CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD


# Fast Video Encoding using Spatio-Temporal Features and Ensemble Classification

by

Muhammad Tahir

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Faculty of Engineering
Department of Electrical Engineering

2019

# Fast Video Encoding using Spatio-Temporal Features and Ensemble Classification

By

Muhammad Tahir

(PE113012)

**Dr. Andrew Ware**

**University of South Wales, UK**

**(Foreign Evaluator 1)**

**Dr. Naeem Ramzan**

**University of West of Scotland, Paisley, Scotland, UK**

**(Foreign Evaluator 2)**

**Dr. Imtiaz Ahmad Taj**

**(Thesis Supervisor)**

**Dr. Noor Muhammad Khan**

**(Head, Department of Electrical Engineering)**

**Dr. Imtiaz Ahmad Taj**

**(Dean, Faculty of Engineering)**

**DEPARTMENT OF ELECTRICAL ENGINEERING**

**CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**ISLAMABAD**

**2019**

Dedicated to my Parents and Teachers

# CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
# ISLAMABAD

Expressway, Kahuta Road, Zone-V, Islamabad
Phone:+92-51-111-555-666 Fax: +92-51-4486705
Email: info@cust.edu.pk Website: https://www.cust.edu.pk


## CERTIFICATE OF APPROVAL

This is to certify that the research work presented in the thesis, entitled "**Fast Video Encoding using Spatio-Temporal Features and Ensemble Classification**" was conducted under the supervision of **Dr. Imtiaz Ahmed Taj**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Electrical Engineering, Capital University of Science and Technology** in partial fulfillment of the requirements for the degree of Doctor in Philosophy in the field of **Electrical Engineering**. The open defence of the thesis was conducted on **September 06, 2019.**

Student Name: Mr. Muhammad Tahir (PE113012)

---

The Examining Committee unanimously agrees to award PhD degree in the mentioned field.

**Examination Committee :**

| | | |
|---|---|---|
| (a) | External Examiner 1: | Dr. Gulistan Raja,<br>Professor<br>UET, Taxila |
| (b) | External Examiner 2: | Dr. Farhan Ullah,<br>ICCC, PAEC,<br>Islamabad |
| (c) | Internal Examiner : | Dr. Muhammad Faisal Iqbal<br>Associate Professor<br>CUST, Islamabad |
| **Supervisor Name :** | | Dr. Imtiaz Ahmad Taj<br>Professor<br>CUST, Islamabad |
| **Name of HoD :** | | Dr. Noor Muhammad Khan<br>Professor<br>CUST, Islamabad |
| **Name of Dean :** | | Dr. Imtiaz Ahmad Taj<br>Professor<br>CUST, Islamabad |

# AUTHOR'S DECLARATION

I, **Mr. Muhammad Tahir (Registration No. PE-113012)**, hereby state that my PhD thesis titled, '**Fast Video Encoding using Spatio-Temporal Features and Ensemble Classification**' is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/ world.

At any time, if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my PhD Degree.

(**Mr. Muhammad Tahir**)

Dated: September, 2019 Registration No : PE-113012

# PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled "**Fast Video Encoding using Spatio-Temporal Features and Ensemble Classification**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/ cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD Degree, the University reserves the right to withdraw/ revoke my PhD degree and that HEC and the University have the right to publish my name on the HEC/ University Website on which names of students are placed who submitted plagiarized thesis.

**(Mr. Muhammad Tahir)**

Dated: September, 2019

Registration No : PE-113012

# *List of Publications*

It is certified that following publication(s) have been made out of the research work that has been carried out for this thesis:-

**Journals**

1. **M. Tahir**, I. A. Taj, P. A. Assuncao and M. Asif, "Fast video encoding based on random forests," *Springer, Journal of Real-Time Image Processing*, pp. 1-21, 2019 [IF : 2.588].

2. M. Asif, Maaz B. Ahmad, I. A. Taj and **M. Tahir**, "A Generalized Multi-Layer Framework for Video Coding to Select Prediction Parameters," *IEEE Access*, Vol. 6, pp. 25277-25291, 2018 [IF : 4.098].

**Conferences**

1. M. Asif, I. A. Taj, S. Ziaudin and **M. Tahir**, "An efficient scheme for intra prediction block size and mode selection in AVC," *13th International Conference on Frontiers of Information Technology (FIT)*, pp. 211–215, 2015.

**Muhammad Tahir**
(PE113012)

# *Acknowledgements*

I would like to express my heartfelt gratitude to my advisor Dr. Imtiaz A. Taj for his continuous support, encouragement and supervision throughout this research work. He has been a constant source of inspiration for me to pursue my PhD. It is indeed an honor for me to work under his supervision both as a student and as a colleague. He has been a great mentor.

I would also like to extend my gratitude to Dr. Pedro A. Assuncao for his valuable suggestions and guidance during my research. His expertise in the field of video coding really helped me in refining the results of my work. I am thankful to him for collaborating in my research.

I am thankful to Muhammad Asif for his help and contribution in this research. I would also like to thank all the other members of Vision and Pattern Recognition (VisPRS) research group for their support and guidance.

I am also thankful to my parents, wife and my children Malaika, Bilal and Ammar for their prayers, support, patience and encouragement during my PhD.

Finally, I would like to thank the Electrical Engineering Department and the higher administration at Capital University of Science and Technology for funding my PhD studies under faculty development program and giving me an opportunity to enhance my research skills and qualification.

# *Abstract*

Video Coding has evolved over the years and new compression standards are being developed at regular intervals. Latest video codecs such as High Efficiency Video Coding (HEVC) have improved compression efficiency to reduce the bit-rates of encoded streams. This improvement has resulted in high computational complexity that becomes a bottleneck in real-time implementation of these codecs. Reduction in this computational complexity without compromise on the video quality is a challenge. Another challenge in fast video encoding is the diverse nature of video content. Hence, there is need for development of intelligent techniques to reduce the computational complexity of latest video codecs that also adapt to the diverse nature of video data.

Research in this thesis focuses on identification of local and global features extracted from video data that can be used to characterize the diverse content in video sequences. Texture variations and motion content in video sequence are quantified to categorize it into simple and complex video sequence. This information is used to develop a content adaptive fast encoding framework to reduce the computational complexity of HEVC. Proposed framework performs equally well both for sequences with simple or complex video content without compromising on video quality. It has been tested with a large set of video sequences and shows promising results as compared to the other recent works.

This research work also focuses on the use of machine learning based algorithms for fast encoding to significantly reduce the complexity of video codecs while keeping bit-rate and PSNR within limits in recent video standards like HEVC. Machine learning based techniques formulate encoding process as a classification problem and use features extracted from video data to model the classifiers that can assist in early predictions during encoding. A large set of spatial and temporal features is extracted from video data and a systematic approach is applied for optimal feature selection. Resultant optimal features are used to train a Random Forests based ensemble classifier for early selection of prediction modes and coding unit sizes in HEVC. Proposed technique has been evaluated with publicly available video data

sets. Experimental results show that the proposed approach significantly reduces the complexity of different profiles in HEVC without compromising on video quality and performs better than other existing fast video encoding implementations.

Fast encoding methods developed in this research have also been validated using emerging applications of HEVC such as compression of light field images. Detailed analysis of different formats in light field image representation has been carried out to identify unique aspects that differentiate them from natural videos. These unique features are used in fast coding of light field images in various formats using HEVC. Experimental results show that proposed technique can be applied in fast coding of light field images without compromise on image quality. These results can be used as benchmark for future research on fast encoding of light field images. Hence, the fast video encoding methods developed in this research can be extended to other applications of image and video coding. These techniques can also be integrated in real life video encoding solutions to enable implementation of latest video codecs on embedded hardware platforms with limited processing power and memory.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **AMP** | Asymmetric Motion Partitioning |
| **ASO** | Arbitrary Slice Ordering |
| **ASIC** | Application Specific Integrated Circuit |
| **AUC** | Area under the Curve |
| **AVC** | Advanced Video Coding |
| **BDPSNR** | Bjontegaard Delta PSNR |
| **BDBR** | Bjontegaard Delta Bit-Rate |
| **CABAC** | Context Adaptive Binary Arithmetic Coding |
| **CART** | Classification and regression trees |
| **CAVLC** | Context Adaptive Variable Length Coding |
| **CTU** | Coding Tree Unit |
| **CU** | Coding Unit |
| **DCT** | Discrete Cosine Transform |
| **DTB** | Digital Television Broadcast |
| **DVD** | Digital Versatile Disc |
| **FM** | Fusion Measure |
| **FMO** | Flexible Macroblock Ordering |
| **FN** | False Negative |
| **FP** | False Positive |
| **FPGA** | Field Programmable Gate Array |
| **GOP** | Group Of Pictures |
| **GPU** | Graphics Processing Unit |
| **HD** | High Definition |
| **HDTV** | High Definition Television |

| | |
|---|---|
| **HEVC** | High Efficiency Video Coding |
| **HVS** | Human Visual System |
| **IBC** | Intra Block Copy |
| **IDCT** | Inverse Discrete Cosine Transform |
| **ITU** | International Telecommunication Union |
| **JCT-VC** | Joint Collaborative Team on Video Coding |
| **JPEG** | Joint Photographic Experts Group |
| **LD** | Low Delay |
| **MAP** | Maximum a posteriori probability |
| **MB** | Macroblock |
| **ME** | Motion Estimation |
| **MHz** | Megahertz |
| **MOOC** | Massive Open Online Course |
| **MOS** | Mean Opinion Score |
| **MPEG** | Motion Picture Experts Group |
| **MSE** | Mean Squared Error |
| **MV** | Motion Vector |
| **MVC** | Multi-view Video Coding |
| **NAL** | Network Abstraction Layer |
| **NZ** | Non Zero |
| **PMD** | Pyramid Motion Divergence |
| **POC** | Picture Order Count |
| **PSNR** | Peak Signal-to-Noise Ratio |
| **PU** | Prediction Unit |
| **QCIF** | Quarter Common Intermediate Format |
| **QP** | Quantization Parameter |
| **RA** | Random Access |
| **RD** | Rate-Distortion |
| **RDO** | Rate-Distortion Optimization |
| **ROC** | Receiver Operating Characteristics |
| **RF** | Random Forests |

| | |
|---|---|
| **SAO** | Sample Adaptive Offset |
| **SAD** | Sum of Absolute Differences |
| **SCC** | Screen Content Coding |
| **SD** | Standard Definition |
| **SMP** | Symmetric Motion Partition |
| **SVC** | Scalable Video Coding |
| **SVM** | Support Vector Machines |
| **TN** | True Negative |
| **TNR** | True Negative Rate |
| **TP** | True Positive |
| **TPR** | True Positive Rate |
| **TU** | Transform Unit |
| **UHD** | Ultra High Definition |
| **VLC** | Variable Length Coding |
| **VLD** | Variable Length Decoding |
| **VVC** | Versatile Video Coding |
| **YUV** | Color format with Luma and two Chroma components |

# Chapter 1

# Introduction

## 1.1 Overview of Video Coding

With the adoption of multimedia technologies such as high definition television (HDTV), online video sharing and video storage in recent years, video coding has become an integral part of broadcast and entertainment media. Applications like video conferencing, video-on-demand, video streaming and social networking websites all use digital video as the main form of media. In all these applications, motivation is to develop video codecs that provide high degree of compression without compromising the quality. MPEG-2 video standard, which was developed about 2 decades ago as an enhancement of MPEG-1, was the first prominent standard adopted world-wide that enabled adoption of digital television systems. MPEG-2 is still used for high definition (HD) and standard definition (SD) television broadcast on satellite, cable networks, and video content storage on DVDs and Blu-Ray disks [1]. Video coding for multimedia applications has evolved over the years through the development of a series of video codecs like ITU-T H.261, MPEG-2, H.263, MPEG-4 and H.264. Every new video standard helped in increasing the coding efficiency albeit the increase in computational complexity. To

meet the increasing demand of diverse video services and popularity of high definition multimedia content, there is constant need for efficient video codecs and sophisticated implementation methodologies.

H.264 or MPEG-4 AVC [2] is currently the most commonly used video codec worldwide. Because of high coding efficiency and other salient features of H.264, nearly half the bits sent on communication networks and stored on video content websites are encoded using H.264 and its share is still growing. H.264 provides 50% coding gain over earlier standard MPEG-2 and 47% coding gain as compared to H.263 baseline profile. This coding efficiency also increased computational complexity of H.264. Nevertheless, usage of H.264 in multimedia applications results in better video quality at much lower bit rates, saving bandwidth and storage costs. H.264 is being used in high definition (HD) TV, Blu-ray storage discs, video conferencing, mobile network video and video content acquisition devices such as camcorders.

High efficiency video coding (HEVC) also known as H.265 [3] is the latest video codec that provides 50% coding gain as compared to its predecessor H.264 while maintaining the same video quality. Main objectives behind the development of H.265 are support for all existing multimedia applications that use H.264 and providing support of new applications like Ultra High Definition (UHD) TV with display resolutions 4kx2k or higher, multi-view video capture and display and support for higher dynamic range and precision for higher quality in terms of color content.

Increase in coding efficiency of H.265 and its predecessors like H.264 has increased computational complexity of these codecs manyfolds. This makes codec implementation on different hardware platforms very difficult and challenging.

## 1.2 Motivation

There has been an enormous increase in the usage of digital video in recent years. Digital video accounts for more than 60% traffic on the Internet. This percentage

is expected to cross 80% by the year 2022 [4, 5]. Social media applications like Facebook, Twitter and online video sharing websites like YouTube, movie content sites like Netflix and online education sites like Edx and Coursera use video as the main medium for sharing information. Video conferencing solutions like Skype and WhatsApp provide video streaming in addition to audio communication. Increase in the processing power of multimedia devices and high network bandwidths has also fueled the increase in demand of video content. This high demand is also fueling the constant drive for increase in compression efficiency of recent video codecs to reduce transmission bandwidths and improve storage capacity for video data.

High efficiency video coding (HEVC) has improved coding efficiency of earlier standards like H.264 by 50%. This high compression efficiency has also increased the computational complexity of the codecs by many times. This computational complexity is the main limiting factor in widespread usage of HEVC in consumer electronics with limited power and processing capability. Hence, there is need for intelligent algorithms to reduce the complexity of these video codecs to enable their implementation in real-time multimedia applications. Diverse nature of video content is another challenge that degrades the performance of fast video encoding techniques. Content adaptive fast encoding schemes are required to enable fast encoding methods to handle all sorts of video data. Together these fast encoding methods can enable widespread adaptation of recent codecs like HEVC on the Internet and multimedia devices with limited resources.

## 1.3 Applications of Video Coding

Applications of multimedia content and video coding are everywhere. Reduction in bit-rate of latest video codecs and increase in processing power of computing devices has increased these applications many-fold. Today video traffic is the biggest load on wired and wireless communication networks. Below are some of the applications where video coding is being used.

**Video Conferencing and Telemedicine**: Low bit-rate video coding is used in video conferencing and telephony applications for live video communication. Video standards like H.263, MPEG-4 or H.264 are used in video conferencing solutions like Skype and WhatsApp. Telemedicine solutions use video streaming to provide remote health care to distant patients.

**Digital Video Storage**: Digital video content in the form of movies and video clips is often stored on DVDs and Blu-Ray Discs. These videos are encoded using Video Codecs like MPEG-2, MPEG-4 and H.264. Online video content sharing websites also store videos on web servers. Efficient video coding methods can reduce the storage space requirements.

**Internet Video Streaming**: Applications of Video on world wide web (www) are large in number. Video content websites like Youtube, Dailymotion, Social media sites like Facebook, P2P networks, live streaming sites like Netflix and MOOC sites like Coursera and Edx, all use digital video . Video codecs used are MPEG-4 and H.264.

**Digital Television Broadcast (DTB)** : Digital television broadcast is being used all over the world in the form of high definition TV (HDTV) and standard definition TV (SDTV). Video Codecs used are MPEG-2 and H.264. Similarly, digital videos and video encoding is being used in high definition (HD) and ultra-high definition cinema for higher quality multimedia experience.

**Video over 3G Wireless**: Video on cellular and 3G Networks is being used for telephony and surveillance applications. Supported video standards are H.263, H.264.

**Home Surveillance and Security**: Security and Surveillance systems are being used in homes and offices. Digital cameras capture and store video content for offline viewing. Video standards being used are MPEG-2 , MPEG-4. Storage of long hours of surveillance videos requires extra storage space. Hence, efficient coding can reduce storage requirements.

**Virtual Reality and 3D Games**: Virtual reality systems and 3D game consoles like Xbox and Play-station use digital video and video encoding in graphics and simulations. These applications often use high quality videos with resolutions upto 720p and 1080p. Video Codecs used are MPEG-4, H.264.

**Multi-view Videos and Light Field Images**: Multiview video captures the same scene from different viewpoints to give more realistic view of the scene. Similarly, light view images is an other emerging field in image representation that captures more than 2D information in a visual scene. Both multiview videos and light field images contain huge amount of data. Efficient coding of this data is a challenge for future video and image coding standards. Recent developments in efficient coding of light field images have shown that using new video codecs like HEVC give more compression efficiency for light fields as compared to image coding standards like JPEG [6].

These applications of video coding are summarized in the Fig. 1.1.


FIGURE 1.1: Applications of Video Coding

## 1.4 Research Objectives

There are two main challenges in video coding using newest video codecs like HEVC. One is the computational complexity of the codec because of the complex structure of coding tree units (CTUs) at multiple depth levels and its sub-partition in coding units (CU) and prediction units (PU). This quad-tree structure makes selection of optimal CU size, prediction modes and block size highly complex. Second challenge is the diverse nature of the video content. Focus of this research is the development of intelligent algorithms to reduce the computational complexity of HEVC. This reduces the encoding time that ultimately enables implementation of video encoders in real-time on multimedia devices with limited computational capabilities.

Main Objectives of this research are :

- Development of a content adaptive fast video encoding framework that can adapt to diverse characteristics of video data and gives equally good performance both for simple and complex video content without degradation in RD performance.
- Identification of appropriate local and global features in spatial and temporal domain of video frames that can efficiently characterize video content and help in effective prediction of complex decisions during video encoding.
- Development and formulation of fast encoding algorithms based on classification methods to reduce the complexity of latest video codecs while maintaining good RD performance.
- Design of a systematic feature selection methodology for video data that can be used to reduce the feature space for classification problems. Using score fusion technique to combine multiple feature selection methods for selection of optimal features.
- Application of fast video encoding methods developed in this research to other emerging domains like compression of light field images for effective

and computationally efficient encoding of various light field image formats using HEVC.

## 1.5 Research Contributions

Main contributions of this research work are :

- Development of a content adaptive fast video encoding technique for HEVC to handle the variations in video content. Proposed technique uses local and global parameters extracted from video data to characterize the video content. Spatial and temporal information in video frames is used to fine tune the fast encoding framework according to the video content. Resulting framework better adapts to the simple and complex video sequences and performs better than other existing implementations. Our contributions are submitted in [7].

- Development of a Random Forests based ensemble classification approach for fast encoding in HEVC with low computational overhead. Random Forests based classifier works as ensemble of online and offline trained classifier models in different configurations and is robust to changes in video content. A systematic approach has been used for tunning of classifier parameters. Proposed ensemble classifier has low run-time overhead and can be used in real-time implementations. Results of the proposed work outperform other state of the art implementations for different HEVC profiles without sacrificing video quality. Our contributions have been published in [8].

- Comprehensive comparative analysis of different feature selection methodologies for selection of optimal features in video data for classifier training and evaluation is carried out. A score fusion technique is developed to combine feature selection techniques for improved results. Classifier models trained using the resultant optimal feature set show improved performance with low computational overhead. Our contributions have been published in [8].

- Application of fast video encoding techniques for low-complexity coding of light field images in different representation formats using HEVC. Detailed analysis of multiple light field image formats has been carried out and unique features are identified that can assist in fast coding of light field images using video codecs like HEVC. Findings of this investigation show that fast encoding methods developed in this research can be used to reduce the coding complexity of emerging media like light field images without degradation of image quality. These novel results can also be used as a benchmark for future research. Our contributions are submitted in [9].

## 1.6 Thesis Organization

This thesis is composed of seven chapters. Chapter 1 starts with the introduction of video coding and significance of the topic under consideration in this research. Applications of video coding are discussed in detail. Research objectives and major contributions of the research are outlined.

Chapter 2 gives detail of background on video coding. Main blocks of a hybrid video codec model are discussed. This is followed by brief history of video coding standards and overview of the tools introduced in video codecs over the years. Main challenges in video coding are described followed by brief overview of different machine learning methods used in fast video coding. Different evaluation methods used for performance evaluation of video codecs are also discussed. Chapter also gives detail of different publicly available data sets used in the experimentation and performance evaluation.

Chapter 3 discusses the literature survey. Different methods used for fast coding of high efficiency video coding are discussed in detail. These methods have been divided into three main categories namely, statistics based methods, machine learning based methods and content adaptive methods. Chapter also gives detail of recent research done in the area of Light Field Images. Deficiencies in each

category are highlighted. Chapter also discusses the gap in the research, problem statement of the proposed research methodology.

In Chapter 4, a comprehensive framework for content adaptive fast coding of video data is described. Computational complexity of different blocks in HEVC is discussed. Usage of spatial and temporal features in video data to develop a content adaptive technique to reduce the computational complexity of HEVC is discussed. Proposed framework for fast coding of video data handles both simple and complex video streams. Video frame is first categorized into simple or complex frame using global and local parameters. Based upon this categorization, frame specific techniques are used for fast encoding of prediction modes and early CU size prediction. Results of the proposed method are discussed and compared with other similar implementations.

Chapter 5 describes proposed methodology for fast coding of HEVC standard using machine learning methods. A detailed feature selection procedure is given where comparative analysis of filtering and wrapper based approach for video data is described in detail. Design and formulation of Random Forests Classifier for CU, TU Split Decision and Skip mode selection is described. Accuracy of RF classifier and selection of optimal classifier parameters is discussed. Results of the proposed fast coding method are described for different HEVC profiles and performance is compared with other recent works.

In Chapter 6, fast video coding techniques developed in this research are applied in fast coding of Light Field images. Different representations of light field images are discussed and analysis has been done on light field data to extract the features and analyze the trends in light field image representation that can be used for fast coding. Detail of the proposed method for feature selection is given that uses combined score of filtering and wrapper based approach as the measure to rank and select the optimal features. Design of proposed Random Forests based classifier for fast selection of prediction mode and CU depth is described. Results section describes the speed-up and RD performance of the proposed fast coding

method for Light Field Images for different Light Field representations and HEVC profiles.

Chapter 7 is last chapter of the thesis that gives the concluding remarks about the research and also discusses how this research work can be extended in the future.

# Chapter 2

# Video Coding Standards and Performance Evaluation

This chapter describes different formats for representation of digital video. Building blocks of a video codec are explained and brief overview of various video coding standards is given. HEVC, that is the focus of this research is discussed in detail. Chapter also describes challenges in video coding. Various machine learning based methods used in fast video coding and evaluation matrices for video coding are briefly discussed. Chapter also gives overview of the video data sets used in this research work.

## 2.1 Digital Video Representation

Digital video is composed of color images sampled over time. These images are generally called video frames. Each video frame is composed of pixels and the total number of pixels in horizontal and vertical dimensions determine frame resolution. Similarly, number of frames per unit time determiners the video frame rate. Color space of a video sequence determines how image brightness (luminance) and color (chrominance) information is described in the video. Two well known color space formats for video representation are [10]:

- RGB Color Space
- YUV Color Space

### 2.1.1 RGB Color Space

In the RGB color space, each pixel in video frame is composed of RED (R), GREEN (G) and BLUE (B) components. Red, green and blue are primary colors for representation of color information. In RGB color space, proportions of red, green and blue components are same. RGB color space is adequate for video capture and display devices. In video capturing devices, separate sensor arrays are used for each of the red, blue and green components.

### 2.1.2 YUV Color Space

YUV color space decomposes a video frame into brightness and color components and stores them separately. Human visual system (HVS) is more sensitive to brightness than color information in a video frame. YUV color space exploits this limitation of HVS and represents brightness component with full resolution while color component is stored in lower sub-sampled resolutions. This results in less number of bits to represent video data. Video codecs often process videos in YUV color space.

In YUV color space, Y represents the luminance (luma) component and U, V represent the two color (chroma) components. In YUV420 format, Y component has the same resolution as size of video frame. U and V components are sub-sampled and have lower resolution. This also results in reduced storage and transmission requirements. In YUV420 format, Y, U and V component are calculated from R,G,B components in a video frame according to following formulas [11].

$$Y = 0.299R + 0.587G + 0.114B, \tag{2.1}$$

$$U = 0.564(BY), \tag{2.2}$$

$$V = 0.713(RY). \tag{2.3}$$

### 2.1.3 Digital Video Resolutions

Digital video uses various spatial and temporal resolutions. Spatial resolution defines size of the video frame in horizontal and vertical dimensions in terms of pixels. Temporal resolution specifies number of frames to be captured/displayed per unit time. Video resolutions highly depend upon applications and available storage or transmission capacity. With increase in demand for high quality multimedia like HDTV and increase in processing power of multimedia devices, demand for large frame resolution and high frame rates is continuously increasing. Common frame resolutions used in video applications are shown in the Table 2.1 [11].

TABLE 2.1: Video Frame Resolutions

| S.No. | Frame Resolution | Dimensions | Comments |
|---|---|---|---|
| 1 | QCIF | 176x144 | Quarter Common Intermediate Format |
| 2 | CIF | 352x288 | Common Intermediate format |
| 3 | SD | 640x480 | Standard Definition |
| 4 | HD | 1280x720 | High Definition Video |
| 5 | UHD | 4096x2160 | Ultra High Definition Video |

Frame rates in video sequences also depend upon the application. Common frame rates used in digital video are 25 fps, 30 fps, 60 fps and 100 fps.

## 2.2 Building Blocks of Video Codec

A video codec is composed of an encoder and a decoder. Encoder encodes source video frames into a compressed bitstream. Encoder exploits spatial and temporal redundancy in the video data in order to compress it. Job of the decoder is to decode the encoded bitstream to reproduce uncompressed video. Block diagram of a generic video encoder and decoder is shown in the Figs. 2.1 and 2.2 [1].

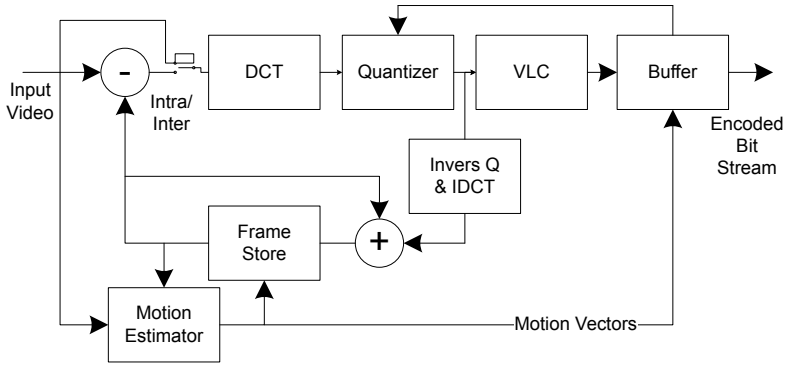


FIGURE 2.1: Generic Video Encoder Building Blocks

Majority of the current video coding standards are based on the hybrid video coding model because the techniques being used are a hybrid of picture coding methods and motion handling[12]. Each video frame is first divided into square shaped blocks called macroblocks (MB). Video frame is processed block by block using the principal of predictive coding [13]. Video encoding and decoding is done separately on luma and chroma components in a video frame. All steps in video encoding or decoding are applied on macroblocks in a video frame in the raster scan order.

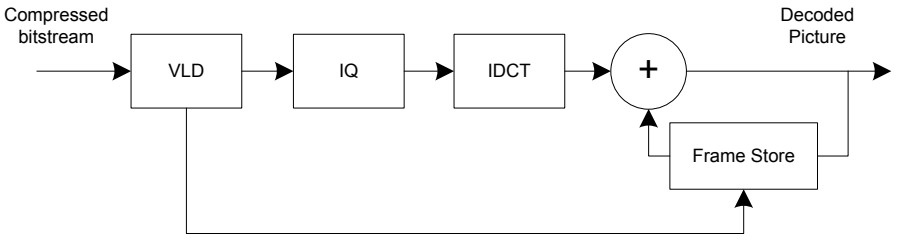


FIGURE 2.2: Generic Video Decoder Building Blocks

Main components in a block based hybrid video codec include residual generation, motion estimation and compensation, transform and quantization, entropy coding and rate distortion optimization. These components are explained in the following subsections.

### 2.2.1 Residual Generation

According to predictive coding principal, difference between macroblock in the current frame and its predicted value from the previous frame is called residual or

error signal. This error signal has reduced dynamic range and requires less number of bits to encode. At encoder, residual signal is transformed and coded to generate the encoded bitstream. At the decoder, after decoding and inverse transformation the residual signal is added to a prediction value to reproduce the macroblock in the video frame. Macroblocks that are predicted from neighboring blocks in the current frame are called intra MBs. On the other hand, those predicted from blocks in previous frames are called inter MBs.

### 2.2.2 Motion Estimation and Compensation

Motion estimation is the process of finding position of a macroblock in the previous frame. Position in previous frame is marked using a displacement vector called motion vector. Sum of absolute differences (SAD) operation is performed to find the position of a macroblock in previous frame. Motion estimation is the most complex operation in video encoding. Motion compensation is the reverse of motion estimation where motion vectors are used to get motion compensated macroblock from previous frame. Motion estimation is performed on the luma block(Y) in the video frame. A scaled version of the motion vector is than used for the chroma blocks U and V.

### 2.2.3 Transform and Quantization

Residual signal is transformed to further exploit spatial redundancy. Energy in a video frame is concentrated in the low frequency region. Transform operation is applied to isolate this low frequency component from the high frequency component. In video codecs, Discrete Cosine Transform (DCT) is used on block by block basis to get frequency coefficients. This transformation is followed by quantization operation that further reduces the insignificant high frequency components. Quantization is a lossy operation that reduces number of non-zero coefficients and results in further compression with insignificant data loss.

### 2.2.4 Entropy Coding

Entropy coding also know as variable length coding (VLC) is the last block in video encoding. DCT coefficients, motion vectors and other data in video encoder is entropy encoded to generate the compressed stream. Two well-know methods for entropy encoding used in video codecs are Context Adaptive Variable Length Coding (CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC). Variable length decoding (VLD) is the reverse of entropy encoding. In video decoder, first VLD is performed on encoded stream to get residual coefficients, motion vectors and other data.

### 2.2.5 Rate Distortion Optimization

In latest video codecs, a macroblock can be encoded in one of many ways. These video codecs support many MB types, prediction modes and block partition sizes. Rate distortion optimization (RDO) is used to select the optimal choices that reduce distortion for generation of minimum number of bits [14]. A Lagrangian cost function is computed with parameter $\lambda$ according to the Eq. 2.4 [15, 16].

$$J = D + \lambda R. \tag{2.4}$$

For every MB, distortion D and bitrate R is computed for all the possible prediction modes, block partitions and MB types. These values are used to compute the Lagrangian cost J and the mode or the partition type with minimum Lagrangian cost is selected as the best option. Parameter $\lambda$ is empirically found using quantization parameter QP. All the latest video encoder implementations use this RDO process to exhaustively evaluate all the available options and selection the optimal one. This exhaustive search results in increase in the complexity of the encoding process. With increase in the number of prediction modes, MB types and block partitions in current video codecs, this complexity has enormously increased.

## 2.3 Video Coding Standards Over the Years

Video coding standards have evolved over the years. Applications like real time video communication, video storage and broadcast of digital multimedia content are the major driving force behind this evolution. Many video coding standards have been released over the years. Fig. 2.3 shows the time line of the major video codecs developed over the years [17, 18]:

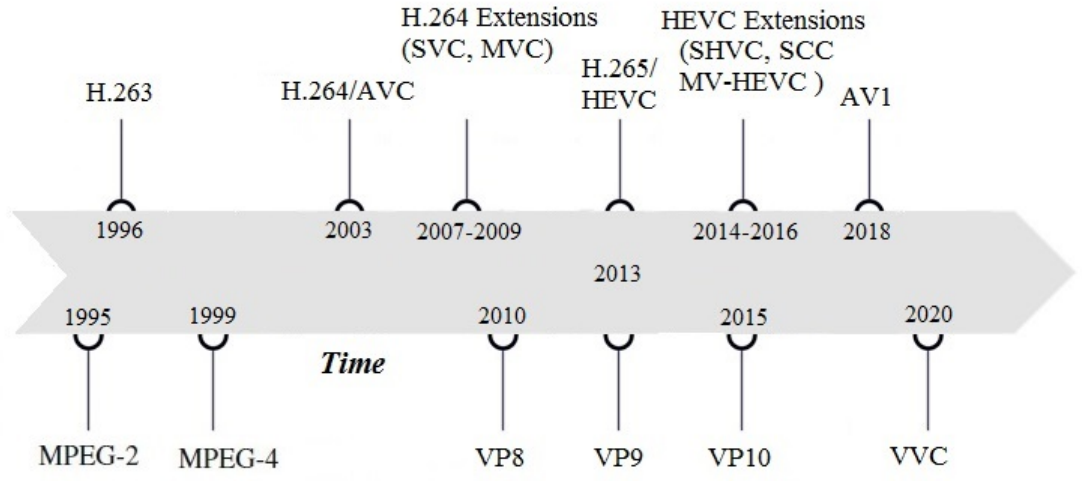


FIGURE 2.3: Video Codecs over the years

Following subsections briefly describe the major video codecs.

### 2.3.1 MPEG-2

MPEG-2 was the first significant video coding standard that was jointly developed by ISO/IEC and ITU-T in 1995 as ISO/IEC 13818-2 [19, 20]. Although there were earlier video standards like H.261 and MPEG-1 but MPEG-2 was a major success with significant enhancements. MPEG-2 used macroblocks of size 16x16 and inter-frame prediction with residual transform coding (DCT) of size 8x8. Other tools included in the standard were quantization and variable length coding. MPEG-2 also employed the concept of profiles and levels for encoded video streams. MPEG-2 was backward compatible with MPEG-1 and was largely used for multimedia video storage on CR-ROMs .

### 2.3.2 H.263

H.263 was developed by ITU-T in 1996 and was a dominant standard as compared to its predecessors [21]. Main application of H.263 was video conferencing solutions at low bit rate. Prominent features in the standard include bi-direction prediction, arithmetic entropy coding and median motion vector prediction. New features like error resilience, deblocking filter mode and improved coding tools were introduced in the standard. Coding efficiency of H.263 was further enhanced in later versions known as H.263+ and H.263++.

### 2.3.3 MPEG-4

MPEG-4 is an ISO/IEC standard that was developed by MPEG in 1999 [22, 23]. MPEG-4 was better than MPEG-2 in coding efficiency and flexibility. Core compression tools supported in the standard were based on H.263. MPEG-4 introduced new features like coding of irregular shaped objects in natural scenes and animation sequences. Support for coding of still images and scalable video coding was also introduced in MPEG-4 in addition to coding of video sequences. MPEG-4 had support for multiple profiles lie simple profile, advanced simple profile and simple scalable profile.

Using a large set of new coding tools, MPEG-4 targeted applications like high definition television broadcast, DVD storage, low resolution surveillance cameras and network bandwidth adaptive video streaming.

### 2.3.4 H.264

H.264 standard also know as advanced video coding (AVC) / MPEG-4 part 10 was jointly developed by ITU-T and MPEG group in 2003 [2, 24]. H.264 improved coding efficiency upto 50% as compared to prior coding standards. This improvement in coding efficiency also increased in codec complexity. Improvement in coding efficiency of H.264 was a result of many unique features that were introduced for

the first time. Significant features in H.264 include introduction of context adaptive binary arithmetic coding (CABAC) for high compression, multiple reference frames for inter-frame motion estimation, variable block size motion compensation, integer discrete cosine transform (DCT) quarter-pel motion compensation and in-loop deblocking filter. Features introduced for error resilience were flexible macroblock ordering (FMO), arbitrary slice ordering (ASO) and data partitioning in extended profile. Block diagram of the codec is shown in the Fig. 2.4.

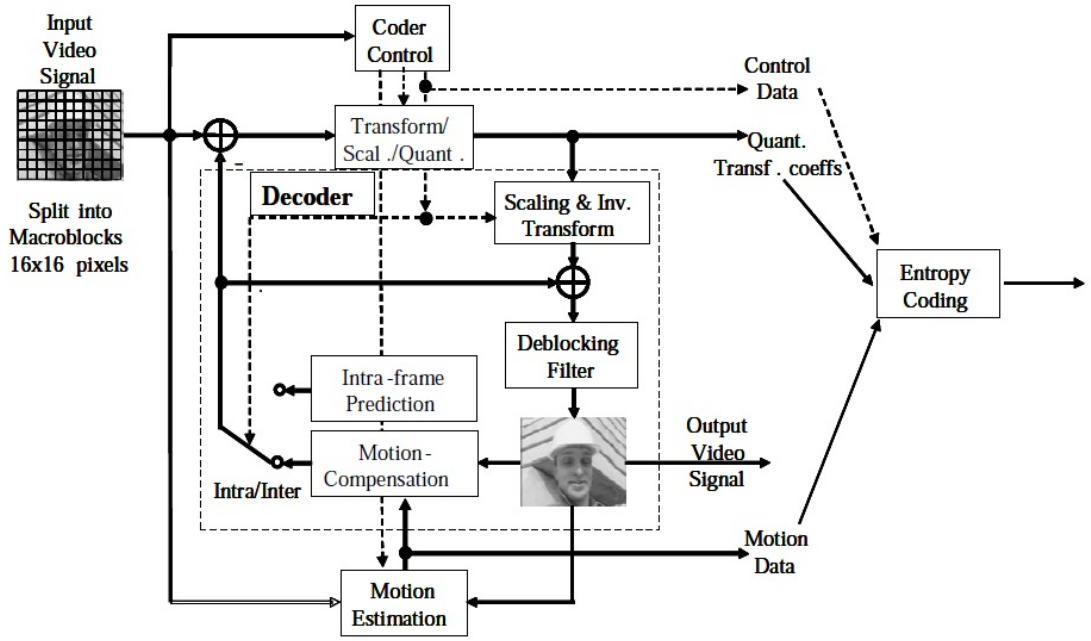


FIGURE 2.4: H.264 Block Diagram [2]

H.264 also introduced the concept of network abstraction layer (NAL) to make the encoding stream network friendly in addition to video coding layer (VCL). Each NAL unit in H.264 encapsulates VCL and non-VCL packets [11]. H.264 supported both progressive and interlaced frames. Each frame was further divided into slices and macroblocks. H.264 supported three profiles namely baseline, main and extended profiles. H.264 was followed by two extensions. First one was Scalable Video Coding (SVC) in 2007 [25] and second one was Multiview Video Coding (MVC) in 2009 [26].

## 2.4 High Efficiency Video Codec (HEVC)

High efficiency video codec (HEVC) a.k.a H.265 is the latest video codec that was jointly developed by ITU-T and ISO MPEG group [3, 27]. It is also based on the hybrid codec architecture that offers many advanced features. Development of HEVC targeted two main areas. One is processing of high resolution video content. Second is codec implementation on parallel processing hardware platforms. Enhancements and modifications done in HEVC focus on these areas. It supports demanding applications and services like Ultra High Definition TV (UHD-TV) with display resolutions 4Kx2K, or higher, and multi-view video capture and display with higher dynamic range in terms of color content. HEVC has 50% more coding efficiency as compared to H.264 while offering the same perceptual visual quality. Main building blocks of the codec are shown in Fig. 2.5.

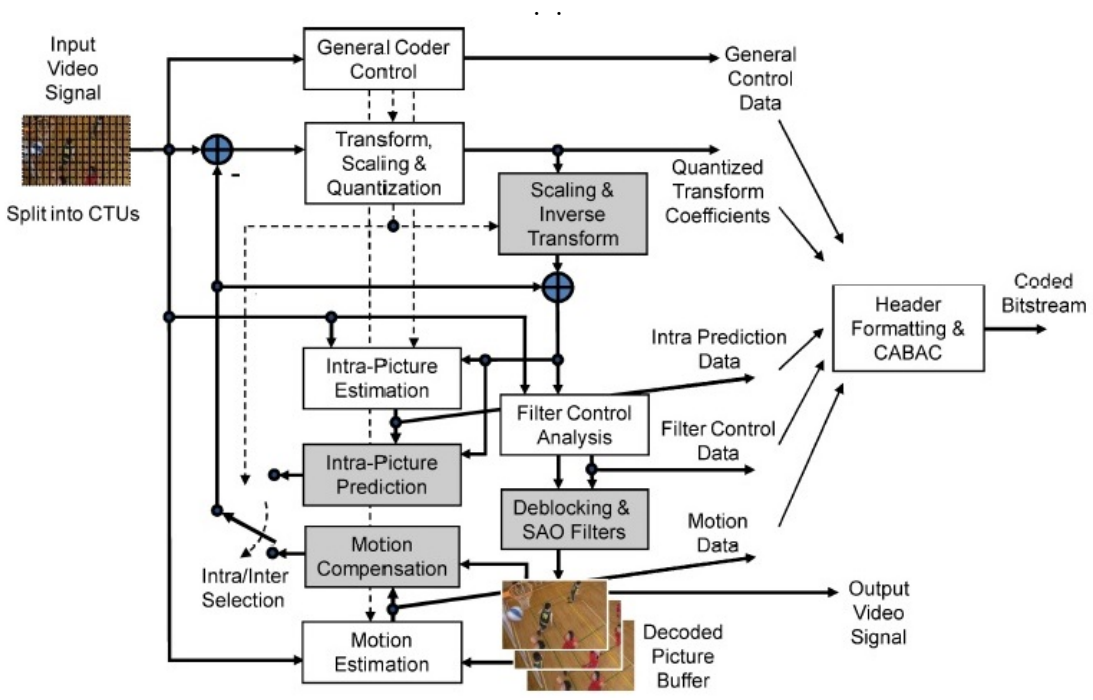


FIGURE 2.5: HEVC Block Diagram [3]

Detail of the new coding tools introduced in HEVC, quad-tree structure, prediction modes and entropy coding is described in following subsections.

