splits horizontal e vertical para determinar quais modos

de predição intra-quadro podem ser descartados antecipadamente para reduzir a complexidade computacional do VTM. A ideia principal é identificar os chamados intervalos pré-podáveis (*pre-prunable ranges*), ou seja, conjuntos de modos ISP que provavelmente não contribuirão para uma melhor codificação e, por isso, podem ser ignorados sem perda significativa de eficiência de codificação.

Para isso, os autores consideram a relação entre uma amostra dentro do bloco atual e sua respectiva amostra de referência. Cada modo de predição intra-quadro tem uma direção específica, que indica o ângulo da predição em relação à vertical. Quanto maior esse ângulo, mais inclinada será a direção da predição dentro do bloco. Para simplificar os cálculos, a posição da amostra de referência é determinada utilizando operações de *bit-shift*, que são computacionalmente mais eficientes do que multiplicações e divisões convencionais.

Quando um bloco é dividido verticalmente no processo de codificação, ele gera subpartições menores. Essas subpartições possuem uma diagonal principal, cuja inclinação depende diretamente da proporção entre sua largura e altura. Se a direção da predição intra-quadro for muito próxima à inclinação da diagonal da subpartição, significa que a predição estará praticamente alinhada com a estrutura do bloco, tornando alguns modos ISP menos prováveis.

A principal observação feita pelos autores é que, nesses casos, as amostras localizadas acima da diagonal principal da subpartição não utilizam novas amostras de referência reconstruídas. Isso indica que certos modos ISP podem ser eliminados antecipadamente sem comprometer a qualidade da predição. Dessa forma, ao analisar a relação entre a inclinação da diagonal do bloco e o ângulo de predição intra, é possível definir um critério para descartar modos desnecessários antes mesmo de realizar cálculos mais complexos.

A implementação foi realizada sobre o VTM 9.0, utilizando as configurações All-Intra com QPs de 22, 27, 32 e 37 para 18 sequências de vídeos que são condições comuns de teste do VVC. Nos testes experimentais, a abordagem obteve uma redução de 12% no esforço computacional, com uma perda de eficiência de codificação de apenas 0,4%.

- **(Liu et al., 2021):** este trabalho propõe um algoritmo de decisão rápida para os modos ISP, fundamentado na complexidade da textura do bloco. A abordagem parte da hipótese de que blocos com texturas simples tendem a não selecionar os modos ISP. Além disso, apresenta-se um método para calcular a complexidade do bloco por meio de um intervalo amostral, otimizando a decisão com base nos custos de distorção.

  A solução adota o desvio médio absoluto como métrica para mensurar a com-

plexidade da textura, porém, em vez de considerar todas as amostras, utiliza-se apenas um subconjunto. Esse subconjunto é formado pelas amostras ímpares de linhas ímpares e pelas amostras pares de linhas pares, o que reduz pela metade o custo computacional sem comprometer a precisão na estimativa da complexidade.

Os resultados das análises indicam que, quando o valor da métrica ultrapassa 20, a probabilidade de seleção dos modos ISP aumenta. Assim, definiu-se um limiar: se o valor calculado for inferior a 30, os modos ISP são previamente desconsiderados.

As implementações foram realizadas utilizando o software de referência VTM 8.0, com a configuração AI e valores de QP de 22, 27, 32 e 37. Em testes com 20 sequências de vídeo do HEVC, os resultados evidenciaram uma redução de 7% no tempo de codificação, acompanhado de uma perda de apenas 0,09% no BD-BR.

- **(Saldanha et al., 2021):** Este trabalho propõe uma solução abrangente para redução de complexidade na predição dos modos intra utilizando aprendizado de máquina e análises estatísticas. A proposta consiste em três soluções distintas que são integradas em uma abordagem unificada. A primeira solução emprega um classificador baseado em árvores de decisão para os modos Planar/DC, a segunda solução utiliza um classificador semelhante para os modos MIP. A terceira solução utiliza o cálculo da variância do bloco para os modos ISP.

  A primeira solução, que avalia a probabilidade de seleção dos modos Planar ou DC e, consequentemente, exclui os modos angulares da RD-List para evitar avaliações desnecessárias, emprega um classificador de árvore de decisão. Esse classificador utiliza dados extraídos do codificador, como a presença dos modos de textura suave no topo da RD-List e a quantidade de modos angulares na RD-List. Esses dados são significativos, pois a presença dos modos suaves, Planar e DC, no topo indica uma maior probabilidade de serem escolhidos como o melhor modo. Além disso, a menor quantidade de modos angulares na lista também sugere que Planar e DC têm uma maior chance de serem os modos preferenciais.

  A segunda abordagem, que determina quando os modos MIP podem ser evitados e removidos da RD-List, também emprega um classificador de árvore de decisão, seguindo a mesma lógica da primeira solução. Este classificador utiliza dados provenientes do codificador, como o número de modos MIP na RD-List e o custo de SATD do primeiro modo convencional dividido pelo primeiro MIP na RD-List. Esses dados apresentam uma correlação significativa com a decisão

de escolher ou não os modos MIP como o melhor modo. Quanto menor o número de modos MIPs na RD-List, menor a probabilidade de um modo MIP ser selecionado. Além disso, a razão entre o custo do SATD do primeiro modo convencional e o primeiro MIP da lista indica uma probabilidade maior de escolha do modo MIP se o resultado desse cálculo for maior que 1.

Já a terceira solução analisa a variância do bloco para decidir quando os modos ISP podem ser removidos da RD-List. O limiar de variância do bloco que indica quando um modo ISP pode ser removido é calculado dinamicamente, uma vez que esse limiar tende a variar conforme o tipo de vídeo e o parâmetro de quantização, conforme apontado pelos autores. Os valores de variância dos blocos do primeiro quadro, que não selecionaram o ISP como o melhor modo, são armazenados, e a média desses valores é calculada. Esse resultado é utilizado como limiar nos quadros subsequentes, sendo ajustado conforme necessário para garantir uma adaptação contínua às características do vídeo. Essa abordagem dinâmica busca otimizar a precisão na decisão de remover os modos ISP ao longo da sequência de vídeo.

Os resultados experimentais foram feitos utilizando o software de referência VTM 10.0 utilizando a CTC do VVC e o modo de configuração AI para cada um dos QPs 22, 27, 32 e 37. Os resultados demonstram que a solução dos autores trouxe uma redução de 18,32% em tempo de codificação com perda no BD-BR de 0,60%. De forma mais detalhada, a solução com as árvores de decisão tanto Planar/DC quanto modos MIP obteve um resultado de 10,47% de redução no tempo de codificação com uma perda no BD-BR de 0,29%. Já para a solução dos ISP foi obtida uma redução no tempo de codificação de 8,32% com perda de BD-BR de 0,31%.

- **(Dong et al., 2022):** Este trabalho apresenta um algoritmo de decisão rápida das novas ferramentas na predição intra-quadro do VVC, abrangendo especificamente as ferramentas ISP, IBC e MRL. O algoritmo desenvolvido pelos autores consiste em duas etapas. A primeira etapa busca remover modos tanto das ferramentas IBC e ISP quanto dos modos de predição intra-quadro normais. Enquanto a segunda busca reduzir a complexidade da codificação eliminando predições desnecessárias das profundidades restantes baseando-se no modo selecionado pelo codificador.

  Os autores demostraram que o modo IBC tende a ser mais utilizado em regiões de textura irregular e complexa, enquanto outros modos lineares são usados em texturas mais simples e direcionais. Já o ISP tende a ser utilizado em regiões cuja textura possui bordas, além de que os valores médios entre as subpartições tendem a ser perceptíveis.

A solução proposta pelos autores utiliza o algoritmo de aprendizado de máquina chamado Árvores de Decisão para a etapa de remover os modos das ferramentas IBC e ISP. Esta etapa envolve utilizar uma árvore de decisão que indica se o IBC deve ser evitado ou não e outra árvore de decisão que indica se os modos ISP devem ser evitados ou não.

A árvore de decisão relacionada às decisões do IBC recebe seis informações, tais quais a consistência do histograma de tons de cinza, número de amostras de alto gradiente e informação de contexto. Outros dados tais quais custo do RDO, resíduos e bits necessários para codificação também são passados para o classificador para avaliar se os modos lineares podem ser removidos. Por outro lado, a árvore de decisão relacionada às decisões do ISP utiliza a força de correlação e características de textura para determinar quando o ISP poderá ser evitado. Características como a distribuição de valores de amostras e o valor médio da amostra são consideradas. Análises feitas pelos autores que indicavam o ganho de informação foram feitas e indicaram que estes dados eram relevantes.

Ainda, uma terceira árvore de decisão é treinada para eliminar predições desnecessárias que venham a ocorrer após as previsões anteriores. Este classificador irá receber informações tais quais informações da textura e informações da codificação. Dentro das informações da textura são passadas informações como o cálculo da magnitude máxima do gradiente, o o número de amostras de alto gradiente, entre outros. Já as informações de codificação são informações tais quais custo do RDO, melhor modo ISP, melhor modo normal, a razão entre o bloco adjacente e o bloco atual, a razão entre o custo do RDO do bloco pai e do bloco atual e os resíduos.

Para os resultados experimentais foi usado o software de referência VTM 4.0 utilizando os valores de QP 22, 27, 32 e 37 para todas as sequências de vídeo que são as condições comuns de teste do VVC. Os resultados demonstram que a solução para evitar os modos IBC e ISP levaram a uma redução de 19,07% em tempo de codificação com 0,23% de perda no BD-BR. Já a solução que busca eliminar predições desnecessárias demonstram uma redução de 11,30% em tempo de codificação com perda no BD-BR de 0,23%. O resultado geral indica uma redução em tempo de codificação de 50,53% com perda no BD-BR de 1,03%.

- **(Park; Kim; Jeon, 2022):** este trabalho tem como objetivo propor um esquema para evitar os ISP antes de testá-los, levando em consideração os benefícios da predição e das transformadas juntas. Com isso, serão descartados modos ISP que provavelmente não seriam escolhidos como o melhor modo, e, consequentemente, será reduzida a complexidade de codificação. O método proposto foi

modelado como um classificador binário onde a decisão será 0 caso os modos ISP não possam ser evitados ou 1 caso os modos ISP possam ser evitados.

A solução proposta pelos autores baseia-se em um algoritmo de aprendizado de máquina chamado LightGBM. A decisão que este modelo deve tomar é se a combinação de ISP pode ou não ser evitada. Esta combinação possui três informações: a primeira indica qual o modo foi escolhido para o ISP, seja este Planar, DC ou Angular. A segunda informação indica como o bloco foi dividido, ou seja, se o bloco foi dividido vertical ou horizontalmente. Já a terceira informação indica se foi utilizada alguma transformada LFNST e, caso afirmativo, qual transformada foi utilizada.

Segundo as análises desenvolvidas pelos autores, a proporção do bloco é um dado que possui relevância uma vez que há uma maior probabilidade de haver uma divisão vertical em blocos alongados de forma horizontal e vice-versa, enquanto que a relevância do tamanho do bloco pode ser percebida ao verificar-se que o ISP tende a ser escolhido com mais frequência em blocos menores. Outra análise indica a importância do modo de predição intra-quadro na escolha dos modos ISP, onde percebeu-se que os modos mais comuns para os ISP são o Planar em primeiro lugar, seguido do DC e dos modos Angulares horizontal e vertical. Foi verificado que, dentro dos modos angulares, o ISP horizontal tende a estar distribuído principalmente dentro dos modos angulares horizontais, enquanto o ISP vertical tende a estar distribuído principalmente dentro dos modos angulares verticais. Por último, a fórmula desenvolvida pelos autores, cujo nome MAST significa média absoluta da soma dos coeficientes de transformada, busca investigar a relevância das transformadas na escolha dos ISPs. Para isto, é realizada uma predição no bloco inteiro e após é aplicada a transformada primária para cada sub partição. O que se observou é que os valores da MAST crescem de forma gradual desde a primeira subpartição até a quarta subpartição quando a decisão final foi utilizar ISP, enquanto os valores da MAST não demonstram diferença significativa ao fazer a mesma comparação quando a decisão final foi não utilizar ISP.

O modelo se utiliza de dois classificadores, sendo um para os ISPs horizontais e outro para os ISPs verticais, desta forma, quando o modelo recebe uma combinação de ISP, ele verifica se a divisão foi feita de forma horizontal ou vertical e passa para o classificador respectivo. Os dados passados para o classificador foram o tamanho do bloco, a proporção do bloco, o modo de predição intra, o valor gerado pela MAST aplicada a primeira subpartição do bloco, o valor gerado pela MAST aplicada a última subpartição do bloco e o valor da divisão entre estes dois valores.

Para os resultados experimentais foi usado o software de referência VTM 11.0, com a configuração AI, utilizando os valores de QP 22, 27, 32 e 37 para as 26 sequências de vídeo que são as condições comuns de teste do VVC. Os resultados demonstram que a solução dos autores trouxe uma redução de 7,2% em tempo de codificação com 0,8% de perda no BD-BR.

4.2 Considerações Finais

A Tabela 2 sintetiza os trabalhos levantados nesta Seção. Essa Tabela indica se os resultados foram obtidos utilizando métodos de Aprendizado de Máquina (ML) ou métodos Heurísticos (H), se foram extraídas características do processo de codificação (C) ou da imagem (I), qual a redução do tempo de codificação, qual a perda em eficiência de codificação e a razão entre a redução no tempo de codificação e a perda em eficiência de codificação.

Tabela 2 – Síntese dos Trabalhos Relacionados com foco nos Modos ISP

Trabalho	**Método**	**características**	**TS**	**BD-BR**	**TS/BD-BR**
(Park; Kim; Jeon, 2020)	H	I	12,11%	0,43%	28,16
(Liu et al., 2021)	H	I	7,00%	0,09%	77,78
(Saldanha et al., 2021)	ML/H	C/I	8,32%	0,31%	26,84
(Dong et al., 2022)	ML	C	50,53%	1,03%	49,05
(Park; Kim; Jeon, 2022)	ML	C	7,20%	0,80%	9,00

Fonte: Elaborada pelo autor.

