Intra Frame Luma Prediction using
Neural Networks in
HEVC

by

DILIP PRASANNA KUMAR

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in Electrical Engineering

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2013

ACKNOWLEDGEMENTS

I gratefully thank Professor K.R. Rao for his supervision, encouragement and guidance in conducting this research. I also wish to thank Dr. Russel and Dr. Dillon for serving as members of my graduate committee.

I would like to thank my collegues at the MPL lab for the discussions I have had with them. I would like to thank Rohit Rawat, Kanishka Tyagi, Soumitro Auddy, Nayana Parashar, Yeshaswini Gowda, Asha B.K Rao and Karthik Mahendra for providing valuable suggetions and feedback during my research.

Finally, I would like to thank my family and friends for supporting me in this undertaking.

April 22, 2013

ABSTRACT

Intra Frame Luma Prediction using
Neural Networks in
HEVC

Dilip Prasanna Kumar, M.S
The University of Texas at Arlington, 2013

Supervising Professor: K. R. Rao

High Efficiency Video Coding, the latest video coding standard proposed by the JVT-VC and three profiles- HEVC main, main 10 and main intraframe were adopted in January 2013, provides significant amount of compression compared to older standards, while retaining similar visual quality. This is achieved at the cost of a computationally expensive encoding method.

Intra frame coding contributes to a large portion of the computational complexity. In this research, a way to speed up the intra frame prediction mode decision using Artificial Neural Networks is proposed. The search for the correct prediction mode is simplified by using neural networks to analyze and reduce the number of modes that must be searched to arrive at the mode decision.

By employing this scheme, a speed up of upto 20% has been observed without significant loss of PSNR or increase in bitrate.

TABLE OF CONTENTS

Appendix

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

LIST OF ACRONYMS

AVC Advanced Video Coding

CABAC Context Adaptive Binary Arithmetic Coding

CB Coding Block

CIF Common Intermediate Format

CTB Coding Tree Block

CTU Coding Tree Unit

CU Coding Unit

DCT Discrete Cosine Transform

DST Discrete Sine Transform

DPCM Differential pulse code modulation

FANN Free artificial neural networks library

HDTV High Definition Television

HEVC High Efficiency Video Coding

HVS human visual system

ITU-T International Telecommunication Union - Telecommunication Standardization Sector

JCT-VC Joint Collaborative Team on Video Coding

MLP Multilayer perceptron

MOS mean opinion score

MPEG Moving Picture Experts Group

MSE mean squared error

PB Prediction Block

PSNR peak signal to noise ratio

PU Prediction Unit

QCIF Quarter CIF

QP Quantization Parameter

RDO rate distortion optimization

RDOQ rate distortion optimized quantization

SAD sum of absolute differences

SATD sum of absolute transform differences

SSIM structural similarity index metric

TB Transform Block

TU Transform Unit

URQ Uniform Reconstruction Quantization

VCEG Video Coding Experts Group

CHAPTER 1

INTRODUCTION

1.1 Video Compression Basics

The properties of an image are height, width, color and brightness of a pixel. Videos add a time dimension to images by stacking several images (also called frames). By displaying the sequence of images quickly, videos create the illusion of a moving image.

A large portion of video compression is related to image compression. Image compression exploits spatial redundancies in an image to compress the image. Video compression algorithm performs the same task to compress individual images and also exploits temporal redundancies (the redundancies between frames) to efficiently compress several frames. [2].

Video frames that are compressed by exploiting temporal redundancies are called inter-frames or p-frames (used interchangeably). To compress a p-frame, the frame is compared to a reference frame and only the difference between the frames (quantized transform coefficients of residual images and motion vectors [3]) are encoded. The frames that are used as a reference for the p-frame can only be compressed using their own spatial redundancies. These frames are called i-frames or intra frames (used interchangeably) . A third type of frames, called b-frames use bidirectional prediction and use the both p-frames and i-frames for exploiting temporal redundancies [4].

1.2 Need for Video Compression

The table 1.1 shows the raw bit-rate for several popular video formats.

Table 1.1: Raw bit-rates for common video formats [3]

Format	**Resolution**	**Raw Bitrate**
CIF	352x288 30Hz	36.5 Mbps
QCIF	176x144 30Hz	9.13 Mbps
HDTV	1280x720 60Hz	663.6 Mbps
	1920x1080 30Hz	1.3905 Gbps

The table 1.1 shows that raw (uncompressed) video sequences require enormous amount of bandwidth and memory. Several video compression schemes have been developed to address how videos should be compressed. The High Efficiency Video Coding (HEVC) video compression standard, for example, achieves a bitrate of around 11.2Mbps for the Basketballdrive test sequence at 1920x1080 60Hz. This is a reduction of 98.3%. Such a high amount of video compression makes several video applications like broadcast HDTV, video conferencing, internet streaming services and many others feasible.

1.3 Video compressing standards

To enable encoders and decoders developed by different vendors to function together, video compression algorithms have been standardized by bodies such as International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), Moving Picture Experts Group (MPEG) and recently the Joint Collaborative Team on Video Coding (JCT-VC). Several video compression standards adopted today, such as MPEG-2 [5], Advanced Video Coding (AVC) [6] and H.264

[4] are the results of the efforts put in by these bodies. The figure 1.1 shows the evolution of video compression standards over the years.

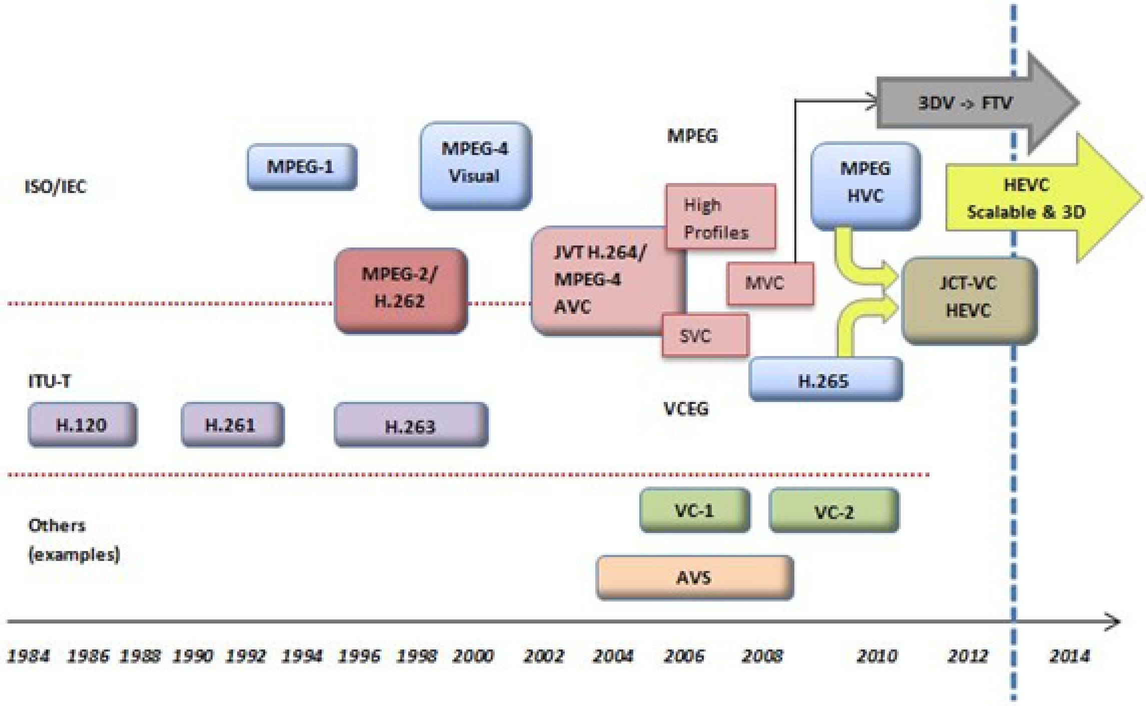


Figure 1.1: Evolution of video compression standards [7]

1.4 Thesis Outline

Chapter 2 provides a brief introduction to the HEVC compression standard along with a description of its encoder block diagram.

In chapter 3, intra prediction is considered in more detail. The complexity of intra frame prediction mode decision is presented along with some previous attempts to reduce this. The reference software HM [8] is introduced and the implementation of intra frame mode decision for luma samples is also presented.

In chapter 4, a new approach to intra frame mode decision using neural networks is presented. In chapters 5 and 6 the proposed approach is explored in detail.

Chapter 7 presents the results obtained in terms of time gain, loss of PSNR and increase in bitrate when compared to the reference software. Chapter **??** presents conclusions and discusses possible further work.

CHAPTER 2

HIGH EFFICIENCY VIDEO CODING, HEVC

HEVC is the latest video compression standard to be adopted as an international standard. It was developed over a period of 6 years by the JCT-VC from 2007 to 2013 [1]. The main goal for HEVC was to achieve 50% lower bitrate than H.264 while retaining the same visual quality. Figure 2.1 shows the block diagram for the HEVC encoder.

2.0.1 Encoder description

Figure 2.1 shows the block diagram of a generic HEVC encoder. The input to this encoder is a sequence of frames from a video source. Each frame is first segmented to generate the quad tree structure. Then, each segment is passed one by one to the encoder.

The encoder decides if the block should be compressed as an i-frame (intra frame, exploiting spatial redundancies) or as a p-frame (inter frame, exploiting temporal redundancies). Based on this decision, the encoder generates an image block that is subtracted from the original image segment. The residual signal is transformed, scaled, quantized and subjected to entropy encoding. Meanwhile, inverse quantization and transforms are applied to the quantized transform coefficients. These are used by the intra or inter prediction blocks.

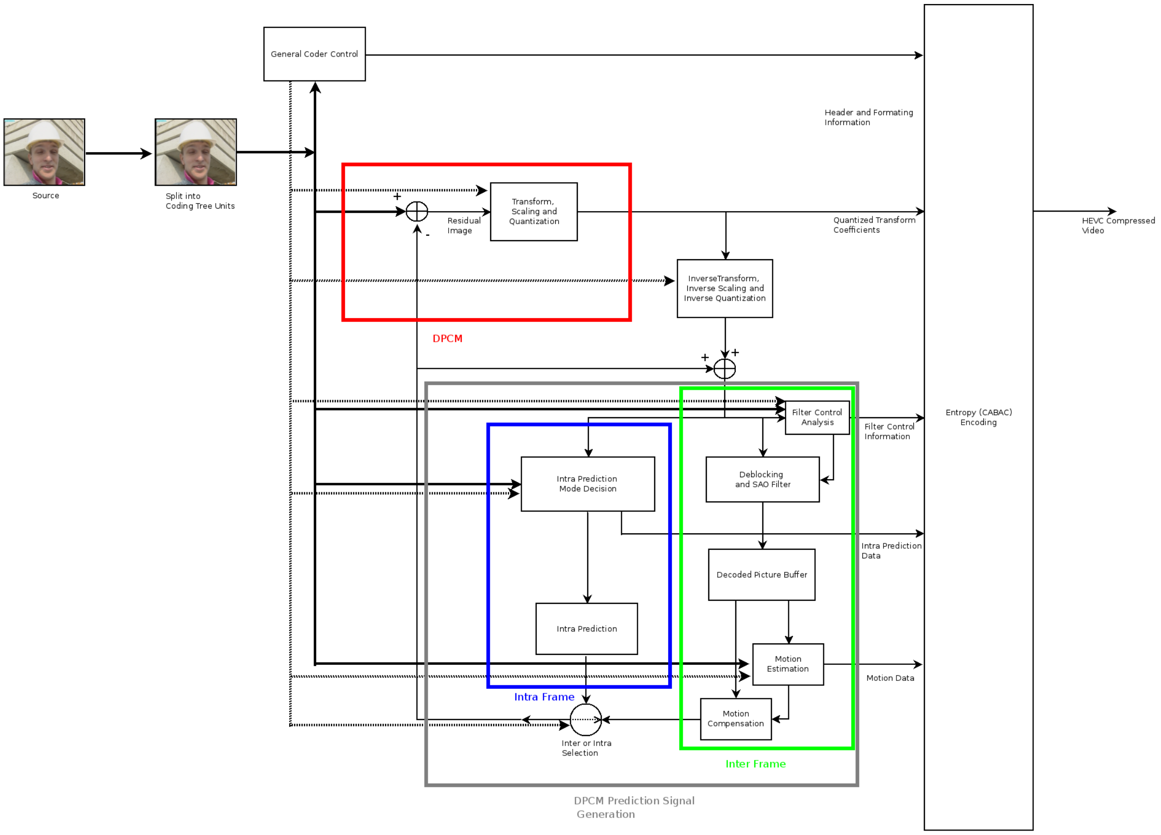


Figure 2.1: HEVC encoder block diagram

2.0.2 Quad Tree Structure

The quad tree structure is introduced in HEVC. The macroblocks used in H.264 [4] are replaced by Coding Tree Units. Figure 2.2 shows how an image block is split into the quad tree structure. The Coding Tree Unit (CTU) consists of a Luma Coding Tree Block (CTB) and the corresponding chroma CTBs and syntax elements. The CTU specifies the positions and the sizes of the luma Coding Block (CB) and chroma CB. One luma CB and generally two chroma CBs together with syntax form a Coding Unit (CU).

