Fast Partition Mode Decision via a Plug-in Fully Connected Network for Video Coding

Jiaqi Zhang, Meng Wang, Chuanmin Jia,
Qi Wang, Shanshe Wang, Siwei Ma and Wen Gao

Institute of Computing Technology, Chinese Academy of Sciences,
University of Chinese Academy of Sciences,
City University of Hong Kong,
Peking University.

Contents

Introduction

- ECM is under development
  - Adopts DIMD, TIMD, MMLM…
  - Nearly 7% and 14% bit-rate saving under AI and RA[1]
  - Encoding complexity dramatically increased

	All Intra Main10				
	Y	U	V	EncT	DecT
Class A1	-6.76%	-10.85%	-12.55%	306%	235%
Class A2	-6.43%	-9.83%	-6.78%	294%	226%
Class B	-5.92%	-9.95%	-11.25%	337%	248%
Class C	-6.73%	-8.79%	-9.19%	329%	243%
Class E	-7.23%	-9.70%	-9.20%	329%	286%
Overall	-6.54%	-9.78%	-9.92%	321%	247%
Class D	-5.70%	-7.02%	-6.59%	332%	256%
Class F	-10.50%	-13.32%	-14.04%	244%	285%

ECM-2.0 over VTM-11.0 AI

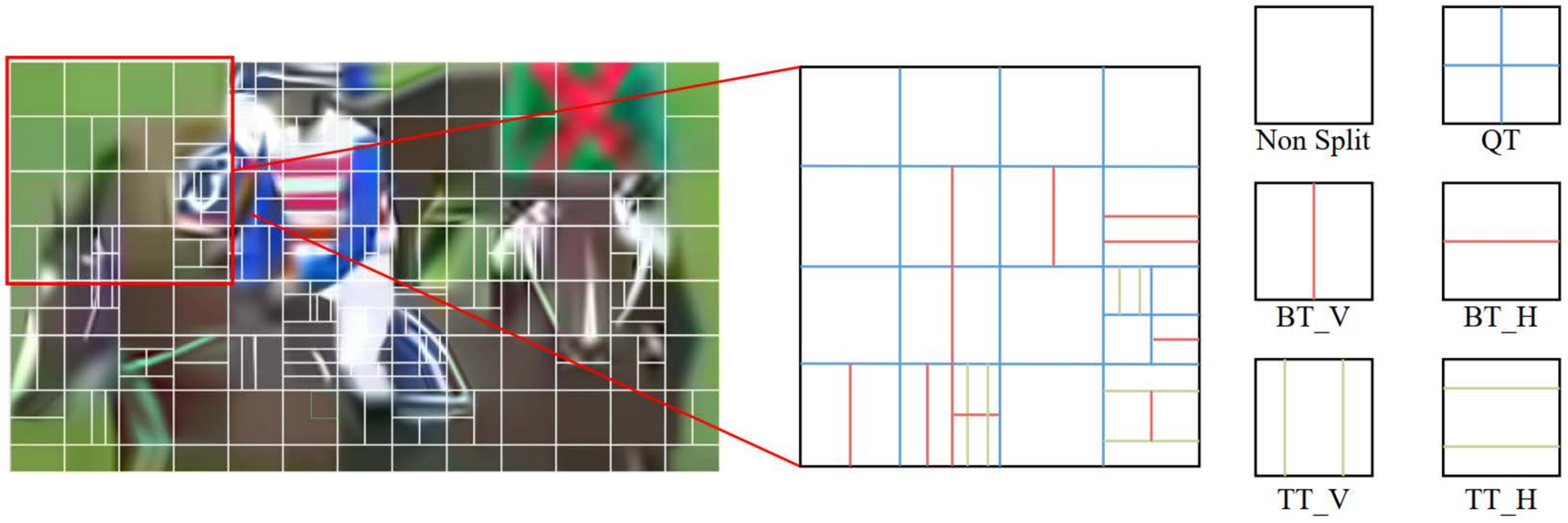
	Random Access Main 10				
	Y	U	V	EncT	DecT
Class A1	-13.50%	-15.91%	-20.31%	342%	504%
Class A2	-14.37%	-17.39%	-16.47%	321%	584%
Class B	-12.47%	-17.52%	-17.43%	355%	548%
Class C	-14.37%	-16.46%	-16.52%	351%	488%
Class E					
Overall	-13.56%	-16.89%	-17.57%	345%	529%
Class D	-15.35%	-16.36%	-15.88%	358%	530%
Class F	-13.20%	-16.71%	-16.88%	319%	438%

ECM-2.0 over VTM-11.0 RA

[1]: Martak Karczewicz, Yan Ye, Li Zhang, Benjamin Bross, and Xiang Li, \JVET AHG report: Enhanced compression beyond VVC capability (AHG12)," in JVET, 24th Meeting, Document JVET-X0012. Teleconference: ITU-T, ISO/IEC, 2021.