Ao analisar a Tabela, observa-se que, embora a maioria dos trabalhos utilize soluções baseadas em ML, também há abordagens fundamentadas em heurísticas. Entre os estudos que adotam ML, (Saldanha et al., 2021) propõe uma solução abrangente que combina três métodos distintos, enquanto (Dong et al., 2022) divide o problema em duas etapas: uma para evitar modos IBC e ISP e outra para eliminar predições desnecessárias. Já (Park; Kim; Jeon, 2022) emprega um esquema de classificação binária para descartar modos ISP antes mesmo de testá-los.

Por outro lado, entre as abordagens heurísticas, destacam-se trabalhos como (Liu et al., 2021), que calcula o desvio médio absoluto para avaliar a complexidade da textura do bloco e eliminar os modos ISP mais simples da RD-List. Além disso, (Park; Kim; Jeon, 2020) utiliza a similaridade entre a direção da predição e a orientação da subpartição para descartar modos menos promissores antes do teste na RD-List.

Diante desse cenário, ainda há espaço para o desenvolvimento de abordagens que combinem tanto características do processo de codificação, permitindo evitar carga computacional adicional, quanto cálculos simples sobre a imagem para identificar modos ISP mais prováveis de serem testados.

5 METODOLOGIA DE MODELAGEM E AVALIAÇÃO

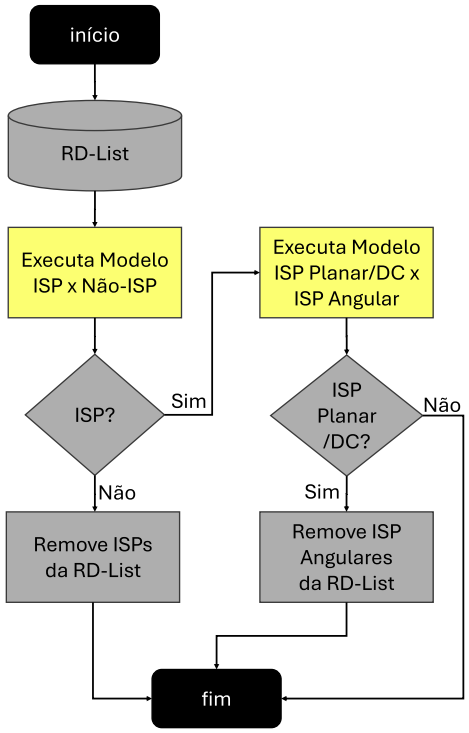
Neste Capítulo, são detalhados os métodos e as ferramentas utilizadas no desenvolvimento e avaliação da solução proposta para otimizar o processo de codificação de vídeo, com ênfase na redução do esforço computacional e na minimização do impacto sobre a eficiência de codificação. O foco é a adoção de estratégias baseadas em aprendizado de máquina para aprimorar a decisão de uso do ISP no software de referência VTM. O Capítulo está estruturado nas seguintes Seções: na Análise e Modelagem do Problema busca-se identificar pontos estratégicos para a introdução de melhorias no funcionamento dos modos intra. Na Aquisição e Geração dos Datasets é explorado como foram coletadas as características utilizadas bem como qual separação foi feita dos conjuntos de dados. No Treinamento de Árvores de Decisão são detalhadas as etapas do processo de treinamento dos conjuntos de dados utilizando as árvores de decisão. Já na Implementação é descrito como as árvores foram inseridas no código do VTM e sua utilização. Por fim, Avaliação indica os métodos utilizados para avaliar como a solução proposta se comporta em comparação com a implementação âncora.

5.1 Modelagem do Problema de Decisão ISP

A Figura 5 apresenta uma visão geral da modelagem proposta para acelerar a decisão dos modos ISP na predição intra-quadro. A solução foi estruturada com base em análises preliminares realizadas sobre um conjunto de vídeos utilizado no trabalho de (Duarte et al., 2023), codificado conforme as Condições Comuns de Teste no VTM 18.0, na configuração *All Intra*. Essas análises forneceram *insights* valiosos sobre o comportamento dos modos ISP, permitindo identificar padrões de uso e orientar a definição de estratégias para redução do esforço computacional, mantendo a eficiência de codificação.

A partir dessas observações, que envolveram, por exemplo, a frequência de utilização dos modos de acordo com o tamanho do bloco e a ocorrência de modos ISP associados aos modos Angulares presentes na RD-List, o problema foi decomposto

Figura 5 – Ilustração do Modo de Decisão Rápido para os Modos ISP



Fonte: Elaborada pelo autor.

em dois subproblemas principais: (i) a decisão entre utilizar ou não os modos ISP e (ii) a seleção do modo ISP mais apropriado, caso o uso do ISP seja considerado vantajoso.

Essas duas etapas, destacadas na Figura 5, são tratadas como decisões sequenciais. A primeira etapa é responsável por determinar se o bloco analisado possui potencial para se beneficiar do uso de ISP. Quando a decisão é Não-ISP, todos os modos ISP são descartados antecipadamente, evitando avaliações desnecessárias e contribuindo significativamente para a redução do tempo de codificação. Já a segunda etapa, executada apenas quando o bloco é classificado como ISP, visa restringir o número de modos ISP a serem testados, reduzindo a carga computacional sem comprometer a qualidade da codificação.

A modelagem de cada subproblema foi feita com base em diferentes conjuntos de características. Para a decisão entre ISP e Não-ISP, foram utilizadas características extraídas diretamente da imagem, calculadas em nível de bloco e de subpartição. As características de imagem foram utilizadas na etapa inicial da decisão principalmente por permitirem, com base em informações visuais simples do bloco, descartar todos os modos ISP de uma só vez quando identificada baixa probabilidade de ganho com seu uso. Além disso, essas características podem ser calculadas com baixo custo computacional e estão disponíveis antes do início efetivo da predição, o que permite

que a decisão ocorra de forma antecipada. Já na decisão do modo ISP, optou-se por características extraídas durante o processo de codificação, escolhidas por já estarem disponíveis no processo de compressão, não gerando sobrecarga adicional na coleta dos dados. Cabe mencionar que também foi avaliada a utilização combinada de características de imagem e tabulares em ambas as etapas, mas os resultados obtidos não demonstraram melhoria significativa no desempenho, o que levou à adoção de conjuntos distintos e especializados para cada subproblema.

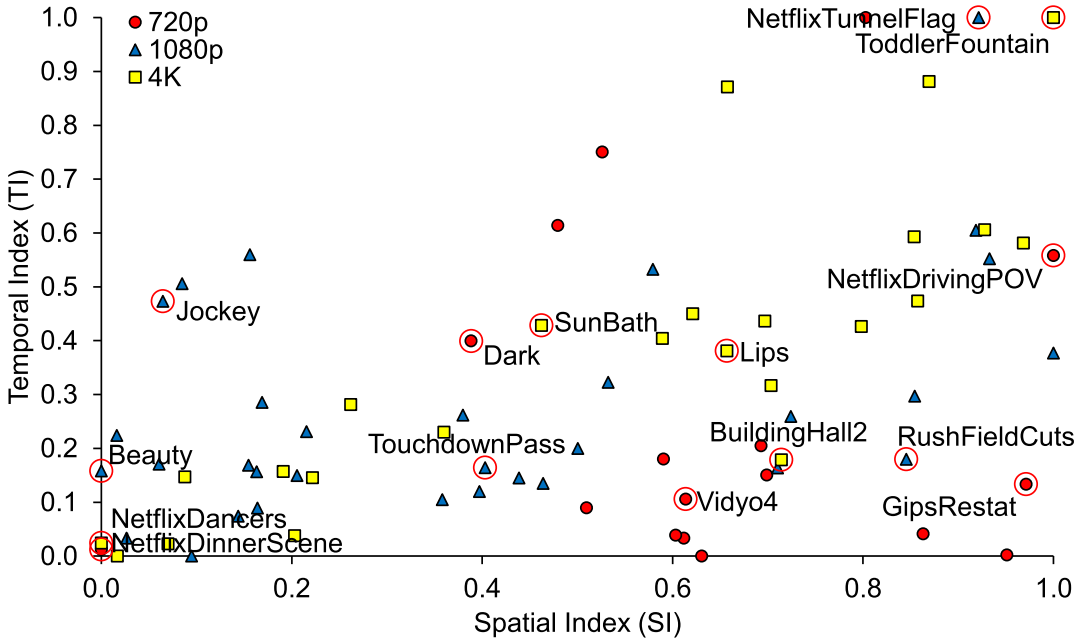
Ambas as decisões foram implementadas com modelos de árvores de decisão, devido à sua boa performance em dados tabulares e ao rápido tempo de inferência. Além disso, a estrutura condicional das árvores facilita sua tradução para C++ e a integração direta com o codificador VTM, o que foi essencial para a implementação prática da solução proposta.

5.2 Aquisição e Geração dos Datasets

Para a realização das análises iniciais, foi utilizado um conjunto de vídeos previamente selecionado no trabalho de (Duarte et al., 2023). Nesse estudo, foram escolhidas 15 sequências a partir de um conjunto de 73 vídeos, utilizando como critério a métrica SIxTI, que avalia a variabilidade espacial (*Spatial Information* - SI) e a variabilidade temporal (*Temporal Information* - TI) das sequências. O valor de SI representa a complexidade espacial dentro de um único quadro, enquanto o TI mede as variações ao longo do tempo entre quadros consecutivos. Os vídeos selecionados possuem três resoluções distintas — HD (720p), Full HD (1080p) e Ultra HD (4K) — sendo escolhidos cinco vídeos de cada categoria. Para garantir diversidade, a seleção foi realizada dividindo as sequências em quadrantes de acordo com a métrica SIxTI, escolhendo, para cada quadrante, um vídeo de cada resolução que estivesse mais distante da mediana, além de um vídeo representativo do centro da distribuição. Destaca-se que os vídeos utilizados neste estudo não pertencem ao conjunto de testes das CTC, o que amplia a diversidade das sequências analisadas e permite explorar diferentes características de conteúdo. A distribuição das sequências selecionadas em relação às métricas SI e TI pode ser observada na Figura 6, que ilustra como os vídeos estão posicionados nesses eixos. Na Figura, os vídeos escolhidos para compor o conjunto de dados estão destacados por círculos vermelhos e identificados pelo nome.

A partir desse conjunto de vídeos, foi realizada a codificação utilizando o VTM 18.0 na configuração All Intra, seguindo as diretrizes das CTCs do VVC. O objetivo principal dessas análises iniciais foi investigar o comportamento dos modos ISP na predição intra-quadro do VVC, buscando identificar padrões e informações relevantes que pudessem contribuir para a redução do esforço computacional sem comprometer a eficiência da compressão. Dessa forma, foi possível estabelecer quais subproble-

Figura 6 – Distribuição das sequências de vídeos de acordo com a métrica SIxTI



Fonte: (Duarte et al., 2023).

mas precisavam ser abordados para efetivamente reduzir o esforço computacional da codificação sem impactar negativamente a taxa de compressão.

A análise envolveu a observação de diferentes aspectos do uso do ISP, incluindo a frequência da utilização dos modos de acordo com o tamanho do bloco, bem como o modo associado escolhido quando os modos ISP são selecionados de acordo com o tamanho do bloco e a frequência da quantidade de modos ISP associados aos modos Angulares presentes na RD-List para serem testados. A partir dessas observações, o problema foi subdividido em dois subproblemas principais: a decisão sobre o uso do ISP e a escolha do modo ISP mais apropriado.

A modelagem dos subproblemas envolveu a identificação das características mais relevantes extraídas do processo de codificação e das imagens, além da definição de critérios para classificar e prever o uso do ISP. Para resolver esses subproblemas, optou-se pelo uso de aprendizado de máquina, especificamente árvores de decisão, devido à sua boa performance em dados tabulares e ao rápido tempo de inferência.

5.2.1 Conjunto de Dados de Características de Imagem

O primeiro conjunto de dados foi construído com base na análise das variâncias dos blocos de luminância das imagens antes da codificação, uma vez que a variância do bloco e de suas subpartições horizontais e verticais é um fator relevante para determinar a complexidade do bloco e pode ser um bom indicativo da utilização dos modos ISP. Aproximadamente 800.000 amostras foram extraídas, sendo elas balanceadas de

Tabela 3 – Características extraídas para os modelos de decisão

Nome	Descrição	Todos os tamanhos
blockVariance	Variância do bloco inteiro.	✓
verticalSubVar1	Variância da primeira subpartição vertical.	✓
verticalSubVar2	Variância da segunda subpartição vertical.	✓
verticalSubVar3	Variância da terceira subpartição vertical.	✗
verticalSubVar4	Variância da quarta subpartição vertical.	✗
horizontalSubVar1	Variância da primeira subpartição horizontal.	✓
horizontalSubVar2	Variância da segunda subpartição horizontal.	✓
horizontalSubVar3	Variância da terceira subpartição horizontal.	✗
horizontalSubVar4	Variância da quarta subpartição horizontal.	✗
blockWidth	Largura do bloco.	✓
blockHeight	Altura do bloco.	✓
qp	Parâmetro de Quantização associado ao bloco.	✓

Fonte: Elaborada pelo autor.

acordo com vídeo, qp, tamanho de bloco e classe.

As características extraídas estão listadas na Tabela 3. A primeira coluna da tabela apresenta os nomes atribuídos a cada uma das características utilizadas, enquanto a segunda fornece uma breve descrição do que cada uma representa. A terceira coluna indica se determinada característica está presente em todos os tamanhos de bloco considerados no conjunto de dados. Essa distinção é importante especialmente no caso das variâncias de subpartições, pois diferente dos blocos maiores, os blocos de tamanho 4×8 e 8×4 são divididos apenas em duas subpartições horizontais ou verticais.

