LEARNING-BASED COMPLEXITY REDUCTION AND SCALING FOR HEVC ENCODERS

Mateus Grellert
Guilherme Corrêa
Bruno Zatt
Sergio Bampi
Luis A. da Silva Cruz

April, 2018

Summary

- Introduction and Motivation
- Problem Definition: HEVC Mode Decision Complexity
- Proposed Method: Coding Unit Split Decision Using Decision Trees
- Results and Discussion
- Conclusions

Summary

- Introduction and Motivation
- Problem Definition: HEVC Mode Decision Complexity
- Proposed Method: Coding Unit Split Decision Using Decision Trees
- Results and Discussion
- Conclusions

Introduction

- Video coding is present in a variety of applications

- To cope with these demands, many products that implement video encoders through hardware and/or software are available on the market


Motivation

- Latest encoder projects:
  - **H.265/HEVC, from ITU-T and ISO/IEC**
  - VP9, from Google
  - AV1, from AOM
- Capable of achieving excellent compression ratios compared to their predecessors

Problem:

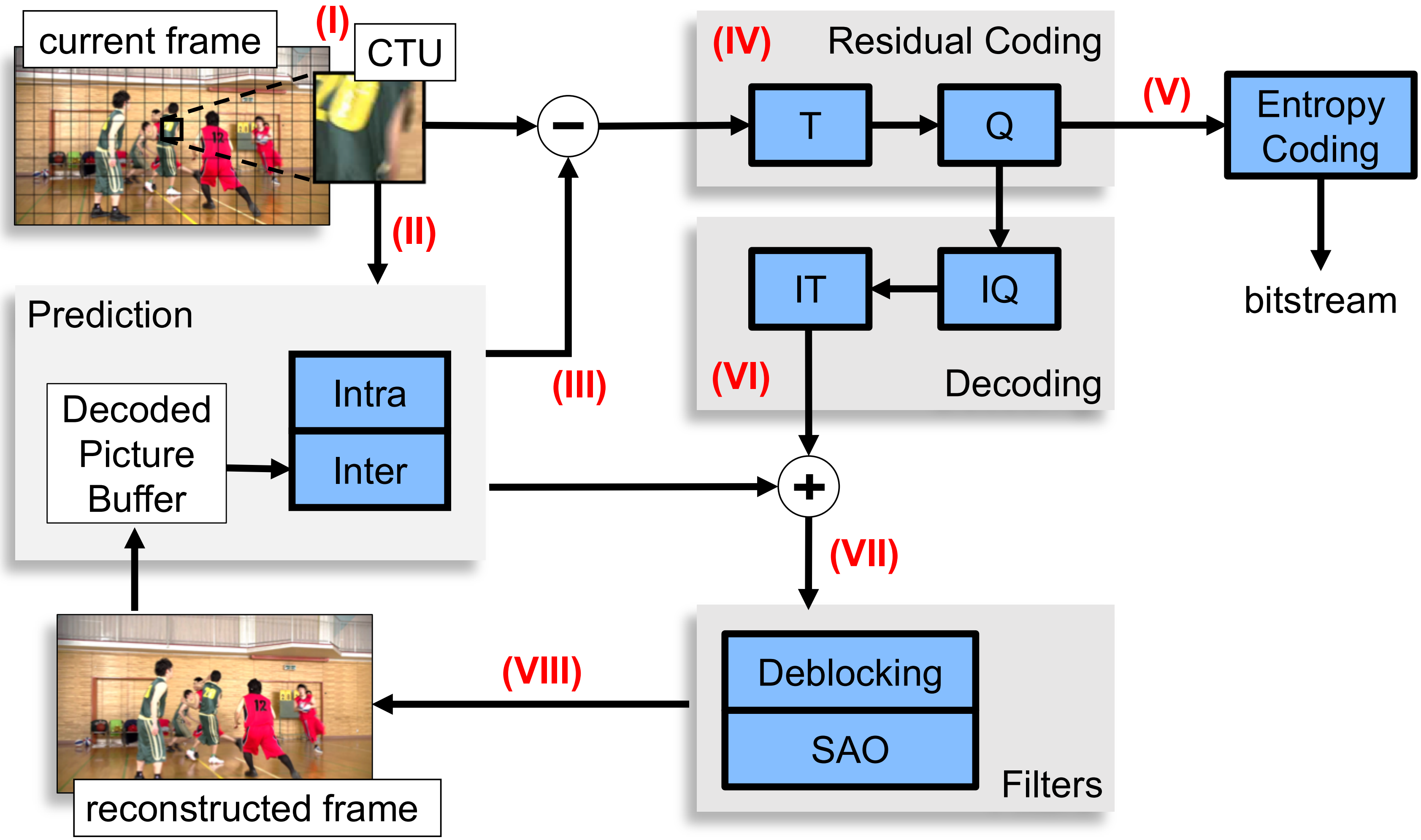
- HEVC-compliant encoders require a lot of computing to produce the bitstreams

- Complexity reduction methods are of utmost importance!

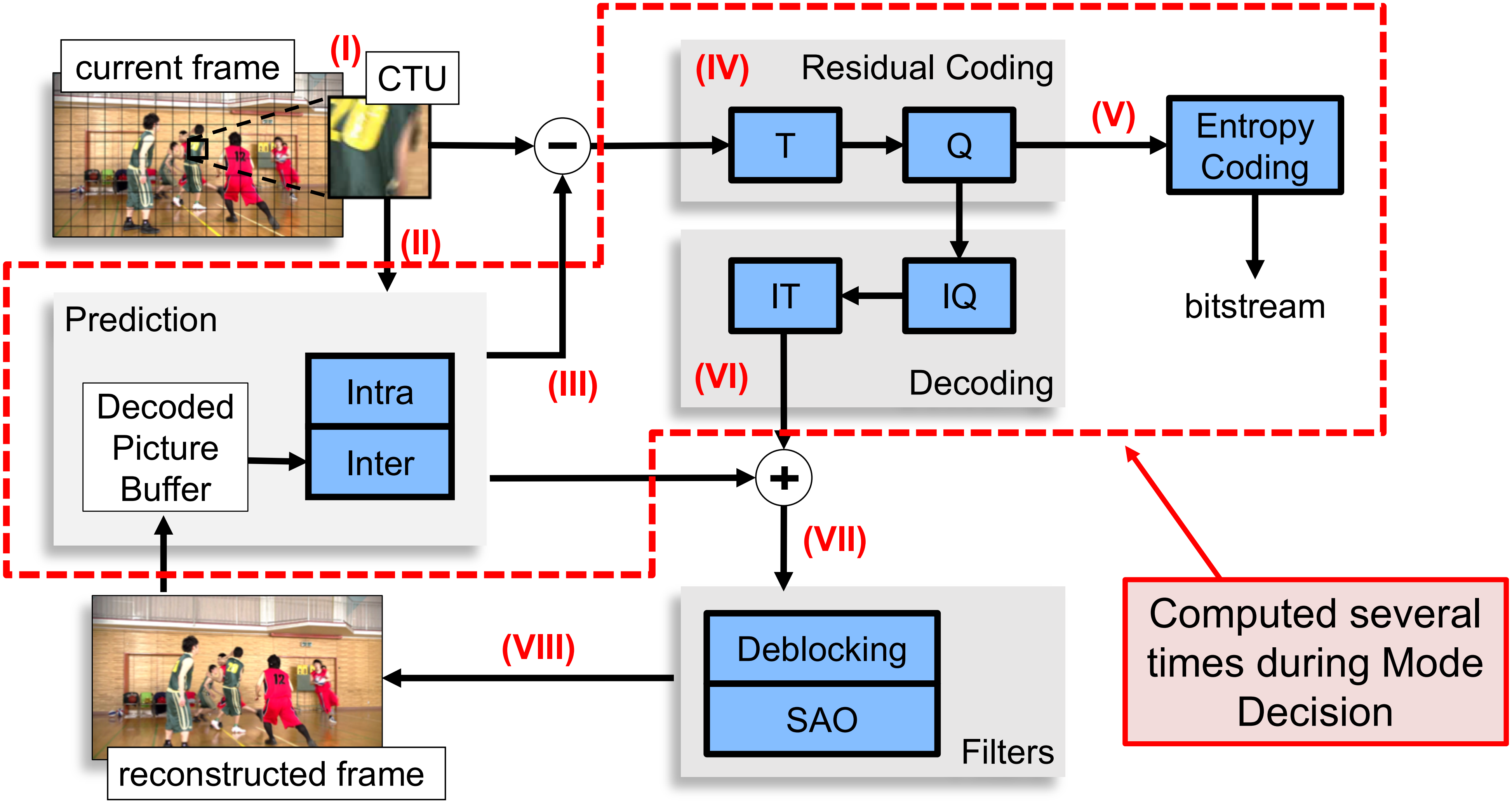
Summary

- Introduction and Motivation
- Problem Definition: HEVC Mode Decision Complexity
- Proposed Method: Coding Unit Split Decision Using Decision Trees
- Results and Discussion
- Conclusions

