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

Dissertação

Redução de Complexidade do Processo de Decisão de Modo da Predição Intra-Quadro do Codificador de Vídeo VVC utilizando Aprendizado de Máquina

Adson Ileon Ripinski Duarte

Pelotas, 2021

Adson Ileon Ripinski Duarte

Redução de Complexidade do Processo de Decisão de Modo da Predição Intra-Quadro do Codificador de Vídeo VVC utilizando Aprendizado de Máquina

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. Guilherme Correa
Prof. Dr. Bruno Zatt

Pelotas, 2021

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

D812r Duarte, Adson Ileon Ripinski

Redução de complexidade do processo de decisão de modo da predição intra-quadro do codificador de vídeo VVC utilizando aprendizado de máquina / Adson Ileon Ripinski Duarte ; Daniel Palomino, orientador ; Guilherme Correa, Bruno Zatt, coorientadores. — Pelotas, 2021.

107 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2021.

1. Versatile video coding. 2. Predição intra-quadro. 3. Modo de decisão. 4. Redução de complexidade. 5. Aprendizado de máquina supervisionado. I. Palomino, Daniel, orient. II. Correa, Guilherme, coorient. III. Zatt, Bruno, coorient. IV. Título.

CDD : 005

Elaborada por Maria Inez Figueiredo Figas Machado CRB: 10/1612

Adson Ileon Ripinski Duarte

Redução de Complexidade do Processo de Decisão de Modo da Predição Intra-Quadro do Codificador de Vídeo VVC utilizando Aprendizado de Máquina

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: 31 de agosto de 2021

Banca Examinadora:

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

Prof. Dr. Claudio Machado Diniz
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Marcelo Porto
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Ricardo Araujo
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

RESUMO

DUARTE, Adson Ileon Ripinski. **Redução de Complexidade do Processo de Decisão de Modo da Predição Intra-Quadro do Codificador de Vídeo VVC utilizando Aprendizado de Máquina**. Orientador: Daniel Palomino. 2021. 107 f. Dissertação (Mestrado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2021.

Este trabalho apresenta uma solução para redução de complexidade do processo de decisão de modo da predição intra do codificador *Versatile Video Coding* (VVC). Para atingir esta redução em complexidade, a ordem em que os modos são avaliados no processo *Rate-Distortion Optimization* (RDO) foi alterada de forma a dar prioridade a modos de predição intra que possuam mais chances de serem escolhidos como o melhor ao final do processo de decisão. Ao fazer isto, foi possível inserir modelos de aprendizado de máquina supervisionado baseados em árvores de decisão binárias capazes de detectar quando é possível desconsiderar a avaliação de alguns tipos de modos de predição intra-quadro baseado nos custos taxa-distorção dos modos que já foram avaliados até o momento. Um total de três modelos baseados em árvores de decisão foram desenvolvidos. O primeiro e o segundo modelo foram desenvolvidos com o objetivo de buscar predizer quando os modos de predição intra Angulares ou *Matrix-based Intra Prediction* (MIPs) podem ser desconsiderados da avaliação do RDO sem que sejam inseridas perdas significativas em eficiência de codificação. O terceiro modelo foi desenvolvido para agir nos casos onde o primeiro e o segundo modelo não foram capazes de trazer uma redução de complexidade, buscando explorar a possibilidade de desconsiderar a avaliação dos modos intra MIPs sem que perdas significativas em eficiência de codificação sejam inseridas. Para o primeiro, segundo e terceiro modelos foram obtidas F1-scores de 85,27%, 78,69% e 84,80%. Como resultados, obteve-se médias de 10,87% de redução de complexidade e 0,39% de BD-BR. Ao comparar o trabalho desenvolvido com trabalhos relacionados, pode-se notar que ainda que o método desenvolvido possua reduções de complexidade inferiores em alguns casos, o mesmo obtém perdas em eficiência de codificação inferiores em todos os casos, mesmo considerando todos os modos intra disponíveis no VVC.

Palavras-chave: Versatile Video Coding. Predição intra-quadro. Modo de Decisão. Redução de Complexidade. Aprendizado de Máquina Supervisionado.

ABSTRACT

DUARTE, Adson Ileon Ripinski. **Low Complexity VVC Intra Mode Decision using Machine Learning**. Advisor: Daniel Palomino. 2021. 107 f. Dissertation (Masters in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2021.

This work presents a solution for reducing the complexity of the intra mode decision for the Versatile Video Coding (VVC). To achieve this complexity reduction, the order in which modes are evaluated in the Rate-Distortion Optimization (RDO) process has been changed in order to give priority to intra prediction modes that are more likely to be chosen as the best at the end of the decision process. By doing this, it was possible to insert supervised machine learning models based on binary decision trees to predict when it is possible to skip the evaluation of some types of intra modes based. A total of three decision tree-based models were developed. The first and second models were developed with the objective of predicting when Angular intra prediction modes or Matrix-based Intra Prediction modes (MIPs) can be skip from the RDO evaluation without introducing significant losses in coding efficiency. The third model was developed to act in cases where the first and second models were not able to skip the evaluation of any types of modes, seeking to predict if MIP modes should be evaluated or not. For the first, second and third models, accuracies of 85.27%, 78.69% and 84.80% were obtained. As a result, we obtained averages of 10.87% on complexity reduction and 0.39% of BD-BR increase. When compared with related works, it can be noticed that even though the developed method has lower complexity reductions in some cases, it obtains lower coding efficiency losses in all cases, even when considering all intra modes available in the VVC.

Keywords: Versatile Video Coding. Intra Prediction. Mode Decision. Complexity Reduction. Supervised Machine Learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AI	All Intra
ALF	Adaptive Loop Filter
BD-BR	Bjøntegaard Delta-Bit Rate
BT	Binary Tree
CTC	Common Test Conditions
CU	Coding Unit
CTU	Coding Tree Unit
CV	Cross-Validation
DF	Deblocking Filter
DCT	Discrete Cosine Transform
DST	Discrete Sine Transform
HEVC	High Efficiency Video Coding
HM	HEVC Test Model
ISP	Intra Sub-Partitions Prediction
LCU	Largest Coding Unit
LD	Low Delay
LD-P	Low Delay Picture
MAD	Mean of Absolute Differences
MIP	Matrix-based Intra Prediction
MPM	Most Probable Modes
MRL	Multiple Reference Line
MTT	Multi-type Tree
PU	Prediction Unit
QP	Quantization Parameter
QT	Quaternary Tree
RA	Random Access

RF	Random Forest
RDO	Rate-distortion Otimization
RMD	Rough Mode Decision
SAD	Sum of Absolute Differences
SAO	Sample Adaptive Offset
SATD	Sum of Absolute Transformed Differences
TT	Ternary Tree
TS	Time Saving

SUMÁRIO

1 INTRODUÇÃO

O consumo de vídeos na internet tem se popularizado cada vez mais nos últimos anos. Um estudo realizado pela CISCO apresentou que em 2016 o fluxo de dados na internet relacionados a transmissão de vídeo correspondeu a 73% do tráfego total e que até 2022 existe uma estimativa deste valor subir para 82% (CISCO, 2019). Serviços como Netflix e YouTube, por exemplo, têm crescido muito nos últimos anos. Estatísticas do YouTube mostram que diariamente são assistidas mais de um bilhão de horas de vídeo (YOUTUBE, 2020) enquanto que a Netflix, no terceiro trimestre de 2019, já possuía 158 milhões de assinantes (WATSON, 2019).

Dada esta alta demanda por conteúdos de vídeo na internet, surge o problema de que vídeos sem compressão são impraticáveis. Um vídeo sem compressão, na resolução 1920x1080 pixels a 30 quadros por segundo, onde cada pixel contém 24 bits, necessitaria de uma taxa de transmissão de 1,5 Gbps. Para armazenar duas horas deste mesmo vídeo, o tempo de um filme, seriam necessários $\approx$1TB. Desta forma, a área de codificação de vídeo desempenha um papel importante ao estudar e desenvolver técnicas que realizam o processo de compressão de um vídeo com baixo impacto em sua qualidade, com o objetivo de reduzir a quantidade de dados que precisa ser armazenada e/ou transmitida. Para atingir este objetivo, diversos padrões de codificação de vídeo já foram desenvolvidos. Dentre eles, está o *Versatile Video Coding* (VVC), padrão que teve seu desenvolvimento concluído em meados de julho de 2020 pelo grupo *Joint Video Experts Team*, e será o foco deste trabalho. O objetivo do desenvolvimento do VVC é aumentar ainda mais as taxas de compressão quando comparado com o padrão anterior *High Efficiency Video Coding* (HEVC) mantendo a mesma qualidade visual.

Para atingir tal objetivo, o VVC incorpora muitas novas ferramentas e algoritmos de codificação. Somente na predição intra, o número de modos de predição, que eram ao todo 35 no HEVC, passaram para 67 no VVC. Além disso, o VVC permite realizar a predição intra sobre blocos não quadrados graças a nova estrutura de particionamento. Também foram adotadas novas estratégias na predição, como *Multiple Reference Line*, *Intra Sub-Partitions* e *Matrix-based Intra Prediction* (CHEN; YE;

KIM, 2019). Estas novas ferramentas contribuem para que se tenha uma eficiência de codificação maior em relação ao HEVC. Entretanto, um número maior de modos intra disponíveis aumenta a complexidade do codificador, pois ainda que estes modos eventualmente possuam pouca complexidade para serem aplicados, cada um destes precisa ser avaliado pelo processo de decisão de modo, o qual além de aplicar o modo realiza outras etapas de codificação.

O processo de decisão de modo é responsável por escolher qual o melhor particionamento e o melhor modo de predição para um determinado bloco do vídeo com base em um custo. Este custo é obtido através do método *Rate-Distortion Optimization* (RDO) (SULLIVAN; WIEGAND, 1998), um processo complexo que precisa realizar todas as etapas de codificação para obter o custo de cada modo de predição. Como seria inviável aplicar o RDO para cada uma das combinações possíveis entre particionamentos e modos disponíveis, apenas um subconjunto de modos é selecionado para ser testado pelo RDO. Este subconjunto é obtido a partir de um método chamado *Rough Mode Decision* (RMD), o qual realiza uma estimativa de custo menos complexa que o RDO. De acordo com estudos apresentados por (TISSIER et al., 2019) (SALDANHA et al., 2020a) (SIQUEIRA; CORREA; GRELLERT, 2020), existe um grande aumento de complexidade do VVC em relação ao HEVC. Sendo assim, como o VVC se trata de um padrão de codificação de vídeo estado da arte existem muitas possibilidades de redução de complexidade a serem exploradas no processo do modo de decisão.

A área de Aprendizado de Máquina supervisionado por sua vez tem estado em grande destaque nos últimos anos devido a sua alta aplicabilidade a diversas áreas. Com a área de codificação de vídeo, não seria diferente. Em Zhang; Kwong; Wang (2020), por exemplo, são mencionados diversos trabalhos que utilizam técnicas de aprendizado de máquina supervisionado para reduzir a complexidade do processo do modo de decisão do HEVC. Também já existem alguns trabalhos que fazem o mesmo para o VVC (AMESTOY et al., 2019) (FU et al., 2019) (YANG et al., 2019) (ZHANG et al., 2020). Entretanto, os trabalhos feitos com o HEVC não servem para o VVC já que o número de particionamentos e modos intra aumentou. Além disso, ainda que já existam trabalhos que apliquem aprendizado de máquina supervisionado para reduzir a complexidade do processo de modo de decisão do VVC, nenhum dos trabalhos encontrados realizou este processo especificamente para reduzir a quantidade de modos de predição intra que são avaliados. Por fim, devido a alta complexidade gerada pelo aumento de particionamentos e de modos intra no VVC, existe a necessidade de se desenvolver modos de decisão rápidos para a predição intra do mesmo.

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver uma solução para a redução de complexidade computacional do processo de decisão de modo da predição intra-quadro do padrão de codificação de vídeo VVC utilizando aprendizado de máquina supervisionado. A solução proposta utiliza árvores de decisão para reduzir a quantidade de modos que é avaliada pelo processo RDO do codificador VVC.

1.1.1 Objetivos Específicos

- Realizar uma análise da probabilidade de ocorrência de cada tipo de modo de predição intra no VVC para cada tamanho de bloco;
- Gerar *datasets* necessários para treinar, validar e testar cada modelo de aprendizado de máquina supervisionado;
- Obter os modelos de aprendizado de máquina supervisionado baseados em árvores de decisão necessários para a solução proposta.

1.2 Organização do Trabalho

Este trabalho está organizado da seguinte forma. No Capítulo 2, são apresentados alguns conceitos essenciais sobre codificação de vídeo, relacionando-os quando possível com o padrão de codificação *Versatile Video Coding* (VVC), o qual será utilizado no desenvolvimento deste trabalho. No Capítulo 3, são apresentados alguns conceitos de Aprendizado de Máquina supervisionado e sobre o problema de Classificação, o qual pertence a esta área. No Capítulo 4, são apresentados os trabalhos relacionados encontrados ao realizar uma busca sobre redução de complexidade na predição intra dos codificadores VVC e *High Efficiency Coding* (HEVC). No Capítulo 5, uma análise sobre a probabilidade de cada modo de predição intra ser escolhido como o melhor é apresentada para os diversos tamanhos de bloco existentes no VVC. No Capítulo 6, apresenta-se como o problema de redução de complexidade na predição intra do VVC foi modelado de acordo com as estatísticas obtidas na análise do capítulo anterior e através de modelos de aprendizado de máquina supervisionado. No Capítulo 7 são apresentados os resultados obtidos para o modo de decisão rápido proposto com o objetivo de reduzir complexidade na predição intra do VVC. Por fim, no Capítulo 8 são apresentadas as conclusões que podem ser feitas sobre o trabalho desenvolvido.

2 VERSATILE VIDEO CODING

O Versatile Video Coding (VVC) (CHEN; YE; KIM, 2019) é um padrão de codificação de vídeo que foi desenvolvido pelo grupo *Joint Video Experts Team* (JVET) e foi lançado no meio de 2020. O motivo do desenvolvimento do VVC deu-se ao fato do *High Efficiency Video Coding* (HEVC), padrão de codificação de vídeo estado da arte anterior ao VVC, não estar preparado para as demandas de novas tecnologias emergentes, tais como vídeos imersivos ou 360º, bem como às estimativas de que as resoluções dos vídeos e tráfego de dados na internet relacionados a vídeo continuem crescendo. Sendo assim, o principal objetivo do VVC é aumentar as taxas de compressão para uma mesma qualidade visual quando comparado ao HEVC. Entretanto, assim como o HEVC, o VVC é um codificador residual híbrido, o qual é apresentado na Figura 1.

Na Figura 1, cada quadro do vídeo a ser codificado é inicialmente dividido em blocos de tamanho igual, os quais são chamados no VVC de *Coding Tree Units* (CTU). Após, cada CTU é enviada para o processo de codificação, onde ainda podem ser particionadas em *Coding Units* (CU) a partir da estrutura de particionamento presente no VVC. Cada CU gerada será processada pelos módulos de predição intra-quadro e inter-quadro, os quais são responsáveis por predizer os *pixels* do bloco original ao explorar as redundâncias espaciais e temporais presentes em vídeos. A redundância espacial diz respeito a similaridade entre *pixels* vizinhos dentro de uma imagem, enquanto a redundância temporal diz respeito a similaridade entre quadros vizinhos. Como existe mais de um tipo de predição possível, o modo de decisão é responsável por escolher entre a predição inter-quadro e intra-quadro e, mais do que isso, entre as muitas possibilidades que existem dentro de cada uma destas predições. Esta escolha se dá com base em um custo taxa-distorção (taxa de bits e qualidade) obtido para cada tipo de predição. Após esta escolha, a diferença entre o bloco predito e o bloco original é gerada, a qual é chamada de resíduo. Este resíduo é então enviado para o módulo de transformada.

Os módulos de transformada e quantização são combinados com o objetivo de descartar dados do vídeo que não são perceptíveis ao sistema visual humano. Para

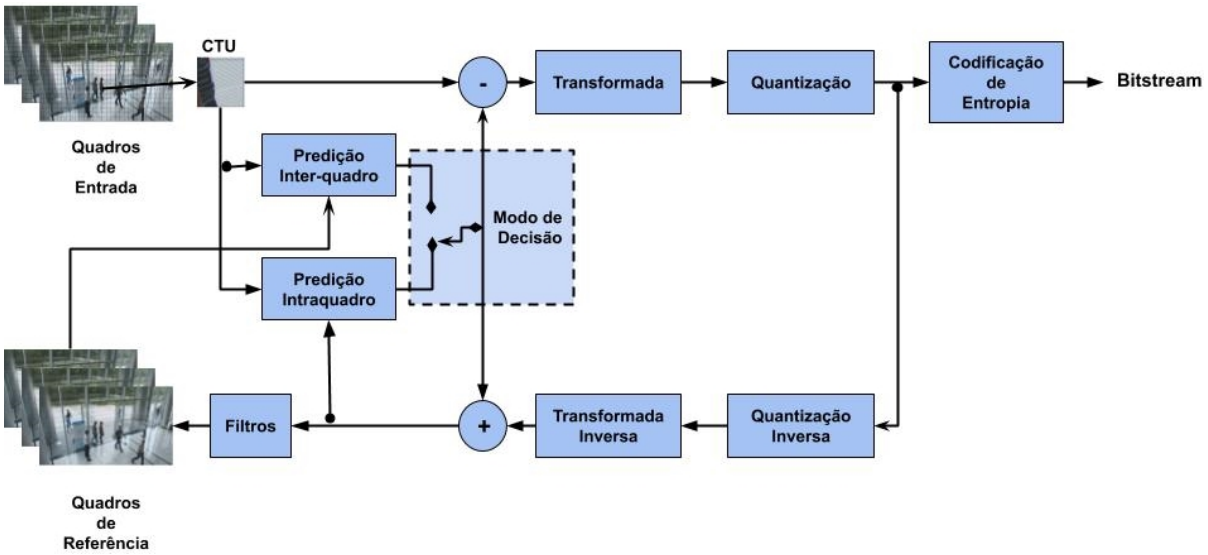


Figura 1 – Ilustração de um Codificador Residual Híbrido

tanto, primeiro a imagem é passada para o domínio das frequências pelo módulo de transformada. No domínio das frequências, as informações que não costumam ser perceptíveis ao sistema visual humano costumam ficar nos coeficientes de alta frequência. Sendo assim, o módulo de quantização corta parte das altas frequências ao realizar uma divisão inteira de todos os coeficientes por um mesmo valor, onde este valor é definido de acordo com o *Quantization Parameter* (QP). Como após esta divisão alguns dos coeficientes podem resultar em zero, o processo de quantização insere perdas.

A saída da quantização é então enviada para a codificação de entropia, módulo que realiza o processo de compressão, ou seja, todos os passos realizados antes da codificação de entropia não realizam um processo de compressão e sim uma preparação do bloco que será enviado a codificação de entropia. Ainda, como os quadros e as amostras de referência no codificador e no decodificador devem ser os mesmos e o módulo de quantização insere perdas, dentro do codificador existem os módulos de quantização inversa e transformada inversa, os quais são responsáveis por reconstruir os blocos e os quadros de acordo com as perdas inseridas. Por fim, cada bloco reconstruído ainda passa por um módulo de filtros, o qual é responsável por suavizar os efeitos de blocagem que são perceptíveis na imagem após realizar a predição.

No VVC, as principais diferenças em relação ao HEVC ocorrem na estrutura de particionamento e nos módulos de predição, transformada e filtros. A Tabela 1 faz um comparativo entre as ferramentas presentes no HEVC e no VVC na estrutura de particionamento, na predição intra-quadro, nas transformadas e nos filtros. De acordo com a tabela, é possível verificar que o VVC estende o número de transformadas que podem ser aplicadas a um bloco. Como existem mais transformadas, um processo chamado *Multiple Transform Selection* (MTS) é executado dentro de cada um dos

Tabela 1 – Ferramentas presentes no HEVC e no VVC

	HEVC	**VVC**
Estrutura de Particionamento	QT	MTT
Predição Intra-Quadro	33 modos de predição intra angulares e 2 modos de predição intra não angulares	65 modos de predição intra angulares e 2 modos de predição intra não angulares, ISP, MRL e MIP
Transformadas	DCT-II	DCT-II, DCT-VIII e DST-VII
Filtros	SAO, DF	SAO, DF e ALF

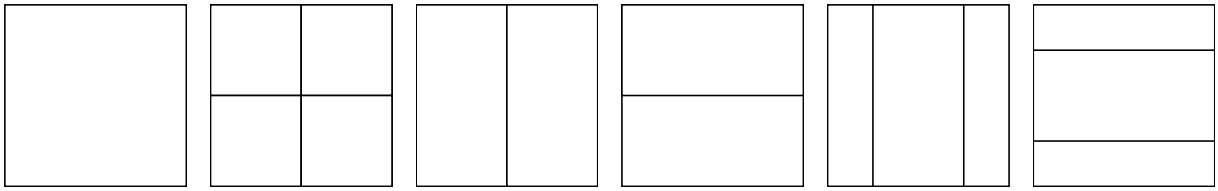
tipos de predição, o qual aplica cada uma das transformadas sobre os resíduos e seleciona ao final a transformada que gerou o melhor custo. Ainda, o VVC adota uma nova estrutura de particionamento chamada de *Multi-type Tree* (MTT) e aumenta os modos de predição intra disponíveis na predição intra-quadro.

Como o foco deste trabalho se dá nos novos modos de predição intra e no modo de decisão do VVC, as seções a seguir tem o objetivo de apresentar conceitos gerais e informações específicas do VVC sombre ambos os assuntos. Na Seção 2.1 são apresentadas informações sobre a nova estrutura de particionamento presente no VVC, já que esta está presente na predição intra-quadro do VVC, na Seção 2.2 são apresentadas informações sobre a predição intra-quadro do VVC, a qual possui novos modos de predição intra em relação ao HEVC, na Seção 2.3 são apresentadas informações sobre o processo do modo de decisão da predição intra-quadro do VVC e na Seção 2.4 são apresentados alguns trabalhos que realizaram uma comparação entre a complexidade do HEVC e do VVC.

2.1 Estrutura de Particionamento

Como visto anteriormente, em um codificador híbrido residual baseado em blocos, inicialmente cada quadro do vídeo é dividido em blocos de tamanho igual para serem processados pelo codificador, os quais são chamados de CTU no VVC. O tamanho inicial dos blocos é definido de acordo com a configuração de cada codificador, sendo que no VVC o tamanho máximo inicial para os blocos é de 128x128, o que aumenta o tamanho máximo inicial do HEVC, que era de apenas 64x64. Posteriormente, cada bloco inicial pode ainda ser particionado através da estrutura de particionamentos presente em cada codificador.

O VVC adota a estrutura de particionamentos chamada MTT, a qual estende a estrutura de particionamentos do HEVC, que possuia apenas o particionamento qua-



(a) Bloco (b) QT (c) BTV (d) BTH (e) TTV (f) TTH

Figura 2 – Particionamentos existentes no VVC

ternário (QT - *Quad-Tree*), para conter particionamentos binários (BT - *Binary Tree*) e ternários (TT), sendo que ambos podem ser no sentido horizontal (BTH e TTH) ou vertical (BTV e TTV). Estes particionamentos são apresentados na Figura 2, onde é possível visualizar que o particionamento QT divide o bloco em quatro partições de tamanhos iguais, o particionamento BT divide o bloco em duas partições na horizontal ou na vertical, ambas possuindo tamanhos iguais, e o particionamento TT divide o bloco em três partições no sentido horizontal ou vertical, onde a partição central possui 50% do tamanho do bloco e as duas partições a esquerda e a direita ou acima e abaixo possuem 25% do tamanho do bloco. Por conta desta nova estrutura de particionamentos, é possível realizar o processo de codificação sobre blocos não quadrados, o que não acontecia no HEVC.

Na Figura 3 é apresentada a ilustração de uma CTU onde os diversos tipos de particionamento disponíveis no VVC foram aplicados. As linhas em negrito representam um particionamento QT, as linhas tracejadas representam um particionamento BTH ou BTV e as linhas tracejadas com pontilhados representam um particionamento TTH ou TTV. Segundo a documentação do VVC (CHEN; YE; KIM, 2019), quando um particionamento do tipo BT ou TT é realizado, não é mais possível realizar particionamentos do tipo QT. Além disso, existem restrições nesta estrutura de particionamentos para que não seja possível chegar no mesmo bloco por mais de um caminho.

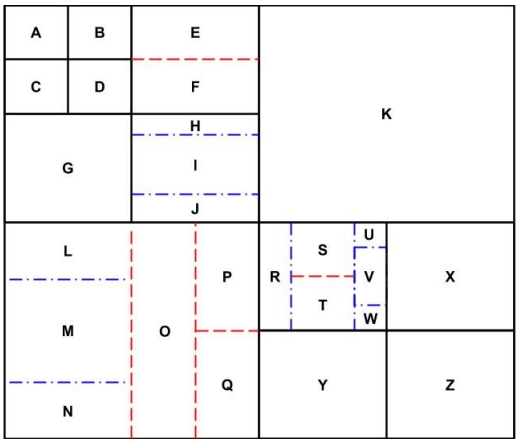


Figura 3 – Exemplo de uma CTU particionada no VVC

2.2 Predição Intra-Quadro

A predição intra-quadro é responsável por reduzir a redundância espacial presente em vídeos. Esta redundância espacial ocorre devido a semelhança entre os *pixels* vizinhos em uma imagem. Sabendo que existe esta semelhança, o que a predição intra-quadro faz é buscar predizer os *pixels* de um bloco com base nos *pixels* acima e a esquerda do bloco, os quais são chamados de *pixels* de referência. Os possíveis tamanhos de bloco (largura x altura) de luminância na predição intra-quadro do VVC são ao total 17, sendo estes 64x64, 32x32, 32x16, 32x8, 32x4, 16x32, 16x16, 16x8, 16x4, 8x32, 8x16, 8x8, 8x4, 4x32, 4x16, 4x8 e 4x4.

Existem diferentes modos de predição intra convencionais que podem ser aplicados a um bloco, onde estes modos podem ser não angulares ou angulares. Os modos não angulares buscam realizar a predição do bloco através da média (DC) ou da interpolação das amostras de referência (Planar), enquanto os modos angulares buscam realizar a predição do bloco através da extrapolação das amostras da referência em diferentes direções (ângulos) e dependendo da direção com o auxílio de filtros. Na Figura 4 é possível visualizar os modos de predição intra presentes no VVC. Existem 67 modos de predição intra no VVC, dos quais dois são não angulares (DC e Planar) e 65 são angulares (2-66). Além disso, os modos destacados em vermelho são novos em relação ao HEVC, ou seja, o VVC estende de 33 para 65 os modos de predição intra angulares.

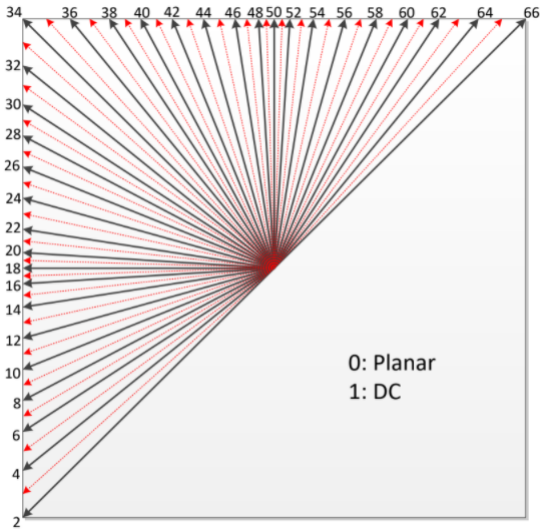


Figura 4 – Modos de Predição Intra-Quadro no VVC

Além dos novos modos de predição intra angulares introduzidos no VVC, também foram introduzidas três principais novas ferramentas no VVC que podem ser chamadas de modos de predição intra não convencionais, sendo estas *Intra Sub-Partitions Prediction* (ISP), *Multiple Reference Line* (MRL) e *Matrix-based Intra Prediction* (MIP).

2.2.1 Intra Sub-Partitions Prediction (ISP)

No ISP, um bloco é dividido horizontalmente ou verticalmente em duas ou quatro subpartições de tamanho igual. Exemplos destas subpartições podem ser vistos na Figura 5 e na Figura 6. Então, para cada subpartição, a predição intra-quadro é realizada, sendo que ao fim da predição de uma subpartição as amostras de referência geradas são utilizadas pela próxima subpartição a ser predita. As restrições do ISP se aplicam ao valor mínimo de *pixels* que uma subpartição deve possuir, sendo este 16, e ao modo de predição intra, o qual deve ser o mesmo para todas as subpartições. Dado o número mínimo de *pixels* que uma subpartição deve possuir, o menor tamanho de bloco aplicável ao ISP é 4x8 ou 8x4, sendo que neste caso somente duas subpartições são geradas. Nos outros casos, quatro subpartições são geradas. Além disso, o ISP não pode ser combinado com MRL ou MIP. Esta ferramenta foi proposta em (DE-LUXÁN-HERNÁNDEZ et al., 2019), onde apresentou-se que ao ser integrada ao VTM 3.0 em média foi obtido um ganho em eficiência de codificação de 0,57% e 0,27% para as configurações AI (*All Intra*) e RA (*Random Access*).

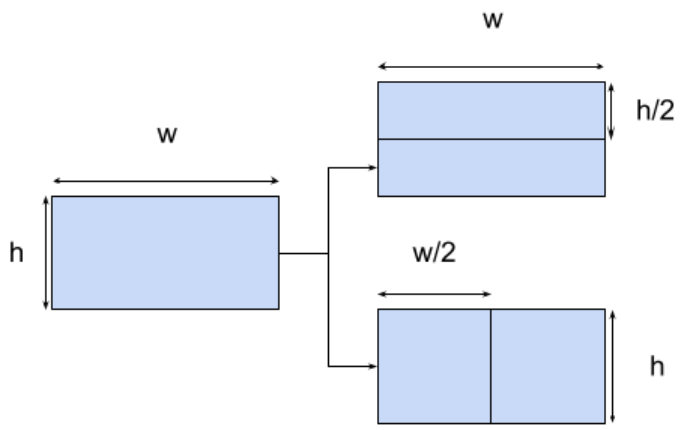


Figura 5 – Subpartições do ISP para blocos de tamanho 4x8 ou 8x4

2.2.2 Multiple Reference Line (MRL)

A ferramenta MRL estende o número de linhas e colunas adjacentes a um bloco que podem ser consideradas como amostras de referência, como pode ser visto na Figura 7. Enquanto no HEVC somente a Referência 0 podia ser utilizada, no VVC também podem ser utilizadas as Referências 1 e 3, ou seja, o espaço de amostras de referência é ampliado para a geração das amostras preditas. Esta ferramenta foi proposta em (CHANG et al., 2019) e de acordo com testes realizados ao integrar esta ferramenta no VTM 2.0.1, em média a eficiência de codificação aumentou em 0,46% e 0,2% para as configurações AI e RA, respectivamente. Além disto, observou-se um ganho maior para vídeos que contém gravações de tela, chegando a até 1,45%. Cabe

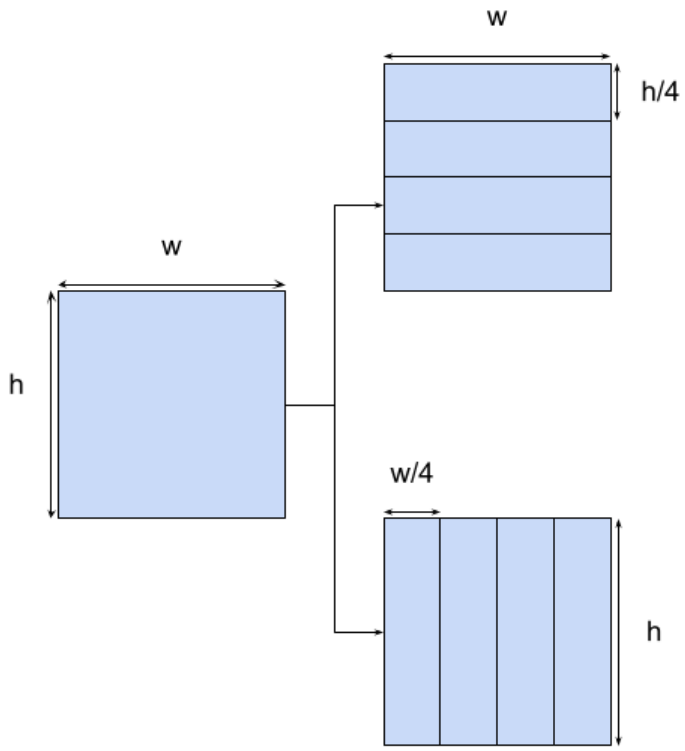


Figura 6 – Subpartições do ISP para blocos maiores que 4x8 e 8x4

destacar que a ferramenta MRL pode ser utilizada somente com os modos de predição intra DC e angulares.