A CTB can have one CU or be split into several CUs. The decision to code an area of image as intra or inter is taken at the CU level. A CU is the root for both Prediction Unit (PU) and Transform Unit (TU). A Prediction Block (PB) can be the

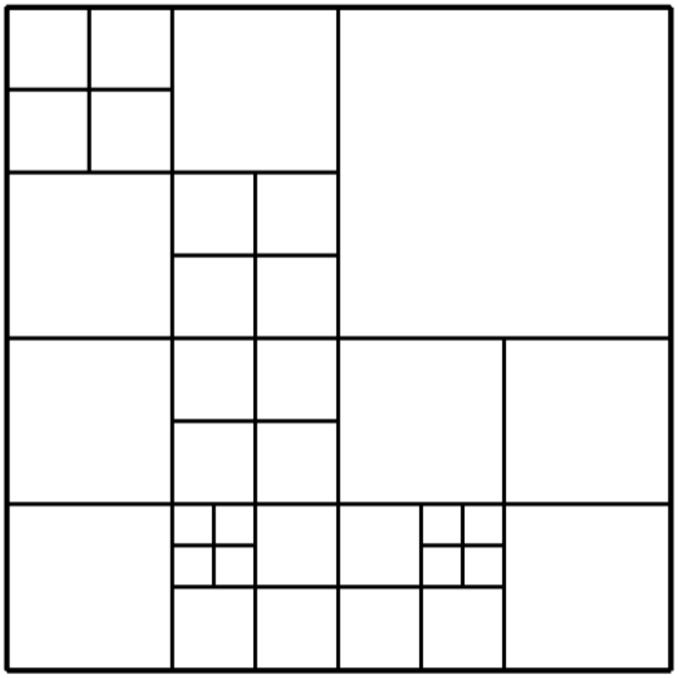


Figure 2.2: HEVC Quad tree structure [1]

size of a CB or be split further into smaller luma and chroma PBs. The supported sizes are 64x64, 32x32, 16x16, 8x8 and 4x4. For inter prediction modes, non square modes are allowed as shown in Figure 2.3. An inter frame PB can not have a size of 4x4.

Similarly, starting at the level of a CU, a CB can have one Transform Block (TB) of the same size as the CB or be split into smaller TBs [1], [9], [10].

2.0.3 DPCM, Quantization and Transform

Differential pulse code modulation (DPCM) is a well known technique used in several communication systems. In video compression technologies, DPCM is extended to two dimensional images [2]. In HEVC, an image block (the PB) is subtracted from a generated prediction block or a motion compensated reference image segment to obtain a difference image block (also called an error image block or a

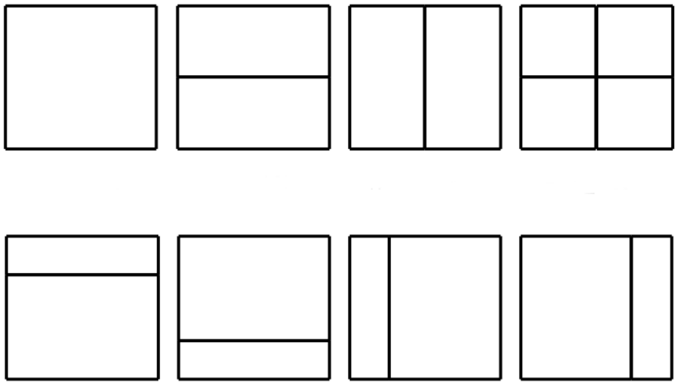


Figure 2.3: Inter frame prediction block modes for luma samples in HEVC [1]

residual image block). The principle behind this operation is that if the histogram of the error signal is narrow, fewer bits are required to represent it [2].

The residual block is partitioned into several TBs and then transformed. The possible sizes for the TB are 4x4, 8x8, 16x16 and 32x32. A single 32x32 matrix is specified for the transform. A 32x32 TB is transformed by simply performing matrix multiplication with this block. For TBs of sizes 16x16, 8x8 or 4x4, corresponding transforms are applied. The matrix is based on the Discrete Cosine Transform (DCT). A special 4x4 transform based on the Discrete Sine Transform (DST) is also available for 4x4 intra coding [1].

The transform coefficients are then subjected to quantization. HEVC uses the same Uniform Reconstruction Quantization (URQ) scheme used in H.264. A Quantization Parameter (QP) is used to control the step size of the quantizer, and has a range from 0 to 51. A QP increase of 6 doubles the step size. [1].

The quantized transform coefficients are then further compressed using entropy encoding. HEVC specifies Context Adaptive Binary Arithmetic Coding (CABAC) as the entropy coding scheme to be used. The number of contexts used in HEVC is less than those in h.264/AVC but are also more efficient [1].

2.0.4 Intra Prediction

As shown in figure 2.1, the residual block is generated by subtracting the original source from a predicted block. The predicted block is generated using either intra or inter frame prediction.

For intra prediction, HEVC specifies 35 different prediction modes for luma samples. For each PB, any one of the 35 prediction modes can be used to generate a prediction. Both the encoder and decoder always use the row of pixels to the top and the column of pixels to the left of the current prediction block to generate the prediction. The prediction mode specifies how the top row or the left column should be used to generate a prediction. In HEVC, there are 33 angular modes,a DC mode and an interpolation mode. Figure 2.4 shows the angular modes and the corresponding mode numbers in HEVC.

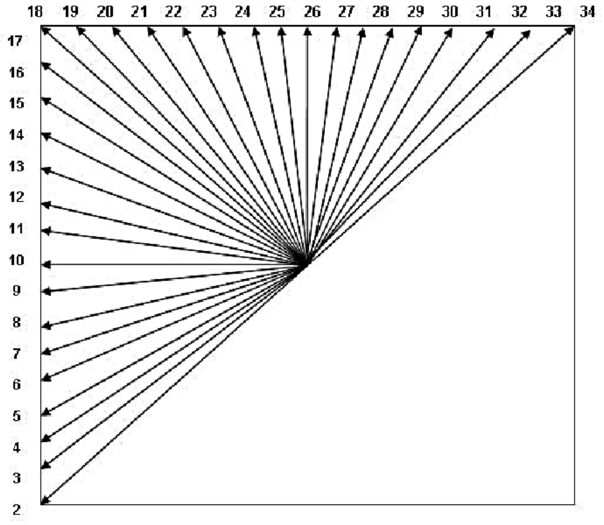


Figure 2.4: Intra prediction modes for luma samples in HEVC [1]

Figure 2.5 shows an example of how the top row or left column reference is copied to create the prediction block for an angular mode.

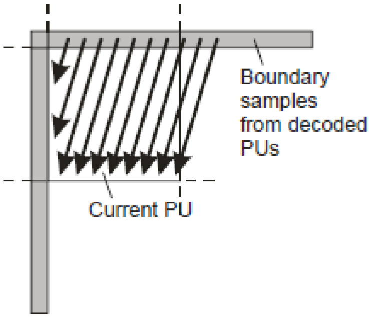


Figure 2.5: Example of an intra prediction mode in HEVC [1]

Table 2.1 shows the prediction modes for the chroma samples. These are similar to the ones used in H.264/AVC standard [4][1].

Table 2.1: Intra prediction modes for chroma samples [1]

Intra chroma prediction mode	**Intra prediction direction**				
	0	26	10	1	$X(0 \leq X < 35)$
0	34	0	0	0	0
1	26	34	26	26	26
2	10	10	34	10	10
3	1	1	1	34	1
4	0	26	10	1	X

2.1 Summary

This chapter has introduced the HEVC video coding standard, the encoder block description and intra prediction used in HEVC. In the next chapter, intra prediction is discussed in detail.

CHAPTER 3

INTRA PREDICTION MODE DECISION IN HEVC

3.1 Mode decision for intra prediction

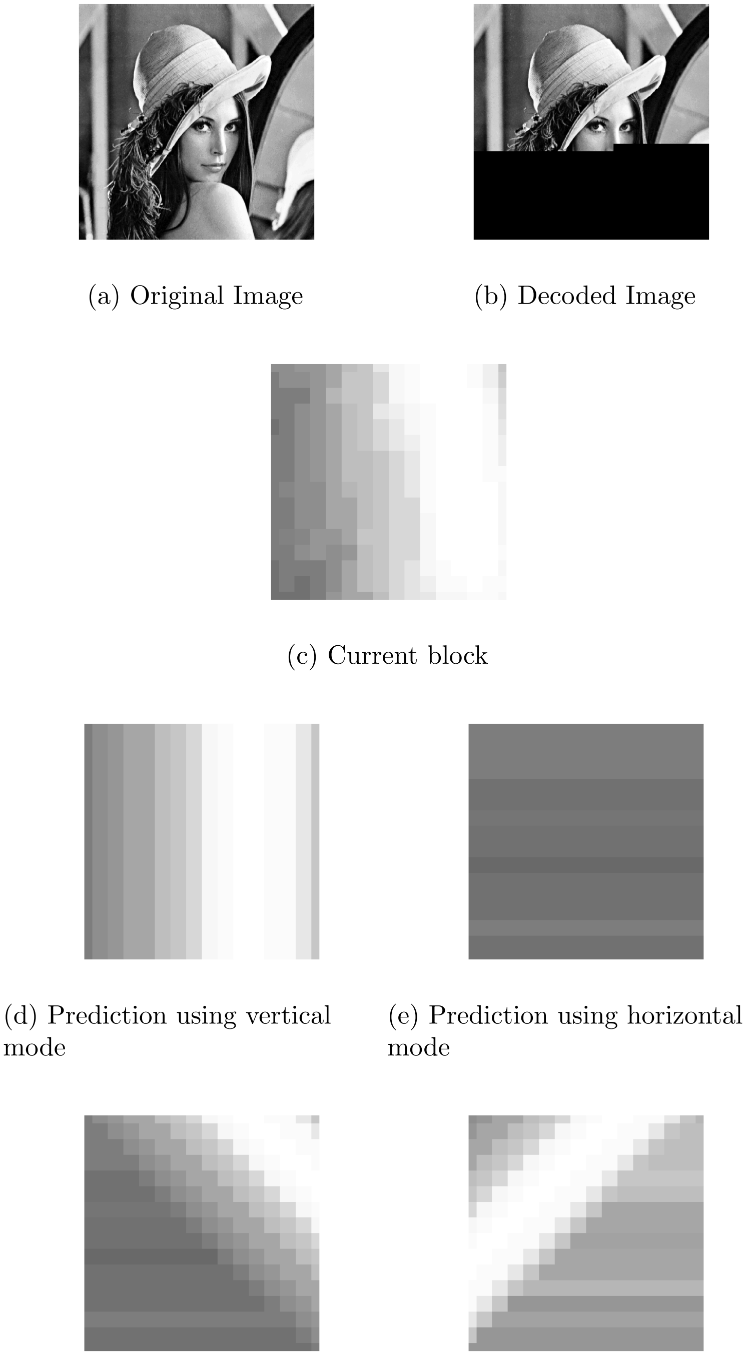
Figure 3.1 shows an example of a fictional intra prediction scheme where four intra prediction modes are available. To generate a prediction for the current block, the decoder has the pixels to the top and the left pixels which can been seen in Figure 3.1b. The current block is shown in Figure 3.1c and the predictions generated for this block are shown in Figures 3.1d - 3.1g.