Entre as características extraídas, destacam-se a variância do bloco completo, que indica sua complexidade global, e as variâncias das subpartições horizontais e verticais, que caracterizam a complexidade em diferentes direções. Além disso, foram incluídas as características de codificação relacionadas ao tamanho do bloco, permitindo diferenciar blocos de diferentes dimensões dentro do conjunto de dados, além do valor do QP associado ao bloco, a fim de analisar sua influência na decisão do modelo. Essas informações possibilitam uma compreensão mais aprofundada da estrutura dos blocos e do impacto de suas características na codificação.

A variância, seja do bloco inteiro ou de uma subpartição, foi calculada conforme a Equação (8). Nessa equação, *B* representa o bloco ou subpartição considerada, x_i denota os valores das amostras, μ é a média das amostras e *N* é o número total de amostras no bloco ou subpartição.

$$Var(B) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{8}$$

5.2.2 Conjunto de Dados de Características de Codificação

O segundo conjunto de dados foi extraído a partir das estatísticas geradas pelo processo de codificação do VTM, abrangendo diferentes estágios desse processo, como RMD, MPM, RD-List e RDO. A Tabela 4 detalha essas características. O conjunto de dados gerado contém um total de 48 características extraídas dessas etapas, com aproximadamente 800.000 exemplos balanceados por vídeo, classe, QP e tamanho de bloco.

Esse conjunto de dados elimina a necessidade de processamento adicional, pois as informações já estão disponíveis. Além disso, ele oferece uma visão detalhada dos fatores que influenciam a seleção dos modos ISP, identificando aqueles com maior probabilidade de serem escolhidos como a melhor opção.

RMD calcula rapidamente os custos de taxa-distorção para os modos Planar, DC, Angular e MIP. Esses custos fornecem indícios de qual modo associado aos modos ISP produzirá o melhor custo. Considerando os melhores custos em RMD para os modos mencionados, foram extraídas as seguintes informações: SAD, SATD, número estimado de bits e os custos rápidos de taxa-distorção. Além disso, foram extraídas as posições *x* e *y* do bloco, os melhores modos Angular e MIP, e os melhores valores de MRL para os modos Angular e DC.

O MPM fornece uma lista com seis modos intra, sendo que o primeiro é sempre o modo Planar, enquanto os demais são o modo DC ou um dos modos Angulares. Esses seis modos intra são considerados os melhores candidatos, pois foram os escolhidos para os blocos vizinhos à esquerda e acima. A partir do MPM, foram extraídas as seguintes informações: número dos cinco modos intra selecionados (excluindo o Planar, pois é constante), número dos melhores modos intra nos blocos vizinhos à esquerda e acima, oito valores booleanos indicando se o melhor modo intra para os blocos vizinhos é Planar, DC, Angular ou MIP, além de um valor booleano indicando se o DC está presente no MPM.

5.3 Treinamento das Árvores de Decisão

Em ambos os conjuntos de dados, foram utilizadas árvores de decisão implementadas utilizando a biblioteca Scikit-learn (Pedregosa et al., 2011). Os conjuntos de dados foram divididos em 75% para treinamento e validação, reservando os 25% restantes para teste. Essa divisão garantiu que os modelos nunca vissem 25% dos dados durante as etapas de treinamento e validação. Para obter o modelo foi feita uma busca de hiperparâmetros em duas etapas: *Random Search* (Bergstra; Bengio, 2012) e *Grid Search*.

A etapa de *Random Search* envolveu a avaliação de 1.000 combinações aleatórias em um amplo espaço de pesquisa dos hiperparâmetros *criterion*, *min samples split*,

Tabela 4 – Características de codificação extraídas do VTM

Nome	Descrição	Etapa	Tipo	Mín	Máx	# de valores
BlockPosition	As posições *x* e *y* do bloco.	RMD	Inteiro	0	4096	2
BestAngular	Melhor modo angular.	RMD	Inteiro	2	66	1
BestMIP	Melhor modo MIP.	RMD	Inteiro	0	7	1
MRLAngular	Linha de referência MRL do melhor modo angular.	RMD	Inteiro	0	2	1
MRLDC	Linha de referência MRL do melhor modo DC.	RMD	Inteiro	0	2	1
ModesMPM	Modos MPM excluindo Planar.	MPM	Inteiro	1	66	5
ModesPosition	Posição da primeira ocorrência de cada tipo de modo intra.	RD-List	Inteiro	1	12	4
FirstAngular	Primeiro modo angular na RD-List.	RD-List	Inteiro	2	66	1
FirstMIP	Primeiro modo MIP na RD-List.	RD-List	Inteiro	0	7	1
NeighborMode	Melhor número de modo intra em blocos vizinhos.	MPM	Inteiro	0	66	2
NeighborType	Melhor tipo de modo intra em blocos vizinhos.	MPM	Booleano	0	1	8
DCMPM	Modo DC é um MPM.	MPM	Booleano	0	1	1
SAD	Soma das Diferenças Absolutas.	RMD	Decimal	0,63	1390,38	4
SATD	Soma das Diferenças Absolutas Transformadas.	RMD	Decimal	0,52	497,81	4
FracBits	Número estimado de bits.	RMD	Decimal	1,19	15669,28	4
RMD Cost	Custo rápido de taxa-distorção.	RMD	Decimal	0,65	502,01	4
RDO Cost	Custo completo de taxa-distorção.	RDO	Decimal	182,14	1881715,88	4
				Total	**48**	

Fonte: Elaborada pelo autor.

min samples leaf, *max depth*, *max leaf nodes* e *max features*. Cada combinação foi avaliada utilizando *5-fold cross-validation* (Pedregosa et al., 2011) sobre 75% dos dados reservados para treinamento e validação. Em seguida, usamos os resultados do *Random Search* para calcular a Correlação de Pearson entre cada hiperparâmetro e o F1-score.

Os dois hiperparâmetros selecionados para cada conjunto de dados foram refinados na etapa de *Grid Search*, enquanto os hiperparâmetros restantes foram configurados com seus valores padrão ou os melhores valores encontrados durante a *Random Search*.

A etapa de *Grid Search* também avaliou combinações por meio de uma *5-fold cross-validation* sobre os mesmos 75% dos dados usados para treinamento e validação. Os modelos finais foram derivados da combinação de hiperparâmetros que obteve o melhor F1-score no *Grid Search*.

5.4 Implementação no Codificador VTM

A implementação da solução proposta foi realizada diretamente no código-fonte do VTM 18.0, codificador de referência do VVC. Para isso, foram inseridas modificações no fluxo de codificação, permitindo a extração das características necessárias e a tomada de decisão baseada nos modelos treinados. Dessa forma, os modelos adicionam critérios para a escolha dos modos ISP.

Ambas as soluções desenvolvidas utilizam árvores de decisão para a escolha dos modos ISP na predição intra-quadro. Os modelos treinados foram integrados ao código do VTM na forma de funções específicas, responsáveis por analisar as características disponíveis e realizar a tomada de decisão.

A solução baseada em características do processo de codificação possui uma fun-

ção dedicada que, quando requerida, recebe como entrada as características selecionadas que foram coletadas durante a codificação, avalia os dados recebidos e retorna a decisão do modelo.

Já a solução baseada em características extraídas das imagens segue uma abordagem diferente. Durante o processo de codificação, o VTM acessa arquivos contendo essas características, carregando-as na memória quadro a quadro. Assim, quando necessário, as informações são fornecidas ao modelo de árvore de decisão para que ele realize a predição com base nas características da imagem. Cabe destacar que, na modelagem proposta, essa etapa foi concebida considerando a possibilidade de execução em unidades de processamento paralelo, o que permite que a consulta às características e a decisão do modelo ocorram de forma eficiente.

5.5 Avaliação dos resultados

Nesta seção, são apresentadas e descritas as métricas utilizadas para avaliar a eficácia da solução proposta no contexto da codificação intra-quadro do VVC. As métricas consideradas permitem analisar tanto a redução no tempo de codificação quanto o impacto na eficiência de codificação.

- ***Time Saving* (TS)**: Representa a redução percentual no tempo total de codificação ao comparar a implementação padrão do VTM com a abordagem proposta, conforme definido na Equação (9). Nessa equação, CP refere-se à versão padrão do VTM utilizada nos experimentos, enquanto CM corresponde à mesma versão do VTM modificada com a solução proposta. TT representa o tempo total de execução para ambas as versões. Embora este trabalho utilize o VTM 18.0, a equação pode ser aplicada a qualquer versão, desde que a comparação seja feita dentro da mesma versão do codificador. O valor final de TS é obtido a partir da média dos resultados para os quatro valores de QP considerados, conforme mostrado na Equação (10), onde $i = 1, 2, 3, 4$ corresponde aos $QPs = 22, 27, 32, 37$, respectivamente.

- **BD-BR**: O BD-BR é uma métrica utilizada para comparar a eficiência de diferentes algoritmos de compressão de vídeo, medindo a variação na taxa de bits necessária para manter a mesma qualidade visual em relação a uma codificação de referência. Essa métrica é amplamente utilizada para avaliar e comparar diferentes *codecs* ou técnicas de compressão em termos de eficiência de codificação, permitindo uma análise mais precisa de sua performance (Bjontegaard, 2001).

- **TS/BD-BR**: Representa a razão entre TS e BD-BR, indicando a redução do esforço computacional obtida para cada 1% de perda na eficiência de codificação.

Essa métrica é especialmente útil para avaliar o compromisso entre desempenho e eficiência de codificação, permitindo quantificar o impacto da solução proposta na complexidade computacional em relação ao comprometimento da taxa de bits. Valores mais altos de TS/BD-BR indicam que a redução no tempo de codificação foi mais significativa em comparação à perda de eficiência na compressão, tornando a abordagem mais vantajosa.

$$TimeSaving = 100 - \left(\frac{TT_{(CM)} \times 100}{TT_{(CP)}} \right) \tag{9}$$

$$TS = \frac{\sum_{i=1}^{4} TimeSaving_{QP_{(i)}}}{4} \tag{10}$$

6 ANÁLISES SOBRE A DECISÃO DE MODO ISP

Este Capítulo busca detalhar os resultados obtidos a partir de diferentes análises realizadas sobre a taxa de ocorrência de modos ISP e sobre a influência das características extraídas das imagens na escolha desses modos. Os experimentos realizados utilizam um conjunto de 15 vídeos selecionados no trabalho de (Duarte et al., 2023) que utilizou as métricas SI e TI, codificados com o VTM 18.0 na configuração *All Intra* e nos valores de QP 22, 27, 32 e 37. Para possibilitar a coleta dos dados necessários às análises, foram implementadas modificações no codificador.

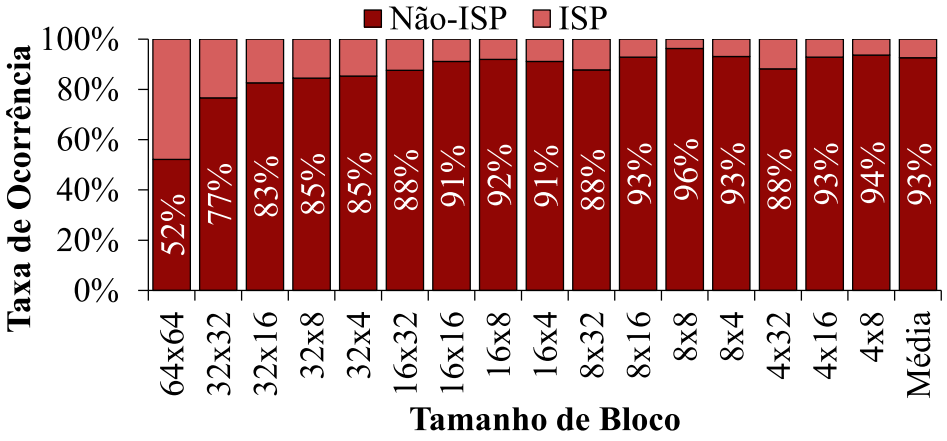
A primeira Seção foca na análise da taxa de ocorrência dos modos ISP, buscando entender tanto a frequência com que esses modos são selecionados em comparação aos modos não-ISP, quanto a frequência dos modos ISP Planar, DC e Angulares em busca de possibilidades de redução do esforço computacional associado aos modos ISP. Já a segunda Seção concentra-se na análise das características, explorando como diferentes características de textura e codificação indicam o modo mais provável a ser escolhido. Os resultados apresentados neste Capítulo servirão de base para a construção da solução proposta, abordada no próximo Capítulo.

6.1 Análise da Taxa de Ocorrência dos Modos ISP

Nesta Seção, apresentamos uma análise da taxa de ocorrência da ferramenta ISP no VTM. Para isso, foram conduzidas três análises distintas. A primeira avalia a ocorrência geral do ISP, ou seja, a frequência com que a decisão final do modo intra do VTM seleciona um modo ISP como a melhor escolha para um determinado bloco. Para isto, os modos intra foram organizados em duas classes: **(i) Não-ISP**, que inclui os modos Planar, DC, Angular e MIP; e **(ii) ISP**, que agrupa os modos ISP, abrangendo subpartições horizontais ou verticais combinadas com os modos Planar, DC ou qualquer um dos modos Angulares.

Nas segunda e terceira análises, focamos nas taxas de ocorrência de modos intra específicos durante a etapa ISP, bem como no número de avaliações realizadas para esses modos. Para essas análises, os modos ISP foram subdivididos em duas classes

Figura 7 – Taxa de Ocorrência das classes Não-ISP e ISP de acordo com o tamanho do bloco



Fonte: Elaborada pelo autor.

adicionais: **(i) ISP Planar/DC** e **(ii) ISP Angular**. Essa classificação é fundamentada na diferença entre esses modos, visto que os modos Planar e DC são mais adequados para texturas homogêneas, enquanto os modos Angulares são mais eficazes para texturas direcionais.