2.2.3 Matrix-based Intra Prediction (MIP)

A ferramenta MIP foi desenvolvida com o objetivo de distinguir-se dos modos de predição intra convencionais. Para tanto, três conjuntos de matrizes foram obtidos após o treinamento de redes neurais com um grande conjunto de dados, sendo o objetivo deste treinamento realizar uma predição intra-quadro não linear. Na Figura 8 é apresentado o processo de predição realizado pelo MIP, onde é possível visualizar que existem três etapas, sendo a primeira a etapa de Média, a segunda a etapa de Multiplicação Vetor-Matriz e a terceira a etapa de Interpolação. Na primeira etapa, as amostras de referência do bloco a ser predito são reduzidas aos vetores de amostras bdry_{top} e $\text{bdry}_{\text{left}}$, os quais são obtidos após realizar a média entre amostras adjacentes. Após, estes dois vetores de amostras são concatenados para formar o vetor bdry_{red}, o qual junto ao *mode* K (modo MIP) formam a entrada da próxima etapa. Na segunda etapa, uma matriz e um vetor de *offset* são selecionados de acordo com o tamanho do bloco e o modo MIP. É realizada então a multiplicação entre o vetor bdry_{red} e a matriz selecionada, e ao resultado desta multiplicação é somado o vetor de *offset* selecionado. O resultado desta etapa consiste em um bloco predito sub-amostrado, o qual pode ser visto na Figura 8 na terceira etapa. Finalmente, na terceira etapa os *pixels* que faltam são preditos através da interpolação bilinear das amostras de refe-

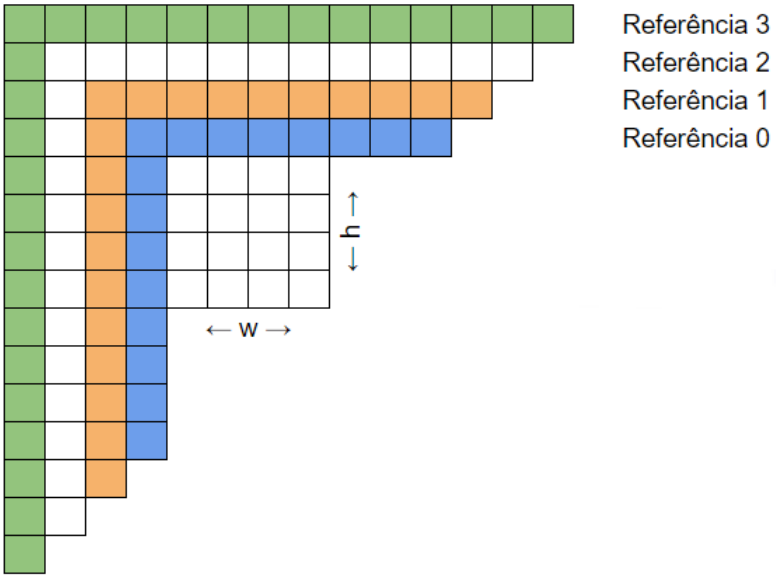


Figura 7 – Possíveis amostras de referência utilizadas pelo VVC com a ferramenta MRL

rência e dos *pixels* obtidos após a segunda etapa. De acordo com informações da documentação do VVC (CHEN; YE; KIM, 2019), existem três conjuntos de matrizes e vetores de *offset*, onde cada matriz é associada a exatamente um vetor de *offset* e ambos compõem um modo MIP. O primeiro conjunto destina-se aos blocos de tamanho 4x4, o qual contém 18 matrizes e vetores de *offset*; o segundo conjunto destina-se aos blocos de tamanho 4x8, 8x4 e 8x8, o qual contém dez matrizes e vetores de *offset*; e o terceiro conjunto destina-se aos tamanhos de bloco remanescentes, o qual contém seis matrizes e vetores de *offset*. De acordo com (SALDANHA et al., 2020a), o MIP é adequado para blocos que possuem mais de uma direção, já que o mesmo realiza uma predição intra-quadro não linear.

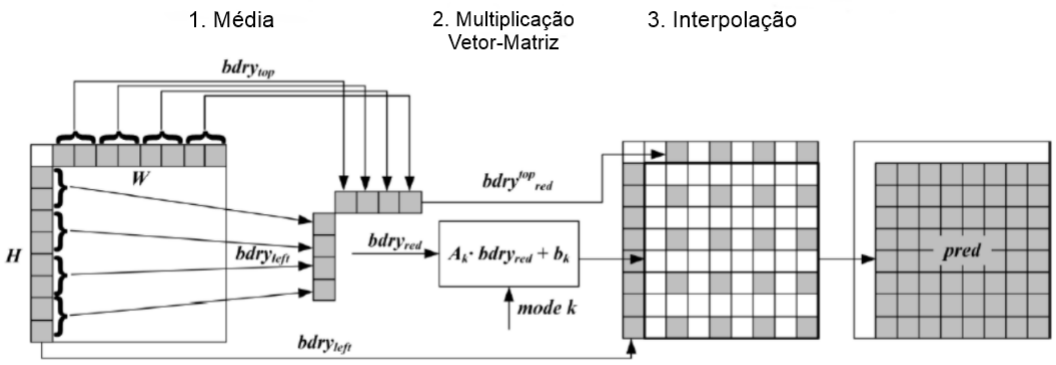


Figura 8 – Processo de predição da ferramenta MIP, adaptado de (CHEN; YE; KIM, 2019)

2.3 Modo de Decisão Intra

Idealmente, um codificador deveria avaliar para cada bloco de entrada a melhor sequência de particionamentos a ser aplicada e o melhor modo de predição intra a ser aplicado para cada uma das partições geradas. Entretanto, devido ao alto número de combinações existente, realizar esta busca exaustiva é inviável do ponto de vista de custo computacional. Sendo assim, o VVC adota um processo de modo de decisão rápido que avalia apenas um subconjunto destas combinações. Este processo é apresentado na Figura 9, onde é possível verificar que o mesmo divide-se em dois principais subprocessos, sendo o primeiro chamado de RMD e o segundo chamado de RDO. Ambos procuram obter os custos dos modos de predição intra quando aplicados a um bloco através da equação em 1.

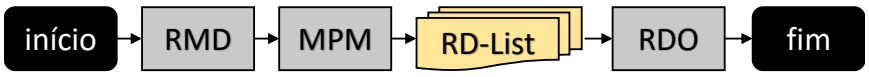


Figura 9 – Processo do Modo de Decisão na predição intra do VVC

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

Nesta equação, *mode* é o modo de predição intra que está sendo aplicado, D é a medida de distorção, R é a taxa de bits utilizada, λ é o multiplicador de Lagrange e QP é o parâmetro de quantização. A distorção é obtida através da SAD (*Sum of Absolute Differences*) conforme equação (2) e da SATD (*Sum of Absolute Transformed Differences*) conforme equação (3) entre o bloco predito e o bloco original, onde a medida que resulta no menor valor é selecionada. Já a taxa de bits é obtida através da quantidade de informações que devem ser enviadas ao decodificador, dentre estas o resíduo e informações laterais, onde informações laterais dizem respeito as sinalizações que devem ser enviadas ao decodificador, como o modo de predição intra, particionamento utilizado, dentre outros. O λ presente na equação define o *tradeoff* entre compressão e qualidade, onde valores acima de 1 irão priorizar a compressão e valores abaixo de 1 irão priorizar a qualidade. Já o QP define a quantidade de informações que serão perdidas na etapa de quantização. Valores menores de QP irão gerar pouca perda de informação e, portanto, uma maior qualidade visual com uma taxa de compressão menor, enquanto valores maiores de QP irão gerar mais perda de informação, o que acarreta em uma menor qualidade visual para uma taxa de compressão maior.

$$SAD = \sum_{i=1}^{height} \sum_{j=1}^{width} |O_(i,j) - P_(i,j)| \tag{2}$$

$$SADT = \sum_{i=1}^{height} \sum_{j=1}^{width} Hadamard(O_(i,j) - P_(i,j)) \quad (3)$$

Como tanto o processo do RMD quanto o processo do RDO baseiam-se na mesma equação para obter os custos dos modos de predição intra, a diferença entre ambos se dá no ponto do processo de codificação em que a distorção e a taxa de bits são obtidas. Para medir a distorção, o RMD realiza a SAD e a SATD entre o bloco predito e o bloco original imediatamente após aplicar o modo de predição intra ao bloco, enquanto o RDO obtém a distorção somente após todo o processo de codificação, o qual inclui as etapas de transformada e quantização. O mesmo ocorre para a taxa de bits, já que o RMD não leva em consideração as informações obtidas após a codificação de entropia, enquanto o RDO leva em consideração. Sendo assim, o RDO é um processo mais complexo que o RMD por obter o custo real de um modo de predição intra, enquanto o RMD é apenas uma estimativa deste custo. Portanto, o objetivo do RMD é gerar um subconjunto de modos de predição intra a serem avaliados pelo RDO, chamado de RD-List. A implementação do processo RMD no VVC possui cinco etapas internas, as quais podem ser chamadas de RMD-1, RMD-2, MRL, MIP e ISP. Todas estas etapas adicionam ou substituem modos de predição intra na RD-List.

As etapas RMD-1 e RMD-2 consistem em adicionar modos de predição intra convencionais a RD-List. No RMD-1, apenas os modos de predição intra não angulares e os angulares de número par passam por uma estimativa de custo. A partir destas avaliações, uma lista ordenada crescentemente pelo custo é gerada, e os N primeiros modos desta lista são enviados para a etapa RMD-2. O valor de N varia de acordo com o tamanho do bloco, onde este valor é inversamente proporcional ao tamanho do bloco. Após, para cada modo de predição intra angular presente na lista retornada pelo RMD-1, o RMD-2 avalia os modos adjacentes a esquerda e a direita, caso estes existam. Se algum destes modos de predição intra angulares apresentar um menor custo do que algum dos modos presentes na lista retornada pelo RMD-1, este modo com maior custo é substituído.

Em seguida, a etapa MRL consiste em avaliar alguns modos de predição intra angulares ao variar as linhas de referência para a Referência 1 e para a Referência 3, como apresentado na Figura 7. Nesta etapa, somente os *Most Probable Modes* (MPM) são avaliados, sendo que estes consistem nos melhores modos de predição intra obtidos para CUs vizinhas.

A etapa do MIP adiciona modos de predição intra MIP a RD-List. Para tanto, é feita uma avaliação de alguns dos modos MIP disponíveis para o bloco. A seleção de qual modo MIP avaliar se dá através de um mapeamento direto que existe entre modos de predição intra angulares e modos de predição intra MIP. Desta forma, a partir dos modos de predição intra angulares com menor custo RMD até o momento,

são selecionados os modos de predição intra MIP a serem avaliados.

Na sequência, a etapa ISP reserva posições ao final da RD-List onde modos de predição intra ISP serão avaliados no processo do RDO para tamanhos de bloco maiores que 4x4. Ao fim do processo RMD, a etapa MPM adiciona, caso ainda não estejam na RD-List, modos de predição intra a partir dos melhores modos de predição intra obtidos para os blocos vizinhos.

A partir da RD-List gerada pelo RMD, o RDO obtém os custos para cada modo presente na lista e ao fim seleciona o modo de predição intra que obteve o menor custo. Como no final da RD-List existem posições reservadas ao ISP, quando o processo RDO chega na primeira posição reservada a um modo de predição intra ISP, esta posição e as demais reservadas são associadas a modos de predição intra convencionais que obtiveram menor RD-Cost até o momento. Dentro do processo do RDO existe uma decisão antecipada para a avaliação de um modo de predição intra ISP, sendo que esta ocorre sempre que o RD-Cost de um modo ISP ultrapassa o maior custo de um dos modos de predição intra convencionais já avaliados.

Na Tabela 2, são apresentadas as quantidades máximas de modos na RD-List após a etapa do RMD para cada tamanho de bloco. Como o VVC possui diferentes combinações de transformadas que podem ser aplicadas em um bloco, as quantidades apresentadas na Tabela 2 são um somatório das quantidades de modos que são avaliados em cada uma das transformadas. As colunas Planar, DC, Angular, ISP DC/Planar, ISP Angular e MIP apresentam as quantidades máximas para cada um destes tipos de modos de predição intra-quadro e a coluna Total Máximo apresenta a quantidade máxima de modos que são avaliados. O que pode ser visto através destes números é que a quantidade máxima de modos avaliados no processo de decisão de modo da predição intra-quadro no VVC pode chegar a números muito altos, possuindo um mínimo de 34 modos até um máximo de 81 modos sendo avaliados para apenas um bloco. Ou seja, ainda que o processo RMD busque gerar um subconjunto de modos de predição intra-quadro para serem avaliados pelo processo de decisão de modo, este subconjunto ainda pode ser grande. Logo, soluções que busquem reduzir a quantidade de modos avaliados no processo do RDO são importantes.

2.4 Complexidade do Versatile Video Coding em relação ao High Efficiency Coding

Como o VVC traz diversas novas ferramentas de predição em relação ao HEVC com o objetivo de aumentar a eficiência de codificação para uma mesma qualidade visual, existe também um aumento de complexidade em relação ao HEVC. Diversos autores realizaram análises de complexidade do VVC em relação ao HEVC e algumas destas análises serão apresentadas a seguir.

Tabela 2 – Quantidade Máxima de Modos na RD-List por Tamanho de Bloco para cada Tipo de Modo e para todos Tipos de Modos

Tamanho de Bloco	Planar	DC	Angular	ISP DC/Planar	ISP Angular	MIP	Total Máximo
4x4	5	15	20	0	0	18	34
4x8	5	15	20	8	25	15	52
4x16	5	15	20	8	22	15	52
4x32	5	15	21	8	22	15	50
8x4	5	15	20	8	25	15	52
8x8	6	15	20	4	11	18	44
8x16	6	15	20	8	22	20	50
8x32	5	15	21	8	22	18	54
16x4	5	15	20	8	25	15	52
16x8	6	18	20	8	22	18	54
16x16	6	15	24	12	30	35	70
16x32	6	15	24	12	30	35	75
32x4	5	15	20	8	22	18	52
32x8	6	15	20	8	22	18	50
32x16	6	15	21	12	30	36	69
32x32	6	18	24	12	30	42	81
64x64	3	9	12	12	30	27	66
Mínimo	3	9	12	0	0	15	34
Máximo	6	18	24	12	30	42	81

Tissier et al. (2019) fez um estudo com o objetivo de identificar onde estariam as principais oportunidades de redução de complexidade no VVC. Para tanto, testes foram realizados no VTM 3.0 com a configuração AI, onde 22 sequências de vídeo da CTC do VVC foram codificadas com os valores de QP 22, 27, 32 e 37. De acordo com as análises de complexidade apresentadas no trabalho, verificou-se que as principais oportunidades de redução de complexidade encontram-se na estrutura de particionamentos, no módulo de predição intra-quadro e no módulo das transformadas, onde esta redução de complexidade pode se dar ao eliminar as avaliações de particionamentos, modos de predição intra ou transformadas que não resultam no melhor custo.

No trabalho de Saldanha et al. (2020a), o objetivo foi realizar uma análise de complexidade da predição intra-quadro no VVC em relação ao HEVC. Nesta análise, o VTM 7.0 e o HM 16.20 foram levados em consideração. De acordo com os resultados obtidos, no melhor cenário o VVC é 21 vezes mais lento que o HEVC, enquanto que no pior cenário este valor chega a até 56 vezes. Além disso, mostrou-se que a complexidade da predição intra-quadro no VVC se dá principalmente na etapa do RDO do processo do modo de decisão, visto anteriormente na Figura 9.

Em Siqueira; Correa; Grellert (2020), uma análise foi feita a respeito da eficiência de codificação e da complexidade do VVC em relação ao HEVC. Neste trabalho, foram utilizados o VTM 5.0 e o HM 16.9, ambos com o arquivo de configuração RA. Nos

testes realizados, foram considerados 19 sequências de vídeo da CTC do VVC as quais foram codificadas com os valores de QP 22, 27, 32 e 37. Como resultados, observou-se que o VVC tem em média uma eficiência de codificação de 44,4% maior para uma mesma qualidade visual em relação ao HEVC. Entretanto, o processo de codificação do VVC é de 10 a 16 vezes mais complexo que o HEVC, onde a predição intra-quadro no VVC, por exemplo, é 109 vezes mais complexa que a predição intra-quadro no HEVC.

Como o VVC apresenta uma complexidade maior em relação ao HEVC e é um padrão de codificação de vídeo estado da arte, existem diversas possibilidades de redução de complexidade que podem ser exploradas, principalmente no processo de escolha do modo de decisão intra, o qual tornou-se mais custoso por conta das novas ferramentas de codificação que foram introduzidas no VVC.

3 APRENDIZADO DE MÁQUINA SUPERVISIONADO

Aprendizado de máquina é uma área da inteligência artificial que se une a outras áreas, tais como computação, probabilidade e estatística. Nesta área, a preocupação se dá em fazer com que computadores sejam capazes de aprender com base em suas experiências passadas, onde estas experiências passadas nada mais são que os dados já conhecidos pela máquina. Isto implica em menos intervenção humana em algoritmos, já que a máquina se torna capaz de aprender e adaptar-se automaticamente de acordo com os dados previamente conhecidos.

Na inteligência artificial, são considerados tanto o raciocínio dedutivo quanto o raciocínio indutivo (NORVIG; RUSSELL, 2014). No raciocínio dedutivo, parte-se de uma premissa geral ou mais ampla e chega-se a uma conclusão mais particular. Já no raciocínio indutivo, parte-se de premissas particulares e chega-se a uma conclusão geral. No aprendizado de máquina, trabalha-se com o raciocínio indutivo, isto porque a máquina realiza seu aprendizado apenas sobre um conjunto seleto de dados, não sobre o todo, e utiliza este aprendizado para inferir características e padrões sobre novos dados que são apresentados a ela. Por conta disto, é necessário ter cuidado na seleção dos dados sobre os quais a máquina irá realizar o seu aprendizado, pois se os mesmos não forem uma boa representação do todo, pode acontecer o que é conhecido na inteligência artificial como *overfitting*, ou sobreajuste, o que indica que o modelo gerado pela máquina funciona muito bem para o conjunto de dados utilizados no treinamento mas não funciona para conjuntos de dados desconhecidos pela máquina.

A área do aprendizado de máquina se divide em várias categorias distintas de acordo com o tipo de entrada, sinal ou feedback que é disponível ao aprendizado (NORVIG; RUSSELL, 2014). Dentre estas várias categorias, três se destacam: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.

No aprendizado supervisionado, a máquina recebe como entrada pares que contém tanto o valor de entrada quanto o valor de saída desejado. Assim, a máquina é capaz de aprender uma função que mapeia uma entrada para uma saída corretamente. Os principais problemas existentes na área de aprendizado supervisionado

são o de classificação e regressão, onde têm-se um conjunto de entradas com sua classificação conhecida, e deseja-se classificar um outro conjunto de entradas com sua classificação desconhecida. O que muda entre a classificação e a regressão é que na classificação a saída é um valor discreto, enquanto que na regressão a saída é um valor contínuo.

Diferente do aprendizado supervisionado, no aprendizado não supervisionado a máquina recebe o conjunto de entrada contendo apenas os valores de entrada, mas não os valores de saída desejados. A partir disto, a máquina tenta realizar um agrupamento dos dados mais parecidos, gerando clusters ou classes de dados. Um problema existente na área de aprendizado não supervisionado é o de *clustering*, onde um conjunto de dados são agrupados de acordo com sua semelhança, o que permite descobrir padrões desconhecidos até então entre os dados.

Por fim, no aprendizado por reforço, a máquina procura uma sequência de ações que maximizem sua "recompensa" no final. Esta procura pela sequência de ações é feita por tentativa e erro. Um exemplo de problema existente na área de aprendizado por reforço é onde a máquina deve jogar um jogo contra um humano, e esta máquina deve aprender qual é a sequência de ações que maximizam as chances dela de ganhar. Este aprendizado geralmente requer que um humano dê *feedback* a máquina a respeito de suas ações.

Neste estudo, o foco será em aprendizado supervisionado e no problema de classificação, e em como este problema pode ser aplicado ao modo de decisão do VVC. Na Seção 3.1 será apresentado o problema de classificação e os passos que costumam ser seguidos ao lidar com este problema e na Seção 3.2 será discutida a relação que existe entre o problema de classificação e o problema do modo de decisão do VVC.

3.1 Classificação

Como visto anteriormente, o problema da classificação faz parte da área de aprendizado supervisionado. Pode-se descrever este problema da seguinte forma: têm-se um conjunto de exemplos de treinamento $S = \{s_1, s_2, ..., s_n\}$, onde cada exemplo do conjunto de treinamento S é associado a um elemento do conjunto de classes $C = \{1, 2, ..., k\}$; através do treinamento, a máquina gera uma função que mapeia um exemplo para uma classe que pertence ao conjunto C; assim, quando um novo conjunto de teste $T = \{t_1, t_2, ..., t_n\}$ é apresentado a máquina, ela é capaz de predizer para cada elemento do conjunto T a qual classe ele pertence.

A classificação lida apenas com problemas onde o conjunto de classes são categóricos, sendo que cada categoria é representada por um número inteiro. Quando o conjunto de classes é do tipo contínuo, tem-se um problema de regressão e não de classificação. Além disso, existem variações no problema de classificação. Por

exemplo, pode-se querer que um determinado exemplo seja associado a exatamente uma classe ou a múltiplas classes; também pode-se querer que a probabilidade de um exemplo estar associado a cada uma das classes seja expressa ao invés de mostrar como saída apenas uma classe ao qual este exemplo é associado.

Quando se trata do problema de classificação, os passos para lidar com este problema geralmente são a obtenção de um *dataset*, escolha das *features* que serão utilizadas no aprendizado, escolha do modelo de classificação e, por fim, treinamento e resultados. Cada um destes passos será descrito nas subseções a seguir.

3.1.1 Obtenção de um dataset

Como o problema de classificação pertence a área de aprendizado de máquina supervisionado, torna-se necessário um *dataset* que contenha exemplos para realizar o treinamento e teste do modelo de classificação. Aqui, o *dataset* pode tanto ser gerado pela pessoa que está trabalhando com o problema quanto buscado em *websites* que disponibilizam *datasets* de forma aberta.

Ao trabalhar com codificação de vídeos, os exemplos que são utilizados para treinamento geralmente são extraídos do próprio processo de codificação. A maioria dos trabalhos desenvolvidos para a área de codificação de vídeo em algum momento extraem dados do processo de codificação, seja para realizar levantamentos estatísticos ou para gerar *datasets*. Sendo assim, a etapa de obter um *dataset* para trabalhar com problemas de classificação aplicado ao processo de codificação de vídeos não é problemática do ponto de vista de se obter uma grande quantidade de vídeos, ao passo que cada vídeo contém vários quadros que são divididos em vários blocos e é possível coletar informações do processo de codificação de cada um desses blocos para gerar um *dataset* com grande quantidade de dados. Entretanto, o problema pode dar-se no tempo levado para obter estes dados, visto que o tempo de codificação de um vídeo em codificadores atuais como o VVC costuma ser longo. Além disso, deve ser levado em consideração que os exemplos deste *dataset* devem estar balanceados e que este balanceamento deve ser realizado não só por classe que se pretende realizar a classificação, mas também é importante considerar um balanceamento considerando as diferentes configurações de codificação que podem ser utilizadas para a geração dos *datasets*, tais como, diferentes sequências de vídeo, tamanhos do bloco, parâmetros de quantização (QPs) entre outros.

3.1.2 Escolha das Features

A etapa de escolha das *features* consiste tanto em selecionar quais *features* obtidas através do processo de codificação são relevantes para o problema de classificação quanto, quando for o caso, gerar novas *features* de forma externa. Para gerar novas *features* de forma externa, geralmente trabalha-se a partir dos arquivos das

sequências de vídeo, de onde são lidos os dados dos blocos e calculadas as *features* desejadas. Por se tratar de imagens, as *features* consideradas geralmente fazem parte de técnicas de processamento digital de imagens. Além disso, *features* extraídas durante o processo de codificação também podem ser importantes para ajudar a resolver os problemas de classificação presentes em um codificador de vídeo. Quanto a seleção de quais *features* são mais relevantes, geralmente é analisado o Ganho de Informação para cada *feature* presente no *dataset*.

3.1.3 Escolha do Modelo de Classificação

Na fase de escolha do modelo de classificação já se tem o dataset pronto para ser passado ao classificador. Entretanto, a preocupação agora se dá em decidir qual algoritmo de classificação utilizar. Esta não é uma tarefa trivial.

Existem diversos classificadores disponíveis, desde os mais simples até os mais robustos. Como geralmente os trabalhos desenvolvidos na área de codificação de vídeo buscam reduzir complexidade de algoritmos de otimização da taxa-distorção, usualmente opta-se por trabalhar com classificadores mais simples que alcancem resultados satisfatórios, ou seja, redução do tempo de codificação com baixas perdas na eficiência de compressão. Dentre estes, destacam-se a Árvore de Decisão e Naïve Bayes, os quais costumam aparecer em trabalhos da literatura conforme será visto no Capítulo 4. Como o objetivo deste trabalho é redução de complexidade, decidiu-se focar em árvores de decisões simples, já que estas costumam apresentar bons resultados com inserção de pouco *overhead*. Na subseção a seguir, será descrito brevemente o modelo de classificação baseado em Árvore de Decisão.

3.1.3.1 Árvore de Decisão

Uma Árvore de Decisão é um dos modelos que mais se parecem com o raciocínio humano. Ela pode ser vista como uma sequência de testes que devem ser verificados para predizer a classe de um determinado dado de entrada.

Para entender melhor, imagine a seguinte situação. Você pede ao seu amigo uma sugestão de filme. Ele então começa a lhe fazer uma série de perguntas, como gênero que você quer assistir, tempo de duração desejado, se você prefere um filme antigo ou recém lançado e se quer algo do cinema nacional ou estrangeiro. Ao fim desta série de perguntas, com base nas suas respostas, seu amigo irá lhe recomendar um filme. Este raciocínio é o que está por trás de uma Árvore de Decisão.

Na Figura 10, têm-se um exemplo de uma possível Árvore de Decisão. Nela, têm-se os nodos de decisão, que são as perguntas que a árvore faz acerca do dado que está sendo classificado, e as folhas, que representam as classes às quais o dado pode ser associado. Um novo dado a ser classificado sempre começa no nodo raiz, e vai descendo na hierarquia de acordo com os resultados dos testes até chegar em um nó

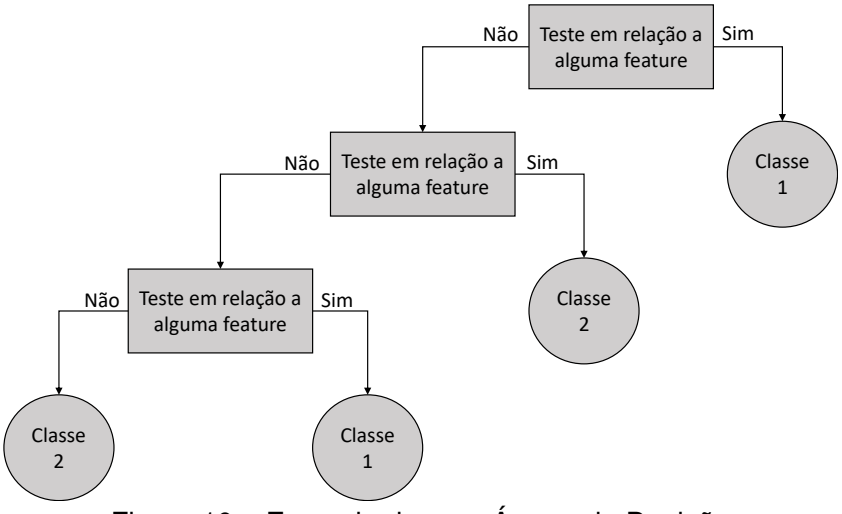


Figura 10 – Exemplo de uma Árvore de Decisão

folha, que irá conter a classe predita para este dado.

Para obter a árvore que será utilizada para classificar novos dados, o modelo recebe um conjunto de treinamento com dados já classificados. Ele começa então a separar hierarquicamente o conjunto de treinamento em vários níveis, selecionando *features* dos dados presentes em cada nodo e elaborando testes que devem ser feitos em relação a estas *features* para ir a um próximo nível. Geralmente, a seleção das *features* é feita a partir de medidas estatísticas, tais como *Information Gain* e *Gini Index*.

Apesar de ser um modelo de fácil entendimento, possui alguns problemas, principalmente em relação a *overfitting*. Por conta deste problema, geralmente quando este modelo é utilizado busca-se limitar a profundidade máxima da árvore para que o modelo não fique superespecializado no conjunto de treinamento.

3.1.4 Treinamento e Resultados

Após escolher o algoritmo de classificação que será utilizado no modelo, surge a preocupação em como dividir o conjunto de documentos entre treinamento e teste. A resposta mais simples para esta pergunta seria dividir o conjunto de documentos entre treinamento e teste pegando aproximadamente 80% para treinamento e 20% para teste. Geralmente, estes subconjuntos são divididos de forma aleatória, para gerar amostragens melhores. Além disso, são mutuamente exclusivos, ou seja, a intersecção entre os dois conjuntos é vazia. Entretanto, mesmo fazendo uma seleção aleatória, corre-se o risco do conjunto de treinamento conter exemplos que não são representativos o suficiente. Isso desencadeia em o classificador não conseguir generalizar, tendo um desempenho ruim na hora de classificar novos dados e funcionando bem apenas para dados do conjunto de treinamento.

Por conta deste problema, uma metodologia muito empregada para avaliar a capacidade de generalização do modelo para um conjunto de dados é a de *cross-validation*

(CV). A técnica de CV consiste em particionar o conjunto de dados disponível em alguns subconjuntos mutualmente exclusivos. Depois, alguns destes subconjuntos são utilizados para treinar o modelo ou estimar hiperparâmetros e o restante dos subconjuntos são utilizados para testar ou validar o modelo. Existem duas principais abordagens sobre como particionar o conjunto de dados: *leave-p-out* e *k-fold*.

Na abordagem *leave-p-out*, p exemplos são usados para teste e o restante é usado para treinamento. Isto é repetido até cobrir todas as possíveis combinações de p exemplos para teste em um conjunto de dados de tamanho n. Desta forma, tem-se:

$$\binom{n}{p} = \frac{n!}{p!(n-p)!} \tag{4}$$

O problema desta abordagem é que mesmo para n razoavelmente pequenos, o número de combinações se torna grande o suficiente para tornar o custo computacional inviável. Desta forma, a abordagem *k-fold* é uma aproximação da *leave-p-out*, pois ao invés de testar exaustivamente todas as combinações possíveis, ela seleciona aleatoriamente uma das possíveis combinações. Por conta disso, a *leave-p-out* é vista como uma abordagem exaustiva e a *k-fold* é vista como uma abordagem não exaustiva.

Na abordagem *k-fold* o conjunto de dados é dividido aleatoriamente em k partições de tamanho igual. Destas k partições, uma é utilizada como teste e as k-1 partições restantes são utilizadas como treinamento. Este processo é repetido k vezes até que todas as partições tenham sido utilizadas tanto como teste quanto como treinamento. Ao final, para medir a capacidade de generalização do modelo, a média de cada um dos resultados das k iterações é calculada.

Após realizar o treinamento, a última etapa consiste em avaliar o desempenho do modelo de classificação. A forma mais comum e intuitiva de visualizar os resultados é através da matriz de confusão. Supondo que o objetivo do classificador seja classificar exemplos de vídeos em duas categorias distintas, sendo estas: "é uma gravação de tela", "não é uma gravação de tela". A Figura 3 representa uma possível matriz de confusão deste cenário.

Tabela 3 – Exemplo de uma possível matriz de confusão

		Valor Predito	
		É uma gravação de tela	**Não é uma gravação de tela**
Valor Real	**É uma gravação de tela**	89	11
	Não é uma gravação de tela	2	98

Observando a Tabela 3, conclui-se que existem 100 exemplos de cada uma das classes. Além disso, é possível visualizar que dos 100 exemplos que eram uma gra-

vação de tela, 89 foram classificados corretamente e 11 foram classificados incorretamente. Dos 100 exemplos que não eram uma gravação de tela, 98 foram classificados corretamente e 2 foram classificados incorretamente. O interessante da matriz de confusão é que ela é fácil de ser interpretada, pois todas as classificações que foram feitas corretamente estarão na diagonal principal. Além disso, com ela é possível visualizar em qual classe o modelo está tendo um pior desempenho.

Uma outra relação que pode-se obter da matriz de confusão é a relação de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos. Aqui, verdadeiros e falsos dizem respeito a predição do classificador estar correta ou não e positivos e negativos dizem respeito à real classe do exemplo. Por exemplo, na Tabela 3 os 89 exemplos que eram uma gravação de tela classificados como sendo uma gravação de tela são verdadeiros positivos; os 11 exemplos que eram uma gravação de tela e que foram classificados como não sendo uma gravação de tela são falsos positivos; os 2 exemplos que não são gravações de tela e que foram classificados como sendo gravações de tela são falsos negativos e os 98 exemplos que não são gravações de tela e que são classificados como não sendo gravações de tela são verdadeiros negativos.

Os falsos positivos e os falsos negativos estão associados aos erros do tipo I e II, respectivamente, da inferência estatística. Além disso, eles podem ser utilizados para calcular métricas a respeito da acurácia do modelo.