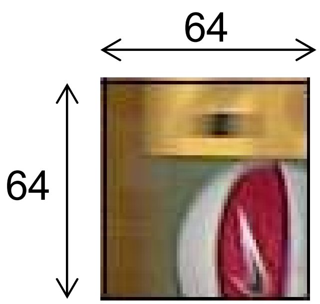
HEVC Encoding Loop

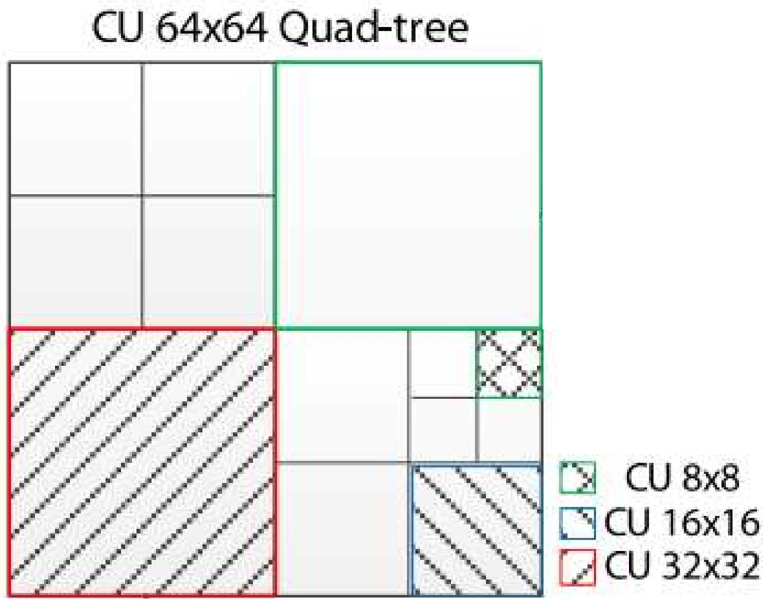


HEVC Encoding Loop

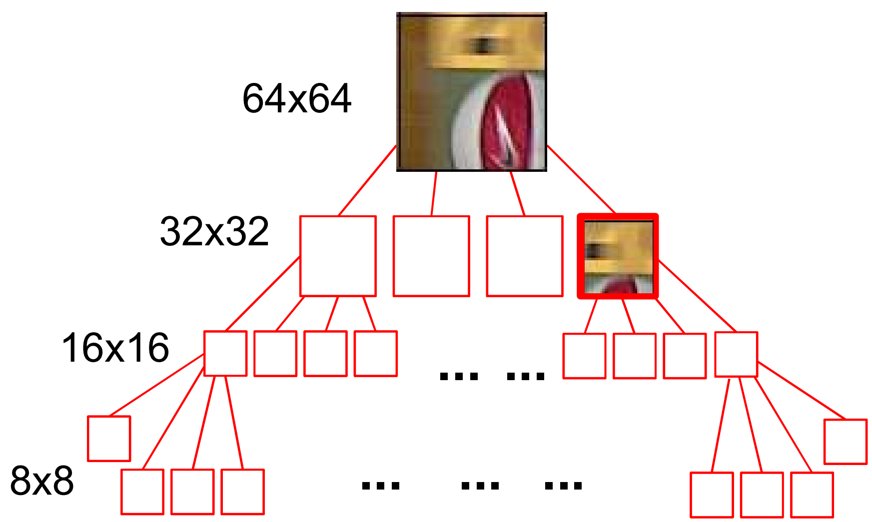


RDO-based Mode Decision

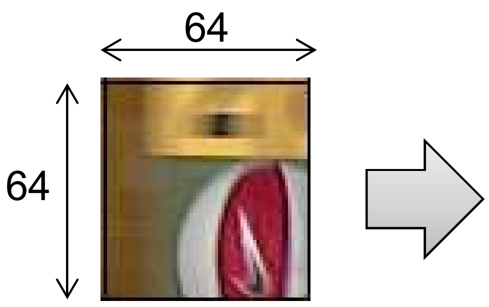




1 – Each 64x64 block can be partitioned into 4 smaller blocks recursively, forming a **CU** quad-tree

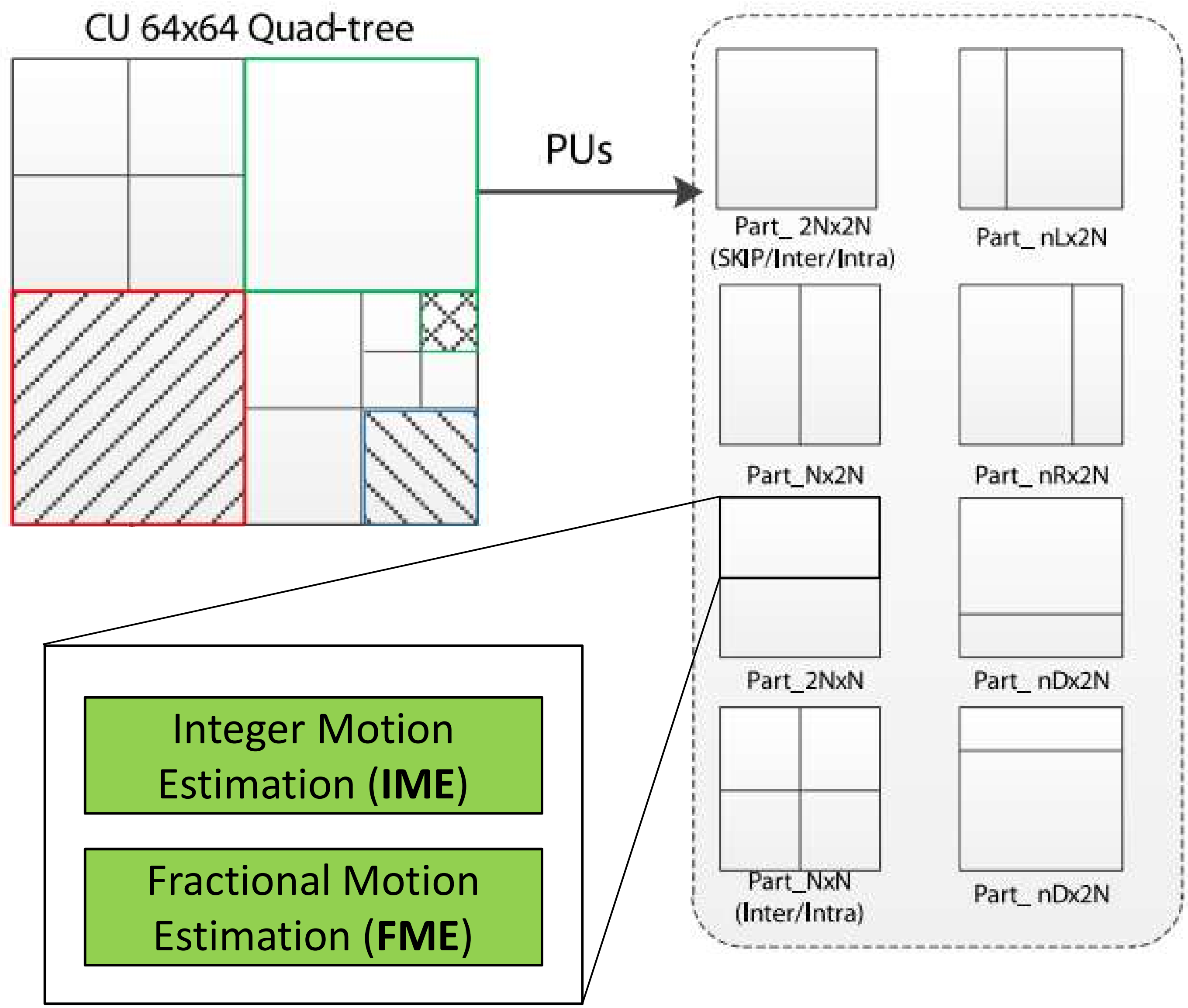


RDO-based Mode Decision

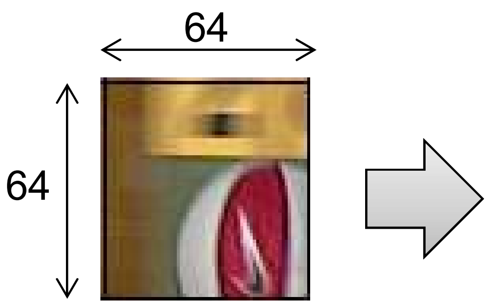


1 – Each 64x64 block can be partitioned into 4 smaller blocks recursively, forming a **CU** quad-tree

2 – Each CU can assume 8 different paritionings, called Prediction Units (**PU**), where ME occurs