### 2.4.1 Video Coding Layer

In HEVC, like the earlier standards, video frame is first partitioned into blocks. Concept of macroblock with fixed size of 16x16 is replaced with larger blocks called coding tree units (CTUs). Size of CTU can vary from 64x64, 32x32 down to 8x8 [28]. CTUs are further divided into square size coding units (CUs) in a hierarchical quad tree structure that exists at different CU depth levels. The CU size at depth level 0 is 64x64 whilst at depth level 3, it is 8x8. This hierarchical structure is shown in in Fig. 2.6.

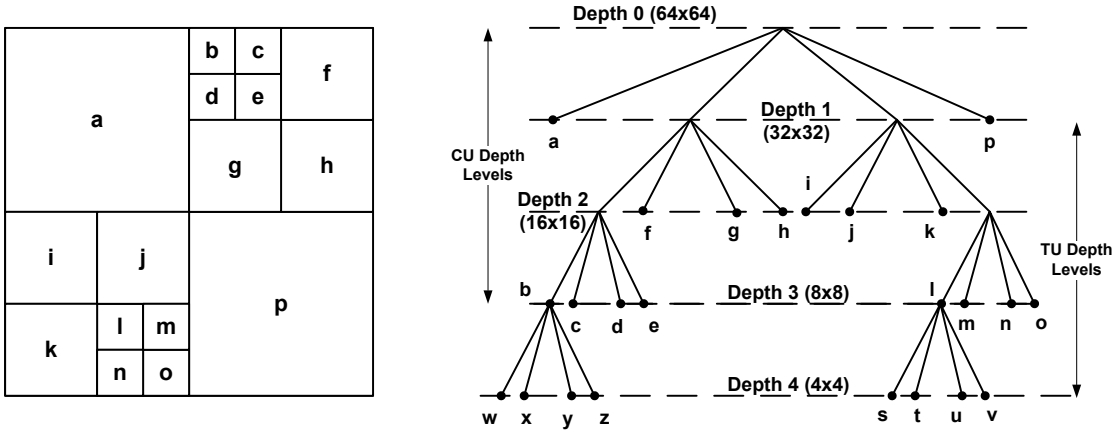


FIGURE 2.6: HEVC CU and TU Quad-tree Structures

### 2.4.2 Coding Units

In HEVC, several new coding structures are introduced to improve the coding efficiency. These units include Coding Unit (CU), Prediction Unit (PU) and Transform Unit (TU). CTU is first divided into square shaped CUs at different depth levels. Each CU may contain one or more PUs. For each PU, several intra and inter prediction modes are evaluated and various partition sizes are evaluated. These partitions can be symmetric and asymmetric partitions. Symmetric partitions include inter 2Nx2N, Nx2N, 2NxN and NxN. In HEVC, asymmetric partitions have also been introduce to increase the coding efficiency in the inter mode. These partitions are inter 2NxnU, inter 2NxnD, inter nLx2N and inter nRx2N . For intra modes, only symmetric partitions 2Nx2N and NxN are used [3]. These partition sizes are shown in Fig. 2.7.

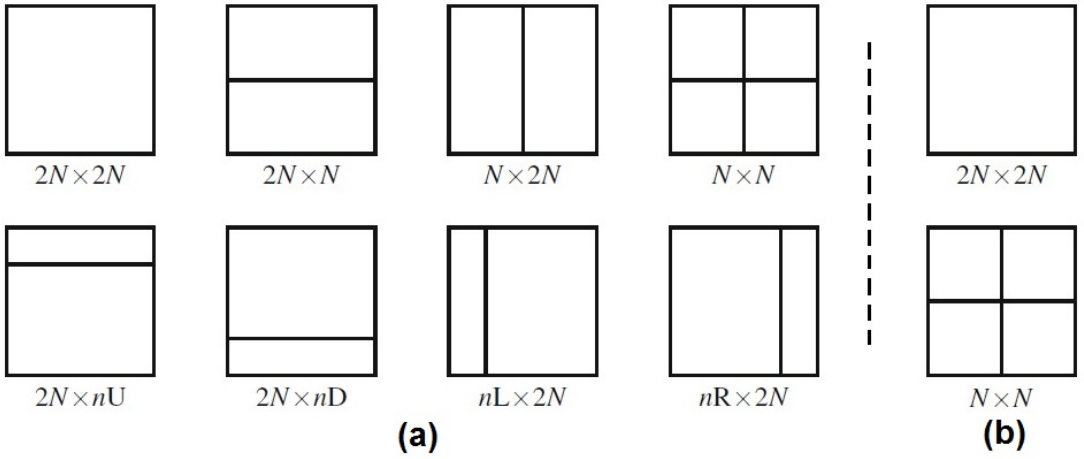


FIGURE 2.7: Partition sizes in HEVC Prediction Units (a) Inter Partitions (b) Intra Partitions

HEVC also introduced variable TU sizes. Inside each PU, TUs of different sizes are evaluated. These sizes include 4x4, 8x8, 16x16 and 32x32. Together these TUs at various depth levels form a quad-tree structure like the CU quad-tree, as shown in Fig. 2.6.

### 2.4.3 Slices, Tiles and Wavefront Processing

In HEVC, a video frame can be partitioned into slices, tiles and wavefronts. A slice can be coded independently and it can be part of a video frame or an entire frame. Slices are further partitioned into CTUs that are processed in raster scan order. Slices provide error resilience against data loss during video transmission. In addition to slices, new partitions have been introduced in HEVC to support parallel processing on highly parallel hardware platforms. These partitions include Tiles and Wavefront parallel processing (WPP). Tiles are rectangular partitions that can be independently decoded like slices. CTUs inside a tile are processed in raster scan order. Partition of video frame into slices and tiles is shown in the Fig. 2.8 [3].

In WPP, a slice is divided into rows of CTUs that can be processed in parallel. With WPP, processing of CTUs in the second row can start, when two CTUs in first row have been processed. This helps in parallel processing of CTUs inside a

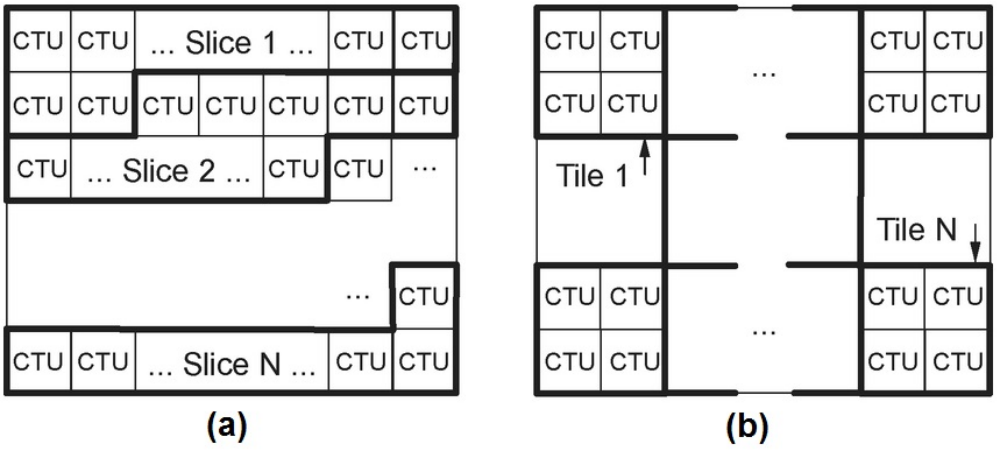


FIGURE 2.8: Video Frame Partitions [3] (a) Slices (b) Tiles

slice and multiple rows of CTUs can be processed using multiple processor cores or multiple software threads inside an application. Concept is shown in the Fig. 2.9.

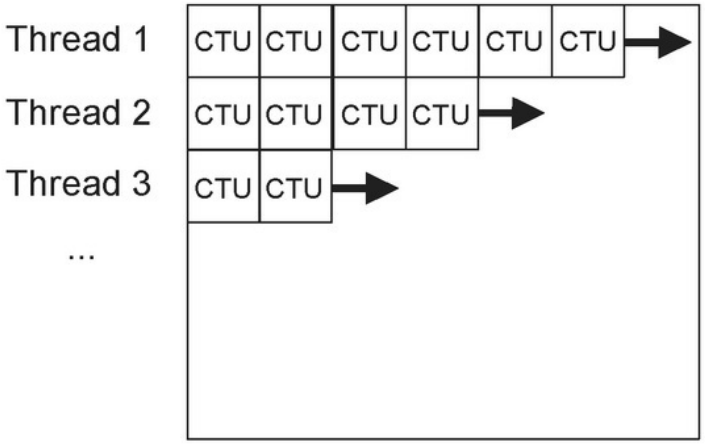


FIGURE 2.9: Wavefront Parallel Processing in HEVC [3]

### 2.4.4 Intra-picture prediction

HEVC like its predecessors, supports block based intra prediction to exploit spatial correlations inside a video frame. There are 35 intra prediction modes in HEVC, that include DC prediction, planer mode and 33 angular prediction modes [29]. Orientation of angles in these prediction modes is shown in Fig. 2.10. Prediction units inside an intra predicted CU can be of size 2Nx2N or NxN.

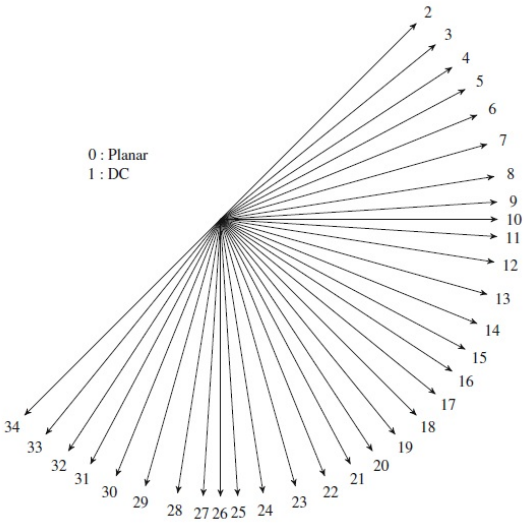


FIGURE 2.10: HEVC Intra Prediction Mode Directions [3]

### 2.4.5 Inter-picture prediction

In inter-prediction, temporally collocated blocks in neighboring frames are used to predict the block in the current frame. Neighboring frame is considered as the reference for the prediction of the block in the current frame. In HEVC, PUs in the inter-predicted CUs can use symmetric and asymmetric partitions sizes as shown in Fig. 2.7. Each PU is represented using a set of attributes that include motion vectors and index of reference frame. HEVC supports 1/2 pixel and 1/4 pixel interpolation for luma samples [3]. Interpolation filters with 8 and 7 taps are used for half pel and quarter pel interpolation, respectively. Integer and fractional sample positions for luma interpolation in HEVC is shown in Fig. 2.11. Motion vectors can be predicted from spatially and temporally collocated neighboring PUs [30].

Skip inter-prediction mode is used to infer motion attributes of the current block from the blocks in the reference frame. In HEVC, a special merge mode is also introduced where motion attributes of spatially and temporally collated neighboring PU blocks are used to signal the motion information of the current PU block [3]. Hence in HEVC, Skip mode is a special case of merge mode when coded residual of the PU block is zero. Skip mode is mostly selected as the optimal prediction mode for homogeneous regions in video frames with very little or no motion.

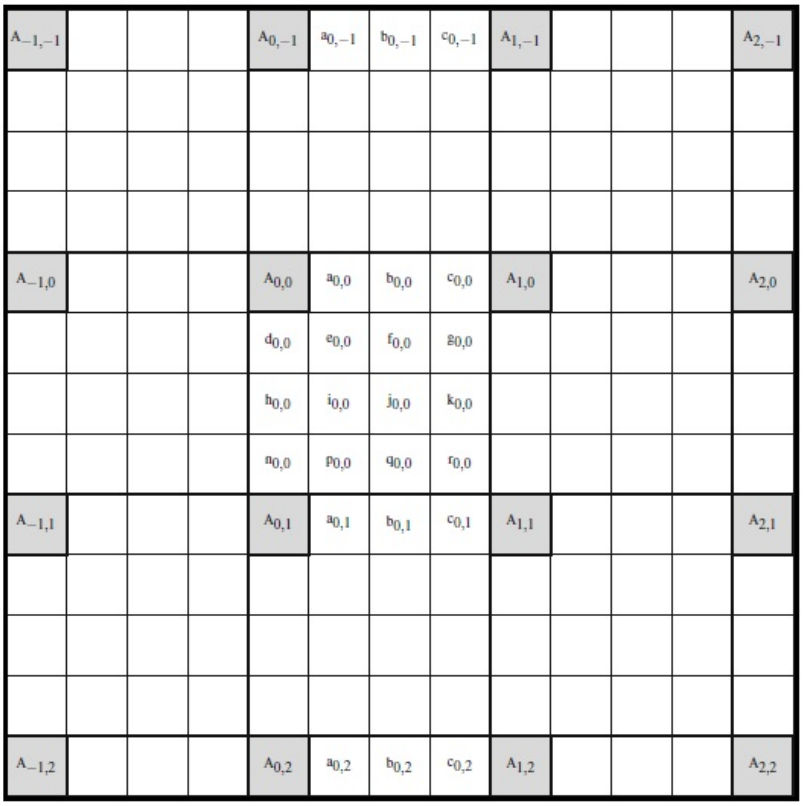


FIGURE 2.11: HEVC Luma Interpolation: Integer and fractional Sample Positions [3]

### 2.4.6 Transform and Entropy Coding

HEVC also uses discrete cosine transform (DCT) to encode the prediction residual. In the TU quad-tree structure, TUs are recursively divided into sub partitions. Size of TUs can vary from 32x32 down to 4x4. For 4x4 luma intra predicted TU blocks, a discrete sine transform is also incorporated [31]. After transformation, prediction residual coefficients and syntax elements are entropy coded. HEVC only uses context adaptive binary arithmetic coding (CABAC) for coding of prediction residual.

### 2.4.7 In-Loop Filtering

Loop filters are used in video codecs to remove the blocking artifacts introduced by the quantization and block prediction. HEVC also uses a deblocking filter in the

prediction loop [32]. Deblocking filter is applied on 8x8 boundaries only. In addition to deblocking filter, HEVC also uses a Sample Adaptive Offset (SAO) after the deblocking filter which is a nonlinear amplitude mapping to better reconstruct the original signal.

One of the issues with HEVC is the royalty and patent fees. Therefore, Google initiated the work on open-source and royalty free codecs by releasing a series of codecs named VP8, VP9 and VP10. Coding efficiency of VP9 is comparable to the HEVC. Later Google co-founded an alliance for open media (AOM) with a consortium of more than 30 tech companies. Open-source video codec AV1 was launched by AOM. Work on the successor of HEVC is also in progress and the codec is known as Versatile Video Coding (VVC). VVC is supposed to be finalized by the year 2020 and the goal is to provide better coding efficiency than HEVC.

### 2.4.8 Comparison of HEVC with other Video Codecs

During the evolution of video codecs, many new coding structures, prediction modes, filtering operations and entropy coding techniques have been introduced, as described in the previous section. These tools and techniques have increased the compression efficiency of the codecs. Table 2.2 gives a comparison of the coding tools used in HEVC with earlier video coding standards [33].

According to the table, new features introduced in HEVC are quad-tree structures for CU and TU blocks, extra intra and inter prediction modes and partition sizes and multiple filters. These features have improved coding efficiency of HEVC as compared to earlier standards like H.264 or MPEG4. However, computational complexity of HEVC has increased enormously. This complexity is a bottleneck in its widespread usage on the Internet and multimedia solutions.

Table 2.2: Comparison of HEVC with other Video Coding Standards

| **Features** | **MPEG-2** | **MPEG-4** | **H.264** | **HEVC** |
|---|---|---|---|---|
| Partition Size | MB(16x16) | MB(16x16) | MB(16x16) | CU(64x64 to 8x8) |
| Picture Coding Type | I,P,B | I,P,B | I,P,B | I,P,B |
| Intra-Prediction | DC Prediction | DC and Simple AC Prediction | 9 Prediction Modes | 35 Prediction Modes |
| Inter-Prediction | 16x16 | 8x8 to 16x16 | 4x4 to 16x16 | 8x4,4x8 to 64x64 |
| Entropy Coding | VLC | VLC | CAVLC, CABAC | CABAC |
| MV Resolution | 1/2 Pel | 1/4 Pel | 1/4 Pel | 1/4 Pel |
| Transform Unit | 8x8 DCT | 8x8 DCT | 4x4 DCT | 32x32 to 4x4 DCT,DST |
| Deblocking Filter | Out of Loop | Out of Loop | In-Loop | In-Loop , SAO Filter |
| Partition Block Size | Symmetric | Symmetric | Symmetric | Symmetric and Asymmetric |
| Multi Reference Prediction | 2 | 2 | 16 | 16 |
| Parallel Processing Tools | Slices | Slices | Slices | Wavefronts,Tiles,Slices |
| Interlaced Coding | Yes | Yes | Yes | No |
| Max Frame Rate (fps) | 15 | 30 | 59.94 | 300 |

### 2.4.9 HEVC Extensions

Many extensions have been added to the High Efficiency Video Coding (HEVC) standard to increase its capability, throughput and flexibility. HEVC Rext extension enabled coding of videos in monochrome 4:2:2 and 4:4:4 chroma formats. Other HEVC extensions include processing of 3D and multi-view videos, scalable video coding and screen content coding. Details of these extensions is provided following sub-sections.

#### 2.4.9.1 Multi-View and 3D Video Coding

In order to compress multi-view videos and depth based 3D videos, two HEVC extensions have been finalized. These are called Multi-view extension of HEVC (MV-HEVC) and 3D extension of HEVC (3D-HEVC) [34]. MV-HEVC exploits redundancy between the video feed of multiple cameras that capture the same scene information from different angles to compress it efficiently. Similarly, new tools have been introduced in 3D-HEVC to efficiently compress depth information in addition to 2D video. These tools exploit dependencies between video texture and depth maps. Both MV-HEVC and 3D-HEVC use a multi-layer coding structure that uses the redundancy between multiple layers to achieve higher compression. Multi-View and 3D coding has applications in stereoscopic displays and 3D TV.

#### 2.4.9.2 Scalable Video Coding

Scalable Video Coding extension of HEVC (SHVC) encodes video data according to the decoder configurations and network conditions [35]. SHVC can also assist in error resilience where data may be lost because of network errors. In SHVC, video data is encoded in multiple layers that consist of one base layer and multiple enhancement layers. SHVC is based upon a multi-loop coding architecture with support for inter layer prediction. Different scalability options available in SHVC include spatial scalability that deals with resolution of video frame, temporal scalability that is related to reducing the number of frames per second and

SNR scalability that is about changing the video quality and bit depth scalability such that the base layer has lower and enhancement layers have higher bit depth.

#### 2.4.9.3 Screen Content Coding

Screen content coding (SCC) extension of HEVC deals with efficient coding of computer generated graphics, computer animations and videos with mixture of camera captured content and computer generated graphics [36]. These videos often contain large uniform areas and repeated patterns with sharp edges and high contrast. Number of unique colors in screen content videos is limited as compared to the camera generated videos. HEVC SCC extension inherits many features of HEVC like CU quad tree structure and coding tools. For efficient coding of screen content, new tools have been introduced. These include Intra Block Copy (IBC), Palette mode, adaptive color transform and adaptive motion vector resolution [37–39]. Because of these added tools, coding efficiency of HEVC SCC extension is approximately twice as compared to the HEVC Rext for videos with computer generated graphics and text.

## 2.5 Challenges in Video Compression

With the increase in usage of multimedia and advancements in imaging technology, demand for video systems and video compression is continuously increasing. Services like high speed broadband, HD display technology, web-based video content and availability of low cost computing devices like smart phones and mobile Internet devices, are another driving force that require innovation and improvement in video compression techniques.

Video compression is being used in applications like video conferencing, high definition video storage in the form of DVD and Blu-Ray, security systems, computer vision, digital cinema and digital television. Video Encoding in latest video codecs

such as HEVC presents a challenging problem of providing high compression efficiency and magnificent video quality while fulfilling limitations of underlying hardware platforms.

Main challenges in video encoding using newest video codecs include:

### 2.5.1 Real-time Performance

With the increase in compression efficiency and complexity of video codecs, real time implementation of codecs like HEVC is one of the biggest challenges that designers are facing in video applications. High definition (HD) and ultra high definition (UHD) television broadcast is targeting FPS figures of 50, 60 , 100 upto 120. Applications like Video conferencing require minimum possible delay in video encoding and decoding in order to compensate for network delays. Video security applications target an end to end delay of less than 150 ms. Meeting these timing requirements even with fast processor architectures is a tough challenge to implement. Dedicated hardware platforms for video technology and intelligent encoding algorithms are being designed to meet these challenges.

### 2.5.2 Video Quality

Demand for high definition video in multimedia solutions is increasing. On-line video streaming websites like Youtube, Dailymotion offer upto 4K video resolutions. Video gaming consoles like Play station and Xbox provide video with 720p and 1080p resolution. Video conferencing applications provide 720p video with bit-rate of of upto 1 Mbps. Smart-phones have capabilities of capturing and playback of 720p and 1080p videos. Video surveillance and security cameras also offer HD resolution. Embedded devices used in these applications have to provide high quality video with strict budget constraints and limited resources. Development of intelligent algorithms that can maintain desired video quality and run these applications on hardware platforms meeting desired budget is a challenge for the designers.

### 2.5.3 Power Consumption

Power consumption is another important performance metric in embedded devices. When an embedded device provide support for high quality videos at higher frame rates, 3D graphic intensive games, it requires high performance processors, GPUs and large memory space which ultimately affects power consumption of the device. In order to extend battery life in embedded platforms, optimal and intelligent algorithms are required to reduce computational complexity of video encoding and decoding tasks.

### 2.5.4 Diverse Video Content

Content in video sequences is another major challenge for video coding standards. Video sequences vary both in terms of texture inside the video frame and motion content among multiple frames. These variations impact the prediction modes, partition sizes, motion estimation and ultimately the coding efficiency of the codec. Similarly, performance of the fast encoding algorithms to reduce the complexity of video codecs is also dependent on the video content. Complex variations in video content may impact the RD performance or the speed-up of fast encoding techniques used to speed-up video codecs.

### 2.5.5 Architecture of Video Codecs

To increase compression efficiency and to support higher frame resolutions, design of video codecs is getting more and more complex. In HEVC, frames are divided into variable sized blocks called CTUs. These CTUs are further divided into CUs and PUs with multiple depth levels and partition sizes. Transform operation is also done on variable size blocks. Multiple reference frames are used for motion search. With increase in number of options, partition sizes and prediction modes, encoding task becomes more and more complex. Real-time Implementation of these standards is challenging, because it requires efficient meta-data structures,

efficient usage of memory and optimal algorithms to reduce computational complexity.

Above mentioned challenges are the bottleneck in widespread implementation of latest video codecs. A great deal of research is being done on development of fast algorithms and design of dedicated hardware architectures to reduce the computational complexity of the video codecs to enable their usage in multimedia solutions. These fast algorithms often use machine learning based classification methods to take optimal decisions during the encoding process and hence, reduce the computational complexity of the video codec. Next section gives overview of popular machine learning methods being used in fast video coding.

## 2.6 Overview of Classification Methods

Classification and machine learning methods are employed in real-life problems to assist in decision making and find optimal solutions. These classification methods can be broadly divided in two categories [40]:

- Supervised Learning
- Un-supervised Learning

In supervised learning methods, data-set is explicitly labeled while un-supervised learning methods use unlabeled data sets. Each instance in the data set has multiple features or attributes that describe the data in different ways. Supervised learning techniques first learn from representative features of given labeled data set for a specific problem. This knowledge is used to predict and classify unseen data. Supervised classification algorithm can be described as a learning function 'f' that maps the input sample X to the output label Y according to the Eq. 2.5.

$$Y = f(X). \tag{2.5}$$

Comparative assessment of different classification methods is given in [41, 42]. Researchers have used classification algorithms in recent years to model complex video encoding tasks as a classification problem to speed-up the encoding process. These classification algorithms can early predict the optimal decisions and reduce the encoder complexity.

Supervised learning based classification methods start with collection of sample data set. Pre-processing is done on the data set to remove outliers and noisy instances. Feature sub-set selection is also performed to remove the irrelevant and redundant features. Feature sub-set selection reduces feature dimensions that helps in reducing complexity of the algorithm and efficiency of the classification problem. Resultant processed data set is partitioned into training and testing data. Training data is used to train the classifier and a classifier model is generated for given problem. Next the generated classifier model is employed to classify test data-set. Sample framework of classification based methods is shown in Fig. 2.12. Accuracy of the classification depends upon the relevance of extracted features, size of test data and model of the classifier while complexity of algorithm depends upon feature dimensions and classifier cost function.

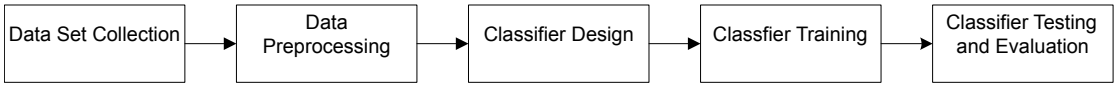


FIGURE 2.12: Sample Classification Framework

Different criteria are used to evaluate the performance of classification techniques. These include prediction accuracy, confusion matrices, region of convergence (ROC) curve and N-fold cross validation. Detail of these can be found in [43, 44].

Below is given brief overview of various classification methods found in literature related to fast video encoding.

### 2.6.1 Bayesian Classification

Bayesian classification forms a probabilistic classifier model that is based on Bayes Rule [45] as described in the Eq. 2.6.

$$P(C \mid X) = \frac{P(C)P(X \mid C)}{P(X)}, \tag{2.6}$$

where X represents the set of attributes and C represents the Class. Classification based on the Bayes theorem uses the maximum A posteriori (MAP) decision rule and can be formulated as

$$Arg_C[Max[P(C \mid X) = P(C)P(X \mid C)/P(X)]]. \tag{2.7}$$

For a given set of features, a test sample is assigned the class that has the maximum A posteriori probability. In the above expression, probability of the attributes P(X) does not change for different classes and can be considered a normalization constant.

Naive Bayes classifier [46] is a simplification in the Bayes classifier model that assumes, all the class attributes are independent of each other .

### 2.6.2 Classification and Regression Trees

Decision trees are used in machine learning to label data by sorting its feature values. Concept of classification and regression trees (CART) was introduced by L. Breiman [47] in 1984. Classification trees are applied to label the data based upon its feature values. On the other hand, regression trees are used for continuous target values and predict value of the target. An example classification tree is shown in Fig. 2.13 [48].

CART trees are generated using a set of questions and feature values. Tree generation starts at the root node. Each node is further split according to some rule using one of the features. Heuristics like information gain and gini-index [49] are used for selection of optimal feature to use for node split. Some stopping criteria

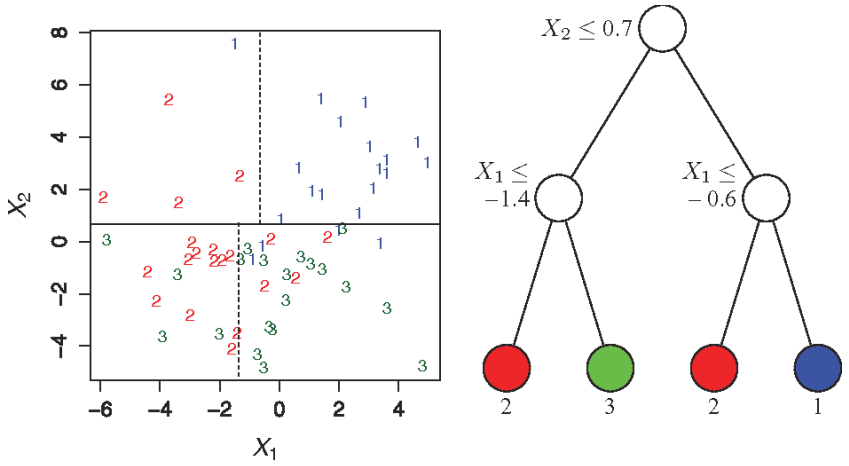


FIGURE 2.13: CART model

is also used to decide, when not to split a node further. This also helps in controlling the size of decision tree. Terminal nodes of classification trees represent class labels.

### 2.6.3 Support Vector Machines

Support vector machines (SVM) is another statistical algorithm for classification [50]. SVM based classifiers can be used to solve problems like object detection, text categorization and face recognition in machine vision. SVM uses the idea of constructing a hyperplane that maximizes the margins between data points of two classes [51]. Hyperplane is defined such that

$$w^T x_i + b \geq 1, \tag{2.8}$$

$$w^T x_i + b \leq -1, \tag{2.9}$$

where w is the weighted vector and b is the bias. SVM hyperplane and the margins are shown in the Fig. 2.14 [52].

SVM based classification uses the decision rule defined by the discriminant function in the Eq. 2.10.

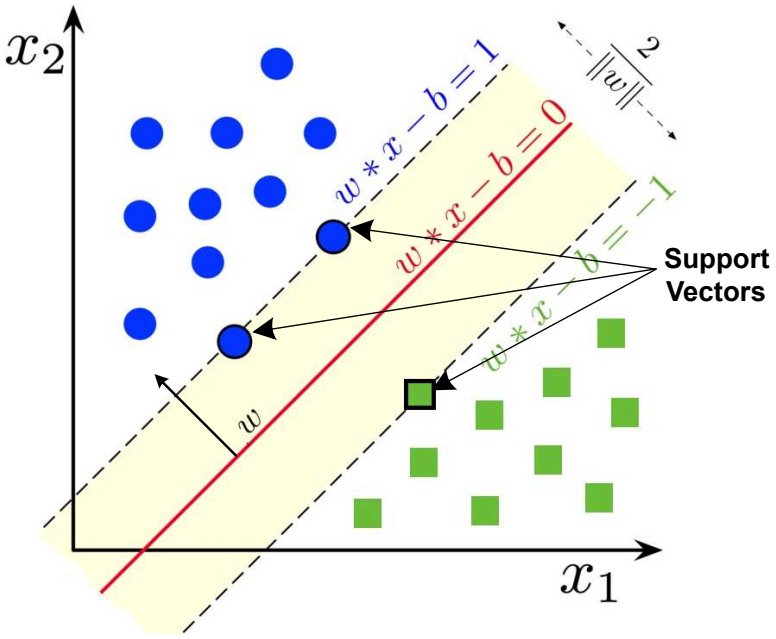


FIGURE 2.14: Margins and support vectors in SVM model

$$f(x) = w^T \Phi(w) + b, \tag{2.10}$$

where $\Phi(w)$ is a non-liner kernel function that maps the input $x_i$ into a higher dimensional kernel space. This hyperplane can be constructed by minimizing the cost function shown below.

$$J(w) = \frac{1}{2} \parallel w \parallel^2, \tag{2.11}$$

under the constraint

$$y_i(w^T \Phi(w) + b) \geq 1. \tag{2.12}$$

Common Kernel functions used in SVM are liner, radial basis functions (RBF) and sigmoid functions.

Majority of the classification algorithms used for fast video encoding in the literature are based on single classifier models. Very few implementations use ensemble classification methods that are more robust to changes in the video data.

## 2.7 Performance Evaluation Metrics for Video Coding

For performance evaluation of video coding standards, comparison of coding efficiency and assessment of optimizations done in video codecs, different subjective and objective evaluation matrices are used [53]. These matrices quantify the extent of distortions introduced in video frames because of quantization and predictive coding blocks or the modifications done in the video codec to reduce its computational complexity. Also relative speed-up attained because of the reduction in computational complexity can be computed for comparison of different fast coding methodologies.

### 2.7.1 Subjective Assessment

Subjective assessment of video quality relates to human visual system (HVS) and how a person perceives quality of video under test. One subjective assessment technique is known as mean opinion score (MOS) [54]. In MOS assessment, a panel of human observers assess the quality of video under test and rate it on a scale of 1 to 5. Here 1 refers to 'Bad' and 5 refers to 'Excellent' quality. Ratings assigned by multiple viewers are averaged to get the MOS figure. To get a reliable MOS figure, a panel of 20 to 25 human observers is required. MOS score of 3.5 or above is considered acceptable for many broadcasting services [53]. Such subjective methods of video quality assessment give reliable results despite being time consuming and expensive. Therefore, majority of research on video coding uses objective assessment measures.

### 2.7.2 Objective Assessment

Objective assessment methods of video quality rely on mathematical models. Block based motion estimation and compensation and lossy operations like quantization introduce distortions in encoded video frames. To measure these artifacts

in decompressed video data and to compare the video quality of different video compression techniques, a well-known quality assessment technique is Peak Signal to Noise Ratio (PSNR). PSNR depends on the Mean Square Error (MSE) between the blocks in original video frame and reconstructed frame during the encoding process. MSE is computed using Eq. 2.13 and PSNR is measured on a logarithmic scale according to the Eq. 2.14 [11].

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [Yref(i,j) - Yprc(i,j)]^2, \tag{2.13}$$

$$PSNR = 10 \log \left[ \frac{255^2}{MSE} \right]. \tag{2.14}$$

where $Yref(i,j)$ and $Yprc(i,j)$ are the pixel values of the blocks in reference frame and reconstructed frame, M and N are the block dimensions in terms of number of pixels, respectively. In the equation, the peak signal magnitude with an 8-bit pixel is 255, and the noise is the square of the difference (error) between the blocks in reference and reconstructed frame. PSNR values of 31 dB and above are considered acceptable for good quality videos [55]. PSNR is a simple and straight forward method of video quality assessment that is commonly used in video encoders for quality assessment and comparison.

### 2.7.3 Bjontegaard Delta Measurements

In order to compare the performance of two video encoding schemes, a method was proposed by Gisle Bjntegaard [56, 57] that is based on Rate Distortion (RD) curves as shown in Fig. 2.15. Proposed method calculates delta in overall bit-rate savings (BDBR) and difference in video quality (BD-PSNR) between the two encoding schemes by fitting a third order polynomial function. Rate distortion curves for two encoding schemes are computed at four different quantization parameter (QP) values. Interpolation is used to extend the curves and calculations are performed on logarithmic scale. For BDBR computation, difference in bit-rate

is computed along the horizontal scale for a given PSNR and for BD-PSNR, difference is computed along the vertical scale for a given bit-rate on the RD curve. Average BD-PSNR is computed as the approximate difference between the integrals of the two RD curves i.e. $D_1(r)$ and $D_2(r)$ divided by the integration interval between the upper $r_H$ and lower $r_L$ bounds according to the Eq. 2.15 [53]. Similar calculations are used to compute the average BDBR on a logarithmic scale according to the Eq. 2.16. Average BDBR percentage and BD-PSNR figures are used to evaluate the RD performance of video coding methods and their comparison with other similar schemes.

$$BDPSNR \approx \frac{1}{r_H - r_L} \int_{r_L}^{r_H} \left[ D_2(r) - D_1(r) \right] dr, \tag{2.15}$$

$$BDBR \approx 10^{\frac{1}{D_H - D_L} \int_{D_L}^{D_H} [r_2(D) - r_1(D)] dD} - 1. \tag{2.16}$$

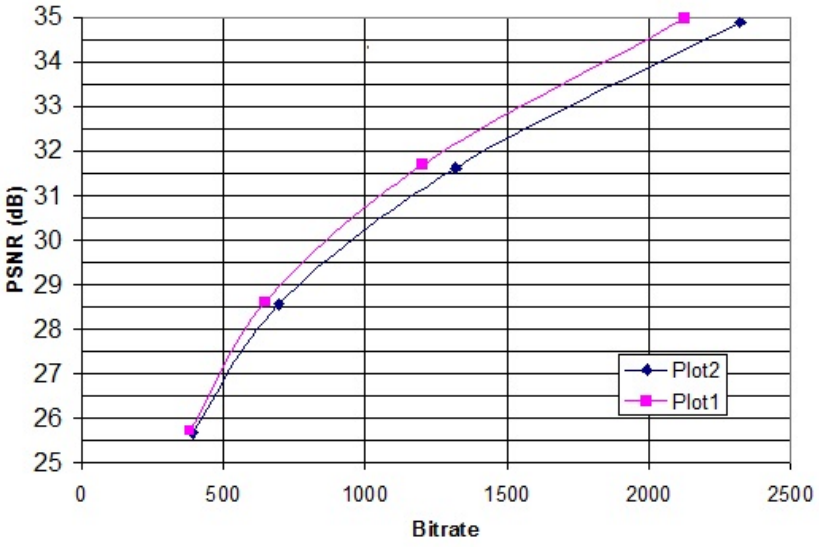


FIGURE 2.15: Example rate distortion (RD) curve between PSNR and bit-rate

### 2.7.4 Speed-up

In fast encoding implementations, reduction in computational complexity of the video codec is measured in terms of percentage increase in the processing speed of

the codec, as given by the Eq. 2.17 [58, 59],

$$TS(\%) = \frac{(T_r - T_p) * 100}{T_r}, \tag{2.17}$$

where TS% represents the percentage speed-up and $T_r$ and $T_p$ are the encoding times of the reference implementation and the proposed complexity reduction technique, respectively. Reduction in computational complexity of video encoding algorithm results in degradation in the RD performance. Hence, an increase in the speed-up will often lead to decrease in BDPSNR and increase in BDBR figures.

## 2.8 Data Set Characteristics

Data sets used to evaluate the performance of the proposed fast encoding techniques in this research consist of openly available video sequences and images. These video sequences and images have diverse characteristics both in terms of content and resolutions. Detail of these data sets is provided in the following subsections.

### 2.8.1 Video Coding Data Set

Video coding data set consists of 20 streams of various resolutions, diverse motion content and texture as shown in Table 2.3.

These video sequences contain high, low and moderate motion content as well as diverse texture and spatial information. These videos are divided into categories A, B, C, D and E based upon frame resolution and according to the common test conditions defined by ITU-T [60]. All the sequences are in YUV420 format with 8-bit pixel depth. Majority of fast video encoding implementations in literature use the same data set for comparison with other existing implementations.

TABLE 2.3: Video Coding Data-Set

| S.No. | Class | Stream | Resolution |
|---|---|---|---|
| 1 | Class-D | BasketballPass | 416x240 |
| 2 | | BlowingBubbles | |
| 3 | | BQSquare | |
| 4 | | Flowervase | |
| 5 | Class-C | BasketballDrill | 832x480 |
| 6 | | PartyScene | |
| 7 | | BQMall | |
| 8 | | RaceHorses | |
| 9 | Class-E | Johnny | 1280x720 |
| 10 | | Vidyo1 | |
| 11 | | Vidyo3 | |
| 12 | | Vidyo4 | |
| 13 | | FourPeople | |
| 14 | | KristenAndSara | |
| 15 | Class-B | BasketballDrive | 1920x1080 |
| 16 | | BQTerrace | |
| 17 | | Cactus | |
| 18 | | Kimono1 | |
| 19 | Class-A | PeopleOnStreet | 2560x1600 |
| 20 | | Traffic | |

### 2.8.2 Light Field Images Data Set

For light field image coding, 12 light field images of diverse texture and content have been used as shown in Table 2.4. This dataset is part of the Ecole Polytechnique Fdrale de Lausanne (EPFL) Light-Field Images Dataset and JPEG Paleno Database [61, 62]. All images are taken using a Lytro Illum B01 (10-bit) camera. Captured images were in raw format that were processed using MATLAB toolbox for light field images [63]. Resultant light field images were transformed into YUV format with resolution 9375x6510.

## 2.9 Summary

This chapter described building blocks of a video codec. Different video codecs developed over the years and coding tools introduced were also discussed and a comparison of these coding tools is given. Common machine learning based

Table 2.4: Light Field Images Data-Set

| S.No. | Category | Light Field Image | Resolution |
|---|---|---|---|
| 1 | Color Charts | Color Chart 1 | 9375x6510 |
| 2 | Color Charts | ISO Chart 12 | |
| 3 | Grid | Danger de Mort | |
| 4 | Mirrors and transparency | Vespa | |
| 5 | Nature | Flowers | |
| 6 | People | Fountain and Vincent 2 | |
| 7 | People | Friends 1 | |
| 8 | Studio | Ankylosaurus And Diplodocus 1 | |
| 9 | Studio | Desktop | |
| 10 | Studio | Magnets 1 | |
| 11 | Urban | Bikes | |
| 12 | Urban | Stone and Pillars | |

techniques used in fast video encoding were briefly introduced and various subjective and objective evaluation matrices to assess the performance of video coding standards and fast encoding techniques were discussed. Chapter also discussed different challenges in video coding. Finally, the chapter gave detail of the video data sets used in this research. In next chapter, various fast video encoding techniques available in literature will be discussed in detail.

# Chapter 3

# Literature Review

This chapter presents overview of different fast video encoding techniques and survey of the latest research done in the literature. Recent work in coding of light field images is also discussed. Limitations of earlier work are highlighted. Chapter also discusses Problem statement for the research in this thesis and the proposed methodology.

## 3.1 Fast Video Encoding Techniques

In majority of current video encoders, optimal prediction modes and partition sizes in a macroblock are selected based on rate distortion optimization (RDO) [14]. An exhaustive search method for selection of optimal modes, partition sizes and motion vectors is implemented that tries to minimize an RDO cost function and selects the option with minimum RD cost. This exhaustive search approach increases computational complexity of video encoders like high efficiency video coding (HEVC) that use the exhaustive search for selection of optimal depth levels in CU and TU quad-tree structures in addition to prediction modes and motion vectors.

Search for the optimal prediction modes, selection of optimal coding unit (CU) and transform unit (TU) depth levels and motion estimation are main complex blocks

in HEVC encoder [64]. To address this computational complexity, lots of research is being done to find intelligent algorithms that can reduce the computation time. Another challenge for fast video encoders is the content of video sequences. Many fast video encoding methods find it hard to adapt to the abrupt changes in the video texture or motion content. Some work is available in the literature on content adaptive methods for fast encoding. Researchers have proposed many fast encoding techniques for video codecs. These fast video encoding techniques can be categorized into the following main classes as shown in the Fig. 3.1.

- Statistics based Techniques
- Learning and Classification Techniques
- Content Adaptive Techniques

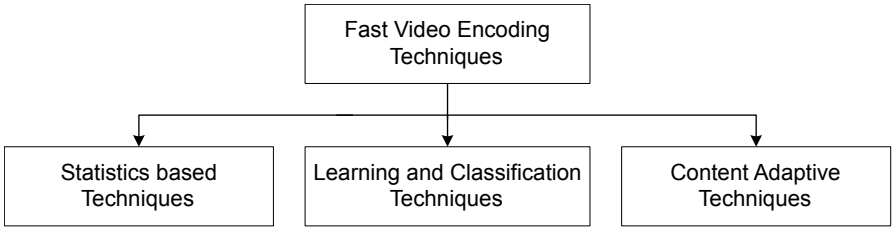


FIGURE 3.1: Fast video encoding methods

Statistics based techniques collect information about spatial and temporal neighboring blocks in a video sequences and use it to fix soft or hard thresholds for taking fast encoding decisions. Learning and classification based techniques are mostly employed in pattern classification, machine learning and signal processing applications to find optimal solutions of complex problems. These techniques first learn from representative features of a given data set. This knowledge is then applied to generate classifier models to predict and classify new data. Both online and offline training is used for generation of classifier models. Content adaptive methods are based on fast encoding schemes that adapt to the changes in the content of video sequences. This adaptability helps in improving the RD performance of fast encoding methods for a diverse set of video sequences.