In this example, the vertical prediction mode in Figure 3.1d provides the closest match. Thus, the encoder can select the vertical mode for intra prediction, and inform the decoder that the vertical mode of prediction must be used while decoding the frame. Mode decision is this process of determining which prediction scheme has to be used for the current block.

HEVC offers 35 different possible prediction modes (figure 2.4). Further, the encoder must also determine the size of the PB to be used. For a CB, the HEVC encoder must check for all possible prediction modes at all allowed PB sizes and select the best combinations of PB sizes and modes to encode a certain CU. This decision is made based on rate distortion optimization (RDO).

3.2 Rate Distortion Optimization

The encoder's decision on the mode of operation to be used (prediction mode, size of PB) affects the signal dependent rate-distortion characteristics. The encoder's



(a) Original Image

(b) Decoded Image

(c) Current block

(d) Prediction using vertical mode

(e) Prediction using horizontal mode

(f) Prediction using diagonal right mode

(g) Prediction using diagonal left mode

Figure 3.1: Example of prediction modes for intra prediction. Here, figure 3.1c is the current block and figure 3.1d produces the lowest SAD

task is then to minimize the distortion D, subject to a constraint R_c, on the number of bits used R.

$$min\{D\}\ subject\ R \leq R_c \tag{3.1}$$

The encoder can also be optimized to minimize the rate R, subject to a constraint D_c on the distortion D as

$$min\{R\}\ subject\ D \leq D_c \tag{3.2}$$

Equations 3.1 and 3.2 represent a constrained problem. This is converted to an unconstrained problem using a Lagrangian multiplier as

$$min\{J\},\ where\ J\ =\ D\ +\ \lambda R \tag{3.3}$$

where λ is the Lagrangian multiplier, R is the rate and D is the distortion observed for a particular combination of modes. The encoder's task is to select a combination of modes such that the cost J is minimized [11]. This process is known as rate distortion optimization (RDO).

In the context of intra prediction in HEVC, the size of the PB and the prediction mode used for a PB must be selected such that the RDO cost J in (3.3) is minimized.

The rate R can be measured objectively for the purposes of rate-distortion optimization. The distortion D is a subjective quality and depends upon the human visual system (HVS). Some subjective measures of distortion are structural similarity index metric (SSIM) and mean opinion score (MOS) while peak signal to noise ratio (PSNR), mean squared error (MSE), sum of absolute differences (SAD) and sum of absolute transform differences (SATD) are bjective measures of distortion.

The process of intra prediction mode decision in HEVC involves the encoder measuring the values of distortion D and rate R for each of the 35 available modes (Figure 2.4) and then selecting the mode that provides the lowest rate-distortion cost J.

3.3 Quad tree structure and intra prediction mode decision complexity in HEVC

In the HEVC encoder, the PB size is also determined using rate-distortion optimization. While larger PBs are more efficient, smaller PBs are required in regions of high detail and texture. The encoder should measure the values of distortion D and rate R for each of the 35 modes at every level of the PU subtree. Following this, the quantization and TB sizes are also determined through R-D optimization.

Table 3.1: Number of possible intra prediction combinations in a 64x64 CB

Size of PB	**Number of PBs in a 64x64 CU**	**Number of Modes to be Tested in each PB**	**Total number of modes to be tested at this level**
32x32	4	35	140
16x16	16	35	560
8x8	64	35	2240
4x4	256	35	8960
		Total	11900

Table 3.1 shows the number of possible prediction modes that exist in a 64x64 CB. To find the optimum combination, the encoder must calculate the values of distortion and rate $11,900$ times. Finding the value of D and R includes performing the complete prediction, transform, scaling and quantization, and all the inverse operations $11,900$ times along with the SATD which is computed as the measure of distortion D.

The quad tree structure imposes enormous computational requirements for HEVC.

3.4 Implementation in HM 8.0 reference encoder

The reference software for HEVC developed by the JCT-VC is called HM. It includes a reference encoder and a decoder that meets the HEVC specifications.

In version 4.0 in the HM reference encoder, a tree pruning algorithm was introduced that reduces the number of subtrees that are explored by the encoder [12]. In HM 8.0, intra prediction is performed in three stages [13], [14]:

Table 3.2: HM 8.0 intra prediction scheme

Step 1	A rough mode decision is performed based on prediction residual SATD and estimated mode bits.
Step 2	The selected modes from step 1 is chosen for RDOQ to obtain the best prediction mode.
Step 3	The best TU partitions are determined for best mode from the previous RDOQ process.

3.5 Decision cost

Despite the improvements made to the HM encoder, it is still very slow because of its extensive search of all modes. In the “all intra” configuration in HM 8.0, rate distortion optimized quantization (RDOQ) was observed to take up 24.4% of the total encoding time followed by the intra mode decision process at 11.8% and 11.0% for memcopy/memset. [15].

From table 3.1, it can be seen that if the number of modes that have to be checked at each PB level can be reduced by 1, the total number of modes that have to be evaluated reduces by $(4 + 16 + 64 + 256 = 340)$. A new parameter called "decision cost" is introduced as a way to measure the number of modes that have to be evaluated. Decision cost is defined as the average number of modes that must be checked by performing intra prediction and calculating distortion and rate in a PB.

Decision cost is related to how computationally intensive the intra prediction mode decision algorithm is. The higher the decision cost, the more number of intra prediction modes are being checked by the encoder and hence, higher cost in terms of computation time.

Let M be the set of possible intra prediction modes for the PB. HEVC specifies that

$$M = \{0, 1, 2, \dots 34\} \tag{3.4}$$

where 0 is the DC mode, 1 is the planar mode and 2 to 34 are angular modes (figure 2.4).

Let S be the set of modes that have to be evaluated and $\mathcal{L}[S]$ be the length of set S. Then decision cost C is given by

$$C = E[\ \mathcal{L}[S]\ ] \tag{3.5}$$

In HM 8.0 reference encoder, all 35 modes are evaluated for a rough mode decision. For a PB of size 16x16 and larger, the best 3 modes from the rough mode decision are selected for RDOQ. Totally, 35+3 = 38 modes are evaluated so $\mathcal{L}[S] = 38$ in all cases where the PB size is 16x16 or larger. The decision cost $C = E[\mathcal{L}[S]] = 38$ here for all PBs of size equal to or larger than 16x16. Similarly for PB sizes of 4x4 and 8x8, the best 8 modes from the rough mode decision are selected for RDOQ and decision cost $C = E[\mathcal{L}[S]] = 43$.

3.6 Summary

This chapter introduced the concept of rate-distortion optimization. The HM-8.0 reference encoder was introduced and the intra prediction scheme implementation in HM-8.0 was discussed. The idea of decision cost was introduced. The next chapter presents the proposed solution to reduce the decision cost for intra prediction.

CHAPTER 4

PROPOSED SOLUTION TO INTRA FRAME MODE DECISION COMPLEXITY

In section 3.3, the computational complexity of HEVC intra prediction in HM 8.0 was covered and section 3.5 introduced the idea of decision cost.

To reduce the computational complexity of intra prediction, an artificial neural network based approach to reduce the decision cost is proposed. The idea is to train an artificial neural network to identify the best prediction modes to be used for any PB. The results suggested by the neural network are analyzed and used to generate a set of most likely possible modes. The best mode is then selected from this set for intra prediction using RDOQ as explained in steps 2 and 3 in table 3.2.

4.1 Artificial Neural Networks

An artificial neural network [16] is introduced into the encoder to quickly identify and classify a PB as belonging to one of 33 possible angular modes (Figure 2.4). Artificial neural networks are used extensively in several pattern recognition techniques because they are faster than conventional computing and they can be trained to perform a wide variety of tasks [16]. An introduction to artificial neural networks is given in Appendix A

In this thesis, a neural network is trained to guess which *angular* mode is most likely going to provide the lowest RDO cost. After the neural network has been trained sufficiently, it is integrated with the HM 8.0 reference encoder [8]. The reference encoder is made to bypass the rough mode decision (step 1 in table 3.2) and feed all PBs to the neural network instead. The neural network generates a guess on which of the angular modes is best suited for the input PB based upon its previous training.

Ideally, the perfect neural network will always guess the best angular mode for intra prediction in the RDO sense. Let this mode be N where $N \in M$ and M is the set described in (3.4). After including the planar and DC modes, the set of modes that need to be evaluated becomes $S = \{N, 0, 1\}$. Then $\mathcal{L}[S] = 3$. The decision cost $C = 3$ for all cases because only three prediction modes need to be evaluated. However, as will be explained in section 5.3, the neural network approach is not perfect and further processing needs to be done.

4.2 Obtaining prediction modes from NN

The accuracy of the neural network is not high as will be explained in section 5.3.

Let Q be the best prediction mode in RDO sense for the current PB. The probability that prediction mode Q is contained in the set of modes to be evaluated S is denoted as $P[Q \in S]$. To increase the probability that the best mode in RDO sense is included in set S, the set S can be expanded to include $S = \{N, N{\pm}1, N{\pm}2 \ldots, 0, 1\}$ where $N \pm 1, N \pm 2 \ldots \in M$ (Values outside the range [2, 34] are not considered). As the number of modes added to S increases, both $P[Q \in S]$ and $\mathcal{L}[S]$ increases as will be explained in section 5.3.

In this scheme, $\mathcal{L}[S]$ depends on the mode N suggested by the neural network. This is because $N \pm 1, N \pm 2 \ldots$ can not be applied in cases where the resulting modes will be outside the range [2, 34]. For example, if $N = 2$, S can only be expanded as $S = \{N, 0, 1, N+1, N+2 \ldots\}$ and if $N = 34$, $S = \{N, 0, 1, N-1, N-2 \ldots\}$.

Let S_i be the set of modes to be evaluated when the neural network outputs mode N_i where $i = 2, 3, 4, \ldots 34$. If $P[N_i]$ is the probability of the neural network outputting N_i, then the decision cost C is given by

$$C = \sum_{i=2}^{i=34} P[N_i].\mathcal{L}[S_i] \tag{4.1}$$

$P[N_i]$ depends upon the training of the neural network, the frequency of occurrence of each mode in videos and the accuracy of the neural network. Chapter 6 covers the procedure and explores ways to further optimize the process using Bayesian probabilities.

4.3 Fast search optimization

The set of modes to be evaluated, S can be further reduced. This scheme uses a simple fast search algorithm that can reduce the size $\mathcal{L}[S]$ at a negligible loss to the accuracy ($P[Q \in S]$). This method provides coding time gain by reducing the number of modes that must be tested using RDOQ.

4.4 Summary

This chapter introduces the proposed solution. All three components of this proposed solution- the neural network, finding the set S from the neural network and the fast search algorithm provide a large number of parameters that can be adjusted to change the system performance. The following chapters will present some of the parameters, how they affect the system and how they are chosen.

CHAPTER 5

SHORTLISTING POSSIBLE INTRA MODES WITH NEURAL NETWORKS

In the first step, the image block is fed to a neural network that has been trained to guess which of the 33 intra prediction schemes are best suited for that block. The neural network was trained using the FANN library [17]. The training algorithm used is RPROP [18] and a simple 3 layer multi layer perceptron architecture is implemented.