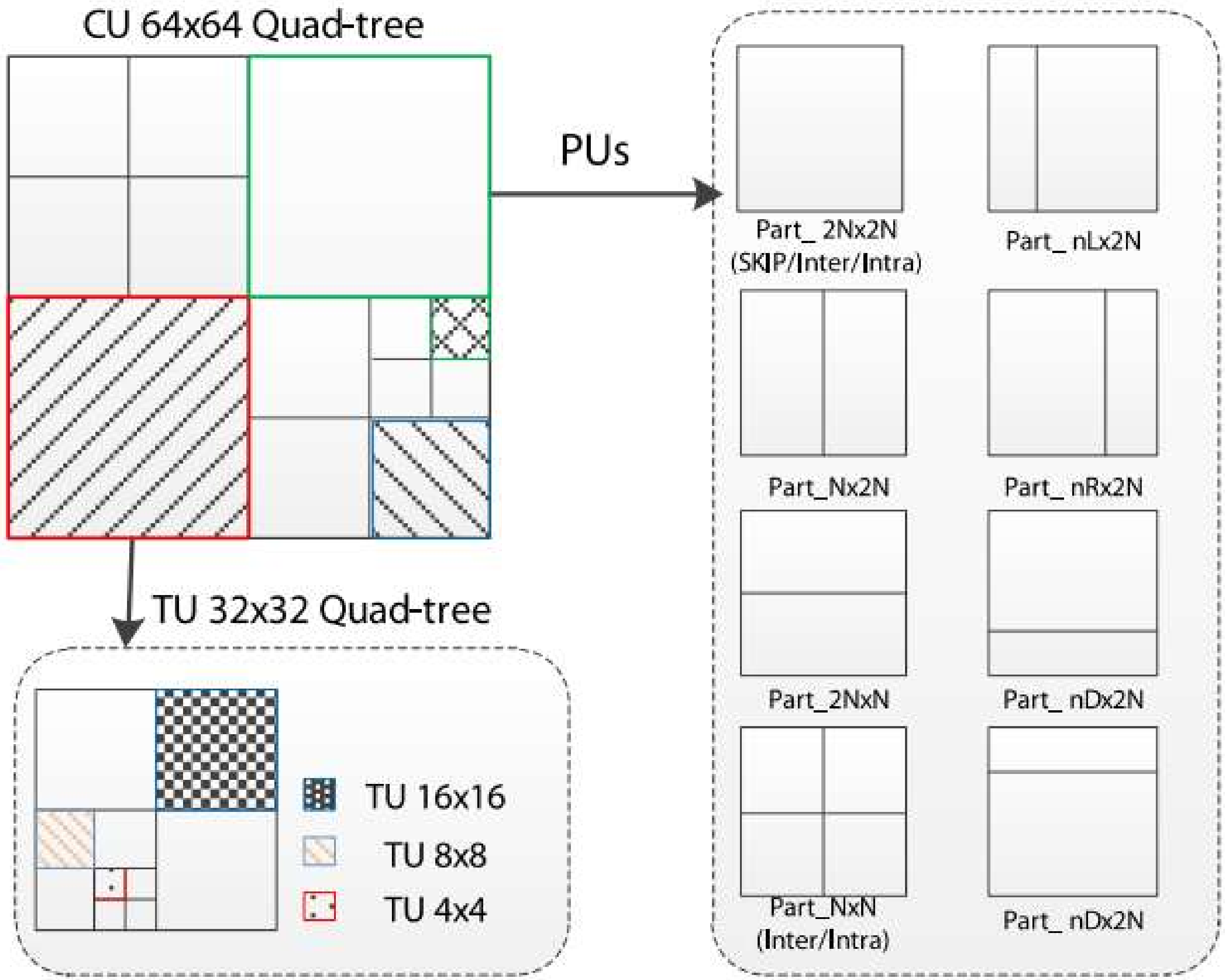
RDO-based Mode Decision



1 – Each 64x64 block can be partitioned into 4 smaller blocks recursively, forming a **CU** quad-tree

2 – Each CU can assume 8 different paritionings, called Prediction Units (**PU**), where ME occurs

3 – The best mode of each PU goes to the transforms stage. The Transform Units(**TUs**) can also be recursively partitioned (TU quad-tree)



Source: Shen, X.; Yu, L. “CU splitting early termination based on weighted SVM”, EURASIP JIVP, 2013

HEVC Mode Decision Complexity

- The exhaustive mode decision evaluates several options, while only some of them are used to encode the pixel data in a CU

- Most decisions are nested inside each node of the CU quad-tree decision

- **Simplifying the CU decision process has a significant impact in the overall encoding complexity (proposed method)**

Summary

- Introduction and Motivation
- Problem Definition: HEVC Mode Decision Complexity
- Proposed Method: Coding Unit Split Decision Using Decision Trees
- Results and Discussion
- Conclusions

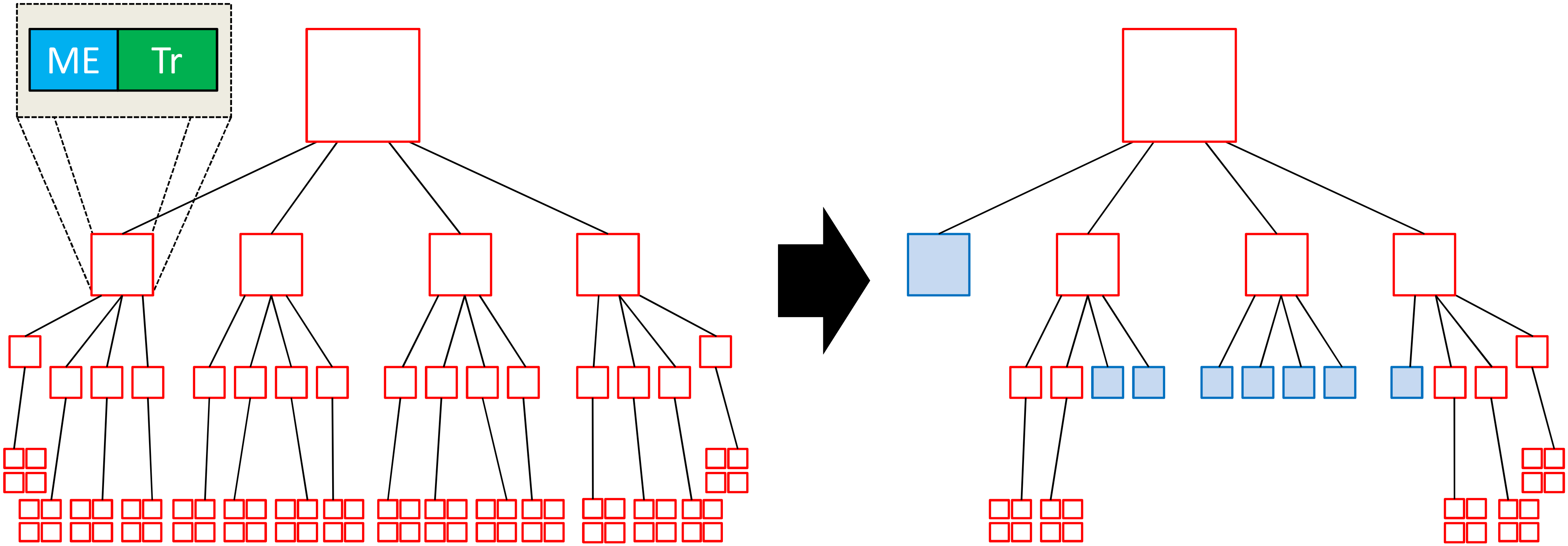
Proposed Method

- The proposed method alleviates the CU split decision process through an early termination of the CU quad-tree analysis

- The early termination is decided by learning-based models using Decision Trees

Proposed Method

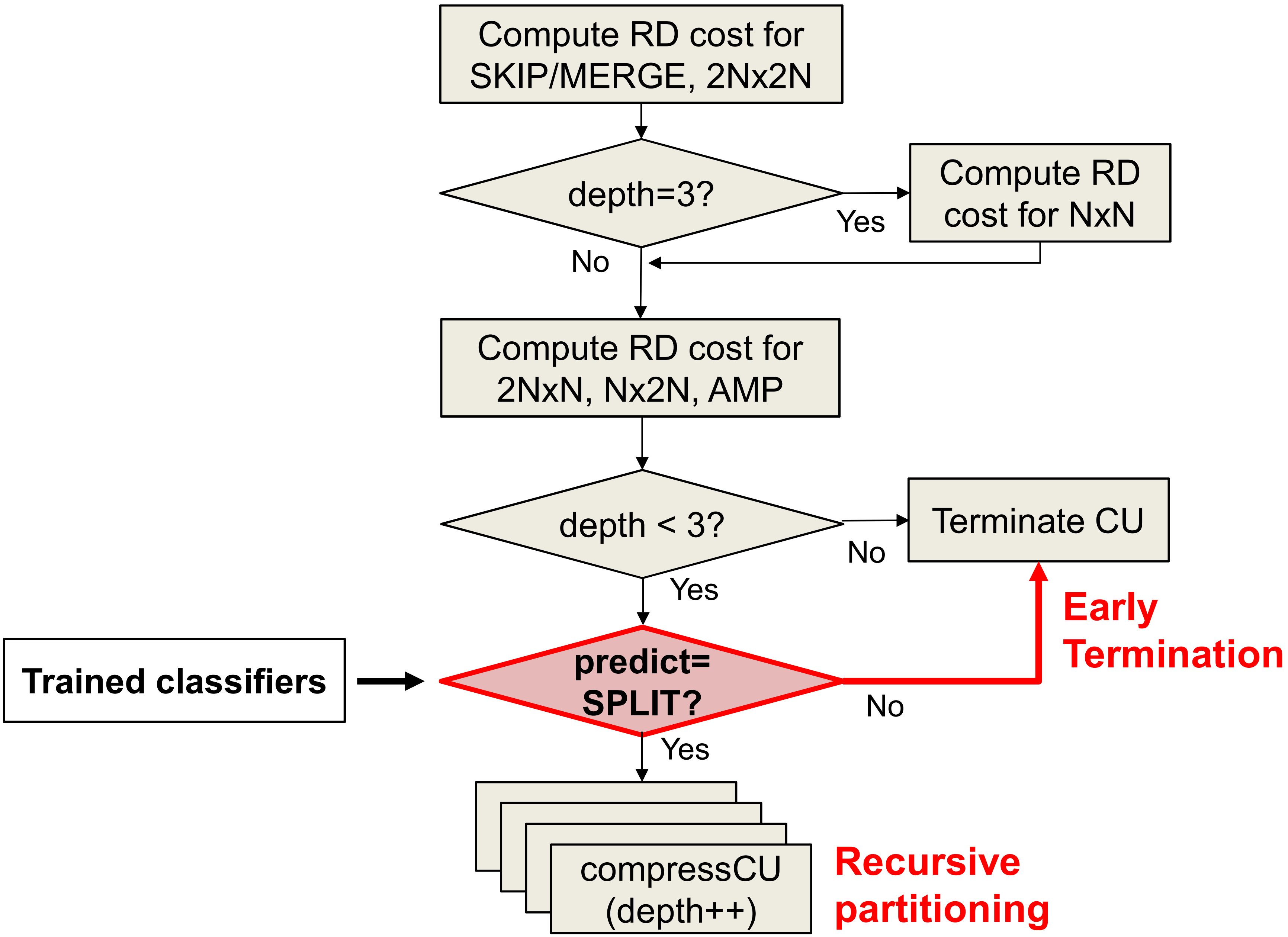
- How do early terminations reduce encoding time?