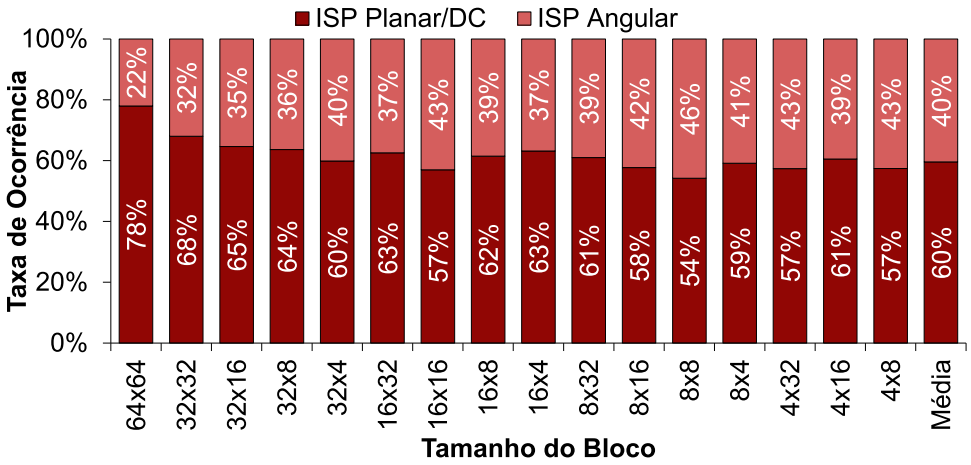
Nas duas primeiras análises, calculamos as taxas de ocorrência de cada classe organizadas por tamanho de bloco, a fim de identificar cenários nos quais determinadas avaliações poderiam ser evitadas. Na terceira análise, investigamos o número de avaliações realizadas especificamente para a classe ISP Angular, visando compreender a frequência com que esses modos são avaliados durante a etapa ISP.

A Figura 7 apresenta a taxa de ocorrência das classes **Não-ISP** e **ISP** categorizada pelo tamanho do bloco. Observamos que, para todos os tamanhos de bloco, a classe Não-ISP apresenta uma taxa de ocorrência superior à classe ISP. Para blocos de tamanho 64×64, a classe ISP se aproxima da ocorrência da classe Não-ISP. No entanto, à medida que o tamanho do bloco diminui, a taxa de ocorrência da classe ISP também reduz. Em média, a classe Não-ISP obtém uma taxa de ocorrência de 93%, enquanto a classe ISP alcança apenas 7%. Isso indica que a avaliação dos modos ISP poderia ser evitada na maioria dos casos.

A Figura 8 apresenta a taxa de ocorrência das classes **ISP Planar/DC** e **ISP Angular**, categorizada por tamanho de bloco. Podemos observar que a classe ISP Planar/DC possui uma ocorrência mais elevada quando comparada à classe ISP Angular para blocos de maior tamanho, especialmente para 64×64 e 32×32. Além disso, a classe ISP Planar/DC mantém uma taxa de ocorrência superior em todos os tamanhos de bloco, atingindo, em média, 60% da taxa total. Dessa forma, quando o modo ISP é selecionado, os modos Planar e DC são mais prováveis de serem escolhidos.

Por fim, a Figura 9 ilustra a frequência dos candidatos ISP associados aos modos

Figura 8 – Taxa de Ocorrência das classes ISP Planar/DC e ISP Angular de acordo com o tamanho do bloco



Fonte: Elaborada pelo autor.

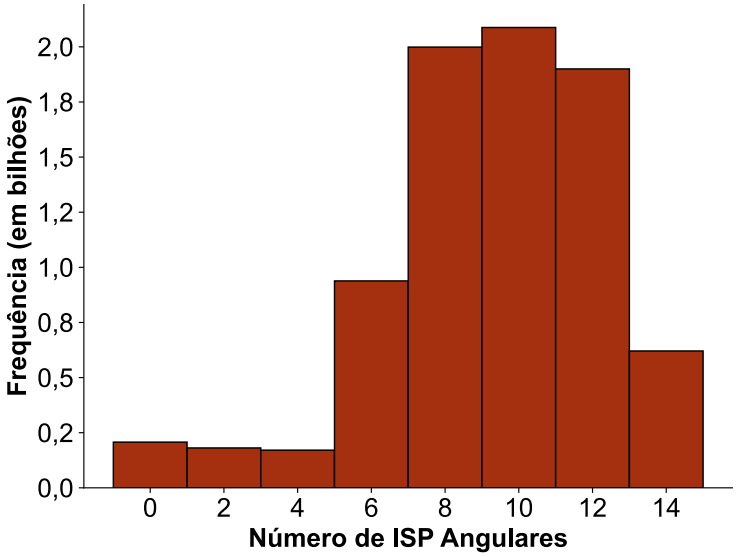
Angulares na RD-List. Nota-se que o número de candidatos ISP associados aos modos Angulares é sempre par, pois o *software* de referência VTM avalia os mesmos modos intra para subpartições horizontais e verticais. Além disso, observa-se que, na maioria dos casos, há 8, 10 ou 12 candidatos ISP associados aos modos Angulares, indicando que o VTM avalia, em média, de 8 a 12 candidatos ISP por bloco. Esse resultado sugere um potencial para a redução do esforço computacional, pois, ao descartar os candidatos ISP associados aos modos Angulares, é possível evitar, em média, a avaliação de 8 a 12 modos Angulares vinculados ao ISP.

Em resumo, considerando a alta taxa de ocorrência da classe Não ISP, abordagens baseadas em aprendizado de máquina para prever quando a avaliação dos modos ISP pode ser evitada são promissoras, uma vez que esses modos são sempre avaliados, apesar de, na maioria dos casos, não atingirem o melhor resultado em taxa de distorção. Além disso, dada a alta taxa de ocorrência da classe ISP Planar/DC e a elevada frequência dos candidatos ISP Angulares na RD-List, é possível reduzir o esforço computacional da decisão do modo ISP caso um modelo de aprendizado de máquina seja empregado para prever quando o melhor modo ISP é Planar ou DC. Nessas situações, a avaliação de vários candidatos ISP Angulares pode ser ignorada.

6.2 Análise das características de imagem e de codificação com a decisão de modo ISP

Nesta Seção, realizamos uma análise detalhada das características extraídas tanto das imagens quanto do processo de codificação. Enquanto as análises relacionadas as características de imagem possuem o objetivo de compreender o impacto de di-

Figura 9 – Frequência dos modos ISP Angulares na RD-List



Fonte: Elaborada pelo autor.

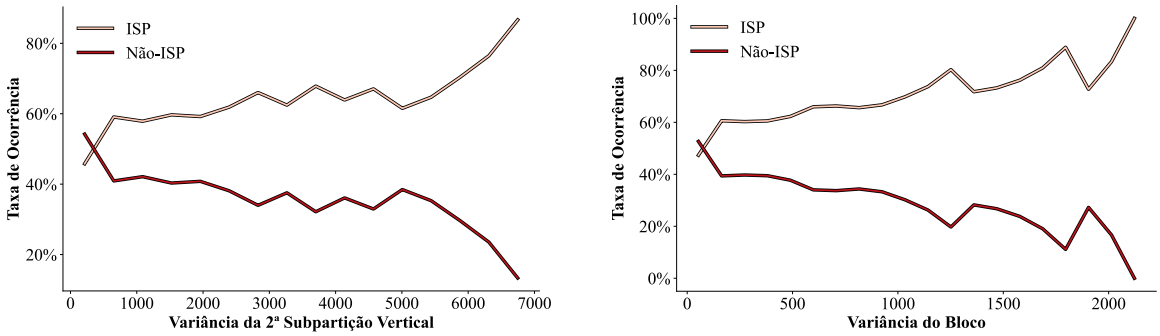
ferentes características nas decisões relacionadas a avaliar os modos ISP, as análises relacionadas as características do processo de codificação podem esclarecer o impacto de diferentes características nas decisões relacionadas aos modos mais prováveis associados aos modos ISP. Para tanto, selecionamos as características que apresentam o maior ganho de informação utilizando a ferramenta WEKA (Hall et al., 2009), dividindo os conjuntos de dados conforme os tamanhos dos blocos e avaliando a influência dessas características na escolha dos modos intra-quadro.

6.2.1 Análise das Características de Variância

No conjunto de dados com características de imagem, considerando os blocos de tamanho 4×8 e 8×4, a característica que apresentou o maior ganho de informação foi a variância da segunda subpartição vertical. Já para os demais tamanhos de bloco, a variância do bloco inteiro se destacou como a característica mais informativa. Para a análise destas características, dividimos os valores observados de cada uma delas em 20 intervalos iguais, definidos com base nos valores mínimo e máximo obtidos para cada característica. Dentro de cada intervalo, calculamos a taxa de ocorrência das classes Não-ISP e ISP.

A Figura 10a ilustra como a variância da segunda subpartição vertical impacta as taxas de ocorrência das classes Não-ISP e ISP. Para valores muito baixos de variância, a classe Não-ISP apresenta uma taxa de ocorrência superior à classe ISP. No entanto, à medida que a variância aumenta, a taxa de ocorrência da classe ISP também sobe, atingindo picos superiores a 80% para valores elevados de variância.

De forma semelhante, a Figura 10b mostra a influência da variância do bloco inteiro



(a) Variância da segunda subpartição (b) Variância do bloco inteiro

Figura 10 – Taxa de Ocorrência das classes Não-ISP e ISP de acordo com o cálculo da variância

nas taxas de ocorrência das classes. Novamente, a classe Não-ISP apresenta uma taxa mais alta para valores baixos de variância, mas conforme a variância do bloco aumenta, a taxa de ocorrência da classe ISP tende a se aproximar de 100%.

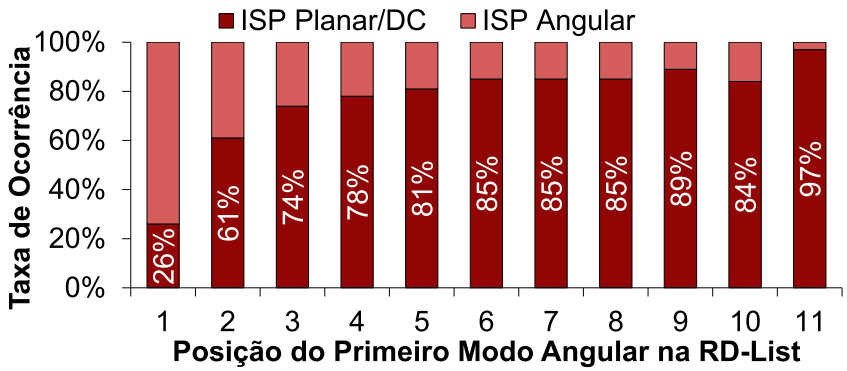
A análise da variância das subpartições e do bloco inteiro é um indicativo da possibilidade de evitar a avaliação dos modos ISP. Especificamente, a variância da segunda subpartição vertical e a variância do bloco inteiro mostraram um aumento na taxa de ocorrência da classe ISP à medida que os valores de variância aumentavam. Isso sugere que, em cenários com alta variância, os modos ISP são mais apropriados, uma vez que esses modos são mais eficazes para modelar a complexidade das texturas nas imagens.

6.2.2 Análise das características de Posição no RD-List e MPM

Além das variâncias, também foi feita a análise de duas características relacionadas ao processo de codificação, que demonstram relevância para a escolha dos modos ISP: a posição do primeiro modo Angular na lista RD-List e o modo intra presente na segunda posição da lista MPM. Esta segunda característica é especialmente significativa, pois, no processo de geração da lista MPM, o modo Planar é sempre inserido como o primeiro candidato. Assim, o modo que aparece na segunda posição representa, de fato, a primeira escolha real baseada no contexto do bloco. Notavelmente, quando o modo DC está presente na lista MPM, ele ocupa invariavelmente essa segunda posição. Ambas as características influenciam diretamente a seleção do modo intra mais adequado e, consequentemente, a necessidade de avaliar os modos ISP.

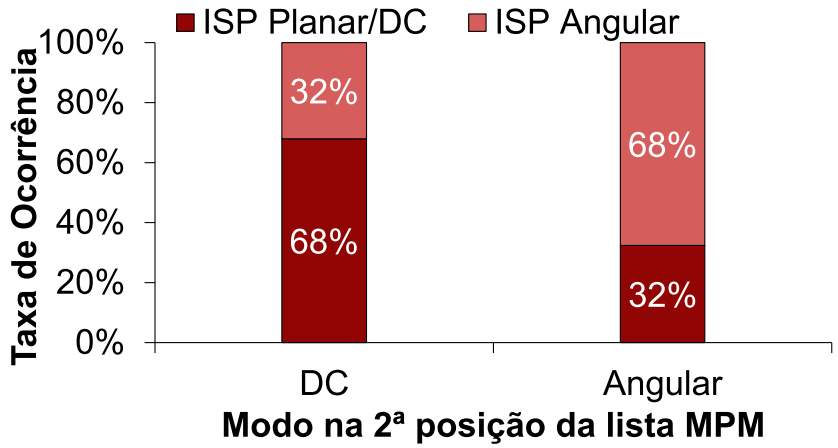
Na Figura 11, podemos observar a taxa de ocorrência das classes ISP Planar/DC e ISP Angular conforme a posição do primeiro modo Angular na RD-List. Quando o primeiro modo Angular aparece nas primeiras posições da lista, a classe ISP Angular tende a ter uma taxa de ocorrência mais alta. No entanto, à medida que o primeiro modo Angular se desloca para posições posteriores, a taxa de ocorrência da classe ISP Planar/DC aumenta, chegando a 97% quando o primeiro modo Angular ocupa a

Figura 11 – Taxa de Ocorrência das classes ISP Planar/DC e ISP Angular de acordo com a posição do primeiro modo angular na RD-List



Fonte: Elaborada pelo autor.

Figura 12 – Taxa de Ocorrência das classes ISP Planar/DC e ISP Angular de acordo com o modo presente na segunda posição da lista MPM



Fonte: Elaborada pelo autor.

décima primeira posição. Esses resultados indicam que a posição do primeiro modo Angular pode ser um indicativo importante para prever quando a avaliação dos modos ISP Angulares pode ser evitada.