5.1 Training the neural network

The HM 8.0 encoder [8] was modified to log pixel values of PBs and the corresponding intra mode chosen. PB blocks of sizes 4x4, 8x8, 16x16 and 32x32 were logged. The logged data was then formatted to be compatible with the FANN library.

The FANN library is an open source neural network library, which implements multilayer artificial neural networks in C. It provides several functions for creating and training a neural network. Among the training methods provided by this library, the RPROP [18] method was found to be the most effective for this problem. The neural network was generated and trained using the FANN library and the training data logged from the HM 8.0 encoder. The neural network was trained to identify the mode that produces the lowest RDO cost.

5.2 HM encoder simulation for testing the accuracy of proposed scheme

To test the neural network, a simulation tool was developed. Figure 5.1 shows the block diagram for the simulation tool.

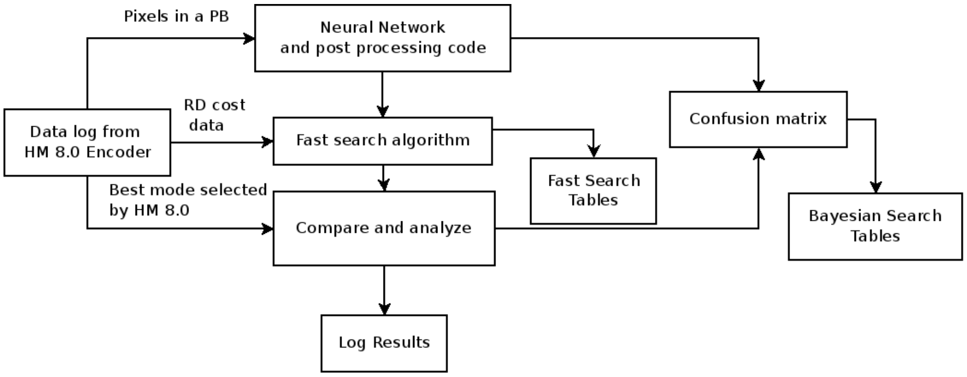


Figure 5.1: HM encoder simulation tool

The tool uses data previously logged from an unmodified HM 8.0 encoder [8]. From these logs, pixel values for several PBs and their corresponding best modes are extracted. The pixels are fed to the neural network being tested. The neural network generates the best guess mode which is used to generate a set of modes to be tested as explained in section 4.2. If a fast search algorithm is used, then this set is further modified by the fast search algorithm. Finally, the set S is compared to the best mode selected by the unmodified HM encoder to calculate the accuracy of the neural network.

The tool also helps compute the Bayesian search tables (section 6.3) and the fast search tables (section 6.4).

5.3 Accuracy of Neural Networks and system performance

With the goal of speeding up the encoding process, the neural network process is designed to be fast and simple. The neural network's input is not subjected to any feature extraction. This is because of two reasons:

1. The suitable feature in this case, edge detection, is very computationally intense.
2. The neural network showed the ability to classify images with sufficient accuracy.

Table 5.1: 7-tap and 8-tap interpolating filters [1]

Filter	-3	-2	-1	0	1	2	3	4
8-tap	-1	4	-11	40	40	-11	4	1
7-tap	-1	4	-10	58	17	-5	1	

The HM 8.0 encoder uses a 7-tap filter and an 8-tap filter for sub-pixel interpolations (Table 5.1 [1]). To make the system faster, it was chosen to not use interpolating filters. As a result, the neural network can not be accurate in deciding between two similar angular modes. However, this is acceptable because the final mode decision is taken by the HM encoder after performing RDOQ and the neural network's guess is equivalent to the rough mode decision process used in HM 8.0 encoder.

Table 5.2 shows the accuracy of a neural network ($P[Q \in S]$) that was trained to identify the angular mode in an 16x16 block. Although it first appears that the neural net has a very low accuracy, it must be observed that the neural network can not distinguish between closely lying angles. 82% of the time, the neural network picked within ±3 of the best mode in the RDO sense. Figure 5.2 shows that the neural network's predicted modes (x-axis) are highly correlated with the best mode (y-axis).

Table 5.2: Accuracy of a neural network for mode decision

Set S	Accuracy of NN
$S = \{N\}$	37.9789%
$S = \{N, N \pm 1\}$	67.7177%
$S = \{N, N \pm 1, N \pm 2\}$	77.4162%
$S = \{N, N \pm 1, N \pm 2, N \pm 3\}$	82.2148%
$S = \{N, N \pm 1, N \pm 2, N \pm 3, N \pm 4\}$	85.3779%

It can be seen from table 5.2 that a neural network can be used to shortlist a set of prediction modes that can contain the best mode in an RDO sense, without the use

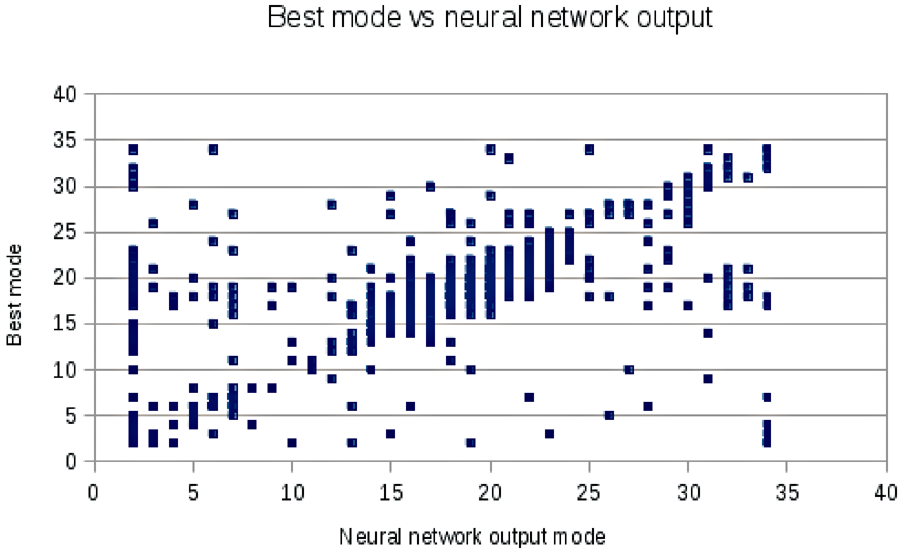


Figure 5.2: Correlation between best mode and neural net output for 16x16 block

of any pre-processing to extract features from the image block. While it is possible to further improve the accuracy of the neural network in several ways (increase number of hidden layers/neurons, cascade neural networks in chains to take decisions in stages and improve pre-processing techniques), the improvement comes at a cost of increased computational load required by the neural network stage in the encoder. In section 6.3, an improved method of determining set S from the output of the neural network is explored which allows higher accuracy (around 85%) at lower $\mathcal{L}[S]$ ($\mathcal{L}[S]$ around 7 to 8). Thus, a low accuracy fast neural network with very few hidden layers and very few neurons within the hidden layers can still be used to generate the set S where $P[Q \in S]$ is very high.

5.4 Neural networks architecture

Block sizes of 8x8 and 4x4 are very small and most angles require half and quarter pixel interpolation. While the neural networks have low accuracy at these sizes, it is still possible to use them to obtain good R-D characteristics while improving the time gain by avoiding upsampling and interpolation.

Table 5.3: Configuration of neural networks used for intra prediction

Neural net	Hidden layers	Number of neurons in		
		Input Layer	Hidden Layer	Output Layer
16x16 Neural Net	1	256	300	33
8x8 Neural Net	1	64	100	33
4x4 Neural Net	1	16	100	33

For block sizes of 16x16 and higher, a single neural network is trained to obtain the guessed mode. Larger block sizes are downsampled to 16x16 by skipping alternate rows and columns prior to being fed to a neural network. For 8x8 block sizes, a separate neural net is trained. This is because upsampling and interpolating an 8x8 block and feeding it to a 16x16 block is slow and does not provide significant accuracy improvements over a simple fast neural network trained for 8x8 blocks. A smaller neural network requiring fewer computations can provide adequate accuracy at this level and help speed the encoding time. Similarly, another low accuracy neural net is trained for 4x4 blocks.

Table 5.3 shows the configuration of the different neural networks used for intra prediction. A sigmoidal activation function shown in (5.1) was used for all the hidden layer neurons. The output layer neurons used a linear activation function shown in (5.2).

$$y = \frac{2}{1 + e^{-2sx}} - 1 \tag{5.1}$$

$$y_{out} = x.s \tag{5.2}$$

where s is a steepness parameter, x is the net input $\left(\sum w_i x_i\right)$, y and y_{out} are the outputs of the neurons in the neural network.

5.5 Summary

It is possible to train neural networks to identify the best prediction mode to use in a PB. Seperate neural networks are used for 4x4, 8x8 and 16x16 blocks. 32x32 blocks are downsampled by skipping alternate rows and columns and fed to the 16x16 neural net. By using neural networks to guess the best prediction mode, the decision cost is reduced.

CHAPTER 6

OBTAINING PREDICTION MODES FROM NEURAL NETWORKS

6.1 Accuracy of Neural Networks and System Performance

Figures 6.1, 6.2 and 6.3 show the probability of the neural network suggesting a mode N when the best mode is known to be Q for the 4x4, 8x8 and 16x16 neural networks. The graphs show the probability $P[N|Q]$ for all the angular modes. It can be seen that the neural network usually suggests a mode which is very close to the best mode. This is exploited to improve the system accuracy as explained below.

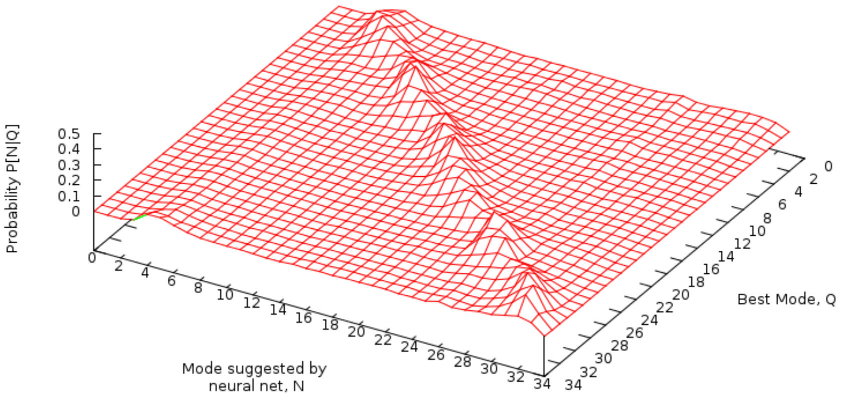


Figure 6.1: Performance of 4x4 neural network

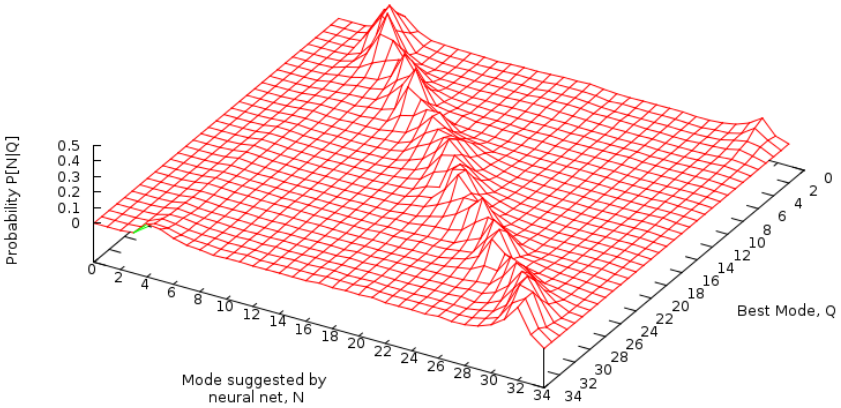


Figure 6.2: Performance of 8x8 neural network

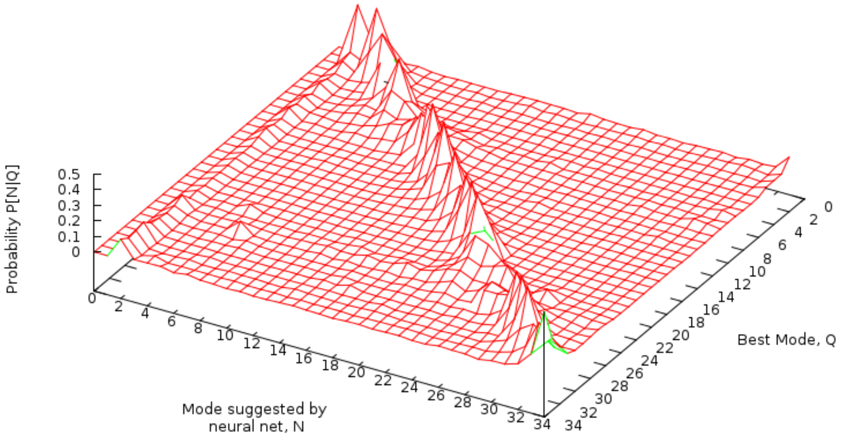


Figure 6.3: Performance of 16x16 neural network

6.2 Linear Search

The set of modes to be tested can be expanded as $S = \{N, N \pm 1, N \pm 2 \ldots\}$ where N is the mode suggested by the neural network. The decision cost C is given by equation 4.1. Figure 6.4 shows a plot of accuracy of the system with increasing decision cost for the 16x16 neural network.