A sensibilidade de um modelo (5) diz respeito a capacidade que o modelo tem de classificar um dado que é positivo como um verdadeiro positivo. No exemplo, seria a capacidade de encontrar condições que levam um exemplo a ser da classe "é uma gravação de tela" em documentos que realmente tem estas condições.

$$sensibilidade = \frac{verdadeiros\_positivos}{verdadeiros\_positivos + falsos\_negativos} \tag{5}$$

A precisão de um modelo (6) pode representar a proporção de resultados que são verdadeiros positivos ou verdadeiros negativos. Ela não deve ser utilizada para tirar conclusões quando a proporção de dados entre as classes for diferente. A equação em (6) leva em consideração os verdadeiros positivos.

$$precisao = \frac{verdadeiros\_positivos}{verdadeiros\_positivos + falsos\_positivos} \tag{6}$$

Com as medidas de sensibilidade e precisão, é possível calcular uma nova medida, chamada *F1 Score* (7). Esta medida é uma representação da acurácia do modelo e seu cálculo se dá a partir da média harmônica entre a precisão e a sensibilidade.

$$f1 = 2 * \frac{precisao * sensibilidade}{precisao + sensibilidade} \tag{7}$$

3.2 O Problema de Classificação na Decisão de Modo de Predição Intra-Quadro

É possível estabelecer uma relação entre o problema de classificação da área de aprendizado de máquina supervisionado e o problema de decisão de modo da área de codificação de vídeo. Como visto anteriormente, o problema de decisão de modo na área de codificação de vídeo consiste em, para cada bloco a ser predito, escolher uma dentre as várias combinações de particionamento e tipo de predição que existem. Logo, dentro do processo do modo de decisão, diversos problemas de classificação podem ser vistos.

Considerando os modos de predição intra-quadro, por exemplo, um problema de classificação pode dar-se no processo de decidir qual é o melhor modo de predição intra-quadro a ser aplicado no bloco, onde existira um conjunto de *features* F = {f_1, f_2, ..., f_n} relacionadas ao bloco em questão e um conjunto de classes C = {Planar, DC, Angular, ISP, MIP}, onde cada classe estaria associada a um tipo de modo de predição intra-quadro que deve ser aplicado ao bloco. De acordo com a predição feita pelo classificador, alguns dos modos de predição intra-quadro que são avaliados pelo codificador no processo do modo de decisão poderiam ser evitados.

Por conta desta relação evidente entre o problema de classificação e o problema de decisão de modo, já existem trabalhos que aplicaram o problema de classificação neste contexto nos codificadores de vídeo HEVC e VVC. Em Zhang; Kwong; Wang (2020), por exemplo, foram apresentados diversos trabalhos que aplicaram técnicas de aprendizado de máquina supervisionado com o objetivo de reduzir complexidade no modo de decisão do HEVC. Ainda que exista esta relação entre os dois problemas, aplicar o problema de classificação ao problema de decisão de modo não é uma tarefa trivial e dois desafios destacam-se. O primeiro desafio se dá na questão de determinar quais *features* são relevantes para o problema de classificação que está sendo considerado. Para o problema de decidir qual modo de predição intra é mais adequado a um bloco, o desafio se dá em determinar quais *features* são relevantes para esta tomada de decisão, sejam estas *features* extraídas do próprio processo de codificação ou calculadas externamente a partir das sequências de vídeo consideradas. O segundo desafio se dá no fato de que existem diversos pontos onde um classificador pode ser aplicado, e dentro de cada ponto existem diversas formas de se modelar o problema. Somente no ponto de decidir qual modo de predição intra deve ser avaliado pelo processo do modo de decisão, existem diversos classificadores que podem ser pensados por conta da alta quantidade de tipos de modo de predição intra existentes, e ainda dentro de cada tipo de modo de predição intra, a alta quantidade de modos.

Sendo assim, no próximo capítulo é apresentado o resultado de uma pesquisa de trabalhos relacionados. Esta pesquisa foi realizada com o objetivo de elucidar quais

features costumam ser consideradas ao realizar um processo de redução de complexidade em um codificador, seja este processo através de um modelo de aprendizado de máquina supervisionado ou não, bem como quais tipos de classificação costumam ser realizadas quando considera-se um modelo de aprendizado de máquina supervisionado.

4 TRABALHOS RELACIONADOS

Neste capítulo são apresentados trabalhos encontrados ao realizar uma pesquisa sobre redução de complexidade no processo de decisão de modo da predição intra-quadro dos codificadores HEVC e VVC. O foco da pesquisa foi em trabalhos que desenvolveram métodos para a redução da quantidade de modos que são avaliados no processo de decisão de modo da predição intra-quadro. Entretanto, também são citados alguns trabalhos que reduzem a quantidade de particionamentos que são avaliados por ambos os codificadores.

Buscou-se descrever para cada trabalho seu objetivo, a metodologia seguida para atingir tal objetivo e os resultados. Na metodologia, quando presentes, são destacados alguns pontos do trabalho, tais como análise estatística por trás da ideia de redução de complexidade e *features* que foram consideradas no trabalho. Também na metodologia é descrito como o algoritmo proposto foi capaz de reduzir a complexidade da predição intra-quadro. Nos resultados, são descritos os testes que foram realizados no trabalho e os ganhos e perdas obtidos em TS (*Time Saving*) e eficiência de codificação, respectivamente. O TS dos trabalhos é medido ao comparar os tempos de codificação do codificador de referência, sendo estes o HM ou o VTM, com os tempos de codificação da mesma versão do codificador de referência após implementar o(s) método(s) proposto(s). Já a eficiência de codificação dos trabalhos é medida através da métrica BD-BR (*Bjontegaard Delta Rate*) (BJONTEGAARD, 2001), que calcula a variação na taxa de bits para uma mesma taxa de qualidade entre o codificador de referência e o codificador modificado com a solução proposta. Valores positivos de BD-BR indicam aumento na taxa de bits para uma mesma qualidade visual, que por consequência demonstram uma perda em eficiência de codificação.

Nas seções 4.1 e 4.2 são apresentados os trabalhos desenvolvidos para os codificadores HEVC e VVC, respectivamente. Na seção 4.3 é feita uma conclusão acerca dos trabalhos encontrados e são apresentadas quatro tabelas, duas para o HEVC e duas para o VVC, as quais resumem os trabalhos de acordo as reduções de complexidade obtidas e as técnicas utilizadas.

4.1 Redução de Complexidade na Predição Intra-Quadro do Codificador de Vídeo HEVC

Em **Da silva; Agostini; Silva cruz (2012)** o objetivo foi reduzir a quantidade de modos de predição intra que são avaliados no processo do modo de decisão do HEVC. Para tanto, os autores dividem uma PU em blocos de tamanho 2x2 e para cada bloco aplicam filtros que calculam a força das bordas em cinco direções diferentes: horizontal, vertical, duas diagonais (45º e 135º) e não direcional. Após calcular a força das bordas em cada uma das direções para cada um dos blocos 2x2 dentro da PU, a direção que obtiver o maior valor será selecionada. A partir da direção selecionada, uma lista de modos de predição intra direcionais associada a esta direção é escolhida para ser avaliada pelo SATD. Esta lista contém 11 modos de predição intra, sendo nove direcionais e dois modos não direcionais (DC e Planar). Com os custos gerados pelo SATD, apenas os cinco e três modos com menor custo mais os MPM são selecionados para os tamanhos de PU 4x4 à 8x8 e 16x16 à 64x64, respectivamente. O método proposto foi implementado no HM 4.0 e cinco sequências de vídeo da CTC do HEVC foram codificadas com a configuração AI com valores de QP 22, 27, 32 e 37. Como resultados, foram obtidas médias de 18,88% e 0,9% de TS e BD-BR, respectivamente.

No trabalho de **Zhang; Zhao; Xu (2012)** o objetivo foi reduzir a quantidade de modos de predição intra que são avaliados no processo RDO do HEVC. O método baseia-se no fato de que existe uma alta probabilidade de que o primeiro ou o segundo modo com menores custos do RMD sejam os melhores modos após a etapa do RDO. Sendo assim, com o objetivo de reduzir a quantidade de modos que são testados no RDO, duas metodologias foram desenvolvidas, uma para PUs de tamanho 4x4 à 8x8 e outra para PUs de tamanho 16x16 à 32x32. Para as PUs de tamanho 4x4 à 8x8, é feita uma análise de acordo com os dois primeiros modos de predição intra presentes na lista pós-RMD. Se ambos são angulares e adjacentes, somente estes dois modos, o próximo modo angular presente na lista e os MPM são avaliados pelo RDO. Por outro lado, se ambos são angulares porém não adjacentes, somente estes dois modos mais os MPM são avaliados pelo RDO. Além disso, se o DC ou o Planar estão presentes na lista pós-RMD, estes também são avaliados pelo RDO. Sendo assim, para as PUs de tamanho 4x4 à 8x8, o autor reduz de oito para no mínimo dois e no máximo cinco modos a serem testados no RDO. Por outro lado, como para as PUs de tamanho 16x16 à 32x32 a lista pós-RMD possui apenas três modos, decidiu-se avaliar apenas o DC e o Planar no processo do RDO quando o primeiro modo na lista pós-RMD não é angular. Para avaliar o método proposto, os autores implementaram a técnica no HM 4.0 e executaram testes de acordo com a CTC do HEVC. Como resultados, obteve-se uma média de 1.06% de BD-BR e uma média de 20% de TS.

Os autores **Fang; Chang; Chung (2013)** propuseram uma forma de reduzir a

quantidade de modos de predição intra que são avaliados no processo de decisão de modo do HEVC. Para tanto, uma PU de tamanho NxN é dividida em cinco partes de tamanho N/2xN/2, sendo uma destas a parte central. Após, para cada parte da PU, as distribuições de energia nos ângulos iguais a 0º, 45º, 90º e 135º são calculadas. Ao fim deste processo, o ângulo que obteve menor distribuição de energia é selecionado, e a partir deste ângulo são selecionados até nove modos de predição intra direcionais a serem avaliados pelo SATD. Ao obter o resultado da primeira avaliação do SATD, o modo que obteve o melhor custo e seus dois modos adjacentes, mais o DC e o Planar, passam por uma segunda avaliação do SATD, onde ao final da segunda avaliação do SATD três ou oito modos com o melhor custo de SATD são selecionados de acordo com o tamanho da PU para serem avaliados pelo RDO. O método foi implementado no HM 6.1 e os testes foram realizados com 14 sequências de vídeo da CTC do HEVC com valores de QP 20, 24, 28 e 32 e com o arquivo de configuração AI. Como resultados, obteve-se uma média de TS de 68% no processo do RMD com uma média de BD-BR de 0,65%.

Em **Chen et al. (2014)** o objetivo foi reduzir tanto a quantidade de particionamentos quanto a quantidade de modos de predição intra que são avaliados no processo de decisão de modo do HEVC. A redução da quantidade de modos de predição intra que são avaliados é baseada em estatísticas de gradiente dos pixels, obtidas através do operador Sobel. Estas estatísticas são obtidas após o processo RMD, e a partir do gradiente de cada pixel e de dados em uma tabela que associam intervalos de gradiente aos modos de predição intra direcionais do HEVC, é gerado um contador com os modos de predição intra direcionais do HEVC e o número de vezes que eles foram associados a PU de acordo com os gradientes dos pixels. A partir deste contador, um modo é descartado da lista a ser avaliada pelo RDO quando ele não possui ocorrências ou quando um de seus dois modos vizinhos possuem um número de ocorrências menor que N*N/33, onde N é o tamanho da PU. Além disto, para manter um balanço entre redução de complexidade e eficiência de codificação, caso restem apenas dois modos na lista do RDO após este processo, uma busca refinada ao redor do modo com o melhor custo RMD é feita com base nas estatísticas de gradiente dos pixels obtida anteriormente e no custo do RDO até que se ache o modo com melhor custo. Em sequência, para diminuir o número de particionamentos que são avaliados, decidiu-se alterar a ordem em que os particionamentos são feitos e verificar a homogeneidade das PUs. Na questão da ordem dos particionamentos, uma PU sempre irá iniciar no nível de particionamento 1 (32x32). A partir deste nível, ela pode ir para o nível 0 (64x64) caso as quatro PUs sejam homogêneas e possuam um mesmo modo de predição; ficar neste nível caso as quatro PUs sejam homogêneas mas não possuam o mesmo modo de predição; ou ir para o nível de particionamento 3 (8x8) caso as PUs sejam consideradas heterogêneas. No nível 3, o mesmo procedimento é adotado, po-

dendo as PUs permanecerem no nível 3, subirem para o nível 2 ou descerem para o nível 4. Uma PU é considerada homogênea pelo algoritmo quando mais de 80% da distribuição dos gradientes de cada pixel está associada a um modo de predição intra ou quando o modo de predição intra da PU é o DC ou o Planar. O método proposto foi implementado no HM 8.0 e 14 sequências de vídeo recomendadas pela CTC do HEVC foram codificadas com valores de QP 22, 27, 32 e 37. Como resultados, obteve-se médias de TS de 28,45% e 42,84% e médias de BD-BR de 0,53% e 0,61% ao ativar o método que reduz a quantidade de modos de predição intra que são avaliados e ao ativar ambos os métodos, respectivamente.

O objetivo do trabalho de **Gwon; Choi; Youn (2015)** foi reduzir a quantidade de modos de predição intra que são avaliados tanto pelo processo RMD quanto pelo processo RDO do HEVC. Para reduzir a quantidade de modos que são testados no RMD, um método baseado em direção de textura da PU foi proposto. A direção de textura foi obtida de forma muito semelhante a apresentada no trabalho Da silva; Agostini; Silva cruz (2012), já descrito anteriormente. Entretanto, uma modificação foi feita para que se leve em consideração a relação entre a quantidade de sub-blocos dentro de uma PU que apresentaram a direção de textura que foi considerada majoritária e a quantidade total de sub-blocos dentro da PU. Foi apresentado pelos autores que conforme esta relação se aproxima do valor um, cresce a probabilidade de que o melhor modo esteja no subconjunto associado a textura majoritária identificada na PU. Ao utilizar esta métrica em uma função cúbica que foi modelada de forma a representar a densidade da probabilidade de que a direção de textura detectada seja adequada para a PU em questão, reduz-se a quantidade de modos da lista do RMD quando o resultado desta função está acima de um treshold. Por outro lado, para reduzir o número de modos que são testados pelo RDO os autores adotaram três técnicas. A primeira técnica consiste no método de Lee (LEE; PARK; JANG, 2014), o qual pula o processo inteiro do RDO quando detecta que o modo com menor custo RMD é um dos MPM. Caso o método de Lee não obtenha sucesso, duas técnicas são rodadas simultaneamente com o objetivo de gerar listas com uma quantidade menor de modos a serem testados pelo RDO. A primeira delas consiste no método de Zhang (ZHANG; ZHAO; XU, 2012), descrito anteriormente. A segunda foi desenvolvida pelos autores e é baseada em um modelo bayesiano, o qual realiza uma classificação binária. Esta classificação binária é feita no sentido de detectar a probabilidade de que, para um determinado modo de índice i na lista pós-RMD, algum dos modos com índice menor que i seja o melhor modo (classe A) ou algum dos modos com índice maior ou igual a i seja o melhor modo (classe B). Sempre que o modelo bayesiano classifica um modo da lista pós-RMD com índice i como classe A, os modos com índice maiores ou iguais a i são removidos da lista pós-RMD. Para treinar este modelo, foi utilizada uma feature que é calculada para cada modo que está na lista pós-RMD, exceto o primeiro. Seu

cálculo é a divisão do menor custo RMD, que está relacionado ao primeiro modo da lista, por cada custo RMD dos modos subsequentes. Esta feature foi escolhida pois os autores mostraram que conforme essa divisão se aproxima de zero, cresce a probabilidade de que um dos modos com índice menor ou igual a i seja o melhor. Este modelo foi treinado com oito sequências de vídeo das classes A à D e testado com quatro sequências de vídeo também das classes A à D da CTC do HEVC, sendo que os vídeos para treinamento e teste são distintos. Por fim, todos os métodos foram implementados no HM 12.0 e quatro sequências de vídeo das classes A à D, as quais não foram utilizadas no conjunto de treinamento do modelo, foram codificados com a configuração AI. Como resultados, obteve-se uma média de TS de 30,3% e uma média de BD-BR de 0,9%.

Em **Jamali; Coulombe; Caron (2015)** o objetivo foi reduzir a quantidade de modos de predição intra que são avaliados no processo do modo de decisão do HEVC. Baseado no operador Sobel, o algoritmo proposto calcula o gradiente horizontal e vertical para cada um dos pixels da LCU, o que possibilita que estes sejam reutilizados nas profundidades inferiores. Após calcular o gradiente para cada pixel, o ângulo da direção é estimado para cada pixel através da divisão entre o gradiente horizontal e o gradiente vertical. A partir desta estimativa, cada pixel é associado de acordo com o ângulo da direção a três modos intra direcionais, sendo o primeiro obtido através do mapeamento direto entre o ângulo obtido e um modo de predição intra direcional do HEVC e os outros dois sendo os modos de predição intra adjacentes. Tendo estas informações, cada pixel é percorrido de forma a incrementar a probabilidade dos três modos associados a ele, tendo um peso maior o modo diretamente mapeado pelo ângulo obtido. No fim, isto gera um histograma dos 33 modos de predição intra direcionais do HEVC, e deste histograma os oito e os três modos com maior valor são selecionados para os tamanhos de PU de 4x4 à 8x8 e de 16x16 à 64x64, respectivamente. Junto a estes modos, ainda são adicionados o DC e o Planar dada a sua alta taxa de ocorrência e até cinco modos MPM. Por fim, antes do RDO os modos são ordenados pelo seu custo medido a partir do SATD. Caso algum dos modos selecionados nos MPM seja o mesmo que obteve o menor SATD, o RDO é pulado e este modo é considerado como melhor. Se isto não ocorrer, alguns modos ainda podem ser removidos da lista a ser avaliada pelo RDO caso exista um alto *gap* entre o modo com menor SATD e o modo com maior SADT. Este gap é medido através de uma função que considera um treshold adaptativo de acordo com o tamanho da PU. O algoritmo proposto foi implementado no HM 15.0 e 20 sequências de vídeo das classes A à E da CTC do HEVC foram codificadas com a configuração AI sob os valores de QP 22, 27, 32 e 37. Como resultados, o trabalho obteve médias de 35,6% e 1,07% de TS e de BD-BR, respectivamente.

Os autores **Cristina; Mihnea; Ionut (2016)** propuseram uma metodologia para re-

duzir a quantidade de particionamentos e modos intra que são testados ao basear-se na complexidade da CU atual. Para medir a complexidade da textura, foi utilizado uma adaptação do algoritmo *Histogram of Oriented Gradients* (HOD) (DALAL; TRIGGS, 2005). No trabalho, um frame é dividido em blocos de tamanho 16x16 (sem sobreposição) e para cada bloco são calculados os gradientes nas direções horizontal e vertical, os quais são utilizados posteriormente para calcular a magnitude e a direção dos gradientes. A partir das direções e magnitudes de gradientes obtidos para cada pixel do bloco 16x16, um histograma é gerado de forma a contar o número de pixels que teve sua direção de gradiente inclusa em subintervalos de tamanho igual (12°) entre 0° e 180°. Após obter este histograma, os blocos 16x16 adjacentes são combinados e formam um bloco 32x32 quando seus histogramas são semelhantes. O mesmo processo é feito para níveis acima de 32x32 e abaixo de 16x16. A similaridade dos histogramas dos blocos é verificada de acordo com a comparação da SAD entre os histogramas e um treshold. Para reduzir o número de modos testados no RDO, os blocos 16x16 são divididos em grupos, onde o primeiro grupo contem os blocos cujo histograma apresentou dois subintervalos de direções de gradiente com alta magnitude, o segundo grupo contem blocos cujo histograma apresentou uma distribuição uniforme de magnitude entre os subintervalos de direções de gradiente e o terceiro grupo contem os blocos cujo histograma apresenta três ou mais subintervalos de direções de gradiente com alta magnitude. Ao ter esta classificação, o número de modos que são testados no RDO são reduzidos a apenas três de acordo com a classificação do bloco, onde dentre estes três o modos DC ou o modo Planar está sempre incluso. Não foi especificado em qual versão do HM o método foi implementado e o teste foi feito apenas com duas sequências de vídeo. Como resultados, obteve-se em média 4,65% de redução de tempo de execução e 0,05% de acréscimo de BD-BR.

Em **Duanmu; Ma; Wang (2016)** foram combinadas diversas formas de reduzir a complexidade no processo do modo de decisão do HEVC através de modelos de aprendizado de máquina supervisionado. O primeiro passo consiste em classificar o bloco que está sendo predito em duas classes possíveis, sendo estas *Natural Image Block* (NIB) ou *Screen Content Block* (SCB). Caso o bloco seja classificado como SCB, apenas um subconjunto de modos de predição intra que costumam ser utilizados para este tipo de bloco são avaliados pelo processo de decisão de modo. Caso o bloco seja classificado como NIB, uma nova classificação é realizada com o objetivo de predizer se o bloco deve ser particionado ou não. Quando classifica-se que o bloco deve ser particionado, a avaliação dos modos de predição intra para o nível de profundidade atual é pulada, e quando classifica-se que o bloco não deve ser particionado, é ainda feita uma nova classificação para saber se o conteúdo do bloco em questão é direcional ou não direcional. Novamente, de acordo com esta última classificação, apenas um subconjunto de modos de predição intra são avaliados pelo processo do

modo de decisão. Para todas as classificações o tipo de modelo utilizado foi árvore de decisão. Dentre as *features* nos conjuntos de dados, estão a soma das diferenças DC absolutas entre as sub-cus que são geradas ao realizar um particionamento, variância da CU, a qual indica a homogeneidade, informações sobre o gradiente da CU calculado a partir do operador Sobel, como direção do gradiente, magnitude e porcentagem de gradientes que foram iguais a zero. Existe um conjunto de dados para cada tamanho de bloco, sendo estes 64x64, 32x32, 16x16, 8x8 e 4x4. Para treinar o modelo, como os autores possuem uma alta quantidade de exemplos para cada conjunto de dados, o *cross-validation* foi desconsiderado e o conjunto de dados foi dividido em metade para treinamento e metade para validação. Os modelos foram implementados no SCM 4.0 e 13 sequências de vídeo do tipo SC da CTC do HEVC foram codificadas com o arquivo de configuração AI e com os valores de QP 22, 27, 32 e 37. Em média, obteve-se um TS de 52,0% com um BD-BR de 2,40%.

Liu et al. (2016) propôs uma forma de reduzir o número de particionamentos e o número de modos de predição intra que são verificados no processo de decisão de modo do HEVC, onde o primeiro método baseia-se na complexidade de textura da CU e o segundo método baseia-se na direção de textura da CU. A complexidade de textura da CU foi obtida através do desvio padrão da complexidade de pixel oposto, onde esta segunda complexidade é medida através da multiplicação entre o desvio padrão da CU atual e a MAD entre cada pixel e seus pixels vizinhos. Foi verificado pelos autores que se a profundidade atual é adequada ou não adequada para a CU, a medida calculada gera valores baixos ou altos, respectivamente. Sendo assim, um nível de profundidade não é avaliado quando a medida calculada está acima de um treshold superior; e um nível de profundidade é selecionado como o melhor, acarretando em os demais níveis não serem avaliados, quando a medida calculada está abaixo de um treshold inferior. Ambos os tresholds são calculados de forma adaptativa com base na profundidade atual e no valor do QP. Por outro lado, para obter a direção de textura da PU, os autores calcularam o desvio padrão dos pixels em oito direções. Para reduzir a complexidade deste cálculo, do tamanho de PU 8x8 para cima é feita uma subamostragem dos pixels que são considerados no cálculo. Após calcular o desvio padrão nas oito direções, uma lista para ser avaliada pelo RMD é gerada com base nas três direções que obtiveram o menor desvio padrão. Esta lista contém no máximo treze modos de predição intra contendo sempre o DC e o Planar. Ao obter o resultado do RMD, o tamanho da lista que é avaliada pelo RDO também foi reduzido, onde são considerados apenas os 2, 2, 2, 3 e 3 melhores modos retornados pelo RMD para os tamanhos de PU 64x64, 32x32, 16x16, 8x8 e 4x4, respectivamente. O método proposto foi implementado no HM 16.0 e os testes foram feitos ao codificar, com diferentes configurações do HEVC (AI, RA, LD, LD-P) e com diferentes QPs (22, 27, 32 e 37), 24 sequências de vídeo distribuídas pelas classes A à F da CTC do HEVC. Como resul-

tados, obteve-se médias de TS de 37,2%, 29,5% e 56% e médias de BD-BR de 0,2%, 0,7% e 1,0% ao habilitar a técnica que reduz o número de particionamentos avaliados, ao habilitar a técnica que reduz o número de modos de predição intra avaliados e ao habilitar ambas as técnicas, respectivamente.

Em **Wang; Xue (2016)** o objetivo foi reduzir a quantidade de particionamentos e modos de predição intra que são avaliados no processo de decisão de modo do HEVC. Para o algoritmo de decisão de tamanho da CU, o método de Otsu foi empregado para determinar as áreas da LCU que devem ser particionadas. Como este é um método custoso, ele é aplicado apenas na LCU, sendo que os demais níveis baseiam-se nos resultados já calculados. A partir do método de Otsu, o trabalho propôs utilizar duas medidas para decidir a profundidade de uma LCU, sendo estas a diferença absoluta entre o treshold separador e o valor médio dos pixels de luminância da LCU e a maior variância inter-classe. Esta decisão foi feita ao realizar um levantamento estatístico com 10.000 LCUs que possuem diferentes valores de profundidade máxima e foram selecionadas pseudo-aleatoriamente, o qual mostrou que existem padrões entre estas duas medidas e os valores de profundidade máxima. A partir destes padrões, um algoritmo que leva em consideração o QP e estas duas medidas obtidas a partir do método de Otsu foi proposto para pular determinadas profundidades quando algumas condições são atendidas. Por outro lado, para o algoritmo de decisão de modo da PU, o ângulo do gradiente da PU é obtido e mapeado para um modo intra direcional. Inspirado pela forma de calcular gradientes a partir do operador Sobel, foram propostas novas máscaras que adaptam-se ao tamanho da PU. São selecionados até 7 modos de predição intra para as PUs do tamanho 64x64 ao 16x16 e até 17 modos de predição intra para as PUs do tamanho 8x8 ao 4x4 com base no modo de predição intra direcional associado ao ângulo do gradiente da PU e seus vizinhos e nos MPM. Os modos DC e Planar sempre são inclusos na avaliação do RMD. Para fins de teste, os métodos propostos foram implementados no HM 12.0 e após 18 sequências de vídeo distribuídas pelas classes A à E da CTC do HEVC foram codificadas sob a configuração AI e sob os valores de QP 22, 27, 32 e 37. Como resultados, os métodos geraram em média 33,32% de TS e em média 0,81% de BD-BR.

Yang; Pi (2016) propôs uma metodologia com o objetivo de reduzir a quantidade de modos de predição intra que são avaliados no processo de decisão de modo do HEVC. Esta redução foi feita com base na direção de textura da PU, sendo esta obtida através do operador Sobel. Como o Sobel torna-se custoso a medida que o tamanho da PU onde o mesmo deve ser aplicado cresce, decidiu-se trabalhar apenas com os tamanhos de PU 8x8 e 4x4. Ainda, para a PU de tamanho 4x4, é feita uma análise do nível de significância da direção de textura obtida. Isto se deve ao fato deste tamanho de PU ser muito pequeno, o que resulta na estimativa da direção de textura obtida a partir dos seus pixels nem sempre ser boa. Sendo assim, sempre que este nível

de significância é baixo, a lista de modos de predição intra a serem testados pelo processo RDO é gerada a partir dos melhores modos obtidos após o processo RMD. Caso contrário, a lista é gerada com base nos modos de predição intra relacionados ao ângulo da direção de textura obtida. A lista gerada no método proposto nunca irá conter mais do que quatro modos de predição intra a serem testados, sendo que dois destes sempre serão o DC e o Planar. O método proposto foi implementado no HM 16.0. Nos testes, foram codificadas dez sequências de vídeo distribuídas pelas classes A à E da CTC do HEVC com a configuração AI e valores de QP iguais a 22, 27, 32 e 37. Como resultados, foi obtida uma média de TS de 30,6% e uma média de BD-BR de 0,8%.

Zhang et al. (2016) propôs uma metodologia para reduzir tanto a quantidade de modos de predição intra quanto a quantidade de particionamentos que são avaliados pelo HEVC. Para reduzir a quantidade de modos de predição intra que são avaliados pelo HEVC, foi proposto utilizar informações sobre o gradiente do bloco que está sendo codificado para reduzir a quantidade de modos de predição intra que são avaliados pelo RMD e pelo RDO. Estas informações são obtidas ao aplicar o operador Sobel sobre o bloco e obter os gradientes horizontais e verticais. Após, é feita a média destes gradientes em cada uma das direções e de acordo com a comparação destas médias e tresholds, são selecionados apenas subconjuntos de modos de predição intra a serem avaliados pelo RMD. Já para reduzir a quantidade de particionamentos que são avaliados, são combinadas uma estratégia com base nas médias dos gradientes horizontais e verticais obtidas previamente e em dois modelos SVM, um que decide realizar um *early termination* caso o bloco possua baixa complexidade de textura e o outro que decide ir para a próxima profundidade sem avaliar a profundidade atual caso o bloco possua uma alta complexidade de textura. Primeiro, verifica-se se de acordo com as médias de gradiente obtidas o bloco atual pode ser considerado homogêneo. Esta verificação é feita ao comparar estas medidas com tresholds que são atualizados de tempos em tempos durante processo de codificação. Caso o bloco seja considerado homogêneo, um *early termination* é realizado. Caso contrário, o processo segue para os modelos SVM. O primeiro modelo, que busca predizer um *early skip*, faz esta predição baseado nas profundidades escolhidas para os blocos vizinhos e nos custos RMD obtidos para os modos de predição intra. Já o segundo modelo busca predizer um *early termination* e baseia-se também nas profundidades escolhidas para os blocos vizinhos e nos custos RDO obtidos para os modos de predição intra. Os conjuntos de dados utilizados nos treinamentos dos SVM foram obtidos ao codificar quatro sequências de vídeo no HM 14.0. O método foi implementado no HM 14.0 onde 20 sequências de vídeo das classes A à E da CTC do HEVC foram codificadas com a configuração AI e com os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se médias de TS de 15,2%, 43,1% e 53,9% e médias de BD-BR de 0,18%,

0,50% e 0,70% ao habilitar o método que reduz a quantidade de modos de predição intra que são avaliados, o método que reduz a quantidade de particionamentos que são avaliados e ambas as técnicas, respectivamente.

Em **Benhajyoussef; Ezzedine; Bouallègue (2017)** o objetivo foi reduzir a quantidade de particionamentos e modos de predição intra que são avaliados durante o processo de decisão de modo do HEVC, onde ambos os métodos foram baseados em estimativa de gradiente. Sendo assim, a direção e magnitude do gradiente de cada pixel em uma PU são estimados através do operador Prewitt. A partir destas estimativas, é gerado um histograma para cada PU contendo um peso associado a cada modo de predição intra direcional do HEVC. Este peso é obtido ao identificar, para cada pixel, qual modo de predição intra está associado ao gradiente para este pixel. Após esta identificação, é somado ao peso do modo o valor da magnitude de cada pixel onde o gradiente corresponde ao modo em questão e também a quantidade de vezes que o modo foi associado à pixels dentro da PU. Além disto, neste processo os dois modos vizinhos também tem seu peso incrementado, porém com um fator de multiplicação menor que o modo central. Ao obter este histograma, a lista a ser avaliada pelo RMD pode conter até 15, 14, 8, 6 e 5 modos de predição intra de acordo os maiores valores obtidos no histograma. Cabe destacar que os modos DC e Planar sempre estão inclusos nesta lista. Com a lista de modos ordenada pelos custos gerados pelo RMD, há ainda a possibilidade de reduzir de oito para três o número de modos avaliados pelo RDO para as PUs de tamanho 4x4 e 8x8 caso o melhor modo retornado pelo RMD seja o DC ou caso os três melhores modos retornados pelo RMD correspondam com os três modos com maior peso no histograma. Já o método de redução do número de particionamentos é baseado na complexidade de textura da CU e é aplicado somente a partir do nível de profundidade 2 (16x6). A complexidade de textura da CU atual é medida através de duas formas dependendo do tamanho da PU. Se a PU é do tamanho 8x8, a medida é obtida através da soma entre a média da magnitude dos gradientes da PU e a soma das diferenças absolutas entre esta média e as médias de magnitude das PUs que serão geradas caso o particionamento ocorra. No caso da PU de tamanho 16x16, somente a soma das diferenças absolutas é considerada. Sempre que a medida que descreve a complexidade da textura estiver abaixo de um treshold, a PU em questão não é particionada. Existe um treshold único para os dois tamanhos de PU em questão e estes foram definidos de forma empírica. O algoritmo foi implementado no HM 14.0 e 20 sequências de vídeo das classes A à E da CTC do HEVC foram codificadas com o arquivo de configuração AI e com os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se médias de TS de 31,8%, 31,0% e 42,8% e médias de BD-BR de 0,9%, 0,7% e 1,1% ao ativar o método que reduz o número de modos de predição intra avaliados, o método que reduz o número de particionamentos avaliados e ambos os métodos, respectivamente.