Detail of these fast video encoding techniques is given in the following sections.

### 3.1.1 Statistics based Techniques

It is a well known fact that frames in a video sequences have high spatial and temporal correlations. Every macroblock or CTU within a frame resembles its spatial neighbors i.e. adjacent blocks within current frame and temporal neighbors i.e. co-located blocks in previous frames in texture detail and motion content. These correlations increases in frames with slow motion and smooth areas and decreases in regions with high motion content and more texture details. Because of these correlations, prediction modes, partition sizes, motion vectors and residual coefficients, a block tend to resemble its spatial and temporal neighbors. Similarly, depth level of a CTU in current frame in HEVC tends to resemble depth levels of its co-located CTUs in neighboring frames and adjacent CTUs in current frame.

Hence, information about prediction modes, block sizes and motion vectors of neighboring blocks in current and previous frames can be used to get highly probable prediction modes and block sizes for current block. This helps in avoiding exhaustive RDO computations on unnecessary prediction modes and block sizes or doing motion estimation searches in unnecessary regions of previous frames.

Researchers have explored many fast encoding techniques to speed up encoding in latest video codecs. Chae Rhee et al. [65] have surveyed many fast encoding algorithms for inter prediction and mode decisions in H.264. They have also analyzed the impact of using these algorithms in HEVC. Liquan et al. [66] proposed an adaptive inter-mode decision scheme for HEVC that exploits the correlations between spatio-temporal parameters of adjacent CUs and various CU depth levels to predict mode of current CU. An early Skip mode decision is made based upon correlation between Skip mode of neighboring CUs and Skip mode at different depth levels of current CU. Unnecessary prediction modes are avoided by defining a mode complexity parameter. Mode decision for current CU is made based upon correlation between RD cost of different CU depth levels and neighboring CUs using an adaptive threshold. Method showed less performance gain for video sequences with high motion content where complexity parameter used by the approach continuously falls in complex mode.

H. Zhang et al. [67] proposed micro and macro level approaches for fast encoding of intra prediction in HEVC. At macro level, they performed early termination of CU depth search based upon RD cost of sub CU blocks and avoid unnecessary computations. At micro level, they perform a progressive rough mode search based upon Hadammard cost to select an appropriate intra prediction mode and checks a small set of 11 intra modes. This avoids unnecessarily checking all 35 intra modes available in HEVC and results in computational efficiency. Approach used for early termination of CU depth search only works for intra CTUs and cannot be used for inter CU depth decisions with more probability of occurrence in an encoded video stream.

Sangsoo et al. [59] proposed fast inter prediction in HEVC based upon spatial parameters like sample adaptive offset (SAO) and temporal parameters like motion vectors, coded block flag (cbf) and transform unit size. SAO parameter values of reference block were used to estimate texture complexity in video frame. Motion complexity of current block was measured based upon motion vectors of 2Nx2N PU blocks. Extracted information was employed to intelligently predict early Skip mode and CU split decision. Similarly Jarno Vanne et al. [68] proposed three techniques for efficient inter prediction in HEVC that do efficient prediction of symmetric motion partition (SMP), selection of symmetric and asymmetric motion partitions SMP and AMP based upon quantization parameter (QP), and optimal SMP and AMP size selection based upon range limitations. Rate distortion complexity (RDC) of inter prediction in HEVC was used to select optimal partitions and mode decisions.

Liquan et al. [69] proposed a fast CU size decision algorithm and early termination of motion estimation schemes in HEVC. Their approach uses motion vectors, RD costs and depth levels of neighboring CUs and uses it to devise early termination schemes. These early termination schemes check motion homogeneity, Skip modes and RD costs to avoid motion estimation on unnecessary CU sizes. This results in speed up of inter prediction in HEVC. Thresholds employed in Liquans method are fixed and do not adapt to changing conditions in video scenes. Also this method uses spatial information of neighboring and collocated CUs so

for regions with abrupt scene change and high motion content this will result in performance degradation. Song-Hak Ri et al. [70] proposed a RD cost based approach to speed up inter mode decisions in H.264. They select two optimal best mode candidates from spatial neighboring blocks and temporal neighboring blocks respectively. Prediction mode with the lower RD cost is chosen as the appropriate member. To avoid error propagation because of mis-prediction they periodically use exhaustive mode decisions. Scheme was implemented in H.264 and cannot be directly extended to HEVC because block sizes and prediction modes in HEVC are different from H.264.

Xunhua et al. [71] proposed a fast CU depth prediction scheme for HEVC inter coding that uses CU depth values of spatial neighboring blocks in current frame and temporal neighboring blocks in collocated frame to predict the depth of current CU. RD cost of neighboring CUs and parent CU at lower CU depth levels is also employed to reduce the depth search range. B. Kim [72] proposed a fast algorithm for Skip or Merge mode prediction in HEVC for smart surveillance applications. The scheme used 9 neighboring CUs in the collocated frame and 4 CUs in the current frame for fast prediction of Skip or Merge mode during CU encoding. Optimal threshold was computed using the conditional probabilities of Skip or Merge mode and used to skip evaluation of unnecessary prediction modes. J. Wu et al. [73] have proposed a fast CU encoding scheme for HEVC based upon a set of three different constraints to select between best and second best PU modes to reduce search space in HEVC inter-coding.

T. Lin et al. [74] proposed two early termination algorithms for optimal CU depth selection using PU residual histogram in current CU and Skip information of neighboring blocks for early termination of CU depth search. Similarly, K. Tai et al. [75] proposed an early CU Split scheme that also skips unnecessary PU predictions to reduce computational complexity of HEVC. Their approach uses depth information of co-located CUs and RD cost correlation of different prediction modes for early termination of the CU Split procedure. F. Chen et al. [76] also proposed a fast inter coding algorithm for HEVC. Their scheme first divides the CU into static or motion block, then spatio-temporal correlations are

used to determine the CU depth range and prediction mode. Hoyoung et al. [77] proposed an early Skip mode decision scheme, with emphasis on video quality, using thresholds on the RD cost of merge mode for early decision of Skip mode. Similarly, [78–81] have used fast encoding approaches for mode decisions in HEVC, based on spatio-temporal neighbors.

These statistics based techniques use the behavior of spatial and co-located neighboring blocks for prediction and speed-up of encoding process and are often based on fixed hard or soft thresholds for prediction and exit strategies. These schemes relay on local features of neighboring blocks in a video frame for decision making. Hence, their performance is susceptible to scene changes and do not easily adapt to the inherent variability of video content.

### 3.1.2 Learning and Classification Techniques

Researchers have applied classification algorithms to model critical video encoding decisions as a classification problem and find optimal solutions. Various classifier based approaches have been used for fast encoding in latest video codecs. Xiaolin Shen et al. [82] proposed a statistical approach for fast encoding of HEVC. They have modeled the CU size decision as a classification problem with two classes i.e. split and non-split. CU split decision is taken based upon Bayesian decision rule. A set of feature vectors is calculated for each CU and split decision is taken using class conditional probability functions (pdfs) of respective classes and minimizing Baysian risk. This classification based approach results in reduction of computational complexity. Approach requires calculations of class-conditional probabilities using non-parametric estimation which requires excess training data, also it requires various pre-computed thresholds for different frame resolutions in Bayesian decision rule which becomes a limitation with change in data sets.

Jian Xiong et al. [83] proposed a pyramid motion divergence (PMD) based approach for CU size selection in HEVC. Motivation for this approach is, higher the degree of motion in a video region, it is highly likely that region will be encoded

using small CU sizes. Optical flow is estimated for down-sampled version of each frame. PMD is estimated for each CU as variance of the optical flows of current CU and sub CUs. Next a k-nearest neighbor (kNN) based classifier is used to take the split decision for each CU as compared to neighboring CUs. Optical flow calculations and PMD computations at pixel level over the entire frame is a computation overhead in this approach.

Jui Chiang et al. [84] proposed a two stage hierarchical neural classifier for fast mode decision in H.264 stereo coding. First stage of neural network predicted probable block partitions while second stage neural network was used for reference frame selection. Features extracted from temporal and neighboring view frames were fed to the Neural network. Similarly, Eduardo Martinez et al. [85] proposed a classification based approach for inter-prediction in H.264 that works in two levels. First classifier takes a decision and selects Skip mode for P Slides and/or DIRECT mode for B slices. If this decision is not taken, second classifier chooses between large and small block sizes for inter-prediction. Both classifiers are based on support vector machines (SVM). Classifiers work in binary mode and use a set of features extracted from video data to take early decisions. Chen Kuo et al. [86] proposed a statistical approach for fast encoding in H.264. SVM is employed in inter-mode decision, reference frame selection for motion estimation (ME) in multiple frames and intra mode decision. Representative features are selected from video frames to train SVM based classifiers in offline mode. Next trained classifier is used in H.264 for fast encoding in run-time. All these schemes were proposed for H.264 and its variants and cannot be directly applied to HEVC because macroblock structure and inter prediction in HEVC is quite different and complex as compared to H.264 with multiple depths and partition sizes.

Yun Zhang et al. [87] proposed a machine learning based approach for fast CU depth decision in HEVC. They model CU depth decision as a three level hierarchical binary decision problem. They have selected SVM based approach for classification. A three output joint classifier is designed to manage false predictions. The approach also employed feature selection optimization. Output of this

approach highly depends on prediction accuracy and right feature selection. Errors in feature selection introduce RD degradations. M. Grellert et al. [88] also used SVM for fast CU partition decisions in HEVC that uses offline training to model the classifier, while H. Kim et al. [89] used online learning based on a Bayesian decision rule, also for fast CU partitioning. Similarly, L. Zhu et al. [90] have implemented multiple binary and multi-class SVM classifier models to speed-up the CU Split decision and prediction mode selection, respectively. They also employed a combination of online and offline trained models to achieve better classifier accuracy. X. Shen et al.[91] also employed weighted SVM for early CU split termination in HEVC. In this case the weights are used to reduce misclassification in the SVM model.

A data mining approach based on decision trees is used in [92] to allow early termination of decision processes that find the best CUs, PUs and TUs. A separate decision tree is applied for each case and their fixed structure does not allow any form of adaptability to different types of content. Furthermore, using single decision trees may lead to over-fitting during the training phase. D. Ruiz et al. [93] have also employed decision trees to speed-up HEVC intra coding. Bochuan Du et al. [94] used Random Forests to speed up intra prediction in HEVC. First CU depth search is reduced by selecting a range using neighboring CU depth levels. Then a Random Forests classifier takes a Split vs. Non-Split decision. The authors have used the individual pixel values as feature vectors for the RF classifier, which may contribute to expand the feature set because of the limited information gain of individual pixels.

Table 3.1 gives summary of the resent research in the area of fast coding of video data. A common aspect of earlier research using machine learning approaches is that most of them use offline training and are based on a single classifier. These offline-trained classifiers do not easily adapt to changing conditions in different type of video sequences. Therefore, feature selection becomes highly critical and this is the reason for the majority of machine learning approaches to compute a large feature set from video data. Thus, some ranking mechanism should be used to short-list the most appropriate and effective features to improve the classifier

TABLE 3.1: Summary of research work in fast video encoding methods

| Title | Methods | Reference |
|---|---|---|
| Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations | Statistics based Methods | Liquan et al.[66] |
| Efficient mode decision schemes for HEVC inter prediction | Statistics based Methods | J. Vanne et al. [68] |
| A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding | Statistics based Methods | Sangsoo et al.[59] |
| Fast encoding algorithm for high-efficiency video coding (HEVC) system based on spatio-temporal correlation | Statistics based Methods | JH. Lee et al.[80] |
| A fast HEVC inter CU selection method based on pyramid motion divergence | Motion Pyramids-Optical Flow | J. Xiong et al.[83] |
| A fast inter coding algorithm for HEVC based on texture and motion quad-tree models | Statistics based Methods | F. Chen et al. [76] |
| A Fast HEVC Encoding Method Using Depth Information of Collocated CUs and RD Cost Characteristics of PU Modes | Spatial-Temporal Methods | K. Tai et al. [75] |
| A Fast H.264 / AVC-Based Stereo Video Encoding Algorithm Based on Hierarchical Two-Stage Neural Classification | Neural Networks | J. Chiang et al. [84] |
| A two-level classification-based approach to inter mode decision in H.264/AVC | SVM | E.Martinez et al. [85] |
| Machine Learning Based Coding Unit Depth Decisions for Flexible Complexity Allocation in High Efficiency Video Coding | SVM | Y. Zhang et al. [87] |
| CU splitting early termination based on weighted SVM | SVM | X. Shen et al.[91] |
| Fast HEVC Encoding Decisions Using Data Mining | Decision Trees | G. Correa et al.[92] |
| Fast CU Partitioning Algorithm for HEVC Using Online Learning Based Bayesian Decision Rule | Bayesian Decision Rule | H.Kim et al. [89] |
| A unified architecture for fast HEVC intra-prediction coding | Decision Trees | D.Ruiz et al. [93] |

accuracy. Although, some techniques use online training or cost functions based upon misclassification error to improve classifier performance, using a single classifier may result in over-fitting of the classifier model. This leads to either poor performance in terms of speed-up for low motion video or poor performance in terms of quality for high motion videos. Moreover, computation of cost functions for classifier accuracy adds to the overall computational complexity. Very few techniques available in the literature employ ensemble classifiers. This may be due to the common understanding that ensemble classifiers are more computationally intensive. However, ensemble classifiers have the unique advantage of diversification to avoid classifier over-fitting and result in better performance [95] besides being quite computationally efficient [96].

## 3.2 Content Adaptive Techniques

Content in video sequences often has large variations. These variations include changes in motion content, abrupt scene changes and texture variations in video frames. This diverse content make task of fast encoding techniques more challenging because fast encoding methods do not perform well for complex video sequences. HEVC reference, HM evaluates all prediction modes and exhaustively searches all the CU depth levels to select optimal prediction modes and CU depth levels. Both of these tasks have a major contribution in increasing the computational complexity of HEVC [97].

Various techniques for fast encoding of HEVC [71, 72, 98] have been proposed by researchers in literature. These schemes use spatio-temporal features for fast HEVC encoding and are based on precomputed fixed thresholds for early prediction of CU depth levels and prediction modes. These schemes rely on local features of neighboring blocks in a video frame for decision making and therefore do not adapt well to changes in the video content. Machine learning based approaches [90, 92, 99] have also been used to reduce the computational complexity of HEVC in the literature. Majority of existing methods used for fast video coding are not

content adaptive and compare parameters extracted from neighboring blocks to precomputed thresholds to predict behavior of the current block during encoding. These comparisons define early termination conditions to speed-up encoding in video streams of different resolutions and video content. Hence, these implementations often do not adapt when motion content of video stream or texture of video frame changes.

Some work has been done on content adaptive approaches for fast encoding in HEVC. G. Tian et al. [100] have used content adaptive approach for prediction unit size decision in HEVC intra coding, that works in two stages. In the first stage, texture complexity of down-sampled LCU helps in filtering out unnecessary PU sizes. In the second stage, PU sizes of neighboring encoded PUs are used to short list candidate prediction units. HR Tohidypour et al. [101] have proposed an adaptive scheme for motion search range reduction and early prediction of intra and inter modes in 3D HEVC. Disparity between base and dependent views is employed for early selection of optimal prediction modes. J. Leng et al. [102] have proposed a fast CU decision algorithm that computes CU utilization at frame level and neighboring CU information at block level to skip rarely used CUs. Approach uses parameters extracted from neighboring blocks in a frame to compare against fixed thresholds for fast prediction. C. Blumenberg et al. [103] have also proposed a content adaptive scheme for tile partitioning in HEVC to reduce coding losses because of tile boundaries selection. DG. Fernandez et al. [104] proposed a smooth region classification algorithm in video frames. Based upon this classification, time complexity because of search of optimal CU size selection, intra and inter prediction modes decision was reduced in HEVC. According to their scheme, if current CU block falls in the smooth region, it is not further split to reduce the computational complexity. Similarly, number of evaluated intra and inter prediction modes are reduced for CUs in the smooth region. Methods based on classification have also been used in [105, 106] to reduce the computational complexity of HEVC.

Table 3.2 gives summary of the resent research in the area of content adaptive coding of video data. Majority of these content adaptive techniques rely on local

TABLE 3.2: Summary of Literature in Content adaptive fast video encoding

| Title | Methods | Reference |
|---|---|---|
| Content adaptive prediction unit size decision algorithm for HEVC intra coding | Content Adaptive HEVC Intra Coding | G. Tian et al. [100] |
| A content adaptive complexity reduction scheme for HEVC-based 3D video coding | Content Adaptive Search Range Selection for 3D HEVC | HR Tohidypour et al. [101] |
| Content based hierarchical fast coding unit decision algorithm for HEVC | Content Adaptive CU Size Selection | J. Leng et al. [102] |
| Adaptive content-based Tile partitioning algorithm for the HEVC standard | Content Adaptive Tile Partitioning | C. Blumenberg et al. [103] |
| Complexity reduction in the HEVC / H265 standard based on smooth region classification | Content Adaptive Smooth Region Classification | DG. Fernandez et al. [104] |

parameters to adapt the encoding of blocks in a video frame. No global information at frame level is calculated to characterize the content in video sequences. Hence, adaptability of these techniques suffer loss in RD performance or encoder speed-up for video sequences with complex spatial and motion content.

## 3.3 Fast Coding of Light Field Images

Light field images represent the visual information in a scene as a function of position and angle of light rays [107, 108]. Hence, light field images contain more information as apposed to conventional 2D images. Because of this additional information, light field images contain huge amount of data and its compression and storage is a challenge. Researchers are experimenting with various light field representation formats with the objective of improving the coding efficiency of standard encoding techniques for light field imaging. I. Viola et al. [109] have compared the performance of different light field coding algorithms using objective and subjective quality assessment for lenslet and 4D light field images. Their study concludes that compression of 4D light fields performs significantly better

than lenslet images. Y. Li et al. [110] have used a displacement intra prediction scheme based on HEVC for coding light field images. Their scheme achieves 60% bit-rate reduction as compared to Intra coding of HEVC. L. Li et al. [111] and C. Perra et al. [112] have proposed different methods based on a pseudo sequence to represent light field images and use HEVC for compression. They first arrange the Light field image views as a pseudo temporal sequence and use HEVC for compression. Waqas et al. [113] have represented sub-aperture light field images as multi-view sequences and multi-view extension of HEVC has been used for efficient compression. A 2D prediction and rate allocation scheme has been employed to achieve higher compression. Their approach gives a higher PSNR for compression of light field images as compared to the compression achieved with HEVC. In general, light field data formats have significant impact on coding efficiency. For instance, it was found by A. Vieira [6], that higher compression ratios are obtained when encoding light fields as sub-aperture images rather than using lenslet format, because sub-aperture images can be organized temporally into a pseudo video sequence and video codec like HEVC can be applied for compression. Similarly, [114–116] have also used pseudo sequence of sub-aperture images along with HEVC for efficient compression of light field images.

Table 3.3 gives summary of the resent research in the area of light fields coding. Most compression schemes for light field imaging investigated so far, such as the ones cited above, are based on HEVC standard coding techniques. In addition to this, majority of research in light field images is focused on achieving higher compression for huge amount of data in the light fields. Using HEVC for coding of light fields gives better coding efficiency than still image coding standards like JPEG. Nevertheless, it also results in increase in the computational complexity of the encoding process. In this context, JPEG Pleno is working on standardization activities to define a new compression standard that aims to not only achieve efficient coding of light field images, but also low computational complexity [117, 118]. Many implementations in the literature [78, 104, 119] are available on fast coding of video data using HEVC. However, to the authors knowledge there are

TABLE 3.3: Summary of Literature in Coding of Light Field Images

| Title | Methods | Reference |
|---|---|---|
| Comparison and evaluation of light field image coding approaches | HEVC based Compression of Light Fields | I. Viola et al. [109] |
| Coding of focused plenoptic contents by displacement intra prediction | HEVC based intra coding of Light Fields | Y. Li et al [110] |
| Pseudo-Sequence-Based 2-D Hierarchical Coding Structure for Light-Field Image Compression | Pseudo Sequence based coding of Light Fields | L. Li et al. [111] |
| High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement | Pseudo Sequence based coding of Light Fields | C. Perra et al. [112] |
| Interpreting plenoptic images as multi-view sequences for improved compression | Multi-view coding of Light Fields | Waqas et al. [113] |
| Data formats for high efficiency coding of Lytro-Illum light fields | HEVC based coding of Light Fields | A. Vieira et al.[6] |

no research studies related to low-complexity or fast coding of light field images, neither benchmark results to enable evaluation of new research in this area.

## 3.4 Gap Analysis

Majority of work done for fast video encoding is based on statistics based techniques. These techniques often use soft or hard thresholds to predict block type, prediction modes and motion vectors of current block. On the other hand, classification based techniques require extraction of relevant features and training of classifier. Training of classifier is computationally intensive but can be done offline.

Main limitations of earlier literature are summarized below:

- Despite higher compression efficiencies and all the work done on fast encoding methods, HEVC codec is still not in widespread usage in main-stream video applications due to the higher computational complexity of the codec

that limits its real-time implementation in multimedia devices. Fast implementation of HEVC with acceptable RD performance is still an open research problem.

- Statistics based approaches for fast encoding, widely available in literature do not require any training albeit being data dependent because thresholds are often computed using experimentation with different video streams. Outcome of these techniques is highly susceptible to sudden scene changes, high motion content and inhomogeneous regions in video sequences.

- Little work is available in the literature on content adaptive fast encoding of video sequences. Majority of existing approaches are susceptible to changes in video texture or motion content. Existing methods usually rely on local parameters in a video frame and with changes in video content, these methods either compromise on encoding speed or the RD performance.

- Majority of existing classification based methods employed in fast video coding are based on single classifier models and use offline training. These techniques do not easily adapt to changing conditions in video scenes. Therefore, selection of features becomes highly critical. Very few techniques in literature use ensemble classifiers like Random Forests or Adaboost. Similarly, work done in literature on classification based approaches on earlier video standards like H.264 cannot be directly extended to HEVC because of many differences in coding structure and prediction modes.

- Latest research on light field images is focused on compression efficiency and HEVC is considered as the codec of choice. However, no contribution in the literature is available that addresses the computational complexity of HEVC for encoding light field images. Consequently, no benchmark results are available for comparative analysis of fast light field encoding techniques.

## 3.5 Statement of Problem

Latest video codecs have high computational complexity despite being compression efficient. This complexity results in implementation challenges when these codecs are used in various multimedia solutions. These challenges are the major bottleneck in widespread usage of newest video codecs like HEVC. Majority of research done on fast video encoding either uses statistics based methods or classification techniques using single classifiers. These solutions do not adapt to the spatial and temporal variations in video content and compromise either on speed or the RD performance.

Hence, there is need for robust fast encoding methods using intelligent algorithms with low computational overhead to reduce the computational complexity of HEVC. Such algorithms should adapt to the variations in video content and give speed-up without compromising on RD performance and should result in generalizable solutions that can also be applied to future image and video coding applications.

## 3.6 Proposed Research Methodology

Focus of this research work is the investigation of fast algorithms and intelligent techniques to reduce the computational complexity of HEVC. Publicly available data sets have been used for analysis of video data. Simulations are run using the data set to identify the complex blocks in HEVC encoder. A large set of spatial and temporal parameters is identified in video data that can be used to correctly predict the behavior of current block in the video frame and characterize the nature of video content. Preprocessing of this large feature set is carried out to remove irreverent features and select the optimal ones. This preprocessing helps in improving the performance of classifier models and reduction in overhead of the fast encoding method. Resultant optimal features are used to train the classifier model and the model is integrated in HEVC to intelligently predict and

simplify the critical tasks to reduce the computations of complex encoding blocks. Before integration, classifier models are evaluated for accuracy using offline data extracted from video sets. After integration in the video codec, performance of the fast encoding method is evaluated based upon the speed-up of encoding process and RD performance of the resultant encoded stream.

Proposed research methodology is shown in the Fig. 3.2. Multiple iterations of this process have been carried out to refine the classifier model and improve the performance of the fast encoding technique. Using this methodology, a computationally efficient fast encoding framework is developed that adapts to the changes in video content without sacrificing the video quality or bit-rate and performs better than other state of the art implementations. Proposed fast video encoding solution can also be extended to other emerging applications of video codecs such as coding of light field images.

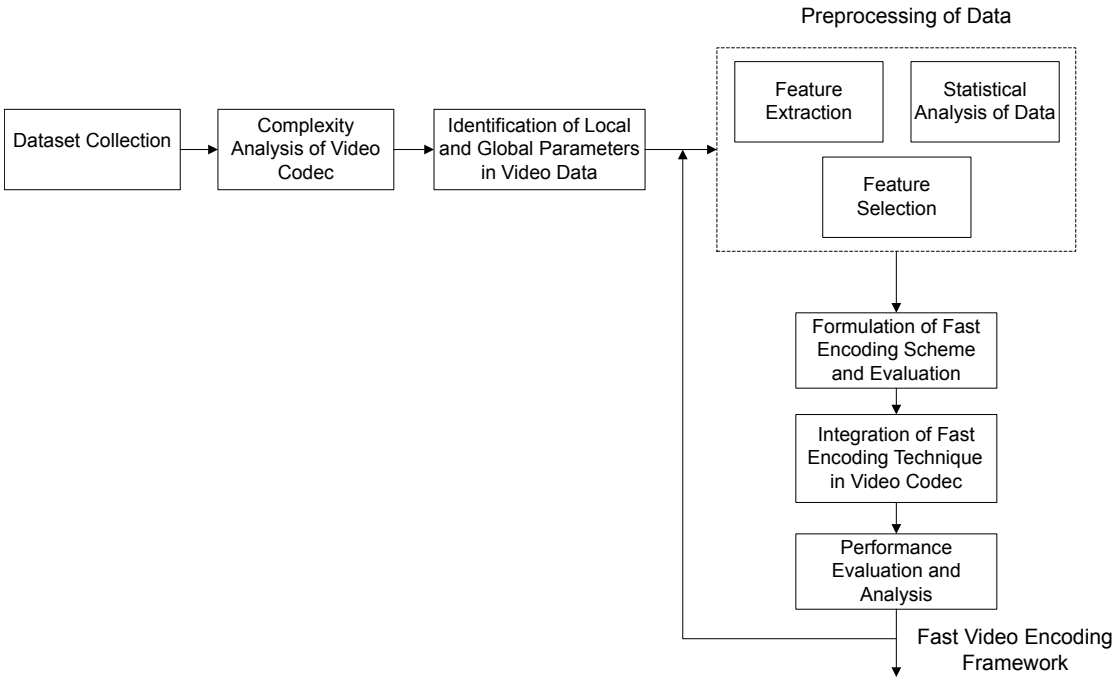


FIGURE 3.2: Proposed research methodology

## 3.7 Summary

This chapter described the recent literature on various fast video encoding techniques. The literature survey was categorized into statistics, classification based methods and content adaptive fast coding. Recent research in coding of light field images was also discussed. Limitations of earlier work were highlighted and gap in the literature was identified. Based upon these limitations, problem statement of this research was formulated and proposed research methodology was outlined. In the following chapters, main contributions of the proposed work will be described.

# Chapter 4

# Content Adaptive Fast Video Encoding

This chapter presents the first contribution of the research work. A content adaptive fast video coding technique is proposed. Spatial and temporal parameters from video data are extracted that give information about the type of video content. These global parameters are used to categorize video frames into simple or complex category content. Accordingly, content specific schemes for fast HEVC encoding have been implemented. Proposed scheme shows promising results for video data with diverse content.

## 4.1 Overview of HEVC Complexity

HEVC gives 50% more coding efficiency as compared to H.264. To achieve this compression efficiency, many new features and coding tools have been introduced in HEVC as discussed in Chapter 2. These tools include quad-tree structure of CUs and TUs, that are different from MBs used in earlier standards. Size of CUs in the quad-tree structure can range from 64x64 to 8x8. These CUs exist at 4 CU depth levels. A similar quad-tree structure exists for TUs. Size of variable size TUs can range from 32x32 to 4x4 at four depth levels.

Similarly, new intra and inter prediction modes and partition sizes have been introduced in HEVC. Intra prediction in HEVC has been increased from 9 modes in H.264 to 35 modes. New PUs and partition sizes have also been introduced including non-symmetric partitions. These prediction modes and partition sizes are evaluated at each CU depth level. HEVC reference software HM, exhaustively checks all the CU depth levels and prediction modes for selection of the optimal ones. For example, 85 different CU size combinations at multiple depth levels are evaluated to select the optimal CU size during encoding of one CTU. And for each CU depth level, all the possible prediction modes and partition sizes are evaluated. This makes the HEVC encoding task many times more complex [120]. Encoding complexity of different blocks in HEVC encoder for Low delay and Random Access profile are shown in Table 4.1 [121]. Average complexity of different encoder blocks for both the profiles is shown in the Fig. 4.1. According to these stats, inter prediction block that includes selection of optimal depth level in CU quad-tree structure, motion estimation, selection of optimal inter prediction mode and partition size, is the most complex block in HEVC encoder and accounts for more than 80% complexity.

TABLE 4.1: Percentage Complexity of different Blocks in HEVC Encoder [121]

| Encoding Block | RA Profile (%) | LD Profile (%) |
|---|---|---|
| Entropy Coding | 2.98 | 2.4 |
| Intra Prediction | 2.25 | 1.95 |
| Inter Prediction | 79.03 | 82.23 |
| Transform and Quantization | 14.48 | 12.5 |
| De-blocking Filter | 0.08 | 0.08 |
| Sample adaptive offset Filter | 0.1 | 0.1 |
| Others | 1.28 | 0.93 |

## 4.2 Content Adaptive Fast Encoding Techniques

Video data has large variations both in texture complexity and motion content. Texture in video frame may change from smooth to complex among different frames in a video or among various video sequences. Similarly, motion content

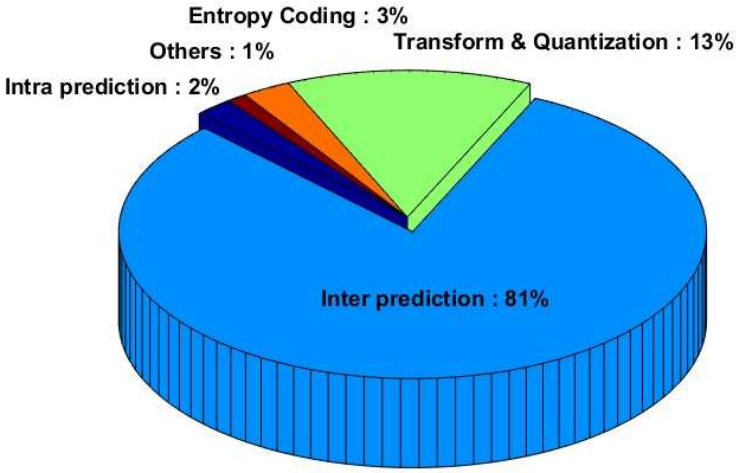


FIGURE 4.1: Complexity Analysis of HEVC Encoder

and the frequency of dynamic scene changes also vary from one video sequence to another. These variations make the task of optimal prediction mode and appropriate CU size selection in HEVC highly complex. Majority of fast video encoding techniques described in Chapter 3 often use fixed thresholds for early selection of prediction modes and CU sizes. These thresholds are often based upon local block level parameters and do not adapt to the changes in video texture or motion content. Therefore, such techniques result in poor RD performance or little speed-up gains during fast video encoding.

Hence, researchers are working on content adaptive techniques to adapt the fast encoding methods to the variations in the video content. These approaches select thresholds based on video data analysis that can be updated with change in texture complexity of video frame or motion content. Some methods in literature also update the threshold values at regular intervals after a fixed number of video frames to handle the variations within the video sequence. Similarly, machine learning based methods used for fast encoding sometimes use content specific feature selection. On-line training of classifier models has also been used to handle the variations in video content among different video sequences. All these techniques extract certain parameters from video data to get information about video characteristics. These extracted parameters are used to adapt the fast encoding methodology to the video data.

## 4.3 Spatial and Temporal Parameters

Video data has high spatial and temporal correlations. These correlations are high for videos with large smooth regions and slow motion content. On the other hand, video sequences with complex texture and high motion content have relatively less correlation. To access the texture complexity of a block in the video frame, information about the spatial neighboring blocks can be used. Similarly, temporal neighboring blocks in the collocated frame can be used to access the motion content in the block being encoded. Spatial perceptual information (SI) and temporal perceptual information (TI) measures represent the spatial details and temporal changes in video sequences [122]. These parameters are computed at frame level in video sequence.

### 4.3.1 Spatial Perceptual Information (SI)

Spatial Perceptual Information (SI) shows the spatial detail and texture complexity in a video frame. SI parameter is based on the Sobel filter [123]. Sobel filters are used for edge detection and texture orientation in images and video frames. Luma component of each frame in the video sequence is filtered using Sobel filtering and standard deviation of the Sobel filtered frame is computed. This process is repeated for all the frames in the video sequence. Finally, SI is computed as the maximum of the standard deviations of all the Sobel filtered frames over the whole sequence as shown in the Eq. 4.1.

$$SI = Max_{time}[Std_{space}[Sobel(F_n)]]. \tag{4.1}$$

Video sequences with large smooth regions inside video frames will have low SI value. On the other hand, complex texture and high spatial detail in video frames results in high SI values.

### 4.3.2 Temporal Perceptual Information (TI)

Temporal Perceptual Information (TI) shows the temporal changes and variations in motion content in the video sequence. For computation of TI, difference between the pixel values of luma component in consecutive frames in video sequence is computed. This is known as the difference frame. Standard deviation of the difference frame is used in computation of TI. This process is repeated for all the consecutive frame pairs in the video sequence. TI is equal to the maximum of standard deviation of all the difference frames in whole sequence as shown in Eq. 4.2.

$$TI = Max_{time}[Std_{space}[D_n\ (i,j)]], \tag{4.2}$$

where

$$D_n\ (i,j) = F_n\ (i,j) - F_{n-1}\ (i,j). \tag{4.3}$$

Video sequences with static regions and low motion will have small TI values. Similarly, large TI values represent high motion content in video sequences.

Both SI and TI are global parameters that can be used to quantify the spatial and temporal changes in the video sequence.

### 4.3.3 Otsu Thresholding

Binarization is used in image segmentation to differentiate the foreground from the background. Thresholding is used to convert gray level images into binary images. Objective of an efficient binarization technique is to find an optimal threshold value for binarization. Otsu thresholding is a well known algorithm for image binarization that can be used for segmentation in images and video frames [124]. Foreground and background are considered as two classes for image binarization.

First step in Otsu thresholding is computation of image gray level histogram. Than, Otsu method searches for an optimal threshold by checking all the gray levels in the image histogram as threshold values. Threshold that maximizes the inter class variance is considered the optimal threshold as shown in the Eq. 4.4 [125].

$$T_{th} = max[\omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2], \tag{4.4}$$

where

$$\omega_0(t) = \sum_{i=1}^{t} P(i), \tag{4.5}$$

$$\omega_1(t) = \sum_{i=t+1}^{L} P(i), \tag{4.6}$$

$$\mu_0(t) = \frac{\sum_{i=1}^{t} i * P(i)}{\omega_0(t)}, \tag{4.7}$$

$$\mu_1(t) = \frac{\sum_{i=t+1}^{L} i * P(i)}{\omega_1(t)}. \tag{4.8}$$

$P(i)$ is the probability of gray level in the image histogram. $\omega_i(t)$ is the probability of class occurrence and $\mu_i(t)$ is the average level of a specific class.

Otsu method shows good performance for the images with bi-modal gray level histograms. Histogram of the 'Newscaster' image and corresponding binarized image using Otsu thresholding are shown in the Fig. 4.2.

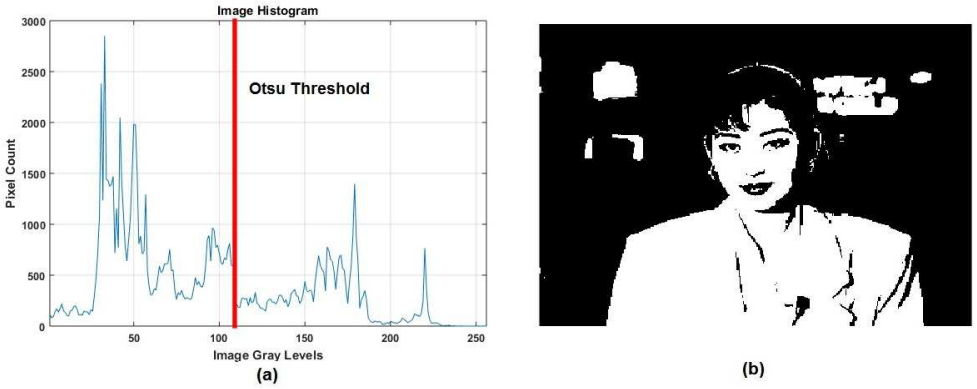


FIGURE 4.2: Otsu Thresholding (a) Histogram of 'Newscaster' Image (b) Binarized 'Newscaster' image using Otsu Thresholding

## 4.4 Analysis of Video Data

In this section, an analysis has been carried out for video sequences of diverse content to observe the variations in CU size and prediction modes with changes in video content. Three video Sequences namely, 'RaceHorse', 'BQMall' and 'Johnny' are used and 20 frames of each sequence is encoded at QP values 22,27,32 and 37 using HEVC low-delay profile.

Tables 4.2 shows percentage CU Split cases at different CU depth levels during selection of optimal CU size. Percentage of CU Split cases at CU depth levels 0, 1 and 2 are considered because at depth level 3, CU is not further split. According to the Table, there are large variations in average percentage of CU Split cases for the three sequences. For 'RaceHorses' sequence with high motion content, average percentage of Split cases is upto 92.93% at depth level 0. For 'Johnny' sequence with large static regions and slow motion, this percentage is 27.23% at depth level 0. For 'BQMall' with moderate motion content, average percentage is 49.18% at CU depth level 0.

Table 4.3 shows percentage of Skip prediction mode at CU depth levels 0,1,2 and 3 for the three video sequences. Again there are large variations in percentage of Skip mode depending on the video content. According to the Table, for videos with low motion content and static regions like Johnny the percentage is upto 95.95% and for videos with high motion content like 'RaceHorses' it is as low as

TABLE 4.2: Percentage Distribution of CU Split Cases in Video Streams

| Stream | CU Depth | Qp = 22 | Qp = 27 | Qp = 32 | Qp = 37 | Average (%) |
|---|---|---|---|---|---|---|
| RaceHorses | Depth 0 | 99.19 | 97.11 | 91.73 | 83.69 | 92.93 |
| | Depth 1 | 79.12 | 65.47 | 49.06 | 33.98 | 56.91 |
| | Depth 2 | 37.58 | 25.51 | 15.09 | 8.66 | 21.71 |
| BQMall | Depth 0 | 67.67 | 53.21 | 42.63 | 33.2 | 49.18 |
| | Depth 1 | 41.47 | 29.18 | 21.13 | 13.79 | 26.39 |
| | Depth 2 | 16.56 | 11.41 | 6.98 | 3.81 | 9.69 |
| Johnny | Depth 0 | 51.21 | 31.21 | 16.46 | 10.05 | 27.23 |
| | Depth 1 | 23.7 | 8.79 | 3.41 | 1.68 | 9.39 |
| | Depth 2 | 5.6 | 1.27 | 0.48 | 0.15 | 1.87 |

2.24% at CU depth level 0. For video sequence with moderate motion content like 'BQMall', percentage of Skip mode is in between.

TABLE 4.3: Percentage Distribution of Skip Mode Cases in Video Streams

| Stream | CU Depth | Qp = 22 | Qp = 27 | Qp = 32 | Qp = 37 | Average (%) |
|---|---|---|---|---|---|---|
| RaceHorses | Depth 0 | 0.00 | 0.87 | 2.02 | 6.07 | 2.24 |
| | Depth 1 | 0.89 | 8.85 | 21.47 | 34.30 | 16.38 |
| | Depth 2 | 9.51 | 32.03 | 51.76 | 66.61 | 39.98 |
| | Depth 3 | 31.12 | 60.28 | 76.36 | 86.53 | 63.57 |
| BQMall | Depth 0 | 31.46 | 46.85 | 56.74 | 64.31 | 49.84 |
| | Depth 1 | 42.87 | 58.19 | 68.95 | 77.58 | 61.90 |
| | Depth 2 | 59.20 | 74.54 | 83.58 | 89.45 | 76.69 |
| | Depth 3 | 75.83 | 87.18 | 92.81 | 95.91 | 87.93 |
| Johnny | Depth 0 | 48.79 | 66.92 | 78.43 | 87.68 | 70.45 |
| | Depth 1 | 64.31 | 82.17 | 90.49 | 94.92 | 82.97 |
| | Depth 2 | 77.46 | 91.87 | 96.71 | 98.67 | 91.17 |
| | Depth 3 | 87.96 | 97.00 | 99.19 | 99.66 | 95.95 |

Stats in Tables 4.2 and 4.3 indicate, majority of blocks in the video frame for sequences with complex texture are encoded using small CU sizes as opposed to the frames with homogeneous texture that are encoded using large CU size. Similarly, large temporal variations and high motion content in video frames is also encoded using small CU blocks. Video frames with complex texture and high motion content also require diverse set of prediction modes and percentage of Skip or merge mode is reduced. On the other hand, video sequences with homogeneous regions and slow motion contain large percentage of Skip mode and are encoded using large CU size. Hence, if spatial or temporal complexity of a video frame can be calculated, this information can assist in fast encoding of the video sequence, as unnecessary prediction modes and block sizes can be ignored.

## 4.5 Proposed Fast Coding System

As discussed in the previous section, HEVC encoding decisions about CU depth level and prediction modes selection are highly dependent on the video content. In this work a content adaptive approach has been devised for fast HEVC encoding that early terminates CU depth level search and fast selection of optimal prediction modes to reduce the computational complexity of HEVC. Approach uses both global and local parameters to adapt the fast encoding method to the video content.