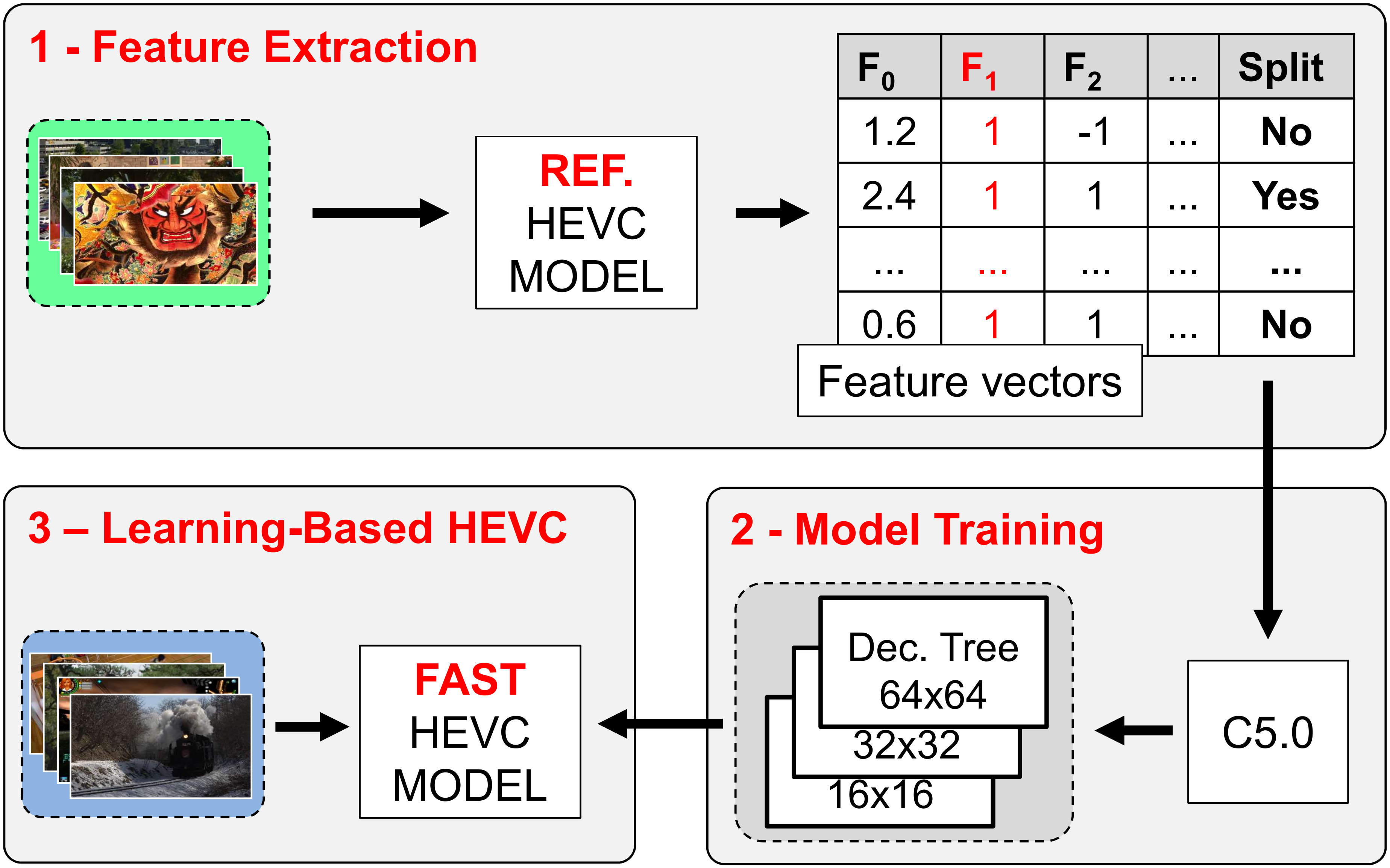
Exhausitve CU quad-tree

Learning-based CU quad-tree

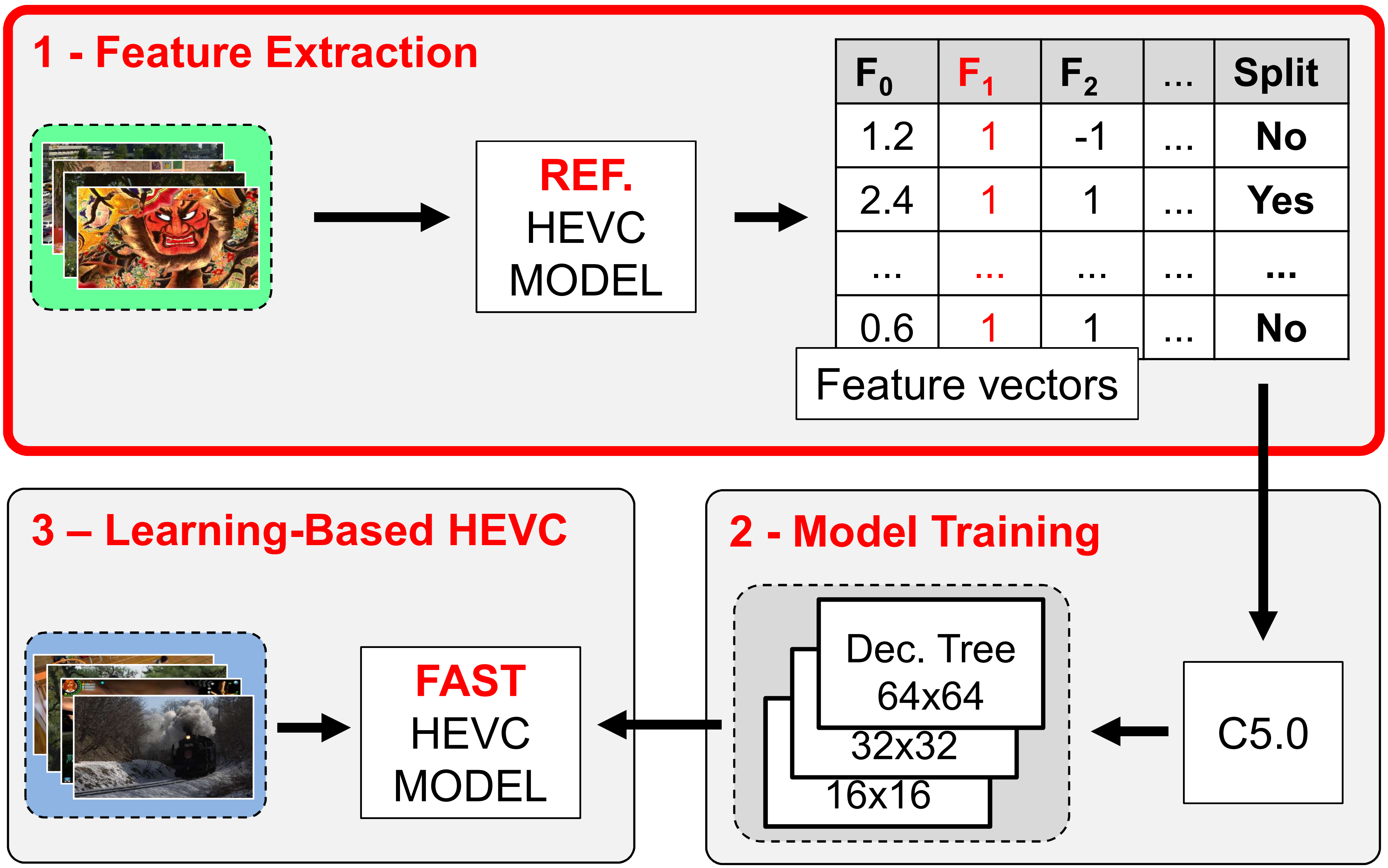
Proposed Method



Methodology



Methodology



Feature Extraction

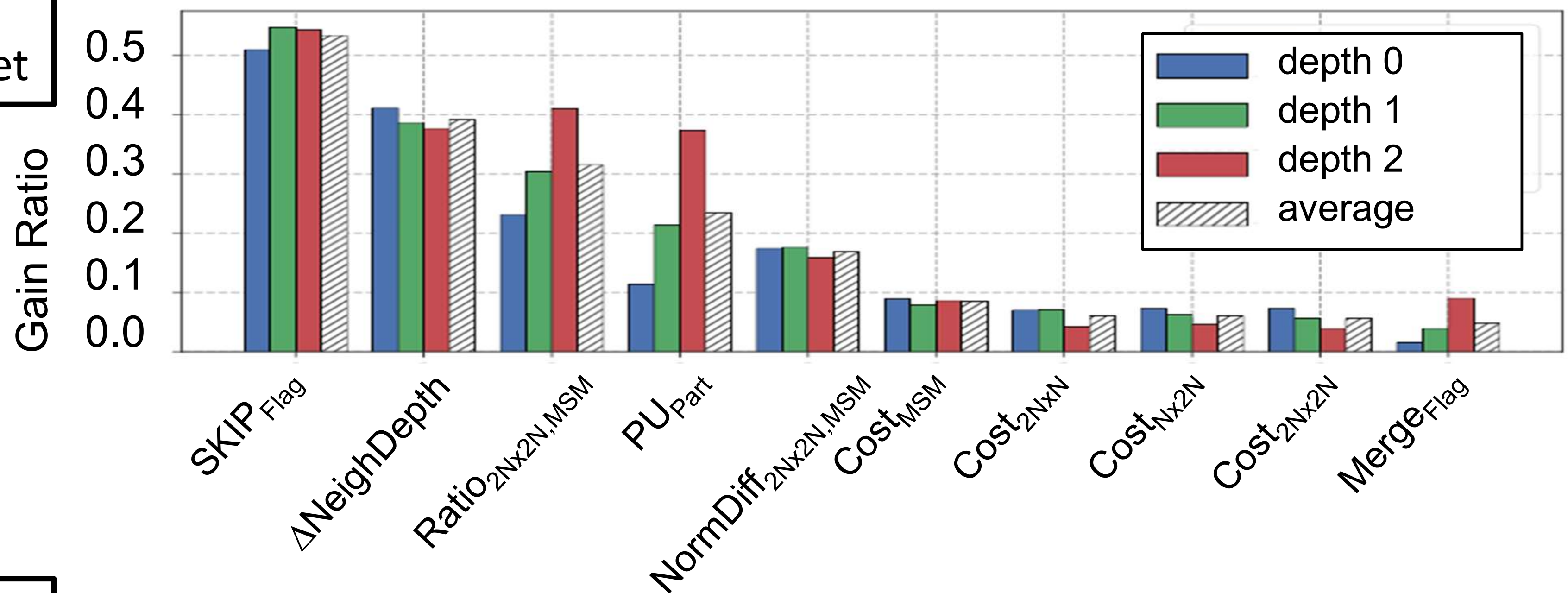
- The set of features used in this work contains the ones used in Corrêa's [1] Decision Trees, but also new features introduced in this work

