Design and Implementation of Deep Depth Decision Algorithm for Complexity Reduction in High Efficiency Video Coding (HEVC)

Helen K Joy[1], Manjunath R Kounte[2]
School of ECE
REVA University
Bengaluru, India

B K Sujatha[3]
Department of Electronics and Telecommunication
Ramaiah Institute of Technology
Bengaluru, India

***Abstract*—High efficiency video (HEVC) coding made its mark as a codec which compress with low bit rate than its preceding codec that is H.264, but the factor that stop HEVC from many applications is its complex encoding procedure. The rate distortion optimisation (RDO) cost calculation in HEVC consume complex calculations. In this paper, we propose a method to cross out the issue of complex calculations by replacing the traditional inter-prediction procedure of brute force search for RDO by a deep convolutional neural network to predict and perform this process. In the first step, the modelling of the deep depth decision algorithm is done with optimum specifications using convolutional neural network (CNN). In the next step, the model is designed and trained with dataset and validated. The trained model is tested by pipelining it to the original HEVC encoder to check its performance. We also evaluate the efficiency of the model by comparing the average time of encoding for various resolution video input. The testing is done with mutually independent input to maintain the accuracy of the system. The system shows a substantial saving in encoding time that proves the complexity reduction in HEVC.**

***Keywords*—CNN; HEVC; deep learning; RDO; encoding time; complexity reduction**

I. Introduction

Video compression is an area to explore while considering the flourish of video acquisition devices, social media, live transfer of videos etc. The high efficiency video encoding HEVC system possess a better compression compared to its previous system advanced video coding AVC. However, the computational complexity of HEVC is a matter of discussion because of its Rate distortion optimisation [1] (RDO) cost calculation for coding tree unit [2] (CTU). There for this computational complexity in HEVC is a matter of research interest, the focus will be to reduce the computational complexity[3] with better efficiency. Before going into the details let's review the evolution of HEVC its drawback and goodness compared to its ancestral system.

ITU-T video compression standard introduced H.261 is the year November 1988. In the H.26x family, first member, H.261 in video coding standards in the domain of the VCEG (ITU-T Video Coding Experts Group) then Specialists Group on Coding for Visual Telephony [4]. H.261 was originally designed to transmit data over ISDN lines with data rates as multiples of 64 Kbit/s. The coding algorithm was designed in such a way to work at video bit rates in 40 Kbit/s to 2 Mbit/s [5].MPEG-2 consists of "three different kinds of coded frames: I-frame /intra-coded frames, P -frame/predictive-coded frames, and B-frame/ bidirectionally-predictive-coded frames" [3]. The I-frame is a single uncompressed or raw frame that is a separately-compressed version. "The I-frame coding takes the advantage of spatial redundancy and the persistence of vision of human eye ie the inability of human eye to detect several changes in the image. I-frames do not depend on data in the previous or the next frames,"[6] Unlike in P-frames and B-frames, and because of that its coding matches with the still photography. The raw frame is spitted into 8X8-pixel blocks. The data in all block is transformed using discrete cosine transform (DCT)[6]and results is a matrix of size 8×8 of coefficients that have real number values. The DCT transform converts spatial domain into frequency domain, but it does not change the information in the block; if the DCT is calculated with perfect precision, the original block can be recovered clearly by applying the inverse discreet cosine transform [5] "H.263 [7]is a popular video compression standard for low-bit-rate compressed format focusing on videoconferencing. It was standardized by the organisation ITU-T, Video Coding Experts Group (VCEG) in 1995/1996"[4] . H.263 is the member of the H.26x family of video coding standards in the domain, ITU-T. Like other H.26x standards, H.263 is also based on (DCT) discrete cosine transform video compression. It was later advanced to add different additional enhanced features in the year 1998 and 2000. "H.264 is one of the popularly used codec on the planet, with significant note in optical disc, broadcast process, and streaming in video markets etc.[5] The applications are noted in Table I. Still, many uses of H.264 are subject to royalties, something that should is taken into considered before Google's WebM, as well as the general availability of decoding abilities on target platforms and devices" [8]. H. 264 mostly called as AVC (Advanced video coding), its block segmentation based, motion compensated with DCT technique. The aim behind AVC was to transfer video in low bit rate with better efficiency for UHD videos to its adaption of it.