To take the advantage of the correlation between the neighboring CU blocks in current and collocated frames, information of four neighboring CUs has been used. These neighboring CU blocks include, upper (U), left (L), upper-left (L-U) CUs in the current frame and collocated CU in the previous frame as shown in Fig. 4.3.

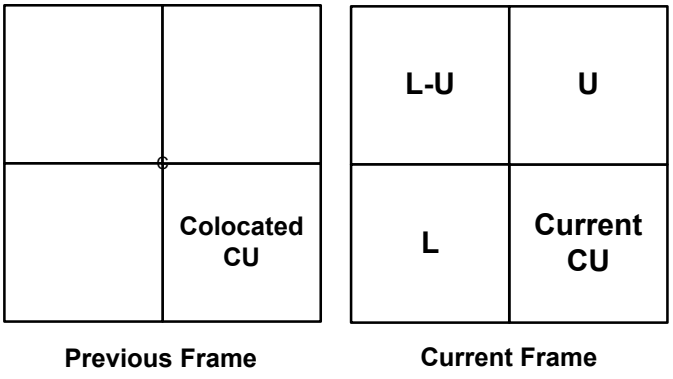


FIGURE 4.3: Neighboring CU Blocks in Current and Previous Frame

### 4.5.1 Classification of Video Content

Encoding decisions in HEVC change with spatial and temporal content in the video data. Global parameters like SI and TI are used to identify frames in video sequences with complex texture and high motion content. SI and TI parameters are computed for the video and these values are normalized to bring them in the range 0 to 1. Normalized values of SI and TI have been computed for ten video sequences of different texture and motion content and shown in the Fig. 4.4. These sequences

include, 'BlowlingBubbles', 'RaceHorses', 'BQMall', 'PartyScene', 'Catcus', 'BasketBallDrive', 'BQTerrace', 'FourPeople', 'Traffic' and 'Video4'. Video Sequences with low motion and smooth texture like 'Video4' and 'Traffic' have normalized SI and TI values less than 0.5. Similarly, video sequences with high motion content and complex texture like 'BlowingBubbles', 'RaceHorses' and 'BQTerrace' have normalized SI and TI values above 0.5.

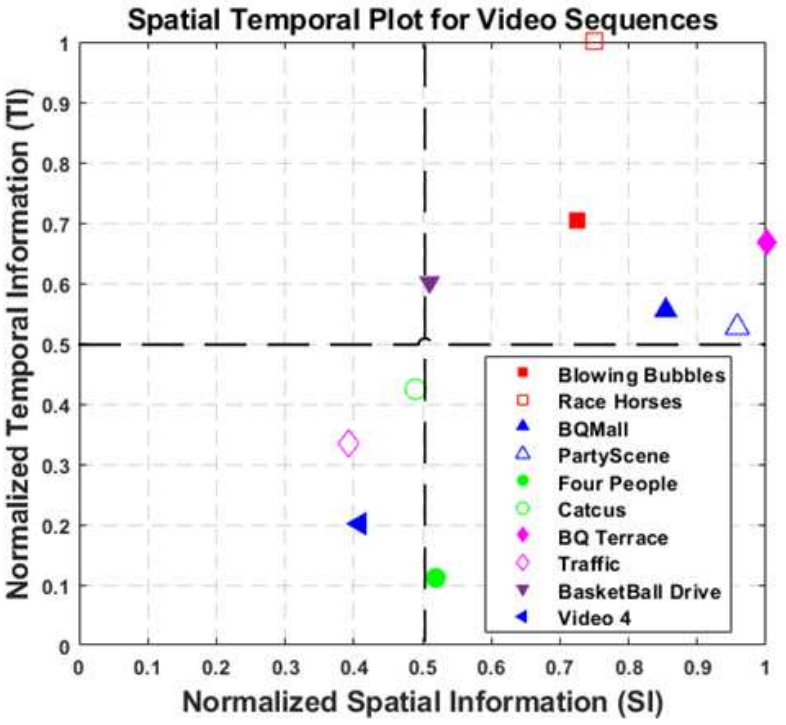


FIGURE 4.4: Classification of Video Content based on SI and TI Parameters

Hence, SI and TI parameters can be used to categorize the video frames into simple or complex frames. In the proposed methodology, when sum of normalized SI and TI values is greater than or equal to 1.0, video frame is considered complex, otherwise it is considered a simple frame.

### 4.5.2 Otsu Difference Maps

Proposed approach uses Otsu thresholding for identification of static regions in the video frame. First a difference frame is computed according to the Eq. 4.3. Otsu thresholding is applied on the difference frame to categorize each pixel into

foreground or background regions. Otsu threshold is computed using histogram of pixel values in difference frame and used to binarize the difference frame as shown in the Eq. 4.9. As a result, a binary difference map is generated. In the difference map, pixels with value of 0 belong to static region, while pixels with value of 1 belong to non-static region. Example difference maps are shown in Figs. 4.5 and 4.6. These maps have been generated for frames in video sequences 'BasketBallDrill' and 'BlowingBubbles', respectively. According to these maps, frame in the 'BasketBallDrill' has more static regions than the frame in the 'BlowingBubbles' sequence.

$$Map(i,j) = D_n\ (i,j) \geq T_{th}\ . \tag{4.9}$$

FIGURE 4.5: Otsu Threshold based Difference Map for 'BasketBallDrill'

Information in the difference maps can assist in early selection of Skip prediction mode without exhaustively checking all the prediction modes for a specific CU block.

FIGURE 4.6: Otsu Threshold based Difference Map for 'BlowingBubbles'

### 4.5.3 Content Adaptive Fast Encoding

In order to adapt the fast encoding method to the video content, first normalized SI and TI parameters are computed at frame level. If sum of normalized SI or TI values is greater than or equal to 1.0, frame is categorized into complex video frame. Otherwise it is categorized as a simple video frame and a specific fast algorithm is used for each frame type according to the flow-chart in Fig. 4.7. Processing of simple and complex video frames is described in the following sections.

### 4.5.4 Processing of Simple Video Frames

Simple video frames have low texture and motion content that results in more percentage of large CU sizes and Skip mode cases during HEVC encoding. Proposed fast encoding method for simple video frames at each CU depth level is outlined below:

- Compute average depth of neighboring CUs using the Eq. 4.10 below :

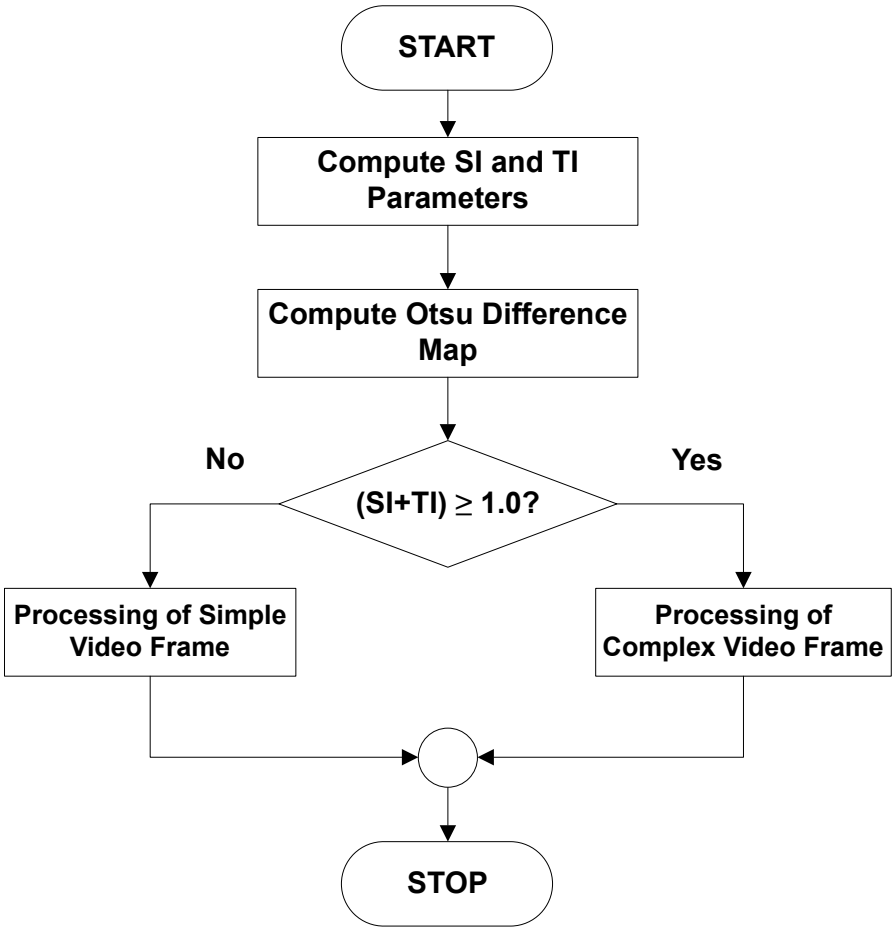


FIGURE 4.7: Classification of Video Frame using spatio-temporal (SI, TI) Parameters

$$\text{DepthN} = \sum_{i=1}^{N} w_i d_i, \tag{4.10}$$

where N = 4 and $d_i$ is the depth level of neighboring CUs. Value of $w_i$ is chosen as 0.25 to compute the average.

- Using the Otsu Difference map, check if current CU is static or not. If current CU is static and depth level of current CU is greater than the average neighbors depth, stop the CU depth search.

- For static CUs, if prediction mode of collocate CU in the previous frame is Skip, select Skip as the optimal mode for current CU without exhaustively checking all the other prediction modes.

- For non-static CUs, after checking merge and 2Nx2N prediction modes, if coefficient count of prediction residual is zero, skip checking of asymmetric and intra prediction modes.

Above steps are repeated for CU depth levels 0, 1, 2 and 3 starting from CU depth level 0.

### 4.5.5 Processing of Complex Video Frames

Complex video frames have high texture and motion content that results in more percentage of small CU sizes and diverse set of prediction modes during HEVC encoding. Proposed fast encoding method for complex video frames at each CU depth level is outlined below:

- Compute average depth of neighbors using the Eq. 4.10. If average neighbors depth is greater than the depth level of the current CU, skip searching for optimal prediction modes at current CU depth level
- Using the Otsu Difference map, check if current CU is static or not. If current CU is static and depth of current CU is greater than the average neighbors depth, stop the search for optimal CU depth.
- For static CUs, if prediction mode of 4 neighboring CUs is Skip, select Skip as the optimal mode for the current CU without exhaustively checking all the other prediction modes.
- For non-static CUs, after checking merge and 2Nx2N prediction modes, if coefficient count of prediction residual is zero, skip checking of asymmetric prediction modes.

Above steps are repeated for CU depth levels 0, 1, 2 and 3 starting from CU depth level 0.

### 4.5.6 Overall Content Adaptive Algorithm

Proposed overall content adaptive algorithm for fast HEVC encoding first evaluates global parameters at frame level to categorize the video frame into simple or complex category. Otsu threshold based difference map is also computed at frame level and used to identify the static regions in the video frame. After categorization of the video frame, frame specific fast encoding methods described in sections 4.5.4 and 4.5.5 are repeated for each CU in the video frame starting from the depth level 0. Flow-chart of the fast encoding method at CU level is shown in Fig. 4.8. According to the flow-chart, exhaustive search for optimal prediction mode can be stopped if prediction residual after checking of Skip or 2Nx2N modes is zero or there is a large percentage of Skip blocks in the neighbors. Similarly, for static CUs, if current CU depth is greater than the average neighbors CU depth, search for optimal CU depth level can also be terminated.

## 4.6 Results and Discussion

Proposed content adaptive approach for fast encoding of videos has been implemented in HEVC reference software HM version 16.10 [126] and results are computed on an Intel Core i5 machine running at 1.7 GHz with 8 GB of memory. These results have been computed for HEVC Random Access (RA) and Low Delay (LD) profiles at Qp values 22, 27, 32 and 37 according to the HEVC common test conditions [60]. Video streams of different resolutions and multimedia content have been used for computation of speed-up in HEVC encoding and RD performance. To compute the speed-up of the proposed approach as compared to the HM reference, Eq. 4.11 has been used.

$$TS(\%) = \frac{(T_r - T_p) * 100}{T_r}, \tag{4.11}$$

where Tr is the execution time of HM reference and Tp is the execution of the proposed fast encoding method. For measurement of RD performance, Bjontegaard

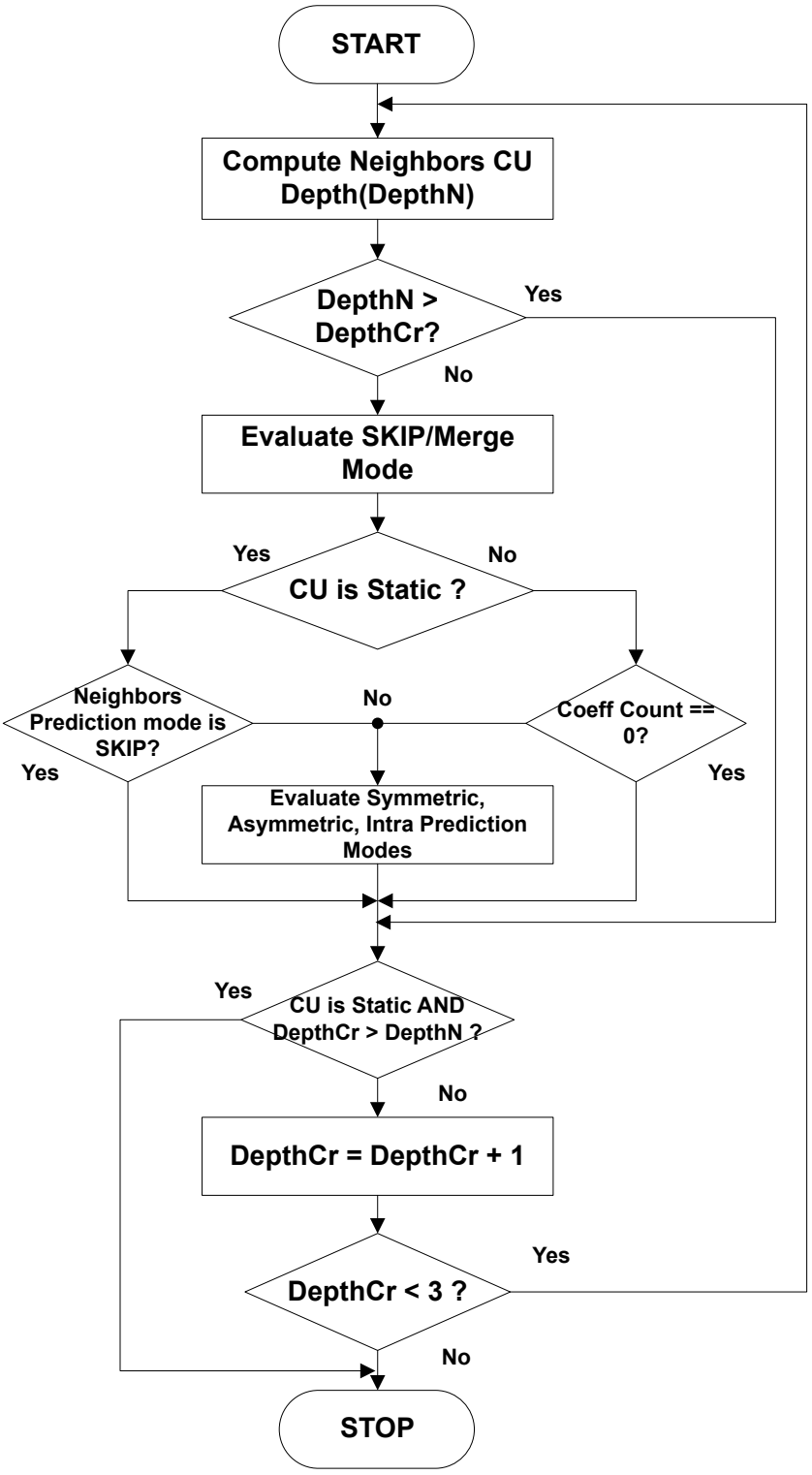


FIGURE 4.8: Flow chart of the fast encoding algorithm at CU Level

delta bit-rate, BDBR (%) and Bjontegaard delta PSNR, BDPSNR (dB) have been used [56].

Table 4.4 shows performance of proposed approach for both RA and LD profiles.

According to these results, proposed approach gives overall speed-up of 56.56% for RA profile and 52.69% for LD profiles, respectively. This speed-up is achieved with increase in BDBR of 1.67 and 2.94, respectively. These results also indicate that proposed content adaptive approach shows equally good results both for simple and complex content video streams. For simple video sequences with low motion and smooth texture like 'FourPeople', proposed approach gives speed-up of 83.52% for RA profile with negligible increase in BDBR of 1.49 and 77.68% for LD profile with increase in BDBR of 5.37, respectively. For video sequences with complex texture and high motion content like 'BQTerrace', proposed approach gives speed-up of 61.88% with increase in BDBR of 1.20 for RA profile and speed-up of 55.23% with increase in BDBR of 1.81 for LD profile, respectively. Hence, proposed approach gives good speed-up for simple video sequences and shows good RD performance for videos of complex nature.

Table 4.4: Results of Proposed Approach for RA and LD Profiles

| Stream | Resolution | RA Profile | | | LD Profile | | |
|---|---|---|---|---|---|---|---|
| | | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) |
| BasketballPass | 416x240 | 1.31 | -0.062 | 47.67 | 1.73 | -0.130 | 45.10 |
| BQSquare | | 0.62 | -0.027 | 46.18 | 1.51 | -0.026 | 43.23 |
| BasketballDrill | 832x480 | 1.93 | -0.081 | 57.46 | 3.89 | -0.163 | 56.21 |
| BQMall | | 2.18 | -0.089 | 48.53 | 3.71 | -0.172 | 46.54 |
| RaceHorses | | 2.21 | -0.086 | 35.04 | 2.11 | -0.078 | 33.12 |
| FourPeople | 1280x720 | 1.49 | -0.048 | 83.52 | 5.37 | -0.198 | 77.68 |
| KristenAndSara | | 1.33 | -0.050 | 77.88 | 3.66 | -0.125 | 66.89 |
| BasketballDrive | 1920x1080 | 3.14 | -0.067 | 59.70 | 3.13 | -0.082 | 58.01 |
| BQTerrace | | 1.20 | -0.025 | 61.88 | 1.81 | -0.051 | 55.23 |
| Cactus | | 1.50 | -0.032 | 61.33 | 2.91 | -0.092 | 57.85 |
| PeopleOnStreet | 2560x1600 | 1.52 | -0.068 | 42.92 | 2.57 | -0.126 | 39.78 |
| Average | | 1.67 | -0.058 | 56.56 | 2.94 | -0.113 | 52.69 |

RD performance of the proposed approach against HM reference for 'Catcus' stream is shown in the Fig. 4.9. According to the RD curve, RD performance of the proposed approach is similar to the HM reference with an average encoding speed-up of 61.33%.

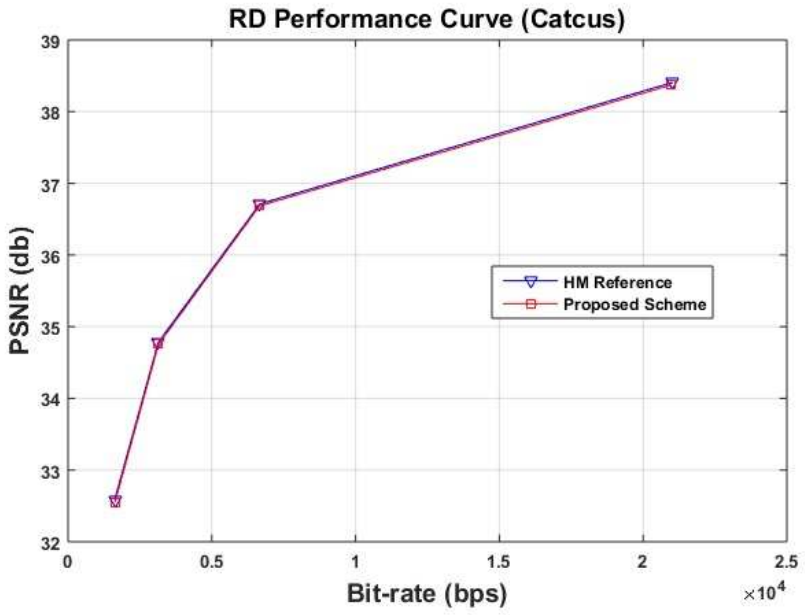


FIGURE 4.9: RD Performance Curve for 'Catcus' (Qp : 22,27,32,37 )

## 4.7 Performance Comparison

Speed-up comparison of the proposed approach and J.Lee [80] has been shown for 'PeopleOnStreet' and 'Catcus' streams at Qp values 22, 27, 32, and 37 in Figs. 4.10 and 4.11. It is clear from these Figs. that proposed approach performance better than the J.Lees method in terms of speed-up for both the streams at all Qp values.

Detailed performance comparison of the proposed approach with other existing implementations is shown in the Table 4.5. According to the table, proposed approach gives average speed-up of 56.56% at BD BR of 1.67% for RA profile. Average speed-up of F.Chen [76], Sangsoo [59] and D. Fernandez [104] for the same profile are 48.60%, 49.35% and 31.57%. Hence, overall speed-up of proposed approach is better than all the three implementations with almost similar RD performance.

For video streams with low motion content and simple texture like 'FourPeople', D. Fernandez [104] gives speed-up of 35.27%, F.Chen [76] 63.6% and Sangsoo [59] 74.1%. For the same sequence, proposed approach adapts better to the video content and gives speed-up of 83.52%. Similarly, for video streams with high

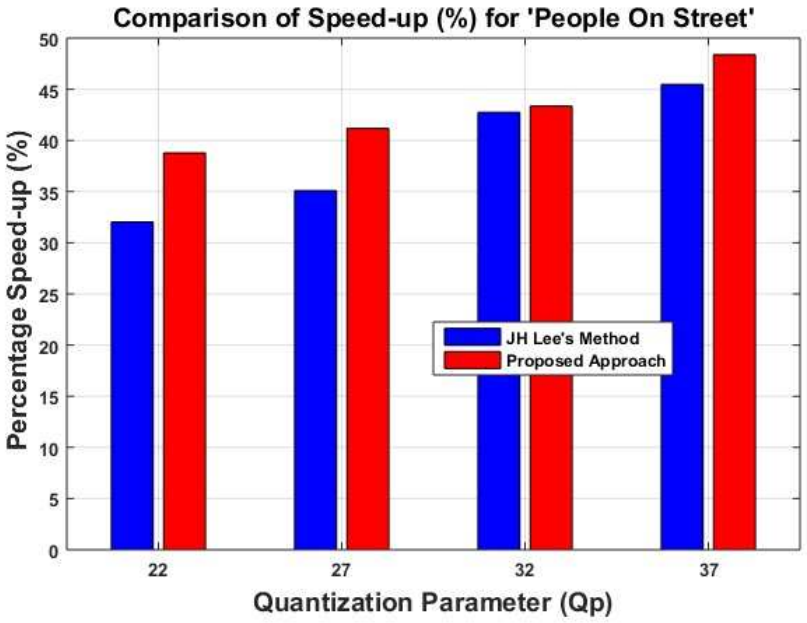


FIGURE 4.10: Speed-up Comparison for 'PeopleOnStreet'

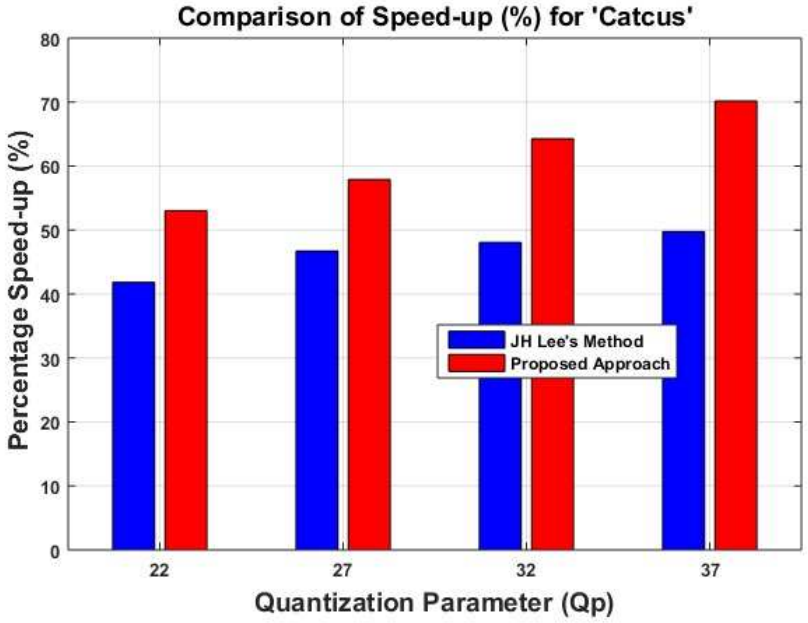


FIGURE 4.11: Speed-up Comparison for 'Catcus'

motion content and complex texture like 'BQTerrace', Sangsoo [59] gives speed-up of 54.70%, F.Chen [76] 50.6% and DG Fernandez [104] 35.47%. On the other hand, proposed approach adapts better to the complex video content and gives speed-up of 61.88% for the same video sequence. Similar trend in performance can be observed for videos with moderate complexity in terms of content like

TABLE 4.5: Comparison of results of the proposed approach with existing works for HEVC Random Access profile

| Stream | Resolution | Proposed Approach | | F.Chen [76] | | Sangsoo [59] | | DG Fernandez [104] | |
|---|---|---|---|---|---|---|---|---|---|
| | | BDBR | TS(%) | BDBR | TS(%) | BDBR | TS(%) | BDBR | TS(%) |
| BasketballPass | 416x240 | 1.31 | 47.67 | 1.8 | 35.5 | 1.50 | 33.60 | 1.4 | 30.49 |
| BQSquare | | 0.62 | 46.18 | 2 | 34.5 | 0.60 | 45.10 | 0.7 | 20.43 |
| BasketballDrill | 832x480 | 1.93 | 57.46 | 1.6 | 47.7 | 1.90 | 45.20 | 2.1 | 32.33 |
| BQMall | | 2.18 | 48.53 | 1.5 | 40.3 | 2.20 | 48.60 | 0.8 | 24.87 |
| RaceHorses | | 2.21 | 35.04 | 1.8 | 43.6 | 2.2 | 33.9 | 0.5 | 16.76 |
| FourPeople | 1280x720 | 1.49 | 83.52 | 1.1 | 63.6 | 1.7 | 74.1 | 1.1 | 35.27 |
| KristenAndSara | | 1.33 | 77.88 | 1.1 | 60.4 | 1.2 | 73.1 | 1.6 | 46.41 |
| BasketballDrive | 1920x1080 | 3.14 | 59.70 | 1.4 | 52.6 | 2.00 | 50.90 | 1.9 | 46.92 |
| BQTerrace | | 1.20 | 61.88 | 0.8 | 50.6 | 1.60 | 54.70 | 1.2 | 35.47 |
| Cactus | | 1.50 | 61.33 | 1.5 | 55.9 | 2.80 | 56.80 | 1.5 | 32.7 |
| PeopleOnStreet | 2560x1600 | 1.52 | 42.92 | 2.1 | 49.9 | 0.90 | 26.90 | 1.1 | 25.63 |
| Average | | 1.67 | 56.56 | 1.52 | 48.60 | 1.69 | 49.35 | 1.26 | 31.57 |

'BasketBallDrill' in the table.

## 4.8 Analysis of Results

In comparison to other state of the art implementations, proposed approach has outperformed all of them in terms of average speed-up with negligible increase in BDBR figures as shown in the Table 4.5. This is because, it adapts better to the nature of video content both for simple videos and streams with high motion and complex texture. Therefore, proposed content adaptive approach can be used in video encoding solutions to reduce complexity of HEVC.

According to the Table 4.4, for LD profile BDBR figures of proposed approach are on the higher side as compared to the RA profile. Similarly, for certain streams like 'BasketballDrive' RD performance is not up-to the mark for both the profiles. For streams like 'PeopleOnStreet' with large temporal variations and complex texture, speed-up needs further improvement. This is mainly because fixed thresholds are being used in categorization of frames as simple or complex in the proposed methodology and any miscalculation in the categorization ultimately affects the speed-up or the RD performance. Using a machine learning based method for this categorization may improve the results.

## 4.9 Summary

A content adaptive fast video encoding technique was implemented in this chapter. Global parameters SI, TI and Otsu Threshold at frame level were used to evaluate the type of video content. Based upon these evaluations, content specific schemes were implemented for simple and complex frames in the video sequence. Results of this implementation showed that proposed approach adapts better to the video content and gives promising results as compared to the existing implementations. In the next chapter, machine learning based approaches will be explored for fast coding of video data.

# Chapter 5

# Fast Video Coding using Random Forests

This chapter presents the second major contribution of the research. A machine learning based fast encoding method to reduce the computational complexity of HEVC is proposed. The approach uses Random Forests (RF) based ensemble classification for selection of prediction mode and early termination of CU size search in HEVC. A large feature set for training of Random Forests classifiers is extracted from video data. Comparative analysis of two feature selection methods has been done to reduce the features set for training of the classifier model. The chapter also presents the performance analysis of classifier models and comparison of the proposed approach with other similar methods.

## 5.1 Machine Learning for Fast Video Coding

Machine learning based methods are being used for fast video encoding and usually perform better than statistics base approaches that use hard or soft thresholds for making early decisions. Researchers have experimented with different machine learning algorithms to reduce the computational complexity of recent video codecs, as discussed in chapter 3. These methods use features extracted from video data

to generate a classifier model and use it for early prediction of critical decisions in the video codec. Majority of literature on machine learning based methods uses single classifiers for fast video encoding and these classifier models are trained offline.

Commonly used machine learning methods in fast video encoding are Bayesian classification, Decision Trees and SVM. Single Decision Tree is highly dependent on input data and tend to overfit that affects its accuracy and performance. Bayesian classification methods often use Naive Bayes classifiers. Naive Bayes assumes, all the input features are independent of each other that is not always true for video data. Another underlying assumption in Bayes classification is that input data has normal distribution. Similarly, SVM heavily rely on the number of support vectors selected from the training data. Secondly, because of slow training speed, it is not preferred in online training scenarios [42].

### 5.1.1 Why Random Forests?

Little work is available in literature on using ensemble classification methods like Random Forests for fast video encoding. Random Forests uses multiple decision trees for classification and each tree is generated using randomly selected features. Random Forests handles the over-fitting problem of single decision tree by majority voting on the output of individual trees and biasness towards specific features through randomness. Ensemble classifiers usually have more computational overhead but Random Forests is computationally efficient and can be used in real-time implementations [96]. Similarly, classifier models used in video codecs are usually trained offline. Offline trained classifier models heavily rely on training data and feature set used for training to handle the variations in video content.

To compare the performance of Random forests with other machine learning based methods, simulations are run using MATLAB 'Classification Learner' application. In these simulations, features extracted from video data are used and accuracy of

TABLE 5.1: Comparison of Random Forests with other machine learning methods

| Classifier Model | AUC | Classifier Accuracy (%) |
|---|---|---|
| Linear Discriminant Function | 0.92 | 84.2 |
| Logistic Regression | 0.93 | 85.7 |
| SVM | 0.93 | 86.4 |
| Decision Tree | 0.93 | 87.0 |
| Random Forests | 0.96 | 89.8 |

Random Forests is evaluated against Discriminant analysis based classifier, Decision Tree, SVM and logistic regression models. Simulations are run for CU split decision in HEVC. Results of these simulations are shown in the Table 5.1. According to these results, Random Forests shows better accuracy and maximum area under the curve (AUC) in receiver operating characteristics (ROC) curves than all the other classification models. ROC curves of these simulations are shown in Fig. 5.1. These results demonstrate that for video data, Random Forests shows better performance as compared to other machine learning methods.

In this work, Random Forests has been used for fast video encoding. Both offline and online trained classifier models are used to handle the variations in video content and evaluate the impact of individual models on the performance of fast encoding technique. Details of the Random Forests classifier and the proposed methodology is given in the following sections.

## 5.2 Overview of Random Forests

Decision trees are used for classification and regression in machine learning based methods. Random Forests (RF) is an ensemble classifier model [127, 128] that uses multiple decision trees to form a forest, as shown in the Fig. 5.2. An ensemble classifier, by definition, is a group of weak classifiers that collaborate to take a strong decision. For classification, the input test sample is fed to all decision trees and then each decision tree associates the sample to a specific class. Majority voting is performed on the outcome of all decision trees to classify the test sample.

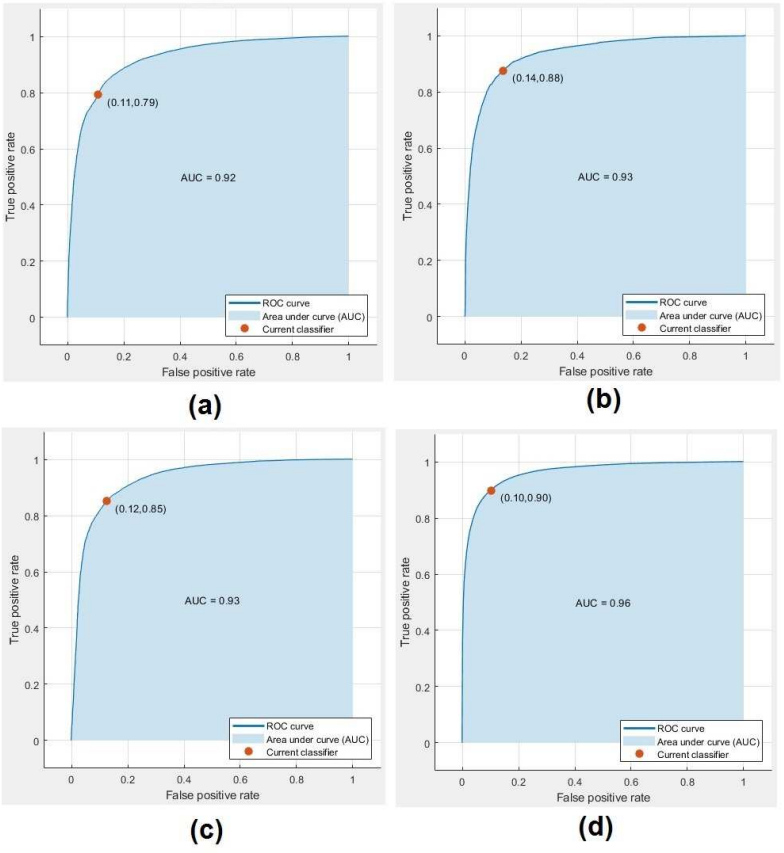


FIGURE 5.1: Comparison of ROC Curves (a) Linear Regression Function (b) Decision Tree (c) SVM (D) Random Forests

For training of the classifier model, samples are randomly selected from the training data with replacement and used for decision tree generation. This reduces the correlation between the generated trees. Also a sub-set of features is used at each node split while generation of decision trees in the forest. Features in the sub-set are randomly selected [129, 130]. This randomness in decision tree generation, along with majority voting on the outcome of all trees in the forest, adds diversification to the Random Forests classifier and makes it immune to over-fitting. During tree generation, an entropy-based impurity measure is used for split decision at each node in the decision tree according to Eq. 5.1, where i(X) is the impurity at node X, p($w_i$) is the fraction of samples that belong to class $w_i$. The decrease in impurity at node X is measured by using Eq. 5.2, where i($X_L$) and i($X_R$) are the impurities of left and right children nodes $X_L$ and $X_R$. $P_L$ is the fraction of training samples that will go to the left child node after split. Eq. 5.3 is used as a stopping criterion in decision tree generation during the training process,

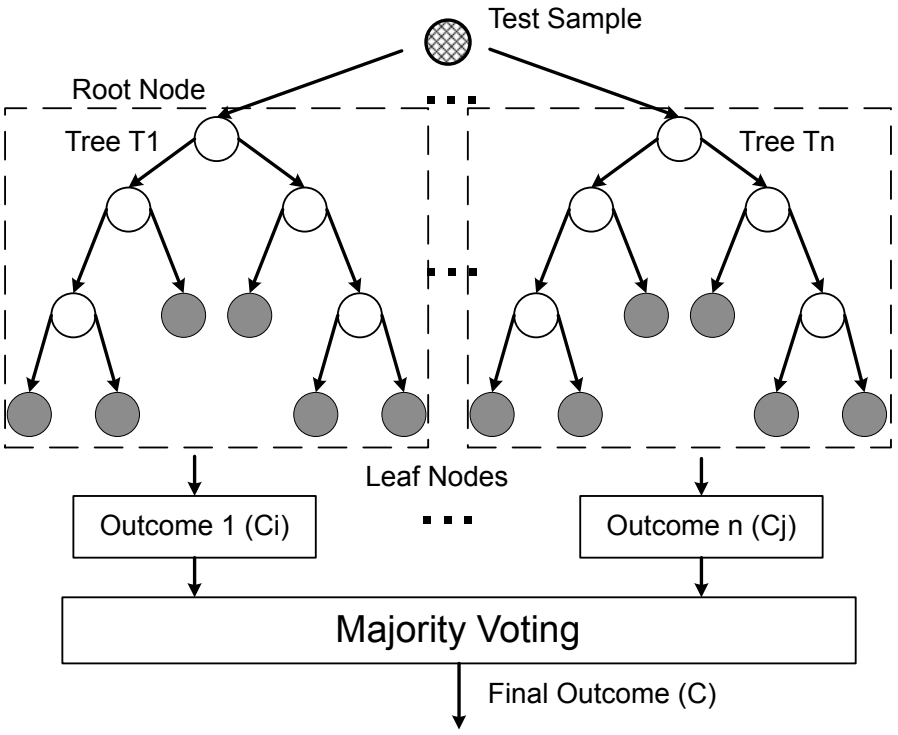


FIGURE 5.2: Random Forests (RF) with N Decision Trees

where $\beta$ is a threshold value [131].

$$i\left(X\right) = \ -\sum_{i \in C}^{n} \left(\left(p\left(w_i\right)\right) \log_2 \left(p\left(w_i\right)\right)\right), \tag{5.1}$$

$$\Delta i\left(X\right) = \ i\left(X\right) - P_L i\left(X_L\right) + \left(1 - P_L\right) i\left(X_R\right), \tag{5.2}$$

$$\Delta i\left(X\right) \leq \ \beta. \tag{5.3}$$

Random Forests uses randomly selected sub-set of features for node split in decision tree generation followed be majority voting on outcome of all decision trees. Because of these randomly selected features, trees in the Random forest are less correlated and perform better than other ensemble classification methods like Bagging[132]. Samples from the training set that are not selected for tree generation can be used to validate the Random Forests classifier. This process is

known as out-of-bag training accuracy and is often used to cross-validate the classifier model.

The accuracy of Random Forests classifier depends upon the number of trees in the forest, number of features used for split at each node and depth of each decision tree. A large number of fully grown trees may increase classifier accuracy but it also increases the computational complexity of the classifier model. Hence, selection of optimal number of trees, tree depth and other parameters is a trade-off between accuracy of the classifier model and computational complexity.

## 5.3 Proposed Fast Encoding Method

As described in Chapters 2 and 4, testing all CU sizes at different depth levels of the CU quad-tree structure, is one of the most computationally intensive tasks during HEVC encoding. Similarly, within each CU, testing of TU quad-tree structure and evaluation of all inter-prediction modes for different CU sizes at all depth levels also adds to the computational complexity. Hence, the proposed methodology targets these three tasks to reduce the computational complexity of HEVC, based on the RF ensemble classification to intelligently predict early CU and TU depth termination rather than testing all depth levels to find the optimal one. Similarly, RF-based classifier is also used for early prediction of Skip mode during inter coding. An ensemble of offline and online classifier models is used, taking advantage of all the characteristics of these type of classifiers. First, it intelligently predicts Skip mode at different CU depth levels. Secondly, it is used to predict the optimal TU size within each CU. Lastly, the classifier is used for prediction of early CU depth termination in the inter frames. The classification problem is formulated as binary classification with two classes in each case. First, a set of features is identified as containing the best ones to be extracted from the video data and used to train the classifier. Than, the generated classifier model is integrated in the HEVC encoder. The fast coding decisions are taken by the classifier based upon the input features extracted from video data during encoding

at runtime. Therefore, selection of features for classifier training is an important process in this work, as described in the following section.

## 5.4 Feature Extraction and Feature Selection

In fast video encoding, the relevant features are derived from the inherent characteristics of the video signal, coding process, mode decision and data structures. These extracted features are used to train the classifier model. Homogeneous regions in a video sequence with smooth texture and small motion content are more often encoded as large CU blocks using HEVC. On the other hand, regions in video frame with complex texture are encoded using small CU sizes. The depth level of a CU is also highly correlated with spatial and temporal neighbors. TU Split decision is highly correlated with the number of non-zero coefficients and the absolute residue of a block. Similarly, Skip inter-prediction mode is usually selected as the optimal mode for homogeneous or static regions in video frames. Skip mode selection is also correlated with prediction modes of neighboring blocks [69].

Selection of optimal features is also important in machine learning approaches due to its strong influence on accuracy of classifier models. Dimensionality of the feature set is also important, because a large set may lead to overfitting and increases the computational complexity. In this work, two different approaches were evaluated for efficient feature selection. (i) Information gain filtering-based approach; (ii) Wrapper-based approach using RF feature importance as elimination criterion.

The proposed methodology to select the best features comprises two steps. First, a large set of relevant features is extracted from video data to predict CU and TU depth levels and Skip inter prediction mode. Than, both the filtering-based approach and wrapper-based approach are used to find the most relevant features and to drop those that do not significantly contribute to the decision process. Overall the large set includes features about texture, motion and other parameters related

to RD optimization and data structures. These include pixel variance and gradient of the current block using the Sobel operator that gives information about the texture in the block. The variance of current CU motion vectors (MV), average MV of neighboring CUs, sum of absolute differences (SAD) with co-located blocks that give information about motion content and homogeneous regions is also used [91]. Features related to RD optimization and data structures include non-zero (NZ) coefficient count in the current block, RD cost of the current block, quantization parameter (QP) and total bits to encode the current block. These features give information about residual content in the current block. Also, the depth of the neighboring CU blocks, partition size, average NZ coefficient count of neighbors and count of Skip blocks in the neighbors give information about the behavior of the neighboring blocks. To compute the features related to the neighboring CU blocks, three spatio-temporal neighboring CU blocks are considered. These include top and left CU in the current video frame and co-located CU in the adjacent frame [87].