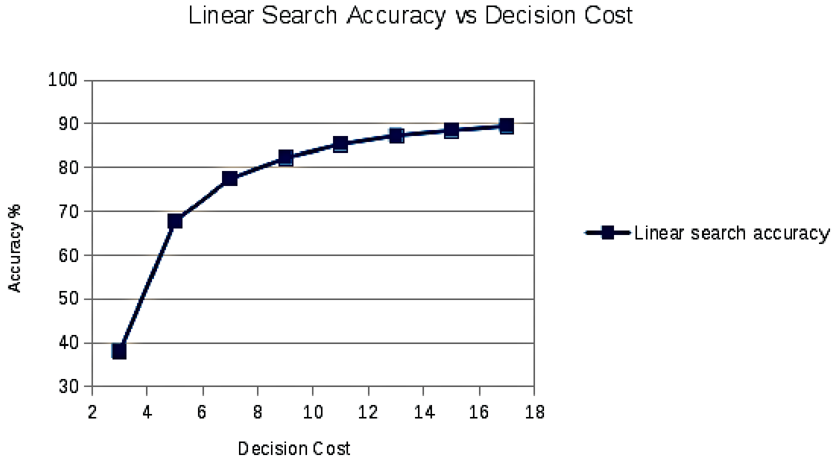


Figure 6.4: Linear search accuracy vs decision cost for 16x16 neural network

6.3 Bayesian Search

The linear search technique helps to improve accuracy but by increasing the decision cost. To reduce the decision cost, the linear search technique is improved to take advantage of the fact that the neural network is more accurate at guessing some modes than others.

Let

$$P[U_{M_i}] = P[N = M_i | Q = M_j] \tag{6.1}$$

where $M_i, M_j \in M$ and $i, j = 0, 1 \dots 34$. Then, from Bayes theorem,

$$\begin{aligned} P[V_{M_i}] &= P[Q = M_j | N = M_i] && (6.2) \\ &= \frac{P[N = M_i | Q = M_j] P[Q = M_j]}{P[N = M_i]} && (6.3) \\ &= \frac{P[U] P[Q = M_j]}{P[N = M_j]} && (6.4) \end{aligned}$$

Then, the set of modes to be evaluated can be generated as

$$S_i = \{M_1, M_2, M_3 \dots\} \tag{6.5}$$

such that

$$P[V_{M_1}] > P[V_{M_2}] > \ldots > P[V_{M_l}] \tag{6.6}$$

The length of set S_i, $\mathcal{L}[S_i]$, can be limited by setting a parameter α where

$$\alpha = P[V_{M_1}] + P[V_{M_2}] + \ldots + P[V_{M_l}] \tag{6.7}$$

By increasing α, the accuracy of the system increases along with the decision cost C. The decision cost C is given by (4.1). The advantage of this method is that when the neural network is more accurate at guessing a certain mode, the length $\mathcal{L}[S]$ required for that mode is very low, and when the neural network is not accurate at guessing a mode, $\mathcal{L}[S]$ increases for that mode. Thus, the decision cost C required for certain accuracy is lower than the decision cost required for the simple linear search described in section 6.2

The probabilities $P[U_{M_i}]$, $P[V_{M_j}]$, $P[Q = M_j]$ and $P[N = M_i]$ are obtained from running the HM simulation tool shown in figure 5.1. Then, a look up table is created to obtain a set S given the neural network output N according to (6.5), (6.6) and (6.7).

Figure 6.5 shows a plot of accuracy of the system with increasing decision cost for the 16x16 neural network when compared to the linear search.

6.4 Fast search technique

To evaluate the modes in set S, the encoder performs intra prediction for each of these modes and then performs RDOQ to determine the best mode. It has been found that RDOQ is very computationally intensive [15]. In order to reduce the number of times the encoder has to perform RDOQ, a fast search technique is implemented to short list the best modes to be tested for RDOQ.

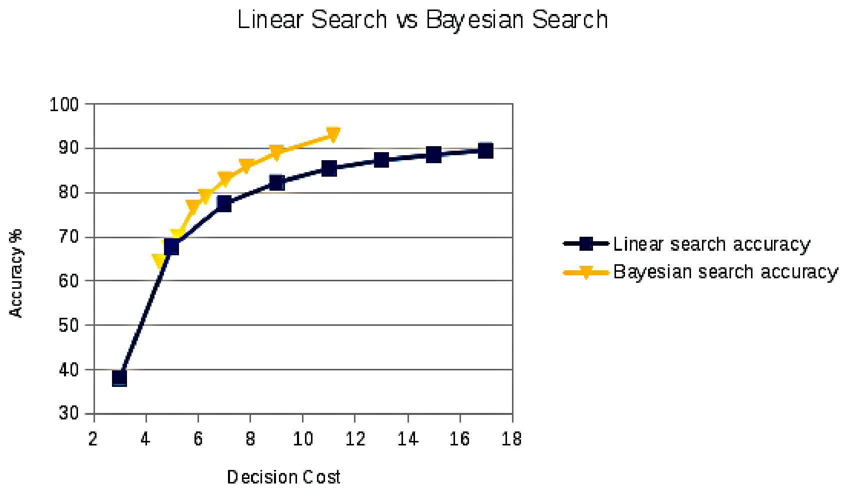


Figure 6.5: Linear search vs Bayesian search for 16x16 neural network

A rough mode decision is performed on every other element of the set S to calculate its rough cost when set S is arranged in descending order. The mode that has the lowest estimated cost is selected along with the two adjoining modes present in the set S. These three modes are selected for RDOQ to obtain the best mode.

Then the decision cost becomes

$$C_{fast} = \left[\frac{1}{2}\sum_{i=3}^{i=33} P[N_i]\mathcal{L}[S_i] + 3\right] + \left[\frac{1}{2}P[N_2](\mathcal{L}[S_2] + 2)\right] + \left[\frac{1}{2}P[N_{34}](\mathcal{L}[S_34] + 2)\right] \tag{6.8}$$

$$\approx \frac{1}{2}C_{Bayesian} + 3 \tag{6.9}$$

where $C_{Bayesian}$ is the cost of the Bayesian search tables without fast search, given by (4.1).The fast search provides two advantages: it reduces the average number of modes to be evaluated at each PU level and it reduces the number of modes that have to be subjected to RDOQ. It comes at a cost of reduced accuracy that leads to negligible loss in performance. When C_{fast} is much greater than $C_{Bayesian}$ for a certain mode, the fast skip technique is dropped and the entire set S from the Bayesian search table is used. Also, fast search technique does not provide significant benefits for 4x4 blocks so it is only used for 8x8 and larger PB blocks.

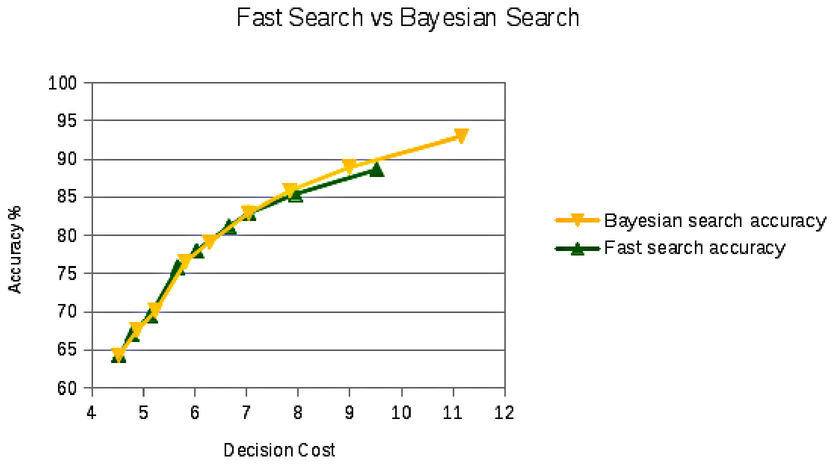


Figure 6.6: Bayesian search vs fast search for 16x16 neural network

Figure 6.6 shows a plot of accuracy of the system with increasing decision cost for the 16x16 neural network. Although the plot does not show any improvements over the Bayesian search technique, the difference is in the number of modes subjected to RDOQ. The fast search technique subjects only 2 or 3 modes to RDOQ while the Bayesian search subjects every mode to RDOQ. This makes fast search run faster despite having the same decision cost as Bayesian.

6.5 Summary

This chapter presents some techniques that are used to improve the accuracy of the intra mode decision process at the cost of increase in decision cost. The Bayesian search technique and the fast search techniques are discussed in terms of their accuracy and decision cost in this chapter.

The proposed algorithm can be summarized as:

Step 1: Feed the normalized pixel values to the neural network based on the size of the PB.

Step 2: Scan the output nodes of the neural network. The prediction mode associated with the higest value output node is the best guess of the neural network

Step 3: Obtain the set of prediction modes to be tested, S, from the Bayesian search tables or the fast search tables.

Step 4: Evaluate the modes in set S to determine which is the best prediction mode for that PB

The next chapter presents the coding time gain, loss of PSNR and increase in bitrate obtained by this algorithm.

CHAPTER 7

RESULTS

For testing the proposed system, the test sequences in table 7.1 were used.

Table 7.1: Test sequences used for testing the proposed system

No.	Sequence	Class	Resolution	Number of Frames
1	BQSquare	Class D	416x240 60Hz	600
2	BasketballDrill	Class C	832x480 50Hz	502
3	BQMall	Class C	832x480 50Hz	600
4	Traffic	Class A	2560x1600 30Hz	150
5	BasketballDrive	Class B	1920x1080 50Hz	502

7.1 Coding time gain over HM-8.0

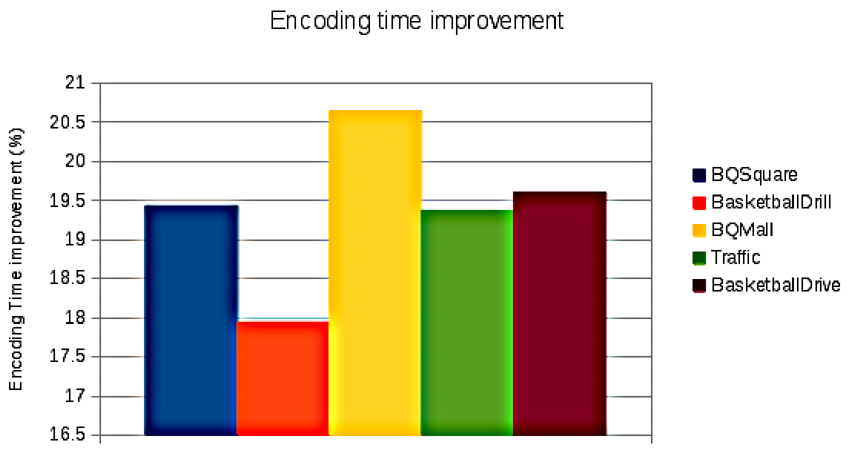


Figure 7.1: Encoding time gain for QP 20

Figure 7.1 shows the encoding time gain for various test sequences tested at QP 20.