- The following list contains a subset of these new features:
  - Best prediction mode (inter, intra, SKIP)
  - Number of encoded bits
  - Total distortion obtained with the predicted CU
  - Coded Block Flag

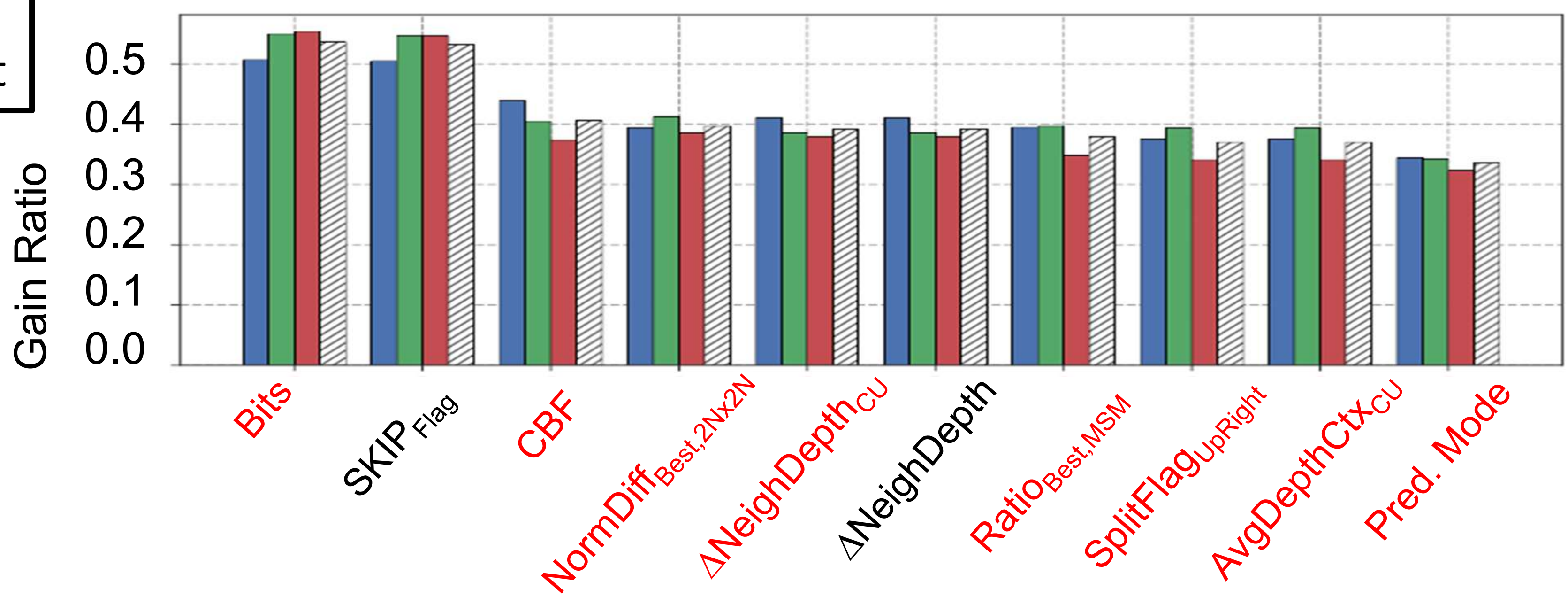
[1] G. Correa, P. A. Assuncao, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC encoding decisions using data mining," IEEE Transactions on Circuits and Systems for Video Technology, 2015.