The large feature sets defined in this work are shown in Table 5.2 for: fast SKIP mode selection (15 features), early CU depth termination (18 features) and early TU depth termination (7 features).

### 5.4.1 Filtering based Approach

The first approach to selectively reduce the size of the large feature sets is the filtering-based approach using the information gain [133, 134]. In the filtering-based approach, individual features are ranked based upon information gain to reduce the size of the feature set by only including those with higher rank. This approach can be described as follows.

If $A_i$ is an attribute and C is the class, a decrease in entropy of C reflects an increase in the information about C because of the attribute $A_i$. This is known as information gain and it is defined by Eq. 5.4 [135].

$$InformationGain = H(C) - H(C|A_i), \tag{5.4}$$

TABLE 5.2: Large feature set for SKIP mode selection, CU and TU SPLIT decision

| Features | Skip Mode | CU Split | TU Split |
|---|---|---|---|
| Average Neighbors CU Depth | ✓ | ✓ | – |
| Average Neighbors Partition Size | ✓ | ✓ | – |
| Pixel Variance Partial (1/4) of Current Block | ✓ | ✓ | – |
| Pixel Average of Current Block | ✓ | ✓ | – |
| Pixel Variance of Current Block | ✓ | ✓ | ✓ |
| MV Variance of Current Block | – | ✓ | – |
| Average MV of Neighboring Blocks | ✓ | ✓ | – |
| Non-Zero Coefficient Count of Neighbors | ✓ | ✓ | – |
| NZ coefficients of Current Block | ✓ | ✓ | ✓ |
| SAD with co-located Blocks | ✓ | ✓ | ✓ |
| RD Cost of SKIP Mode Encoding | – | ✓ | – |
| Neighboring CUs average RD Cost | ✓ | ✓ | – |
| RD Cost of Current Block | ✓ | ✓ | ✓ |
| Skip Mode Count of Neighboring Blocks | ✓ | ✓ | – |
| Gradient of Current Block using Sobel | ✓ | ✓ | – |
| Total Encoded Bits for Current Block | ✓ | ✓ | ✓ |
| QP of Current Block | ✓ | ✓ | – |
| Skip Flag in Current Block | – | ✓ | – |
| Abs Sum of coefficients of Current Block | – | – | ✓ |
| Prediction Residual of Current Block | – | – | ✓ |

where

$$H\left(C\right) = -\sum_{c \in C} \left(\left(p\left(c\right)\right) \log_2 \left(p\left(c\right)\right)\right), \tag{5.5}$$

$$H\left(C|A_i\right) = -\sum_{a \in A} p\left(a\right) \sum_{c \in C} \left(\left(p\left(c|a\right)\right) \log_2 \left(p\left(c|a\right)\right)\right). \tag{5.6}$$

Eq. 5.5 and 5.6 represent entropy of class C, before and after observing the attribute. Hence, information gain measures the importance of an individual attribute $A_i$ and how effectively it can discriminate between different classes. To filter out the least important features, each feature $A_i$ is assigned a score based upon its information gain and then all features are ranked according to such score.

### 5.4.2 Wrapper based Approach

In the wrapper-based approach, the combined effect of different feature subsets is analyzed using a predefined classifier and than the least important ones are excluded [136, 137]. In this work Random Forests is used as the predefined classifier. First a RF model is trained using all the features in the data set. To compute the importance of each feature, it is permuted and the classifier accuracy is computed with the resultant modified data set. A decrease in the classifier accuracy because of the permuted feature represents the importance of that feature [127]. This process is repeated to compute the importance of all features in the data set and then they are ranked according to their importance.

The information gain and feature importance of each feature was computed for Skip mode, CU and TU Split decision for the following four different video sequences: 'BasketBallDrive', 'BQMall', 'BlowingBubbles' and
'Johnny'. The relevant features are extracted for 20 frames in each video sequence encoded at four different QP values i.e. 24, 28, 32, 38. The features are extracted for all CU layers in the CTU i.e., from layer 0 (64x64) to layer 2 (16x16). Then the information gain of each feature is computed as the average for all CU sizes. Ranking of each feature based upon average information gain (filter) and average feature importance (wrapper) is shown in Table 5.3 for CU Split, Skip mode and TU Split cases. The rank order obtained for each feature, based on the information gain is shown in columns "F", while the rank order obtained from wrapper is shown in columns "W". Those features with the highest information gain/feature importance are ranked as '1' in the table, while higher rank values correspond to lower information gain/feature importance.

To select the optimal reduced feature sets for CU Split, Skip mode and TU Split cases, multiple RF classifier models were generated. Starting with the complete feature set in each case, features were gradually eliminated based upon their rank (according to information gain/feature importance shown in Table 5.3) and then new RF classifier models were generated. The number of trees in these classifiers was kept constant at 20 and the accuracy was calculated for each case. The

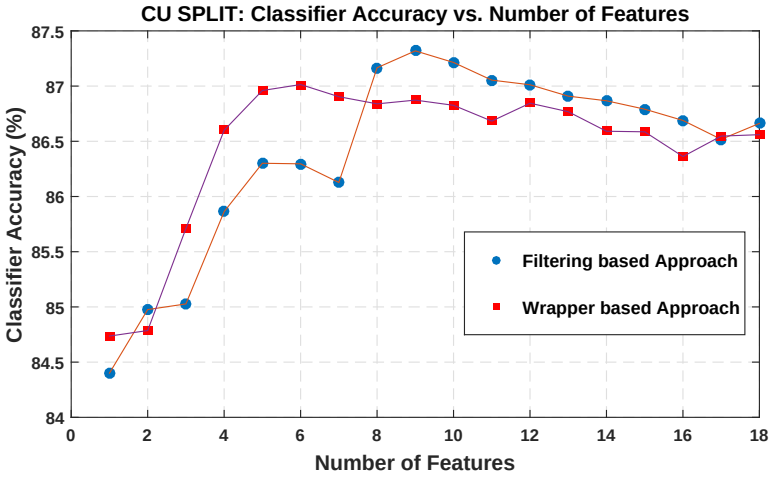


FIGURE 5.3: CU Split Decision: RF classifier accuracy vs. Number of features for filtering and wrapper based approaches

graphs of the classifier accuracy vs. number of features for filtering and wrapper approaches are shown in Figs. 5.3, 5.4 and 5.5 for CU Split, Skip mode and TU Split cases, respectively. In each graph, the number of features used in the x-axis corresponds to the subset defined by the same rank order in Table 5.3. For example, 6 features in the graphs of Figs. 5.3, 5.4 and 5.5 include those numbered from 1 to 6 in Table 5.3 for each case. This can also be seen as building the graph from the right to the left, by starting to compute the classifier accuracy for both the filtering and wrapper approaches using the full set of features and then removing those with lowest information gain/feature importance, one by one, till a single feature is left.

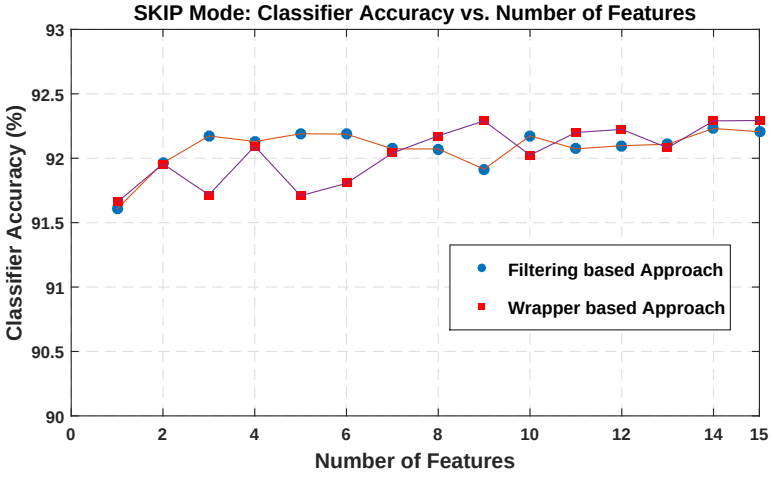


FIGURE 5.4: Skip Mode Selection: RF classifier accuracy vs. Number of features for filtering and wrapper based approaches

Selection of optimal features is a trade-off between the most relevant features and the feature set diversity. According to the graphs of Figs. 5.3, 5.4 and 5.5, as irrelevant features are removed (moving in the graph from the right to the left), classifier accuracy gradually increases up to a certain point. Then, beyond that point it again starts decreasing because of the lack of diversity in the feature set. Similarly, when features are gradually removed, the impact on accuracy depends on their information gain/feature importance, which may not be same for all the features. For instance in Fig. 5.3, for filtering based approach when feature no. 8 is removed, the classifier accuracy significantly decreases because this feature has a relatively high feature importance as shown in Table 5.3. According to the Figs. 5.3 and 5.5, the filtering-based approach reaches the maximum accuracy, higher than the wrapper-based approach, at 9 and 4 features for CU Split and TU Split, respectively. For CU Skip mode, as shown in Fig. 5.4, variations in accuracy are small i.e. 91.5% to 92.3%, as features are reduced from 15 to 1. Despite being small, this range of variability should not be neglected and approximated to zero (i.e., horizontal lines in Fig. 5.4), as it would lead to an ambiguous solution where any single feature would be equally good. Moreover, a very small feature set may not guarantee enough diversity for all types of video content. In Fig. 5.4, the filtering-based approach shows maximum accuracy at 10 features while the wrapper-based approach shows maximum accuracy at 9 features.

Based upon this analysis, any of the two feature selection approaches can be used to select the optimal features in video data. In this work the filtering based approach has been used as it gives better accuracy out of the two. From the previous results using the filtering-based approach, 9 features were found adequate to generate the classifier model for CU Split decision, 10 features for Skip mode selection and 4 features for TU Split decision. The reduced feature sets for the three cases are identified in Table 5.3 in bold figures, which also represent the rank order of the selected features. Examples of other previous works that also followed a similar type of heuristic approach for feature selection are [91] and [138].

TABLE 5.3: Reduced feature set for SKIP mode, CU and TU Split decision using the Filtering-based approach

| Features | CU Split | | Skip Mode | | TU Split | |
|---|---|---|---|---|---|---|
| | F | W | F | W | F | W |
| Total Encoded Bits for Current Block | **1** | 4 | **1** | 2 | **2** | 3 |
| NZ coefficients of Current Block | **2** | 2 | **2** | 1 | **1** | 7 |
| Average Neighbors CU Depth | **3** | 16 | **5** | 13 | - | - |
| Skip Flag in Current CU Block | **4** | 1 | - | - | - | - |
| Skip Mode Count of Neighbor Blocks | **5** | 6 | **3** | 7 | - | - |
| Average Neighbors Partition Size | **6** | 7 | **6** | 8 | - | - |
| NZ Coefficient Count of Neighbors | **7** | 14 | **4** | 10 | - | - |
| MV Variance of Current Block | **8** | 3 | - | - | - | - |
| Pixel Variance of Current Block | **9** | 5 | **10** | 3 | **4** | 4 |
| Average MV of Neighboring Blocks | 10 | 8 | **7** | 5 | - | - |
| Gradient of Current Block using Sobel operator | 11 | 11 | 11 | 9 | - | - |
| SAD with co-located Blocks | 12 | 12 | **8** | 4 | 5 | 2 |
| Pixel Variance Partial (1/4) of Current Block | 13 | 10 | 12 | 11 | - | - |
| QP of Current Block, | 14 | 17 | **9** | 12 | - | - |
| RD Cost of Skip Mode Encoding | 15 | 13 | - | - | - | - |
| RD Cost of Current Block | 16 | 9 | 13 | 6 | 7 | 6 |
| Neighboring CUs average RD Cost | 17 | 15 | 15 | 14 | - | - |
| Pixel Average of Current Block | 18 | 18 | 14 | 15 | - | - |
| Abs Sum of coefficients of Current Block | - | - | - | - | **3** | 1 |
| Prediction Residual of Current Block | - | - | - | - | 6 | 5 |

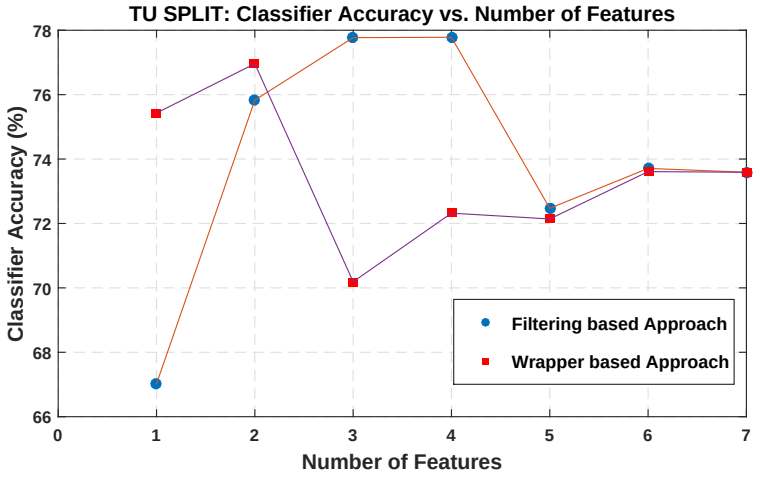


FIGURE 5.5: TU Split Decision: RF classifier accuracy vs. Number of features for filtering and wrapper based approaches

## 5.5 Design of Random Forests Classifier and Parameter Optimization

The proposed method to achieve fast coding decisions is based on separate RF classifiers trained for the Skip mode prediction and early termination of CU and TU depth search.

### 5.5.1 Early Skip Mode Prediction

Skip mode in HEVC is usually chosen as the optimal prediction mode in smooth image regions with low motion content. Moreover, Skip mode is the most likely used mode at low bit-rates, i.e. high QP values. If Skip mode is selected as the best mode at current CU depth level, then checking other inter prediction modes can be skipped, avoiding further processing. The ultimate objective of early Skip mode prediction is to determine the optimal mode, without exhaustively evaluating all intra and inter prediction modes.

The proposed model for early Skip mode prediction uses RF classification to speed-up selection of optimal prediction mode. The 10 input features shown in Table 5.3 are fed to the RF classifier model at run-time, where the Skip mode classification is done before checking all other inter prediction modes at current depth level. The outcome of the binary classifier is either Skip or Non-Skip. If Skip mode is predicted as the optimal mode, checking of all other inter-prediction modes is skipped at current depth level.

### 5.5.2 Early CU Depth Termination

In a full search implementation, such as the reference software HM, the CU depth selection process starts from depth level 0 and then all depth levels (0-3) are recursively evaluated based upon RD cost before selection of the optimal CU depth level. Usually depth level 0 is selected for smooth regions in video frame

and deeper depth levels i.e. 1,2 and 3 are selected for regions with more texture detail and motion content. Due to the intrinsic nature of the search algorithm and data structures, large depth levels and small CU sizes result in fewer encoding bits but increased computational complexity.

The proposed method extracts the 9 features shown in Table 5.3, from the video data and uses them as input to the RF classifier model. At each CU depth level, after evaluation of all prediction modes, the RF binary classifier takes a Split vs. Non-Split decision. If the classifier takes a Non-Split decision, all deeper CU depth levels are skipped. This results in major speed boost as all unnecessary CU depth levels are not checked.

### 5.5.3 Early TU Depth Termination

After prediction, a residual quad-tree is formed at the root of each CU, which recursively divides each TU into four sub TUs starting from TU size 32x32. While an exhaustive search approach checks all possible TU sizes to find the optimal one, the proposed model takes an early TU Split decision at each TU depth level. If a Non-Split decision is taken, then checking of sub TUs is skipped to speed up encoding. The 4 features identified in Table 5.3 are extracted and fed to the RF binary classifier model to speed-up the selection of best depth level in each TU quad-tree.

### 5.5.4 Configurations of Ensemble Classifier Model

In the proposed approach, three different configurations were investigated for training and testing the RF classifier model. In the first configuration, the RF model is trained offline and integrated in the HM Reference. In the second configuration, the RF model is trained online at run-time. In this case, few initial frames from each video stream are used for training and the remaining ones are encoded using the trained classifier model. In the third configuration, an ensemble of online and offline classifier models is used, in order to achieve the best possible results.

The combined classifier score is computed as a weighted average of both classifier outputs according to the Eq. 5.7 . In this work a weight of 0.5 was used for each one. For those configurations where only one classifier is used, the weight of the enabled classifier is set to 1, while that of the disabled one is set to zero. The generic diagram of such classifier is shown in Fig. 5.6, where the combined scores obtained from both the online and offline classifier models are used to take the CU/TU Split and Skip mode decisions.

$$C = \sum_{i=1}^{N} w_i C_i. \tag{5.7}$$

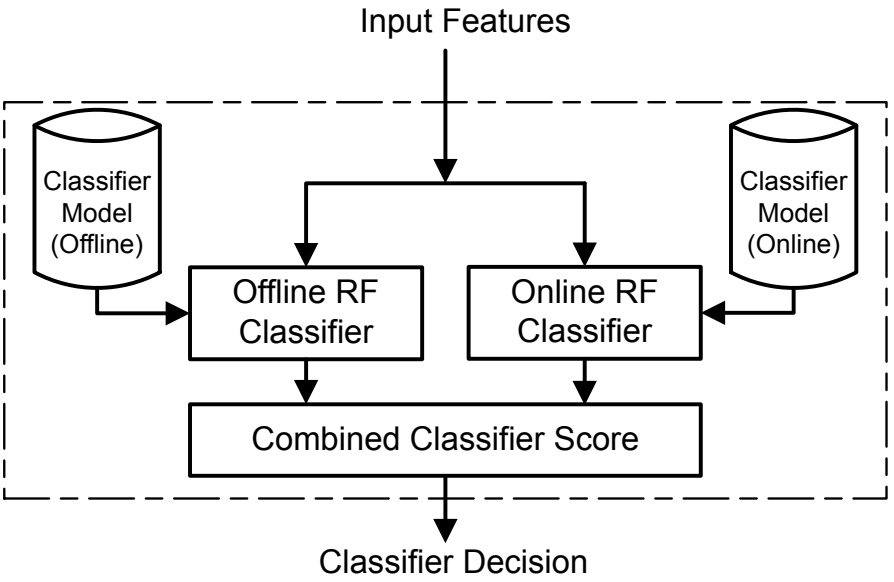


FIGURE 5.6: RF Ensemble Classifier with Online and Offline Classifier models

### 5.5.5 Overall Fast Encoding Algorithm

The proposed fast encoding decision algorithm based on ensemble classifiers for Skip mode prediction, TU Split and CU Split early decisions is shown in the flow-charts of Fig. 5.7 and 5.8. This algorithm is repeated during HEVC encoding for each CTU in inter frames starting from depth level 0. The ensemble classifiers for Skip mode prediction and early CU depth termination run at CU level as shown in Fig. 5.7. After training the classifier models either offline or online, the main steps of the proposed fast algorithm are summarized as follows:

**Step 1**: Test the Skip prediction mode at current CU depth level and compute its RD cost.

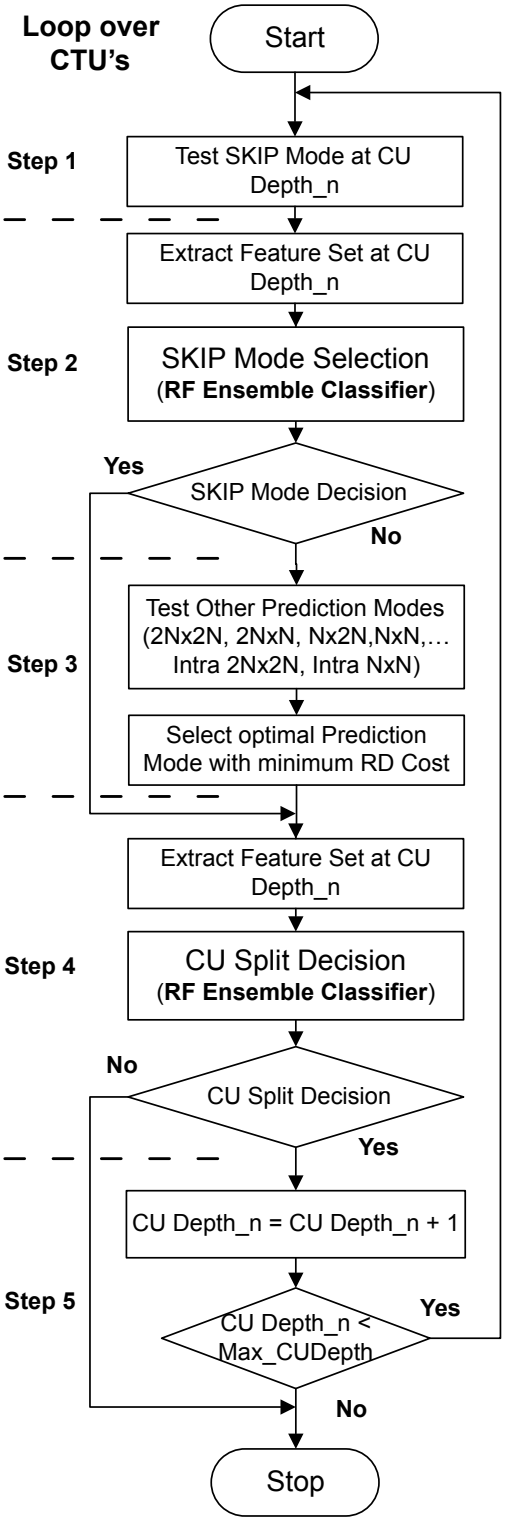


FIGURE 5.7: Flow-Charts of the Overall Fast Encoding Algorithm : Early Skip mode selection and CU Split Decision at different depth levels in a CTU

**Step 2**: Compute the feature set shown in Table 5.3 for early Skip mode selection and run the RF ensemble classifier. If the outcome is Skip, then select Skip as the optimal mode and goto Step 4, otherwise proceed to Step 3.

**Step 3**: Test all the other inter and intra prediction modes and possible PU sizes at current depth level and select the optimal prediction mode with minimum RD cost.

**Step 4**: Compute the feature set shown in Table 5.3 for early CU depth termination and run RF ensemble classifier. If classifier outcome is Non-Split, stop the RDO process at current CU depth level and skip the next step for current CU otherwise proceed to Step 5.

**Step 5**: Split the current CU into 4 sub-CUs, increment CU depth level by 1 and repeat Steps 1 to 5 for the next CU depth level.

The ensemble classifier for early TU depth termination runs at PU level inside every CU, as shown in Fig. 5.8. The main steps of the proposed fast algorithm are summarized as follows:

**Step 1**: Compute the residual transform at current TU depth Level.

**Step 2**: Extract the feature set shown in Table 5.3 for early TU depth termination and run RF ensemble classifier. If classifier outcome is Non-Split, stop the RDO process at current TU depth level and skip the next step for current TU, otherwise proceed to Step 3.

**Step 3**: Split the current TU into 4 sub-TUs, increment the TU depth level by 1 and repeat Steps 1 to 3 for the next TU depth level.

The RF ensemble classifier for fast selection of Skip mode is run for all the CU depth levels from 0 to 3. On the other hand, the RF ensemble classifier for fast CU Split decision is only run at CU depth levels 0,1, and 2 because CU is not further split at depth level 3. Similarly, the RF ensemble classifier for fast TU Split decision is only run at TU depth levels 1, 2, and 3.

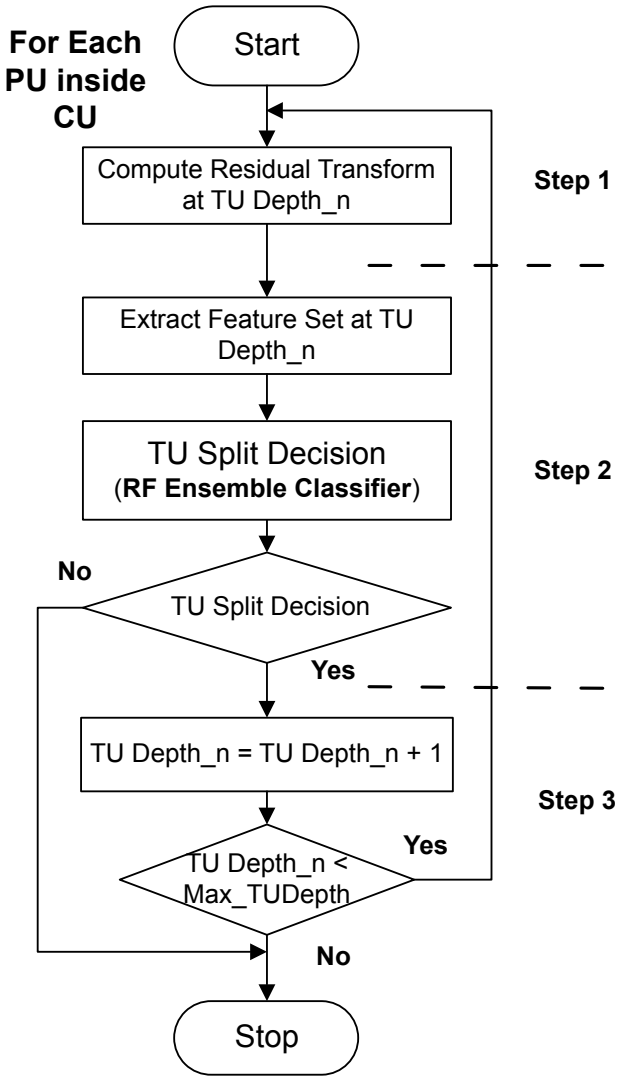


FIGURE 5.8: Flow-Charts of the Overall Fast Encoding Algorithm : TU Split Decision in residual quad-tree inside a PU

The proposed ensemble classifier can also be run in offline-only mode or online-only mode by disabling the other model. In offline-only mode, the online training is disabled.

### 5.5.6 Classifier Training and Evaluation

Training of the RF classifier is done in both the offline and online modes to generate classifier models for Skip mode selection and TU and CU Split decision. For offline training model, three video sequences 'BasketBallDrill', 'Vidyo1' and 'PartyScene' are used. 'BasketBallDrill' contain high motion content and large

temporal variations. Similarly, 'PartyScene' contain scene change variations. On the other hand, 'Vidyo1' has low motion content and static regions with homogeneous texture. Using three different sequences enables the classifier to train for different type of video content. Similarly, to avoid overlap between training and testing data and for better validation of classifier performance, training video sequences are encoded at QP values 24, 28, 32, 38, that are different from the QP values used during classifier testing.

For all three sequences, 20 frames were encoded and the features previously selected were extracted for training the three RF classifiers, i.e. for Skip mode decision, TU Split decision and CU Split decision. To build the decision trees in the forest, samples were randomly selected from the input data with replacement. This random selection with replacement may result in replication of some samples from the original data set while some others are left out. Samples that are left out can be used for OOB training accuracy of the classifier. In the design of the RF model, different parameters such as the number of trees in the forest, depth of each decision tree, number of samples at the leaf node and the threshold $\beta$ directly affect the classifier accuracy. To get the optimal values for these parameters, multiple simulations were run for each of these parameters. In these simulations, one parameter was varied at a time while keeping the others constant and its impact on the classifier accuracy was observed. The results of these simulations are shown in Figs. 5.9 to 5.12. The impact of the number of trees on the classifier accuracy is shown in Fig. 5.9, where the number of trees is increased from 1 to 100. As observed in the Fig., after 20 trees the classifier accuracy remains constant, hence the optimum number of trees was defined as 20.

Similar simulations were run by varying the depth of each tree in the forest from 1 to 50 and results are shown in Fig. 5.10. From these simulations, it is clear that increasing the depth of the decision tree beyond 10 does not significantly affect the classifier accuracy but increases the tree size. Hence, the depth of the decision tree was selected as 10. Similarly based upon the results shown in Fig. 5.11, the optimal value for number of samples at the leaf node was defined as 100. To minimize the correlation between different trees in the forest, the number of

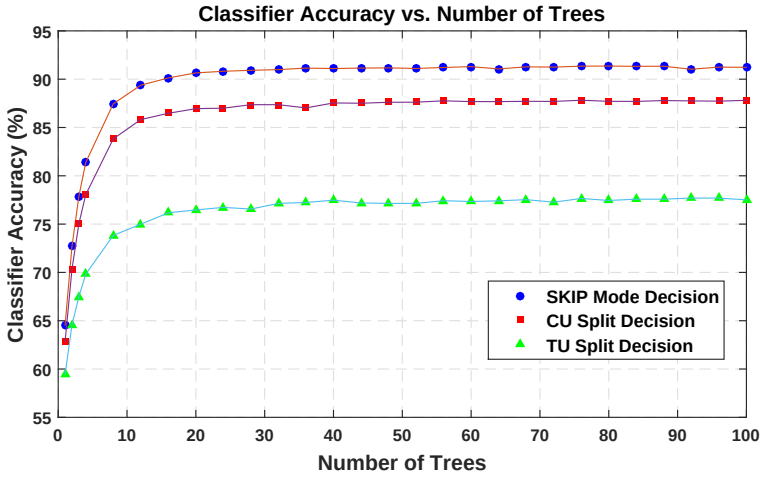


FIGURE 5.9: Relation of RF Classifier Accuracy with Number of Trees in the Forest for Skip Mode, CU and TU Split Decisions

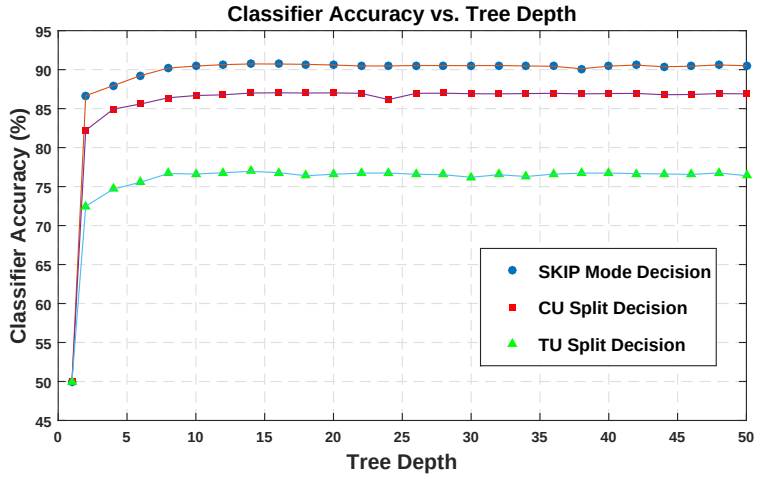


FIGURE 5.10: Relation of RF Classifier Accuracy with Trees Depth in the Forest for Skip Mode, CU and TU Split Decisions

features (K) for the Split decision at each node is kept less than or equal to the total number features (N) according to the Eq. $K \leq \sqrt{N}$. Similarly, to terminate the decision tree, the threshold $\beta$ is used as stopping criteria at each node (Eq. 5.3). Since the choice of this threshold influences the tree size and also the accuracy of the classification, a simulation study was carried out to objectively evaluate its influence on the accuracy, for the three types of fast decision. Based upon the results shown in Fig. 5.12, $\beta$=0.3 was chosen as a good trade-off between speed and accuracy. Moreover, it was found that small variations around this value do not significantly influence the results.

Using the previous optimal values for classifier parameters, the accuracy of the classifier model was evaluated using five video sequences of different resolutions and different video content in terms of texture and motion. Twenty frames of each sequence were used for accuracy measurement. Based on the confusion matrix, which is often used to evaluate the performance of binary classifier models, the true positive rate $TPR = \frac{TP}{PP}$and the true negative rate $TNR = \frac{TN}{NN}$ were computed. The classifier accuracy was computed using Eq. 5.8, where (TP) represents true positive (TP), (TN) represents true negative, (FP) represents false positive, and (FN) represents false negative [43]. PP represents total positive cases and NN represents total negative cases in the data set. For Skip mode decision, true positive represents Skip mode predicted as Skip and true negative represents Non-Skip mode predicted as Non-Skip. Similarly, for CU and TU Split decisions, true positive represents a Split case predicted as Split and true negative represents Non-Split case predicted as Non-Split by the classifier.

$$Accuracy(\%) = \frac{(TP + TN) * 100}{PP + NN}. \tag{5.8}$$

For Skip mode decision, higher TNR rate means a large number of cases correctly classified as Non-Skip, that leads to improvement in video quality. While higher TPR rate means a large number of cases correctly classified as Skip, that results in encoder speed-up. Similarly, for CU and TU Split decisions, a higher TPR rate corresponds to a large number of cases correctly classified as Split, that results in increased video quality. While speed-up gain from classification depends upon higher TNR rate because a large number of cases are correctly classified as Non-Split.

For CU Skip mode decision, the classifier model has an average accuracy of 90.04%, average TPR of 0.88 and average TNR of 0.89 as shown in Table 5.4. Specially for sequences with high motion content, like 'PeopleOnStreet' and 'BlowingBubbles', where the percentage of Skip blocks is small, the proposed classifier achieves small TNR values of '0.96 and 0.90' respectively, as shown in Table 5.4. Small values of TNR results in good video quality. Similarly, for sequences like 'Johnny', with

static regions and low motion, the percentage of Skip blocks is large, which results in TPR of 0.97 and good speed-up.

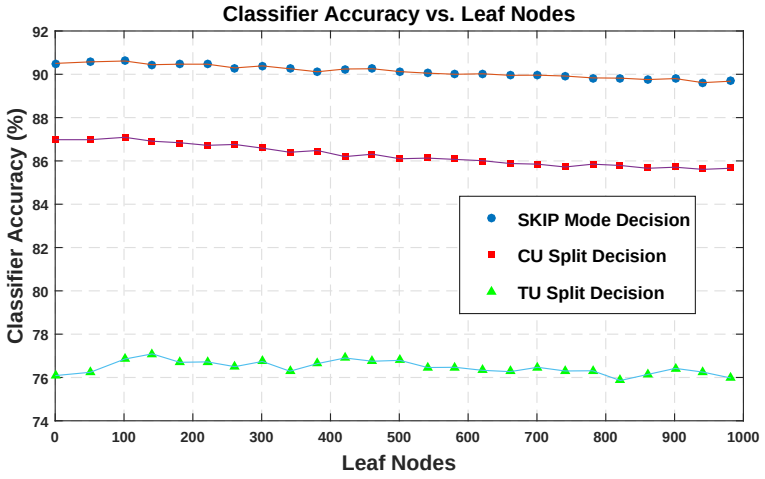


FIGURE 5.11: Relation of RF Classifier Accuracy with Number of Leaf Nodes in the Forest for Skip Mode, CU and TU Split Decisions

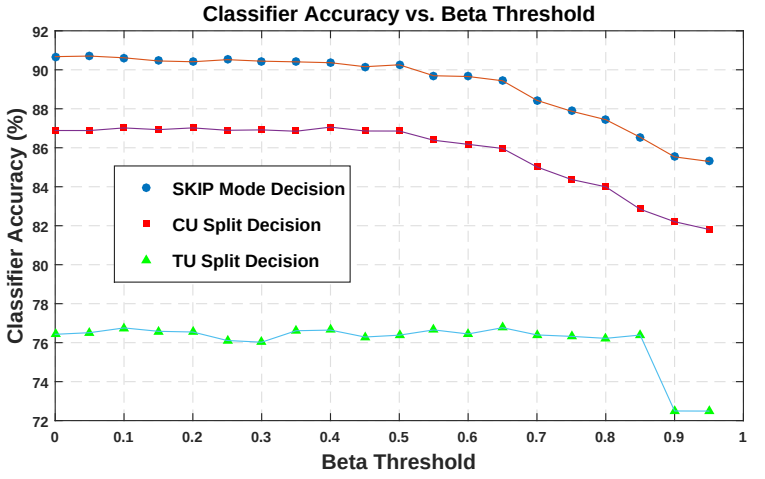


FIGURE 5.12: Relation of RF Classifier Accuracy with Beta Threshold in the Forest for Skip Mode, CU and TU Split Decisions

For CU Split decision, the classifier has an average accuracy of 85.55%, TPR of 0.88 and TNR of 0.83 as shown in Table 5.4. Similarly, for TU Split decision, the classifier has an average accuracy of 82.35%, TPR of 0.74 and TNR of 0.82 as shown in Table 5.4. For sequences with high motion content, like 'BlowingBubbles' and 'PeopleOnStreet' where Split CUs is often necessary, TPR of 0.93 and 0.96 results in good video quality. For sequences with low motion content like 'Johnny', where Split CUs is rarely necessary, high TNR of 0.98 results in good speed-up.

To select the optimal number of training frames for online RF model, several simulations were run for Skip mode selection, CU and TU Split decision, while increasing the number of training frames from 5 to 30. It was found that increasing the number of training frames, results in lower bitrate increase (BDBR) and lower speed-up, with all three cases following an almost linear relationship. A median value of 15 frames was used in the experiments, which is also inline with other previous works [139].

TABLE 5.4: RF Classifier evaluation for Skip Mode selection, CU and TU Split decision

| Stream | Qp | CU Split Decision | | | Skip Mode Selection | | | TU Split Decision | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | TPR | TNR | Accuracy(%) | TPR | TNR | Accuracy(%) | TPR | TNR | Accuracy(%) |
| BlowingBubbles | 27 | 0.93 | 0.57 | 70.42 | 0.84 | 0.88 | 86.50 | 0.85 | 0.72 | 74.81 |
| | 37 | 0.84 | 0.81 | 81.84 | 0.76 | 0.90 | 80.83 | 0.73 | 0.92 | 90.59 |
| BQMall | 27 | 0.92 | 0.82 | 84.38 | 0.94 | 0.89 | 92.42 | 0.80 | 0.79 | 79.00 |
| | 37 | 0.89 | 0.90 | 90.39 | 0.92 | 0.89 | 91.19 | 0.72 | 0.92 | 90.95 |
| Johnny | 27 | 0.85 | 0.93 | 92.87 | 0.97 | 0.77 | 95.05 | 0.70 | 0.88 | 86.57 |
| | 37 | 0.79 | 0.98 | 97.48 | 0.97 | 0.72 | 96.74 | 0.62 | 0.98 | 97.77 |
| BasketBallDrive | 27 | 0.86 | 0.86 | 85.59 | 0.88 | 0.96 | 91.44 | 0.85 | 0.64 | 65.54 |
| | 37 | 0.80 | 0.94 | 93.66 | 0.87 | 0.96 | 88.96 | 0.77 | 0.93 | 92.23 |
| PeopleOnStreet | 27 | 0.96 | 0.67 | 77.39 | 0.90 | 0.95 | 93.09 | 0.79 | 0.53 | 62.49 |
| | 37 | 0.94 | 0.79 | 81.43 | 0.76 | 0.96 | 84.13 | 0.58 | 0.89 | 83.52 |
| Average | | 0.88 | 0.83 | 85.55 | 0.88 | 0.89 | 90.04 | 0.74 | 0.82 | 82.35 |

## 5.6 Results and Discussion

To evaluate the performance of the proposed method, the RF models for early Skip mode selection, early TU Depth termination and early CU depth termination were integrated in the HEVC reference software, HM version 16.10 [126]. Twenty video sequences of different resolutions and diverse content have been used. The RF library 'librf' was used for generation of the proposed RF classifier model [140]. As previously mentioned, offline RF classifier models for Skip mode prediction and CU Split decision have been generated using three sequences, 'BasketBallDrill', 'PartyScene' and 'Vidyo1' using the bootstrap method. For online generation of the RF model, 15 frames of each stream were used for training, assuming that the video content characteristics do not significantly change during the limited duration of these test sequences. In longer video sequences without real-time constraints, temporal segments (i.e., video shots) should be identified and retraining

should be done at shot boundaries. Simulation results were obtained for Low-Delay Main B profile and Random Access profile in HEVC for QP values of 22, 27, 32, 37 under the common test conditions [60]. The platform used for evaluation is an Intel Core i5 machine running at 1.70 GHz with 8 GB of RAM.

Coding efficiency is measured using BDPSNR (dB) and BDBR (%) [56]. The gain in processing speed (TS%) is measured in terms of the difference between the coding time of the HM reference encoder and the proposed RF-based fast encoding, as given by Eq. 5.9, where $T_r$ and $T_p$ are the encoding times of HM reference and proposed RF, respectively. For comparison of different approaches, analyzing the speed-up gains i.e. TS alone, does not provide an absolute and fully meaningful performance measure because in general an increase in TS also increases BDBR. Hence, we have used a fusion measure (FM) that merges both BDBR and TS into a single performance indicator, defined by Eq. 5.10. Such indicator measures the amount of bit rate overhead per complexity reduction unit. Therefore, any fast coding method presenting a smaller value of FM means that complexity reduction is achieved at a lower cost in terms of bit rate overhead.

$$TS(\%) = \frac{(T_r - T_p) * 100}{T_r}, \tag{5.9}$$

$$FM = \frac{BDBR * 100}{TS}. \tag{5.10}$$

To evaluate the contribution of each RF classifier, the speed-up results individually achieved by the Skip mode selection, CU Split and TU Split decisions, for Random Access profile, are shown in Table 5.5. These results were obtained using RF offline models for Skip mode, CU Split and TU Split decisions. According to these results, the maximum contribution to speed-up is due to early Skip mode selection scheme that gives an average speed-up of 47.47%. The speed-up due to early CU depth termination is 47.01% while early TU depth termination scheme yields an average speed-up of 10.48%.