Liu et al. (2017) propôs uma metodologia baseada em aprendizado de máquina supervisionado para decidir o nível de profundidade de uma CU de acordo com informações da complexidade de textura da CU atual. Para medir a complexidade de textura da CU atual, diversas *features* foram consideradas, dentre estas *Neighboring Mean Squared Error* (NMSE), a qual calcula a variância da CU atual para cada pixel de acordo com os oito pixels em torno, os gradientes da CU medido pelo operador Sobel em quatro direções distintas, a variância das variâncias das sub-cus que são geradas por um particionamento também é levada em consideração e por fim o QP ao qual o vídeo está sendo codificado. A partir de um conjunto de dados contendo estas *features* e as decisões tomadas pelo codificador em relação ao particionamento, um modelo SVM foi treinando com o objetivo de classificar uma CU em três classes possíveis, sendo estas CUs com alta complexidade de textura, CUs com baixa complexidade de textura e CUs indeterminadas. Para as CUs com alta complexidade de textura, o processo de avaliação para a CU em questão é pulado e parte-se direto para o próximo nível de profundidade, ao passo que para as CUs com baixa complexidade de textura um *early termination* é realizado e próximas profundidades não são avaliadas. Por fim, quando uma CU é classificada como indeterminada, o processo natural do HEVC é seguido. Quanto aos dados obtidos para treinamento, só é mencionado pelo autor que foram extraídos do HM 15.0 ao codificar duas sequências de vídeo. Para fins de teste, o método proposto foi implementado no HM 15.0 e 24 sequências de vídeo da CTC do HEVC foram codificadas com a configuração AI e com os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se uma média de TS de 59,6% com uma média de BD-BR de 1,26%.

No trabalho de **Lu; Yu; Jin (2018)** o objetivo foi diminuir a quantidade de particionamentos e a quantidade de modos de predição intra que são avaliados em uma LCU no processo de decisão de modo do codificador HEVC. A redução da quantidade de particionamentos que são avaliados foi feita com base em dois fatores: a complexidade de textura do bloco medida no domínio das frequências a partir dos coeficientes da transformada DCT; e as profundidades das LCUs vizinhas e da LCU co-localizada no *frame* anterior. Já a redução da quantidade de modos de predição intra que são testados foi obtida através de um estudo apresentado no trabalho, o qual mostra que existe uma alta dependência entre o melhor modo de codificação obtido para uma PU e o melhor modo de codificação obtido para a mesma PU em um nível acima na árvore de particionamentos ou co-localizada no *frame* anterior. A partir destas dependências, os autores propuseram diferentes formas de gerar uma lista contendo modos de predição intra a serem testados pelo processo RMD. Ao final do processo RMD, no máximo dois modos intra são selecionados de acordo com o seu *Hadamard Cost* para passarem pelo processo RDO. O método proposto pelos autores foi implementado no HM 13.0 e os testes foram realizados ao codificar 17 sequências de vídeo

das classes A à E da CTC do HEVC com a configuração AI e com os valores de QP 22, 27, 32 e 37. Como resultados, os autores obtiveram médias de TS de 28,8%, 39,0% e 55,2% e médias de BD-BR de 0,30%, 1,51% e 1,60% ao habilitar a técnica que reduz a quantidade de particionamentos que são avaliados, ao habilitar a técnica que reduz a quantidade de modos de predição intra que são avaliados e ao habilitar ambas técnicas, respectivamente.

Em **Sun et al. (2019)** o objetivo foi diminuir o número de particionamentos e o número de modos de predição intra que são avaliados durante o processo de decisão de modo do HEVC. Este método baseia-se na complexidade e na direção de textura da CU atual, ambas obtidas a partir da variância dos pixels em quatro direções, sendo estas horizontal, vertical, diagonal superior esquerda e diagonal superior direita. Após medir a variância nestas quatro direções e selecionar a variância direcional com maior valor, a complexidade de textura da CU atual pode ser considerada homogênea quando este valor está abaixo de um treshold, complexa quando este valor está acima de um treshold ou indeterminada quando nenhum dos casos anteriores ocorre. Para obter os tresholds, verificou-se a taxa de erro de profundidade ao comparar as profundidades selecionadas pelo HM 16.7 com o método implementado e as profundidades selecionadas pelo HM 16.7 original. Neste teste, foram considerados cinco sequências de vídeo contendo diferentes características de movimento, quatro valores de QPs (22, 27, 32 e 37) e tresholds variando no intervalo [1,9], sendo que ao fim deste processo foi obtido um treshold para cada QP. A partir da classificação da complexidade de textura da CU, a profundidade atual é selecionada como a melhor quando a textura é considerada homogênea ou não avaliada quando a textura é considerada complexa. Quando a textura é considerada indeterminada, o processo normal do HEVC ocorre. Na metodologia de redução dos modos que são avaliados, os modos de predição intra direcionais foram agrupados em quatro classes, o que gerou um total de 11 modos por classe, já que o DC e o Planar sempre estão inclusos. A partir destas quatro classes que estão associadas a cada uma das quatro variâncias calculadas, existem diferentes possibilidades do que pode acontecer. Primeiro, se a diferença entre a maior variância e a menor variância for menor que 10% da menor variância, a textura da PU é considerada suave, o processo do RMD é pulado e somente o DC, o Planar e os MPM são avaliados no RDO. Por outro lado, se a diferença entre a segunda menor variância e a primeira menor variância é menor do que 10% da primeira menor variância, considera-se que a PU contem duas direções de textura, e as duas classes associadas a estas direções mais os MPM são avaliadas pelo RMD. Se nenhuma das duas condições anteriores ocorrem, apenas a classe associada a direção que obteve a menor variância e os MPM são avaliados pelo RMD. Por fim, considerando o balanço entre complexidade e eficiência de codificação, decidiu-se avaliar no RDO os 5, 4, 3, 2 e 2 melhores modos retornados pelo RMD para os tamanhos de PU

4x4, 8x8, 16x16, 32x32 e 64x64, respectivamente. Para testar os métodos, ambos foram implementados no HM 16.7 e 15 sequências de vídeos distribuídas pelas classes A à E da CTC do HEVC foram codificadas sobre a configuração AI com os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se em média 57,13% de TS e 0,65% de BD-BR.

No trabalho de **Huang et al. (2020)** foi proposto um método para reduzir o número de particionamentos que são avaliados. Este método baseia-se em dois modelos, sendo o primeiro um modelo bayesiano, o qual verifica a probabilidade de cada profundidade, e o segundo um modelo SVM, o qual faz uma classificação binária que indica se a CU atual já está em uma profundidade adequada ou se a CU atual deve ser particionada. Para o modelo bayesiano, foi considerada a correlação entre a profundidade escolhida para a CU atual e as profundidades escolhidas para as CUs vizinhas. Sendo assim, o que o modelo calcula é a probabilidade de que as CUs vizinhas escolham uma profundidade i dado que a CU atual também escolheu a profundidade i. A partir da soma das probabilidades obtidas pelo modelo bayesiano para cada profundidade e para cada CU vizinha, sempre que uma profundidade obtém uma probabilidade menor ou igual a 0,05 (muito baixa) ou maior ou igual a 0,95 (muito alta) a profundidade atual é pulada ou considerada a melhor, respectivamente. Quando nenhum dos dois casos anterior ocorre, o processo segue para o modelo SVM somente quando a probabilidade é considerada alta (maior ou igual a 0,6) ou quando a probabilidade é considerada baixa (menor ou igual a 0,3). No modelo SVM, foram consideradas *features* que descrevem a complexidade da textura e o QP que está sendo utilizado para codificar o vídeo, já que estes possuem uma alta relação com as profundidades que são escolhidas para as LCUs. Dentre as *features* que descrevem a complexidade de textura do bloco, foram consideradas a variância do bloco todo e a média e a variância do bloco ao dividi-lo em duas partes iguais na horizontal e na vertical. As médias e as variâncias das duas partes na horizontal são então subtraídas e seu valor absoluto é considerado. O mesmo é feito para as médias e as variâncias na vertical. Após, as diferenças absolutas das médias e das variâncias na horizontal e na vertical são somadas. A partir desta soma final, é possível verificar se o bloco possui uma textura complexa ou não, já que esta soma resultará em valores altos quando o bloco possuir uma textura complexa. O método proposto foi implementando no HM 15.0 e 21 vídeos da CTC do HEVC foram codificados sob os QPs 22, 27, 32 e 37. Como resultados, obteve-se médias de redução de tempo de execução de 18,24% e 36,28% e médias de acréscimo de BD-BR de 0,20% e 0,80% com o modelo bayesiano e com o modelo SVM, respectivamente.

4.2 Redução de Complexidade na Predição Intra-Quadro do Codificador de Vídeo VVC

No trabalho de **Amestoy et al. (2019)**, foi proposta uma metodologia baseada em Random Forests para reduzir a quantidade de particionamentos que são testados no VVC. A classificação feita pela Random Forest consiste em predizer se para a CU atual um particionamento quaternário ou binário é mais apropriado. Para tanto, foram gerados datasets para os tamanhos de bloco 128x128, 64x64, 32x32 e 16x16. Estes datasets contém dez vídeos distribuídos pelas classes A1, A2, B, C e D da CTC do VVC, sendo que estes vídeos foram selecionados de acordo com as suas informações de SITI. Cada um dos vídeos foram codificados com a configuração RA sob os valores de QP 22, 27, 32 e 37 no JEM 7.0, onde durante a codificação foi extraído o melhor particionamento selecionado para a CU após o RDO. Estes datasets foram balanceados de forma a conter um mesmo número de exemplos por sequência de vídeo e por classe. Várias features foram consideradas no dataset, sendo um total de 19. Dentre estas, existem features que descrevem a textura da CU, como variância e gradientes, e features que foram extraídas do codificador, como informações relacionadas aos vetores de movimento da predição inter-quadros e QP. O método proposto foi implementado no JEM 7.0 e testes foram conduzidos ao codificar 18 sequências de vídeo, sendo três por classe, da CTC do VVC. As sequências utilizadas no teste não foram utilizadas no treinamento dos modelos. O arquivo de configuração utilizado foi o RA. Como resultados, obteve-se uma média de TS de 30% com uma média de BD-BR de 0,57%. Ainda, de acordo com os autores, o overhead da técnica implementada é de cerca de 0,21% do tempo total.

Nos trabalhos de **Chen et al. (2019)** e **Fan et al. (2020)** é apresentada a proposta de um algoritmo para reduzir a quantidade de particionamentos que são avaliados no VVC. Este algoritmo sempre inicia nas CUs de tamanho 32x32 e existem três pontos que são explorados, sendo o primeiro evitar que uma CU de tamanho 32x32 seja particionada, o segundo escolher entre um particionamento quadrado ou retangular e o terceiro escolher apenas uma entre as cinco estruturas de particionamentos possíveis da QTMT. Para evitar que uma CU de tamanho 32x32 seja particionada, a variância da CU em questão é calculada e esta não é particionada quando esta medida significa que a CU já é homogênea o suficiente. Na escolha entre um particionamento quadrado ou retangular, os gradientes na vertical e horizontal são calculados com o Sobel. Após, a soma do gradientes absolutos para cada pixel é calculada. Se de acordo com estas medidas o gradiente na horizontal e na vertical são muito semelhantes, apenas o particionamento quadrado é avaliado e os retangulares na horizontal e na vertical são pulados. Por fim, para escolher um dentre os cinco particionamentos possíveis, calcula-se a variância de cada um dos sub-blocos que são gerados ao

aplicar o particionamento. Após, calcula-se também a variância da variância de cada um dos sub-blocos. Por fim, o tipo de particionamento que apresentar o maior valor para esta última medida é selecionado. Isto foi feito baseando-se no fato de que esta variância da variância dos sub-blocos gerados apresenta um valor alto quando os sub-blocos possuem texturas diferentes entre si, o que justifica o particionamento aplicado ser o melhor. Todos estes três pontos são explorados em sequência pela técnica, sendo que um ponto só é explorado quando o anterior não obteve sucesso. Para medir a eficácia do método, este foi implementado no VTM 4.0 e 20 sequências de vídeo distribuídas pelas classes B à F da CTC do VVC foram codificadas com a configuração AI e com os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se uma média de TS de 53,17% e uma média de BD-BR de 1,62%.

Em **Fu et al. (2019)** foi proposta uma metodologia para reduzir o número de particionamentos que são avaliados no VVC. Esta metodologia explora dois pontos: detectar quando um particionamento binário e ternário no sentido vertical podem ser pulados e detectar quando um particionamento ternário no sentido horizontal pode ser pulado. O método baseia-se na correlação que existe entre o melhor particionamento encontrado para os sub-blocos de um super-bloco. De acordo com estatísticas levantadas no trabalho, se o melhor particionamento dos sub-blocos não é no sentido vertical existe uma alta probabilidade de que o melhor particionamento do super-bloco também não seja no sentido vertical. Além disso, o melhor modo intra selecionado para os sub-blocos também é um grande indicativo se um particionamento no sentido vertical será a melhor opção. Sendo assim, um modelo bayesiano foi treinado a partir de seis sequências de vídeo que foram codificadas no VTM 1.0 com os QPs 22, 27, 32 e 37 e com a configuração AI. As features consideradas no modelo bayesiano foram os particionamentos escolhidos para os sub-blocos e os modos intra escolhidos para os sub-blocos e para o super-bloco. De acordo com a saída deste modelo, um particionamento binário ou ternário no sentido vertical é pulado sempre que a probabilidade de um particionamento no sentido vertical é menor em relação a probabilidade de um particionamento no sentido horizontal. Como no momento em que um particionamento ternário no sentido horizontal vai ser avaliado já existe informações de RD-Cost sobre o particionamento binário nos sentidos horizontal e vertical, a decisão de pular o particionamento ternário no sentido horizontal é feita com base nestes dois RD-Costs e nos modos intra que foram selecionados. O método foi implementado no VTM 1.0 e 22 sequências de vídeo das classes A1 à E foram codificadas com os valores de QP 22, 27, 32 e 37 sob a configuração AI. Como resultados, obteve-se uma média de TS de 45% e uma média de BD-BR de 1,01%.

Os autores **Lei et al. (2019)** propuseram um método para reduzir a quantidade de particionamentos que são avaliados no VVC. Como no VVC os particionamentos binários e ternários podem ser tantos no sentido horizontal quanto no sentido vertical, o

método busca detectar se para a CU atual a avaliação de um destes particionamentos na vertical ou na horizontal pode ser pulada. Para tanto, o método busca estimar o custo de realizar o particionamento da CU atual na horizontal e na vertical. Esta estimativa é obtida ao realizar um processo de RMD simplificado para cada uma das sub CUs geradas após o particionamento, onde este processo envolve apenas 7 dos 67 modos intra direcionais do VVC. Ao detectar quais dos sete modos intra obteve um menor custo RMD, este é avaliado pelo RDO e assim tem-se uma estimativa do custo para cada uma das sub CUs que foram geradas. Com esta estimativa, é possível verificar se o particionamento na horizontal ou na vertical obteve um melhor desempenho e assim pular um dos dois. Quando ambos tipos de particionamento obtém estimativas de custo parecidas, o processo normal do VVC é realizado. O método proposto foi implementado no VTM 3.0 e os primeiros 200 frames de 22 vídeos da CTC do VVC foram codificados com a configuração AI sob os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se uma média de TS de 40,87% e uma média de BD-BR de 0,85%.

Park; Kang (2019) propuseram uma metodologia para reduzir a complexidade dos particionamentos ternários. Esta metodologia baseia-se no fato de que, em média, 88,9% dos particionamentos ternários que são testados durante a codificação não resultam no melhor RD-Cost. Com o objetivo de explorar esta redundância, foi verificado que quando o custo de uma partição binária no sentido horizontal possui um menor RD-Cost que uma partição binária no sentido vertical, existe uma alta probabilidade de que uma partição ternária no sentido horizontal também resulte em um RD-Cost menor que uma partição ternária no sentido vertical. O mesmo ocorre para o caso inverso. Sendo assim, a proposta é pular a avaliação de partições ternárias no sentido horizontal ou vertical de acordo com as probabilidades calculadas com base no RD-Cost das partições binárias já avaliadas. Este método foi implementado no VTM 4.0 e 22 sequências de vídeo das classes A1 à E da CTC do VVC foram codificadas. Como resultados, obteve-se uma média de TS de 34% e uma média BD-BR de 1,02%. Além disso, em média a redução de complexidade das partições ternárias foi de 62% em comparação ao VTM 4.0.

Peng et al. (2019) desenvolveu uma metodologia para reduzir o número de particionamentos que são avaliados para um CU no VVC. Para reduzir o número de particionamentos que são avaliados, uma CU é classificada de acordo com a sua textura em três categorias: simples, comum ou complexa. Se a textura da CU é considera simples, somente o modo de particionamento NS é avaliado. Por outro lado, se a textura da CU é considerada complexa, o modo de particionamento NS não é verificado. E por fim, se a textura da CU é considerada comum, o fluxo normal do VVC é adotado. Esta classificação da textura da CU se dá com base na comparação do desvio padrão dos pixels da CU e dois tresholds, um inferior e um superior. Estes dois tresholds são obtidos a partir do primeiro frame do vídeo, o qual é codificado normalmente no VVC.

Ainda, para as CUs que são consideradas comuns ou complexas, uma avaliação é feita para possivelmente pular os particionamentos no sentido horizontal ou vertical. Para tanto, a variância dos pixels da CU nos sentidos horizontal e vertical é medida. Esta última técnica é aplicada somente a CUs quadradas. O método foi implementado no VTM e 18 sequências de vídeo foram codificadas com os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se médias de TS de 22,83%, 29,30% e 51,72% e médias de BD-BR de 1,40%, 1,00% e 2,53% ao ativar a técnica que avalia somente o modo de partição NS ou pula o modo de partição NS, ao ativar a técnica que possivelmente pula as partições no sentido horizontal ou vertical e ao ativar ambas as técnicas, respectivamente.

No trabalho de **Tang et al. (2019)** foi proposta uma metodologia para reduzir a complexidade dos particionamentos no VCC tanto para a predição intra quanto para a predição inter. Na predição intra, o operador Canny foi utilizado para detectar se a direção das bordas do bloco que está sendo predito é majoritariamente vertical ou horizontal. De acordo com a direção majoritária obtida, os particionamentos no sentido horizontal ou vertical são pulados. Já para a predição inter, é considerada a diferença entre os pixels do bloco que está sendo predito e os pixels do bloco co-localizado em dois frames de referência, um anterior e um posterior. A partir destas diferenças uma imagem binária é gerada, onde os zeros e os uns indicam que a diferença entre os pixels do bloco atual e seus dois blocos co-localizados foram menores ou maiores que um treshold, respectivamente. Sendo assim, sempre que a quantidade de zeros nesta imagem binária em relação a quantidade total de pixels presente na imagem é maior ou igual a 95% nenhum particionamento adicional é realizado no bloco. Caso contrário, a mesma metodologia empregada na predição intra é avaliada para possivelmente pular particionamentos no sentido horizontal ou vertical. Por fim, o método foi implementado no VTM 4.0 e 20 sequências de vídeo da CTC do VVC foram codificadas com as configurações AI e RA sob os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se médias de TS de 36,18% e 31,43% e médias de BD-BR de 0,71% e 1,34% para as configurações AI e RA, respectivamente.

Yang et al. (2019) propôs uma metodologia para reduzir a quantidade de particionamentos e modos intra que são avaliados no VVC. Descobrir qual dos seis possíveis particionamentos é mais adequado para uma CU foi tratado como um problema de classificação. Entretanto, este problema foi dividido em cinco classificações binárias. Primeiro, um modelo prediz se o bloco atual deve ou não utilizar o particionamento quadrático. Caso positivo, todos os particionamentos não quadrados são pulados. Caso negativo, os quatro modelos restantes, um para cada particionamento binário e ternário nos sentidos horizontal e vertical, predizem se um dos particionamentos podem ser pulados ou não. Para realizar tais classificações binárias foram utilizados cinco modelos baseados em Árvore de Decisão. As features selecionadas para trei-

namento dos modelos foram dividas em três categorias: informação de textura global, informação de textura local e informação de contexto. Com o intuito de medir a homogeneidade da CU atual, na categoria informação de textura global foram considerados o tamanho da CU e os gradientes obtidos a partir do operador Sobel, dentre estes a magnitude máxima de gradiente obtida e as médias dos gradientes na direção horizontal e vertical. Para medir a eficácia de um particionamento na horizontal ou na vertical, na categoria informação de textura local foram consideradas as diferenças entre as variâncias dos blocos que são gerados após o particionamento. Por fim, como existe uma correlação entre o melhor particionamento selecionado para uma CU e os melhores particionamentos selecionados para CUs vizinhas, na categoria informação de contexto foi levado em consideração a profundidade dos particionamentos quadrados e não quadrados das CUs vizinhas. Os modelos foram treinados de forma offline a partir de seis sequências de vídeo codificadas no VTM, entretanto, não é deixado claro pelos autores como o balanceamento dos datasets foi realizado ou qual profundidade máxima foi setada para as árvores de decisão. Para reduzir a quantidade de modos intra avaliados para cada CU, uma Busca baseada em Gradiente Descendente foi realizada para reduzir a quantidade de modos intra direcionais que são avaliados no processo do RMD. Como o resultado desta busca está altamente relacionado com o ponto inicial de busca e existe uma alta probabilidade do melhor modo intra estar dentre os MPMs, o ponto inicial de busca foi setado como o modo MPM com menor custo RMD. A partir deste ponto inicial, uma busca é feita a esquerda do modo e a direita do modo, indo de quatro em quatro modos. Após, uma busca com um passo de um modo é feita ao redor do modo com menor custo RMD encontrado após realizar a busca a esquerda e a direita. Por fim, apenas um modo intra com menor custo RMD é considerado para passar pelo processo RDO. Ambos os métodos foram implementados no VTM 2.0 e 26 sequências de vídeo das classes A1 à F da CTC do VVC foram codificadas com a configuração AI e com os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se médias de TS de 52,59%, 25,51% e 62,46% e médias de BD-BR de 1,56%, 0,54% e 1,93% ao ativar a técnica que reduz a complexidade dos particionamentos, ao ativar a técnica que reduz a complexidade dos modos de predição intra e ao ativar ambas as técnicas, respectivamente.

Zouidi et al. (2019) propôs uma forma de reduzir a quantidade de modos que são avaliados pelo RMD. Para tanto, foram levantadas estatísticas dos modos que costumam aparecer na lista final do RMD para cada uma das profundidades das partições binárias. A partir destas estatísticas, foram geradas listas de modos a serem avaliados pelo RMD para cada uma das profundidades possíveis das partições binárias. Estas listas levam em consideração o QP ao qual o vídeo está sendo codificado e os MPM são sempre considerados na avaliação do RMD. O método foi implementado no JEM 7.0 e seis sequências de vídeo das classes A1, A2, C e D da CTC do VVC foram

codificadas com a configuração AI e com os valores de QP 22 e 37. Como resultados, obteve-se uma média de TS de 5,09% e uma média de BD-BR de 2,67%.

No trabalho de **Chen et al. (2020)** o objetivo foi reduzir a complexidade da decisão de modo intra do VVC ao considerar apenas informações que já estão sendo calculadas dentro do próprio codificador. Esta redução de complexidade se dá em dois principais aspectos, sendo o primeiro deles diminuir a quantidade de modos que são avaliados pelo RDO e a segunda terminar o processo do RDO previamente. A redução da quantidade de modos que são testados no RDO foi feita de acordo com uma análise que mostrou que quando o modo com menor custo RMD é um dos MPM, existe uma chance de 95% de que o melhor modo para esta CU seja um dos MPM. Sendo assim, quando esta condição é atendida, apenas os modos da lista dos MPM são avaliados pelo RDO. Por outro lado, o término prévio do processo RDO foi desenvolvido com base em um modelo linear obtido através da correlação positiva que existe entre os custos RMD e os custos RDO. Através deste modelo, é possível mapear um custo de RMD em um custo de RDO, ao passo que verifica-se se a diferença entre este custo obtido e o custo do primeiro modo RMD está acima de um treshold. Caso positivo, considera-se que o melhor modo já deve ter sido encontrado e o processo RDO é terminado previamente. De acordo com testes apresentados no trabalho, mostrou-se que este modelo linear obtém o melhor modo igual ao do codificador sem nenhuma alteração em 90% das vezes. O método proposto foi implementado no VTM 2.0 e 26 sequências de vídeo distribuídas pelas classes A1 à F foram codificadas com a configuração AI e com os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se médias de TS de 15,65%, 27,69% e 30,59% e médias de BD-BR de 0,10%, 0,81% e 0,86% ao ativar a técnica que reduz a quantidade de modos pré-RDO, ao ativar a técnica que termina previamente o processo RDO e ao ativar ambas as técnicas, respectivamente.

No trabalho de **Saldanha et al. (2020b)** foram propostas duas estratégias para pular avaliações de partições binárias ou ternárias de acordo com informações de textura da CU, melhor modo intra e melhor modo ISP obtido para a CU. Para a primeira estratégia, foi verificado pelos autores que uma partição binária ou ternária na vertical costuma ser melhor que uma horizontal quando a variância vertical dos blocos que são gerados após a partição é menor em relação a variância horizontal destes mesmos blocos. O mesmo é verificado para a partição binária ou ternária no sentido horizontal. Além disso, o modo de predição intra que foi considerado o melhor para a CU atual também é levado em consideração. Para tanto, os modos de predição intra direcionais foram divididos em duas categorias, sendo que a primeira contém modos próximos a direção horizontal e a segunda contém modos próximos a direção vertical. Sendo assim, de acordo com as variâncias obtidas e a direção do modo intra, todas as partições binárias ou ternárias no sentido horizontal ou vertical são puladas. Para

a segunda estratégia, foi feita uma relação entre o melhor modo ISP obtido para a CU e os particionamentos no sentido horizontal ou vertical. Sendo assim, quando o melhor modo ISP obtido para a CU foi no sentido horizontal, os particionamentos no sentido vertical podem ser pulados, e vice-versa. Ambas as estratégias foram implementadas no VTM 5.0 e 26 sequências de vídeo das classes A1 à F da CTC do VVC foram codificados com a configuração AI sob os valores de QP 22, 27, 32 e 37. Como resultados, obteve-se médias de TS de 31,41% e de BD-BR de 0,92%.

Em **Zhang et al. (2020)** foi proposta uma metodologia para reduzir o número de particionamentos e o número de modos intra que são avaliados no VVC. Para reduzir a quantidade de particionamentos que são avaliados, a CU atual é classificada em três regiões possíveis, sendo estas região homogênea, região heterogênea e região comum. Esta classificação é feita ao comparar a complexidade de textura da CU atual com tresholds obtidos a partir das CUs vizinhas, sendo que a complexidade de textura é medida através do desvio padrão da complexidade de pixel relativo, método que já foi utilizado no trabalho descrito anteriormente (LIU et al., 2016). A partir desta classificação, existem três possibilidades: não realizar mais particionamentos caso a CU seja classificada em uma região homogênea; seguir o processo normal do VVC caso a CU seja classificada em uma região regular; e seguir para um classificador baseado em Random Forest caso a CU seja classificada em uma região heterogêna. Este classificador tem o objetivo de predizer qual particionamento deve ser realizado na CU de região heterogênea. Sendo assim, sua saída são uma das seis classes possíveis, dentre estas QT, BTH, BTV, TTH, TTV e NS. As features consideradas foram a entropia do bloco, que representa a quantidade de informação presente em uma imagem, o contraste, que representa a profundidade de textura da imagem e o momento da diferença inversa, que representa a magnitude das diferenças locais em um bloco. Estas três features são calculadas em quatro ângulos diferentes, sendo estes 0º, 45º, 90º e 180º, e foram obtidas ao codificar quatro sequências de vídeo com os QPs 22, 27, 32 e 37 no VTM. Após, os conjuntos de treino e teste foram gerados a partir do método chamado *Bootstrap Resampling*. O número de árvores de decisão presentes na Random Forest foi setado para 25 e a profundidade máxima de cada árvore de decisão foi setado para 30. Ao fim do treino e teste, o modelo obteve em média uma acurácia de 89,45%. Para a redução de modos intra a serem avaliados no VVC, existem duas etapas. A primeira etapa consiste em reduzir o número de modos que são avaliados no processo do RMD. Para tanto, os 35 modos que são avaliados na primeira etapa do RMD são divididos em quatro classes direcionais, sendo estas nos ângulos de 0º, 45º, 90º e 180º. Todas as quatro classes sempre incluem o modo DC e o modo Planar. A partir da direção de textura da CU atual obtida a partir do operador Canny, a CU é classificada em no máximo duas destas quatro classes. Sendo assim, o número de modos avaliados na primeira etapa no RMD é reduzido no mínimo pela

metade. A segunda etapa consiste em reduzir a quantidade de modos que são avaliados pelo RDO. Para tanto, a lista de modos a ser testada pelo RDO é ordenada do menor custo RMD para o maior custo RMD e todos os modos que possuem um custo RMD maior que um dos modos MPM são removidos da lista a ser avaliada pelo RDO. Uma exceção se dá para os modos MPMs, DC, Planar, Horizontal e Vertical, os quais nunca são removidos da lista. Por fim, o método foi implementado no VTM 4.0 e 18 sequências de vídeo das classes A1 à E das CTCs do VVC foram codificadas com a configuração AI e com os QPs 22, 27, 32 e 37. Como resultados, obteve-se médias de TS de 39,05%, 30,13% e 54,91% e médias de BD-BR de 0,69%, 0,58% e 0,93% ao ativar a técnica que reduz o número de particionamentos que são avaliados, a técnica que reduz o número de modos avaliados e ambas as técnicas, respectivamente.

4.3 Considerações Finais

Na Tabela 4 e na Tabela 5 são apresentados os trabalhos encontrados para o codificador HEVC e VVC, respectivamente. Para cada um dos trabalhos, é apresentado seu **Tipo**, média de **TS**, média de **BD-BR**, se o trabalho extraiu *features* da textura do bloco **(TF)**, se o trabalho extraiu *features* do processo de codificação **(CF)** e se o trabalho utilizou *Machine Learning* **(ML)**. Os trabalhos podem ser classificados em três **tipos**, sendo estes **Decisão de Modo Rápida (MR)**, o qual engloba trabalhos que reduzem a quantidade de modos de predição intra que são avaliados no processo do RMD ou do RDO, **Decisão de Particionamento Rápida (PR)**, o qual engloba os trabalhos que reduzem a quantidade de particionamentos que são avaliados e **Híbrido (H)**, o qual engloba trabalhos que pertencem aos dois tipos. Quando o trabalho é do tipo H, é apresentado o TS e o BD-BR ao considerar ambas as técnicas de redução de complexidade. Ainda que sejam poucos, alguns trabalhos realizaram testes com predição inter-quadro e predição intra-quadro. Nestes casos, o maior TS é apresentado.

De acordo com os dados apresentados na Tabela 4, os quais dizem respeito aos trabalhos desenvolvidos para o HEVC, é possível visualizar que a maioria dos trabalhos encontrados são do tipo MR ou H. Isto se deve ao fato de que o foco deste trabalho é em desenvolver um trabalho do tipo MR e de que a pesquisa foi realizada a partir de trabalhos que citaram (DA SILVA; AGOSTINI; SILVA CRUZ, 2012), o qual é um trabalho do tipo MR. Além disto, também é possível verificar que 16 e 14 dos 17 trabalhos extraíram *features* da textura do bloco e do processo de codificação (TF e CF), respectivamente, e apenas 5 dos 17 utilizaram ML. Dentre as *features* de textura consideradas nos trabalhos, as que mais ocorreram foram estimativas de gradiente, como Sobel ou Prewitt, ou ainda medidas estatísticas direcionais, como Variância ou Desvio Padrão. Estas medidas foram calculadas com o intuito de medir a homoge-

Tabela 4 – Trabalhos Relacionados encontrados para o Codificador HEVC

Trabalho	Tipo	TS	BD-BR	TF	CF	ML
(LIU et al., 2017)	PR	59,6%	1,26%	✓	✓	✓
(HUANG et al., 2020)	PR	36,28%	0,80%	✓	✓	✓
(DA SILVA; AGOSTINI; SILVA CRUZ, 2012)	MR	18,88%	0,90%	✓	✗	✗
(ZHANG; ZHAO; XU, 2012)	MR	20,00%	1,06%	✗	✓	✗
(FANG; CHANG; CHUNG, 2013)	MR	68,00%	0,65%	✓	✓	✗
(GWON; CHOI; YOUN, 2015)	MR	30,30%	0,90%	✓	✓	✓
(JAMALI; COULOMBE; CARON, 2015)	MR	35,60%	1,07%	✓	✓	✗
(YANG; PI, 2016)	MR	30,6%	0,80%	✓	✗	✗
(CHEN et al., 2014)	H	42,84%	0,61%	✓	✓	✗
(CRISTINA; MIHNEA; IONUT, 2016)	H	4,65%	0,05%	✓	✗	✗
(DUANMU; MA; WANG, 2016)	H	52,0%	2,40%	✓	✓	✓
(LIU et al., 2016)	H	56,00%	1,00%	✓	✓	✗
(WANG; XUE, 2016)	H	33,32%	0,81%	✓	✓	✗
(ZHANG et al., 2016)	H	59,9%	0,70%	✓	✓	✓
(BENHAJYOUSSEF; EZZEDINE; BOUALLÈGUE, 2017)	H	42,80%	1,10%	✓	✓	✗
(LU; YU; JIN, 2018)	H	55,20%	1,60%	✓	✓	✗
(SUN et al., 2019)	H	57,13%	0,65%	✓	✓	✗

neidade de um bloco ou extrair a direção da textura de um bloco, quando o objetivo era reduzir a quantidade de particionamentos ou reduzir a quantidade de modos de predição intra que são avaliados, respectivamente. Para as *features* extraídas do processo de codificação, estas também foram extraídas para tomadas de decisão ou com o objetivo de calcular *tresholds*, e consistiram em *features* baseadas nos RMD-Costs ou RD-Costs dos modos intra, profundidade da CU que está sendo codificada, QP, dentre outros. Quanto aos valores de TS e BD-BR, cabe destacar que o trabalho de Fang; Chang; Chung (2013) possui um alto valor de TS, entretanto este valor é em relação ao tempo do RMD, e não ao tempo total de codificação. Já o trabalho de Cristina; Mihnea; Ionut (2016) possui um TS baixo se tratando de um trabalho do tipo H, entretanto, seu BD-BR também é baixo, o que justifica já que a solução provavelmente está muito próxima ao processo normal do HEVC. Um ponto negativo dos trabalhos encontrados para o HEVC é que nenhum deles mencionou o *overhead* causado pela implementação de suas técnicas, já que a maioria delas extrai *features* da textura do bloco e algumas delas também utilizam modelos de ML.