Table 7.2: Encoding time gain, loss in PSNR and increase in bitrate of the proposed encoder compared to original HM encoder at QP 20

No.	Sequence	Class	Resolution	Number of Frames	Speedup (%)	PSNR loss (dB)	Increase in Bitrate (%)
1	BQSquare	Class C	416x240 60Hz	600	19.430	-0.154	-1.384
2	BasketballDrill	Class C	832x480 50Hz	502	17.939	-0.087	-2.109
3	BQMall	Class C	832x480 50Hz	600	20.640	-0.079	-0.740
4	Traffic	Class A	2560x1600 30Hz	150	19.370	-0.073	-0.658
5	BasketballDrive	Class B	1920x1080 50Hz	502	19.602	-0.098	0.006
6	Average				19.396	-0.098	-0.977

Table 7.2 shows the change in PSNR, bitrate along with the coding time gain for the various sequences.

7.2 Bitrate and PSNR Loss

The proposed system shows negligible bitrate increase and PSNR loss. Figures 7.2, 7.3, 7.4 and 7.5 show the bitrate-psnr graphs for the test sequences BQMall, BQSquare, BasketballDrill and Traffic respectively. It can be seen that the performance is very similar to the original HM 8.0 encoder.

The encoding was performed for QP values of 20, 24, 27 and 32. The change in average coding time gain with respect to QP is shown in figure 7.6

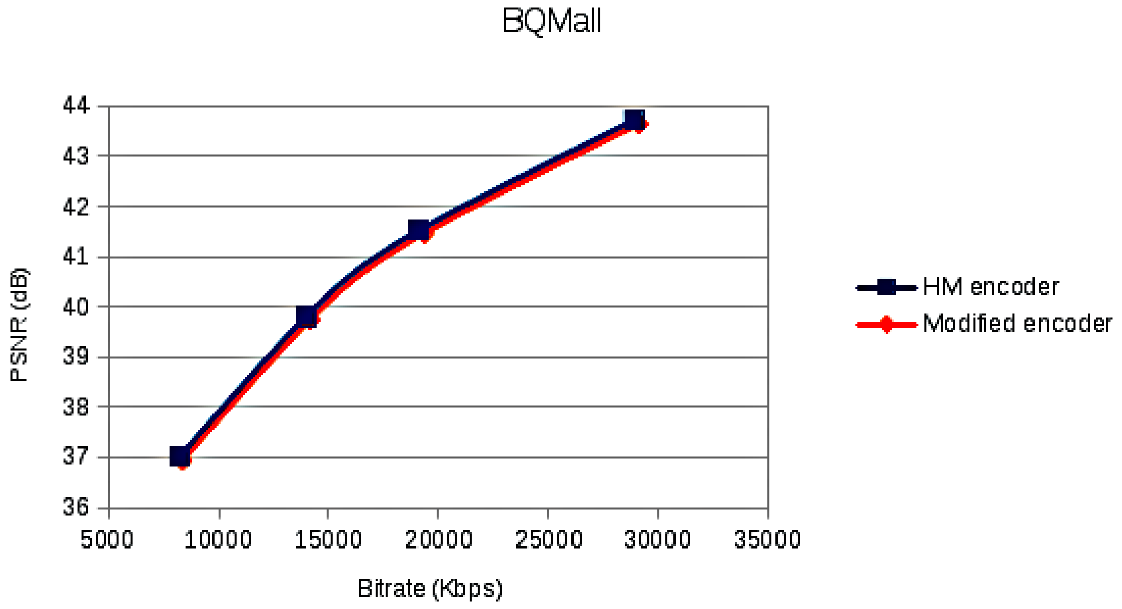


Figure 7.2: Bitrate vs PSNR for BQMall sequence

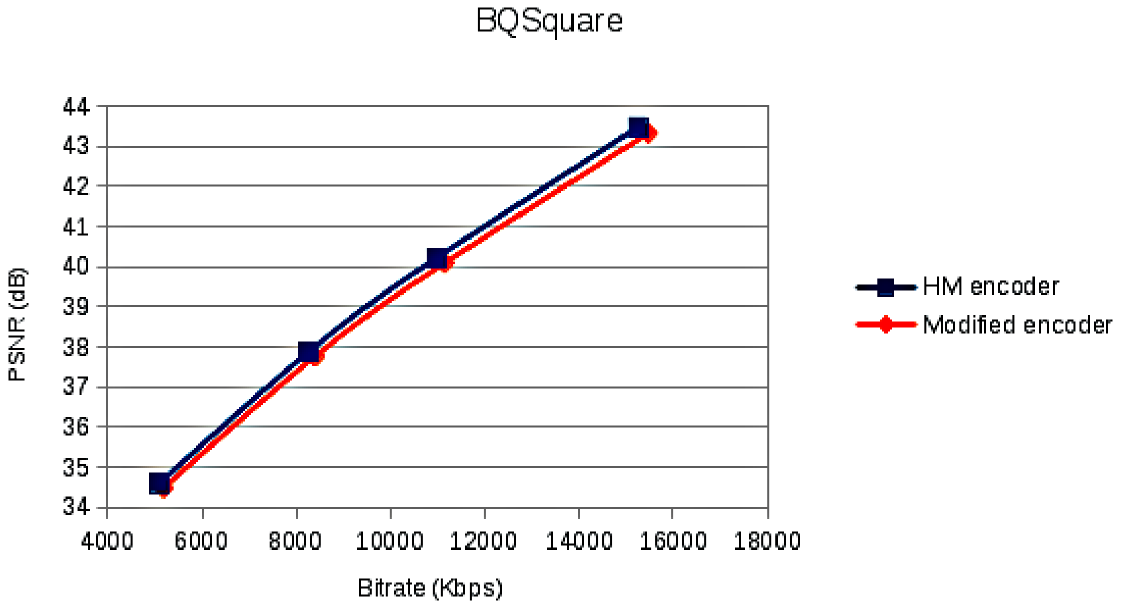


Figure 7.3: Bitrate vs PSNR for BQSquare sequence

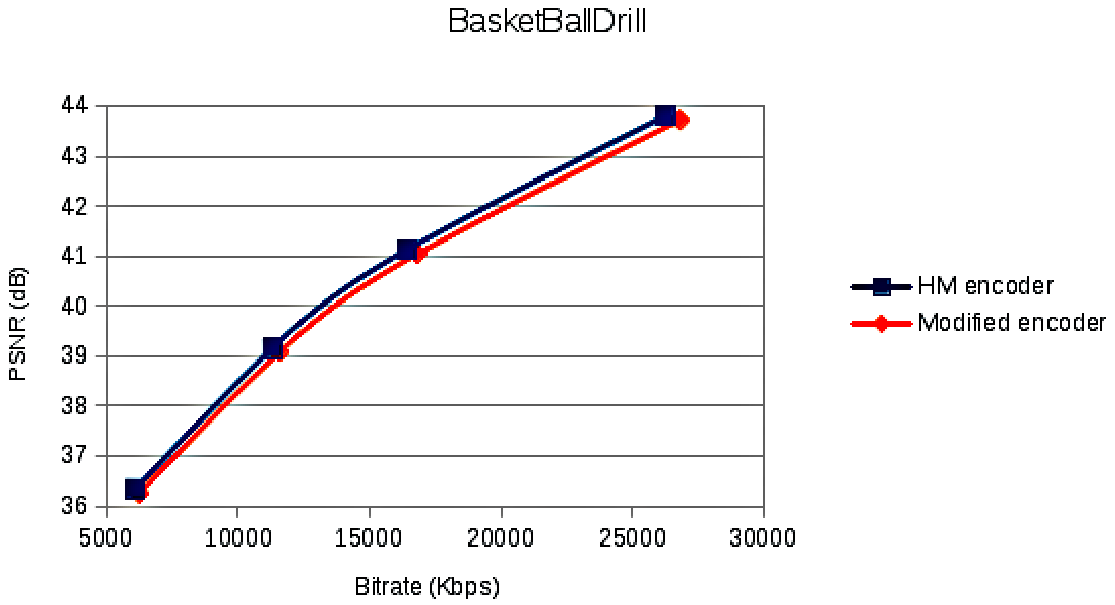


Figure 7.4: Bitrate vs PSNR for BasketballDrill sequence

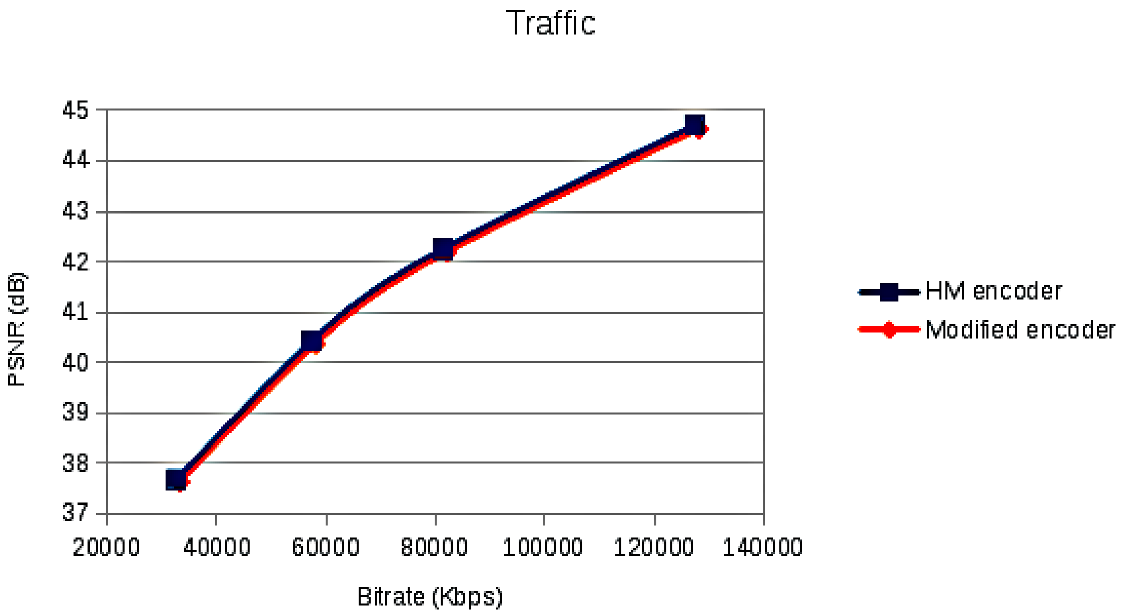


Figure 7.5: Bitrate vs PSNR for Traffic sequence

7.3 Summary

This chapter discusses the performance of the HEVC encoder with the proposed intra prediction scheme. An encoding time gain of upto 20% has been observed at a