Table 5.5: Individual performance achieved by Skip mode selection, CU and TU Split decision (RA, Offline)

| Stream | Resolution | Skip Mode Selection | | | | CU Split Decision | | | | TU Split Decision | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BDBR | BDPSNR | TS(%) | FM | BDBR | BDPSNR | TS(%) | FM | BDBR | BDPSNR | TS(%) | FM |
| BasketballPass | 416x240 | 0.51 | -0.024 | 31.35 | 1.63 | 0.72 | -0.035 | 30.42 | 2.36 | 0.20 | -0.010 | 9.07 | 1.78 |
| BlowingBubbles | | 0.49 | -0.020 | 34.40 | 1.44 | 0.57 | -0.024 | 34.05 | 1.67 | 0.07 | -0.003 | 9.58 | 0.68 |
| BQSquare | | 0.22 | -0.009 | 30.76 | 0.72 | 0.33 | -0.014 | 36.64 | 0.89 | -0.04 | 0.002 | 10.95 | -0.45 |
| Flowervase | | 0.50 | -0.027 | 66.93 | 0.75 | 0.15 | -0.008 | 59.76 | 0.26 | -0.01 | 0.000 | 12.22 | -0.11 |
| BasketballDrill | 832x480 | 0.58 | -0.025 | 36.06 | 1.61 | 0.60 | -0.025 | 39.81 | 1.50 | 0.03 | -0.001 | 9.81 | 0.32 |
| PartyScene | | 0.29 | -0.013 | 25.11 | 1.15 | 0.58 | -0.026 | 29.33 | 1.97 | 0.18 | -0.008 | 8.17 | 1.47 |
| BQMall | | 0.39 | -0.015 | 34.54 | 1.14 | 0.63 | -0.023 | 40.07 | 1.57 | 0.12 | -0.004 | 10.65 | 1.23 |
| RaceHorses | | 0.32 | -0.012 | 15.69 | 2.06 | 0.58 | -0.022 | 24.60 | 2.37 | 0.39 | -0.015 | 7.36 | 2.90 |
| Johnny | 1280x720 | 1.26 | -0.035 | 73.12 | 1.72 | 0.26 | -0.006 | 65.33 | 0.40 | -0.02 | 0.001 | 12.64 | -0.28 |
| Vidyo1 | | 0.54 | -0.024 | 73.67 | 0.73 | 0.28 | -0.012 | 64.63 | 0.43 | -0.05 | 0.002 | 12.69 | -0.61 |
| Vidyo3 | | 2.17 | -0.097 | 71.76 | 3.03 | 0.31 | -0.012 | 63.33 | 0.49 | 0.09 | -0.004 | 12.58 | 1.08 |
| Vidyo4 | | 0.93 | -0.038 | 71.14 | 1.31 | 0.30 | -0.013 | 62.96 | 0.48 | 0.03 | -0.001 | 12.19 | 0.36 |
| FourPeople | | 0.62 | -0.024 | 73.22 | 0.84 | 0.08 | -0.003 | 63.55 | 0.12 | -0.01 | 0.000 | 12.89 | -0.12 |
| KristenAndSara | | 0.72 | -0.027 | 71.00 | 1.02 | 0.25 | -0.010 | 64.52 | 0.38 | -0.11 | 0.003 | 12.19 | -1.31 |
| BasketballDrive | 1920x1080 | 0.67 | -0.015 | 36.65 | 1.82 | 1.01 | -0.022 | 45.84 | 2.20 | 0.33 | -0.007 | 9.04 | 2.98 |
| BQTerrace | | 0.62 | -0.009 | 43.50 | 1.42 | 0.60 | -0.010 | 48.09 | 1.24 | 0.19 | -0.004 | 10.01 | 1.86 |
| Cactus | | 0.53 | -0.012 | 45.05 | 1.18 | 1.02 | -0.022 | 46.44 | 2.20 | 0.16 | -0.004 | 10.06 | 1.59 |
| Kimono1 | | 0.59 | -0.018 | 41.00 | 1.43 | 0.83 | -0.025 | 48.03 | 1.72 | 0.27 | -0.008 | 8.82 | 2.39 |
| PeopleOnStreet | 2560x1600 | 0.31 | -0.014 | 20.01 | 1.55 | 0.58 | -0.026 | 22.12 | 2.63 | 0.36 | -0.016 | 7.88 | 2.83 |
| Traffic | | 0.93 | -0.033 | 54.41 | 1.71 | 0.52 | -0.018 | 50.72 | 1.03 | 0.08 | -0.003 | 10.80 | 0.88 |
| Average | | 0.66 | -0.025 | 47.47 | 1.41 | 0.51 | -0.018 | 47.01 | 1.30 | 0.11 | -0.004 | 10.48 | 0.97 |

Table 5.6: HEVC RD performance comparison for offline, online and ensemble classifier models (RA)

| Stream | Resolution | Offline | | | | Online | | | | Ensemble of Offline and Online | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BDBR | BDPSNR | TS(%) | FM | BDBR | BDPSNR | TS(%) | FM | BDBR | BDPSNR | TS(%) | FM |
| BasketballPass | 416x240 | 3.91 | -0.182 | 45.58 | 8.59 | 3.54 | -0.164 | 46.45 | 7.62 | 2.44 | -0.114 | 41.79 | 5.85 |
| BlowingBubbles | | 3.72 | -0.150 | 49.95 | 7.45 | 4.28 | -0.172 | 50.25 | 8.51 | 2.85 | -0.117 | 45.36 | 6.28 |
| BQSquare | | 1.63 | -0.069 | 51.61 | 3.16 | 2.92 | -0.124 | 50.54 | 5.78 | 1.05 | -0.045 | 45.16 | 2.33 |
| Flowervase | | 2.24 | -0.120 | 80.20 | 2.79 | 1.66 | -0.090 | 72.24 | 2.30 | 1.28 | -0.069 | 73.70 | 1.74 |
| BasketballDrill | 832x480 | 2.37 | -0.099 | 54.08 | 4.38 | 2.74 | -0.115 | 54.31 | 5.04 | 1.76 | -0.074 | 50.13 | 3.51 |
| PartyScene | | 2.48 | -0.107 | 41.07 | 6.03 | 5.62 | -0.243 | 52.39 | 10.72 | 2.96 | -0.127 | 42.76 | 6.93 |
| BQMall | | 2.29 | -0.085 | 52.44 | 4.36 | 5.25 | -0.193 | 56.51 | 9.30 | 3.00 | -0.113 | 51.09 | 5.87 |
| RaceHorses | | 2.20 | -0.081 | 33.89 | 6.49 | 7.36 | -0.264 | 47.88 | 15.37 | 4.51 | -0.165 | 37.01 | 12.20 |
| Johnny | 1280x720 | 2.55 | -0.072 | 84.40 | 3.03 | 1.18 | -0.033 | 74.37 | 1.59 | 0.93 | -0.026 | 77.18 | 1.20 |
| Vidyo1 | | 1.93 | -0.085 | 84.73 | 2.28 | 0.86 | -0.037 | 73.52 | 1.17 | 0.68 | -0.030 | 76.40 | 0.89 |
| Vidyo3 | | 3.92 | -0.166 | 83.23 | 4.71 | 2.43 | -0.102 | 73.92 | 3.29 | 1.52 | -0.065 | 76.61 | 1.98 |
| Vidyo4 | | 2.03 | -0.080 | 82.77 | 2.45 | 0.80 | -0.032 | 69.77 | 1.14 | 0.60 | -0.024 | 73.73 | 0.82 |
| FourPeople | | 1.71 | -0.069 | 83.79 | 2.04 | 1.11 | -0.046 | 69.83 | 1.59 | 0.77 | -0.031 | 70.57 | 1.09 |
| KristenAndSara | | 2.12 | -0.079 | 83.25 | 2.55 | 1.03 | -0.040 | 73.73 | 1.40 | 0.72 | -0.027 | 76.63 | 0.94 |
| BasketballDrive | 1920x1080 | 2.41 | -0.053 | 56.92 | 4.24 | 4.90 | -0.105 | 63.32 | 7.74 | 2.81 | -0.061 | 56.42 | 4.98 |
| BQTerrace | | 2.53 | -0.041 | 61.48 | 4.11 | 3.07 | -0.049 | 56.92 | 5.40 | 1.79 | -0.029 | 54.90 | 3.26 |
| Cactus | | 3.26 | -0.071 | 61.50 | 5.29 | 4.80 | -0.101 | 66.86 | 7.18 | 3.22 | -0.074 | 61.44 | 5.23 |
| Kimono1 | | 2.48 | -0.075 | 59.74 | 4.16 | 5.59 | -0.167 | 68.26 | 8.19 | 4.38 | -0.131 | 62.67 | 6.99 |
| PeopleOnStreet | 2560x1600 | 2.57 | -0.114 | 33.88 | 7.59 | 11.49 | -0.497 | 51.75 | 22.20 | 5.55 | -0.243 | 40.30 | 13.78 |
| Traffic | | 3.20 | -0.111 | 68.33 | 4.69 | 5.17 | -0.168 | 73.80 | 7.00 | 3.00 | -0.102 | 68.77 | 4.37 |
| Average | | 2.58 | -0.095 | 62.64 | 4.52 | 3.79 | -0.137 | 62.33 | 6.63 | 2.29 | -0.08 | 59.13 | 4.51 |

Table 5.6 shows a comparison of HEVC implementation with the offline RF model, online model and ensemble of online and offline models for Random Access profile. It is clear from the table that, when implemented individually, both online and offline approaches yield almost the same speed-up, however the online approach produces higher BDBR that results in higher FM score of 6.63 as compared to the FM score of 4.52 for offline implementation. Since the offline RF model has been trained using samples from multiple video streams encoded at different QP values, it adapts better to video sequences with diverse characteristics, such as the high motion content of 'RaceHorses' and 'PeopleOnStreet'. This is the main reason that justifies the lower FM score obtained for these sequences when encoded with the offline model and compared with the online model. On the other hand, the online model performs better than offline, i.e., lower FM score, for video sequences with low motion and homogeneous regions like 'Johnny' and 'KristanAndSara'.

When the ensemble of both online and offline approaches is used, this results in lower BDBR. Hence, the FM score of the ensemble approach shows improved results as compared to either online or offline individual implementations.

### 5.6.1 Reduction in HEVC computation time and RF classifier overhead

In real-time implementation scenarios, computational overhead of fast encoding scheme becomes critical. Table 5.7 shows the reduction in HEVC encoding times and the computational overhead of RF classifier during fast HEVC encoding for different video streams. 'Ref.Enc Time' represents encoding time of HM reference without RF classifier. 'RF.Enc Time' represents the encoding time with RF Classifier integrated in the HM reference software. RTime1, RTime2 and RTime3 represent the reduction in HM Reference encoding time, when each of the three RF classifier for Skip mode selection, CU Split Decision and TU Split Decision are run one by one. OTime1, OTime2 and OTime3 represent the computational overhead time for feature extraction and classification for each of the Skip mode selection, CU Split Decision and TU Split Decision respectively. 'Total Overhead'

represents the time for total computations involved in feature extraction and RF classification during fast HEVC encoding. According to the Table, the computational overhead of TU Split Classifier is higher than the other two classifiers. This is because TU Split Classifier is invoked multiple times for different prediction modes in each CU at a specific depth level. These results show that the maximum overhead of proposed classifier model is 4.13% for 'Johnny'. This sequence has homogeneous regions and low motion, which results in higher speed-up. As a result, the classifier overhead time increases as compared to the total encoding time. Overall, the percentage of classifier overhead for different sequences is far less than the reduction in computational complexity of HEVC. Hence, the proposed classifier design can be integrated in real-time video encoders with quite small computational overhead.

Table 5.7: Reduction in HEVC computation time (sec) and RF classifier overhead (%)

| Stream | Qp | **Ref.Enc** Time | RTime1 Skip | RTime2 C-Split | Rtime3 T-Split | **RF.Enc** Time | OTime1 Skip | OTime2 C-Split | OTime3 T-Split | Overhead Time | Overhead (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BlowingBubbles | 27 | 690.41 | -188.33 | -185.08 | -55.96 | 402.10 | 1.51 | 0.84 | 8.20 | 10.55 | 2.62 |
| | 37 | 521.06 | -229.10 | -257.43 | -49.58 | 195.62 | 0.89 | 0.44 | 3.70 | 5.03 | 2.57 |
| BQMall | 27 | 2786.88 | -711.56 | -783.94 | -215.59 | 1467.07 | 4.83 | 3.52 | 32.02 | 46.14 | 2.75 |
| | 37 | 2246.71 | -896.03 | -1018.83 | -234.39 | 815.92 | 3.69 | 2.26 | 16.34 | 25.47 | 2.73 |
| Johnny | 27 | 4365.71 | -3145.84 | -2822.22 | -479.68 | 698.87 | 5.31 | 3.97 | 14.00 | 23.28 | 3.33 |
| | 37 | 4031.16 | -3297.01 | -2972.77 | -437.75 | 331.71 | 4.09 | 3.39 | 6.22 | 13.69 | 4.13 |
| BasketBallDrive | 27 | 14540.68 | -4158.18 | -6637.82 | -1295.22 | 6692.79 | 21.05 | 15.34 | 136.99 | 173.38 | 2.59 |
| | 37 | 11829.66 | -5741.54 | -7059.54 | -1112.38 | 3512.49 | 14.70 | 10.79 | 63.30 | 88.79 | 2.53 |
| PeopleOnStreet | 27 | 24190.60 | -4084.42 | -4230.83 | -1208.80 | 17432.04 | 58.56 | 34.30 | 401.96 | 494.82 | 2.84 |
| | 37 | 19392.21 | -5006.33 | -6449.83 | -1619.07 | 10817.29 | 41.53 | 29.13 | 224.36 | 295.02 | 2.73 |

## 5.7 Performance Comparison

Tables 5.8 and 5.9 show comparison of the fast encoding method with other state of the art implementations for Low-Delay and Random Access (Main profile), respectively. Since the works of X. Huang [78] and F. Chen [76] do not fully cover both coding configurations, their methods were implemented for the purpose of this comparison. In the case of K.Tai [75], the comparison is made with the results obtained by the authors.

For the case of Low Delay Main profile, the proposed approach gives a maximum speed-up of 79.71% with an average of 54.57%, as shown in Table 5.8. For Random Access profile, the results are slightly better. The maximum speed-up is 84.73% with an average of 62.64%, as shown in the Table 5.9. For video sequences with low motion content and homogeneous regions, such as 'FourPeople', where the number of Skip CU blocks is large and the percentage of Split cases is lower, the proposed approach achieves higher speed-up. For Low Delay Main profile, this is 78.98% with an increase in BDBR of 1.83%, as shown in Table 5.8 for sequence 'FourPeople'. In comparison with recent works, for the same video sequence, X.Huang [78] and F. Chen [76] achieve lower speed-up (35.45%) with an increase in BDBR of 0.72% and 71.59% with an increase in BDBR of 7.03%, respectively. Similarly, for the same sequence in Random Access profile, the speed-up of the proposed approach is 83.79% with an increase in BDBR of 1.71%, as shown in Table 5.9. X.Huang [78] reaches a speed-up of 36.47% with 0.36% of BDBR increase and F. Chen [76] gives speed-up of 62.72% with 0.69% BDBR increase for the same stream. Hence, for sequences with low-motion content, the proposed approach performs better as compared to other fast encoding methods, in terms of speed-up in both HEVC profiles. Similar results can be seen for other sequences like 'Johnny' and 'Vidyo1'. This clearly shows that the proposed approach adapts better to low-motion video content and gives higher speed-up with negligible increase in BDBR as compared to the existing implementations, which under-perform both in terms of speed-up and BDBR.

TABLE 5.8: HEVC RD performance comparison for Low Delay profile

| Stream | Resolution | Proposed Approach | | | X.Huang [78] | | | K.Tai [75] | | | F.Chen [76] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) |
| BasketballPass | 416x240 | 3.12 | -0.147 | 38.24 | 2.58 | -0.120 | 32.93 | 0.60 | -0.029 | 32.00 | 2.86 | -0.133 | 36.21 |
| BlowingBubbles | | 3.99 | -0.154 | 39.87 | 1.85 | -0.072 | 34.18 | 0.74 | -0.030 | 33.00 | 5.39 | -0.206 | 41.45 |
| BQSquare | | 3.02 | -0.117 | 34.30 | 2.71 | -0.109 | 28.70 | 0.77 | -0.028 | 34.00 | 0.93 | -0.038 | 23.61 |
| Flowervase | | 2.99 | -0.140 | 67.83 | 1.68 | -0.079 | 43.36 | — | — | — | 1.88 | -0.086 | 45.59 |
| BasketballDrill | 832x480 | 2.24 | -0.089 | 46.55 | 1.15 | -0.046 | 36.25 | 0.53 | -0.021 | 37.00 | 4.36 | -0.170 | 58.19 |
| PartyScene | | 2.58 | -0.108 | 30.33 | 1.70 | -0.074 | 27.10 | 0.64 | -0.027 | 32.00 | 2.86 | -0.124 | 33.74 |
| BQMall | | 1.86 | -0.071 | 43.32 | 2.08 | -0.085 | 33.73 | 0.85 | -0.033 | 41.00 | 4.55 | -0.183 | 45.94 |
| RaceHorses | | 1.25 | -0.048 | 27.08 | 1.72 | -0.071 | 28.19 | 0.62 | -0.024 | 30.00 | 2.02 | -0.082 | 31.06 |
| Johnny | 1280x720 | 5.45 | -0.139 | 79.31 | 1.31 | -0.036 | 38.77 | 0.88 | -0.024 | 67.00 | 3.94 | -0.101 | 60.40 |
| Vidyo1 | | 3.53 | -0.140 | **79.71** | 0.93 | -0.040 | 37.98 | 0.56 | -0.017 | 61.00 | 6.59 | -0.270 | 72.51 |
| Vidyo3 | | 6.96 | -0.279 | 78.49 | 1.32 | -0.059 | 36.73 | 1.30 | -0.036 | 62.00 | 4.17 | -0.177 | 66.09 |
| Vidyo4 | | 3.70 | -0.130 | 78.29 | 0.99 | -0.036 | 35.81 | 0.88 | -0.021 | 63.00 | 3.96 | -0.145 | 65.92 |
| FourPeople | | **1.83** | -0.063 | **78.98** | **0.72** | -0.027 | **35.45** | 1.00 | -0.034 | 63.00 | **7.03** | -0.238 | **71.59** |
| KristenAndSara | | 3.30 | -0.108 | 77.58 | 1.30 | -0.042 | 38.11 | 0.75 | -0.024 | 63.00 | 3.78 | -0.121 | 62.12 |
| BasketballDrive | 1920x1080 | **1.93** | -0.046 | **49.16** | **0.63** | -0.014 | **39.87** | 0.75 | -0.017 | 41.00 | 3.41 | -0.077 | 61.65 |
| BQTerrace | | 1.74 | -0.032 | 51.89 | 0.71 | -0.014 | 39.48 | 0.95 | -0.016 | 46.00 | 1.76 | -0.036 | 47.89 |
| Cactus | | 2.46 | -0.060 | 53.06 | 1.14 | -0.028 | 37.99 | 1.07 | -0.025 | 43.00 | 3.07 | -0.073 | 59.94 |
| Kimono1 | | 2.43 | -0.079 | 50.49 | 0.73 | -0.025 | 39.77 | 0.68 | -0.022 | 40.00 | 2.23 | -0.076 | 47.66 |
| PeopleOnStreet | 2560x1600 | 1.85 | -0.085 | 27.22 | 1.49 | -0.069 | 29.38 | 0.96 | -0.030 | 50.00 | 3.01 | -0.139 | 38.71 |
| Traffic | | 3.20 | -0.102 | 59.66 | 1.33 | -0.044 | 39.52 | 0.39 | -0.018 | 33.00 | 4.72 | -0.157 | 52.23 |
| Average | | 2.97 | -0.107 | 54.57 | 1.40 | -0.05 | 35.66 | 0.79 | -0.025 | 45.84 | 3.63 | -0.13 | 51.13 |

Table 5.9: HEVC RD performance comparison for Random Access profile

| Stream | Resolution | Proposed Approach | | | X.Huang [78] | | | K.Tai [75] | | | F.Chen [76] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) |
| BasketballPass | 416x240 | 3.91 | -0.182 | 45.58 | 1.78 | -0.083 | 32.03 | 0.97 | -0.047 | 34.00 | 1.03 | -0.049 | 38.52 |
| BlowingBubbles | | 3.72 | -0.150 | 49.95 | 1.52 | -0.062 | 32.20 | 1.42 | -0.060 | 47.00 | 0.60 | -0.024 | 27.95 |
| BQSquare | | 1.63 | -0.069 | 51.61 | 1.42 | -0.062 | 34.73 | 1.11 | -0.043 | 52.00 | 0.40 | -0.017 | 26.30 |
| Flowervase | | 2.24 | -0.120 | 80.20 | 1.08 | -0.058 | 46.93 | — | — | — | 0.66 | -0.035 | 47.55 |
| BasketballDrill | 832x480 | 2.37 | -0.099 | 54.08 | 0.90 | -0.038 | 39.33 | 0.95 | -0.039 | 42.00 | 1.70 | -0.072 | 52.73 |
| PartyScene | | 2.48 | -0.107 | 41.07 | 1.31 | -0.061 | 29.71 | 1.35 | -0.058 | 46.00 | 0.46 | -0.022 | 25.51 |
| BQMall | | 2.29 | -0.085 | 52.44 | 2.03 | -0.083 | 33.62 | 1.59 | -0.061 | 47.00 | 1.85 | -0.076 | 40.93 |
| RaceHorses | | 2.20 | -0.081 | 33.89 | 1.94 | -0.076 | 27.69 | 1.18 | -0.044 | 32.00 | 1.94 | -0.076 | 25.24 |
| Johnny | 1280x720 | 2.55 | -0.072 | 84.40 | 0.75 | -0.023 | 40.13 | 0.88 | -0.024 | 67.00 | 1.67 | -0.051 | 58.67 |
| Vidyo1 | | 1.93 | -0.085 | **84.73** | 0.43 | -0.019 | 38.28 | 1.44 | -0.046 | 69.00 | 1.18 | -0.054 | 66.66 |
| Vidyo3 | | 3.92 | -0.166 | 83.23 | 1.11 | -0.050 | 37.96 | 1.41 | -0.046 | 69.00 | 1.03 | -0.046 | 60.68 |
| Vidyo4 | | 2.03 | -0.080 | 82.77 | 0.67 | -0.026 | 36.83 | 1.26 | -0.039 | 68.00 | 1.55 | -0.061 | 61.51 |
| FourPeople | | **1.71** | -0.069 | **83.79** | **0.36** | -0.015 | **36.47** | 1.00 | -0.034 | 63.00 | **0.69** | -0.026 | **62.72** |
| KristenAndSara | | 2.12 | -0.079 | 83.25 | 0.52 | -0.021 | 39.78 | 0.75 | -0.024 | 63.00 | 1.18 | -0.046 | 59.03 |
| BasketballDrive | 1920x1080 | **2.41** | -0.053 | **56.92** | 0.94 | -0.021 | 39.93 | **1.57** | -0.034 | **45.00** | 3.07 | -0.066 | 55.43 |
| BQTerrace | | 2.53 | -0.041 | 61.48 | 0.78 | -0.016 | 40.42 | 2.02 | -0.032 | 56.00 | 0.95 | -0.020 | 47.08 |
| Cactus | | 3.26 | -0.071 | 61.50 | 1.33 | -0.030 | 38.87 | 1.84 | -0.039 | 52.00 | 1.42 | -0.031 | 55.18 |
| Kimono1 | | 2.48 | -0.075 | 59.74 | 0.96 | -0.031 | 39.72 | 1.25 | -0.037 | 47.00 | 1.99 | -0.064 | 42.89 |
| PeopleOnStreet | 2560x1600 | 2.57 | -0.114 | 33.88 | 1.44 | -0.064 | 28.54 | 1.11 | -0.052 | 38.00 | 1.40 | -0.063 | 33.54 |
| Traffic | | 3.20 | -0.111 | 68.33 | 0.84 | -0.030 | 40.58 | 1.89 | -0.062 | 60.00 | 0.73 | -0.026 | 44.23 |
| Average | | 2.58 | -0.095 | 62.64 | 1.11 | -0.04 | 36.69 | 1.32 | -0.043 | 52.47 | 1.27 | -0.05 | 46.62 |

For sequences like 'PeopleOnStreet' and 'RaceHorses' that have high texture and camera movement, the speed-up of the proposed approach is slightly lower than the average. This is because these sequences are mostly encoded in small CU blocks due to very high motion, which also results in a very low percentage of Skip blocks. Therefore, despite accurate prediction of SPLIT and Non-Skip cases by the classifier, the overall speed-up is relatively low. For video sequences with high motion content like 'BasketBallDrive', the proposed approach achieves a speed-up of 49.16% for Low Delay profile at BDBR rate of 1.93% and 56.92% at BDBR rate of 2.41% for Random Access profile. For the same sequence, X.Huang [78] achieves a lower speed-up of 39.87% at BDBR of 0.63% for Low Delay Profile and K.Tai [75] achieves 45.0% at BDBR of 1.57% for Random Access profile. Hence, the proposed approach also outperforms these recent works in terms of speed-up and BDBR for video sequences with high motion content. Furthermore, the proposed approach based on an ensemble of offline and online classifier models, reveals better adaptation to changes in video content as compared to other approaches that are only based on offline classifier models or use fixed thresholds for prediction.

Performance comparison of the RF based fast video encoding methodology with the content adaptive techniques developed in Chapter 4 is also shown in the Table 5.10. According to these results, RF classifier based technique performs better than content adaptive method in terms of speed-up and gives average speed-up of 62.64% against 61.12% by the content adaptive approach for HEVC RA profile.

## 5.8 Analysis of Results

Overall the average speed-up of the proposed approach is higher than all the other state of the art implementations under comparison in this work for both the Random Access profile and Low Delay Main profile. This speed-up is achieved while maintaining comparative RD performance. Taking into account that 20 video sequences with quite diverse content characteristics and formats were used in the above experiments, generalization to other video sequences is expected to

TABLE 5.10: Comparison of Classification based and Content Adaptive Methods

| Stream | Resolution | RF Classifier | | | Content Adaptive | | |
|---|---|---|---|---|---|---|---|
| | | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) |
| BasketballPass | 416x240 | 3.91 | -0.182 | 45.58 | 1.31 | -0.062 | 47.67 |
| BlowingBubbles | | 3.72 | -0.150 | 49.95 | 0.80 | -0.032 | 45.67 |
| BQSquare | | 1.63 | -0.069 | 51.61 | 0.62 | -0.027 | 46.18 |
| Flowervase | | 2.24 | -0.120 | 80.20 | 0.87 | -0.046 | 64.38 |
| BasketballDrill | 832x480 | 2.37 | -0.099 | 54.08 | 1.93 | -0.081 | 57.46 |
| PartyScene | | 2.48 | -0.107 | 41.07 | 0.63 | -0.030 | 45.73 |
| BQMall | | 2.29 | -0.085 | 52.44 | 2.18 | -0.089 | 48.53 |
| RaceHorses | | 2.20 | -0.081 | 33.89 | 2.21 | -0.086 | 35.04 |
| Johnny | 1280x720 | 2.55 | -0.072 | 84.40 | 1.98 | -0.060 | 79.24 |
| Vidyo1 | | 1.93 | -0.085 | 84.73 | 1.77 | -0.082 | 83.84 |
| Vidyo3 | | 3.92 | -0.166 | 83.23 | 1.54 | -0.068 | 78.83 |
| Vidyo4 | | 2.03 | -0.080 | 82.77 | 2.18 | -0.084 | 78.51 |
| FourPeople | | 1.71 | -0.069 | 83.79 | 1.49 | -0.048 | 83.52 |
| KristenAndSara | | 2.12 | -0.079 | 83.25 | 1.33 | -0.050 | 77.88 |
| BasketballDrive | 1920x1080 | 2.41 | -0.053 | 56.92 | 3.14 | -0.067 | 59.70 |
| BQTerrace | | 2.53 | -0.041 | 61.48 | 1.20 | -0.025 | 61.88 |
| Cactus | | 3.26 | -0.071 | 61.50 | 1.50 | -0.032 | 61.33 |
| Kimono1 | | 2.48 | -0.075 | 59.74 | 2.01 | -0.065 | 53.04 |
| PeopleOnStreet | 2560x1600 | 2.57 | -0.114 | 33.88 | 1.52 | -0.068 | 42.92 |
| Traffic | | 3.20 | -0.111 | 68.33 | 1.34 | -0.047 | 71.05 |
| Average | | 2.58 | -0.095 | 62.64 | 1.58 | -0.058 | 61.12 |

yield very similar performance. Therefor, the proposed fast encoding method is an efficient solution to ease real-time implementation of highly efficient video encoders.

Proposed machine learning based fast video encoding methodology shows promising results both for Low Delay and Random Access profiles in HEVC. These results are also better than the content adaptive methodology developed in Chapter 4. Main difference in the results is because of the ensemble classification based approach used in this work. Random Forests based classifiers designed for CU size selection and Skip mode early prediction perform well for a diverse set of video sequences and have very little classifier overhead. Hence, it can be used both in offline and online configurations as shown by the results. Although the results for online configuration need further improvement in terms of RD performance, especially for video sequences with high motion content. Main reason for poor RD performance of online RF configuration for video sequences with high motion

content is little data for classifier training. As classifier model is not updated once trained, it does not adapt to the variations in video content. This performance can be improved by updating the RF classifier model at regular intervals for video sequences with large texture and motion variations.

The relative gain in encoding time, given as the percentage of computational time reduction of the codec, is the most commonly used metric to evaluate and compare results of fast video coding algorithms. Although, a direct link to real-time video encoding cannot be established with such relative measure, it allows validation and comparison of results obtained from different methods published in the literature. For instance, other works previously published within the scope of real-time processing [74, 93, 141], also use percentage of computational time reduction to evaluate the performance of different fast video coding methods. Nevertheless, a relationship with real-time video processing can be established by measuring the real-time performance of the reference video encoder, or equivalent implementation running the same coding decision functions. Once the run-time of the reference is known for any specific hardware platform, it is straightforward to infer the processing time expected for an implementation using the proposed fast methods. Therefore, an indirect link can always be established between the results presented in this research and real-time performance on specific platforms.

## 5.9 Summary

In this chapter, a machine learning based fast encoding method to reduce the computational complexity of HEVC was developed. Detailed comparative analysis of filtering and wrapper based methods for selection of optimal features set was carried out. According to this analysis, any of the two methods can be used for feature selection. Random Forests classifier models were trained for prediction mode selection and CU and TU quad-trees early termination. Both online and offline trained classifier models were used, that can work in standalone or ensemble configuration. Performance results showed that Random Forests classifier model

is robust against changes in video content and performed equally well both for low and high motion video streams. According to the results, classifier overhead in HEVC implementation was minimal that helps in using such robust solutions in real-time video applications. In the next chapter, this ensemble classification based approach will be used for fast coding of light field images.

# Chapter 6

# Fast Coding of Light Field Images

This chapter presents the third contribution of the research work. Chapter 5 described Random Forests based fast video encoding of HEVC. Encoding of Light Field (LF) images is an emerging field where HEVC can be used to get maximum compression efficiency. In order to reduce the high computational cost of HEVC, fast video encoding techniques developed in the last chapter are applied for low complexity coding of Light field images. First, brief overview of different Light field image representations is given. This is followed by detailed analysis of light field images data to explore the unique features of light fields that can assist in fast encoding. Proposed fast encoding methodology is based on machine learning that encodes light field images in sub-aperture and pseudo video representations. A score fusion based feature selection approach has been proposed for selection of optimal parameters to train the classifier model. Selected features are used to train Random Forests based classifiers that intelligently predicts CU depth level and Skip prediction mode for fast encoding. At the end, results of the proposed work for different light field image formats are presented and compared with other state of the art.

## 6.1 Overview of Light Field Images

Light field images provide a three-dimensional representation of visual scenes by capturing the light intensity as well as the direction of the light rays, which gives information about scene depth and volume in addition to spatial information and color [142]. The multidimensional nature of light field images is characterized by both spatial and angular resolutions expanding conventional 2D images beyond light intensity representation. Light field imaging has applications in computer vision, multimedia communications, depth estimation, medical imaging, object recognition, computational photography and virtual reality [143, 144]. Light field images are represented using a 4D plenoptic function L(u,v,s,t), where u and v represent the spatial dimensions and s and t represent the angular dimensions [145] . These dimensions are described in the Fig. 6.1. This results in huge amount of data to be stored, delivered and shared using different types of devices and networks. Hence, efficient compression techniques for light field images are under investigation and further research is going on to meet the requirements of JPEG Pleno standardization [117]. Since light fields can be represented in various formats, the compression efficiency has been investigated for such representation formats because different data structures lead to significant variations in correlation among various blocks in a light field image.

## 6.2 Light Fields Representation Formats

Various formats are used for representation of light fields data. The two most common formats to represent light filed images are :

- Sub-Aperture Images
- Lenslet Images

According to the 4D plenoptic function L(u,v,s,t), s and t axes refer to the number of viewpoints to visualize the same scene from various angles while u and v axes

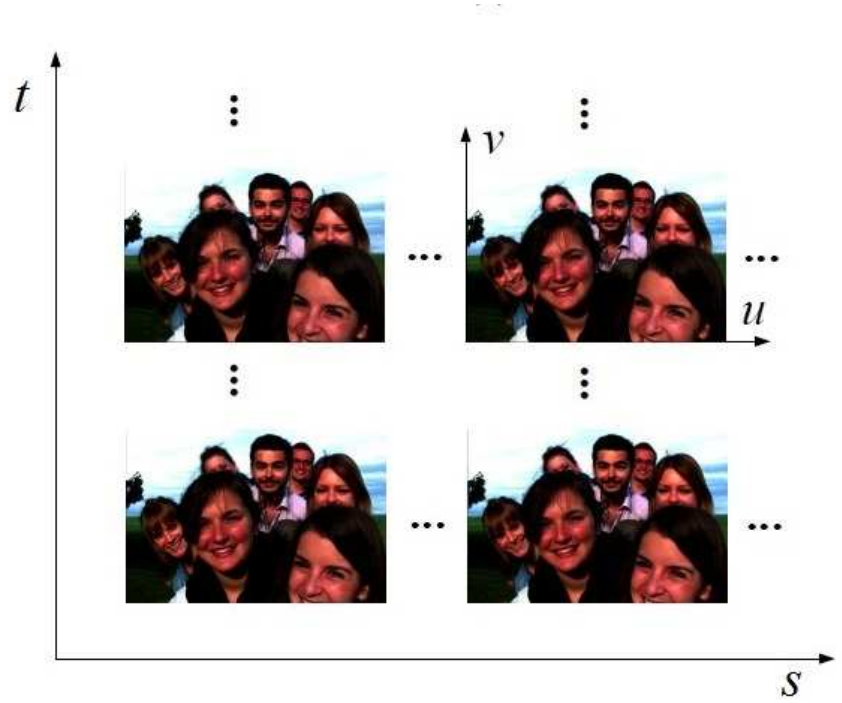


FIGURE 6.1: Light Field Image Axes

refer to the spatial dimensions. Thus the 2D array of images that represent the slightly different perspectives of the visual scene are Sub-Aperture Image representation of the light field. The second representation is a matrix of micro-images, which result from the micro-lens structure of the optical acquisition system. This representation known as Lenslet Images shows the collection of light rays coming from different view points on to a single point [146]. These formats are shown in Figs. 6.2 a and b, respectively.

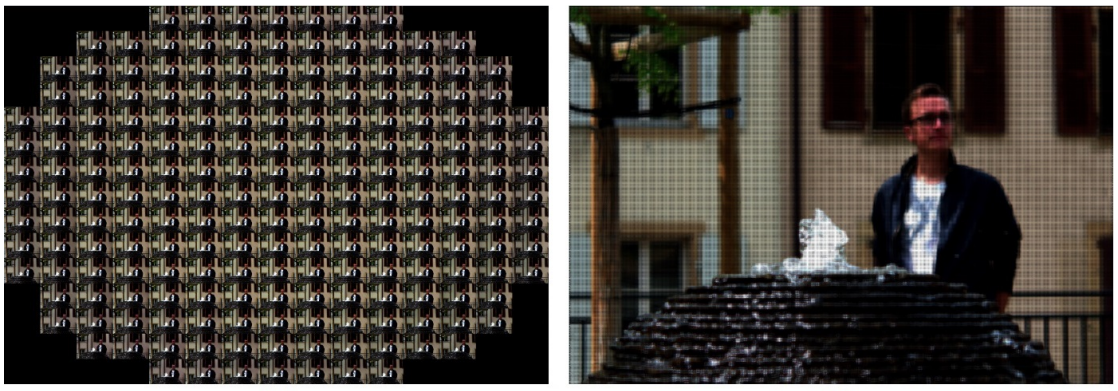


FIGURE 6.2: Light Field Image Representation Formats (a) Sub-Aperture Image (b) Light View Image

A third representation of light field images is formed by arranging the sub-aperture images over time to form a pseudo sequence of light fields. Raster or spiral scanning order can be used to arrange the sub-aperture images into a pseudo-sequence [143] as shown in the Fig. 6.3. Pseudo-sequence representation of light field images enables their efficient encoding using latest video codecs like HEVC because of high correlation among sub-aperture images in the pseudo-sequence.

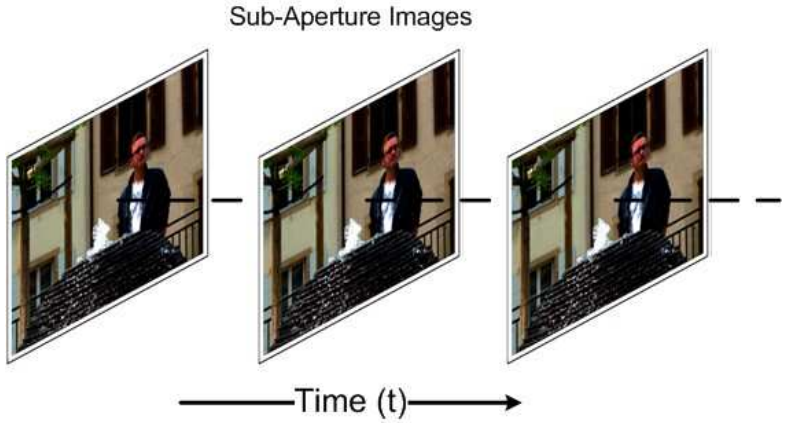


FIGURE 6.3: Light Field Image Representation Formats - Pseudo Sequence

Many researchers [112, 113, 115] have shown that coding of light field images in different representation formats using HEVC gives better coding efficiency as compared to still image coding standards like JPEG. This has been discussed in detail in Chapter 3. Using HEVC for coding of light fields increases the computational complexity that can affect processing of light fields in real-time scenarios. In this research, a machine learning based solution has been developed for fast encoding of light field images using HEVC.

## 6.3 Analysis of Light Field Images

As mentioned in the Chapter 3, fast coding of light field images using standard video coding algorithms and various representation formats has not been investigated so far. HEVC is the most efficient coding standard for coding of light field images while lenslet and stack of sub-aperture images are mostly used formats. It is known that HEVC achieves high compression efficiency at the cost

of high computational complexity [64, 97]. Although many fast coding methods have been investigated to reduce the computational complexity of HEVC for coding conventional video content that include statistics based methods [69, 147, 148] and machine learning approaches, such as decision trees, support vector machines (SVM) and Random Forests [87, 94, 138], the impact of the specific characteristics of light field representation formats in the computational complexity of HEVC is mostly unknown. For instance, pseudo video sequences generated from light fields are quite different from conventional videos because the sub-aperture images are highly correlated with little disparity among different views, as compared to the high temporal variations and dynamic scene changes in natural video sequences.

Computational complexity of HEVC has been discussed in Chapter 4 and it is known that the two main sources of high complexity in HEVC are selection of optimal depth level in CU quad-tree structure and optimal prediction mode. No study is currently available in the literature, relating the coding speed due to algorithmic complexity and the coding efficiency of light field images. To study the characteristics of standard light field encoding, a statistical analysis is presented in this section using lenslet, sub-aperture and pseudo video formats. Pseudo video is formed by temporally organizing the stack of sub-aperture views in raster or spiral scan order [114]. For this analysis, two light field images, represented in different formats were encoded at QP values 22, 27, 32 and 37, using HEVC still image and low delay profiles.

In pseudo video format, high correlation exists among collocated sub-aperture images. To analyze the correlation between collocated neighboring blocks in pseudo video using different scanning orders, the correlation between CU Depth levels in current CU and collocated CU in the neighboring sub-aperture image was calculated and results shown in Fig. 6.4. These graphs show the correlation coefficient for spiral and raster scan order for light field images 'Friends_1' and 'StonePillarsOutside'. According to these graphs, higher correlation exists between CU depth in current CU and neighboring collocated CU for the case of spiral scan order as compared to the raster scan order.

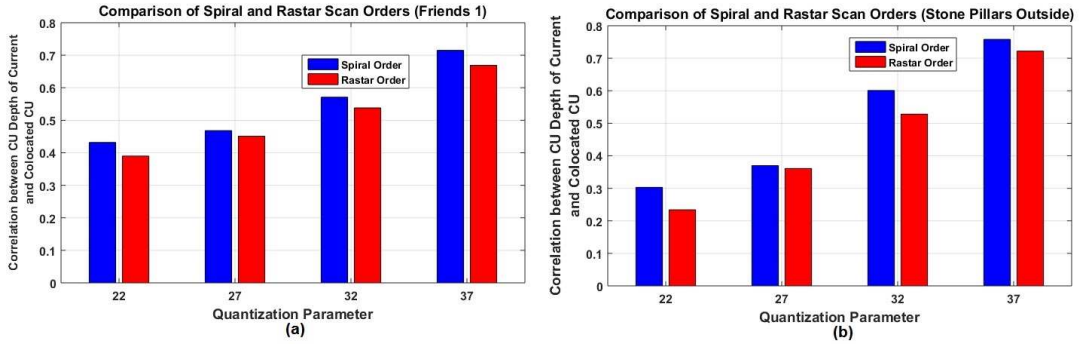


FIGURE 6.4: Correlation of CU Depth Levels with Spiral and Raster Scan Orders (a) 'Friends-1' (b) 'StonePillarsOutside'

### 6.3.1 Distribution of CU Split Decisions

In order to analyze the distribution of CU Split cases, only the percentage of CU Split cases at CU depth levels 0, 1 and 2 is considered because at depth level 3, CUs are not further split. As shown in Fig. 6.5, for light field pseudo videos, the percentage of CU Split cases decreases with increase in QP value, however average values almost remain the same for different Light Fields at CU depth levels 0, 1 and 2. Maximum percentage of CU Split cases exist at depth level 0 and Qp value 22.

According to the Fig. 6.5, average percentage of split cases for different light field pseudo videos is less than 50% i.e. more than 50 cases are non-split. If these non-split cases can be accurately predicted using a machine learning based approach, it will greatly reduce the computational complexity in HEVC. Fig. 6.5 also shows, for the light field pseudo videos, percentage of CU Split cases is less than 1 % at depth level 2 and QP value 37. Hence this is a strong indication that for QP values 37 and above, search for optimal CU depth level can be stopped at depth level 2 without much loss in RD performance.