A Figura 12 apresenta a análise da posição do modo intra na segunda posição da lista MPM. Quando o modo DC ocupa essa posição, a classe ISP Planar/DC tem uma taxa de ocorrência de 68%, sugerindo que, nesses casos, a avaliação dos modos ISP Angulares pode ser evitada em até 68% das situações. Em contraste, quando um modo Angular ocupa a segunda posição, a classe ISP Angular apresenta uma taxa de ocorrência mais alta, indicando que a avaliação dos modos ISP Angulares é mais necessária nesse contexto.

A análise destas características permite a possibilidade do desenvolvimento de modelos preditivos, tais quais árvores de decisão, que podem ser treinados para an-

tecipar a necessidade de avaliação dos modos ISP, com base nas características de textura e nos parâmetros de codificação das imagens.

7 MLISP: ESQUEMA DE DECISÃO ISP BASEADO EM APRENDIZADO DE MÁQUINA

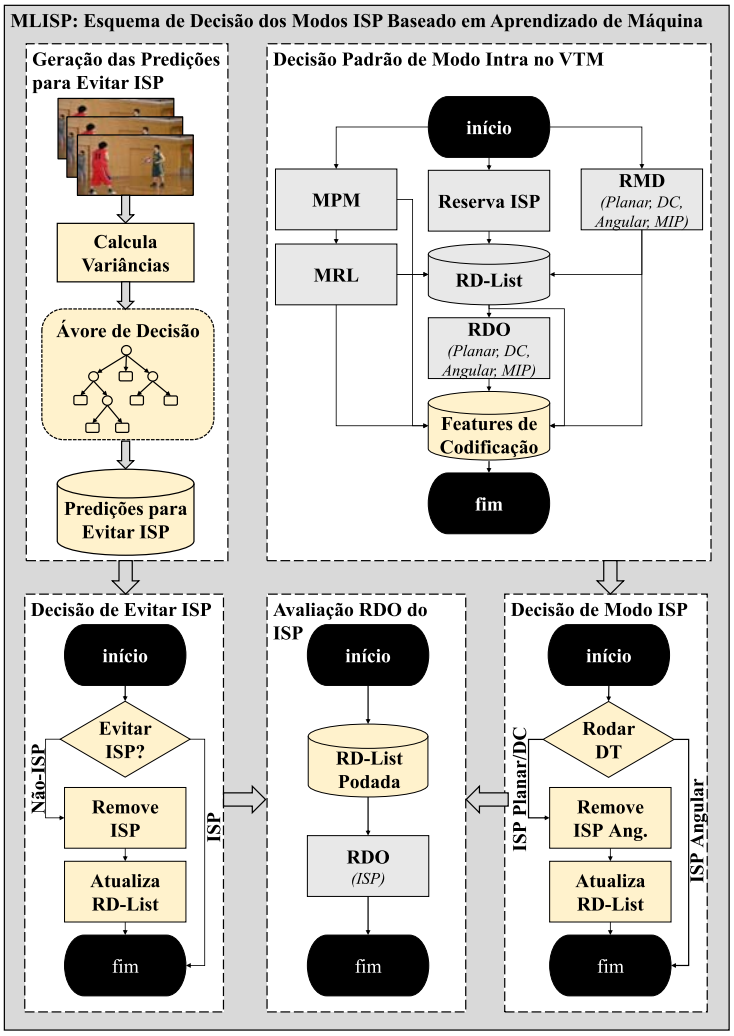
Este Capítulo descreve a proposta de esquema de decisão baseada em aprendizado de máquina para acelerar o processo de decisão dos modos ISP dentro da predição intra-quadro. As análises conduzidas no Capítulo 6 trazem um bom indicativo de que é possível reduzir o esforço computacional atrelado aos modos ISP através de uma avaliação prévia que indique a probabilidade dos modos serem escolhidos como o melhor, ou até mesmo qual o modo associado aos modos ISP seria mais provável de ser selecionado. O esquema proposto, denominado **MLISP**, é composto por duas soluções complementares, que abordam o processo de decisão dos modos ISP de duas formas diferentes.

A primeira solução, denominada **Decisão de Evitar ISP (DE-ISP)**, emprega um modelo de Árvore de Decisão treinado a partir das características extraídas das imagens, classificando os blocos em duas classes. A classe **Não-ISP** que indica blocos onde o menor custo taxa-distorção é obtido pelos modos Planar, DC, Angular ou MIP, e a classe **ISP** que indica blocos onde um dos modos ISP proporciona o menor custo taxa-distorção. A classificação dos blocos em uma das classes do modelo permite determinar se a avaliação dos modos ISP é necessária. Se a classe atribuída indicar que a avaliação pode ser omitida, os modos ISP são eliminados do processo de decisão do RDO, reduzindo a complexidade computacional.

Se este modelo decide por indicar que a avaliação do ISP é necessária, a segunda solução, denominada **Decisão de Modo ISP (DM-ISP)**, é acionada. Essa abordagem também utiliza um modelo de Árvore de Decisão, mas com o objetivo de classificar os blocos em outras duas classes, sendo elas, a classe **ISP Planar/DC** que é onde a combinação do ISP com os modos Planar ou DC apresenta o melhor custo taxa-distorção e a classe **ISP Angular** que é onde a combinação do ISP com um modo Angular proporciona o melhor custo taxa-distorção.

Essa classificação permite reduzir o esforço computacional da etapa RDO para os modos ISP, pois, para blocos classificados como ISP Planar/DC, apenas os modos Planar e DC são avaliados, eliminando a necessidade de testar os modos ISP

Figura 13 – MLISP: Esquema de Decisão dos Modos ISP baseado em Aprendizado de Máquina



Fonte: Elaborada pelo autor.

associados aos modos Angulares. Esta decisão equilibra a redução do tempo de processamento e a eficiência de codificação, dado que a classe ISP Planar/DC ocorre com alta frequência.

A Figura 13 ilustra a integração do MLISP ao processo padrão de decisão de modos intra. Inicialmente, a decisão segue o procedimento já estabelecido, passando pela avaliação RDO para os modos Planar, DC, Angular e MIP. Durante essa etapa, são extraídas as características de codificação pertinentes ao modelo para servir de entrada para a Árvore de Decisão que decide pelo modo ISP.

Simultaneamente, os quadros de vídeo YUV brutos são processados para gerar blocos de luminância de todos os tamanhos definidos pela predição intra. Para cada bloco, são calculadas a variância de cada bloco da imagem, considerando tanto o bloco inteiro quanto suas subpartições horizontais e verticais. Essas informações ali-

mentam a Árvore de Decisão da DE-ISP.

A interação entre as duas soluções durante o processo de codificação ocorre de maneira coordenada, iniciando com a DE-ISP que realiza a predição sobre a necessidade de avaliar os modos ISP para o bloco atual. Caso a predição indique Não-ISP, todos os modos ISP são removidos da RD-List, eliminando sua avaliação no processo de codificação. Por outro lado, se a predição for ISP, os modos ISP são mantidos na RD-List, permitindo que a próxima solução seja chamada, a DM-ISP, que refina a seleção dos candidatos. Nesse estágio, se a predição resultar em ISP Planar/DC, os candidatos ISP correspondentes a modos Angulares são descartados da RD-List, restringindo a avaliação apenas aos modos Planar e DC. No entanto, se a predição indicar ISP Angular, todos os candidatos ISP permanecem e seguem para avaliação, garantindo que os modos mais adequados ao contexto do bloco sejam devidamente analisados, neste caso, os modos ISP Planar/DC não são removidos pois ambos os modos possuem alta taxa de ocorrência.

Com essa abordagem, o MLISP otimiza a composição da RD-List durante a etapa RDO dos modos ISP, reduzindo o esforço computacional necessário sem comprometer significativamente a eficiência de codificação.

De acordo com os trabalhos relacionados discutidos no Capítulo 4, não há, até o momento, propostas que abordem a decisão dos modos ISP da forma apresentada neste trabalho. Nenhum estudo anterior utiliza características extraídas diretamente do processo de codificação para orientar a decisão dos modos ISP, e, embora algumas abordagens empreguem informações extraídas da imagem, a deste trabalho possui cálculos mais simples. Dessa forma, o MLISP representa uma solução inovadora ao combinar diferentes fontes de informação e técnicas de aprendizado de máquina para aprimorar a seleção dos modos ISP.

7.1 Geração dos Datasets

A partir do conjunto de dados extraído no Capítulo 5, Seção 5.2, foram gerados três conjuntos de dados balanceados: dois para a solução de DE-ISP e um para a solução de DM-ISP.

A divisão do conjunto de dados da solução da DE-ISP em dois ocorreu devido à natureza do subparticionamento dos modos ISP. Blocos menores, de tamanhos 4×8 e 8×4, são subdivididos em duas partes, enquanto os demais tamanhos de bloco são divididos em quatro subpartições, tanto na direção vertical quanto na horizontal. Para acomodar a diferença entre a quantidade de características, os conjuntos de dados foram separados de acordo com essa diferença.

O conjunto de dados referente aos blocos menores contém cinco características de variância: a variância do bloco inteiro, duas variâncias das subpartições verticais e

duas variâncias das subpartições horizontais. Já o conjunto de dados dos blocos maiores possui nove características de variância, incluindo a variância do bloco inteiro, quatro variâncias das subpartições verticais e quatro variâncias das subpartições horizontais. Além dessas características, ambos os conjuntos incluem informações sobre o QP, altura, largura e classe. Em termos de volume, o conjunto de dados dos blocos menores contém aproximadamente 100.000 exemplos, enquanto o conjunto referente aos blocos maiores possui cerca de 700.000 exemplos, ambos balanceados por classe, vídeo, qp e tamanho de bloco.

Por outro lado, o conjunto de dados da solução de DM-ISP que também foi balanceado por classe, vídeo, qp e tamanho de bloco possui aproximadamente 800.000 exemplos, sendo que uma vez que o número de características é sempre o mesmo, apenas um dataset se faz necessário. As características foram aquelas descritas na Tabela 4. É importante ressaltar que, apesar do alto número de características de codificação, não há necessidade de cálculos adicionais, pois todas essas informações já estão disponíveis durante o processo de codificação. Foi aplicada a normalização das características relacionadas ao custo de taxa-distorção conforme a Equação (11), onde X representa o conjunto de características relacionadas ao custo de taxa-distorção, incluindo SAD, SATD, FracBits, Custo RMD e Custo RDO. Além disso, w e h representam a largura e altura do bloco, respectivamente. Essa normalização é importante para reduzir o impacto da variação no tamanho dos blocos sobre os valores dessas características, uma vez que blocos maiores tendem a acumular valores absolutos mais elevados. Com isso, busca-se garantir que a análise e os modelos subsequentes considerem as métricas em uma escala comparável, independentemente da dimensão do bloco.

$$x_{\text{norm}} = \frac{x}{w \cdot h}, \quad x \in X, \quad w, h \in \{4, 8, 16, 32, 64\} \tag{11}$$

7.2 Seleção de Características

A fim de reduzir a dimensionalidade dos dados e garantir que apenas as características mais relevantes fossem utilizadas no treinamento dos modelos, foi aplicada a técnica de *Recursive Feature Elimination with Cross-Validation* (RFECV) (Pedregosa et al., 2011) em todos os três conjuntos de dados. O principal objetivo dessa etapa foi eliminar características com baixa contribuição preditiva, reduzindo a complexidade do modelo e potencialmente melhorando sua capacidade de generalização.

O processo de RFECV foi conduzido utilizando um modelo de árvore de decisão treinado com *5-fold cross-validation* (Pedregosa et al., 2011). Inicialmente, o modelo foi treinado com todas as características disponíveis, e, com base na importância atribuída a cada característica, aquela com menor relevância foi removida. O modelo

foi então reavaliado com as N-1 características restantes, repetindo-se esse procedimento iterativamente até que restasse apenas uma característica. No final do processo, o subconjunto de características selecionado foi aquele que resultou no melhor F1-score médio ao longo das execuções da validação cruzada.

Para os conjuntos de dados da DE-ISP, o RFECV não indicou a remoção de nenhuma característica, mantendo o conjunto original de características. Especificamente, para o conjunto de blocos menores (4×8 e 8×4), todas as oito características foram mantidas, incluindo a variância do bloco inteiro, as duas variâncias das subpartições horizontais, as duas variâncias das subpartições verticais, além da largura, altura e QP do bloco. Da mesma forma, para o conjunto de blocos maiores, todas as 12 características foram mantidas, abrangendo a variância do bloco inteiro, as quatro variâncias das subpartições horizontais, as quatro variâncias das subpartições verticais e as informações de largura, altura e QP.

Já para o conjunto de dados da DM-ISP, a aplicação do RFECV resultou na redução do número de características de 48 para 28, mantendo apenas aquelas com maior impacto na predição. Entre as características selecionadas, foram incluídas as posições x e y do bloco, além das seguintes características extraídas das diferentes etapas do processo de codificação:

- **RMD:** Foram retidos os valores de SAD, SATD, número estimado de bits e os custos rápidos de taxa-distorção para os modos Planar, DC, Angular e MIP, com exceção do custo rápido para os modos Angulares. Além disso, foram incluídos o número MRL e o número do modo correspondente ao melhor modo Angular no passo RMD.
- **MPM:** Foi mantido o número do modo que ocupa a segunda posição na lista MPM.
- **RD-List:** Foram selecionadas as posições da primeira ocorrência dos modos Planar, DC, Angular e MIP, além do número de modos Angulares que aparecem primeiro.
- **RDO:** Foram mantidos os custos completos de taxa-distorção para os modos Planar, DC e Angular, considerando apenas os modos que não utilizam ISP.