Feature Set Comparison

Corrêa's Feature Set

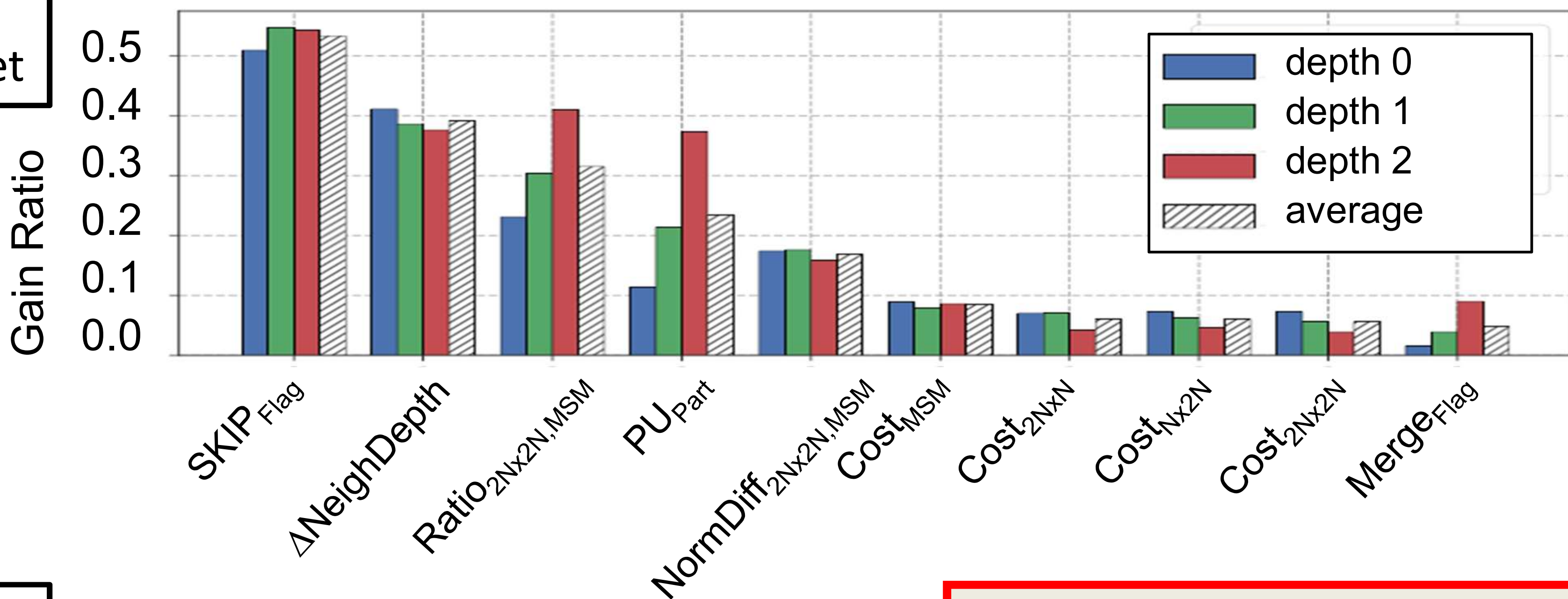


New Feature Set

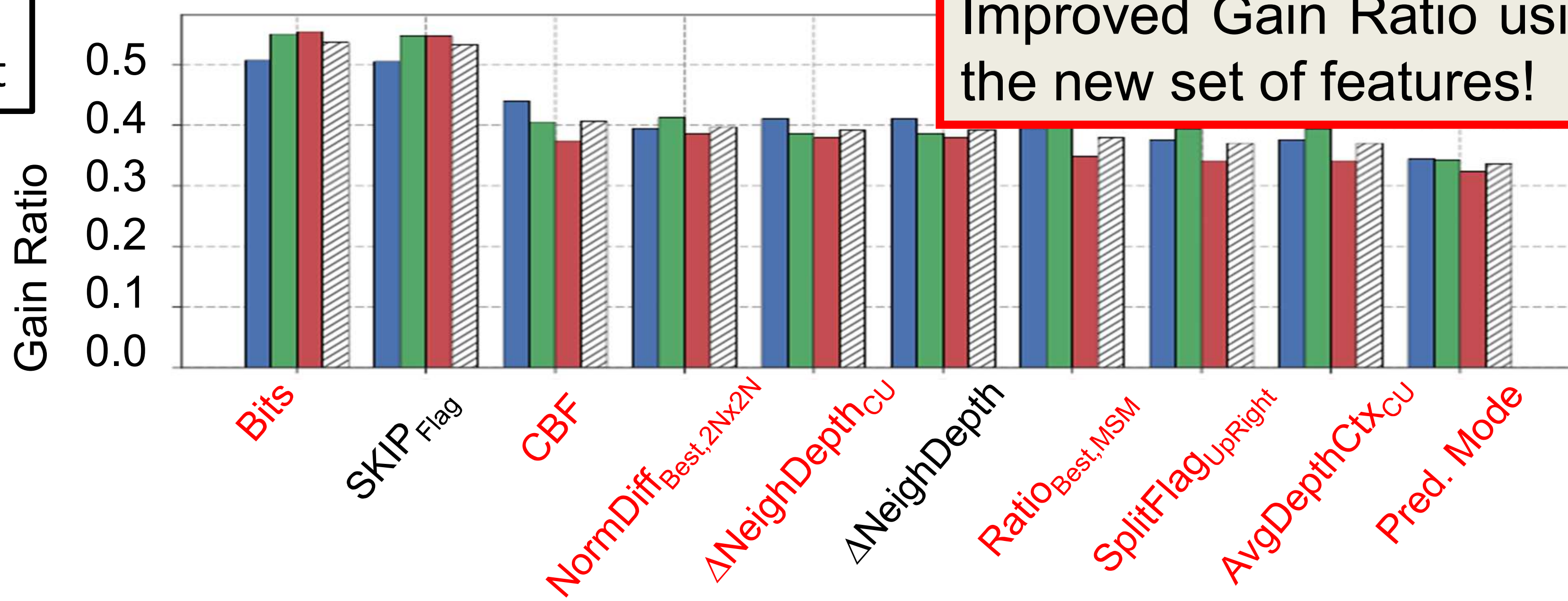


Feature Set Comparison

Corrêa's Feature Set



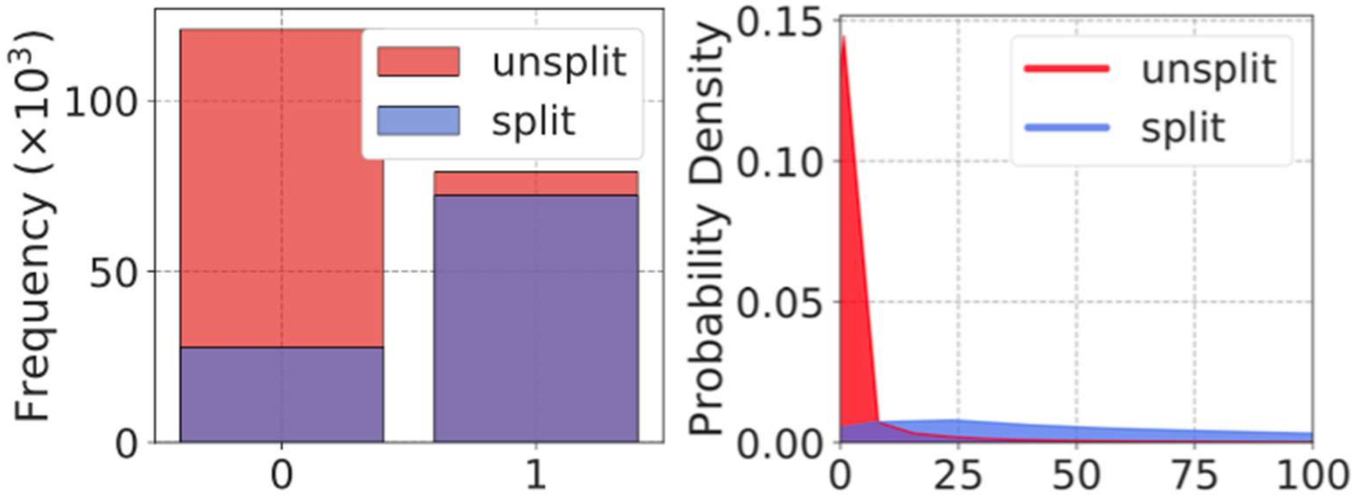
New Feature Set



Improved Gain Ratio using the new set of features!

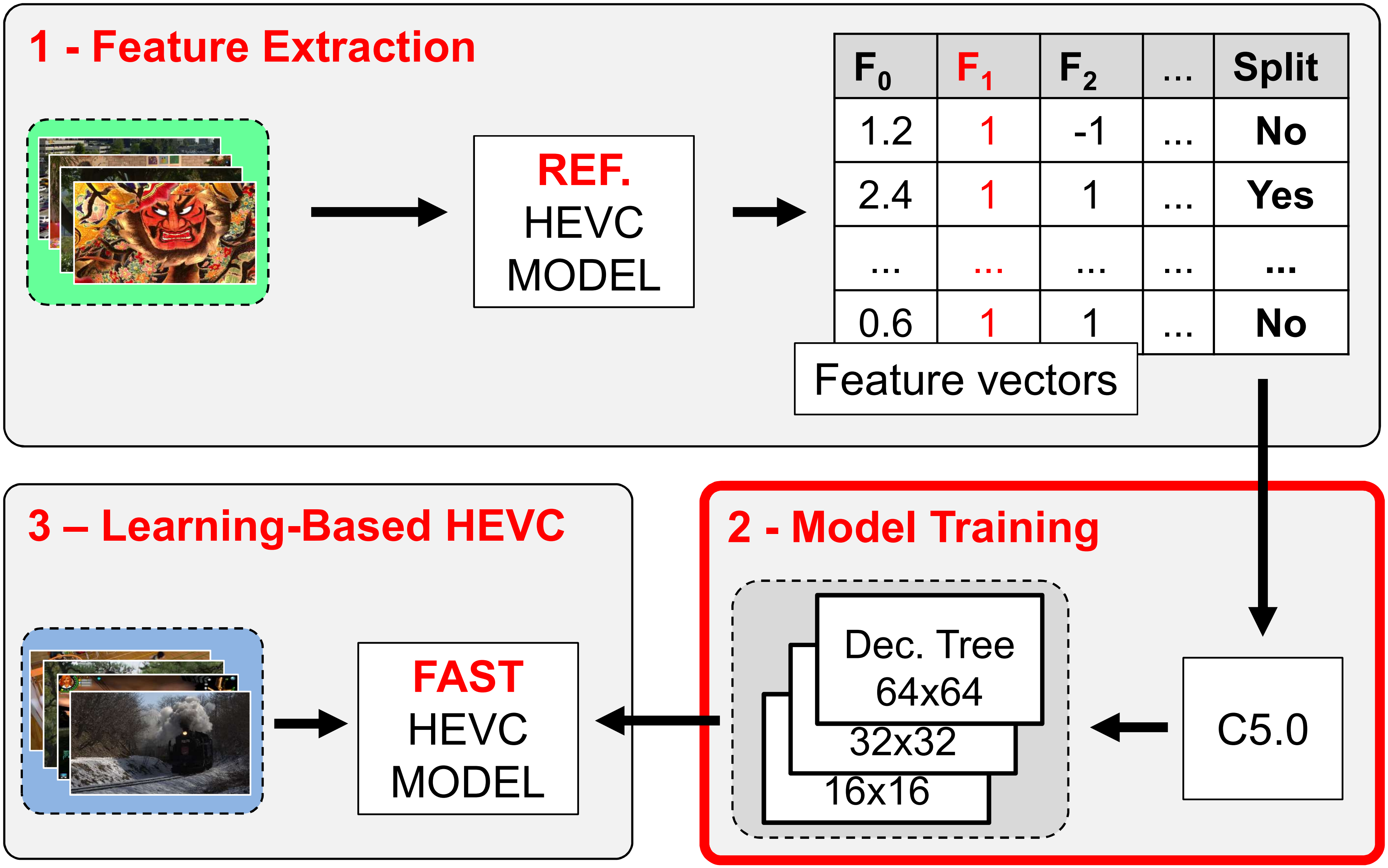
Split Distribution Analysis

- Here we can see how the extracted features can be useful to make decisions of splitting or not a CU