Introduction

- QTMT partition structure
  - Extend CTU size and the maximum transform unit size are extended to 256×256
  - Maximum intra coding block is set as 128×128

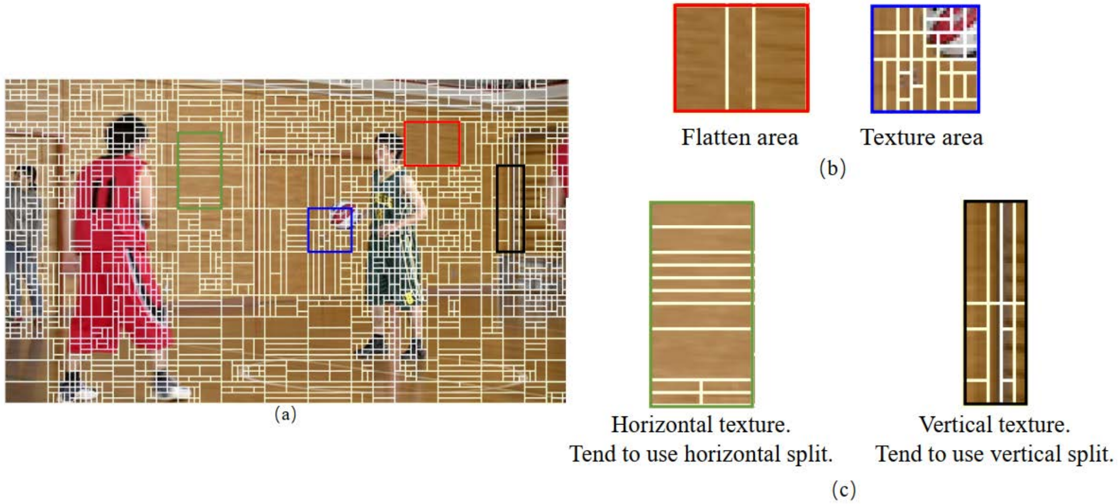


The Proposed Method

- Learning-based approach - fully connected network
  - Shallow architecture with only one hidden layer
  - Selected features are easy to acquire
  - Easily integrated into the video codec, extricating from the interfaces or platforms

Feature Extraction

- The texture information
  - Texture information is highly related to CU partition structure
    - CU is prone to choose simple and large structure in flatten area. More fine-grain CU partition structure is preferred in the complex
    - Directional texture is beneficial for choosing the partition direction



Feature Extraction

- The texture information
  - Four commonly used directional gradients

$$G_h = \sum_{i=0}^{W-1}\sum_{j=0}^{H-1} \boldsymbol{P} \times A_h, \qquad G_v = \sum_{i=0}^{W-1}\sum_{j=0}^{H-1} \boldsymbol{P} \times A_v,$$

$$G_{45^\circ} = \sum_{i=0}^{W-1}\sum_{j=0}^{H-1} \boldsymbol{P} \times A_{45^\circ}, \qquad G_{135^\circ} = \sum_{i=0}^{W-1}\sum_{j=0}^{H-1} \boldsymbol{P} \times A_{135^\circ},$$

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

(a) A_h

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(b) A_v

$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

(c) A_{45°

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(d) A_{135°

  - $GR_0, GR_1, and\ \ GR_2$:

$$(GR_0, GR_1, GR_2) = \begin{cases} (G_h/G_v, G_h/G_{45^\circ}, G_h/G_{135^\circ}), & \mathcal{M} \in \{BT\_H, TT\_H\}, \\ (G_v/G_h, G_v/G_{45^\circ}, G_v/G_{135^\circ}), & \mathcal{M} \in \{BT\_V, TT\_V\}. \end{cases}$$

Feature Extraction

- The CU dimension

$$CSR = \begin{cases} \dfrac{H}{W+H}, & \mathcal{M} \in \{BT\_H, TT\_H\}, \\ \dfrac{W}{W+H}, & \mathcal{M} \in \{BT\_V, TT\_V\}, \end{cases}$$