7.3 Treinamento da Árvore de Decisão

Para o treinamento dos modelos, os passos descritos na Seção 5.3 do Capítulo 5 foram seguidos. Durante a etapa de *Random Search* foi realizada uma análise da influência dos hiperparâmetros no desempenho dos modelos. Para isso, calculou-se o coeficiente de correlação de Pearson entre cada hiperparâmetro e a F1-Score. Esse

coeficiente mede a intensidade e a direção da relação linear entre duas variáveis, indicando o impacto de cada hiperparâmetro no desempenho do modelo. Valores positivos significam que o aumento do hiperparâmetro está associado a uma melhora no F1-Score, enquanto valores negativos sugerem um efeito contrário.

Os modelos utilizados podem ser divididos em duas categorias: os modelos para **Decisão de Evitar ISP (DE-ISP)**, responsáveis por decidir quando o modo ISP deve ser evitado para diferentes blocos, e o modelo para **Decisão de Modo ISP (DM-ISP)**, focado na escolha do melhor modo ISP quando essa decisão não é evitada. Assim, os modelos DE-ISP foram treinados tanto para os blocos 4x8 e 8x4 quanto para os demais blocos, enquanto o modelo DM-ISP foi treinado para a escolha do modo. Essa terminologia será adotada nas tabelas e análises seguintes, tornando mais claro o tipo de decisão modelado em cada caso.

Tabela 5 – Coeficientes de Correlação de Pearson dos Hiperparâmetros em Relação ao Aumento do F1-Score

Modelo	**criterion**	**min samples split**	**min samples leaf**	**max features**	**max depth**	**max leaf nodes**
DE-ISP (4x8 e 8x4)	-0,0038	0,0019	0,0351	0,2630	0,3600	0,4532
DE-ISP (Demais Blocos)	0,0152	0,0127	-0,0386	0,1865	0,3464	0,5051
DM-ISP	-0,0026	0,0032	0,0061	0,4337	0,1766	0,3196

Fonte: Elaborada pelo autor.

Os resultados, apresentados na Tabela 5, revelam quais hiperparâmetros tiveram maior influência sobre o desempenho dos modelos. Observa-se que os hiperparâmetros *max leaf nodes* e *max depth* apresentaram as maiores correlações positivas para ambos os modelos DE-ISP. Isso sugere que uma maior profundidade da árvore e um número maior de nós folha tendem a beneficiar a capacidade do modelo de capturar padrões complexos nesses cenários. Já no modelo DM-ISP, os hiperparâmetros *max features* e *max leaf nodes* demonstraram as maiores correlações positivas.

Com base nesses resultados, para cada modelo foram escolhidos os dois hiperparâmetros com maior coeficiente de correlação para refinamento via *Grid Search*. Os demais foram mantidos em seus valores padrão ou definidos com base nos melhores valores encontrados na busca inicial. Esse refinamento foi realizado utilizando validação cruzada para garantir que as combinações selecionadas resultassem na melhor configuração possível.

A Tabela 6 apresenta as combinações de valores de hiperparâmetros obtidas após o processo de refinamento para cada modelo, comparando-as com os valores padrão utilizados nas árvores de decisão. O valor padrão de cada hiperparâmetro serve como uma referência para avaliar o impacto das mudanças realizadas durante o processo de otimização. É importante destacar que, quando um hiperparâmetro está configu-

rado como **None**, isso indica que não há um limite específico para a quantidade de informação relacionada a esse parâmetro. Ou seja, None significa que a configuração não impõe restrições, permitindo que o modelo utilize toda a informação disponível ou tome decisões sem limitações adicionais.

Tabela 6 – Valores finais de Hiperparâmetros obtidos para cada Modelo

Modelo	**criterion**	**min samples split**	**min samples leaf**	**max features**	**max depth**	**max leaf nodes**
Padrão	*gini*	*2*	*1*	*None*	*None*	*None*
DE-ISP (4x8 e 8x4)	gini	2	80	None	15	210
DE-ISP (Demais Blocos)	gini	2	1	None	15	800
DM-ISP	gini	2	60	28	None	500

Fonte: Elaborada pelo autor.

É possível observar, através da Tabela 6, os ajustes realizados nos hiperparâmetros de cada modelo, comparados aos valores padrão das árvores de decisão. Os hiperparâmetros *criterion* e *min samples split* foram mantidos como gini e 2, respectivamente, em todos os modelos, de acordo com o padrão já estabelecidos nas árvores de decisão. Para o *min samples leaf*, o modelo DE-ISP (4x8 e 8x4) apresentou um ajuste para 80, já o modelo DE-ISP (Demais Blocos) manteve o valor padrão de 1. O modelo DM-ISP obteve um ajuste para 60. O parâmetro *max features* foi ajustado para 28 no modelo DM-ISP, limitando as características utilizadas, enquanto nos outros modelos manteve-se como None, permitindo o uso de todas as características disponíveis. O *max depth*, foi configurado como 15 para ambos os modelos de DE-ISP, enquanto para o modelo DM-ISP foi mantido o mesmo valor padrão de None. Por fim, o hiperparâmetro *max leaf nodes* obteve uma maior variação, sendo 210 para o modelo DE-ISP (4x8 e 8x4), 800 para DE-ISP (Demais Blocos) e 500 para DM-ISP, controlando a flexibilidade e a complexidade de cada modelo. Esses ajustes visam otimizar o desempenho, equilibrando a complexidade e a capacidade de generalização dos modelos.

Tabela 7 – Total de Nodos, Profundidade e Acurácias obtidas para cada Modelo

Modelo	**Configuração**	**Total de Nodos**	**Profundidade**	**Acurácia**
DE-ISP (4x8 e 8x4)	Padrão	33575	45	60%
	Final	419	15	63%
DE-ISP (Demais Blocos)	Padrão	187907	61	65%
	Final	1599	15	68%
DM-ISP	Padrão	146373	50	74%
	Final	999	16	79%

Fonte: Elaborada pelo autor.

A Tabela 7 apresenta os resultados relacionados à complexidade e desempenho

dos modelos, comparando os valores padrão com os valores obtidos após o refinamento. Para os modelos DE-ISP (4x8 e 8x4), DE-ISP (Demais Blocos) e DM-ISP, observou-se uma redução significativa no número de nós e na profundidade das árvores após os ajustes. Por exemplo, o modelo DE-ISP (4x8 e 8x4), inicialmente com 33575 nós e 45 de profundidade, foi reduzido para 419 nós e 15 de profundidade, resultando em um aumento da acurácia de 60% para 63%. De forma semelhante, o modelo DE-ISP (Demais Blocos) teve uma redução de 187907 nós e 61 de profundidade para 1599 nós e 15 de profundidade, com um aumento da acurácia de 65% para 68%. O modelo DM-ISP seguiu a mesma tendência, com uma diminuição de 146373 nós e 50 de profundidade para 999 nós e 16 de profundidade, resultando em uma melhoria na acurácia de 74% para 79%.

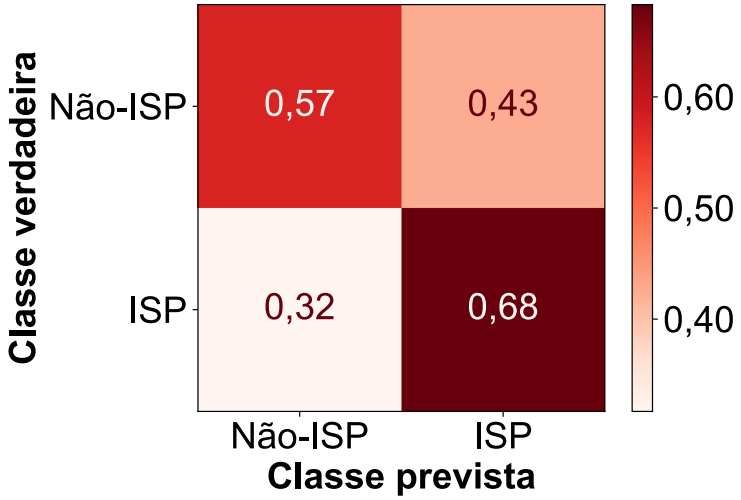
Esses ajustes indicam que, ao reduzir a complexidade dos modelos, com menos nós e menor profundidade, foi possível melhorar a acurácia, sugerindo uma maior generalização dos modelos. Além disso, ao final do processo, as árvores de decisão para todos os modelos foram extraídas utilizando os hiperparâmetros refinados, garantindo que a configuração final fosse a mais eficiente possível. Essa redução de complexidade também pode ser vista como uma estratégia para evitar o *overfitting*, tornando os modelos mais eficientes e robustos. A seguir, é possível observar uma análise mais detalhada do comportamento do modelo através das matrizes de confusão, que fornecem uma visão mais clara sobre os acertos e erros de classificação.

A Figura 14 apresenta a matriz de confusão do modelo final de Árvore de Decisão para classificação das classes Não-ISP e ISP da **DE-ISP** nos blocos de tamanho 4x8 e 8x4, avaliados sobre os 25% dos dados reservados para teste. A diagonal principal representa as predições corretas do modelo, enquanto os elementos fora da diagonal indicam erros de classificação. O modelo final obteve acurácias de 57% e 68% para as classes Não-ISP e ISP, respectivamente.

A Figura 15 apresenta a matriz de confusão do modelo final para os demais tamanhos de bloco. O modelo alcançou acurácias de 72% e 63% para as classes Não-ISP e ISP, respectivamente. No contexto da solução **DE-ISP**, os erros de predição podem ser classificados em dois tipos: **erros de tempo** e **erros de eficiência de codificação**. Um erro de tempo ocorre quando o modelo prediz erroneamente a classe ISP para um bloco que pertence à classe Não-ISP, levando à execução desnecessária do processo de RDO para os modos ISP. Já um erro de eficiência de codificação ocorre quando o modelo classifica erroneamente um bloco como Não-ISP, removendo indevidamente os modos ISP da lista de decisão, o que pode impactar a eficiência da codificação.

A análise das matrizes de confusão revela que, para os blocos 4x8 e 8x4, os erros de tempo ocorrem em 43% dos casos, enquanto os erros de eficiência de codificação representam 32%. Para os demais tamanhos de bloco, observa-se o comportamento oposto, com erros de tempo em 28% dos casos e erros de eficiência de codificação

Figura 14 – Matriz de confusão para o modelo final de árvore de decisão classificando as classes Não-ISP e ISP para os blocos de tamanho 4×8 e 8×4 no conjunto de teste



Fonte: Elaborada pelo autor.

em 37%.

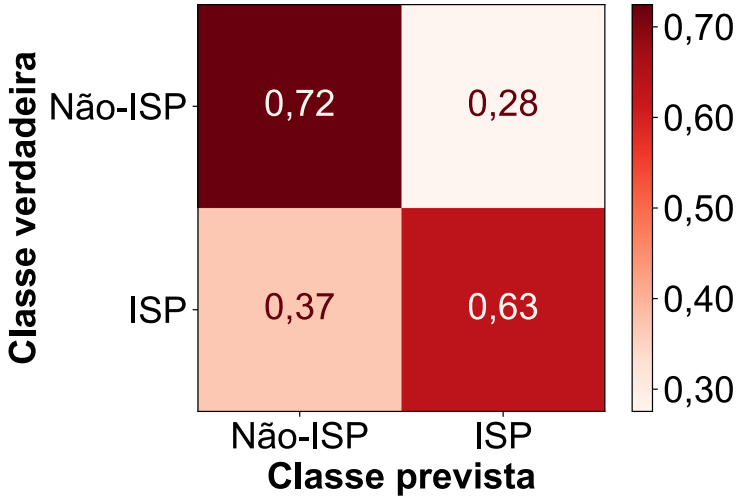
A Figura 16 apresenta a matriz de confusão do modelo final treinado para predizer entre as classes ISP Planar/DC e ISP Angular dentro da **DM-ISP**. O modelo obteve acurácias de 77% e 81% para as classes ISP Planar/DC e ISP Angular, respectivamente. Na solução **DM-ISP**, um erro de tempo ocorre quando um bloco da classe ISP Planar/DC é erroneamente classificado como ISP Angular. Nesse caso, não há perda de eficiência de codificação, mas há um aumento desnecessário no tempo de codificação, pois a avaliação RDO será realizada para um conjunto maior de candidatos ISP. Já um erro de eficiência de codificação acontece quando um bloco da classe ISP Angular é erroneamente classificado como ISP Planar/DC, levando à remoção indevida dos candidatos ISP associados aos modos angulares, reduzindo o tempo de codificação, mas podendo comprometer a eficiência da codificação.

A análise da Figura 16 revela que os erros de tempo ocorrem em 23% dos casos, enquanto os erros de eficiência de codificação representam 18%. Esses resultados demonstram que o modelo consegue um bom equilíbrio entre a redução do tempo de codificação e a pouca perda na eficiência da codificação.

7.4 Implementação no codificador VTM

Como mencionado anteriormente, os modelos de decisão foram desenvolvidos em Python, utilizando a biblioteca Scikit-learn para a construção e treinamento das árvores de decisão. Essa abordagem inicial em Python permitiu a criação de um modelo flexível e fácil de ajustar. No entanto, para integrar esses modelos ao VTM, que é implementado em C++, foi necessário traduzir as árvores de decisão geradas em

Figura 15 – Matriz de confusão para o modelo final de árvore de decisão classificando as classes Não-ISP e ISP para os demais tamanhos de bloco no conjunto de teste

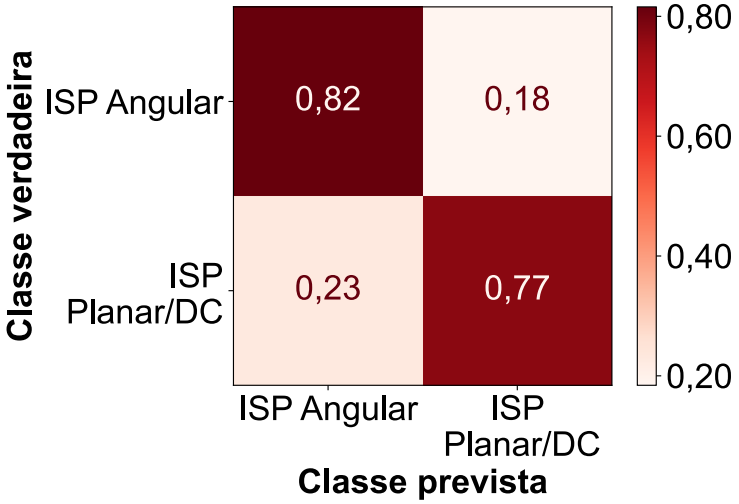


Fonte: Elaborada pelo autor.