Em relação aos dados dos trabalhos desenvolvidos para o VVC, os quais são

Tabela 5 – Trabalhos Relacionados encontrados para o Codificador VVC

Trabalho	Tipo	TS	BD-BR	TF	CF	ML
(AMESTOY et al., 2019)	PR	30,00%	0,57%	✓	✓	✓
(CHEN et al., 2019)	PR	53,17%	1,62%	✓	✗	✗
(FU et al., 2019)	PR	45,00%	1,01%	✗	✓	✓
(LEI et al., 2019)	PR	40,87%	0,85%	✗	✓	✗
(PARK; KANG, 2019)	PR	34,00%	1,02%	✗	✓	✗
(PENG et al., 2019)	PR	51,72%	2,53%	✓	✗	✗
(TANG et al., 2019)	PR	36,18%	0,71%	✓	✓	✗
(SALDANHA et al., 2020b)	PR	31,41%	0,92%	✓	✓	✗
(ZOUIDI et al., 2019)	MR	5,09%	2,67%	✗	✓	✗
(CHEN et al., 2020)	MR	30,59%	0,86%	✗	✓	✗
(YANG et al., 2019)	H	62,46%	1,93%	✓	✓	✓
(ZHANG et al., 2020)	H	54,91%	0,93%	✓	✓	✓

apresentados na Tabela 5, é possível visualizar que mais da metade dos trabalhos encontrados são do tipo PR, o que indica que ainda existem possibilidades a serem exploradas em trabalhos do tipo MR. Mais da metade dos trabalhos extraíram *features* de textura dos blocos, sendo que novamente estas *features* geralmente foram estimativas de gradiente ou estatísticas direcionais do bloco. Entretanto, como no VVC existem novos tipos de particionamento e estes podem ser nos sentidos horizontal ou vertical, a diferença entre o HEVC e o VVC se dá na forma como estas *features* são utilizadas, já que agora para reduzir a quantidade de particionamentos não basta verificar somente a homogeneidade de um bloco, mas também é necessário verificar a direção de textura do bloco. O mesmo comportamento pode ser analisado para as *features* que foram extraídas do processo de codificação, onde mais da metade dos trabalhos utilizaram este tipo de informação para tomada de decisões, seja através de modelos de aprendizado de máquina ou não. O número de trabalhos que utilizou ML no VVC cresceu em relação aos trabalhos desenvolvidos para o HEVC, entretanto, este número ainda é baixo em relação ao total, sendo menos da metade. Dentre os modelos de ML utilizados nos trabalhos do VVC, apareceram modelos baseados em árvores ou bayeseanos, o que é justificável já que a proposta é reduzir complexidade e portanto os modelos considerados devem gerar pouco *overhead*. Outro ponto a ser considerado é que os modelos de ML desenvolvidos nestes trabalhos foram aplicados somente na redução de particionamentos que são avaliados, e não na redução de modos de predição intra que são avaliados. Cabe destacar que o trabalho de Zouidi et al. (2019), o qual buscou reduzir a complexidade somente no processo RMD, possui um baixo valor de TS para um alto valor de BD-BR, o que possivelmente indica que buscar reduzir complexidade somente no processo RMD do VVC não seja uma boa alternativa. Novamente, grande parte dos trabalhos não apresentou o *overhead*

Tabela 6 – Médias de TS e BD-BR para os trabalhos desenvolvidos no HEVC

Tipo	TS	BD-BR
PR	39,33%	0,63%
MR	27,93%	0,86%
H	44,20%	0,99%

Tabela 7 – Médias de TS e BD-BR para os trabalhos desenvolvidos no VVC

Tipo	TS	BD-BR
PR	41,40%	1,15%
MR	28,74%	0,66%
H	58,69%	1,43%

introduzido por suas técnicas, onde somente um apresentou.

Finalmente, na Tabela 6 e na Tabela 7 são apresentadas as médias de TS e BD-BR para os trabalhos, de acordo com o seu tipo, desenvolvidos para o HEVC e para o VVC, respectivamente. No cálculo das médias dos trabalhos do tipo MR ou PR, foram considerados os valores obtidos em trabalhos do tipo H quando estes foram apresentados de forma separada. Também, os trabalhos do tipo MR que reduziram complexidade somente no processo RMD foram deixados de fora. O que é possível observar para ambos codificadores é que trabalhos do tipo MR possuem uma média de TS menor em relação a trabalhos do tipo PR, o que é esperado já que o processo de decisão de particionamento engloba o processo de decisão de modo, pois cada bloco precisa realizar um processo de decisão de modo. Ao comparar as médias obtidas para os codificadores HEVC e VVC para os trabalhos do tipo PR e H, observa-se que existe uma redução de complexidade maior no VVC. Isto deve-se a dois fatores, sendo o primeiro a estrutura de particionamentos do VVC, a qual é mais complexa que o HEVC, e o segundo o aumento de modos de predição intra no VVC em relação ao HEVC. Também é possível notar que as soluções do tipo H trazem mais ganhos em TS, o que é esperado já que este tipo de trabalho engloba soluções do tipo PR e MR ao mesmo tempo.

5 ANÁLISE DA PREDIÇÃO INTRA-QUADRO NO VVC

Como visto no Capítulo 2, o VVC traz diversas ferramentas novas na predição intra-quadro, dentre estas os novos modos de predição intra, as novas transformadas e os novos tipos de particionamento de blocos. Todas estas novas ferramentas contribuem para uma maior eficiência de codificação, ao passo que também geram um aumento na complexidade do VVC por tornar o processo de decisão de modo mais custoso.

Com o objetivo de encontrar maneiras possíveis de reduzir a complexidade do processo de decisão de modo com pouco impacto na eficiência de codificação, neste capítulo é apresentado um estudo que foi feito acerca da probabilidade de que para cada tamanho de bloco um determinado tipo de modos de predição intra seja escolhido como o melhor ao final do processo de decisão de modo do VVC. Os passos que foram seguidos para realizar este estudo são apresentados a seguir.

5.1 Metodologia de Análise

Para realizar este estudo, inicialmente os modos de predição intra presentes no VVC foram divididos em quatro tipos, os quais são listados abaixo:

- **Não Angulares:** este tipo inclui os modos de predição intra **DC** e **Planar**, os quais não estão associados a uma direção específica. O modo **DC** consiste na média dos *pixels* vizinhos e é adequado para blocos homogêneos, enquanto que o modo **Planar** consiste em uma interpolação dos *pixels* vizinhos e é adequado para blocos com diferentes direções de textura;

- **Angulares:** este tipo inclui os 65 modos de predição intra angulares presentes no VVC. Cada um destes 65 modos de predição intra são associados a um ângulo distinto, como pode ser visto na Figura 4, e quando aplicados a um bloco copiam os *pixels* vizinhos no ângulo determinado pelo modo;

- **ISP:** neste tipo, é incluso o modo de predição intra **ISP**, nova ferramenta presente no VVC que foi apresentada no Capítulo 2. Cabe destacar que um modo de predição intra ISP ainda pode estar associado aos tipos **Não Angulares** ou

Angulares, visto que um modo de predição intra ISP irá predizer as subpartições que são geradas com o modo de predição intra DC, Planar ou um dos 65 modos angulares;

- **MIP:** por fim, neste tipo é incluso o modo de predição intra **MIP**, nova ferramenta de predição intra que foi introduzida ao VVC e que também foi apresentada no Capítulo 2.

Após dividir todos os modos de predição intra do VVC em quatro principais tipos, obteve-se um conjunto de 73 sequências de vídeo, das quais 17 são da resolução HD (720p), 32 são da resolução Full HD (1080p) e 24 são da resolução Ultra HD (4K). Estas sequências de vídeo foram obtidas através de bases de dados de três grupos de pesquisa, sendo estes o *Internet Video Codec* (NETVC) (DAEDE; NORKIN; BRAILOVKKIY, 2018), um grupo de pesquisa voltado a codificadores de código aberto; o *Ultra Video Group* (UVG) (MERCAT; VIITANEN; VANNE, 2020), grupo de pesquisa em sistemas de processamento de vídeo e imagem da Finlândia; e o *Joint Video Experts Team* (JVET) (BOSSEN et al., 2019a), grupo de pesquisa responsável pelo desenvolvimento de codificadores como o HEVC e o próprio VVC. Este conjunto foi obtido de forma a não conter nenhum vídeo utilizado nos testes de desempenho do VTM de acordo com as CTCs (Condições Comuns de Teste) do mesmo, o que evita que as inferências aqui realizadas algum nível viés com os resultados finais da dissertação.

Como é inviável codificar 73 sequências de vídeo por conta do tempo demandado para tal, decidiu-se utilizar um sub-conjunto destes vídeos que foi selecionado a partir da métrica *Spatial Information-Temporal Information* (SITI) (ITU-T, 2008), a qual faz uma relação das sequências de vídeo em dois eixos. O primeiro eixo é o *Spatial Information* (SI), que indica a variabilidade de dados em um mesmo quadro, e o segundo é o *Temporal Information* (TI), que indica a variabilidade de dados ao longo dos quadros. A partir desta métrica, foi possível dividir as sequências de vídeo em quatro quadrantes de acordo com os seus valores de SI e TI, e então selecionar cinco sequências de cada uma das três resoluções ao escolher as quatro mais distantes da mediana em cada quadrante e uma próxima ao centro. O resultado final deste processo é apresentado na Figura 11, onde as sequências circuladas foram as escolhidas.

Cada uma das 15 sequências de vídeo distribuídas igualmente entre as resoluções HD, Full HD e Ultra HD passaram pelo processo de codificação do VTM 9.0 com a configuração AI e com os valores de QP 22, 27, 32 e 37. Para cada bloco que foi predito durante o processo de codificação, foi extraído seu tamanho, posição e melhor modo de predição intra de acordo com a decisão final do processo de decisão de modo. O número de quadros foi limitado a 120, 60 e 30 para as resoluções HD, Full HD e Ultra HD, respectivamente. Foram escolhidos estes números pois dentro de cada conjunto de resoluções existem sequências de vídeo que possuem no máximo este

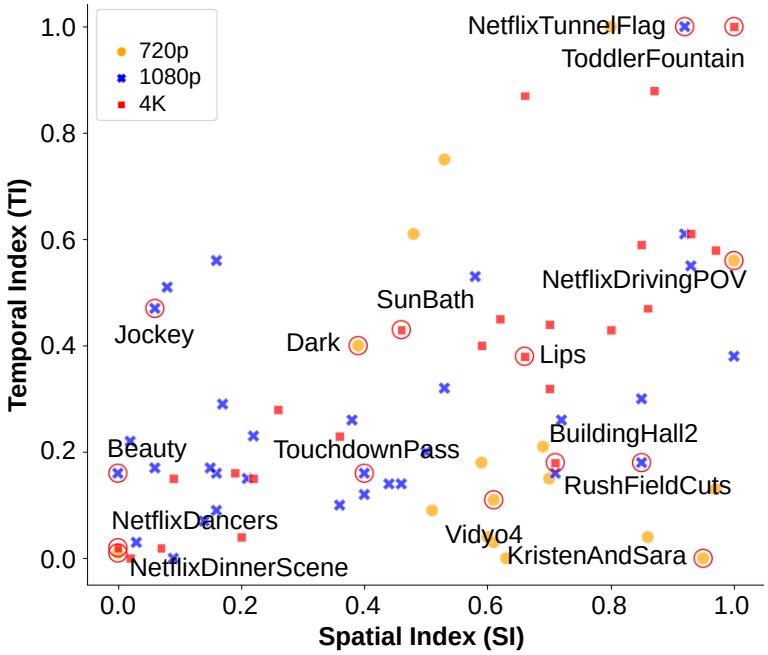


Figura 11 – Sequências de Vídeo distribuídas de acordo com a Métrica SIxTI

número de quadros. Ao final das codificações, os resultados do estudo foram obtidos e são apresentados a seguir.

5.2 Resultados da Análise

A Tabela 8 apresenta a quantidade de blocos que foram extraídos do processo de decisão de modo para realizar a análise. Esta quantidade é apresentada por tamanho de bloco e por QP. Os números estão de acordo com o esperado, visto que conforme o valor de QP aumenta a quantidade de blocos extraídos tende a diminuir, isto porque com valores de QP maiores o codificador prioriza a compressão ao invés da qualidade de imagem e, portanto, o processo de decisão de modo avalia menos blocos e converge a uma decisão mais rapidamente. Também é possível notar que existe uma relação crescente do número de blocos extraídos conforme o tamanho do bloco diminui, e isto também é esperado já que um tamanho de bloco maior ocupa uma maior região de cada *frame*, o que resulta em uma quantidade menor de blocos extraídos para tamanhos de bloco maiores. Dada a forma como os vídeos codificados foram selecionados e a alta quantidade de dados extraídos, com aproximadamente um bilhão ao somar as quantidades de todos os tamanhos de blocos e QPs, pode-se concluir que a análise foi realizada sobre um conjunto de dados diverso e amplo.

Na Tabela 9 é apresentado proporcionalmente a quantidade de vezes que um tipo de modo é escolhido como o melhor para cada tamanho de bloco. Nestes números, os QPs foram agrupados. De acordo com os dados presentes na Tabela 9, é possível visualizar que o modo de predição intra **Planar** detém em média a maior probabilidade de ser escolhido como o melhor. Ao considerar que o **Planar** e o **DC** pertencem ao

Tabela 8 – Quantidade de blocos extraídos do processo de decisão de modo por tamanho de bloco e QP

	22	27	32	37	Total
4x4	67.756.263	39.722.899	23.219.679	10.239.511	140.938.352
4x8	71.627.351	46.174.840	27.734.801	13.938.572	159.475.564
4x16	43.479.046	32.153.273	21.384.639	12.674.079	109.691.037
4x32	14.869.679	13.080.161	10.443.951	5.971.303	44.365.094
8x4	71.360.270	46.028.898	27.887.511	14.058.124	159.334.803
8x8	66.832.959	48.810.988	32.760.709	19.630.542	168.035.198
8x16	40.427.034	33.682.194	25.666.131	15.786.629	115.561.988
8x32	10.526.409	10.414.224	8.625.509	5.757.075	35.323.217
16x4	42.751.171	31.960.964	21.515.597	12.731.102	108.958.834
16x8	40.369.137	33.801.122	25.921.049	15.953.758	116.045.066
16x16	17.188.955	16.032.289	13.362.556	9.185.924	55.769.724
16x32	5.355.406	5.673.562	5.355.478	4.391.569	20.776.015
32x4	14.851.890	12.910.801	10.360.354	6.030.091	44.153.136
32x8	10.651.511	10.429.029	8.870.688	5.933.373	35.884.601
32x16	5.372.139	5.698.516	5.443.567	4.486.576	21.000.798
32x32	2.336.295	2.345.101	2.328.395	2.283.112	9.292.903
64x64	566.066	577.390	579.631	580.530	2.303.617
Total	**526.321.581**	**389.496.251**	**271.460.245**	**159.631.870**	**1.346.909.947**

tipo **Não Angulares**, conforme foi definido no início deste capítulo, chega-se a uma média de 41,29%. Sabendo que o modo de predição intra ISP consiste em gerar subpartições na horizontal ou na vertical do bloco atual e para cada subpartição aplicar o mesmo modo de predição intra, pode-se somar a probabilidade de ocorrência dos modos de predição intra **ISP (Planar-DC)** ao tipo **Não Angulares**. Ao realizar isso, chega-se a uma média de 49,58%, quase metade.

Existem exemplos específicos onde a probabilidade de um dos modos de predição intra Angulares serem escolhidos como o melhor é maior em relação ao Planar, como é o caso do tamanho de bloco 4x32. Entretanto, cabe destacar que esta diferença é pouca e que o modo de predição intra Planar detém esta alta probabilidade sozinho, enquanto que para os angulares esta probabilidade é dividida entre os 65 modos de predição intra angulares. É interessante notar também que para o tamanho de bloco 64x64 a probabilidade do modo de predição intra ISP (Planar-DC) cresce significativamente, chegando a 36,27%. Isto provavelmente ocorre por se tratar de um tamanho de bloco maior, onde as amostras de referência utilizadas possuem maior distância das amostras que estão sendo preditas, e portanto o ISP gera predições melhores já que para cada subpartição que é predita as amostras dessa subpartição são utilizadas como amostras de referência para a subpartição vizinha.

Por fim, o que se pode concluir desta análise é que os modos de predição intra do tipo **Não Angulares** (**Planar**, **DC**, **ISP Planar-DC**) possuem em média maior probabilidade de serem escolhidos como o melhor em relação aos outros tipos. Isto justifica a

Tabela 9 – Proporção de vezes que um determinado Tipo de Modo é escolhido como o melhor para um Tamanho de Bloco

	Planar	**DC**	**Angular**	**ISP (Planar-DC)**	**ISP (Angular)**	**MIP**
4x4	50,00%	7,60%	32,04%	0,00%	0,00%	10,36%
4x8	37,19%	7,56%	28,43%	3,86%	2,69%	20,27%
4x16	37,73%	9,83%	34,08%	4,74%	3,11%	10,52%
4x32	32,35%	10,33%	37,29%	7,10%	5,43%	7,50%
8x4	38,52%	7,58%	28,33%	4,11%	2,83%	18,64%
8x8	39,26%	7,51%	29,64%	2,16%	1,82%	19,61%
8x16	33,19%	7,13%	28,37%	4,98%	3,56%	22,78%
8x32	31,52%	9,02%	32,8%	7,89%	5,04%	13,72%
16x4	38,08%	9,55%	32,93%	5,97%	3,63%	9,84%
16x8	33,98%	7,48%	28,09%	5,60%	3,58%	21,27%
16x16	30,52%	6,97%	28,26%	5,45%	3,99%	24,81%
16x32	29,77%	7,52%	26,59%	7,58%	4,56%	23,98%
32x4	31,84%	10,09%	35,71%	9,16%	6,30%	6,91%
32x8	31,01%	9,48%	31,50%	10,03%	5,79%	12,19%
32x16	28,07%	7,07%	25,00%	10,67%	5,87%	23,32%
32x32	25,44%	6,27%	20,14%	15,43%	7,24%	25,49%
64x64	17,13%	5,42%	11,62%	36,27%	10,36%	19,20%
Média	**33,27%**	**8,02%**	**28,87%**	**8,29%**	**4,46%**	**17,08%**

maioria dos trabalhos relacionados encontrados nunca excluírem o Planar ou o DC da lista a ser avaliada pelo processo de decisão de modo e também justifica o próprio algoritmo do VTM sempre incluir o modo de predição intra Planar na lista a ser avaliada, mesmo que ele fique de fora de acordo com os custos RMD obtidos e com o número de modos que devem estar na lista. Sabendo que existe esta maior probabilidade de ocorrência para modos de predição intra do tipo Não Angulares e que os outros tipos de modos de predição intra costumam estar na lista a ser avaliada, existe a possibilidade de reduzir o tempo do processo de decisão de modo ao detectar quando estes outros tipos de modos de predição intra não tão prováveis de serem escolhidos como o melhor podem ser descartados da avaliação.

6 SOLUÇÃO PARA REDUÇÃO DE COMPLEXIDADE DA DECISÃO DE MODO INTRA NO VVC

Com base na análise realizada no Capítulo 5, concluiu-se que existe a possibilidade de reduzir o tempo do modo de decisão do VVC ao detectar tipos de modos de predição intra que podem ser descartados da avaliação. Além disso, também concluiu-se que os modos de predição intra Não Angulares, sendo estes o Planar, o DC e os ISP DC-Planar possuem em média uma maior chance de serem escolhidos como o melhor para os diversos tamanhos de bloco existentes no VVC. Logo, o modelo de decisão rápido utilizado como base para o desenvolvimento deste trabalho é apresentado na Figura 12. Este modo de decisão rápido possui duas ideias principais:

- **Ordenar a RD-List** de acordo com os tipos de modos de predição intra e suas chances de serem escolhidos como o melhor modo. Desta forma, os modos de predição intra Não Angulares, sendo estes Planar e DC, sempre serão os primeiros a serem avaliados. Em seguida, são avaliados os modos de predição intra Angulares e por fim os modos de predição intra MIP. Os modos de predição intra ISP continuam sendo os últimos a serem avaliados. A ideia em ordenar a RD-List de acordo com os modos que ocorrem com mais frequência é possibilitar que decisões antecipadas possam ser realizadas sobre os modos que ainda serão avaliados, ou seja, decidir se o custo taxa-distorção de um modo com alta probabilidade de ocorrência já indica que não é necessário realizar as avaliações de outros modos;
- **Utilizar modelos de aprendizado de máquina supervisionado baseados em árvores de decisão no contexto da tomada de decisão do modo de decisão rápido**. A principal razão pela qual decidiu-se trabalhar com árvores de decisão foi o seu fácil entendimento e treinamento, bem como o uso deste tipo de modelo em trabalhos relacionados e o fato dele inserir pouca complexidade na solução final. Para treinar estes modelos, decidiu-se utilizar os custos calculados pelo próprio codificador. Ao utilizar *features* extraídas do próprio processo de codificação evita-se que exista um maior *overhead* no método proposto. Além disso,

como os modos de predição intra são agora ordenados pelo tipo, é possível que ao final da avaliação de cada tipo de modo de predição intra os RD-Costs dos modos que já foram avaliados sirvam como *features* para os modelos.

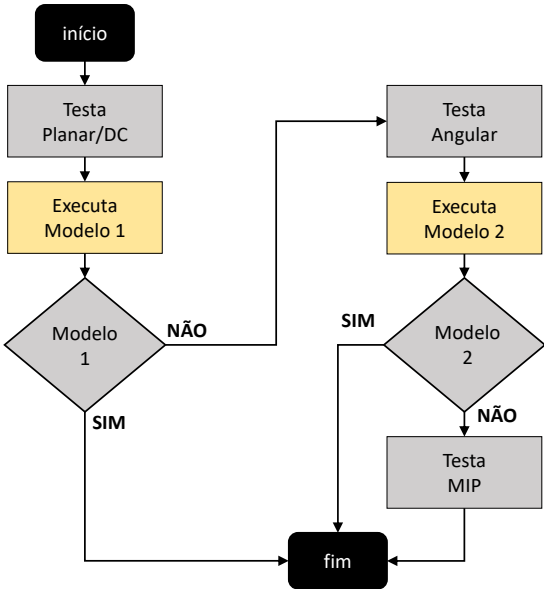


Figura 12 – Ilustração do Modo de Decisão Rápido Proposto

Na decisão de modo rápida proposta na Figura 12 é possível visualizar que os modos Planar e DC são os primeiros a serem avaliados. Ao término desta avaliação, com base nos custos taxa-distorção obtidos para estes modos, o Modelo 1 decide se os modos Planar e DC já são suficientes, ou seja, se já geraram um custo taxa-distorção adequado de acordo com as *features* disponíveis. Se o Modelo 1 decidir que os modos **Planar e DC** são suficientes, as demais avaliações dos modos de predição intra podem ser ignoradas, caso contrário, a avaliação dos modos de predição intra deve seguir adiante para os modos **Angulares**. Caso o Modelo 1 decida que as avaliações dos demais modos de predição intra deva seguir, os modos angulares são os próximos a serem avaliados. Ao término desta avaliação, com base nos custos taxa-distorção dos modos **Planar e DC** e dos modos **Angulares**, o Modelo 2 decide se estes modos já são suficientes e as demais avaliações de modos de predição intra podem ser ignoradas ou se estes modos ainda não são suficientes e os modos **MIPs** devem ser avaliados.

Existem algumas considerações a serem feitas acerca do modo de decisão rápido proposto. Primeiro, as predições dos modelos e possíveis não avaliações de modos de predição intra pelo processo do RDO são realizadas somente na primeira transformada. Ainda assim, existe o impacto nas outras transformadas, isto porque o VVC recupera a RD-List entre as transformadas e portanto ao desconsiderar um modo de predição intra da avaliação de uma transformada também se desconsidera das outras. Segundo, não há um modelo específico para tomada de decisão a respeito dos mo-

dos de predição intra ISP. Entretanto, como no processo do RDO no VVC os modos de predição intra ISP são avaliados de acordo com os resultados obtidos para os modos de predição intra Planar, DC e Angulares, caso o Modelo 1 chegue a uma tomada de decisão onde a ação seja ignorar os modos de predição intra Angulares e MIPS, isto implica que os modos de predição intra ISP Angulares também não serão avaliados. Por fim, até o momento não existe nenhum trabalho que tenha proposto um modo de decisão rápido para a predição intra do VVC que busca realizar a tomada de decisão de forma hierárquica agrupando e ordenando os modos de predição intra em classes. Portanto, o método aqui proposto é inovador neste sentido.

Para desenvolver o modo de decisão rápido proposto, o principal desafio consiste em obter os modelos de aprendizado de máquina supervisionado responsáveis pela tomada de decisão. Esta não é uma tarefa trivial, visto que os modelos devem possuir boa F1-score para que não se tenha uma perda significativa de eficiência de codificação, ao mesmo tempo em que devem ser simples e rápidos para que o método não insira um *overhead* alto, o que resultaria em um aumento de complexidade. Neste trabalho, todos os testes e desenvolvimentos dos modelos baseados em árvores de decisão foram realizados com a biblioteca Scikit-Learn (PEDREGOSA et al., 2011). Nas seções a seguir, serão apresentados e descritos os passos seguidos na metodologia para obter os modelos de aprendizado de máquina supervisionado.

6.1 Obtenção de Dados

Na primeira etapa, o objetivo foi obter os dados que são utilizados para treinar, testar e validar os modelos de aprendizado de máquina supervisionado utilizados no método apresentado na Figura 12. Para tanto, os mesmos 15 vídeos que foram selecionados no Capítulo 5 de acordo com suas métricas SI e TI foram codificados no VTM 9.0 com a configuração AI (*All Intra*) e com os valores de QP 22, 27, 32 e 37. No processo de codificação foram extraídas as *features* apresentadas na Tabela 10 para cada bloco que passou por um processo de predição.

Na Tabela 10, **RMDC** e **RDC** referem-se a um **custo taxa-distorção obtido no processo do RMD e do RDO**, respectivamente, e na coluna tipo o valor **Num.** refere-se a uma *feature* do tipo numérica e o valor **Cat.** refere-se a uma *feature* do tipo categórica. Pode ser visualizado que foram consideradas como *features* principalmente os custos que são calculados no processo do RMD e no processo do RDO para cada tipo de modo de predição intra. No caso dos modos de predição intra DC, Angulares e MIP apenas o menor custo é considerado. Ainda que o DC seja apenas um modo, existem diferentes custos pois ele pode ser combinado com diferentes amostras de referência através da técnica MRL apresentada no Capítulo 2. Além disso, também extraiu-se a posição na lista a ser avaliada pelo processo do RDO (RD-List) da primeira ocorrên-

Tabela 10 – Features Extraídas do Processo de Decisão de Modo

Feature	Descrição	Tipo
RMDC Planar	RMDC do modo Planar	Num.
RMDC DC	Menor RMDC do modo DC	Num.
RMDC Angular	Menor RMDC dos modos Angulares	Num.
RMDC MIP	Menor RMDC dos modos MIPs	Num.
Posição Planar	Posição do modo Planar na RD-List	Num.
Posição DC	Posição do primeiro modo DC na RD-List	Num.
Posição Angular	Posição do primeiro modo Angular na RD-List	Num.
Posição MIP	Posição do primeiro modo MIP na RD-List	Num.
Modo Angular	Número do primeiro modo Angular na RD-List	Num.
Modo MIP	Número do primeiro modo MIP na RD-List	Num.
RDC Planar	RDC do modo Planar p/ 1ª transformada	Num.
RDC DC	Menor RDC do modo DC p/ 1ª transformada	Num.
RDC Angular	Menor RDC dos modos Angulares p/ 1ª transformada	Num.
RDC Planar/RDC DC	Razão entre os RDCs Planar e DC	Num.
RDC Planar/RDC Angular	Razão entre os RDCs Planar e Angular	Num.
RDC DC/RDC Angular	Razão entre os RDCs DC e Angular	Num.
Best Mode	Modo escolhido ao final	Cat.

cia de cada tipo de modo de predição intra. Nos casos dos modos de predição intra Angulares e MIPs, ainda são extraídos os números do primeiro modo Angular e do primeiro modo MIP na RD-List. Por fim, também extraíram-se as razões entre os custos taxa-distorção obtidos no processo do RDO para os modos intra Planar, DC e Angular. Estas razões foram extraídas já que a partir delas é possível saber o quão grande é a diferença entre o custo taxa-distorção para cada tipo de modo intra. Por fim, também extraiu-se o modo de predição intra escolhido ao final do processo do modo de decisão (*Best Mode*) para ser possível separar os dados em classes posteriormente.

Como dito anteriormente, decidiu-se trabalhar com estas *features* principalmente por não existir nenhum *overhead* no seu cálculo, já que elas são geradas dentro do próprio processo de codificação. Além disso, diversos trabalhos relacionados apresentados no Capítulo 4 também utilizaram os custos gerados dentro do processo de codificação para tomadas de decisão, o que implica que estas informações são úteis para o problema que está sendo atacado.

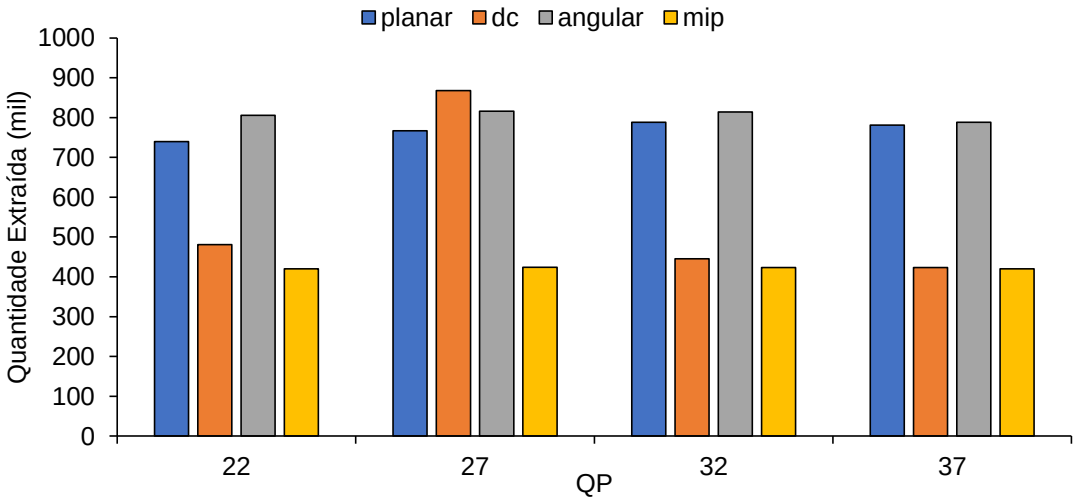
Novamente, limitou-se a quantidade de quadros para 120, 60 e 30 para as resoluções HD, Full HD e Ultra HD. Entretanto, dada a alta quantidade de *features* que são extraídas e o armazenamento que seria necessário caso todo bloco que passa pelo processo de predição fosse salvo, decidiu-se extrair cerca de 100.000 exemplos para cada combinação de modo intra e tamanho de bloco. A Figura 13 traz a quantidade de exemplos que foram extraídos para cada modo intra por QP 13(a), tamanho de bloco 13(b) e sequência de vídeo 13(c). As quantidades de Planar, DC e Angular que foram extraídas consideram os modos ISP, por esta razão possuem valores maiores em

relação ao modo MIP. Como o tamanho de bloco 4x4 não possui modos ISP, a quantidade de exemplos extraído para este tamanho de bloco é menor. Ainda que não tenha sido possível realizar a extração de exemplos de forma balanceada, foram extraídos exemplos o suficiente para que na etapa de subamostragem destes exemplos para formar os *datasets* necessários para treinar os modelos os *datasets* tenham exemplos balanceados por QP, vídeo, tamanho de bloco e classe.