Light Field images in lenslet format were also encoded using HEVC still image profile for the analysis. The distribution of CU Split Cases for lenslet images are shown in Fig. 6.6, where it can be observed that the percentage of split cases at different depth levels is always higher as compared to the pseudo video format. For lower QP values, this percentage is more than 90%, which implicitly leads

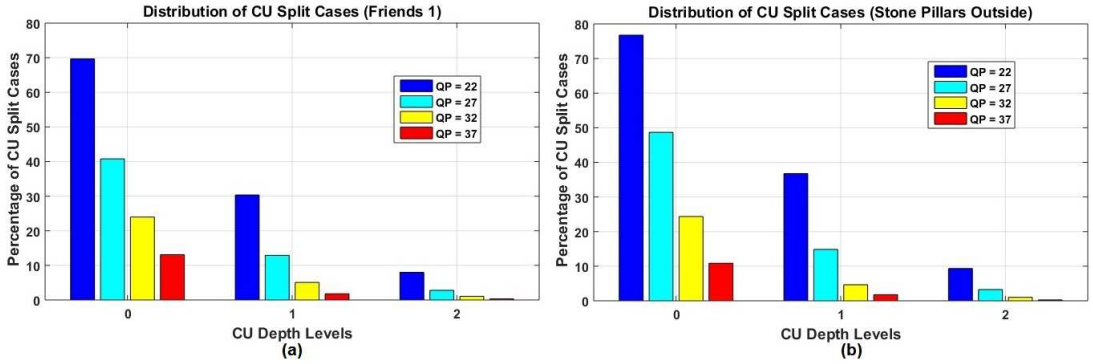


FIGURE 6.5: Distribution of CU Split Cases at various QP values in LF Pseudo Video (a) 'Firends-1' (b) 'StonePillarsOutside'

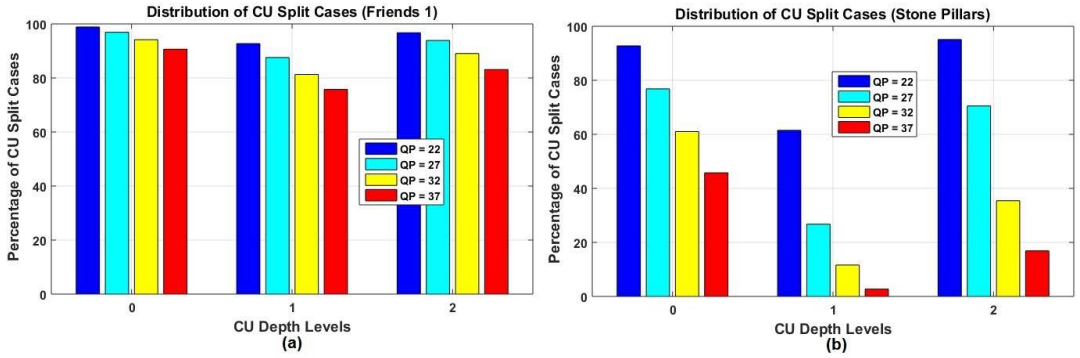


FIGURE 6.6: Distribution of CU Split Cases at various QP Values for Lenslet Images (a) 'Friends-1' (b) 'StonePillarsOutside'

to higher computational complexity. This also means that to efficiently encode lenslet images with high percentage of split cases, Split vs Non-Split classification accuracy will directly impact the RD performance. It is also clear from the Fig. 6.6, that for depth level 0 and Qp = 22, the percentage of split cases is higher than 95%. Hence, this is also a strong indication that search for the optimal prediction mode at depth level 0 can be skipped to speed-up the encoding without much loss in RD performance.

Furthermore, for the stack of sub-aperture image format, there exists significant correlation between the co-located CU blocks in the neighboring sub-aperture images, in addition to the neighboring CU blocks in the current image. This correlation between the collocated neighboring blocks in the stack of sub-aperture images is shown in Fig. 6.7. CU blocks in the top, left and top-left sub-aperture images are similar to the block in the current image and CU depth level of current

FIGURE 6.7: Stack of Sub-Aperture Images: Colocated Neighboring CUs

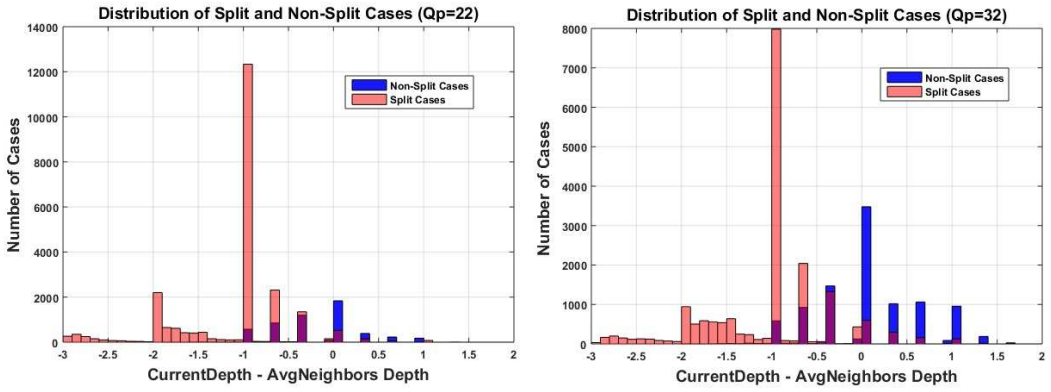


FIGURE 6.8: 'Bikes': Distribution of Split and Non-Split CU Cases (Qp = 22,32)

CU block can be efficiently inferred from its co-located neighbors. Figs. 6.8 and 6.9 show distribution of split and non-split cases for 'Bike' and 'Friends-1' images at QP values 22 and 32 in sub-aperture image format. This distribution has been drawn by calculating the number of cases occurring for each difference (depth level of current CU minus average depth level of collocated neighboring CUs).

This analysis provides significant insight about the occurrence of Split cases. When the depth level of current CU is greater than or equal to the average depth level of collocated neighboring CUs, the percentage of split CU cases is low. Similarly,

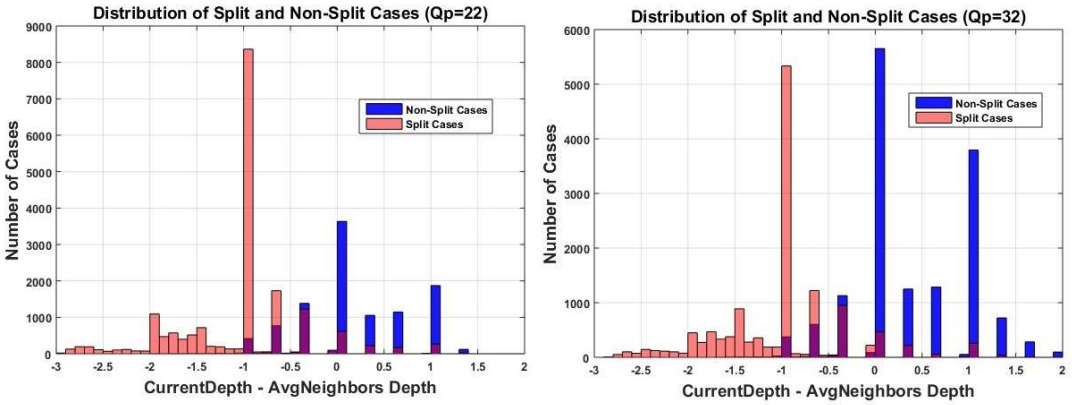


FIGURE 6.9: 'Friends-1': Distribution of Split and Non-Split CU Cases (Qp = 22,32)

when depth level of current CU is less than the average neighboring depth level, the percentage of split cases is high. If difference between the current CU depth level and average neighboring depth level is -1 or lower, then the current block can be split into 4 sub-blocks with high probability of being the correct decision.

### 6.3.2 Distribution of Prediction Modes

During HEVC encoding at each CU depth level, different inter and intra prediction modes are evaluated for selection of the optimal mode inside each CU. Fig. 6.10 and 6.11 show the percentage of Skip and Intra modes in inter frames at CU depth levels 0,1,2 and 3 for two Light Field pseudo-videos. According to Fig. 6.10, the percentage of Skip mode for different light fields is greater than 60% on average. At higher QP values and CU depth levels, this percentage is greater than 80%. This shows consistent and high pseudo-temporal correlations in light field images. If a machine learning based classifier can accurately predict the Skip mode, then prediction of unnecessary PU modes can be skipped to speed up the HEVC encoding. Also the percentage of Skip mode is more than 95% for CU depth 3 at QP = 37. Hence for depth level 3, if QP is equal to or greater than 37, the Skip mode can be selected as the optimal mode without exhaustive search throughout all prediction modes.

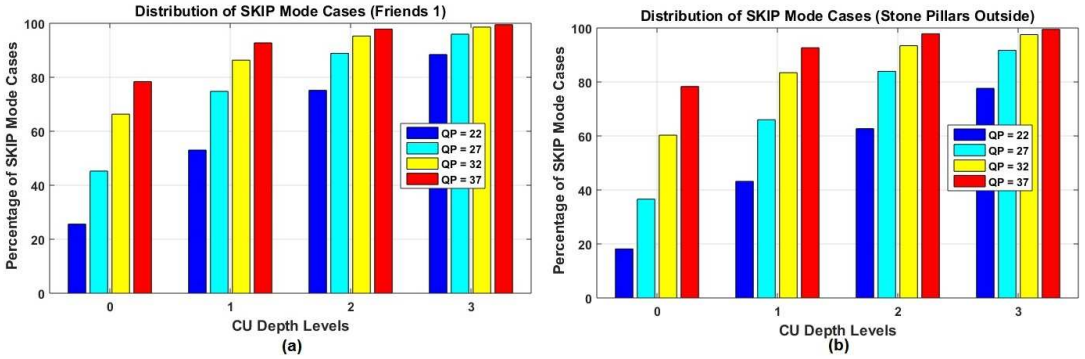


FIGURE 6.10: Distribution of Skip mode at various QP values in LF Pseudo Videos (a) 'Friends-1' (b) 'StonePillarsOutside'

In HEVC inter frames, intra prediction modes are used for prediction of CU blocks that cannot be optimally predicted from the co-located neighboring blocks. According to Fig. 6.11, the percentage of Intra modes at different CU depth levels is around 0.1% on average, which clearly shows that high pseudo-temporal correlation exists and checking of intra mode in the light field pseudo video format can be skipped without much degradation in the RD performance.

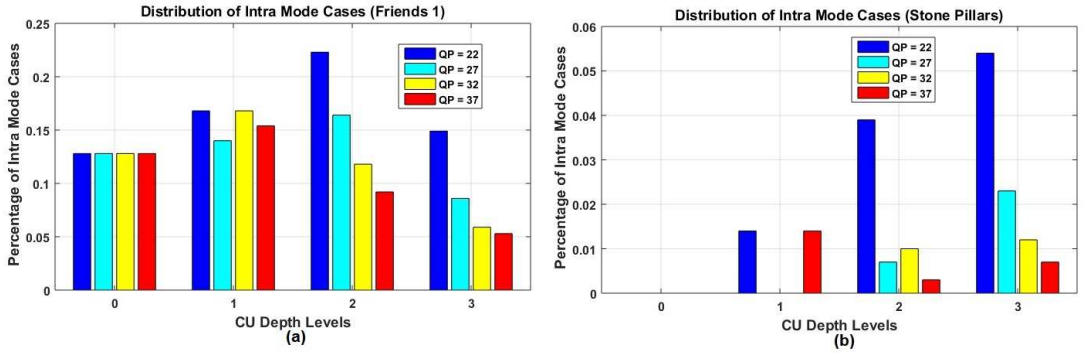


FIGURE 6.11: Distribution of intra modes in inter frames at various QP values in LF Pseudo Videos (a) 'Friends-1' (b) 'StonePillarsOutside'

This distribution of prediction modes was confirmed with other light field images from the data set. Overall this analysis indicates that in light field pseudo-video representation, there are high percentages of non-split CU cases and Skip modes, which can be used to reduce the encoding computational complexity, if accurately predicted and used in early encoding decisions in HEVC. Also additional correlations in stack of sub-aperture images can be exploited to further speed-up the encoding process for still images. For the lenslet representation there is a high

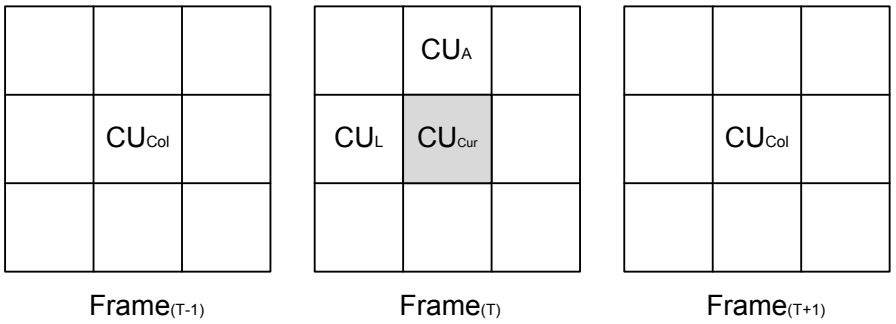


FIGURE 6.12: Neighboring CU Blocks in Current, forward and backward co-located images

percentage of CU split cases, thus accuracy of the prediction scheme is also critical to preserve the RD performance. In the following sections, a new approach for fast encoding of light fields is proposed, based on these specific characteristics and using Random Forests for early prediction of the best coding mode decisions and early CU depth termination. Feature selection methodology for classifier training is also discussed in detail.

## 6.4 Feature Selection Methodology

In classification based methods, a set of relevant features is extracted from the video data and a classifier model is trained to take the early decisions. Selection of optimal features is critical to obtain better classifier accuracy, as the performance of the classifier model highly depends upon the selected feature set. In the proposed methodology, a large set of relevant features is first identified and extracted from the light fields data. To exploit the high spatial and temporal correlations in the light field pseudo-video, many features that depend on the neighboring blocks are used in the feature set.

For computation of such features, top, left and co-located blocks in the forward and backward sub-aperture images were used, as shown in Fig. 6.12. The feature value is computed according to the Eq. 6.1 below [69]:

$$\mathrm{F} = \sum_{i=1}^{N} w_i a_i, \tag{6.1}$$

where N is taken as 4 and value of $\mathrm{w}_i$ is 0.2 for top and left neighbors and 0.3 for co-located neighboring CU blocks in forward and backward reference images. Features that depend on the neighboring blocks include average neighbors depth, neighbors Skip mode count, neighbors average coefficient count and neighbors average motion vectors.

Pixel Variance in a block gives information about texture and smoothness in the video frame. Variance can be computed using the Eq. 6.2 below where M and N represent the width and height of the CU block and $\mu_{m,n}$ represent the pixel mean.

$$Var_{m,n} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(F\left(i,\ j\right) - \mu_{m,n}\right)^2, \tag{6.2}$$

$$\mu_{m,n} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} F(i,\ j). \tag{6.3}$$

Similarly Sobel gradient is also used as a texture feature of each CU block. [123]. Magnitude of Sobel edge can be computed using Eq. 6.4.

$$\mathrm{Mag}_{m,n} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \ |\mathrm{H}_{i,j}| + \ |V_{i,j}|, \tag{6.4}$$

where

$$\mathrm{H}_{i,j} = \ (\ \mathrm{X}_{i-1,j+1} + \ 2 * \mathrm{X}_{i,j+1} + \ \mathrm{X}_{i+1,j+1}) - (\mathrm{X}_{i-1,j-1} + \ 2 * \mathrm{X}_{i,j-1} + \ \mathrm{X}_{i+1,j-1}), \tag{6.5}$$

$$V_{i,j} = (X_{i+1,j-1} + 2 * X_{i+1,j} + X_{i+1,j+1}) - (X_{i-1,j-1} + 2 * X_{i-1,j} + X_{i-1,j+1}). \tag{6.6}$$

$H_{i,j}$ and $V_{i,j}$ represent horizontal and vertical components of the Sobel gradient. Sum of absolute differences (SAD) among pixels of current and collocated block in neighboring frame gives information about temporal variations in a video frame. Hence, SAD with collocated CU has also been used as a feature.

Features related to the current block are non-zero coefficient count of current block, motion vector variance of the current block, total encoded bits and RD Cost of the current block.

A total of 19 features were used for early CU Depth prediction in inter frames, 13 features for early CU depth prediction in intra frames and 14 features for early Skip mode prediction.

### 6.4.1 Score Fusion

For selection of the optimal features, information gain and Random Forests based feature importance were used to rank the feature set [149]. Information gain provides a measure of the significance of each individual feature for classification. While RF based feature importance measures the impact of a sub-set of features on the classification. Using both the ranking criteria, a score fusion function was implemented to combine the ranks assigned by information gain and feature importance into a single measure of feature relevance. Combining both the methods merges the goodness of each into a single metric. First information gain and Random Forests based feature importance are used to assign different scores to each feature. Then both scores are normalized according to the Eq. 6.7 [150], where Fs(x) is the score to be normalized, Fmin and Fmax are the minimum and maximum scores obtained for each method, respectively. Fn(x) is the normalized score obtained for each case.

$$Fn(x)=\frac{\mathrm{Fs\,(x)\,-Fmin}}{\mathrm{Fmax-Fmin}}, x\in FeatureSet \tag{6.7}$$

Finally, the normalized scores of both the methods are combined using the fusion function in Eq. 6.8 below:

$$\mathrm{C(x)} = \sum_{k=1}^{m} \frac{\mathrm{Fn(x)}}{\mathrm{m}}, x\in FeatureSet. \tag{6.8}$$

The algorithm proposed for optimal feature selection, using the combined score C(x), is shown as pseudo code below. Initial feature set contains all the N features. In each iteration of the feature selection loop, an RF classifier model is trained using the feature set S and classifier accuracy is computed. Then the feature with the minimum combined score is removed from the set S and classifier model is generated again with the reduced set S. This process is repeated N times, till only one feature is left. Among all the N classifier models, the feature set for which the classifier accuracy is highest, is selected as the optimal feature set.

**Algorithm 1:** Optimal Feature Selection

**input :** Training Data, N Features, Combined Score
**output:** Optimal Feature Set

```
FeatureSet(S) ←FeatureSet(N)
for i ← 1 to N do
    ClassifierModel ←RFTraining (FeatureSet)
    Accuracy[i] ←Accuracy of ClassifierModel
    MinIndex ← 1
    for j ← 1 to S do
        if CombinedScore[j]<MinScore then
            MinScore ←CombinedScore[ j ] MinIndex ←j
    Remove Feature at MinIndex from FeatureSet(S)
MaxAccuracy ←Accuracy[ 1 ]
MaxIndex ← 1
for k ← 1 to N do
    if Accuracy[k] >MaxAccuracy then
        MaxAccuracy ←Accuracy[k]
        MaxIndex ←k
Optimal Feature Set ←Feature Set with MaxAccuracy
```

TABLE 6.1: Optimal Feature Set for Skip mode Selection and Early CU Depth Termination

| | **Skip Mode Selection** | **CU Split Decision (Intra)** | **CU Split Decision (Inter)** |
|---|---|---|---|
| 1 | CU Depth Current | CU Depth Current | CU Motion Vectors Variance |
| 2 | Neighbors Skip Count | CU Neighbors Average Depth | CU Neighbors Motion Vectors Average |
| 3 | Sobel Gradient | CU Pixel Average | Prediction Mode of Current Block |
| 4 | CU Pixel Variance | CU Pixel Variance | CU Pixel Variance |
| 5 | Total Encoded Bits | Non-zero Coefficients of Current Block | Non-zero Coefficients of Current Block |
| 6 | – | RD Cost of Current Block | Neighbors Skip Count |
| 7 | – | Sobel Gradient of Current Block | Sobel Gradient of Current Block |
| 8 | – | Total Encoded Bits | Total Encoded Bits |
| 9 | – | – | Skip Flag |

Using the above algorithm for optimal feature selection, 5 features were selected for Skip mode prediction, 8 features for early CU split decision in Intra blocks and 9 features for early CU split decision in inter blocks. These features are shown in the Table 6.1.

## 6.5 Proposed Fast Coding System

This section describes the proposed fast encoding framework for low complexity coding of light field images using HEVC, which uses Random Forests based ensemble classifiers for early decisions about prediction modes and early CU depth termination during HEVC encoding of light fields in different representation formats. Based upon the detailed analysis of specific characteristics in light field images, certain fixed thresholds are also used to simplify the HEVC encoding process. Random Forests has been selected because of the promising results obtained for fast video coding using ensemble classification in the last chapter. Other reasons for preference of Random Forests over various machine learning based methods

for fast video encoding are already discussed in detail in Chapter 5, section 5.1.1. Structure of Random Forests Classifier is also described in Chapter 5.

### 6.5.1 Proposed Fast Prediction Mode Selection

Light field pseudo videos have high percentage of Skip prediction units (PU) as compared to all the other PUs at different CU depth levels as shown in Fig. 6.10. Similarly, according to Fig. 6.11, percentage of intra PUs at different CU depth levels in inter frames, is very low. The proposed methodology for fast prediction mode selection comprises the following steps:

- If current CU depth level is 3 and Qp value is greater than or equal to 37, Skip mode is selected as the optimal mode and checking of all the other prediction modes is skipped. This is a fixed decision, which deals with an extreme case, based on the evidence found in the previous statistical study.
- The generic case uses a binary classifier to take Skip vs. non-Skip decision at each CU depth level. If classifier takes a Skip decision, checking of all the other prediction modes at current CU depth level is skipped. In the case of non-Skip decision, all other prediction modes are evaluated
- For inter frames, after evaluation of all the other prediction modes if value of coded block flag (cbf) is zero, checking of intra modes is skipped.

### 6.5.2 Proposed Fast CU Split Decision for Intra and Inter Frames

The percentage of non-Split CUs in Light Field Pseudo videos is high especially at higher CU depth levels and high Qp values, as shown in Fig. 6.5. On the other hand, in Lenslet images, the percentage of Split CUs is higher at lower CU depth levels and lower Qp values, as shown in Fig. 6.6. The proposed methodology for

fast CU split decision for different light field image formats comprises the following steps :

- For Lenslet images, if Qp value is less than 27 and CU depth level is 0, search for optimal prediction mode is skipped at current depth level and CU is split into 4 sub CUs. This also is a fixed decision, which deals with an extreme case, based on the evidence found in the previous statistical study.
- For sub-aperture images, if the difference between the average depth level of neighboring blocks and current CU depth level is greater than or equal to 1, search for optimal prediction modes at current CU depth level is skipped and CU is split in 4 sub CUs.
- For light field pseudo videos, if Qp value is greater than or equal to 37 and current CU depth level is 2, CU is not further split.
- For all light field image formats, a binary classifier is used at current CU depth level to take the Split vs. non-Split decision. If a non-split decision is taken, searching of optimal CU depth at higher depth levels is skipped. The proposed binary classifier works after all the prediction modes at current CU depth have been evaluated.

### 6.5.3 Design of Random Forests Classifier

Proposed fast encoding framework is based on Random Forests. Three separate RF classifiers were trained for early Skip mode selection, early CU depth termination for intra and inter view coding, respectively. All the three classifiers work in binary mode to take the Skip vs. non-Skip decision or Split vs. non-Split decisions. Light field images in pseudo-video format, do not have scene changes or high temporal variations, hence the RF classifier models were trained offline. Four light field images, 'Magnets-2', 'Bush', 'RustyHandle' and 'Sphynx' were used as thr training data-set. For classifier training, these images were encoded at QP values 22, 27, 32 and 37 using HEVC. Light field images used for classifier training were excluded from the light field images used in testing of the fast encoding methodology.

### 6.5.4 Overall Fast Encoding Algorithm

Overall, the proposed fast encoding framework for light field images is based on three RF classifiers in addition to fixed decisions taken to speed up the encoding process in different representation formats. The first RF classifier is used for early CU depth termination in intra blocks. This classifier model is used in Lenslet and stack of sub-aperture image formats for still image profile, and all intra profile in pseudo video format. The second RF classifier works for early selection of Skip mode while the third classifier works for early termination of CU depth, in inter coding of sub-aperture images in pseudo video format. These classifiers are used for fast encoding of light field images in LD Main B and RA profiles. All the three classifiers work in binary mode and classifier models are trained offline. During encoding at CU level, features are extracted from the video data at run-time and fed to the classifier model. RF Classifier model for early Skip mode prediction works at all CU depth levels i.e. 0,1,2 and 3. On the other hand, classifier models for early CU depth termination in intra and inter modes work only at CU depth levels 0, 1 and 2 as CU is not further split at depth level 3. Flow-charts of the proposed overall algorithm at CU level are shown in Figs. 6.13 and 6.14.

## 6.6 Results and Discussion

The proposed fast encoding framework for encoding of light field images was integrated in HEVC reference software HM version 16.10 [126]. The hardware platform used for evaluation is an Intel Core i5 machine running at 1.70 GHz with 8 GB of RAM. A set of twelve light field images of different texture and lighting conditions is selected as test set from light field images dataset [62]. For sub-aperture image format, the size of one sub-aperture image was cropped to make it multiple of 64. The pseudo-video format was obtained by converting sub-aperture images into a pseudo-temporal video sequence using the spiral scan order. Random Forests (RF) classifier models described in previous section have been trained offline using four light field images, 'Bush', 'RustyHandle', 'Magnets-2' and 'Sphynx'. These

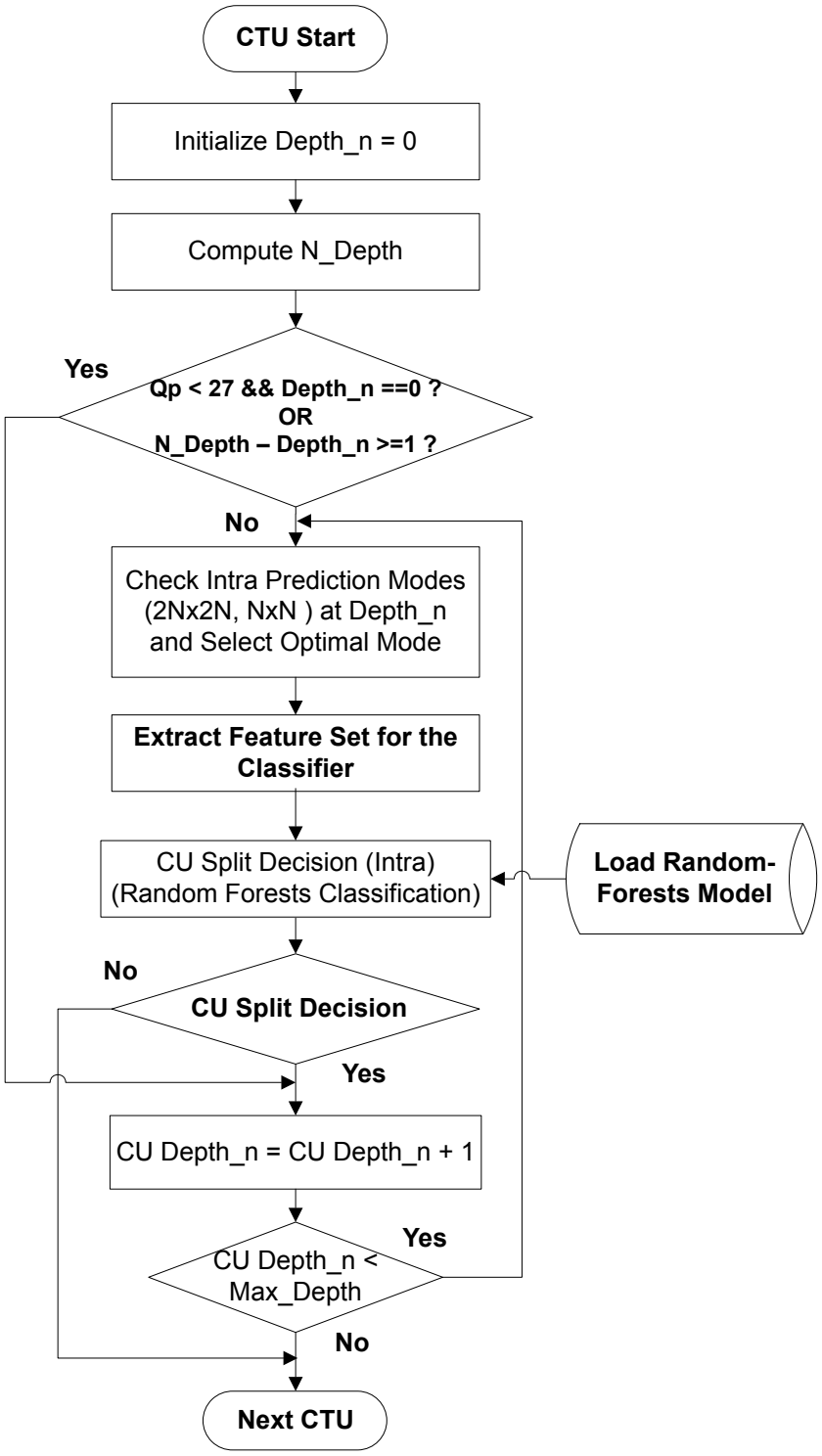


FIGURE 6.13: Flow-Chart of the Fast Encoding Algorithm for All Intra Frame

training images are not part of the testing set. RD performance of the proposed fast implementation and speed-up in terms of execution time are obtained for four HEVC profiles, namely still image profile for lenslet and stack of sub-aperture images, All Intra profile, Low Delay Main B and Random Access profile for pseudo

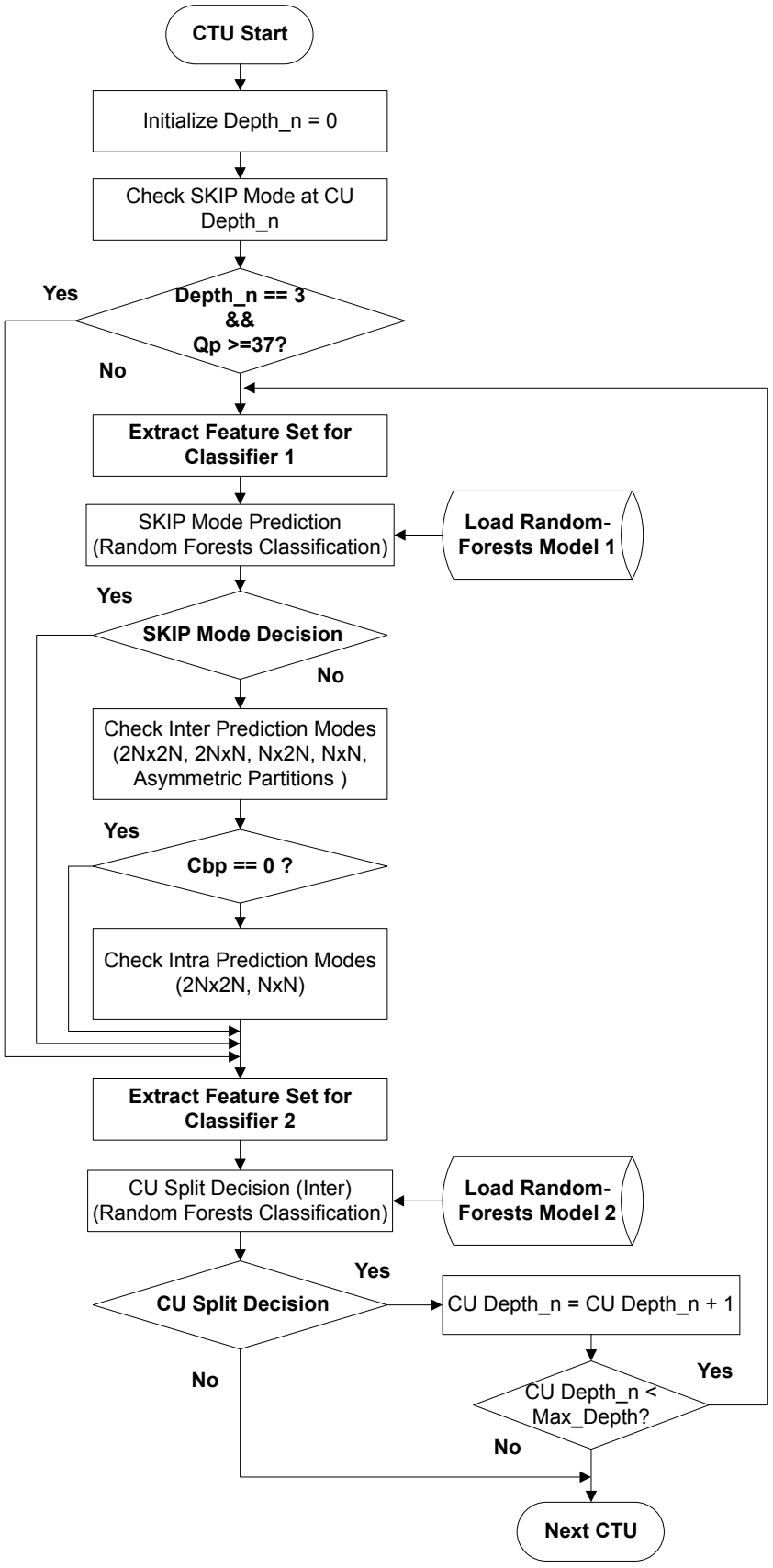


FIGURE 6.14: Flow-Chart of the Fast Encoding Algorithm for Inter Frame

TABLE 6.2: Performance Results of early CU Split Decision for HEVC Still Image Profile

| Stream | LensLet Image | | | Sub-aperture Image | | |
|---|---|---|---|---|---|---|
| | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) |
| Ankylosaurus_&_Diplodocus_1 | 0.36 | -0.032 | 25.54 | 2.52 | -0.067 | 73.46 |
| Bikes | 0.27 | -0.018 | 27.68 | 0.64 | -0.037 | 47.63 |
| Color_Chart_1 | 0.42 | -0.035 | 28.08 | 4.05 | -0.141 | 70.11 |
| Danger_de_Mort | 0.25 | -0.013 | 28.75 | 0.53 | -0.030 | 44.46 |
| Desktop | 0.10 | -0.011 | 23.62 | 0.72 | -0.041 | 54.95 |
| Flowers | 0.26 | -0.016 | 30.08 | 0.25 | -0.017 | 38.43 |
| Fountain_Vincent_2 | 0.27 | -0.021 | 18.76 | 0.65 | -0.036 | 54.45 |
| Friends_1 | 0.17 | -0.015 | 31.55 | 0.56 | -0.029 | 53.32 |
| ISO_Chart_12 | 0.31 | -0.027 | 23.57 | 1.56 | -0.090 | 52.73 |
| Magnets_1 | 0.33 | -0.028 | 23.11 | 2.27 | -0.077 | 70.95 |
| Stone_Pillars_Outside | 0.21 | -0.015 | 30.33 | 0.74 | -0.044 | 55.42 |
| Vespa | 0.22 | -0.012 | 37.60 | 1.70 | -0.078 | 58.85 |
| Average | 0.26 | -0.020 | 27.39 | 1.35 | -0.057 | 56.23 |

video format.

Speed-up in Execution Time (TS%) is measured in terms of the difference between the coding time of the HM reference encoder and the proposed RF-based fast encoding, as given by Eq. 6.9 below, where $T_r$ and $T_p$ are the encoding times of HM reference and proposed RF implementations, respectively.

$$TS(\%) = \frac{(T_r - T_p) * 100}{T_r}. \tag{6.9}$$

For RD performance measurement of the proposed scheme, light field images in different formats were encoded using QP values 22, 27, 32 and 37 and BDBR (%) and BDPSNR (dB) have been computed [56].

Results of the proposed approach for early CU split decision for still image profile in the case of lenslet and sub-aperture image formats are shown in Table 6.2. According to these results, early CU split decision gives average speed-up of 27.39% and 56.23% for lenslet and sub-aperture formats respectively. This speed-up is achieved with an increase in BDBR of 0.26% for lenslet images and 1.35% for sub-aperture images. The speed-up achieved for sub-aperture image format is high as compared to the lenslet image format. This is because of the high correlation between co-located neighboring blocks in top and left sub-aperture views, which

has been exploited to avoid computations at unnecessary CU depth levels. Moreover, in lenslet images, complex texture pattern, even in the smooth regions of the image, leads to small CU sizes and proportionally large percentage of CU split cases during HEVC encoding. This large number of CU split cases reduces the gain in speed-up.

Results of the prediction schemes for early CU split decision and early Skip mode prediction for pseudo video format is shown in Table 6.3. According to these results, early CU-Split decision for intra and inter frames give average speed-up of 44.73% and 57.60% respectively. This speed-up is achieved at the expense of increase in BDBR of 1.5 % and 1.44% respectively. Similarly, speed-up because of early Skip mode selection is 39.68% with increase in BDBR of 0.76%. Speed up for early CU split decision in inter frames is more than the intra frames because for inter coding, high correlation among co-located neighbors in successive frames result in higher percentage of non-split cases. Early prediction of these non-split cases gives more speed up.

The overall performance of the proposed approach for pseudo videos in three HEVC profiles is shown in Table 6.4. Average speed-up for All Intra, Low Delay Main B and Random Access profiles is 44.73%, 56.54% and 62.18% respectively. This speed-up is achieved at the expense of a small increase in BDBR of 1.5%, 2.20% and 1.5% respectively. A maximum speed-up of 71.68% is achieved for the pseudo sequence 'Magnet-1' in Random Access profile, because it contains smooth texture and plain background that results in higher percentage of Skip prediction mode and non-split cases. On the contrary, pseudo sequence 'Bike' with high texture gives minimum speed up of 52.44% for Random Access profile as the high texture increases number of CU split cases that leads to lower speed up. Overall, for encoding of pseudo video representation, maximum average speed-up is achieved with Random Access profile as compared to other two profiles.

Table 6.3: Performance Results of CU Split Decision Intra and Inter CU and Skip Mode Decision

| Stream | Resolution | CU Split Decision Intra | | | CU Split Decision Inter | | | Skip Mode Decision | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) |
| Ankylosaurus_&_Diplodocus_1 | | 3.06 | -0.089 | 67.71 | 3.05 | -0.064 | 66.03 | 1.52 | -0.031 | 42.20 |
| Bikes | | 0.57 | -0.035 | 29.60 | 1.18 | -0.044 | 48.21 | 0.25 | -0.009 | 31.91 |
| Color_Chart_1 | | 4.62 | -0.185 | 63.17 | 1.84 | -0.045 | 56.10 | 1.72 | -0.040 | 36.35 |
| Danger_de_Mort | | 0.46 | -0.026 | 30.17 | 1.39 | -0.052 | 58.29 | 0.58 | -0.022 | 40.94 |
| Desktop | | 0.69 | -0.040 | 42.52 | 1.33 | -0.042 | 56.42 | 0.63 | -0.020 | 42.94 |
| Flowers | 624x432 | 0.16 | -0.011 | 20.64 | 0.68 | -0.026 | 55.23 | 0.29 | -0.011 | 41.53 |
| Fountain_Vincent_2 | | 0.52 | -0.031 | 39.63 | 0.99 | -0.032 | 51.21 | 0.28 | -0.009 | 31.56 |
| Friends_1 | | 0.56 | -0.031 | 39.75 | 0.78 | -0.027 | 59.35 | 0.57 | -0.019 | 42.98 |
| ISO_Chart_12 | | 2.13 | -0.129 | 39.02 | 0.85 | -0.024 | 56.67 | 0.34 | -0.008 | 36.66 |
| Magnets_1 | | 2.70 | -0.095 | 65.57 | 2.76 | -0.062 | 65.88 | 2.39 | -0.052 | 44.88 |
| Stone_Pillars_Outside | | 0.69 | -0.039 | 49.63 | 0.83 | -0.026 | 58.37 | 0.37 | -0.011 | 41.55 |
| Vespa | | 1.78 | -0.086 | 49.34 | 1.62 | -0.051 | 59.47 | 0.17 | -0.006 | 42.62 |
| Average | | 1.50 | -0.066 | 44.73 | 1.44 | -0.041 | 57.60 | 0.76 | -0.020 | 39.68 |

Table 6.4: Performance Results for All Intra, Low Delay B and Random Access Profiles

| Stream | Resolution | All Intra Profile | | | Low Delay B Profile | | | Random Access Profile | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) | BDBR | BDPSNR | TS(%) |
| Ankylosaurus_&_Diplodocus_1 | | 3.06 | -0.089 | 67.71 | 1.99 | -0.042 | 67.74 | 2.25 | -0.045 | 71.14 |
| Bikes | | 0.57 | -0.035 | 29.60 | 2.33 | -0.081 | 44.77 | 1.28 | -0.048 | 52.44 |
| Color_Chart_1 | | 4.62 | -0.185 | 63.17 | 6.42 | -0.146 | 57.54 | 2.33 | -0.057 | 60.03 |
| Danger_de_Mort | | 0.46 | -0.026 | 30.17 | 1.98 | -0.070 | 54.36 | 1.54 | -0.057 | 62.13 |
| Desktop | | 0.69 | -0.040 | 42.52 | 1.68 | -0.045 | 57.78 | 1.87 | -0.059 | 62.75 |
| Flowers | 624x432 | 0.16 | -0.011 | 20.64 | 1.71 | -0.055 | 50.79 | 0.76 | -0.030 | 59.47 |
| Fountain_Vincent_2 | | 0.52 | -0.031 | 39.63 | 1.21 | -0.035 | 47.69 | 0.85 | -0.028 | 54.90 |
| Friends_1 | | 0.56 | -0.031 | 39.75 | 1.12 | -0.031 | 58.47 | 0.88 | -0.030 | 63.77 |
| ISO_Chart_12 | | 2.13 | -0.129 | 39.02 | 2.39 | -0.060 | 55.89 | 1.39 | -0.043 | 60.92 |
| Magnets_1 | | 2.70 | -0.095 | 65.57 | 3.07 | -0.056 | 67.75 | 2.17 | -0.046 | 71.68 |
| Stone_Pillars_Outside | | 0.69 | -0.039 | 49.63 | 0.70 | -0.018 | 57.08 | 0.71 | -0.022 | 62.70 |
| Vespa | | 1.78 | -0.086 | 49.34 | 1.77 | -0.050 | 58.59 | 1.93 | -0.060 | 64.24 |
| Average | | 1.50 | -0.066 | 44.73 | 2.20 | -0.057 | 56.54 | 1.50 | -0.044 | 62.18 |

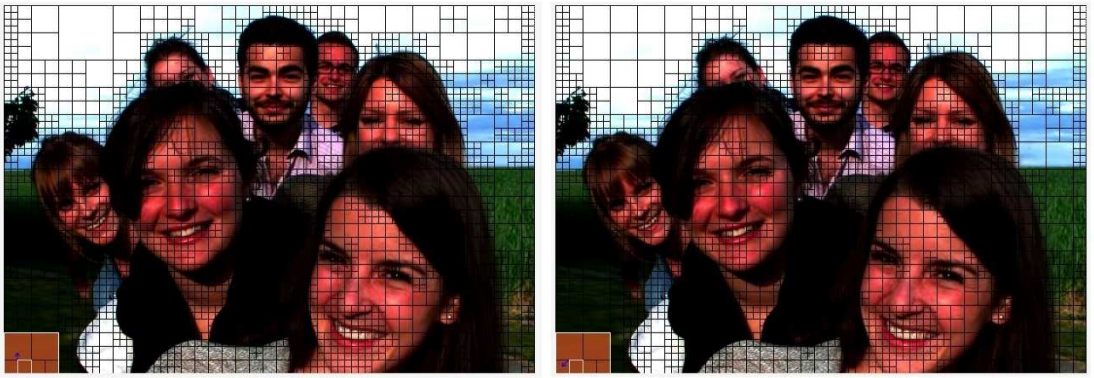


FIGURE 6.15: CU Partitions in All Intra Profile 'Friends-1' (a) HM Reference (b) Proposed Scheme

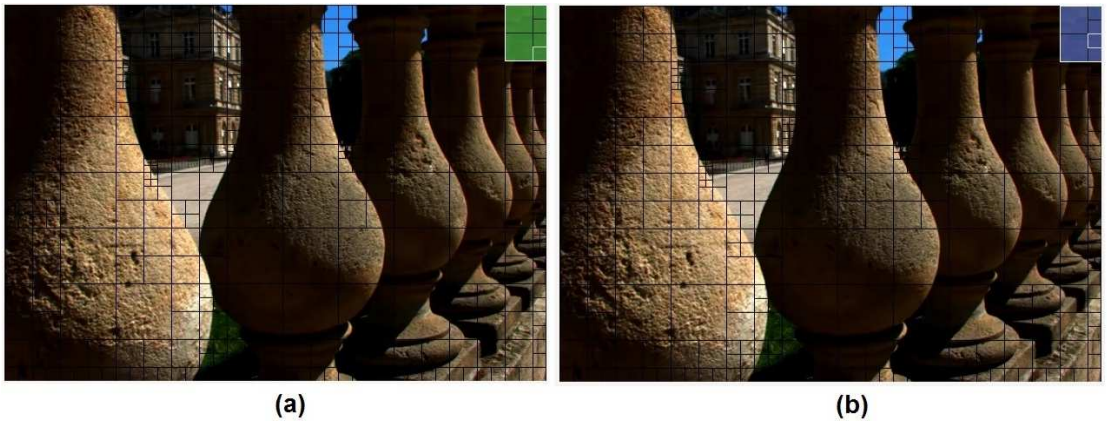


FIGURE 6.16: CU Partitions in RA Profile 'StonePillarsOutside' (a) HM Reference (b) Proposed Scheme

Results of the proposed RF based CU partition decisions for Intra and Inter frames are shown in Fig. 6.15 and 6.16. Fig. 6.15 shows CU partitions in one frame of light field image 'Friends-1' after HM reference encoding and after proposed fast encoding for intra CUs at Qp value 27. According to these results, both for smooth regions and the regions with high texture in the frame, CU partitions of proposed scheme are similar to those of the HM reference encoder partitions.