Python para a linguagem C++ e adicionar as funções correspondentes no código do VTM. Essa tradução foi necessária para garantir a implementação do esquema de decisão no ambiente do VTM, permitindo que as árvores de decisão fossem aplicadas diretamente durante o processo de codificação dos vídeos, sem a necessidade de um sistema de integração entre as duas linguagens.

Além disso, para avaliar a eficácia e eficiência do esquema proposto, realizou-se experimentos comparando o VTM na versão 18.0 padrão com a versão modificada, que incorpora a solução proposta. Para essas análises, foram utilizados vídeos da *Common Test Condition* (CTC) (Bossen et al., 2020) do VVC, que serviram como base para os testes. O tempo de execução de ambas as versões foi registrado para permitir uma comparação direta. Com esses dados, foi possível avaliar não apenas a eficiência da codificação, mas também o impacto da solução proposta no tempo de processamento, considerando o ganho ou a perda de desempenho com a inclusão do esquema de decisão.

Figura 16 – Matriz de confusão do modelo final de Árvore de Decisão para classificação das classes ISP Planar/DC e ISP Angular no conjunto de teste



Fonte: Elaborada pelo autor.

8 RESULTADOS

Este Capítulo apresenta a avaliação do desempenho do esquema MLISP, proposto neste trabalho, e de seus subproblemas isoladamente, com foco na redução do tempo de codificação. Os resultados foram obtidos a partir de experimentos conduzidos conforme as CTC do VVC, utilizando a configuração All Intra e diferentes valores de QP (22, 27, 32 e 37). A análise busca compreender os impactos da abordagem proposta em termos de eficiência computacional e taxa de compressão, comparando seus efeitos em diferentes cenários de codificação.

A solução implementada no codificador VTM 18.0 é composta por dois subproblemas principais, denominados **DE-ISP** e **DM-ISP**. O DE-ISP tem como objetivo identificar situações em que os modos ISP possuem baixa probabilidade de serem selecionados como o melhor modo. Nessas circunstâncias, esses modos são removidos da RD-List, reduzindo a complexidade computacional sem comprometer significativamente a eficiência da compressão. Caso o DE-ISP indique que os modos ISP ainda podem ser escolhidos, o DM-ISP é acionado para determinar se o melhor modo pertence à classe ISP Angular ou ISP Planar/DC. Se a decisão for pela classe ISP Planar/DC, os modos ISP associados a modos angulares são eliminados, garantindo uma maior economia de tempo de codificação. Dessa forma, o esquema MLISP opera em duas etapas para reduzir o esforço computacional da predição intra-quadro.

Para avaliar o desempenho da abordagem proposta, foram codificadas 22 sequências de vídeo apresentadas na Tabela 8 seguindo as recomendações da CTC do VVC. O conjunto de vídeos abrange uma variedade de resoluções e classes, sendo composto por seis vídeos com resolução 3840×2160 (três da classe A1 e três da classe A2), cinco vídeos com resolução 1920×1080 (classe B), quatro vídeos com resolução 832×480 (classe C), quatro vídeos com resolução 416×240 (classe D) e três vídeos com resolução 1280×720 (classe E). Essa diversidade permite avaliar o impacto do esquema MLISP em diferentes cenários de codificação, considerando tanto vídeos de alta definição quanto aqueles com resoluções mais baixas.

Ao todo, foram realizados 264 experimentos, considerando as 22 sequências de vídeo, os quatro valores de QP e as três configurações avaliadas (DE-ISP, DM-ISP e

MLISP completo). Para garantir que os tempos de codificação não fossem influenciados por variações no ambiente de execução, os testes foram conduzidos de forma sequencial, executando um vídeo por vez em um servidor dedicado equipado com processador Intel® Core™ i7-8700K e 16 GB de RAM. Além disso, para assegurar uma avaliação justa, nenhum dos vídeos utilizados nos testes foi empregado no treinamento dos modelos de aprendizado de máquina do MLISP, garantindo que os resultados refletissem um desempenho realista em dados não vistos previamente.

Tabela 8 – Sequências de Vídeo para testes de acordo com a CTC do VVC

Classe Resolução	**Sequência**	**Número de Frames**	**FPS**	**Bit Depth**
A1 3840x2160	Tango2	294	60	10
	FoodMarket4	300	60	10
	Campfire	300	30	10
A2 3840x2160	CatRobot	300	60	10
	DaylightRoad2	300	60	10
	ParkRunning3	300	60	10
B 1920x1080	MarketPlace	600	60	10
	RitualDance	600	60	10
	Cactus	500	50	8
	BasketballDrive	500	50	8
	BQTerrace	600	60	8
C 832x480	RaceHorses	300	30	8
	BQMall	600	60	8
	PartyScene	500	50	8
	BasketballDrill	500	50	8
D 416x240	RaceHorses	300	30	8
	BQSquare	600	60	8
	BlowingBubbles	500	50	8
	BasketballPass	500	50	8
E 1280x720	FourPeople	600	60	8
	Johnny	600	60	8
	KristenAndSara	600	60	8

Fonte: Elaborada pelo autor.

A análise dos resultados considerou duas métricas principais: *Time Saving* (TS) e *Bjøntegaard Delta Bit Rate* (BD-BR). O TS indica a redução percentual no tempo de codificação ao comparar o VTM modificado com o VTM âncora, ou seja, quanto tempo foi economizado pela solução proposta. Já o BD-BR mede a perda ou ganho na eficiência de codificação do vídeo, onde valores positivos indicam um aumento na taxa de bits necessária para manter a mesma qualidade visual, e valores negativos

Tabela 9 – Resultados de redução de tempo e eficiência de codificação

		DE-ISP		DM-ISP		MLISP	
Classe	**Vídeo**	**TS**	**BD-BR**	**TS**	**BD-BR**	**TS**	**BD-BR**
A1	Tango2	9,51%	0,09%	6,98%	0,08%	10,86%	0,10%
	FoodMarket4	7,85%	0,05%	6,41%	0,05%	9,84%	0,09%
	Campfire	10,60%	0,03%	8,61%	0,05%	12,05%	0,08%
A2	CatRobot	8,43%	0,16%	6,71%	0,15%	11,09%	0,23%
	DaylightRoad2	10,49%	0,23%	7,82%	0,12%	12,65%	0,26%
	ParkRunning3	6,49%	0,03%	5,89%	0,04%	8,84%	0,05%
B	MarketPlace	10,36%	0,08%	10,21%	0,09%	13,19%	0,11%
	RitualDance	7,92%	0,17%	6,98%	0,21%	10,20%	0,25%
	Cactus	9,19%	0,21%	8,15%	0,25%	12,28%	0,34%
	BasketballDrive	9,48%	0,38%	6,38%	0,20%	11,33%	0,46%
	BQTerrace	7,39%	0,22%	4,35%	0,12%	9,86%	0,28%
C	RaceHorsesC	9,16%	0,22%	8,89%	0,20%	11,96%	0,29%
	BQMall	8,30%	0,37%	7,10%	0,36%	9,29%	0,59%
	PartyScene	8,48%	0,27%	7,73%	0,20%	11,84%	0,36%
	BasketballDrill	8,81%	0,64%	6,50%	0,37%	11,04%	0,67%
D	RaceHorses	9,09%	0,13%	9,20%	0,13%	11,20%	0,25%
	BQSquare	6,13%	0,21%	6,49%	0,22%	10,39%	0,35%
	BlowingBubbles	8,92%	0,35%	7,24%	0,27%	12,14%	0,49%
	BasketballPass	7,85%	0,34%	6,27%	0,27%	10,05%	0,53%
E	FourPeople	7,43%	0,30%	6,29%	0,31%	10,13%	0,48%
	Johnny	7,79%	0,26%	4,19%	0,30%	10,69%	0,41%
	KristenAndSara	7,96%	0,18%	5,76%	0,22%	10,34%	0,36%
	Média	**8,53%**	**0,22%**	**7,01%**	**0,19%**	**10,97%**	**0,32%**

Fonte: Elaborada pelo autor.

representam uma melhoria na compressão.

A Tabela 9 apresenta os resultados obtidos pelo esquema MLISP e de suas soluções complementares, DE-ISP e DM-ISP. Com base nesses dados, é possível avaliar o impacto de cada abordagem na redução do tempo de codificação e na eficiência de codificação. A seguir, são analisados os principais resultados, destacando os cenários de melhor e pior desempenho observados para cada solução.

A solução DE-ISP conseguiu uma redução média de 8,53% no tempo de codificação, com uma perda de apenas 0,22% em BD-BR. O melhor resultado foi obtido no vídeo Campfire, que apresentou uma diminuição de 10,60% no tempo de codificação, com um impacto mínimo na eficiência, com apenas 0,23% de perda. Esse desempenho se deve, principalmente, ao fato de a solução DE-ISP ter predito com mais frequência a classe Não-ISP. Isso levou à remoção de modos ISP da RD-List e, consequentemente, a diminuição no número de avaliações, otimizando o tempo de processamento.

Apesar dos bons resultados obtidos em várias sequências de vídeo, a solução não teve o mesmo desempenho em todas elas. O pior resultado foi observado no vídeo

BQSquare, que teve uma redução de 6,13% no tempo de codificação, mas com uma perda de 0,21% na eficiência de codificação. Esse comportamento pode ser explicado pela alta frequência com que a solução DE-ISP previu a classe ISP. Como resultado, houve menos oportunidades para evitar a avaliação dos modos ISP, o que acabou limitando a economia de tempo que poderia ser alcançada.

A solução DM-ISP, por sua vez, obteve uma redução média de 7,01% no tempo de codificação, com uma perda média de 0,19% na eficiência de codificação. O vídeo que obteve o melhor desempenho com esta solução foi o MarketPlace, que apresentou uma redução de 10,21% no tempo de codificação, com uma perda de apenas 0,09% na eficiência. Esse resultado pode ser atribuído à predominância de texturas suaves no vídeo, que favorecem os modos Planar/DC. Como consequência, a solução DM-ISP pôde eliminar com mais frequência os modos ISP associados a modos angulares, contribuindo para uma redução significativa do tempo de processamento.

Por outro lado, o vídeo Johnny apresentou o pior desempenho dentro dessa solução, com uma redução de tempo de apenas 4,19%, acompanhada de uma perda de 0,30% na eficiência de codificação. Esse resultado pode ser explicado pela natureza da textura do vídeo, que é mais favorável aos modos angulares. Como a solução DM-ISP previu com maior frequência a classe ISP Angular, a remoção dos modos ISP teve um impacto menor, o que limitou o potencial de a economia de tempo que poderia ser alcançado.

A combinação das soluções DE-ISP e DM-ISP no esquema MLISP teve como objetivo maximizar a redução de tempo de codificação, equilibrando a remoção de modos desnecessários sem comprometer significativamente a eficiência da compressão. Como resultado, o MLISP apresentou um desempenho superior às suas soluções individuais, alcançando uma redução média de 10,97% no tempo de codificação, com uma perda de 0,32% na eficiência de codificação. O melhor desempenho foi novamente observado no vídeo MarketPlace, que apresentou uma redução de 13,19% no tempo de codificação, com uma perda mínima de 0,11% de eficiência de codificação. Esse resultado reforça a eficiência da abordagem, especialmente para vídeos cujas características de textura favorecem a remoção seletiva de modos ISP.

Em todos os casos, a solução MLISP superou DE-ISP e DM-ISP no quesito redução de tempo. A média geral de 10,97% é superior aos 8,53% da DE-ISP e aos 7,01% da DM-ISP, evidenciando que a combinação das duas abordagens realmente melhora a eficiência temporal do codificador. O MLISP, no entanto, teve a maior perda de eficiência (0,32% de BD-BR), o que já era esperado, pois ele maximiza a redução de tempo ao custo de um leve aumento da taxa de bits. Mesmo assim, essa perda é relativamente pequena e pode ser considerada um compromisso aceitável para a redução do tempo de processamento.

Embora a maioria dos vídeos tenha apresentado uma redução de tempo superior

a 10%, alguns cenários tiveram desempenhos menores. O pior resultado foi apresentado para o vídeo ParkRunning3, que obteve uma redução de 8,84% no tempo de codificação, com uma perda de 0,05% na eficiência de codificação. Mesmo nesse cenário, a baixa perda de eficiência de codificação indica que, embora a redução de tempo tenha sido menor, o impacto na qualidade da compressão permaneceu praticamente imperceptível.

Ao analisar os resultados por classe de vídeo, observa-se que o esquema MLISP apresentou um desempenho especialmente promissor para vídeos de alta definição, pertencentes às classes A1, A2 e B. Nessas classes, não só foi observada uma redução significativa no tempo de codificação, mas também as perdas de eficiência de codificação permaneceram em níveis baixos, geralmente abaixo de 0,30%. Por exemplo, os vídeos FoodMarket4 e Campfire apresentaram perdas de apenas 0,09% e 0,08%, respectivamente. Isso sugere que a solução proposta é particularmente eficiente para vídeos em resolução 4K e 1080p, nos quais a complexidade da codificação tende a ser maior e, consequentemente, a redução de tempo obtida tem um impacto mais relevante.

8.1 Comparação com Trabalhos Relacionados

A fim de contextualizar os resultados obtidos pelo esquema proposto, esta Seção apresenta uma comparação detalhada entre o MLISP e trabalhos relacionados que também abordam a decisão sobre a avaliação dos modos ISP. Para isso, a Tabela 10 resume os principais resultados em termos de redução de tempo (TS), impacto na eficiência de codificação medido pelo BD-BR, a razão TS/BD-BR e a versão do VTM utilizada nos experimentos de cada estudo. É importante considerar a diferença nas versões do VTM, as versões mais novas são mais otimizadas, devido a isso os ganhos em tempo de codificação tendem a ser mais difíceis de acontecer.