(a) Coded Block Flag (f_{28}) (b) Encoded Bits (f_{12})

Methodology

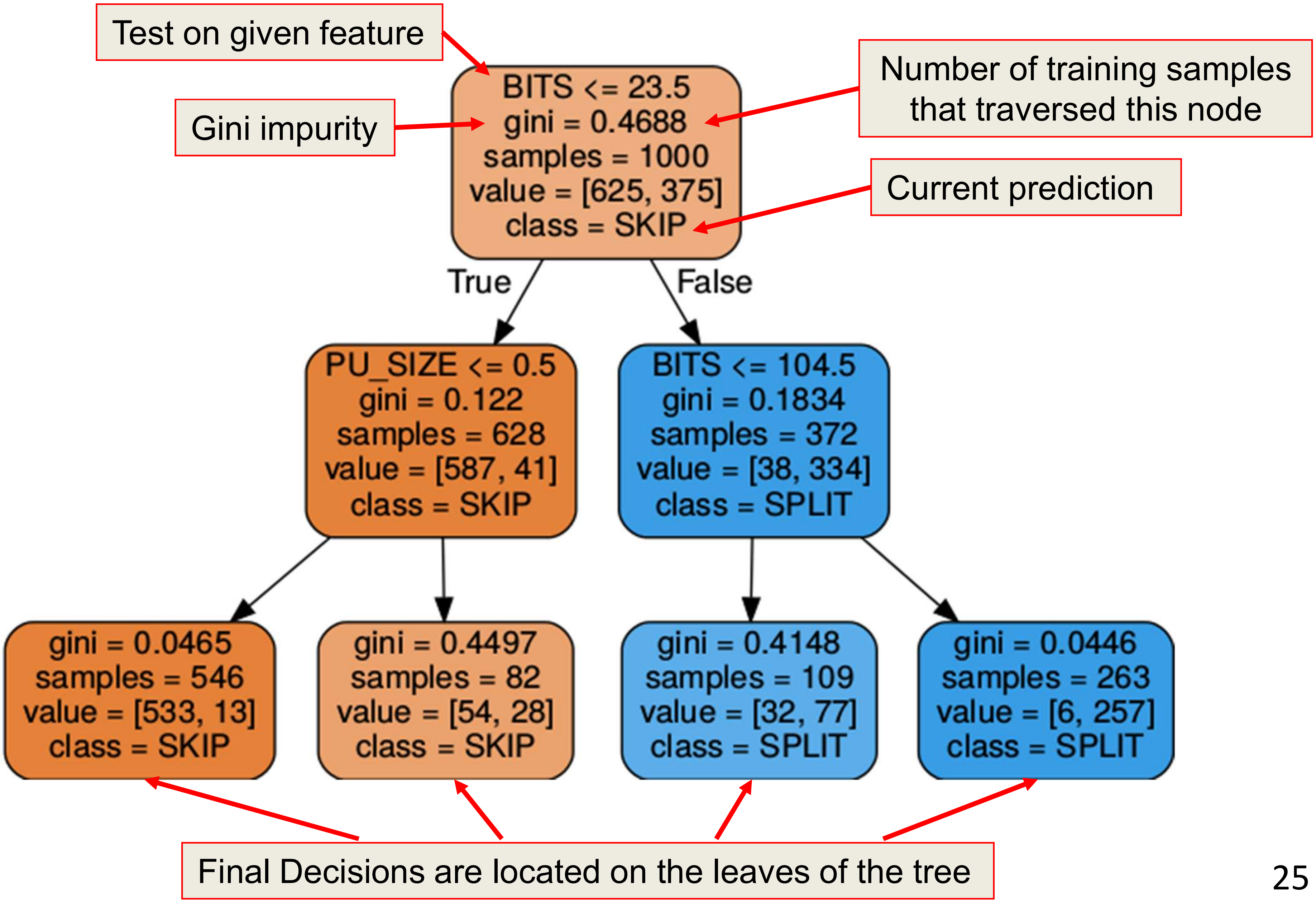


Model Training

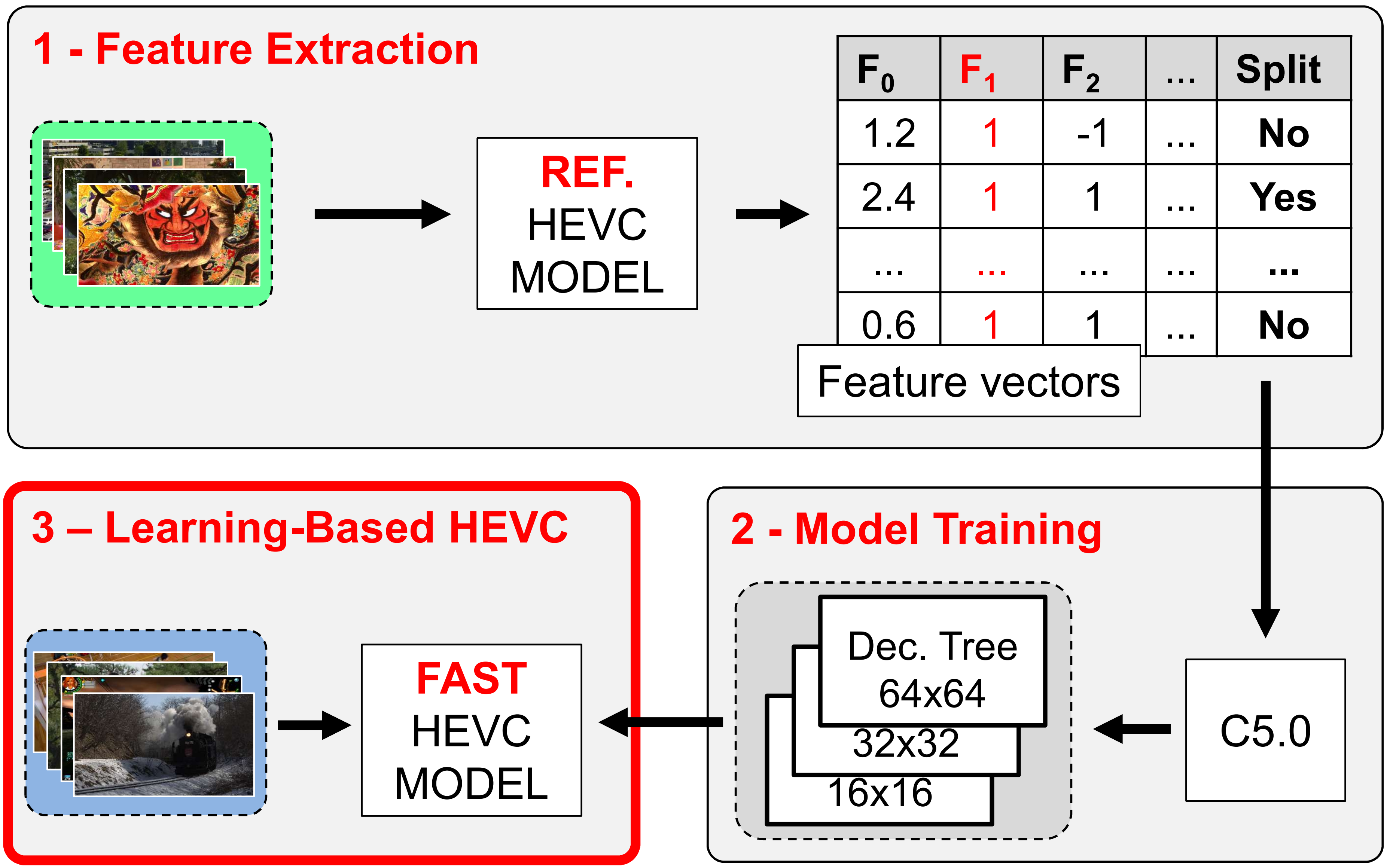
- We trained Decision Trees that are able to predict whether to early-terminate the CU splitting or to continue with the RDO process
- One model was trained for each CU size (64x64, 32x32, 16x16)
- The C5.0 [2] algorithm was used to build our models

[2] R. Quinlan, “Data mining tools see5 and c5.0,” 2015

Decision Trees – Example for CU Splitting



Methodology



Summary

- Introduction and Motivation
- Problem Definition: HEVC Mode Decision Complexity
- Proposed Method: Coding Unit Split Decision Using Decision Trees
- **Results and Discussion**
- Conclusions

Metrics

- Time Savings (TS) for time reduction

$$TS = \frac{T_{Ref} - T_{Test}}{T_{Ref}}$$

- BD-Bitrate (BD-BR) for coding efficiency loss

$$BD_{BR} = \frac{\int_a^b (BR_{Test} - BR_{Ref})}{(b - a)}$$

- $\text{BDTS}_{\text{Ratio}}$ for a combined BD-BR/TS assessment

$$BDTS_{Ratio} = \frac{BD_{BR}}{TS}$$

Methodology

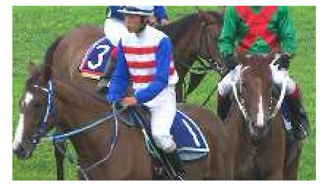
Training Sequences

Traffic	2160x1600
NebutaFestival	2160x1600
ParkScene	1920x1080
BasketballDrive	1920x1080
BQMall	832x480
PartyScene	832x480
BlowingBubbles	416x240
RaceHorses	416x240