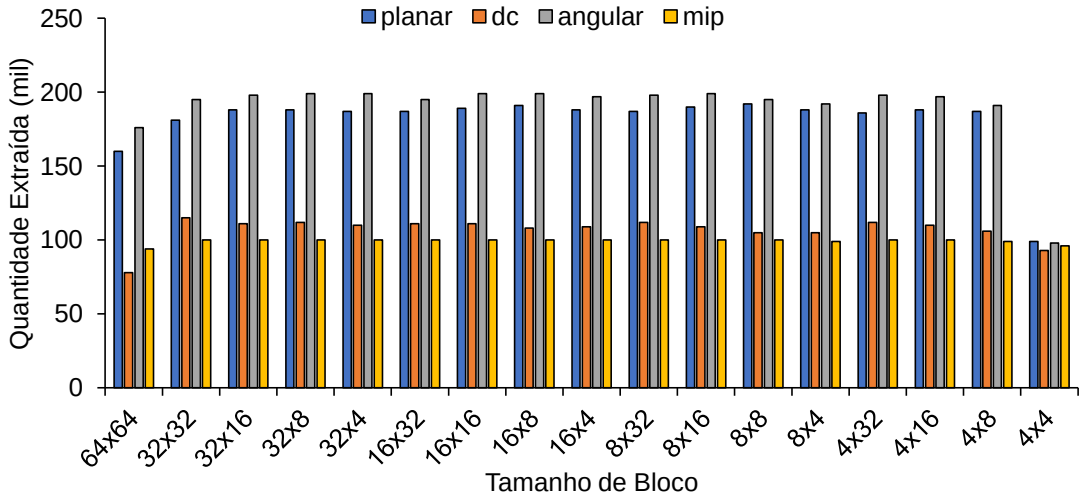
6.2 Produção dos Datasets

Ao obter os dados extraídos do processo de codificação do VTM 9.0 na primeira etapa, torna-se necessário gerar os *datasets* a partir destes dados, os quais são utilizados para treinar, testar e validar os modelos. De acordo com a Figura 12, é possível verificar que existem dois *datasets*, um para cada modelo, e que as tomadas de decisões são binárias. Desta forma, para o Modelo 1 a primeira classe contém exemplos onde o modo escolhido é Planar ou DC e a segunda classe contém exemplos onde o modo escolhido é Angular, MIP ou ISP Angular. Já para o Modelo 2, a primeira classe contém exemplos onde o modo escolhido é Angular e a segunda classe contém exemplos onde o modo escolhido é MIP. Entretanto, é importante levar em consideração que existem diferentes formas de agrupamento possíveis dos exemplos por tamanho de bloco. Por exemplo, os Modelos 1 e 2 apresentados na Figura 12 podem ser cada um apenas um modelo que é treinado com exemplos de todos os tamanhos de bloco ou podem ser divididos cada um em 17 modelos, um para cada tamanho de bloco. Sendo assim, nesta etapa além de gerar os *datasets* necessários para treinar, testar e validar os modelos propostos na Figura 12, também buscou-se verificar se é melhor ter apenas um modelo para todos os tamanhos de bloco ou se é melhor ter um modelo para cada tamanho de bloco. Como o modelo baseado em árvore de decisão da biblioteca Scikit-Learn (PEDREGOSA et al., 2011) não impõe nenhum limite na profundidade das árvores que são geradas após o treinamento, para evitar possíveis *overfittings* e para manter todos os modelos em um mesmo nível de complexidade decidiu-se limitar a profundidade de todas os modelos que foram gerados nesta etapa a no máximo 10.

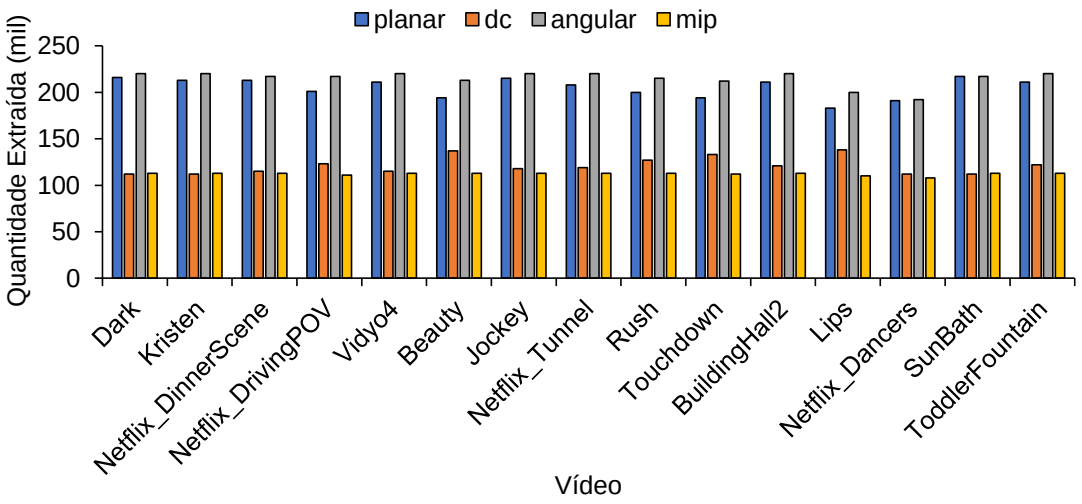
A Tabela 11 apresenta as F1-scores obtidas ao termos um *dataset* para todos os tamanhos de bloco e um *dataset* para cada tamanho de bloco, sendo que para esta segunda F1-score é apresentada a média das F1-scores obtidas para cada tamanho de bloco. Para obter estas F1-scores, foram gerados *datasets* para cada um dos modelos a partir dos dados apresentados na Figura 13. Estes *datasets* possuem exemplos balanceados por classe, QP, sequência de vídeo e tamanho de bloco. Como as classes dos *datasets* podem possuir mais de um tipo de modo de predição intra, também considerou-se o balanceamento de cada tipo de modo de predição intra den-



(a) Quantidade de dados extraídos para cada QP



(b) Quantidade de dados extraídos para cada tamanho de bloco



(c) Quantidade de dados extraídos para cada vídeo

Figura 13 – Quantidade de dados extraídos para cada QP, Tamanho de Bloco e Vídeo por modo intra

Tabela 11 – F1-scores ao ter um modelo para todos e para cada tamanho(s) de bloco

			Um Modelo p/ todos Tamanhos de Bloco		Um Modelo p/ cada Tamanho de Bloco	
Modelo	**Classe SIM**	**Classe NÃO**	**Fscore CV=5**	**Fscore Teste**	**Fscore CV=5**	**Fscore Teste**
1	Planar DC	Angular ISP Ang. MIP	81,18	81,27	81,25	81,34
2	Angular	MIP	84,48	84,41	84,98	85,03

tro de cada classe. Os *datasets* que contém exemplos de todos os tamanhos de bloco possuem cerca de 800.000 exemplos e os *datasets* que contém apenas um tamanho de bloco possuem cerca de 200.000 exemplos. Além disso, no caso dos *datasets* que contém exemplos de todos os tamanhos de bloco os RMD-Costs e RD-Costs foram normalizados ao dividir seu valor pelo número de *pixels* de acordo com o tamanho do bloco. Com os *datasets* prontos, cada um destes foi dividido em 75% para a fase de desenvolvimento 25% para a fase de teste. Então os 75% foram utilizados para treinar e testar um modelo baseado em árvore de decisão com profundidade máxima igual a 10. Este treino e teste foi feito através de um Cross-Validation (CV) igual a cinco. Por fim, um modelo baseado em árvore de decisão com profundidade máxima igual a 10 é treinado com os 75% e os 25% restantes são utilizados para obter a F1-score na fase de teste.

De acordo com as F1-scores apresentadas na Tabela 11, é possível perceber que ter um *dataset* para cada tamanho de bloco não gerou melhoras significativas nesta métrica em relação a ter um *dataset* para todos os tamanhos de bloco. Desta forma, ter apenas um *dataset* que contém exemplos de todos tamanhos de bloco é mais vantajoso não só porque a diferença entre as F1-scores não é significativa, mas também porque o número de modelos cai de 17 para um, o que torna o processo de treinar, testar e validar os modelos mais simples, principalmente na próxima etapa da metodologia, a qual consiste em realizar buscas para cada modelo em alguns dos hiperparâmetros da árvore de decisão. Sendo assim, decidiu-se adotar modelos que realizam predições sobre todos os tamanhos de bloco.

A Tabela 12 traz a acurácia para os dois modelos que realizam predições sobre todos os tamanhos de bloco. Para cada modelo, são apresentadas as proporções de acertos e erros para cada classe e para cada tipo de modo de predição intra dentro de cada classe. Estas proporções foram obtidas ao cruzar as classes preditas pelos modelos na fase de teste com as classes e tipos reais dos exemplos que foram preditos. O que pode ser visto através destas proporções é que para o Modelo 1, dentro da classe **SIM**, existe uma alta taxa de acerto para o modo de predição intra DC enquanto a taxa de acerto para o modo de predição intra Planar é mais baixa. Como o modo de

Tabela 12 – Taxa de Acertos por Classe e por Tipo de Modo de Predição Intra para cada Modelo

Modelo	Classe	Tipo	Acertos (%)	Erros (%)	Média Acertos (%) Classe	Média Erros (%) Classe
1	SIM	Planar	65,02	34,98	81,18	18,82
		DC	97,33	2,67		
	NÃO	Angular	86,21	13,79	81,63	18,37
		ISP Ang.	88,48	11,52		
		MIP	70,19	29,81		
2	SIM	Angular	82,20	17,80	82,20	17,80
	NÃO	MIP	86,63	13,37	86,63	13,37

predição intra DC costuma trazer bons resultados somente para blocos com textura homogênea, acredita-se que esta alta taxa de acertos se deve ao fato de ser mais fácil diferenciar este modo de predição intra dos modos de predição intra Angulares, ISP Angulares e MIPs através dos RMD-Costs e dos RD-Costs, já que para estes três últimos modos de predição intra os blocos costumam possuir texturas heterogêneas, o que resulta em RMD-Costs e RD-Costs para o modo DC não satisfatórios.

Por outro lado, o modo de predição intra Planar possui uma alta probabilidade de ser escolhido como o melhor, como já foi demonstrado na Tabela 9. Além disso, ele é aplicado a blocos com diversos tipos de textura, já que o mesmo realiza uma interpolação das amostras vizinhas ao realizar a predição. Desta forma, este tipo de modo de predição intra é mais difícil de ser diferenciado dos modos de predição intra Angulares, ISP Angulares e MIPs, o que justifica sua taxa de acertos ser mais baixa. Cabe destacar também que ainda que o modo de predição intra Planar possua uma taxa de acertos mais baixa, possuir um erro para a classe Sim traz impactos negativos para a redução de tempo de execução mas não para a eficiência de codificação, já que um erro deste tipo implica em realizar a avaliação dos outros tipos de modos de predição intra mesmo que não seja necessário. Ainda no Modelo 1, para a classe Não, os modos de predição intra Angulares e ISP Angulares possuem boas taxas de acerto, chegando a quase 90%, enquanto que o tipo de modo de predição intra MIP apresenta menor taxa de acerto. Possuir uma taxa de acertos baixa na classe Não traz impactos negativos na eficiência de codificação, já que um erro deste tipo implica em não realizar a avaliação de tipos de modos de predição intra que deveriam ser avaliados e escolhidos.

O Modelo 2 por sua vez traz taxas de acertos boas para ambas as classes e ambos os tipos de modo de predição intra, o que implica que na maioria das vezes através dos RMD-Costs e RD-Costs é possível separar os exemplos do tipo de modo de predição intra Angular e MIP. Esta observação é importante de ser avaliada no contexto do Modelo 1, pois se os modos de predição intra do tipo Angular e MIP tendem a ser bem

separáveis por um modelo de decisão baseado em árvore, é provável que a F1-score do modelo possa ser melhorada se esses dois tipos de modo forem considerados em classes diferentes no Modelo 1. Isso seria importante, pois neste caso a taxa de erro para o modo de predição intra do tipo MIP atinge quase 30%, o que possivelmente levaria a perdas significativas na eficiência de codificação quando o Modelo 1 decidisse pelo **SIM** quando o melhor modo deveria ser o MIP.

Desta forma, com o objetivo de verificar se é possível aumentar a proporção de acertos para o modo de predição intra do tipo MIP no Modelo 1, principalmente para evitar que o método não introduza perdas significativas na eficiência de codificação, avaliou-se dividir o Modelo 1 em dois modelos, onde ambos possuem exemplos do tipo Planar e DC para a classe **SIM** mas para a classe **NÃO** um deles possui exemplos do tipo Angular e ISP Angular (Modelo 1.1) e o outro possui exemplos do tipo MIP (Modelo 1.2). Para realizar esta avaliação, dois novos *datasets* foram gerados e a partir destes uma nova fase de desenvolvimento e teste foi realizada sobre um modelo baseado em árvore de decisão com profundidade máxima igual a 10. Os resultados obtidos desta avaliação são apresentados na Tabela 13.

Tabela 13 – F1-scores obtidas ao dividir o Modelo 1 em dois modelos

Modelo	Classe SIM	Classe NÃO	Fscore CV=5	Fscore Teste
1.1	Planar DC	Angular ISP Ang.	85,15	85,16
1.2	Planar DC	MIP	78,63	78,87

De acordo com os resultados obtidos na Tabela 13, o Modelo 1.1 obtém uma F1-score maior em relação ao Modelo 1, chegando a 85,16%. Entretanto, ao considerar as F1-scores obtidas para o CV=5 e para a fase de teste, os Modelos 1.1 e 1.2 apresentam uma média de F1-score de 81,96%, a qual é muito próxima da média de F1-score obtida para o Modelo 1 na Tabela 11. A primeira vista, não é possível dizer se houve melhorias nas F1-scores ao dividir o Modelo 1 em dois modelos. Logo, para facilitar esta análise, novamente as acurácias são apresentadas na Tabela 14, onde para os Modelos 1.1 e 1.2 são apresentadas as proporções de acertos e erros para cada classe e para cada tipo de modo de predição intra dentro de cada classe. Estas proporções foram obtidas a partir dos resultados da etapa de teste.

Ao analisar os resultados obtidos na Tabela 14 e compará-los com os resultados obtidos para o Modelo 1 na Tabela 12, nota-se que existem melhorias ao dividir o Modelo 1 em dois modelos. Essas melhorias podem ser explicadas por alguns motivos. Primeiro, ao considerar a média de acertos obtidos com os Modelos 1.1 e 1.2 para o modo de predição intra do tipo Planar, esta é de 68,17%, maior que os 65,02% obtidos para o Modelo 1. Segundo, houve um aumento significativo na taxa de acertos para o

Tabela 14 – Taxa de Acertos por Classe e por Tipo de Modo de Predição Intra ao dividir o Modelo 1 em dois modelos

Modelo	Classe	Tipo	Acertos (%)	Erros (%)	Média Acertos (%) Classe	Média Erros (%) Classe
1.1	SIM	Planar	72,95	27,05	85,78	14,22
		DC	98,61	1,39		
	NÃO	Angular	83,95	16,05	84,62	15,38
		ISP Ang.	85,29	14,71		
1.2	SIM	Planar	63,39	36,61	75,97	24,03
		DC	88,55	11,45		
	NÃO	MIP	81,79	18,21	81,79	18,21

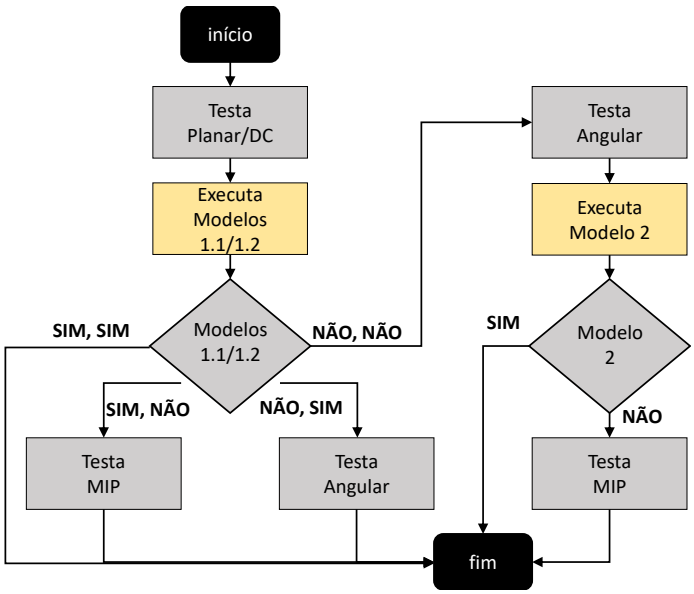


Figura 14 – Ilustração do Modo de Decisão Rápido Proposto Modificado

modo de predição intra do tipo MIP, chegando a 81,79%. Por estas razões, optou-se por dividir o Modelo 1 em dois Modelos e esta modificação no modo de decisão rápido proposto pode ser vista na Figura 14.

A principal diferença que esta alteração traz é que na proposta inicial apresentada na Figura 12 quando o Modelo 1 optava por não avaliar modos, os modos de predição intra do tipo Angular, ISP Angular e MIP eram desconsiderados juntos e o fluxo seguia para a decisão final. Agora, existem quatro possibilidades do que pode ocorrer. Na primeira possibilidade, o Modelo 1.1 pode optar por avaliar os modos Angulares mas o Modelo 1.2 pode optar por não avaliar os modos MIPs, o que significa que somente os modos angulares devem ser avaliados. Na segunda possibilidade, o Modelo 1.1 pode optar por não avaliar os modos Angulares mas o Modelo 1.2 pode optar por avaliar os modos MIPs, o que significa que os modos MIPs devem ser avaliados. Em ambas possibilidades, como os modos Angulares ou os modos MIPs não serão avaliados, não faz sentido seguir para a predição do Modelo 2. Logo, após a avaliação dos modos

Angulares ou dos modos MIPs, o fluxo de execução segue para a decisão final. Na terceira possibilidade, tanto o Modelo 1.1 quanto o Modelo 1.2 podem optar por não avaliar os modos Angulares e os modos MIPs, respectivamente, o que significa que o fluxo deve seguir para a decisão final. Por fim, na quarta possibilidade, tanto o Modelo 1.1 quanto o Modelo 1.2 podem optar por avaliar os modos Angulares e MIPs, respectivamente, o que significa que o fluxo deve seguir para a avaliação dos modos Angulares e em seguida executar a predição do Modelo 2 para verificar se os modos MIPs podem ser desconsiderados.

6.3 Busca de Hiperparâmetros

Para simplificar o processo e considerando a alta quantidade de modelos que foram avaliados na etapa anterior, apenas o hiperparâmetro referente a profundidade das árvores de decisão foi alterado de ilimitado para no máximo 10 com o objetivo de evitar *overfitting* e manter todos os modelos testados em um mesmo nível de complexidade. Entretanto, com o objetivo de realizar uma busca mais concisa em alguns dos hiperparâmetros presentes na árvore de decisão da biblioteca Scikit-Learn (PEDREGOSA et al., 2011), um processo de busca de hiperparâmetros baseado na metodologia apresentada no trabalho de Andrades; Grellert; Fonseca (2019) foi utilizado, o qual consiste em duas principais etapas. Na primeira etapa, realiza-se um *RandomSearch* sobre alguns dos hiperparâmetros em um espaço de busca grande. Ainda que o espaço de busca seja grande, o *RandomSearch* não avalia todas as combinações possíveis, e sim um subconjunto aleatório destas combinações, o que torna a execução do ponto de vista de custo computacional viável. O tamanho deste subconjunto é definido através do número de iterações que o *Random Search* deve realizar. Ao obter os resultados do *RandomSearch*, através do Coeficiente de Pearson é possível detectar quais hiperparâmetros possuem uma maior correlação com o aumento da F1-score dos modelos. Desta forma, são selecionados apenas os N hiperparâmetros que possuem maior correlação com o aumento desta métrica, e apenas estes tem seus valores alterados na avaliação do *Grid Search*, o qual avalia todas as combinações possíveis dentre os valores dispostos na busca.

Para aplicar este processo, torna-se necessário definir quais hiperparâmetros serão levados em consideração e quais espaços de busca serão utilizados no *Random Search* e no *Grid Search*. Também é necessário definir o número de iterações que será utilizado no *Random Search* e o valor de N hiperparâmetros que serão avaliados no Grid Search. Neste trabalho, decidiu-se utilizar um número de iterações igual a 500 e um valor de N igual a dois. Para ser possível validar ao final os resultados obtidos desta busca, foram selecionados aleatoriamente 25% dos dados de cada um dos três *datasets* e estes foram deixados para a etapa de validação. A seguir, serão apresen-

tadas as duas etapas principais deste processo, as decisões que foram tomadas e os resultados em que chegou-se.

6.3.1 Random Search

Como foi mencionado anteriormente, para aplicar este processo é necessário definir quais hiperparâmetros serão avaliados no *Random Search* bem como o espaço de busca para estes. Sendo assim, primeiramente foram selecionados seis hiperparâmetros da árvore de decisão presente na biblioteca sklearn para realizar o *Random Search*. Uma descrição destes hiperparâmetros, seus possíveis tipos de valores e comportamento padrão são apresentados abaixo:

- **criterion:** especifica a métrica que é utilizada para medir a qualidade de um particionamento em um nodo da árvore de decisão. Existem duas métricas disponíveis, sendo estas *Gini* e *Entropy*. Por padrão, este hiperparâmetro utiliza a métrica *Gini*.

- **min samples split:** define o número mínimo de exemplos que um nodo deve possuir para que seja possível realizar um particionamento. É um dos hiperparâmetros que controla a altura da árvore, já que valores menores ou maiores fazem com que os nodos sejam mais ou menos expandidos, respectivamente. Os valores deste hiperparâmetro podem ser um número inteiro, o qual indica o número mínimo exato de exemplos que o nodo deve possuir antes de realizar o particionamento, ou uma fração, a qual indica a proporção mínima de exemplos que um nodo deve possuir antes de realizar o particionamento. Esta proporção é em relação ao número total de exemplos presentes no *dataset*. O valor padrão deste hiperparâmetro é dois, o que indica que os nodos da árvore de decisão são expandidos até que tenha-se folhas puras. Uma folha pura é aquela que contém exemplos de apenas uma classe.

- **min samples leaf:** define o número mínimo de exemplos que um nodo folha deve possuir após um particionamento. Também é um dos hiperparâmetros que controla a altura da árvore e possui o mesmo comportamento que o hiperparâmetro *min samples split* ao aumentar e diminuir seus valores. Seus valores também podem ser números inteiros e frações. O valor padrão deste hiperparâmetro é um, o que também indica que os nodos da árvore de decisão são expandidos até que tenha-se folhas puras.

- **max depth:** define a profundidade máxima da árvore de decisão. Este hiperparâmetro possui impacto tanto no *min samples split* quanto no *min samples leaf*, já que ainda que um particionamento seja permitido de acordo com estes hiperparâmetros, se o mesmo resultar em uma profundidade além do que a qual

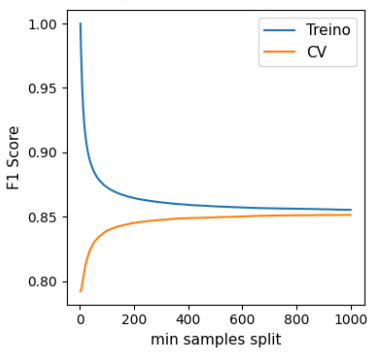
o hiperparâmetro *max depth* define, este não pode ser realizado. Seus valores podem ser números inteiros e o comportamento padrão é não limitar a profundidade máxima.

- **max features:** define a quantidade de *features* que são avaliadas para decidir qual é a *feature* que gera o melhor particionamento em um nodo da árvore de decisão. Esta avaliação é feita de acordo com a métrica definida em *criterion*. Os valores deste hiperparâmetro podem ser números inteiros e o comportamento padrão é avaliar todas as *features* do *dataset*. Caso o valor seja menor do que o número total de *features* presentes no *dataset*, de acordo com o valor definido são selecionadas N *features* aleatórias para realizar a avaliação.
- **max leaf nodes:** define a quantidade máxima de nodos folha que a árvore de decisão pode possuir. É um dos hiperparâmetros que também possui impacto na altura da árvore de decisão, já que limitar a quantidade de nodos folha que a árvore de decisão pode possuir faz com que os nodos sejam expandidos somente até certo ponto. Os valores deste hiperparâmetro podem ser números inteiros e o comportamento padrão é não limitar a quantidade máxima de nodos folha.

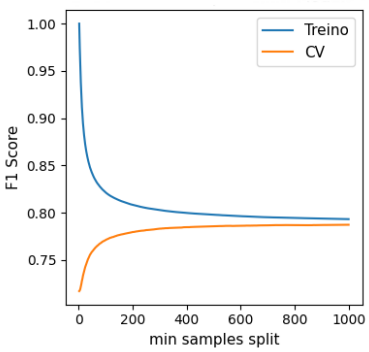
Como pode ser visto pela descrição acima, a maior parte dos hiperparâmetros selecionados controlam a altura da árvore de decisão. Decidiu-se trabalhar com estes hiperparâmetros principalmente porque de acordo com seus valores padrão, o comportamento é não limitar a altura ou a quantidade de nodos de decisão e de nodos folha presentes na árvore de decisão. Isto claramente resulta em modelos complexos e superespecializados no conjunto de treinamento, o que não é desejado, e reforça a necessidade de ter-se um processo de busca de hiperparâmetros mais conciso.

Após selecionar os hiperparâmetros que são considerados no *Random Search*, definiu-se um espaço de busca para cada um destes. Para os hiperparâmetros *criterion* e *max features*, decidiu-se considerar todo o espaço de busca disponível, já que este não é muito grande. Para o restante dos hiperparâmetros, avaliou-se o comportamento de cada um através de curvas de validação. Estas curvas de validação demonstram o comportamento de cada hiperparâmetro conforme seu valor cresce e os outros hiperparâmetros permanecem no valor padrão. Dentro de cada curva, existe a Fscore para o conjunto de treinamento e para o conjunto de teste, sendo que a última é obtida através da média da Fscore de um CV = 5. Cada uma das curvas geradas são apresentadas na Figura 15.

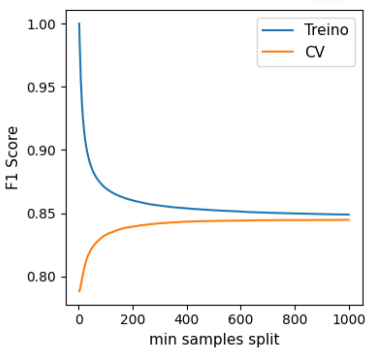
De acordo com a Figura 15, para o hiperparâmetro *min sample split*, os valores menores geram *overfitting*, já que a F1-score para o conjunto de treinamento é igual ou está muito próxima à 100% enquanto que para o conjunto de teste a F1-score está



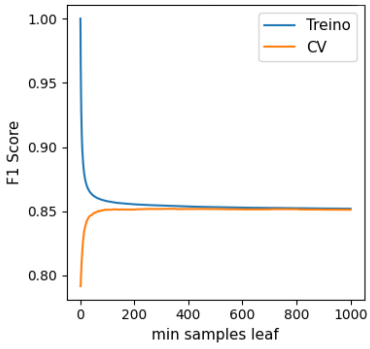
(a) Min Samples Split: Modelo 1.1



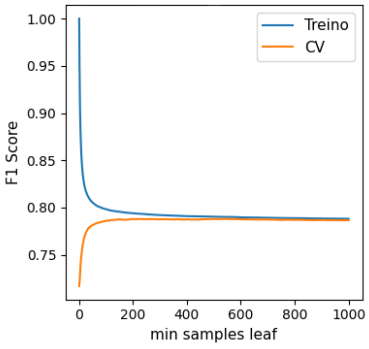
(b) Min Samples Split: Modelo 1.2



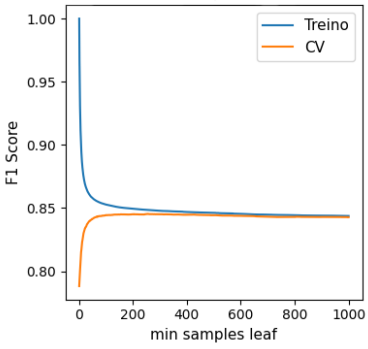
(c) Min Samples Split: Modelo 2



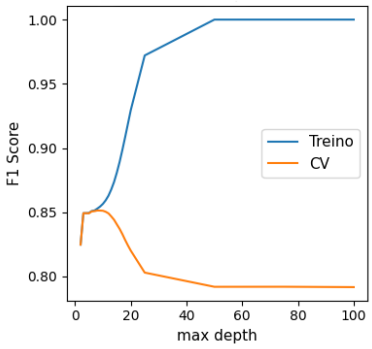
(d) Min Samples Leaf: Modelo 1.1



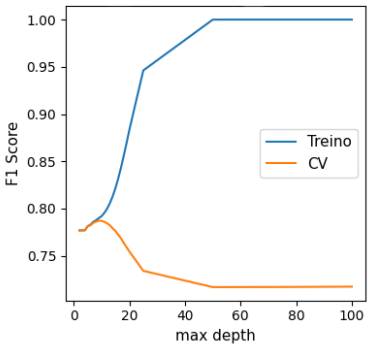
(e) Min Samples Leaf: Modelo 1.2



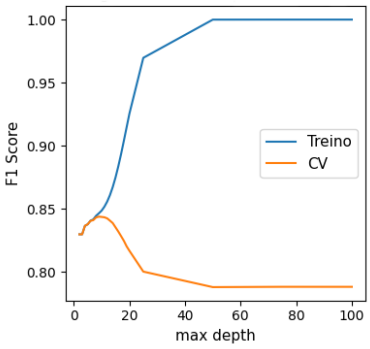
(f) Min Samples Leaf: Modelo 2



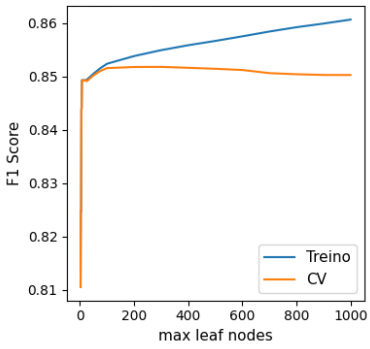
(g) Max Depth: Modelo 1.1



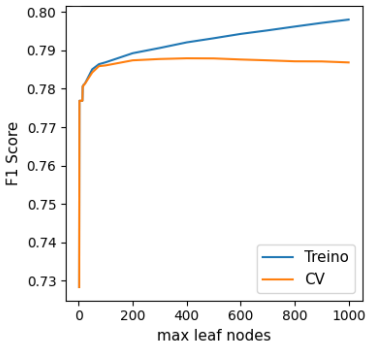
(h) Max Depth: Modelo 1.2



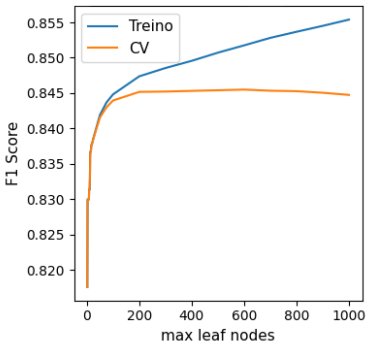
(i) Max Depth: Modelo 2



(j) Max Leaf Nodes: Modelo 1.1



(k) Max Leaf Nodes: Modelo 1.2



(l) Max Leaf Nodes: Modelo 2

Figura 15 – Curvas de Validação para cada Hiperparâmetro e cada Modelo

abaixo de 85%. Conforme o valor do hiperparâmetro cresce, as curvas dos conjuntos de treinamento e teste tendem a convergir para um mesmo ponto. Se os valores deste hiperparâmetro continuassem a crescer além do que o que é mostrado na Figura 15, a tendência é que ambas as curvas caíssem juntas em F1-score, já que começaria a se ter um caso de *underfitting*. O mesmo comportamento do hiperparâmetro *min samples split* pode ser observado para o hiperparâmetro *min samples leaf*, onde o que muda é o ponto onde ambas as curvas começam a convergir para um mesmo valor. Para o hiperparâmetro *max depth*, pode ser visto que para todos os modelos a F1-score cai drasticamente em profundidades máximas que se aproximam, são maiores ou iguais a 20. Por fim, para o hiperparâmetro *max leaf nodes*, existe uma melhora significativa na F1-score conforme o valor deste hiperparâmetro cresce. Entretanto, pode ser observado que para todos os modelos esta melhora na F1-score permanece estável ou começa a cair a partir de valores próximos, superiores ou iguais a 300. Baseado nestas análises, o espaço de busca para cada hiperparâmetro apresentado na Tabela 15 é proposto para o *Random Search*.