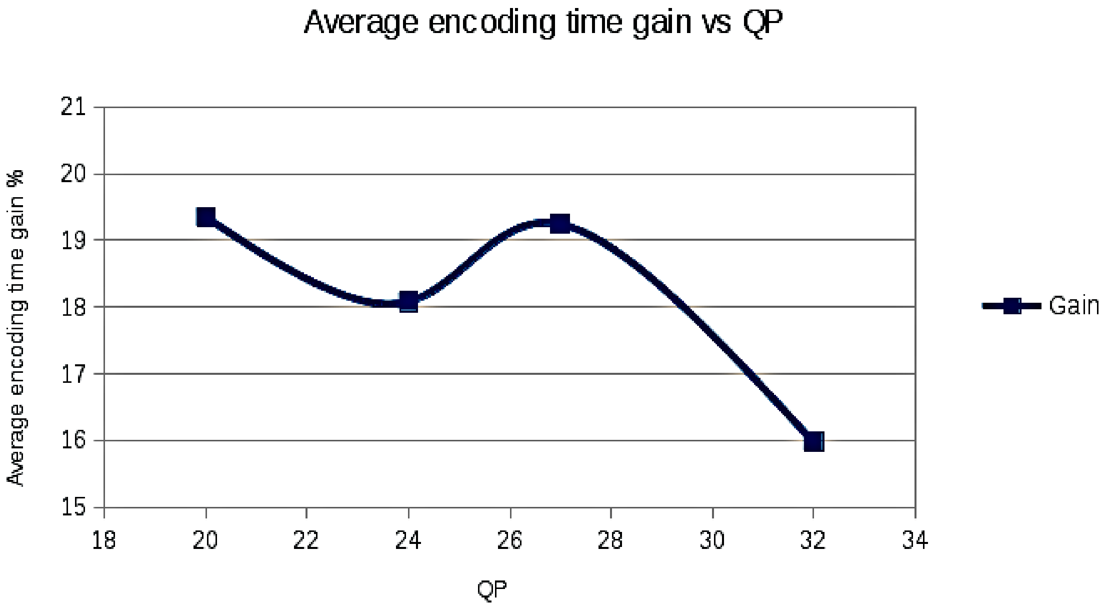


Figure 7.6: Gain vs QP

negligible loss of PSNR and negligible increase in bitrate for various test sequences at different values of the quantization parameter QP.

CHAPTER 8

CONCLUSIONS

In the current configuration, the proposed system provides upto 20% encoding time gain at negligible loss in performance. The average encoding time gain is 18.16%. The average increase in bitrate is 1.57% and the average loss in PSNR is 0.0856 dB.

8.1 Scope for future work

The results show that neural networks are a feasible way to speed up intra coding in HEVC. In the current implementation, the neural networks are used from a generic neural networks library. Specialized libraries exist that can run the neural networks on GPUs to achieve extremely high performance [19]. Since the neural networks account for roughly 20% in the total encoding time in this scheme, significant gains can be obtained by simply optimizing the neural networks.

Neural networks can also be run on dedicated custom hardware [20]. It is possible to reduce the time spent computing the state of the neural networks significantly with these technologies. When neural networks can be made faster, it is possible to design more complex systems that provide even greater accuracy at very little extra cost. When the accuracy of the neural networks increases, the decision cost C required for reaching a specified performance limit also reduces.

The neural networks in this thesis are trained to only recognize angular modes. It may be possible to train a neural network to recognize the planar and DC modes as well. This would directly reduce the decision cost by a factor of 2 when compared to this work.

The proposed method reduces the encoding time by reducing the number of modes to be evaluated. This is complementary to another approach taken by researchers where the focus is on limiting the quad tree depth that is traversed [21]. When combined with that technique, the total encoding gain will be significantly high.

Since the 4x4, 8x8 and 16x16 block sizes use separate neural networks in this thesis, it can be possible to parallelize them to check for all the modes in the quad tree simultaneously. This will produce significant encoding time gain over the current work. Also, when the neural nets are run in parallel, the combined outputs of all the neural networks may lead to an even more powerful algorithm for reducing the decision cost C.

By reducing the decision cost, one significant improvement is the number of times a buffer has to be loaded with the image, predicted signal and residual signal. This can be further investigated to find efficient ways to store data in a buffer to share the buffer between 4x4, 8x8 and 16x16 neural networks to further reduce the number of times data has to be moved.

This system can be implemented on an FPGA to evaluate the performance of the neural networks on hardware in terms of power consumption and encoding time. This information can lead to additional projects regarding implementing the neural net on a mobile device with custom hardware.

Finally, this system can be implemented for HEVC lossless, HEVC HE10 profile [22] and also be ported back to H.264/AVC standards [4].

APPENDIX A

Introduction to Artificial Neural Networks

The basics of artificial neural networks are covered in Appendix A.

The introductory book on neural networks [16] defines neural networks as a massively parallel distributed processor made up of simple processing units, which has a natural property for storing experimental knowledge and making it available for use which resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process
2. Intra neuron connection strengths, known as synaptic weights, are used to store acquired knowledge

A.1 Artificial Neurons

A single artificial neuron is implemented as shown in (A.1)

$$y = g\Bigg(\Big(\sum_i x_i w_i\Big) + b\Bigg) \tag{A.1}$$

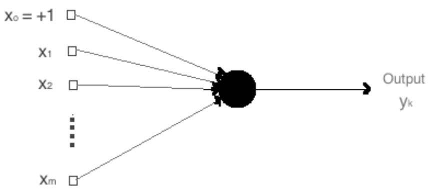


Figure A.1: An artificial neuron

where x_i are the inputs to the neural network, w_i is the weight of that input, b is a bias that is added and $g()$ is the activation function. The inputs to the network can be the outputs of other neurons, or can be supplied from an external source. The activation function is a function like (5.1) and (5.2). The output from each neuron is y in (A.1).

A connection of these neurons into a network forms an artificial neural network. Figure A.1 shows a schematic for a neuron and Figure A.2 shows how neurons can be connected to form a type of neural network called multi layer perceptron (MLP).

In general, there are three types of neural networks:

1. Single Layer perceptron
2. Multilayer perceptron
3. Reccurrant networks

Single layer perceptrons are neural networks that have only a single layer of neurons and recurrent networks are neural networks that have feedback loops with delays [16].

Neural networks that have one or more hidden layers are called multilayer perceptrons. Figure A.2 shows a 3 stage multilayer perceptron with 1 input layer, 1 hidden layer and 1 output layer. The source nodes to the input supply respective elements of the input vector (activation pattern) to the neurons in the hidden layer. The outputs of the third layer are used to drive the third layer and so on. [16].

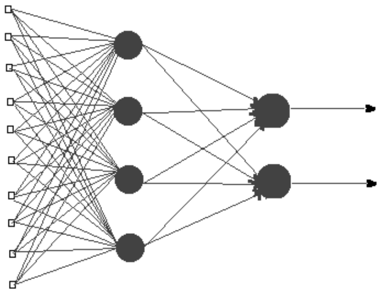


Figure A.2: 3 layer MLP

A.2 Training an artificial neural network

The knowledge of the neural network is stored in the form of the values of values of the weights and biases. Thus, the process of training the neural network involves adjusting the values of the weights and biases such that the output of the neural network is similar to the desired response for a set of training data. The training process can be seen as an optimization problem, where the mean square error of the entire set of training data should be minimized. This problem can be solved in many ways, from standard optimization huristics to special optimization techniques like genetic algorithms and gradient descent algorithms like back-propogation [16].

Although the back-propogation algorithm is popular, it has some limitations that are overcome with more advanced algorithms like RPROP [18].

A.3 Running cost of a multilayer perceptron

During the execution of a neural network, (A.1) must be computed for each neuron present in the network. This means one addition and one multiplication needs to be performed for each connection, and for the bias of the neuron. Then, the call to the activation function must be performed. If there are c connections, n is the total number of neurons and n_i is the number of input neurons in a net, A is the cost of multiplying and adding one weight and G is the cost of the activation function, total cost is

$$T(n) = cA + (n - n_i)G \tag{A.2}$$

If the neural network is fully connected, then we can express cost as

$$T(n) = (l - 1)(n_l^2 + n_l)A + (l - 1)n_l G \tag{A.3}$$

where l is the number of layers and n_l is the number of neurons in each hidden layer.

APPENDIX B

Test sequences used

Frames from each video are shown here. These test videos are accessed from `ftp://hvc:US88Hula@ftp.tnt.uni-hannover.de/testsequences`