Test Sequences

SteamLocomotive	2160x1600
PeopleOnStreet	2160x1600
Kimono	1920x1080
Cactus	1920x1080
BQTerrace	1920x1080
SlideEditing	1280x720
ChinaSpeed	1024x768
RaceHorsesC	832x480
BasketballDrill	832x480
BasketballPass	416x240
BQSquare	416x240








Rate, Distortion, Complexity Analysis

Sequence	BD-BR	TS	$BDTS_{Ratio}$
PeopleOnStreet	0.28%	23.00%	1.21
SteamLocomotive	0.32%	53.90%	0.59
Kimono	0.42%	43.70%	0.96
BQTerrace	0.47%	53.10%	0.88
Cactus	0.42%	47.80%	0.88
RaceHorsesC	0.14%	20.80%	0.68
BasketballDrill	0.17%	40.70%	0.41
BasketballPass	0.16%	43.10%	0.37
BQSquare	0.22%	45.50%	0.48
Kristen&Sara	0.12%	69.10%	0.18
Johnny	0.19%	69.10%	0.27
SlideEditing	0.01%	73.30%	0.01
ChinaSpeed	0.44%	37.80%	1.17
Average	**0.26%**	**47.80%**	**0.54**

Rate, Distortion, Complexity Analysis

Sequence	BD-BR	TS	$BDTS_{Ratio}$
PeopleOnStreet	0.28%	23.00%	1.21
SteamLocomotive	0.32%	53.90%	0.59
Kimono	0.42%	43.70%	0.96
BQTerrace	0.47%	53.10%	0.88
Cactus	0.42%	47.80%	0.88
RaceHorsesC	0.14%	20.80%	0.68
BasketballDrill	0.17%	40.70%	0.41
BasketballPass	0.16%	43.10%	0.37
BQSquare	0.22%	45.50%	0.48
Kristen&Sara	[illegible]	[illegible]	[illegible]
Johnny	[illegible]	[illegible]	[illegible]
SlideEditing	[illegible]	[illegible]	[illegible]
ChinaSpeed	0.44%	37.80%	1.17
Average	**0.26%**	**47.80%**	**0.54**

More than **45%** time reduction with less than **0.3%** loss in coding efficiency

Comparisons with related work

- Comparison with other CU early termination methods

Reference	Technique	BD-rate	Time Savings	BDC_{Ratio}
(ZHANG, KWONG, *et al.*, 2015)	SVM	1.98%	51.45%	3.84
(SHEN and YU, 2013)	SVM	1.35%	44.7%	3.02
(MOMCILOVIC, ROMA, *et al.*, 2015)	Neural Networks	1.17%	47.5%	2.46
(CORREA, P, *et al.*, 2015)	Decision Trees	0.284%	36.7%	0.774
This Work	**Decision Trees**	**0.26%**	**47.8%**	**0.54**

Comparisons with related work

- Comparison with other CU early termination methods

Reference	Technique	BD-rate	Time Savings	BDC_{Ratio}
(ZHANG, KWONG, *et al.*, 2[illegible]	SVM	1.08%	51.45%	3.84
(SHEN a[illegible] 201[illegible]	[illegible]	[illegible]	[illegible]	3.02
(MOMC[illegible] ROMA, *et al.*, 2015)	[illegible]	[illegible]	[illegible]	2.46
(CORREA, P, *et al.*, 2015)	Decision Trees	0.284%	36.7%	0.774
This Work	**Decision Trees**	**0.26%**	**47.8%**	**0.54**

The proposed method **outperforms** state-of-the-art solutions in complexity reduction and coding efficiency

What about complexity scaling?

Complexity-Scalable HEVC Encoder

- Decision Trees can estimate the probability of each decision being correct using offline training statistics
- This probability is called a Confidence Factor (C)
- Originally, any decision with confidence above 0.5 (50%) is taken
- Using this variable, we modified the decision function to make more or less split decisions based on an input threshold ($Split_{TH}$)

Complexity-Scalable HEVC Encoder

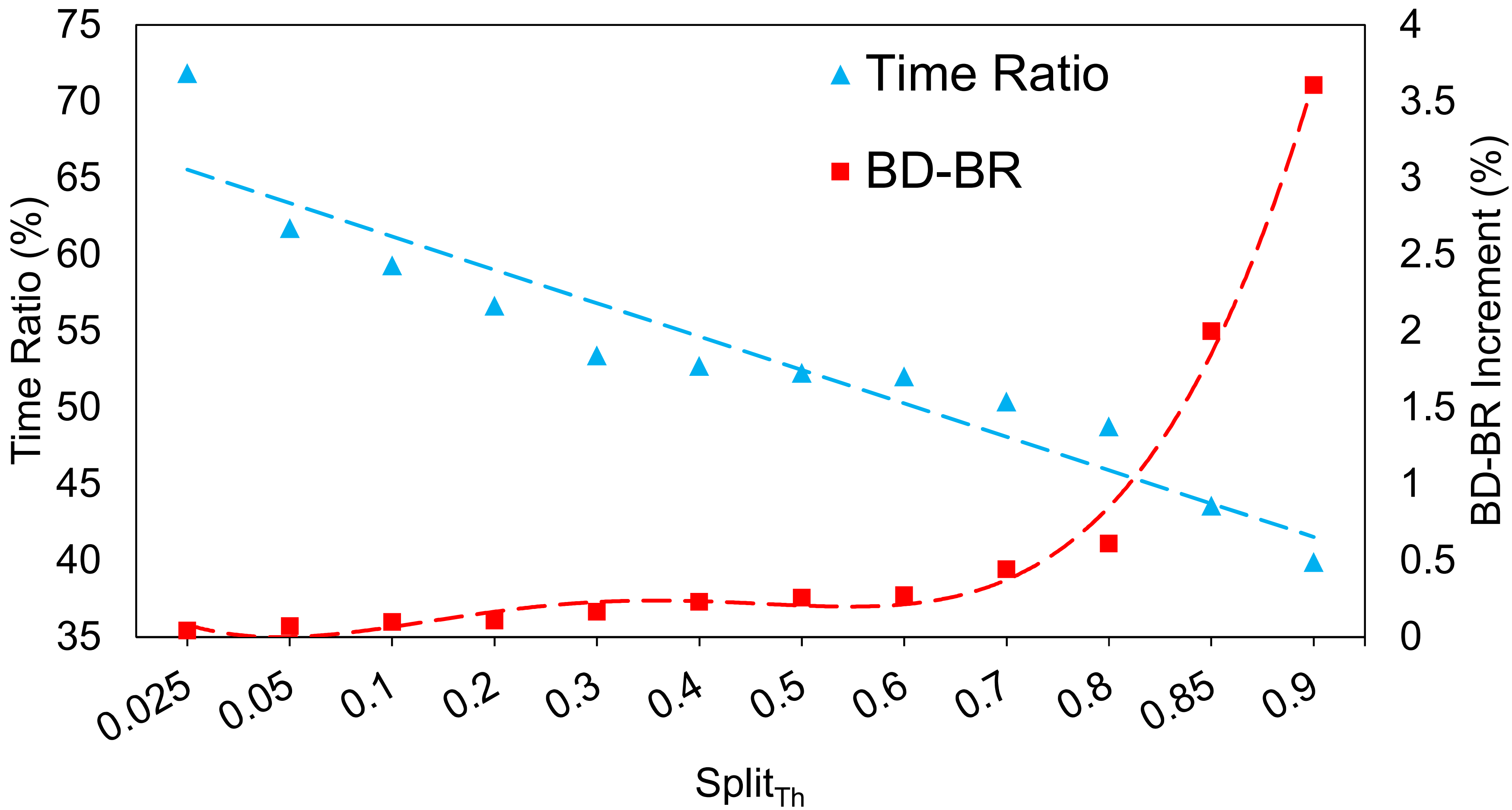
- Considering the original decision of splitting or not (*split* = True or False), we compute a new decision with the following equation:

$$split' = \begin{cases} 1 & \text{if } Split_{th} < 0.5 \text{ and } C < (1 - Split_{th}) \\ 0 & \text{if } Split_{th} > 0.5 \text{ and } C < Split_{th} \\ split & \text{otherwise} \end{cases}$$

$Split_{TH} = 0.5$ – original Decision Tree
$Split_{TH} > 0.5$ – more early terminations (more savings)
$Split_{TH} < 0.5$ – more decisions to split (more coding efficiency)

Split Threshold Analysis

- Employing the decision threshold technique, several complexity points can be achieved



Summary

- Introduction and Motivation
- Problem Definition: HEVC Mode Decision Complexity
- Proposed Method: Coding Unit Split Decision Using Decision Trees
- Results and Discussion
- Conclusions

Conclusions

- Decision Trees are simple and very efficient to predict the CU split decision
- The new set of features proved useful and produced models whose results are **better than other solutions**, even the ones that also use Decision Trees
- The split threshold technique allows **several operating points** with different complexity reduction and coding performance loss
- Average time savings ranging from **28%** up to **61%** can be achieved, with coding losses between **0.04%** and **3.6%**

Thank you!
Questions?

LEARNING-BASED COMPLEXITY REDUCTION AND SCALING FOR HEVC ENCODERS

Mateus Grellert, Guilherme Corrêa, **Bruno Zatt.**
Sergio Bampi, Luis A. da Silva Cruz

April, 2018