- Intermediate coding information

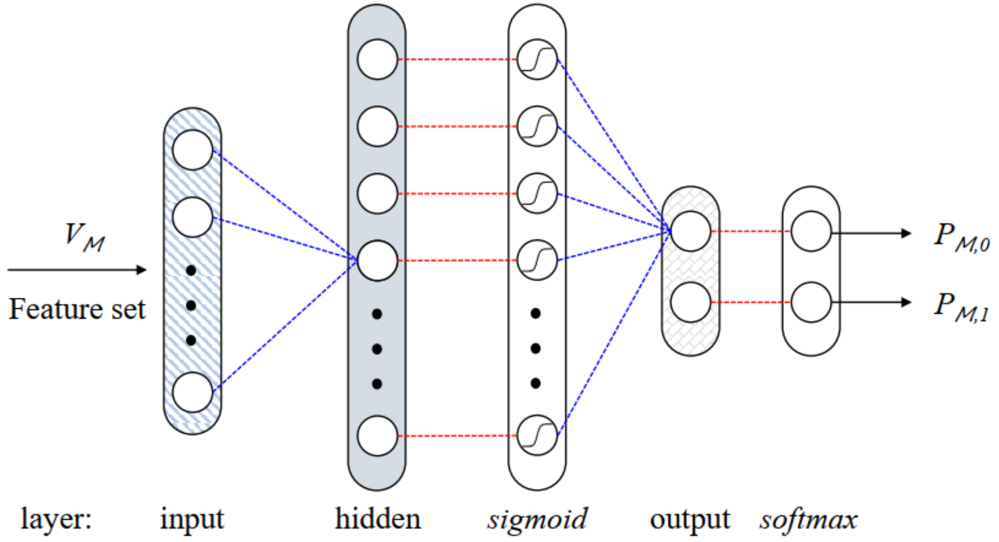
$$\mathcal{S} = \begin{cases} 1, & if\ (R_Q > R_{BT\_V} || R_Q > R_{BT\_H}), \\ 0, & \text{otherwise}, \end{cases} \qquad \mathcal{D} = \begin{cases} 1, & if\ (R_{BT\_H} > R_{BT\_V}), \\ 0, & \text{otherwise}. \end{cases}$$

- Feature set

$$\mathcal{V}_{\mathcal{M}} = \begin{cases} (GR_0, GR_1, GR_2, CSR), & \mathcal{M} \in \{BT\_H, BT\_V\}, \\ (GR_0, GR_1, GR_2, CSR, \mathcal{S}, \mathcal{D}), & \mathcal{M} \in \{TT\_H, TT\_V\}. \end{cases}$$

The Proposed Method

- FCN Model Architecture
  - Four models are designed for BT and TT
  - Only one hidden layer, 30 neuron nodes
  - Non-linear activation function: *Sigmoid* and *Softmax* for hidden layer and output layer