B.1 BQMall

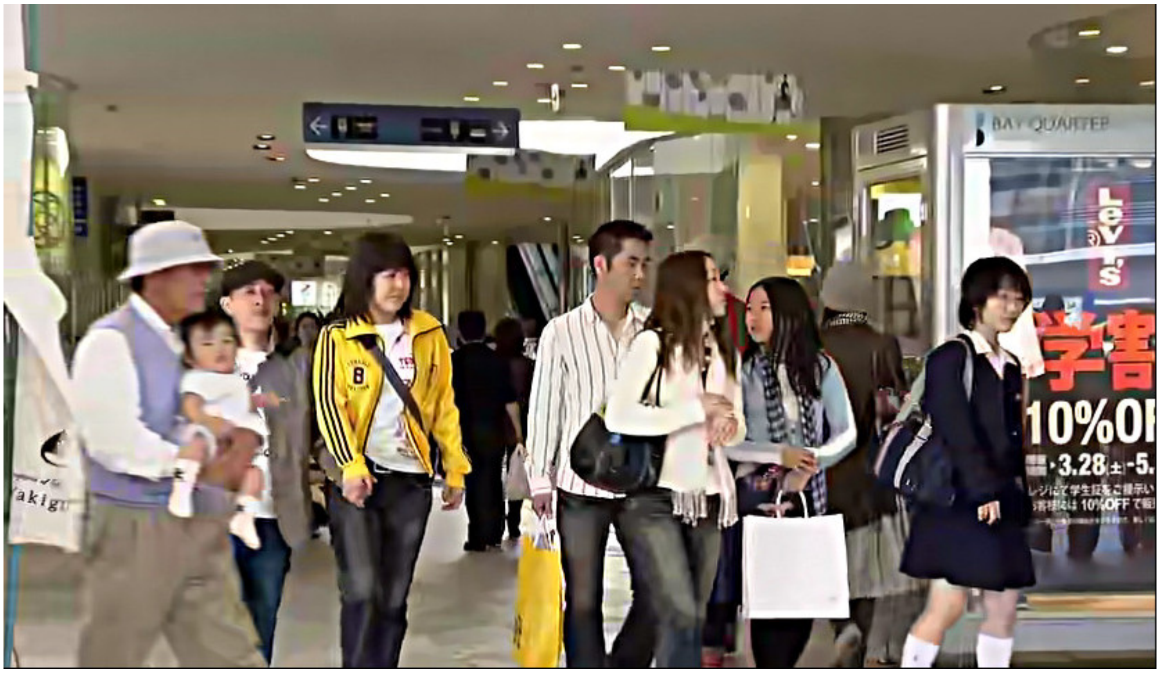


Figure B.1: BQMall

Width	:	832
Height	:	480
Frame rate	:	50Hz
Number of Frames	:	600

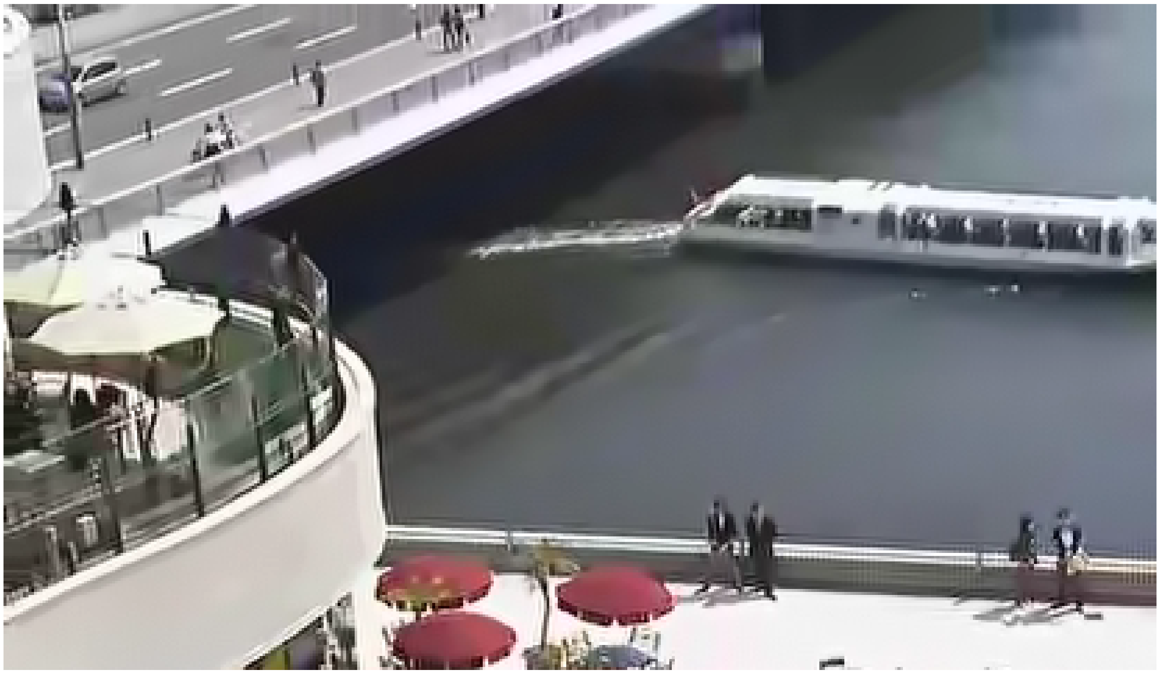


Figure B.2: BQSquare

B.2 BQSquare

Width	:	416
Height	:	240
Frame rate	:	60Hz
Number of Frames	:	600

B.3 BasketballDrill

Width	:	832
Height	:	480
Frame rate	:	50Hz
Number of Frames	:	502

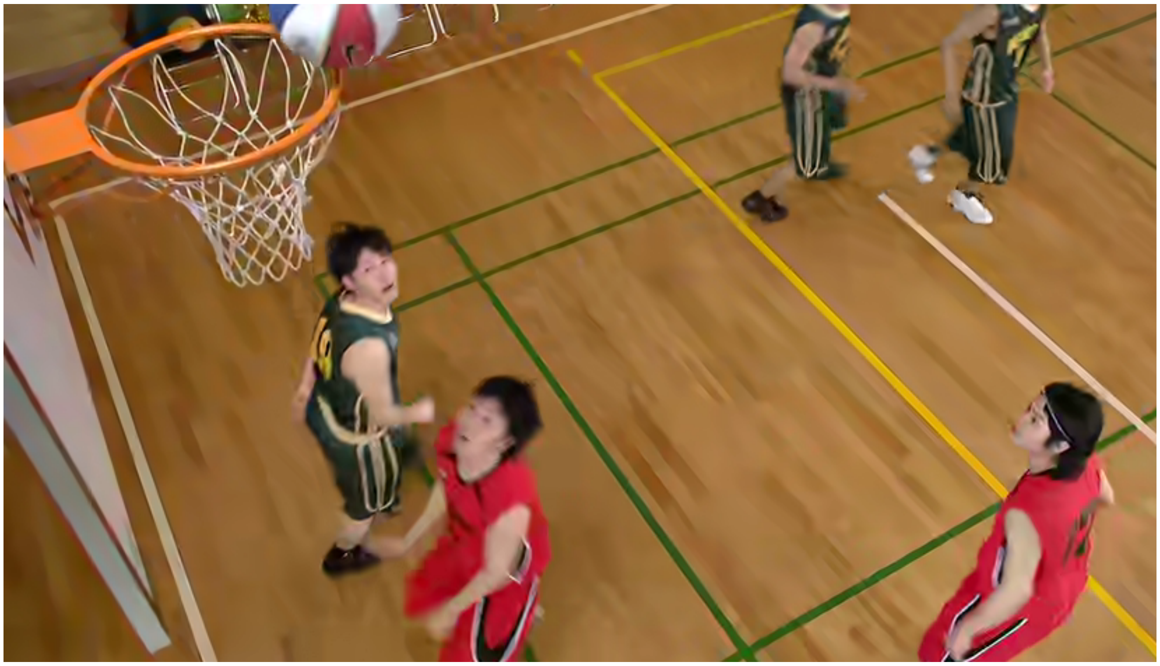


Figure B.3: BasketballDrill

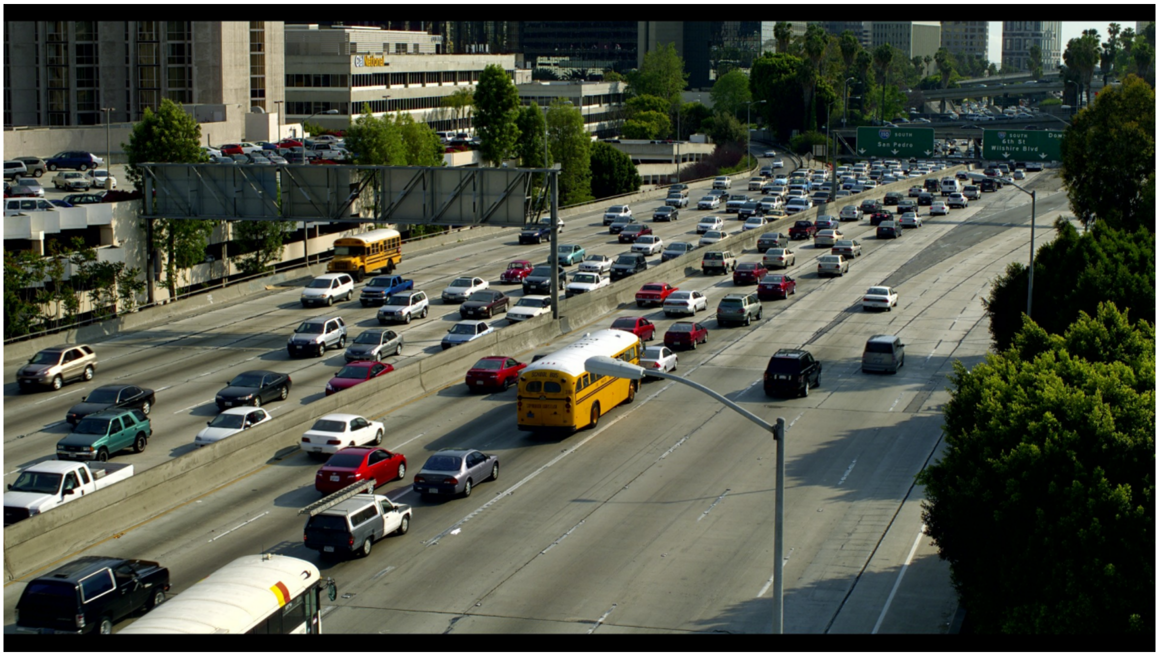


Figure B.4: Traffic

B.4 Traffic

Width	:	2560
Height	:	1600
Frame rate	:	30Hz
Number of Frames	:	150

REFERENCES

[1] G. Sullivan, *et al.*, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1649–1668, Dec 2012.

[2] K. Sayood, *"Introduction to Data Compression"*, 3rd ed. Morgan Kaufmann, 2005.

[3] J. Chen, U. Koc, and K. Liu, *"Design of Digital Video Coding Systems A Complete Compressed Domain Approach"*. Marcel Dekker, 2002.

[4] I. Richardson, *"The H.264 Advanced Video Compression Standard"*. Wiley, 2010.

[5] P. N. Tudor, "Mpeg-2 video compression," *Electronics & Communications Engineering Journal*, vol. 7, pp. 257 – 264, Dec 1995.

[6] T. Wiegand, *et al.*, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 560 – 576, Jul 2003.

[7] N. Ling, "High efficiency video coding and its 3d extension: A research perspective," *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 2150–2155, Jul 2012.

[8] I.-K. Kim, *et al.*, "HM8: High efficiency video coding (hevc) test model 8 encoder description," *JCTVC-J1002*, Oct 2012.

[9] G. Sullivan and R.Baker, "Efficient quadtree coding of images and video," *IEEE Transactions on Image Processing*, vol. 3, pp. 327–331, May 1994.

[10] P. Helle, *et al.*, “Block merging for quadtree-based partitioning in hevc,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1720–1731, May 2012.

[11] G. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, 1998.

[12] Choi, *et al.*, “Fast coding unit decision method based on coding tree pruning for high efficiency video coding,” *Optical Engineering*, vol. 51, no. 3, pp. 030 502–1–030 502–3, 2012. [Online]. Available: http://dx.doi.org/10.1117/1.OE.51.3.030502

[13] Y. Piao, J. Min, and J. Chen, “Encoder improvement of unified intra prediction,” *JCTVC-C207*, Dec 2010.

[14] H. Zhang and Z. Ma, “Fast intra prediction for high efficiency video coding,” *Pacific-Rim Conf. on Multimedia (PCM)*, Dec 2012.

[15] F. Bossen, *et al.*, “Hevc complexity and implementation analysis,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, Dec 2012.

[16] S. Haykin, *“Neural Networks A Comprehensive Foundation”*, 2nd ed. Pearson Education, 2006.

[17] S. Nissen, “Implementation of a fast artificial neural networks library (fann),” 2003, software available at http://leenissen.dk/fann/.

[18] M. Riedmiller. and H. Braun, “A direct adaptive method for faster backpropagation learning: the rprop algorithm,” pp. 586–591 vol.1, 1993.

[19] N. Lopes and B. Ribeiro, “Gpumlib: An efficient open-source gpu machine learning library,” *International Journal of Computer Information Systems and Industrial Management Applications*, p. 355362, 2011.

[20] F. M. Dias, A. Antunes, and A. M. Mota, "Artificial neural networks: a review of commercial hardware," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 8, pp. 945 – 952, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197604000934

[21] H. Zhang and Z. Ma, "Early termination schemes for fast intra mode decision in high efficiency video coding," *IEEE International Symposium on Circuits and Systems, ISCAS 2013*, May 2013.

[22] B. Bross, *et al.*, "High efficiency video coding (hevc) text specification draft 10 (for fdis & last call)," *JCTVC-L1003*, Jan 2013.

[23] C. H. Lampert, "Machine learning for video compression: Macroblock mode decision," vol. 1, pp. 936–940, 2006.

[24] Y. H. Tan, *et al.*, "On residual quad-tree coding in hevc," pp. 1–4, 2011.

[25] M. Viitanen, *et al.*, "Complexity analysis of next-generation hevc decoder," pp. 882–885, 2012.

[26] G. Correa, *et al.*, "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Transactions on Consumer Electronics*, Nov 2011.

[27] A. Leon-Garcia, *Probability, Statistics and Random Processes for Electrical Engineering*, 3rd ed. Pearson Education, 2008.

[28] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," *ITU-T SG16, Doc. VCEG-M33, 13th VCEG meeting, Austin, TX*, April 2001.

[29] G. Bjontegaard, "Improvements of the bd-psnr model," *ITU-T SG16 Q.6, Doc. VCEG-AI11, Berlin, Germany*, July 2008.

[30] G. Sullivan, "Hevc: The next generation in video compression," Nov 2012.

[31] X. Shen, L. Yu, and J. Chen, "Fast coding unit size selection for hevc based on Bayesian decision rule," pp. 453–456, 2012.

[32] "HM reference software svn repository." [Online]. Available: http://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[33] "Test video sequences." [Online]. Available: ftp://hvc:US88Hula@ftp.tnt.uni-hannover.de/testsequences

BIOGRAPHICAL STATEMENT

Dilip Prasanna Kumar was born in Bangalore, India, in 1989. He received his B.E. degree from Visvesvaraya Technological University, India, in 2011. In 2011, he enrolled at the University of Texas at Arlington to pursue a M.S degree in Electrical Engineering. He joined the Multimedia Processing Lab at the University of Texas at Arlington at this time. He worked at Qualcomm Incorporated as a summer intern from May 2012 to August 2012. After graduation he will return to Qualcomm Incorporated.