Tabela 15 – Espaço de Busca para cada Hiperparâmetro no Random Search

Hiperparâmetro	**Espaço de Busca Random Search**	**Total de Valores**
criterion	[gini, entropy]	2
min samples split	[2] U [25, 500, passo 25]	21
min samples leaf	[1] U [10, 100, passo 10]	11
max depth	[2, 9, passo 1] U [10, 100, passo 10]	18
max leaf nodes	[2, 9, passo 1] U [25, 300, passo 25]	20
max features	[1, 15, passo 1] ou [1, 18, passo 1]	15 ou 18
	Total de Combinações	**2.494.800 ou 2.993.760**

De acordo com a Tabela 15, com exceção dos hiperparâmetros *criterion*, *max depth* e *max features*, o restante dos hiperparparâmetros teve um espaço de busca definido com base na Figura 15, sendo que este espaço foi gerado ao observar o ponto em que o valor do hiperparâmetro gerava uma F1-score estável ou começava a declinar. Quanto ao hiperparâmetro *max depth*, decidiu-se aumentar seu espaço de busca além do ponto em que o mesmo mostra uma perda significativa de F1-score de acordo com a Figura 15. Esta decisão foi tomada para que se tivesse um número de combinações maiores no espaço de busca para a etapa do Random Search, já que isto é necessário de acordo com o trabalho de Andrades; Grellert; Fonseca (2019). O espaço de busca do hiperparâmetro *max features* pode possuir dois intervalos distintos e isto deve-se ao fato de que para os Modelos 1.1 e 1.2 existem 15 *features* no *dataset* e para o Modelo 2 existem 18 features no *dataset*, pois o menor RD-Cost do modo angular é adicionado, bem como as razões entre os RD-Costs do modo planar e do modo DC com o RD-Cost do modo angular.

Após definir os hiperparâmetros e o espaço de busca para cada um destes, o

Random Search foi executado para cada modelo. Para esta execução, cada um dos três datasets tiveram seus dados divididos em 75% para desenvolvimento e 25% para teste. Estes 75% foram utilizados na execução do *Random Search*, onde o mesmo teve seu número de iterações igual a 500, ou seja, 500 combinações de valores de hiperparâmetros foram selecionadas aleatoriamente dentro do espaço de busca proposto na Tabela 15, e estas foram avaliadas. Ainda, para cada combinação avaliada no *Random Search*, o método de realizar o treinamento e teste é um *Cross-Validation* (CV) igual a cinco. Sendo assim, foram treinados e testados 2.500 modelos. Desta execução, extraiu-se cada combinação que foi testada e a média da F1-Score dos cinco CV associados a esta combinação. A partir destes dados, o Coeficience de Pearson entre cada hiperparâmetro e a F1-Score foi calculado, e este é apresentado na Tabela 16.

Tabela 16 – Coeficiente de Pearson dos Hiperparâmetros em relação ao aumento da Fscore

Modelo	**criterion**	**min samples split**	**min samples leaf**	**max features**	**max depth**	**max leaf nodes**
1.1	-0,064	0,0414	-0,0565	0,3514	0,0829	0,4229
1.2	-0,0335	0,0574	0,0313	0,4695	0,0888	0,3806
2	-0,0025	-0,0737	-0,0106	0,4702	0,1151	0,2369

Com base nos coeficientes apresentados na Tabela 16, é possível concluir que para todos os modelos os dois hiperparâmetros que apresentaram uma maior correlação com o aumento da F1-score foram o *max features* e o *max leaf nodes*. Esta correlação foi positiva, o que indica que ao crescer o valor destes dois hiperparâmetros a F1-score também cresceu. A seguir será apresentada a etapa seguinte ao *Random Search*, a qual consistiu em executar um *Grid Search* em cima dos dois hiperparâmetros que apresentaram uma maior correlação com o aumento da *F1-score*.

6.3.2 Grid Search

A etapa seguinte ao *Random Search* consiste em executar um *Grid Search* sobre os dois hiperparâmetros que apresentaram uma maior correlação com o aumento da *Fscore* dos modelos. De acordo com os dados apresentados na Tabela 16, para todos os modelos os dois hiperparâmetros foram o *max features* e o *max leaf nodes*. Como o *Grid Search* é uma busca mais custosa já que nela são avaliadas todas as combinações possíveis dentro do espaço de busca, a ideia principal nesta etapa é variar apenas os valores destes dois hiperparâmetros enquanto os outros permanecem no valor padrão.

Na Tabela 17, o espaço de busca proposto para o *Grid Search* é apresentado. Como pode ser visto, para todos os hiperparâmetros que não os dois que possuem maior correlação com o aumento da *Fscore*, seu espaço de busca foi definido para

o valor padrão. Entretanto, decidiu-se adicionar também o valor encontrado como melhor dentro do Random Search. Para o hiperparâmetro *max features* novamente todo o espaço de busca foi considerado e para o hiperparâmetro *max leaf nodes* o espaço de busca foi definido baseado na Figura 15. Neste espaço de busca existe um total de 3.720 ou 4.464 combinações, dependendo do total de *features* no *dataset*. Para cada combinação, o *Grid Search* realiza um *Cross-Validation* igual a cinco, o que faz com que nesta etapa tenham sido treinados e testados 18.600 ou 22.320 modelos. Este treino e teste foi realizado ao utilizar os mesmos 75% dos dados que haviam sido utilizados na etapa do *Random Search*.

Tabela 17 – Espaço de Busca para cada Hiperparâmetro no Grid Search

Hiperparâmetro	**Espaço de Busca Grid Search**	**Total de Valores**
criterion	[gini]	1
min samples split	[2] U [Best Random Search]	2
min samples leaf	[1] U [Best Random Search]	2
max depth	[None] U [Best Random Search]	2
max leaf nodes	[100, 400, passo 10]	31
max features	[1, 15, passo 1] ou [1, 18, passo 1]	15 ou 18
	Total de Combinações	**3720 ou 4464**

A Tabela 18 apresenta as melhores combinações de valores de hiperparâmetros encontradas para cada modelo na etapa do *Random Search* e do *Grid Search*. Para referência, a configuração padrão também é apresentada. Nesta Tabela, para os hiperparâmetros *max features*, *max depth* e *max leaf nodes*, quando seu valor é igual a *None* significa que não há limite para a quantidade de *features*, profundidade e nodos folha, respectivamente. Por fim, a Tabela 19 apresenta para cada configuração a quantidade total de nodos resultantes na árvore de decisão, a profundidade máxima e os *Fscores* obtidos na etapa de treinamento e validação. O que pode ser visto através dos dados apresentados na Tabela 19 é que tanto o *Random Search* quanto o *Grid Search* encontraram combinações de valores de hiperparâmetros melhores que a padrão, já que a complexidade dos modelos reduziu drasticamente, como pode ser visto através das quantidades de nodos e das profundidades máximas, enquanto que a F1-score subiu consideravelmente. Desta forma, ao fim desta etapa, adotaram se os modelos 1.1, 1.2 e 2 com as combinações de valores de hiperparâmetros encontradas pelo *Grid Search*.

6.4 Integração ao *Versatile Test Model* (VTM)

Ao final da etapa anterior, chegou-se aos modelos necessários para o modo de decisão rápido proposto na Figura 12. Agora, o desafio consiste em integrar estes modelos obtidos ao VTM, já que a biblioteca sickit-learn é desenvolvida para a lingua-

Tabela 18 – Combinações de Valores de Hiperparâmetros obtidas para cada Configuração e cada Modelo

Modelo	Config.	criterion	min samples split	min samples leaf	max features	max depth	max leaf nodes
1.1	Padrão	gini	2	1	None	None	None
	Random		150	10	14	100	150
	Grid		150	1	8	None	270
1.2	Padrão	gini	2	1	None	None	None
	Random		25	70	9	10	250
	Grid		2	70	10	None	400
2	Padrão	gini	2	1	None	None	None
	Random		350	20	16	50	225
	Grid		350	1	15	None	400

Tabela 19 – Total de Nodos, Profundidade e F1-scores obtidas para cada Configuração e cada Modelo

Modelo	Config.	Total de Nodos	Profundidade	Fscore CV=5	Fscore Teste
1.1	Padrão	165.637	55	79,20	79,47
	Random	299	12	85,18	85,27
	Grid	539	13	85,16	85,27
1.2	Padrão	219.727	51	71,66	71,75
	Random	499	10	78,74	78,62
	Grid	799	15	78,80	78,69
2	Padrão	158.707	51	78,83	79,24
	Random	449	14	84,49	84,73
	Grid	799	15	84,53	84,80

gem Python e consequentemente os modelos obtidos encontram-se nesta linguagem, enquanto o VTM é desenvolvido na linguagem C++, bem como implementar o modo de decisão rápido proposto. Primeiramente, como a etapa de treino, teste e validação foi concluída, agora todo o conjunto de dados presente em cada um dos três *datasets* pode ser utilizado para treinar cada um dos modelos. Desta forma, para os Modelos 1.1, 1.2 e 2 aplicou-se os valores de hiperparâmetros encontrados no *Grid Search*, os quais são apresentados na Tabela 18, e então treinou-se cada modelo com seu respectivo *dataset*. Após os treinos, os modelos resultantes foram extraídos para funções em C++ através de um *script* encontrado em Papkov (2016). Estas funções foram então integradas ao VTM 9.0 e utilizadas no desenvolvimento do processo de decisão de modo rápido conforme a Figura 14.

Para verificar o desempenho do método proposto, é necessário codificar as sequências de vídeo presentes na CTC do VVC no codificador de referência e no codificador de referência com o método proposto implementado e então comparar os resultados de tempo de execução e eficiência de codificação obtidos em ambas ver-

sões. Por padrão, ao final de cada codificação o VTM já disponibiliza as métricas relacionadas ao tempo de execução e a eficiência de codificação. Entretanto, decidiu-se obter duas outras medidas relacionadas ao tempo de execução. A primeira delas consiste no tempo total de execução para cada um dos modelos, já que com esta é possível saber o *overhead* que o método insere, e a segunda delas consiste no tempo total de execução do processo de decisão de modo, já que este difere do tempo total de execução do processo de codificação como um todo e portanto as reduções de complexidade podem ser diferentes quando vistas sob o ponto de vista do tempo total de execução de todo o processo de codificação ou do tempo total de execução do processo de decisão de modo. Esta última medida foi adicionada tanto no codificador de referência quanto no codificador de referência com o método proposto implementado.

Por fim, com o objetivo de poder medir o desempenho do método em partes, este foi implementado de forma a poder ser executado com uma ativação incremental dos modelos, ou seja, é possível ter três tipos de ativação dos modelos. Somente o Modelo 1.1, os Modelos 1.1 e 1.2 e, finalmente, a ativação com os Modelos 1.1, 1.2 e 2. No capítulo a seguir, serão apresentados os resultados obtidos com a implementação do modo de decisão rápido proposto.

7 RESULTADOS

O modo de decisão rápido proposto na Figura 14 foi implementado no VTM 9.0. Este pode possuir três versões diferentes, onde a primeira versão possui somente o Modelo 1.1 ativado, a segunda versão possui os Modelos 1.1 e 1.2 ativados e a terceira versão possui os Modelos 1.1, 1.2 e 2 ativados. O método foi implementado desta forma para ser possível avaliar o desempenho do modo de decisão rápido proposto de maneira incremental, ou seja, para verificar o quanto se ganha em redução de complexidade e se perde em eficiência de codificação a cada modelo que é inserido e consequentemente a cada tipo de modo de predição intra-quadro que pode ser possivelmente não avaliado de acordo com a decisão deste modelo.

Com o objetivo de obter os resultados de desempenho do modo de decisão rápido, as instruções presentes nas condições comuns de teste (CTCs) (BOSSEN et al., 2019b) do VVC foram seguidas. Estas instruções trazem 26 sequências de vídeo distribuídas por sete classes e cinco resoluções, conforme é apresentado na Tabela 20. Cada sequência de vídeo deve ser codificada com quatro valores de QP, sendo estes 22, 27, 32 e 37 e com o arquivo de configuração AI (*All Intra*), o qual faz com que o codificador utilize apenas predição intra-quadro para a codificação dos quadros do vídeo. Especificamente para as sequências de vídeo da classe F, as quais contém *frames* que são gravações de tela, ao codificar estas sequências um outro arquivo de configuração referente a esta classe também deve ser utilizado, o qual ativa ferramentas de predição intra específicas para este tipo de vídeo.

Como existem 26 sequências de vídeo que devem ser codificadas com quatro valores de QP no codificador de referência e no mesmo codificador de referência com a solução proposta implementada, existem no mínimo 208 codificações que devem ser executadas, isto sem considerar as três versões existentes do método proposto. Dada a alta quantidade de codificações a serem executadas para a primeira e segunda versão do método proposto, ou seja, as que contém apenas o Modelo 1.1 ativado e apenas os Modelos 1.1 e 1.2 ativados, respectivamente, decidiu-se selecionar aleatoriamente apenas três sequências de vídeos das classes A1 (*FoodMarket4*), B (*BasketballDrive*) e E (*FourPeople*) para compor a sequência de testes para estas

Tabela 20 – 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	50	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
F	* ArenaOfValor	600	60	8
* 1920x1080	§ BasketballDrillText	500	50	8
# 1280x720	# SlideEditing	300	30	8
§ 832x480	# SlideShow	500	20	8

duas versões. Para a terceira versão do método, a qual contém todos os modelos ativados, todas as sequências de vídeo foram codificadas. Sendo assim, para obter os resultados aqui apresentados, foram executadas 104 codificações no codificador de referência, 12 codificações tanto na primeira versão quanto na segunda versão do método proposto e novamente 104 codificações na terceira versão do método proposto, o que gera um total de 232 codificações executadas. Estas codificações foram executadas em um servidor dedicado para isto, o qual conta com um processador Intel Xeon Gold 5120 2.20GHz e uma memória RAM de 64GB com 2666MHz. Foram executadas uma codificação por vez para que fatores externos como superaquecimento do processador ou escalonamento dos processos não influenciem nos tempos totais de execução obtidos. É importante salientar que esta metodologia para medição de tempo é a mais utilizada em trabalhos que propõe soluções de redução de complexidade para a codificação de vídeo.

Para cada sequência de vídeo que foi codificada, calculou-se o *Time Saving* (TS),

o *Time Saving* durante o processo de decisão de modo intra (TSD), o BD-BR (BJONTEGAARD, 2001), a relação entre o TS e o BD-BR e o Tempo do(s) modelo(s). O TS (8) indica a proporção de tempo que foi reduzida através do método ao comparar os tempos totais de execução do VTM 9.0 e do VTM 9.0 com o método implementado ao codificarem um mesmo vídeo. O TSD (9) é análogo ao TS, entretanto, sua comparação se dá no tempo total de execução do processo de decisão de modo. O BD-BR indica a perda ou o ganho em eficiência de codificação para uma mesma qualidade visual ao comparar os *bitrates* obtidos para o VTM 9.0 e para o VTM 9.0 com o método implementado. Isso significa que valores positivos indicam uma perda na eficiência de codificação, pois mais bits são necessários para representar o vídeo com a mesma qualidade. O cálculo do BD-BR foi realizado utilizando um *script* em Python (ANSERW, 2016). A relação entre o TS e o BD-BR é uma divisão entre estes dois valores e demonstra, para cada 1% de perda em eficiência de codificação, o quanto se ganha em redução de tempo de execução. Por fim, o Tempo do(s) Modelo(s) (12) indica proporcionalmente o quanto do tempo total de execução do(s) modelo(s) corresponde ao tempo total de toda a codificação do vídeo. Como para cada sequência de vídeo existem quatro resultados por conta dos valores de QP, para calcular estas medidas considerou-se a média dos valores obtidos para cada um dos QPs (10) (11). Nas equações apresentadas em (8) e (9), *FastVTM9.0* diz respeito ao VTM 9.0 com o modo de decisão rápido implementado e nas equações (10) e (11), $QP_{(1)} = 22$, $QP_{(2)} = 27$, $QP_{(3)} = 32$ e $QP_{(4)} = 37$.

$$TimeSaving = 100 - \left(\frac{TempoTotal_{(FastVTM9.0)} * 100}{TempoTotal_{(VTM9.0)}} \right) \tag{8}$$

$$TimeSavingDecisao = 100 - \left(\frac{TempoTotalDecisao_{(FastVTM9.0)} * 100}{TempoTotalDecisao_{(VTM9.0)}} \right) \tag{9}$$

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

$$TSD = \frac{\sum_{i=1}^{4} TimeSavingDecisao_{QP_{(i)}}}{4} \tag{11}$$

$$TempoModelo_{(i)} = \frac{TempoTotalModelo_{(i)} * 100}{TempoTotal_{FastVTM9.0}},\ onde\ i = 1.1,\ 1.2\ ou\ 2 \tag{12}$$

7.1 Resultados para o Modelo 1.1

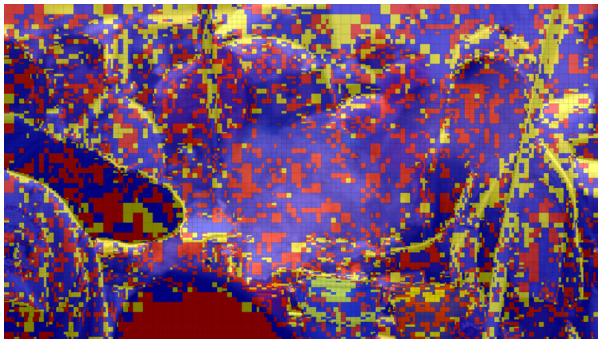
A Tabela 21 apresenta os resultados obtidos para a primeira versão do método, a qual possui apenas o Modelo 1.1 sendo executado. Como pode ser visto na Figura 14, o Modelo 1.1 é responsável por decidir se os custos taxa-distorção dos modos

Tabela 21 – Time Savings, BD-Rates e Tempos do Modelo obtidos para a primeira versão do método proposto.

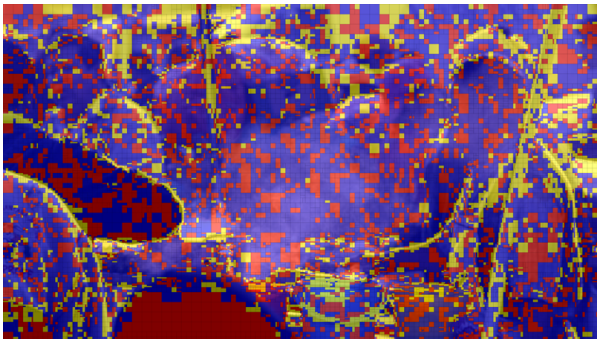
Sequência	TS (%)	TSD (%)	BD-BR (%)	TS/BD-BR	Tempo Modelo 1.1 (%)
FoodMarket4	6,7	8,64	0,2548	26,30	0,07
BasketballDrive	5,86	7,16	0,3526	16,62	0,08
FourPeople	5,81	7,11	0,3312	17,54	0,09
Média	**6,12**	**7,64**	**0,3129**	**20,15**	**0,08**

não angulares (DC e Planar) já são suficientes para gerar uma boa eficiência de codificação e portanto não avaliar os modos angulares. Desta forma, nesta versão do método os modos MIPs nunca são desconsiderados. Ao observar os valores apresentados na Tabela 21, o melhor e pior cenário dependem do ponto de vista em que estes são considerados. O melhor cenário pode ser considerado aquele em que se obtém o maior TS, entretanto ao fazer isto ignora-se o quanto de BD-Rate este ganho em TS trouxe. O mesmo ocorre ao considerar apenas o menor BD-Rate obtido, já que neste caso ignora-se o quanto de TS foi obtido em relação a este pouco acréscimo em BD-BR. Desta forma, neste trabalho o melhor cenário é considerado aquele em que a relação TS/BD-Rate resulta no maior valor, pois neste caso um maior valor para esta medida indica que a solução foi capaz de obter altos valores de TS para baixos valores de BD-BR. Sendo assim, pode ser visto que o melhor cenário obtido é para a sequência de vídeo FoodMarket4, onde para cada 1 de perda em eficiência de codificação foi possível ganhar 26,30 em tempo de execução. O pior cenário ocorre para a sequência de vídeo BasketballDrive, onde para cada 1 de perda em eficiência de codificação ganhou-se 17,54 em tempo de execução. Por fim, ao observar o Tempo do Modelo 1.1, nota-se que o método proposto traz pouco *overhead*, já que o tempo total de execução do Modelo 1.1 é em média apenas 0,08% do tempo total de execução.

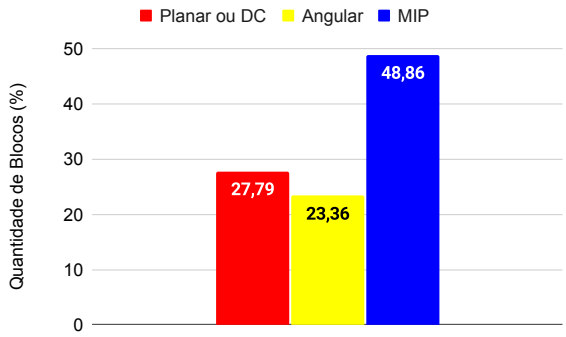
Com o objetivo de poder visualizar o comportamento da primeira versão do modo de decisão rápido proposto no melhor e no pior cenário, as Figuras 16 e 17 trazem um quadro selecionado de forma aleatória para as sequências de vídeo FoodMarket4 e BasketballDrive. Cada quadro possui duas versões, onde a primeira versão foi extraída do processo de codificação do VTM 9.0 e a segunda versão foi extraída do processo de codificação do VTM 9.0 com a primeira versão do modo de decisão rápido implementado (Fast VTM 9.0). Para cada quadro, as decisões de particionamento e de tipo de modo intra resultantes do processo de decisão de modo foram sobrepostas. Nesta sobreposição, a cor vermelha representa os modos de predição intra Planar e DC, a cor amarela representa os modos de predição intra Angulares e a cor azul representa os modos de predição intra MIP. Abaixo de cada quadro também é apresentado estatisticamente a proporção de cada tipo de modo intra.



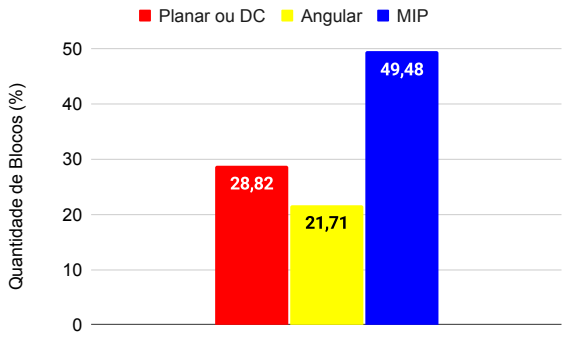
(a) Quadro no VTM 9.0



(b) Quadro no Fast VTM 9.0



(c) Estatísticas do Quadro no VTM 9.0

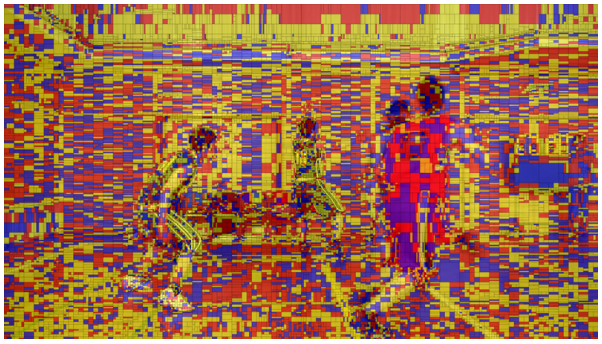


(d) Estatísticas do Quadro no Fast VTM 9.0

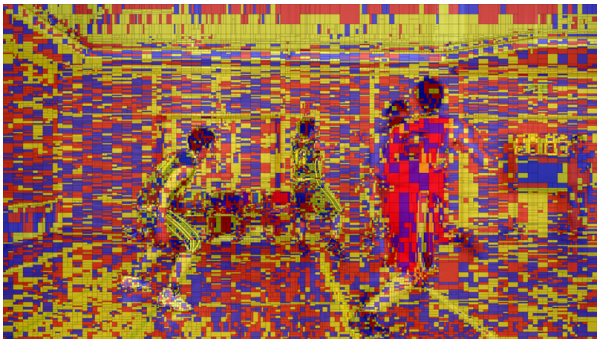
Figura 16 – Quadros e Estatísticas dos Quadros para o vídeo FoodMarket4 no VTM 9.0 e na primeira versão do Fast VTM 9.0

Ao observar a Figura 16, a qual refere-se ao melhor resultado obtido na tabela 21, nota-se que existe uma proporção maior de blocos que foram preditos com os modos de predição intra-quadro Planar, DC e MIP em relação aos modos de predição intra-quadro Angulares. Supondo que isto seja característico desta sequência e portanto sejam mantidas proporções parecidas no mínimo na maioria dos quadros, a razão pela qual a primeira versão do modo de decisão rápido obteve um melhor resultado para a sequência de vídeo FoodMarket4 é porque existe uma baixa proporção de blocos para os quais o modo de predição intra-quadro do tipo Angular é o melhor. Desta forma, o método possui um espaço maior para redução de complexidade e menor para erros, já que existem mais blocos onde a avaliação dos modos intra do tipo angular pode ser desconsiderada sem impactos negativos na eficiência de codificação.

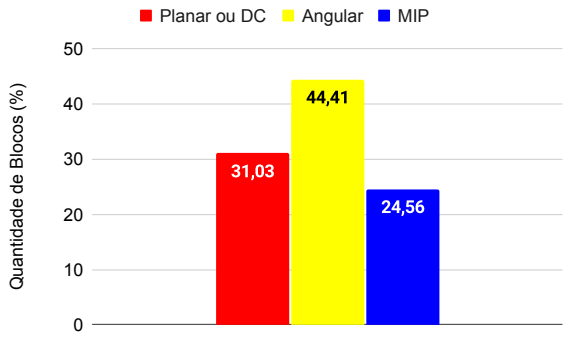
Em contrapartida, ao observar a Figura 17, a qual refere-se ao pior resultado do Modelo 1.1, nota-se que o oposto ocorre. Para este caso, existe uma proporção maior de blocos para os quais os modos de predição intra-quadro angulares foram melhores em relação a proporção observada na Figura 16. Sendo assim, o espaço para redução de complexidade é menor enquanto o espaço para cometer erros torna-se maior, já que agora existem mais blocos onde desconsiderar a avaliação dos modos de predição intra-quadro angulares pode trazer impactos negativos na eficiência de codificação.



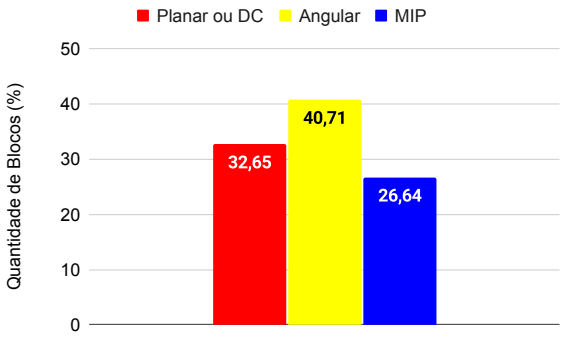
(a) Quadro no VTM 9.0



(b) Quadro no Fast VTM 9.0



(c) Estatísticas do Quadro no VTM 9.0



(d) Estatísticas do Quadro no Fast VTM 9.0

Figura 17 – Quadros e Estatísticas dos Quadros para o vídeo BasketballDrive no VTM 9.0 e na primeira versão do Fast VTM 9.0

Por fim, ao comparar as proporções obtidas nas Figuras 16c e 16d e nas Figuras 17c e 17d, é possível notar que a primeira versão do modo de decisão rápido traz resultados próximos ao processo de decisão de modo original do VTM 9.0, o que é um indicativo de que o Modelo 1.1 está sendo capaz de realizar novas predições com uma boa precisão. Como os erros nesta primeira versão do método podem ocorrer ao desconsiderar a avaliação dos modos Angulares quando esta não deveria ser desconsiderada, nota-se que existe uma baixa redução na proporção de blocos que foram preditos com os modos Angulares, com baixo acréscimo nas proporções de blocos que foram preditos com os modos DC e Planar e os modos MIP. Isso ocorre pois nesta versão do método, no caso de um erro, um bloco que era predito com um modo Angular passa a ser predito com um modo DC, Planar ou MIP.

7.2 Resultados para os Modelos 1.1 e 1.2

Na Tabela 22 são apresentados os resultados obtidos para a segunda versão do método, ou seja, quando os Modelos 1.1 e 1.2 estão ativados. Neste caso, o Modelo 1.2 decide de acordo com os custos taxa-distorção dos modos DC e Planar se estes já são suficientes e portanto os modos MIPs não são avaliados. Logo, esta segunda versão do método é capaz de pular a avaliação dos modos angulares e também a avaliação dos modos MIPs de acordo com as decisões retornadas pelos modelos.

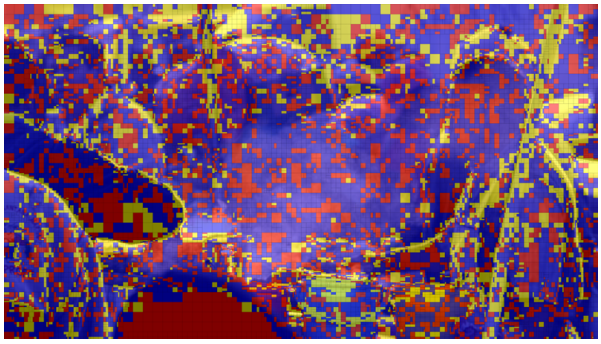
Tabela 22 – Time Savings, BD-Rates e Tempos dos Modelos obtidos para a segunda versão do método proposto

Sequência	TS (%)	TSD (%)	BD-BR (%)	TS/BD-BR	Tempo Modelos 1.1 e 1.2 (%)
FoodMarket4	9,82	12,58	0,3676	26,71	0,13
BasketballDrive	10,83	13,02	0,4058	26,69	0,16
FourPeople	9,58	11,53	0,4089	23,43	0,16
Média	**10,08**	**12,38**	**0,3941**	**25,61**	**0,15**

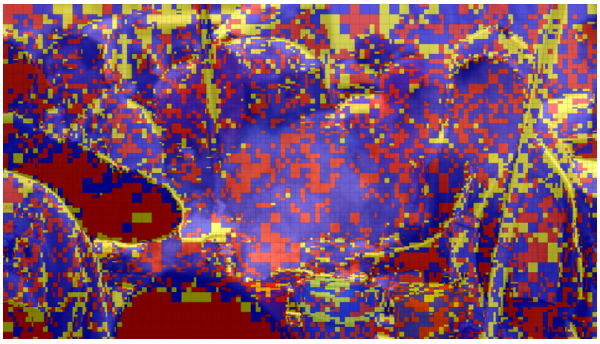
O melhor resultado em termos de TS/BD-BR, assim como na primeira versão do método, repete-se para a sequência de vídeo FoodMarket4, onde para cada 1 de perda em eficiência de codificação foi possível obter 26,71 em tempo de execução. O pior resultado em termos de TS/BD-BR, desta vez ocorre para a sequência de vídeo FourPeople, onde para cada 1 de perda em eficiência de codificação obteve-se 23,43 em tempo de execução. Também é possível notar ao comparar as médias de TS/BD-Rate obtidas nas Tabelas 21 e 22 que a segunda versão do método trouxe em média um ganho de cerca de 5 em tempo de execução para cada 1 de perda de eficiência de codificação quando comparada a primeira versão, o que é esperado já que na primeira versão do método somente os modos angulares eram passíveis de terem sua avaliação desconsiderada e agora na segunda versão do método a avaliação dos modos MIPs também pode ser desconsiderada. Também nota-se um aumento em média da proporção do tempo total de execução dos modelos que corresponde ao tempo total de execução do processo de codificação, a qual passou de 0,08 para 0,15, o que faz sentido já que agora existem dois modelos sendo executados ao invés de somente um.

Com o objetivo de visualizar o desempenho da segunda versão do método no melhor e no pior caso, as Figuras 18 e 19 trazem novamente quadros extraídos do processo de codificação do VTM 9.0 e do processo de codificação da segunda versão do Fast VTM 9.0, ambos com a sobreposição dos modos de predição intra-quadro resultantes do processo de decisão de modo. Abaixo de cada quadro também são apresentadas as estatísticas das proporções de cada modo de predição intra-quadro.