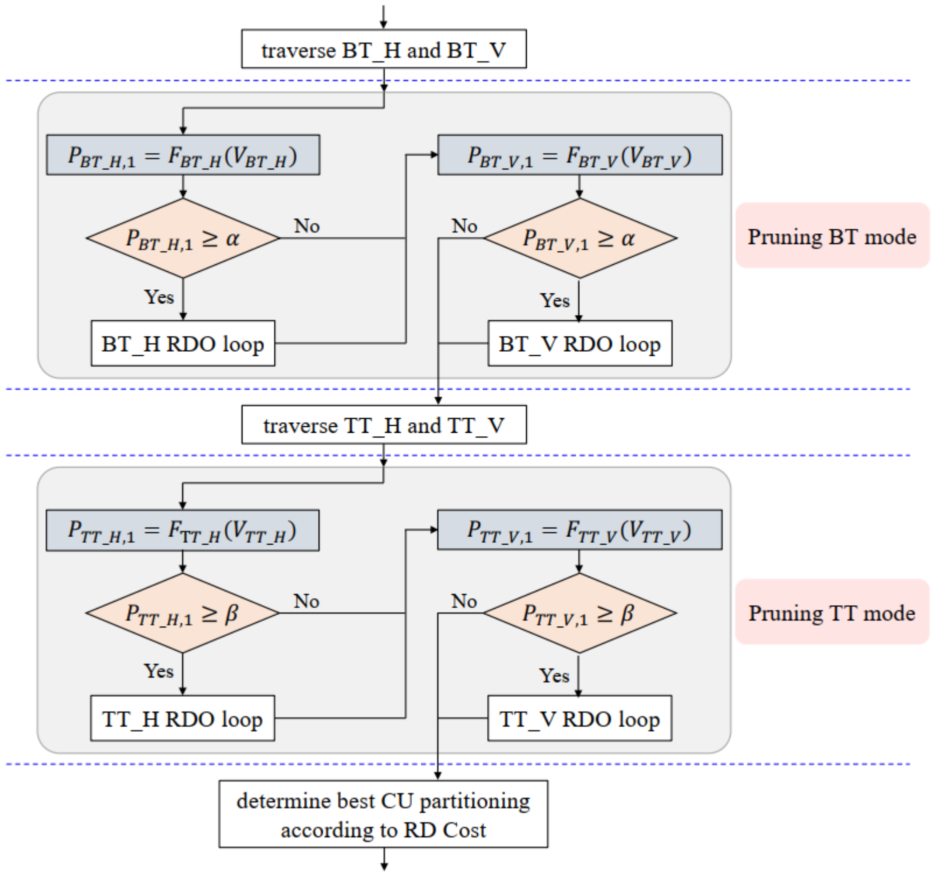
The Proposed Method

- Limited memory cost

Model Type	Number of neurons in each layer			Number of parameter (N_P)	Memory (KB)
	1 (input)	2 (hidden)	3 (output)		
$\mathcal{F}_{\mathcal{BT}\_\mathcal{H}}$	4	30	2	212	0.828
$\mathcal{F}_{\mathcal{BT}\_\mathcal{V}}$	4	30	2	212	0.828
$\mathcal{F}_{\mathcal{TT}\_\mathcal{H}}$	6	30	2	272	1.06
$\mathcal{F}_{\mathcal{TT}\_\mathcal{V}}$	6	30	2	272	1.06

The Proposed Method

- Working flow
  - Two pre-defined threshold α and β for BT and TT



Training and Implementation

- Training with Pytorch[1]
  - Dataset: BVI-DVC
  - Loss function: Cross-entropy-loss
  - Learning rate: $2 * 10^{-5}$
  - ADAM optimizer
- Implementation
  - Collect the weight and bias matrix of the optimal FCN model
  - The implementation in video codec conforms to C++ standard format without deep learning library interface

[1]: https://pytorch.org/

Experimental Results

- Tunable computational complexity reduction
  - Achieving 14.62%~50.39% time savings
  - Better performance on 4K

Class	Sequence	$\mathcal{C}_1$		$\mathcal{C}_2$		$\mathcal{C}_3$	
		BD-BR	*TS*	BD-BR	*TS*	BD-BR	*TS*
A1 3840×2160	Tango2	-0.10%	17.55%	0.19%	32.76%	0.30%	44.17%
	FoodMarket4	0.21%	9.64%	0.03%	24.21%	0.16%	34.26%
	Campfire	0.31%	21.32%	0.65%	36.19%	1.68%	52.69%
A2 3840×2160	CatRobot	0.26%	19.88%	0.69%	39.14%	1.23%	48.26%
	DaylightRoad2	0.30%	19.40%	0.95%	35.33%	1.49%	49.62%
	ParkRunning3	0.02%	9.11%	0.13%	29.09%	0.36%	51.82%
B 1920×1080	MarketPlace	0.02%	6.88%	0.33%	30.92%	0.90%	51.97%
	RitualDance	0.35%	9.07%	0.58%	23.81%	2.71%	45.58%
	Cactus	0.27%	13.46%	0.66%	31.46%	1.43%	52.50%
	BasketballDrive	0.18%	13.41%	0.49%	27.55%	1.12%	49.97%
	BQTerrace	0.38%	12.52%	0.62%	28.00%	1.18%	50.78%
C 832×480	BasketballDrill	0.12%	25.32%	1.02%	36.88%	3.39%	55.62%
	BQMall	-0.03%	16.22%	0.51%	31.00%	1.80%	53.53%
	PartyScene	0.00%	17.97%	0.38%	33.00%	1.06%	55.20%
	RaceHorses	0.19%	17.30%	0.68%	35.88%	1.42%	55.58%
D 416×240	BasketballPass	0.32%	13.29%	0.72%	24.61%	2.09%	48.38%
	BQSquare	0.12%	14.42%	0.31%	31.39%	1.36%	50.65%
	BlowingBubbles	0.32%	20.93%	0.77%	33.70%	1.40%	56.37%
	RaceHorses	0.00%	13.39%	0.53%	32.00%	1.41%	51.70%
E 1280×720	FourPeople	0.09%	9.92%	0.53%	26.03%	1.75%	51.12%
	Johnny	-0.26%	9.49%	0.18%	25.54%	1.61%	49.83%
	KristenAndSara	0.23%	11.05%	0.71%	24.22%	1.84%	49.05%
Average		0.15%	14.62%	0.53%	30.58%	1.44%	50.39%
Average(A1,A2)		**0.17%**	**16.15%**	**0.44%**	**32.79%**	**0.87%**	**46.80%**

Conclusion

- A partition mode pruning method based on fully connected network is proposed
  - By jointly utilizing local texture information and intermediate coding information
  - FCN models are performed as a plug-in module to eliminate the unnecessarily attempted partition modes
- Tunable computational complexity reduction, 15%～50%, can be achieved

[1]: https://pytorch.org/

Thanks!