Fig. 6.16 shows similar results for inter CU partitions for RA profile in one frame of light field image 'StonePillarOutside' after encoding at Qp value 27. These results indicate that the proposed approach closely adapts to different texture and scene depth conditions without compromising the RD performance.

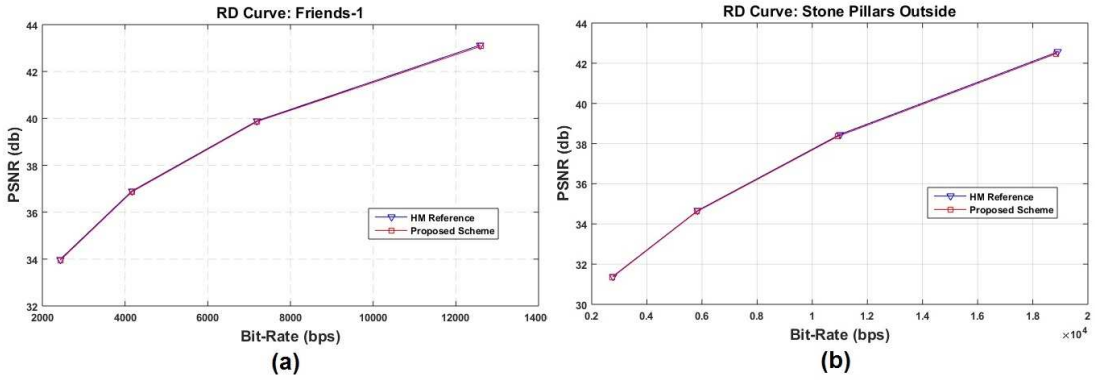


FIGURE 6.17: RD Performance Curves for All Intra Profile at Qp 22,27,32,37 (a) 'Friends-1' (b) 'StonePillarsOutside'

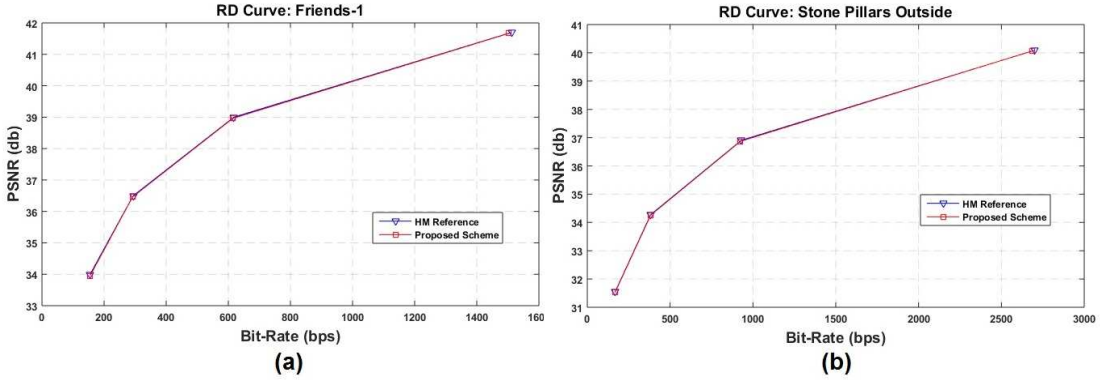


FIGURE 6.18: RD Performance Curves for Random Access Profile at Qp 22,27,32,37 (a) 'Friends-1' (b) 'StonePillarsOutside'

RD performance curves of the proposed approach and HM reference for all intra profile are shown in Fig. 6.17 for two light field images 'Friends-1' and 'StonePillarsOutside'. Fig. 6.18 shows similar results for Random Access Profile. These RD performance curves have been drawn at Qp values 22, 27, 32, 37. According to these curves, RD performance of the proposed approach closely resembles the HM reference. Moreover, this RD performance is achieved with substantial speed-up for all three HEVC profiles as shown in the Table 6.4.

### 6.6.1 Performance Comparison

For performance comparison of the proposed approach with other recent works, three techniques [78, 104, 119] used for fast encoding of video sequences are considered. As no literature is available for low complexity coding of light field images,

TABLE 6.5: Performance comparison of the proposed approach with recent works

| Stream | Proposed Approach | | X.Huang [78] | | D. Fernandez [104] | | T.Shi [119] | |
|---|---|---|---|---|---|---|---|---|
| | BDBR | TS(%) | BDBR | TS(%) | BDBR | TS(%) | BDBR | TS(%) |
| Ankylosaurus_&_Diplodocus_1 | 2.25 | 71.14 | 0.50 | 45.19 | 2.61 | 39.39 | 1.47 | 45.68 |
| Bikes | 1.28 | 52.44 | 1.26 | 37.32 | 0.77 | 32.74 | 0.43 | 38.36 |
| Color_Chart_1 | 2.33 | 60.03 | 0.94 | 42.54 | 1.90 | 34.01 | 1.36 | 43.53 |
| Danger_de_Mort | 1.54 | 62.13 | 1.56 | 41.72 | 1.00 | 36.04 | 0.50 | 42.75 |
| Desktop | 1.87 | 62.75 | 0.74 | 41.06 | 2.32 | 45.37 | 0.45 | 43.43 |
| Flowers | 0.76 | 59.47 | 0.76 | 38.64 | 0.75 | 38.68 | 0.30 | 42.87 |
| Fountain_Vincent_2 | 0.85 | 54.90 | 0.73 | 38.39 | 0.99 | 30.34 | 0.43 | 40.08 |
| Friends_1 | 0.88 | 63.77 | 0.48 | 42.98 | 1.29 | 41.95 | 0.41 | 46.94 |
| ISO_Chart_12 | 1.39 | 60.92 | 1.44 | 39.81 | 1.67 | 37.62 | 0.95 | 46.22 |
| Magnets_1 | 2.17 | 71.68 | 0.63 | 45.91 | 2.12 | 42.25 | 1.44 | 47.15 |
| Stone_Pillars_Outside | 0.71 | 62.70 | 0.44 | 41.43 | 0.98 | 36.86 | 0.34 | 45.60 |
| Vespa | 1.93 | 64.24 | 0.71 | 44.51 | 1.70 | 36.83 | 0.61 | 45.48 |
| Average | 1.50 | 62.18 | 0.85 | 41.62 | 1.51 | 37.67 | 0.72 | 44.01 |

the comparison is made with approaches developed for fast coding of video data. These techniques have been implemented and results are compiled for light fields data. Comparison of these techniques with the proposed method is shown in the Table 6.5. These results are for light fields in pseudo video format encoded using HEVC Random Access profile.

According to the results, the average speed-up of the proposed approach is 62.18%. While speed-up of X.Huang et al.[78] is 41.62%, T.Shi et al. [119] 44.01% and D. Fernandez et al.[104] 37.67%, for the same profile. T.Shi et al.[119] used machine learning technique based on decision trees at different CU depth levels to reduce the complexity of HEVC encoding. Similarly, the statistics based analysis performed by X.Huang et al. [78] and D. Fernandez et al.[104] was also specifically done for video sequences. However in the proposed approach, selection of optimal features was performed for light field images. Moreover, Random Forests classifiers in different representation formats were also designed for light field data and the statistics based analysis was carried out using Lenselet, stack of sub-aperture images and light fields pseudo videos. Methods developed for fast coding of video data do not fully exploit the spatial and temporal correlations available in the light field images. This is the major reason, the proposed approach performs better than the three implementations in terms of speed-up with comparative BDBR figures.

## 6.7 Analysis of Results

Overall, the proposed fast encoding approach based on ensemble classifiers significantly reduces HEVC encoding complexity for all the light field image representation formats. Percentage of speed-up for pseudo video formats is better than the lenslet and sub-aperture image formats, because in pseudo videos early prediction of Skip mode gives extra speed-up. Maximum average speed is achieved for Random Access profile in pseudo videos i.e. 62.18%. For Still Image profile, sub-aperture image format gives more speed-up as compared to the lenslet format. This is because the lenslet format does not have spatial correlation present within the sub-aperture image or among neighboring sub-aperture images. This makes it difficult to predict the CU sizes and prediction modes during encoding. Increase in BDBR for all the image formats is less than 2.5%. Hence, the proposed scheme for fast encoding of light field images finds useful application in fast coding for storage and delivery of light fields using HEVC, with negligible degradation in RD performance.

Performance comparison of the proposed methodology with other recent works also shows that proposed approach performs better than the previous works for light fields data. This also shows that fast coding techniques developed for video sequences may not necessarily ensure better performance when applied to light field images without content specific enhancements. Nevertheless, novel results obtained in this research can be used as benchmark for future work on the fast coding of light field images. This implementation also demonstrates that fast video encoding schemes developed in this research can be seamlessly applied in fast coding of visual media in other fields related to image and videos.

## 6.8 Summary

In this chapter fast video encoding techniques were used to reduce the computational complexity of light field images coding using HEVC. Light field image

representations, stack of sub-aperture images and pseudo videos were used as the input to the HEVC. Unique features of light field image representations were identified and used as input to the RF classifier. Score fusion was applied to select the optimal features for classifier training. Proposed RF classifiers were used for prediction mode selection and CU depth level prediction in Still Image, Low Delay and Random Access profiles of HEVC. According to the results, extra correlations in stack of sub-aperture images helped in speed-up of encoding in Still image profiles. Overall, light filed images encoded as pseudo videos using Random Access profiles gave the highest speed-up. These findings also indicate, fast video encoding techniques developed in this research can be extended to other image and video coding fields.

# Chapter 7

# Conclusion and Future Work

In this research, multiple fast encoding techniques have been developed to reduce the computational complexity of HEVC. These techniques are based on machine learning based algorithms. Systematic approach has been used for selection of relevant features and classifier parameters. Findings of this research show that proposed classifier models are robust against changes in video content and outperform other recent implementations in terms of encoder speed-up and RD performance. Modern trends in video coding are usage of HD and UHD frame resolutions in online video streaming and high frame-rate in multimedia applications. Computational complexity of HEVC is the major limiting factor in its widespread usage on the Internet and end-user multimedia products despite 50% decrease in bit-rates as compared to earlier standards like H.264. Promising results obtained in this research can help in real-time implementation of HEVC in multimedia players and other end-user products with limited processing power and memory. This chapter summarizes and concludes the contributions of the research work. Future enhancements in the research are also discussed.

## 7.1 Concluding Remarks

Overall, this research can be divided into three parts. Focus of first two parts is the development of fast encoding techniques to reduce the encoding complexity of HEVC. Third part is application of the fast encoding techniques in low complexity coding of light field images.

First part of this research is the development of a content adaptive fast encoding framework for HEVC that can be used for video sequences of diverse nature and content. Changes in the video content are a major challenge for fast encoding methods as it may degrade the RD performance or the speed-up. Proposed approach uses global and local spatio-temporal parameters extracted from the video data and intelligently uses them to predict CU depth level and inter prediction modes for fast encoding. Otsu threshold based difference maps are used to quantify the motion content in video frames. Approach intelligently adapts to the changes in video frames for a variety of video sequences. Experimental results show that the proposed approach reduces computational complexity of HEVC by 56.34% on average with negligible increase in BDBR of 1.64% for Random Access profile and performs better than existing implementations. Usage of both global and local parameters to characterize the video data enables better adaptation of the proposed approach to the video content. Hence, video sequences with diverse texture and motion content can be encoded without compromising on video quality or encoding speed.

Second part of this research is the development of a fast encoding method for HEVC based on Random Forests based ensemble classification. A customizable framework of online and offline trained Random Forests classifier models is proposed that can work in multiple user selected configurations. A detailed analysis is carried out for selection of optimal features from video data using filtering and wrapper based approach and a consistent and reproducible methodology is developed to select the best feature sets. Classifier design parameters are optimized through a systematic procedure. The proposed approach models CU and TU

splitting decision and Skip mode selection as a binary classification problem, using Random Forests for early splitting decision and Skip mode prediction. The results provide a conclusive evidence that RF ensemble classifier shows good performance both for sequences with high and low motion content. Particularly, the experimental results demonstrate that the proposed approach achieves increased speed-up for low and high motion video streams while ensuring comparative video quality as compared to the other state of the art. Maximum and average speed-up achieved by the proposed approach is 84.40% and 62.64%, for Random Access profile while for Low Delay Main profile is 78.98% and 54.57%, respectively. Taking into consideration that these gains are obtained with an average BDBR of 2.58% (Random Access) and 2.97% (Low Delay Main), the overall results demonstrate that Random Forests despite being an ensemble classifier, has low classifier overhead and provides an efficient solution for fast video encoding that can be used in multimedia applications and real-time implementations.

Third and the last part is the application of fast video encoding methods developed in this research to the fast encoding of emerging image processing domain like compression of light field images. A machine learning approach has been implemented for low-complexity coding of light field images using HEVC with lenslet, matrix of sub-aperture images and pseudo video formats. Detailed analysis is carried out to identify unique features of light field images that can assist in low complexity encoding and a score fusion based technique is proposed for selection of optimal classifier features. An ensemble classifier using Random Forests is designed for early prediction of CU split decision for Intra and Inter blocks and early Skip mode prediction in inter blocks. The proposed method gives average speed-up of 56.23% for Still Image profile, 62.18% for Random Access, 56.54% for Low Delay Main B and 44.73% for All Intra profile. These results have been obtained with negligible increase in BDBR and negligible change in BDPSNR. These findings can be used as benchmark for further research on fast coding of light field images. Overall, this work demonstrates that using ensemble classifiers to speed-up high efficiency coding of light fields, is an efficient solution to significantly reduce the

computational complexity that enables coding and transmission of light field images in real-time. These promising results also show that fast encoding methods developed in this research can be seamlessly applied to other areas of image and video coding.

## 7.2 Future Work

Findings of the research carried out in this thesis on development of fast video encoding techniques using machine learning are promising in many ways. This work can be used in real-time applications of video coding. This research can be extended in the future in following directions.

- A comprehensive content adaptive framework has been developed for fast encoding that performs equally well for video sequences with simple or complex texture or motion content. Proposed framework only works for HEVC inter profile. Performance of this content adaptive framework can be improved in the future by fast prediction of intra coded frames. Impact of machine learning methods on content adaptive encoding can also be explored in the future.

- Fast encoding techniques developed in this work using ensemble classifier models are robust against changes in video content. These classifiers help in reducing the encoding overhead in HEVC. One limitation of this implementation is that online classifier model is trained only once. To make the scheme more robust against abrupt scene changes, in future implementations classifier model can be updated by retraining it at regular intervals. Similarly, extensions of HEVC such as screen content coding (SCC) or multi-view (MV) coding use the same coding structure as HEVC. Hence, proposed fast encoding methods can be extended in the future to reduce the computational complexity of latest HEVC extensions.

- In this research, Random Forests based machine learning method has shown promising results for fast video encoding. In literature, people have used other machine learning methods such as support vector machines (SVM), decision trees and Bayesian classification for fast encoding. A comparative analysis can be done in the future to compare the performance of Random Forests with other commonly used methods in fast video encoding to analyze the classifier accuracy, overhead of classifier model and impact of various machine learning methods on speed-up and RD performance of the video codec.

- Random Forests classifiers used in this research are based on decision trees. These decision trees are implemented using nested if-else structures in software and no complex mathematical operations like multiplication or devision are required. Such control structures support parallelism and can be easily mapped in Field Programmable Gate Array (FPGA) based hardware implementations [151, 152]. Hence, dedicated hardware based fast video coding solutions can use the proposed research in video playback devices or set-top multimedia boxes in the future.

- For smooth exchange of video content over heterogeneous networks and to match the diverse capabilities of end user devices, video transcoding or transratting solutions are implemented [153, 154]. These solutions enable adjustments of video format, bit-rate or frame resolutions according to the requirements of receiving devices. Such video transcoders are composed of video decoder and encoder running back to back, making it a computationally intensive framework. Fast video encoding methods developed in this research can also be employed in such video transcoding solutions for fast processing.

# Bibliography


[1] M. Ghanbari, *Standard codecs: Image compression to advanced video coding*. IET, 2003.

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/ AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[3] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[4] V. Cisco Public, "Cisco visual networking index: Forecast and trends, 2017–2022," *White Paper*, 2018.

[5] SANDVINE, "Global internet phenomena report," 2018. [Online]. Available: http://www.sandvine.com/resources

[6] A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assuncao, "Data formats for high efficiency coding of Lytro-Illum light fields," *5th International Conference on Image Processing, Theory, Tools and Applications 2015, IPTA 2015*, pp. 494–497, 2015.

[7] M. Tahir, I. A. Taj, and M. Asif, "Content adaptive fast hevc encoding using spatio-temporal parameters," in *17th International Conference on Frontiers of Information Technology (FIT) (Submitted)*. IEEE, 2019, p. xxx.

[8] M. Tahir, I. A. Taj, P. A. Assuncao, and M. Asif, "Fast video encoding based on random forests," *Journal of Real-Time Image Processing*, pp. 1–21, 2019.

[9] M. Tahir, I. A. Taj, P. A. Assuncao, and M. Asif, "Low complexity high efficiency coding of light fields using ensemble classifiers," *Submitted in Journal of Visual Communication and Image Representation*, p. xxx, 2019.

[10] K. Jack, *Video demystified: A handbook for the digital engineer*. Elsevier, 2011.

[11] I. E. Richardson, *H.264 and MPEG-4 video compression: Video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[12] T. Wedi and H. G. Musmann, "Motion-and aliasing-compensated prediction for hybrid video coding," *IEEE transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 577–586, 2003.

[13] P. Symes, *Video compression demystified*. McGraw-Hill Professional, 2000.

[14] G. J. Sullivan, T. Wiegand *et al.*, "Rate-distortion optimization for video compression," *IEEE signal processing magazine*, vol. 15, no. 6, pp. 74–90, 1998.

[15] H. Everett III, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operations research*, vol. 11, no. 3, pp. 399–417, 1963.

[16] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standardsincluding high efficiency video coding (HEVC)," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1669–1684, 2012.

[17] M. Wien, *High Efficiency Video Coding, Coding Tools and Specification*. Springer, 2015.

[18] T. Zhang and S. Mao, "An overview of emerging video coding standards," *GetMobile: Mobile Computing and Communications*, vol. 22, no. 4, pp. 13–20, 2019.

[19] P. Tudor, "MPEG-2 video compression," *Electronics & communication engineering journal*, vol. 7, no. 6, pp. 257–264, 1995.

[20] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital video: an introduction to MPEG-2*. Springer Science & Business Media, 1996.

[21] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Transactions on circuits and systems for video technology*, vol. 8, no. 7, pp. 849–866, 1998.

[22] F. C. Pereira, F. C. Pereira, F. M. B. Pereira, F. Pereira, and T. Ebrahimi, *The MPEG-4 book*. Prentice Hall Professional, 2002.

[23] T. Ebrahimi and C. Horne, "MPEG-4 natural video coding–An overview," *Signal Processing: Image Communication*, vol. 15, no. 4-5, pp. 365–385, 2000.

[24] G. J. Sullivan and T. Wiegand, "Video compression-from concepts to the H.264/AVC standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, 2005.

[25] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on circuits and systems for video technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[26] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, 2011.

[27] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (HEVC)," *Integrated circuit and systems, algorithms and architectures*, pp. 1–375, 2014.

[28] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1697–1706, 2012.

[29] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1792–1801, 2012.

[30] J.-L. Lin, Y.-W. Chen, Y.-P. Tsai, Y.-W. Huang, and S. Lei, "Motion vector coding techniques for HEVC," in *2011 IEEE 13th International Workshop on Multimedia Signal Processing*. IEEE, 2011, pp. 1–6.

[31] J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas, "Transform coefficient coding in HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1765–1777, 2012.

[32] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Van der Auwera, "HEVC deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1746–1754, 2012.

[33] A. Puri, X. Chen, and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard," *Signal processing: Image communication*, vol. 19, no. 9, pp. 793–849, 2004.

[34] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35–49, 2016.

[35] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of SHVC: Scalable extensions of the high efficiency video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 20–34, 2016.

[36] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging HEVC screen content coding extension," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 50–62, 2016.

[37] X. Xu, S. Liu, T.-D. Chuang, Y.-W. Huang, S.-M. Lei, K. Rapaka, C. Pang, V. Seregin, Y.-K. Wang, and M. Karczewicz, "Intra block copy in HEVC screen content coding extensions," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 409–419, 2016.

[38] L. Zhang, X. Xiu, J. Chen, M. Karczewicz, Y. He, Y. Ye, J. Xu, J. Sole, and W.-S. Kim, "Adaptive color-space transform in HEVC screen content coding," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 446–459, 2016.

[39] W. Zhu, K. Zhang, J. An, H. Huang, Y.-C. Sun, Y.-W. Huang, and S. Lei, "Inter-palette coding in screen content coding," *IEEE Transactions on Broadcasting*, vol. 63, no. 4, pp. 673–679, 2017.

[40] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, "An overview of machine learning," in *Machine learning*. Elsevier, 1983, pp. 3–23.

[41] M. Y. Kiang, "A comparative assessment of classification methods," *Decision support systems*, vol. 35, no. 4, pp. 441–454, 2003.

[42] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[43] T. Fawcett and Tom, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006.

[44] P. Burman, "A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods," *Biometrika*, vol. 76, no. 3, pp. 503–514, 1989.

[45] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.

[46] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *European conference on machine learning*. Springer, 1998, pp. 4–15.

[47] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[48] W.-Y. Loh, "Classification and regression trees," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[49] C. Strobl, A.-L. Boulesteix, and T. Augustin, "Unbiased split selection for classification trees based on the gini index," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 483–501, 2007.

[50] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[51] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[52] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[53] T. K. Tan, R. Weerakkody, M. Mrak, N. Ramzan, V. Baroncini, J. R. Ohm, and G. J. Sullivan, "Video quality evaluation methodology and verification testing of HEVC compression performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 76–90, 2016.

[54] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.

[55] R. Raghavendra and E. M. Belding, "Characterizing high-bandwidth real-time video traffic in residential broadband networks," in *8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*. IEEE, 2010, pp. 597–602.

[56] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," in *ITU - T SG16 Q. 6 VCEG-M33*, Austin, Texas, 2001.

[57] G. Bjontegaard, "Improvements of the BD-PSNR model," in *ITU-T SG16/Q6 VCEG-AI11*, Berlin, Germany, 2008.

[58] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach*. Elsevier, 2011.

[59] S. Ahn, B. Lee, and M. Kim, "A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 422–435, 2015.

[60] F. Bossen, "Common test conditions and software reference configurations," in *12th Meeting of JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, Geneva, 2011.

[61] M. Rerabek and T. Ebrahimi, "New Light Field Image Dataset," in *8th International Workshop on Quality of Multimedia Experience (QoMEX)*, Lisbon, Portugal, 2016.

[62] M. Rerabek, "JPEG Pleno Database: EPFL Light-field data set." [Online]. Available: https://jpeg.org/plenodb/lf/epfl/

[63] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1027–1034.

[64] G. Correa, P. Assuncao, L. Agostini, and L. A. Da Silva Cruz, "Performance and computational complexity assessment of high-efficiency video encoders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1899–1909, 2012.

[65] C. E. Rhee, K. Lee, T. Kim, and H. J. Lee, "A survey of fast mode decision algorithms for inter-prediction and their applications to high efficiency video coding," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 4, pp. 1375–1383, 2012.

[66] L. Shen, Z. Zhang, and Z. Liu, "Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1709–1722, 2014.

[67] H. Zhang and Z. Ma, "Fast intra mode decision for high efficiency video coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 660–668, 2014.

[68] J. Vanne, M. Viitanen, and T. D. Hamalainen, "Efficient mode decision schemes for HEVC inter prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 9, pp. 1579–1593, 2014.

[69] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An effective CU size decision method for HEVC encoders," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 465–470, 2013.

[70] S.-H. Ri, Y. Vatis, and J. Ostermann, "Fast inter-mode decision in an H.264/AVC encoder using mode and Lagrangian cost correlation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 302–306, 2009.

[71] X. Zhang, W. Zhu, C. Wang, and H. Zhang, "A fast Coding Tree Unit depth Prediction for HEVC," in *2015 8th International Congress on Image and Signal Processing (CISP)*. IEEE, oct 2015, pp. 114–118.

[72] B.-g. Kim, "Fast coding unit (CU) determination algorithm for high-efficiency video coding (HEVC) in smart surveillance application," *The Journal of Supercomputing*, vol. 73, no. 3, pp. 1063–1084, 2017.

[73] J. Wu, B. Guo, J. Hou, Y. Yan, and J. Jiang, "A fast CU encoding scheme based on the joint constraint of best and second-best PU modes for HEVC inter coding," in *2015 IEEE International Conference on Imaging Systems and Techniques (IST)*. IEEE, 2015, pp. 1–5.

[74] T. L. Lin, C. C. Chou, Z. Liu, and K. H. Tung, "HEVC early termination methods for optimal CU decision utilizing encoding residual information," *Journal of Real-Time Image Processing*, pp. 1–17, 2016.

[75] K.-h. Tai, M.-y. Hsieh, M.-j. Chen, C.-y. Chen, and C.-H. Yeh, "A Fast HEVC Encoding Method Using Depth Information of Collocated CUs and

RD Cost Characteristics of PU Modes," *IEEE Transactions on Broadcasting*, vol. 63, no. 4, pp. 680 – 692, 2017.

[76] F. Chen, P. Li, Z. Peng, G. Jiang, M. Yu, and F. Shao, "A fast inter coding algorithm for HEVC based on texture and motion quad-tree models," *Signal Processing: Image Communication*, vol. 47, pp. 271–279, 2016.

[77] B. Lee, Hoyoung and Shim, Huik Jae and Park, Younghyeon and Jeon, "Early Skip Mode Decision for HEVC Encoder With Emphasis on Coding Quality," *IEEE Transactions on Broadcasting*, vol. 61, no. 3, pp. 388 – 397, 2015.

[78] X. Huang, Q. Zhang, X. Zhao, W. Zhang, Y. Zhang, and Y. Gan, "Fast inter-prediction mode decision algorithm for HEVC," *Signal, Image and Video Processing*, vol. 11, no. 1, pp. 33–40, 2017.

[79] E. Tamunoiyowuna, J. Zaid, O. Ab, A.-h. A. Rahman, and M. Mun, "Enhanced inter-mode decision algorithm for HEVC/H.265 video coding," *Journal of Real-Time Image Processing*, pp. 1–14, 2015.

[80] J.-H. Lee, K. Goswami, B.-G. Kim, S. Jeong, and J. S. Choi, "Fast encoding algorithm for high-efficiency video coding (HEVC) system based on spatio-temporal correlation," *Journal of Real-Time Image Processing*, vol. 12, no. 2, pp. 407–418, 2016.

[81] L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 207–213, 2013.

[82] L. Shen, Z. Zhang, X. Zhang, P. An, and Z. Liu, "Fast TU size decision algorithm for HEVC encoders using Bayesian theorem detection," *Signal Processing : Image Communication*, pp. 1–8, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.image.2015.01.008

[83] J. Xiong, H. Li, Q. Wu, and F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 559–564, 2014.

[84] J.-c. Chiang, W.-c. Chen, L.-m. Liu, K.-f. Hsu, and W.-n. Lie, "A Fast H.264/ AVC-Based Stereo Video Encoding Algorithm Based on Hierarchical Two-Stage Neural Classification," *IEEE Journal of selected topics in signal processing*, vol. 5, no. 2, pp. 309–320, 2011.

[85] E. Martinez-Enriquez, A. Jimenez-Moreno, M. Angel-Pellon, and F. Diaz-de Maria, "A two-level classification-based approach to inter mode decision in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 11, pp. 1719–1732, 2011.

[86] C.-K. Chiang, W.-H. Pan, C. Hwang, S.-S. Zhuang, and S.-H. Lai, "Fast H.264 encoding based on statistical learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 9, pp. 1304–1315, 2011.

[87] Y. Zhang, S. Kwong, X. Wang, H. Yuan, Z. Pan, and L. Xu, "Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding," *IEEE Transactions on Image Processing*, vol. 24, no. 7, pp. 2225–2238, 2015.

[88] M. Grellert, B. Zatt, S. Bampi, and L. A. d. S. Cruz, "Fast Coding Unit Partition Decision for HEVC Using Support Vector Machines," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1741–1753, 2018.

[89] H.-s. Kim and R.-h. Park, "Fast CU Partitioning Algorithm for HEVC Using Online Learning Based Bayesian Decision Rule," *IEEE transactions on circuits and systems for video technology*, vol. 26, no. 1, pp. 130 – 138, 2016.

[90] L. Zhu, Y. Zhang, Z. Pan, R. Wang, S. Kwong, and Z. Peng, "Binary and Multi-Class Learning Based Low Complexity Optimization for HEVC Encoding," *IEEE Transactions on Broadcasting*, vol. 63, no. 3, pp. 547 – 561, 2017.

[91] X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 4, pp. 1–11, 2013. [Online]. Available: http://jivp.eurasipjournals.com/content/2013/1/4

[92] G. Correa, S. Member, P. Assuncao, and S. Member, "Fast HEVC Encoding Decisions Using Data Mining," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 660 – 673, 2015.

[93] L. Mart, D. Ruiz, G. Ferna, and P. Cuenca, "A unified architecture for fast HEVC intra-prediction coding," *Journal of Real-Time Image Processing*, pp. 1–20, 2017.

[94] B. Du, W. C. Siu, and X. Yang, "Fast CU partition strategy for HEVC intra-frame coding using learning approach via random forests," *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2015*, no. 0, pp. 1085–1090, 2016.

[95] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, mar 2014.

[96] M. Fern and E. Cernadas, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems ?" *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133–3181, 2014.

[97] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC Complexity and Implementation Analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, 2013.

[98] Y. Gao, P. Liu, Y. Wu, K. Jia, and G. Gao, "Fast Coding Unit Encoding Mechanism for Low Complexity Video Coding," *PLOS ONE*, pp. 1–14, 2016.

[99] J. F. D. Oliveira and M. S. Alencar, "Online learning early skip decision method for the HEVC Inter process using the SVM- based Pegasos algorithm," *Electronics Letters*, vol. 52, no. 14, pp. 1227–1229, 2016.

[100] G. Tian and S. Goto, "Content adaptive prediction unit size decision algorithm for HEVC intra coding," in *2012 picture coding symposium*. IEEE, 2012, pp. 405–408.

[101] H. R. Tohidypour, M. T. Pourazad, P. Nasiopoulos, and V. Leung, "A content adaptive complexity reduction scheme for HEVC-based 3D video coding," in *2013 18th International Conference on Digital Signal Processing (DSP)*. IEEE, 2013, pp. 1–5.

[102] J. Leng, L. Sun, T. Ikenaga, and S. Sakaida, "Content based hierarchical fast coding unit decision algorithm for HEVC," in *2011 International Conference on Multimedia and Signal Processing*, vol. 1. IEEE, 2011, pp. 56–59.

[103] C. Blumenberg, D. Palomino, S. Bampi, and B. Zatt, "Adaptive content-based Tile partitioning algorithm for the HEVC standard," in *2013 Picture Coding Symposium (PCS)*. IEEE, 2013, pp. 185–188.

[104] D. Fernández, A. A. Del Barrio, G. Botella, C. García, M. Prieto, and R. Hermida, "Complexity reduction in the HEVC/H265 standard based on smooth region classification," *Digital Signal Processing*, vol. 73, pp. 24–39, 2018.

[105] H. Zhang, Q. Liu, and Z. Ma, "Priority classification based fast intra mode decision for high efficiency video coding," in *2013 Picture Coding Symposium (PCS)*. IEEE, 2013, pp. 285–288.

[106] P. K. Podder, M. Paul, and M. Murshed, "A novel motion classification based intermode selection strategy for HEVC performance improvement," *Neurocomputing*, vol. 173, pp. 1211–1220, 2016.

[107] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 31–42.

[108] M. Levoy, "Light fields and computational imaging," *Computer*, no. 8, pp. 46–55, 2006.

[109] I. Viola, M. Řeřábek, and T. Ebrahimi, "Comparison and evaluation of light field image coding approaches," *IEEE Journal of selected topics in signal processing*, vol. 11, no. 7, pp. 1092–1106, 2017.

[110] Y. Li, M. Sjostrom, R. Olsson, and U. Jennehag, "Coding of focused plenoptic contents by displacement intra prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 7, pp. 1308—-1319, 2016.

[111] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo-Sequence-Based 2-D Hierarchical Coding Structure for Light-Field Image Compression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1107–1119, 2017.

[112] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement," in *IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2016*, 2016.

[113] W. Ahmad, R. Olsson, and M. Sjöström, "Interpreting plenoptic images as multi-view sequences for improved compression," in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4557–4561.

[114] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2016*, 2016, pp. 3–6.

[115] J. M. Santos, P. A. A. Assuncao, A. Luis, S. Cruz, L. M. N. Tavora, R. Fonseca-pinto, and S. M. M. Faria, "Lossless Coding of Light Field Images based on Minimum-Rate Predictors," *Journal of Visual Communication and Image Representation*, 2018. [Online]. Available: https://doi.org/10.1016/j.jvcir.2018.03.003

[116] C. Perra, "Light field image compression based on preprocessing and high efficiency coding," in *Telecommunications Forum (TELFOR), 2016 24th*. IEEE, 2016, pp. 1–4.

[117] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG Pleno: Toward an efficient representation of visual reality," *IEEE Multimedia*, vol. 23, no. 4, pp. 14–20, 2016.

[118] WG1, "JPEG Pleno Call for Proposals on Light Field Coding," in *74th Meeting, Geneva, Switzerland, January 15-20*, vol. 1, 2017, pp. 1–37.

[119] T. Shi and J. Wang, "Exploiting data mining for fast inter prediction mode decision in HEVC," *Mobile Networks and Applications*, pp. 1–9, 2018.

[120] K. Goswami, B.-G. Kim, D. Jun, S.-H. Jung, and J. S. Choi, "Early coding unit–splitting termination algorithm for high efficiency video coding (HEVC)," *Etri Journal*, vol. 36, no. 3, pp. 407–417, 2014.

[121] Y.-J. Ahn, T.-J. Hwang, D.-G. Sim, and W.-J. Han, "Implementation of fast HEVC encoder based on simd and data-level parallelism," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 16, 2014.

[122] P. ITU-T Recommendation, "Subjective video quality assessment methods for multimedia applications," *International telecommunication union*, 1999.

[123] D. Ruiz, G. Fernández-Escribano, J. L. Martínez, and P. Cuenca, "Fast intra mode decision algorithm based on texture orientation detection in HEVC," *Signal Processing: Image Communication*, vol. 44, pp. 12–28, 2016.

[124] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[125] X. Xu, S. Xu, L. Jin, and E. Song, "Characteristic analysis of otsu threshold and its applications," *Pattern recognition letters*, vol. 32, no. 7, pp. 956–961, 2011.

[126] "High Efficiency Video Coding (HEVC) — JCT-VC." [Online]. Available: https://hevc.hhi.fraunhofer.de/

[127] L. Breiman, "Random Forests," *Journal of Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[128] L. Breiman, "Consistency for a simple model of random forests. Statistical Department," *University of California at Berkeley. Technical Report,(670)*, 2004.

[129] G. Biau, "Analysis of a random forests model," *Journal of Machine Learning Research*, vol. 13, no. Apr, pp. 1063–1095, 2012.

[130] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *2009 ieee 12th international conference on computer vision workshops, iccv workshops*. IEEE, 2009, pp. 1393–1400.

[131] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2001.

[132] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[133] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *Icml*, vol. 1, 2001, pp. 74–81.

[134] S. R. Singh, H. A. Murthy, T. A. Gonsalves *et al.*, "Feature selection for text classification based on gini coefficient of inequality." *Fsdm*, vol. 10, pp. 76–85, 2010.

[135] M. A. Hall and G. Holmes, "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 6, pp. 1437–1447, 2003.

[136] L. Yu and H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy," *Journal of Machine Learning Research*, vol. 5, pp. 1205–1224, 2004.

[137] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper–filter feature selection algorithm using a memetic framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 1, pp. 70–76, 2007.

[138] L. Zhu, Y. Zhang, N. Li, G. Jiang, and S. Kwong, "Machine learning based fast H . 264 / AVC to HEVC transcoding exploiting block partition similarity," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 824–837, 2016.

[139] T. Mallikarachchi, D. S. Talagala, H. K. Arachchi, and A. Fernando, "Content-Adaptive Feature-Based CU Size Prediction for Fast Low-Delay Video Encoding in HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 693–705, 2016.

[140] B. N. Lee, "librf: C++ random forests library." [Online]. Available: http://mtv.ece.ucsb.edu/benlee/librf.html

[141] X. Sun, X. Chen, Y. Xu, Y. Xiao, Y. Wang, and D. Yu, "Fast CU size and prediction mode decision algorithm for HEVC based on direction variance," *Journal of Real-Time Image Processing*, Mar 2017. [Online]. Available: https://doi.org/10.1007/s11554-017-0682-7

[142] A. Gershun, "The light field," *Journal of Mathematics and Physics*, vol. 18, no. 1-4, pp. 51–151, 1939.

[143] G. Wu, B. Masia, A. Jarabo, Y. Zhang, and L. Wang, "Light Field Image Processing : An Overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1–29, 2017.

[144] P. A. A. Assuncao, "Evolution from 3D video to light-field coding and transmission over future media networks," in *18th Mediterranean Electrotechnical Conference MELECON*, no. April, 2016, pp. 18–20.

[145] D. Lelescu and F. Bossen, "Representation and coding of light field data," *Graphical Models*, vol. 66, no. 4, pp. 203–225, 2004.

[146] M. Martínez-Corral and B. Javidi, "Fundamentals of 3D imaging and displays: a tutorial on integral imaging, light-field, and plenoptic systems," *Advances in Optics and Photonics*, vol. 10, no. 3, pp. 512–566, 2018.

[147] Z.-y. Chen and P.-c. Chang, "Rough mode cost based fast intra coding for high-efficiency video coding," *Journal of Visual Communication and Image Representation*, vol. 43, pp. 77–88, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.jvcir.2016.12.007

[148] J. Tariq, S. Kwong, and H. Yuan, "Spatial/temporal motion consistency based MERGE mode early decision for HEVC," *Journal of Visual Communication and Image Representation*, vol. 44, pp. 198–213, 2017.

[149] N. Dessì and B. Pes, "An evolutionary method for combining different feature selection criteria in microarray data classification," *Journal of Artificial Evolution and applications*, vol. 2009, 2009.

[150] Y. Li, D. F. Hsu, and S. M. Chung, "Combining multiple feature selection methods for text categorization by using rank-score characteristics," in *IEEE International Conference on Tools with Artificial Intelligence, ICTAI*, 2009, pp. 508–517.

[151] M. Barbareschi, "Implementing hardware decision tree prediction: a scalable approach," in *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, 2016, pp. 87–92.

[152] R. J. Struharik and L. A. Novak, "Hardware implementation of decision tree ensembles," *Journal of Circuits, Systems and Computers*, vol. 22, no. 05, p. 1350032, 2013.

[153] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal processing magazine*, vol. 20, no. 2, pp. 18–29, 2003.

[154] L. P. Van, J. De Cock, G. Van Wallendael, S. Van Leuven, R. Rodriguez-Sánchez, J. L. Martínez, P. Lambert, and R. Van de Walle, "Fast transrating for high efficiency video coding based on machine learning," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 1573–1577.