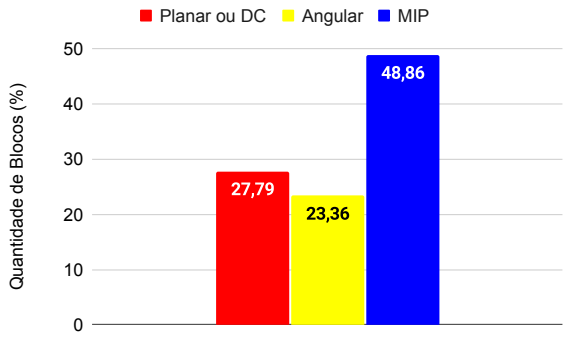
Como pode ser visto através das proporções apresentadas em 18c e 18d e em 19c e 19d, na segunda versão do modo de decisão rápido há redução na proporção de blocos que foram preditos com modos Angulares e MIPs enquanto há um acréscimo na proporção de blocos que foram preditos com modos DC e Planar. Isto é esperado já que na segunda versão do método podem existir erros tanto nos blocos onde a melhor alternativa é utilizar modos Angulares quanto nos blocos onde a melhor alternativa é utilizar modos MIPs. Ao comparar o melhor e o pior caso, os quais correspondem as Figuras 18 e 19 respectivamente, nota-se que em ambos os casos a proporção



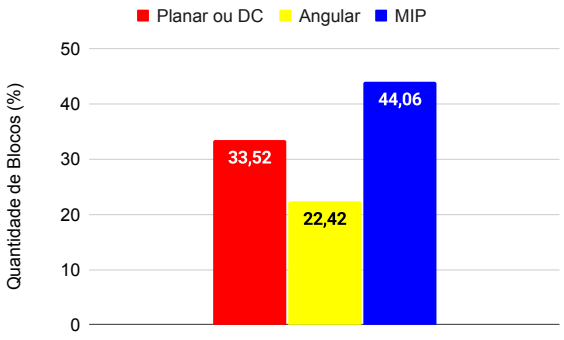
(a) Quadro no VTM 9.0



(b) Quadro no Fast VTM 9.0



(c) Estatísticas do Quadro no VTM 9.0

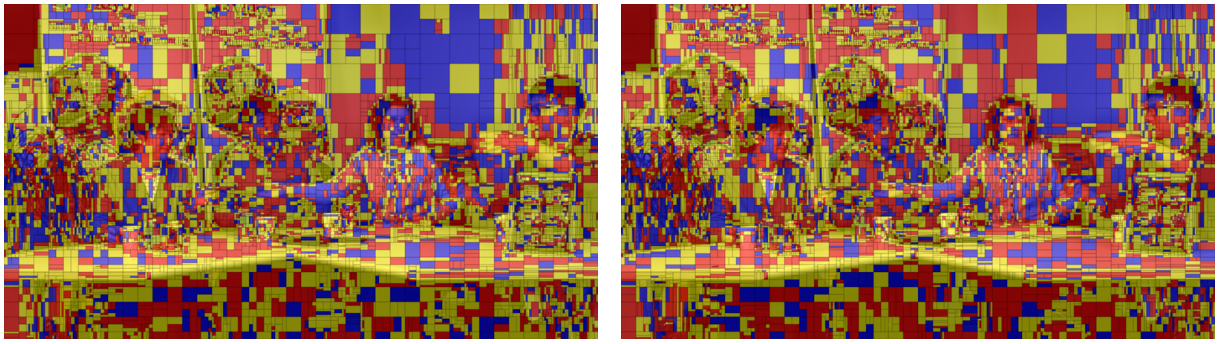


(d) Estatísticas do Quadro no Fast VTM 9.0

Figura 18 – Quadros e Estatísticas dos Quadros para o vídeo FoodMarket4 no VTM 9.0 e na segunda versão do Fast VTM 9.0

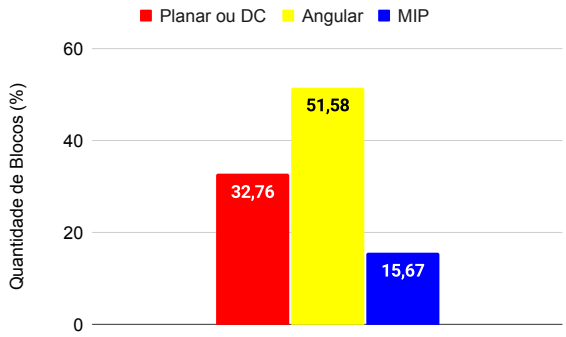
de erros que o método inseriu foi similar, onde o que muda é em qual tipo de modo de predição intra-quadro a maioria destes erros ocorreu. No melhor caso, há uma proporção maior de blocos onde a melhor alternativa foi utilizar modos de predição intra-quadro MIPs, o que indica que há um espaço menor de redução de complexidade e um espaço maior para erros ao considerar o Modelo 1.2. No pior caso, o oposto ocorre, já que há uma proporção maior de blocos onde a melhor alternativa foi utilizar modos Angulares, ou seja, há um espaço menor de redução de complexidade e um espaço maior para erros ao considerar o Modelo 1.1.

Com esta avaliação é possível concluir que a razão pela qual os resultados para a sequência de vídeo FourPeople são um pouco inferiores aos resultados obtidos para a sequência de video FoodMarket4 é porque os erros inseridos pelo Modelo 1.1 para a sequência de vídeo FourPeople ocorreram em áreas mais críticas deste vídeo e portanto trouxeram perdas mais significativas em eficiência de codificação enquanto que os erros inseridos pelo Modelo 1.2 ocorreram em áreas menos críticas da sequência de vídeo FoodMarket4 e portanto inseriram perdas menos significativas em eficiência de codificação. Em outras palavras, isto significa que um erro no Modelo 1.1 gera uma escolha de modo que está mais distante do modo ótimo enquanto que um erro no Modelo 1.2 gera uma escolha de modo sub-ótima que está mais próximo do modo ótimo.

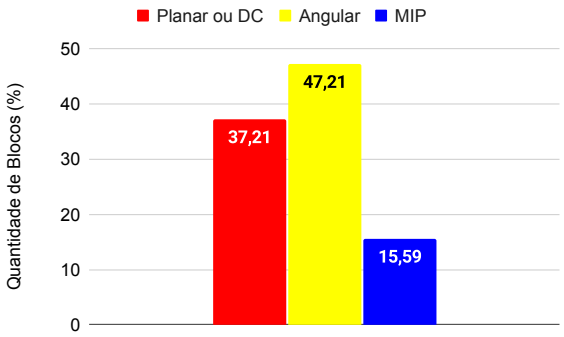


(a) Quadro no VTM 9.0

(b) Quadro no Fast VTM 9.0



(c) Estatísticas do Quadro no VTM 9.0



(d) Estatísticas do Quadro no Fast VTM 9.0

Figura 19 – Quadros e Estatísticas dos Quadros para o vídeo FourPeople no VTM 9.0 e na segunda versão do Fast VTM 9.0

7.3 Resultados para a Solução Completa

Por fim, a Tabela 23 apresenta os resultados obtidos para a versão completa do modo de decisão rápido proposto, a qual conta com os Modelos 1.1, 1.2 e 2 ativados. Na terceira versão do modo de decisão rápido proposto, o Modelo 2 está em execução, o qual é responsável por tentar predizer quando a avaliação dos modos MIPs pode ser pulada de acordo com os custos taxa-distorção obtidos para os modos DC e Planar e modos Angulares. Este modelo só é utilizado no processo de decisão de modo quando os o Modelos 1.1 e 1.2 optaram simultaneamente por não pular a avaliação dos modos Angulares e dos modos MIPs, respectivamente. Desta forma, o principal objetivo do Modelo 2 é explorar os casos onde os Modelos 1.1 e 1.2 não foram capazes de trazer uma redução de complexidade e isto se dá ao detectar quando a avaliação dos modos de predição intra-quadro MIPs pode ser descartada sem que sejam inseridas perdas significativas em eficiência de codificação.

De acordo com os resultados apresentados na Tabela 23, é possível perceber que a terceira versão do modo de decisão rápido traz boas reduções no tempo de execução (TS e TSD) para baixas perdas em eficiência de codificação (BD-BR) para todas as sequências de vídeo presentes na CTC do VVC, o que é um possível indicativo de que os Modelos 1.1, 1.2 e 2 estão sendo capazes de evitar avaliações de modos desnecessárias com uma boa precisão. Para as três últimas sequências de vídeo é

Tabela 23 – Time Savings, BD-Rates e Tempos dos Modelos obtidos para a terceira versão do método proposto

Sequência	TS (%)	TSD (%)	BD-BR (%)	TS/BD-BR	Tempo Modelos 1.1, 1.2 e 2 (%)
Tango2	13,67	16,93	0,6667	20,50	0,17
FoodMarket4	10,70	13,73	0,4484	23,86	0,17
CampFire	15,21	20,06	0,4292	35,44	0,16
CatRobot	11,49	15,05	0,4319	26,60	0,17
DaylightRoad2	14,12	16,86	0,3544	39,84	0,19
ParkRunning3	9,17	14,20	0,2269	40,41	0,15
MarketPlace	13,53	16,58	0,3248	41,66	0,18
RitualDance	11,88	14,50	0,4138	28,71	0,19
Cactus	13,43	16,46	0,3754	35,78	0,19
BasketballDrive	12,30	14,85	0,4357	28,23	0,19
RaceHorses	13,01	16,69	0,2766	47,04	0,17
BQMall	11,13	13,88	0,3921	28,39	0,19
PartyScene	11,84	14,96	0,3174	37,30	0,17
BasketballDrill	8,30	10,72	0,4966	16,71	0,20
RaceHorses	12,56	16,63	0,3806	33,00	0,17
BQSquare	12,51	15,12	0,3354	37,30	0,18
BlowingBubbles	11,30	14,84	0,3473	32,54	0,17
BasketballPass	10,68	14,08	0,4039	26,44	0,18
FourPeople	10,91	13,21	0,4720	23,11	0,21
Johnny	10,67	13,13	0,5083	20,99	0,19
KristenAndSara	9,99	12,48	0,4690	21,30	0,19
ArenaOfValor	4,07	13,25	0,4067	10,01	0,08
BasketballDrillText	3,52	10,46	0,3655	9,63	0,08
SlideEditing	4,90	12,30	0,0710	69,01	0,08
Média	**10,87**	**14,62**	**0,3896**	**30,58**	**0,17**

possível notar uma maior diferença entre os valores de TS e TSD quando comparado com os outros vídeos. Estas sequências de vídeo pertencem a classe F da CTC do VVC, como foi apresentado na Tabela 20, e todas contém *frames* que são gravações de tela de um computador. Por se tratar de uma classe que contém sequências de vídeo com características muito específicas, ao codificar estas sequências a CTC do VVC define que um arquivo de configuração referente a classe F seja utilizado, o qual ativa ferramentas de predição intra específicas para estes tipos de vídeo. Logo, a razão pela qual existe esta grande diferença para os valores de TS e TSD para estas três sequências de vídeo é porque estas ferramentas de predição ativadas por este arquivo de configuração não foram consideradas no modo de decisão rápido proposto e são avaliadas fora do processo de avaliação dos modos tradicionais. Sendo assim, é esperado que para as sequências de vídeo desta classe os resultados obtidos em TS sejam menores do que os resultados obtidos em TSD.

Também é possível notar uma variação nas reduções de tempo de execução obti-

das entre as sequências de vídeo. Isto provavelmente acontece por conta da ordem em que a solução proposta avalia os modos intra e também por conta da complexidade de textura de cada sequência de vídeo. Para sequências de vídeo com complexidade de textura alta, como é o caso da sequência de vídeo *BasketballDrill*, a solução proposta tende a não desconsiderar a avaliação de modos Angulares e MIPs com tanta frequência, o que reduz o potencial de redução de tempo de execução. Por outro lado, para sequências de vídeo com baixa complexidade de textura, como é o caso da sequência de vídeo *CampFire*, a solução proposta tende a desconsiderar a avaliação de modos Angulares e MIPs com maior frequência, já que os custos taxa-distorção dos modos Planar e DC tendem a ser bons o suficiente, e isto leva a maiores ganhos em redução de tempo de execução.

Uma comparação direta da segunda versão do método com a terceira versão do método, para as quais os resultados são apresentados nas Tabelas 22 e 23, não é justa pois para a terceira versão do método existe uma maior quantidade de sequências de vídeo que foram codificadas e portanto comparar as médias obtidas para a segunda versão do método com a terceira versão do método não seria válido. Entretanto, ao observar os resultados obtidos somente das três sequências de vídeo para as quais a segunda versão do método foi aplicada, sendo estas FoodMarket4, BasketballDrive e FourPeople, é possível perceber que ainda que a terceira versão do método traga mais ganhos em TS e TSD quando comparado a segunda versão do método, a relação TS/BD-BR para estas três sequências de vídeo é mais baixa, o que pode indicar que para cada 1 de perda em eficiência de codificação ganhou-se menos em tempo de execução na terceira versão do método do que se ganharia na segunda versão do método. Entretanto, isto são só suposições, já que para se ter uma conclusão mais concisa acerca da comparação entre as duas versões do método seria necessário que para ambas versões todas as sequências de vídeo fossem codificadas.

Ao comparar os Tempos dos Modelos obtidos para a terceira versão do método com a segunda versão do método, nota-se que novamente houve um acréscimo neste tempo, já que agora existem três modelos sendo executados ao invés de dois. Entretanto, este acréscimo foi menor do que o acréscimo obtido ao comparar a primeira e a segunda versão do método e isto se deve provavelmente ao fato de que o Modelo 2 é executado menos vezes do que os Modelos 1.1 e 1.2, o que faz sentido já que para o Modelo 2 ser executado os Modelos 1.1 e 1.2 precisam optar simultaneamente por avaliar os modos de predição intra-quadro Angulares e MIPs, o que deve acontecer com pouca frequência.

Tabela 24 – Comparação do método desenvolvido com trabalhos relacionados

Trabalho	ML	TS (%)	BD-BR (%)	TS/BD-BR	Codificador
Solução Proposta	✓	**10,87**	**0,39**	**30,58**	**VTM 9.0**
Yang et al. (2019)	✗	25,51	0,54	47,24	VTM 2.0
Zouidi et al. (2019)	✗	5,09	2,67	1,91	JEM 7.0
Chen et al. (2020)	✗	30,59	0,86	35,57	VTM 2.0
Zhang et al. (2020)	✗	30,13	0,58	51,95	VTM 4.0

7.4 Comparação com Trabalhos Relacionados

A Tabela 24 apresenta uma comparação do modo de decisão rápido proposto neste trabalho com trabalhos relacionados. As métricas utilizadas para comparação são as mesmas apresentadas nos resultados anteriores, ou seja, TS (*Time saving*), BD-BR e a razão entre TS e BD/BR. Para cada trabalho, também é apresentada a versão do VTM ou do JEM em que este foi desenvolvido e implementado. Estes trabalhos foram selecionados a partir dos resultados da pesquisa realizada no Capítulo 4, onde selecionou-se somente aqueles que realizam reduções de complexidade no processo de decisão de modo da predição intra. No caso dos trabalhos que realizam uma redução de complexidade tanto na decisão de modo intra quando na decisão de particionamento, caso o trabalho apresente o resultado separado somente para a redução de tempo e perdas em eficiência de codificação para a decisão de modo intra, este também é selecionado. Os trabalhos que reduzem complexidade apenas no processo de decisão de modo dos particionamentos do VVC não foram considerados para uma comparação pois estes são complementares a nossa solução, e eventualmente o método aqui proposto pode ser integrado com estes.

De acordo com os valores apresentados na Tabela 24, é possível perceber que a solução proposta neste trabalho apresenta o melhor resultado em termos de BD-BR quando comparado a todos os trabalhos. Também, de acordo com as versões do VTM (ou do JEM quando o caso) onde os métodos apresentados em trabalhos relacionados foram implementados, é possível notar que todos os trabalhos tiveram suas soluções implementadas em versões do VTM iniciais, e, no caso do trabalho Zouidi et al. (2019), o método foi implementada em uma versão ainda mais antiga do VVC, a qual chamava-se JEM e possuía implementações bem inicias do que viria a se tornar o padrão de codificação VVC.

O trabalho de Yang et al. (2019) foi implementado no VTM 2.0. Nesta versão do VTM, ainda não existiam os modos ISPs nem os modos MIPs, sendo que estes só foram inseridos a partir do VTM 4.0 e do VTM 5.0, respectivamente. Como no trabalho de Yang et al. (2019) somente os modos angulares são considerados ao avaliar a possibilidade de descartá-los do processo de decisão de modo intra, se este mesmo método fosse implementado no VTM 9.0 provavelmente não se obteria os mesmo resultados de TS, já que a versão 9.0 conta com os modos intra ISPs e MIPs e estes

não são considerados pelo método de Yang et al. (2019). Logo, a solução proposta é competitiva com a de Yang et al. (2019) não só porque provavelmente podemos obter um melhor TS na versão 9.0, mas também porque mesmo com a nossa solução considerando os modos de predição intra MIPs os resultados em BD-BR foram melhores.

O mesmo comportamento observado no trabalho de Yang et al. (2019) pode ser visualizado para o trabalho de Chen et al. (2020), já que este também foi implementado no VTM 2.0. Para este trabalho, nossos resultados em TS/BD-BR são ainda mais próximos, mesmo que o trabalho de Chen et al. (2020) tenha a vantagem de não considerar os modos intra MIPs, o que indica que a sua solução no VTM 9.0 poderia obter TS/BD-BR menos satisfatórios quando comparados a solução proposta.

Para o trabalho de Zouidi et al. (2019), a solução proposta ganha em todas as métricas. O método de Zouidi et al. (2019) foi implementado em uma versão ainda anterior ao VTM, chamada de JEM. Nas versões relacionadas ao JEM, ferramentas experimentais eram adicionadas para tentar definir-se quais destas estariam nas versões mais estáveis do VVC, as quais migraram para as versões sequentes do VTM. Desta forma, os resultados obtidos no trabalho de Zouidi et al. (2019) são justificáveis do ponto de vista de que o VVC ainda estava em uma fase de desenvolvimento muito experimental e portanto os trabalhos desenvolvidos para redução de complexidade neste momento eram muito experimentais também. Entretanto, cabe destacar que a principal razão pela qual a solução proposta por Zouidi et al. (2019) traz resultados ruins para a relação TS/BD-BR é porque seu processo de decisão de modo dá-se somente no processo do RMD, o qual possui menos impactos em tempo de execução quando comparado ao processo RDO. Desta forma, além do método proposto por Zouidi et al. (2019) estar desconsiderando modos críticos na avaliação do RMD, a qual resulta em impactos negativos na eficiência de codificação, nota-se também que o ganho em TS é muito baixo.

Por fim, o trabalho de Zhang et al. (2020) traz sua solução implementada no VTM 4.0. Nesta versão, os modos ISPs já estão presentes enquanto os modos MIPs ainda não foram trazidos para o VTM. O método de Zhang et al. (2020) utiliza ML, entretanto, o uso se dá para desconsiderar avaliações de particionamentos e não de modos intra. O principal problema com a solução proposta por Zhang et al. (2020) é que o autor não menciona o que ocorre com os modos ISPs quando um modo angular é desconsiderado pela sua solução. Logo, a impressão que fica é que possivelmente estes modos não são considerados pelo autor e até desativados no processo de codificação. Ainda que a solução proposta por Zhang et al. (2020) considere de fato os modos ISPs, o que não fica claro no texto, os modos MIPs não estão presentes no VTM 4.0. Logo, possivelmente os resultados de TS em Zhang et al. (2020) seriam mais baixos na versão do VTM 9.0 e novamente ainda que a solução proposta possua a desvantagem de ter que lidar com os modos MIPs, ainda assim obtém-se um valor menor para BD-BR.

8 CONCLUSÃO

Este trabalho apresentou uma solução para a redução de complexidade no processo de decisão de modo da predição intra do codificador VVC utilizando aprendizado de máquina supervisionado. Esta solução foi implementada no VTM 9.0 e como resultados obteve-se uma média de redução de complexidade de 10,87% em tempo de execução com baixas perdas em eficiência de codificação. No caso onde obteve-se a maior redução de complexidade, a solução proposta foi capaz de reduzir em 15,21% o tempo de execução.

Foi possível atingir esta redução de complexidade principalmente devido a alteração na ordem de avaliação dos modos intra presentes na lista a ser avaliada pelo processo RDO. Como a lista a ser avaliada pelo RDO passou a conter os modos intra pela ordem de probabilidade de serem escolhidos como o melhor ao final do processo de decisão, a cada tipo de modo intra que termina de ser avaliado foram inseridos modelos baseados em árvores de decisão binárias para desconsiderar a avaliação de determinados tipos de modo intra quando os modos intra avaliados até o momento são considerados suficientes. Foram inseridos um total de três modelos, os quais obtiveram Fscores de até 85,27% na etapa de validação. Também é importante destacar que esta alteração na ordem de avaliação dos modos intra na lista do RDO também possibilitou que se trabalhasse apenas com modelos binários ao invés de se trabalhar com modelos que representariam cada modo intra ou ainda um conjunto de modos intra como classes separadas, o que eventualmente resultaria em maiores dificuldades para obter uma acurácia razoável.

Quando comparado a trabalhos relacionados, foi possível notar que nenhuma das soluções encontradas utilizam aprendizado de máquina supervisionado como estratégia para desconsiderar modos de predição intra da avaliação. Além disto, a maioria das soluções foram desenvolvidas para versões do VTM antigas, para as quais os modos ISPs e MIPs não estavam presentes. Logo, ainda que quando comparado a alguns destes trabalhos a solução desenvolvida apresente uma redução de complexidade menor, isto se deve ao fato de que estas soluções encontradas não levam os modos ISPs e MIPs em consideração e portanto caso estas soluções fossem imple-

mentadas no VTM 9.0 como a solução proposta, possivelmente os ganhos em redução de complexidade seriam menores. Destaca-se também que ainda que a solução proposta leve em consideração os modos MIPs e tenha impactos nos modos ISPs, o BD-BR obtido foi menor em relação a todos os trabalhos relacionados encontrados.

Como trabalhos futuros, existe a possibilidade de investigar diferentes arranjos hierárquicos nas avaliações dos modos intra, novas possibilidades de redução de complexidade para os modos ISPs e também novas possibilidades de redução de complexidade para a etapa das transformadas. Como neste trabalho não houve uma comparação dos resultados obtidos com diferentes famílias de modelos de aprendizado de máquina supervisionado, também existe a possibilidade de realizar testes neste sentido para verificar se com modelos da família dos lineares ou dos bayesianos, por exemplo, uma melhor acurácia pode ser atingida e portanto eventuais maiores ganhos em redução de complexidade e menores impactos em eficiência de codificação possam ser obtidos.

REFERÊNCIAS

AMESTOY, T. et al. Random forest oriented fast QTBT frame partitioning. In: ICASSP 2019-2019 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), 2019. **Anais...** IEEE, 2019. p.1837–1841.

ANDRADES, R. S. de; GRELLERT, M.; FONSECA, M. B. Hyperparameter Tuning and its Effects on Cardiac Arrhythmia Prediction. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS), 2019., 2019. **Anais...** IEEE, 2019. p.562–567.

ANSERW. **Bjontegaard metric calculation**. Acessado em 17/08/2021, `https://github.com/Anserw/Bjontegaard_metric`.

BENHAJYOUSSEF, A.; EZZEDINE, T.; BOUALLÈGUE, A. Gradient-based preprocessing for intra prediction in high efficiency video coding. **EURASIP Journal on Image and Video Processing**, Tunes, v.2017, n.1, p.9, 2017.

BJONTEGAARD, G. Calculation of average PSNR differences between RD-curves. **VCEG-M33**, Austin, 2001.

BOSSEN, F. et al. **JVET-N1010**: JVET common test conditions and software reference configurations for SDR video. [S.l.: s.n.], 2019.

BOSSEN, F. et al. **JVET-N1010**: JVET common test conditions and software reference configurations for SDR video. [S.l.: s.n.], 2019.

CHANG, Y.-J. et al. Multiple reference line coding for most probable modes in intra prediction. In: DATA COMPRESSION CONFERENCE (DCC), 2019., 2019. **Anais...** IEEE, 2019. p.559–559.

CHEN, G.; SUN, L.; LIU, Z.; IKENAGA, T. Fast Mode and Depth Decision for HEVC Intra Prediction Based on Edge Detection and Partition Reconfiguration. **IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences**, Naha, v.97, n.11, p.2130–2138, 2014.

CHEN, J. et al. Fast QTMT Partition Decision Algorithm in VVC Intra Coding based on Variance and Gradient. In: IEEE VISUAL COMMUNICATIONS AND IMAGE PROCESSING (VCIP), 2019., 2019. **Anais...** IEEE, 2019. p.1–4.

CHEN, J.; YE, Y.; KIM, S. H. **Test Model 6 of Versatile Video Coding (VTM 6)**. [S.l.: s.n.], 2019.

CHEN, Y. et al. A novel fast intra mode decision for versatile video coding. **Journal of Visual Communication and Image Representation**, China, v.71, p.102849, 2020.

CISCO. **Cisco Visual Networking Index**: Forecast and Trends, 2017–2022 White Paper. Acessado em 11/02/2020, `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html`.

CRISTINA, O. C.; MIHNEA, U. R.; IONUT, P. HEVC intra partitioning and mode decision using histograms of oriented gradients. In: IEEE INTERNATIONAL SYMPOSIUM ON ELECTRONICS AND TELECOMMUNICATIONS (ISETC), 2016., 2016. **Anais...** IEEE, 2016. p.277–280.

DA SILVA, T. L.; AGOSTINI, L. V.; SILVA CRUZ, L. A. da. Fast HEVC intra prediction mode decision based on EDGE direction information. In: PROCEEDINGS OF THE 20TH EUROPEAN SIGNAL PROCESSING CONFERENCE (EUSIPCO), 2012., 2012. **Anais...** IEEE, 2012. p.1214–1218.

DAEDE, T.; NORKIN, A.; BRAILOVKKIY, I. **Video Codec Testing and Quality Measurement**. Acessado em 03/07/2021, `https://tools.ietf.org/html/draft-ietf-netvc-testing-07`.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR'05), 2005., 2005. **Anais...** IEEE, 2005. v.1, p.886–893.

DE-LUXÁN-HERNÁNDEZ, S. et al. An intra subpartition coding mode for vvc. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2019., 2019. **Anais...** IEEE, 2019. p.1203–1207.

DUANMU, F.; MA, Z.; WANG, Y. Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension. **IEEE Journal on Emerging and Selected Topics in Circuits and Systems**, Nova Iorque, v.6, n.4, p.517–531, 2016.

FAN, Y. et al. A Fast QTMT Partition Decision Strategy for VVC Intra Prediction. **IEEE Access**, China, v.8, p.107900–107911, 2020.

FANG, C.-M.; CHANG, Y.-T.; CHUNG, W.-H. Fast intra mode decision for HEVC based on direction energy distribution. In: IEEE INTERNATIONAL SYMPOSIUM ON CONSUMER ELECTRONICS (ISCE), 2013., 2013. **Anais...** IEEE, 2013. p.61–62.

FU, T.; ZHANG, H.; MU, F.; CHEN, H. Fast CU Partitioning Algorithm for H. 266/VVC Intra-Frame Coding. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO (ICME), 2019., 2019. **Anais...** IEEE, 2019. p.55–60.

GWON, D.; CHOI, H.; YOUN, J. M. HEVC fast intra mode decision based on edge and SATD cost. In: ASIA PACIFIC CONFERENCE ON MULTIMEDIA AND BROADCASTING, 2015., 2015. **Anais...** IEEE, 2015. p.1–5.

HUANG, Y.; WANG, D.; SUN, Y.; HANG, B. A fast intra coding algorithm for HEVC by jointly utilizing naive Bayesian and SVM. **Multimedia Tools and Applications**, China, p.1–15, 2020.

ITU-T. **Subjective video quality assessment methods for multimedia applications**. Acessado em 03/07/2021, `https://www.itu.int/rec/T-REC-P.910-200804-I`.

JAMALI, M.; COULOMBE, S.; CARON, F. Fast HEVC intra mode decision based on edge detection and SATD costs classification. In: 2015 DATA COMPRESSION CONFERENCE, 2015. **Anais...** IEEE, 2015. p.43–52.

LEE, S.-H.; PARK, S.-H.; JANG, E. S. Fast intra prediction mode decision based on rough mode decision and most probable mode in HEVC. **Journal of Broadcast Engineering**, Coreia do Sul, v.19, n.2, p.158–165, 2014.

LEI, M. et al. Look-Ahead Prediction Based Coding Unit Size Pruning for VVC Intra Coding. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2019., 2019. **Anais...** IEEE, 2019. p.4120–4124.

LIU, X. et al. An adaptive mode decision algorithm based on video texture characteristics for HEVC intra prediction. **IEEE Transactions on Circuits and Systems for Video Technology**, China, v.27, n.8, p.1737–1748, 2016.

LIU, X. et al. An adaptive CU size decision algorithm for HEVC intra prediction based on complexity classification using machine learning. **IEEE Transactions on Circuits and Systems for Video Technology**, China, v.29, n.1, p.144–155, 2017.

LU, X.; YU, C.; JIN, X. A fast HEVC intra-coding algorithm based on texture homogeneity and spatio-temporal correlation. **EURASIP Journal on Advances in Signal Processing**, China, v.2018, n.1, p.37, 2018.

MERCAT, A.; VIITANEN, M.; VANNE, J. UVG dataset: 50/120fps 4K sequences for video codec analysis and development. In: ACM MULTIMEDIA SYSTEMS CONFERENCE, 11., 2020. **Proceedings...** IEEE, 2020. p.297–302.

NORVIG, P.; RUSSELL, S. **Inteligência artificial**: Tradução da 3a Edição. Rio de Janeiro: Elsevier Editora Ltda., 2014.

PAPKOV. **Decision Tree to CPP**. Acessado em 16/08/2021, `https://github.com/papkov/DecisionTreeToCpp`.

PARK, S.-H.; KANG, J.-W. Context-Based Ternary Tree Decision Method in Versatile Video Coding for Fast Intra Coding. **IEEE Access**, Coreia do Sul, v.7, p.172597–172605, 2019.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, [S.l.], v.12, p.2825–2830, 2011.

PENG, S.; PENG, Z.; REN, Y.; CHEN, F. Fast intra-frame coding algorithm for Versatile Video Coding based on texture feature. In: IEEE INTERNATIONAL CONFERENCE ON REAL-TIME COMPUTING AND ROBOTICS (RCAR), 2019., 2019. **Anais...** IEEE, 2019. p.65–68.

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...** IEEE, 2020. p.3119–3123.

SALDANHA, M.; SANCHEZ, G.; MARCON, C.; AGOSTINI, L. Fast Partitioning Decision Scheme for Versatile Video Coding Intra-Frame Prediction. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), 2020., 2020. **Anais...** IEEE, 2020. p.1–5.

SIQUEIRA, Í.; 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...** IEEE, 2020. p.1–4.

SULLIVAN, G. J.; WIEGAND, T. Rate-distortion optimization for video compression. **IEEE signal processing magazine**, [S.l.], v.15, n.6, p.74–90, 1998.

SUN, X. et al. Fast CU size and prediction mode decision algorithm for HEVC based on direction variance. **Journal of Real-Time Image Processing**, China, v.16, n.5, p.1731–1744, 2019.

TANG, N. et al. Fast CTU Partition Decision Algorithm for VVC Intra and Inter Coding. In: IEEE ASIA PACIFIC CONFERENCE ON CIRCUITS AND SYSTEMS (APCCAS), 2019., 2019. **Anais...** IEEE, 2019. p.361–364.

TISSIER, A. et al. Complexity Reduction Opportunities in the Future VVC Intra Encoder. In: IEEE 21ST INTERNATIONAL WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING (MMSP), 2019., 2019. **Anais...** IEEE, 2019. p.1–6.

WANG, X.; XUE, Y. Fast HEVC intra coding algorithm based on Otsu's method and gradient. In: IEEE INTERNATIONAL SYMPOSIUM ON BROADBAND MULTIMEDIA SYSTEMS AND BROADCASTING (BMSB), 2016., 2016. **Anais...** IEEE, 2016. p.1–5.

WATSON, A. **Number of paid Netflix subscribers worldwide in the 3rd quarter of 2019, by region**. Acessado em 11/02/2020, `https://www.statista.com/statistics/483112/netflix-subscribers/`.

YANG, H. et al. Low complexity CTU partition structure decision and fast intra mode decision for versatile video coding. **IEEE Transactions on Circuits and Systems for Video Technology**, China, 2019.

YANG, W.; PI, Y. A low-complexity intra prediction mode selection algorithm in HEVC. In: IEEE 14TH INTL CONF ON DEPENDABLE, AUTONOMIC AND SECURE COMPUTING, 14TH INTL CONF ON PERVASIVE INTELLIGENCE AND COMPUTING, 2ND INTL CONF ON BIG DATA INTELLIGENCE AND COMPUTING AND CYBER SCIENCE AND TECHNOLOGY CONGRESS (DASC/PICOM/DATACOM/CYBERSCITECH), 2016., 2016. **Anais...** IEEE, 2016. p.598–603.

YOUTUBE. **YouTube para a imprensa**. Acessado em 11/02/2020, `https://www.youtube.com/intl/pt-BR/about/press/`.

ZHANG, M.; ZHAO, C.; XU, J. An adaptive fast intra mode decision in HEVC. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2012., 2012. **Anais...** IEEE, 2012. p.221–224.

ZHANG, Q.; WANG, Y.; HUANG, L.; JIANG, B. Fast CU partition and intra mode decision method for H. 266/VVC. **IEEE Access**, China, v.8, p.117539–117550, 2020.

ZHANG, T.; SUN, M.-T.; ZHAO, D.; GAO, W. Fast intra-mode and CU size decision for HEVC. **IEEE Transactions on Circuits and Systems for Video Technology**, China, v.27, n.8, p.1714–1726, 2016.

ZHANG, Y.; KWONG, S.; WANG, S. Machine learning based video coding optimizations: A survey. **Information Sciences**, China, v.506, p.395–423, 2020.

ZOUIDI, N.; BELGHITH, F.; KESSENTINI, A.; MASMOUDI, N. Fast intra prediction decision algorithm for the QTBT structure. In: IEEE INTERNATIONAL CONFERENCE ON DESIGN & TEST OF INTEGRATED MICRO & NANO-SYSTEMS (DTS), 2019., 2019. **Anais...** IEEE, 2019. p.1–6.