Os trabalhos selecionados para essa comparação foram aqueles previamente discutidos no Capítulo 4 que apresentavam resultados específicos para os modos ISP. Essa seleção é fundamental, pois alguns estudos tratam do ISP juntamente com outras ferramentas, como o IBC, mas reportam apenas resultados globais da solução completa, sem detalhar o impacto individual do ISP. Como o objetivo desta dissertação é analisar a tomada de decisão específica para os modos ISP, esses trabalhos foram excluídos da comparação. Essa decisão garante que a análise seja precisa, evitando interpretações imprecisas ao comparar abordagens que apresentam métricas específicas do ISP com aquelas que fornecem apenas resultados de soluções completas.

A análise da Tabela 10 evidencia que o esquema MLISP apresenta os melhores resultados em redução de tempo quando comparado aos trabalhos de (Park et al.,

Tabela 10 – Comparação com trabalhos relacionados

Solução	VTM	TS	BD-BR	TS/BD-BR
MLISP (Nosso)	**VTM 18.0**	**10,97%**	**0,32%**	**34,28**
(Park et al., 2022)	VTM 11.0	7,20%	0,08%	90,00
(Saldanha et al., 2021)	VTM 10.0	8,32%	0,31%	26,84
(Liu et al., 2021)	VTM 8.0	7,00%	0,09%	77,78
(Park; Kim; Jeon, 2020)	VTM 9.0	12,11%	0,43%	28,16

Fonte: Elaborada pelo autor.

2022), (Saldanha et al., 2021) e (Liu et al., 2021). Esse resultado era esperado, uma vez que esses trabalhos têm como foco principal a identificação de blocos nos quais a avaliação dos modos ISP pode ser evitada. Em contrapartida, o MLISP propõe uma abordagem mais abrangente, que não apenas determina se os modos ISP devem ser avaliados, mas também identifica a classe de modos intra mais promissora para essa avaliação, distinguindo entre as categorias **ISP Planar/DC** e **ISP Angular**. Essa estratégia combinada de **Decisão de Evitar ISP** e **Decisão de Modo ISP** permite ao MLISP atingir um maior potencial de redução de tempo ao abordar dois subproblemas dentro do processo de decisão dos modos intra no ISP.

Outro ponto relevante diz respeito à eficiência de codificação. O trabalho de (Park; Kim; Jeon, 2020) alcança uma redução de tempo superior à do MLISP, porém, essa vantagem vem acompanhada de uma perda mais expressiva na eficiência de codificação. Dessa forma, o esquema MLISP demonstra um melhor equilíbrio entre os critérios de redução de tempo e eficiência de codificação, aspecto refletido na métrica TS/BD-BR.

Adicionalmente, vale destacar que tanto o MLISP quanto a abordagem proposta em (Park et al., 2022) são os únicos métodos que consideram a extração de características de imagem não apenas para o bloco completo, mas também para suas subpartições. Esse aspecto é particularmente importante, visto que a ferramenta ISP realiza predições no nível das subpartições, e não do bloco como um todo. No entanto, enquanto (Park et al., 2022) emprega a soma absoluta média dos coeficientes de transformação como métrica para avaliar as subpartições, o MLISP adota uma abordagem mais simplificada, utilizando a variância do bloco. Essa escolha se justifica pelo fato de que a métrica utilizada em (Park et al., 2022) requer a soma de todos os coeficientes transformados dentro do bloco, tornando o cálculo computacionalmente mais oneroso. Por outro lado, o MLISP consegue obter uma redução significativa no tempo de codificação utilizando características mais simples e eficientes, mantendo métricas competitivas de BD-BR e TS/BD-BR.

Dessa forma, os resultados apresentados indicam que o esquema MLISP oferece

um compromisso vantajoso entre redução de tempo e impacto na eficiência de codificação, ao mesmo tempo que mantém uma abordagem computacionalmente viável. Essa combinação de fatores reforça a relevância da proposta e seu potencial de aplicação prática em cenários reais de codificação de vídeo.

9 CONCLUSÃO

A constante evolução da era digital trouxe consigo a crescente demanda por soluções mais eficientes de compressão de vídeo, dado o aumento exponencial do consumo de conteúdo audiovisual. Nesse cenário, o VVC se destaca como um dos padrões mais avançados, oferecendo uma significativa redução na taxa de bits sem comprometer a qualidade visual. Entre as inovações desse padrão, os modos ISP ganham destaque na predição intra-quadro. No entanto, apesar dos avanços, a implementação do ISP no VVC impõe desafios substanciais no que diz respeito ao esforço computacional necessário para a decisão do modo intra. O processo de decisão, que envolve a avaliação de diversas combinações por meio do RDO, é complexo e exige otimizações contínuas para garantir eficiência.

Neste trabalho, foi abordada a utilização de técnicas de aprendizado de máquina para otimizar o processo de decisão do modo intra. A ideia central foi aplicar modelos de ML para prever os modos ISP mais apropriados, com base nas características extraídas do processo de codificação ou da própria imagem. Ao fazer isso, foi possível reduzir o esforço computacional envolvido, sem prejudicar significativamente a qualidade da compressão. As análises do ISP revelam que, dada a alta taxa de ocorrência da classe Não-ISP e a frequência dos modos ISP Planar/DC, a utilização de abordagens baseadas em aprendizado de máquina para prever quando a avaliação dos modos ISP pode ser evitada e, quando não, quando a avaliação de candidatos ISP Angulares pode ser ignorada, mostra-se promissora.

Duas soluções complementares foram propostas para otimizar a decisão do modo ISP: a **DE-ISP**, que utiliza Árvores de Decisão treinadas com características da imagem para predizer quando a avaliação do modo ISP pode ser ignorada; e a **DM-ISP**, que aplica uma Árvore de Decisão treinada com características de codificação para selecionar as classes de modos intra mais adequadas para avaliação. Os resultados experimentais mostraram a eficácia de ambas as soluções individuais, assim como do esquema completo **MLISP**, que combina as duas abordagens. Esse esquema apresentou uma redução significativa no tempo de codificação, mantendo perdas mínimas na eficiência da compressão, quando comparado a soluções existentes.

Este trabalho contribui para a área de compressão de vídeo ao integrar o aprendizado de máquina no processo de decisão do modo intra, oferecendo uma solução inovadora que balanceia a redução de tempo de codificação com a preservação da qualidade de compressão. Ao combinar características simples da imagem com características de codificação, conseguimos um desempenho competitivo, destacando o potencial das técnicas de aprendizado de máquina na melhoria de codificadores de vídeo modernos.

Embora os resultados obtidos sejam satisfatórios, ainda há espaço para melhorias. Uma possibilidade de trabalho futuro é a extração de novas características de imagem, capazes de capturar de forma mais precisa as características visuais dos quadros, o que pode melhorar a predição dos modos ISP. Além disso, testar modelos de árvores de decisão mais simples, como o LightGBM, pode ser uma alternativa interessante, oferecendo uma boa relação entre desempenho e eficiência computacional. Outra possibilidade seria explorar outros tipos de modelos, como SVM ou modelos bayesianos, que podem trazer melhorias na acurácia da predição, especialmente ao lidar com dados mais complexos. Essas abordagens podem contribuir para uma maior eficiência na decisão do modo e, consequentemente, reduzir o esforço computacional.

10 TRABALHOS SUBMETIDOS E PUBLICADOS

10.1 Fast ISP Mode Decision for the Versatile Video Coding Intra Prediction Using Machine Learning

Larissa Araújo, Adson Duarte, Bruno Zatt, Guilherme Correa, Daniel Palomino

Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia), 2024.

Qualis A4.

10.2 MLISP: Machine-Learning-based ISP Decision Scheme for VVC Encoders (submetido)

Larissa Araújo, Adson Duarte, Bruno Zatt, Guilherme Correa, Daniel Palomino Journal of the Brazilian Computer Society (JBCS), 2025.

Qualis A2.

REFERÊNCIAS

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **Journal of machine learning research**, [S.l.], v.13, n.2, p.281–305, 2012.

BJONTEGAARD, G. **Calculation of average PSNR differences between RD-curves**. 2001. VCEG-M33. Disponível em: <https://www.itu.int/wftp3/av-arch/video-site/0104_Aus/VCEG-M33.doc>.

BOSSEN, F.; BOYCE, J.; SüHRING, K.; LI, X.; SEREGIN, V. **VTM common test conditions and software reference configurations for SDR video**. 2020. JVET-T2010-v1. Disponível em: <https://jvet-experts.org/doc$_{e}nd_{u}ser/current_{d}ocument.php?id = 10545 >$.

BOSSEN, F.; SUEHRING, K.; LI, X. **VTM reference software for VVC**. 2018. Disponível em: <https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM>.

BROSS, B.; WANG, Y.-K.; YE, Y.; LIU, S.; CHEN, J.; SULLIVAN, G. J.; OHM, J.-R. Overview of the Versatile Video Coding (VVC) Standard and its Applications. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v.31, n.10, p.3736–3764, 2021.

CECI, L. **Live streaming - Statistics & Facts**. 2023. Disponível em: <https://www.statista.com/topics/8906/live-streaming/#topicOverview>.

CHANG, Y.-J.; JHU, H.-J.; JIANG, H.-Y.; ZHAO, L.; ZHAO, X.; LI, X.; LIU, S.; BROSS, B.; KEYDEL, P.; SCHWARZ, H.; MARPE, D.; WIEGAND, T. Multiple Reference Line Coding for Most Probable Modes in Intra Prediction. In: DATA COMPRESSION CONFERENCE (DCC), 2019., 2019, Snowbird, UT, USA. **Anais...** IEEE, 2019. p.559–559.

DE-LUXáN-HERNáNDEZ, S.; GEORGE, V.; MA, J.; NGUYEN, T.; SCHWARZ, H.; MARPE, D.; WIEGAND, T. An Intra Subpartition Coding Mode for VVC. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.1203–1207.

DE-LUXáN-HERNáNDEZ, S.; SCHWARZ, H.; MARPE, D.; WIEGAND, T. Fast Line-Based Intra Prediction for Video Coding. In: IEEE INTERNATIONAL SYMPOSIUM ON MULTIMEDIA (ISM), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.135–138.

DONG, X.; SHEN, L.; YU, M.; YANG, H. Fast Intra Mode Decision Algorithm for Versatile Video Coding. **IEEE Transactions on Multimedia**, [S.l.], v.24, p.400–414, 2022.

DUARTE, A.; ZATT, B.; CORREA, G.; PALOMINO, D. Fast Intra Mode Decision Using Machine Learning for the Versatile Video Coding Standard. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), 2023., 2023, Monterey, CA, USA. **Anais...** IEEE, 2023. p.1–5.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The WEKA data mining software: an update. **SIGKDD Explor. Newsl.**, New York, NY, USA, v.11, n.1, p.10–18, Nov. 2009.

LIU, Z.; DONG, M.; GUAN, X.; ZHANG, M.; WANG, R. Fast ISP coding mode optimization algorithm based on CU texture complexity for VVC. **EURASIP Journal on Image and Video Processing**, [S.l.], v.2021, 07 2021.

PARK, J.; KIM, B.; JEON, B. Fast VVC intra prediction mode decision based on block shapes. In: APPLICATIONS OF DIGITAL IMAGE PROCESSING XLIII, 2020, Basel, Switzerland. **Anais...** SPIE, 2020. v.11510, p.115102H.

PARK, J.; KIM, B.; JEON, B. Fast VVC Intra Subpartition based on Position of Reference Pixels. In: INTERNATIONAL CONFERENCE ON ELECTRONICS, INFORMATION, AND COMMUNICATION (ICEIC), 2022., 2022. **Anais...** [S.l.: s.n.], 2022. p.1–2.

PARK, J.; KIM, B.; LEE, J.; JEON, B. Machine Learning-Based Early Skip Decision for Intra Subpartition Prediction in VVC. **IEEE Access**, [S.l.], v.10, p.111052–111065, 2022.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY Édouard. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, [S.l.], v.12, n.85, p.2825–2830, 2011.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence**: a modern approach. 3.ed. [S.l.]: Pearson, 2009.

SALDANHA, M.; SANCHEZ, G.; MARCON, C.; AGOSTINI, L. Complexity Analysis Of VVC Intra Coding. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.3119–3123.

SALDANHA, M.; SANCHEZ, G.; MARCON, C.; AGOSTINI, L. Learning-Based Complexity Reduction Scheme for VVC Intra-Frame Prediction. In: INTERNATIONAL CONFERENCE ON VISUAL COMMUNICATIONS AND IMAGE PROCESSING (VCIP), 2021., 2021, Munich, Germany. **Anais...** IEEE, 2021. p.1–5.

SIQUEIRA, I.; CORREA, G.; GRELLERT, M. Rate-Distortion and Complexity Comparison of HEVC and VVC Video Encoders. In: IEEE 11TH LATIN AMERICAN SYMPOSIUM ON CIRCUITS & SYSTEMS (LASCAS), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.1–4.

SULLIVAN, G.; WIEGAND, T. Rate-distortion optimization for video compression. **IEEE Signal Processing Magazine**, [S.l.], v.15, n.6, p.74–90, November 1998.

TAN, P.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. [S.l.]: Pearson Addison Wesley, 2006. (Always learning).

Video Coding Concepts. [S.l.]: John Wiley Sons, Ltd, 2010. p.25–79.

ZHAO, L.; ZHANG, L.; MA, S.; ZHAO, D. Fast mode decision algorithm for intra prediction in HEVC. In: VISUAL COMMUNICATIONS AND IMAGE PROCESSING (VCIP), 2011., 2011, Tainan, Taiwan. **Anais...** IEEE, 2011. p.1–4.