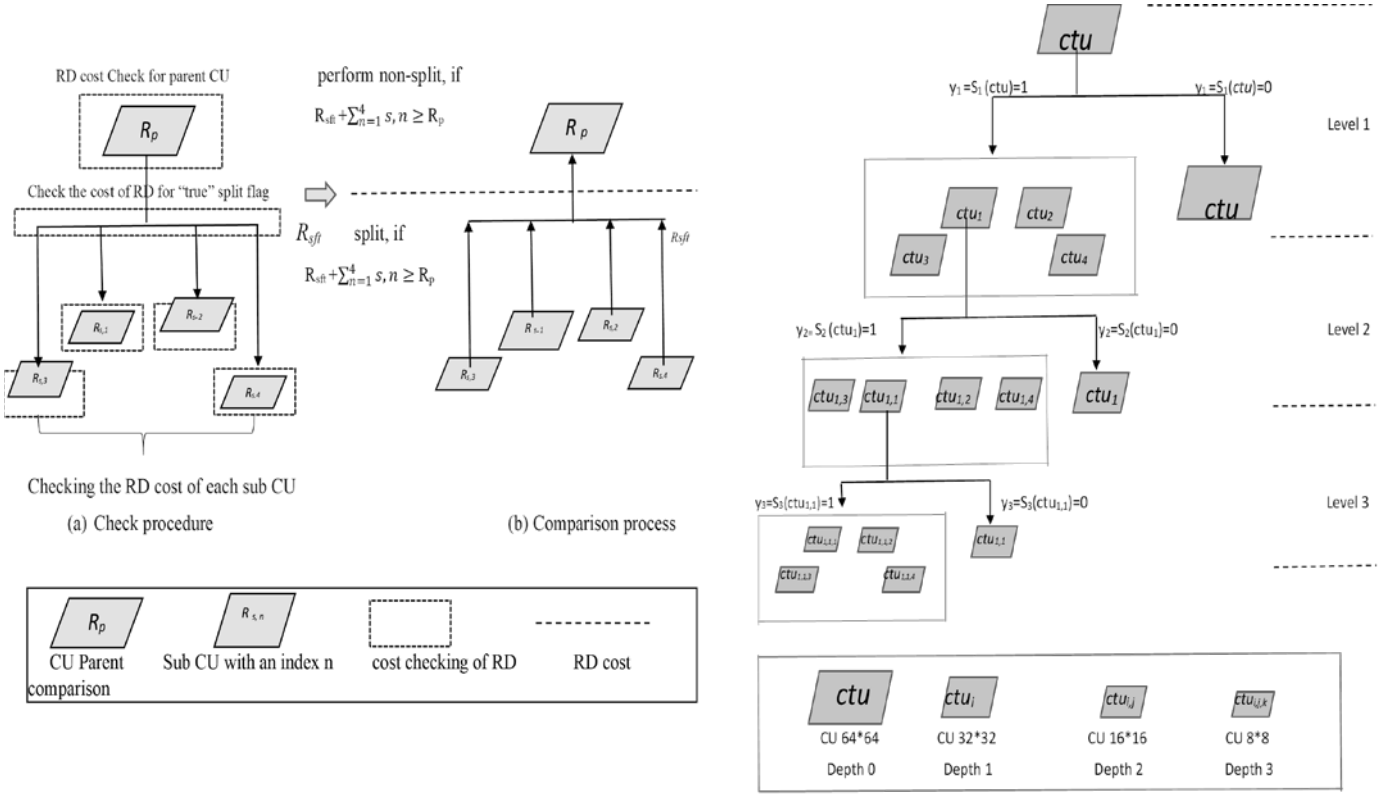


Fig. 1. (a) Rate Distortion Cost Calculation of CU Procedure. (b) The Representation of CU Classification as Layers to Analyze Depth.

"High Efficiency Video Coding, is also known as HEVC or H.265, is the step in this evolution. It builds off a lot of the techniques used in AVC/H.264 to make video compression even more efficient. When AVC looks at multiple frames for change those macroblock chunks can be a few different shapes and sizes, up to a maximum of 16 pixels by 16 pixels. With HEVC, those chunks can be up to 64×64 in size much larger than 16×16, which means the algorithm can remember fewer chunks, thus decreasing the size of the overall video" [9]. HEVC's quad tree [10] partitioning uses the brute force search for RDO (rate distortion optimisation) cost calculation. The complexity of the procedure is more when used with normal signal processing steps that makes the HEVC [11]complex. Fig. 1(a) shows the procedure of rate distortion optimisation as a flowchart. It is divided into check procedure and comparison procedure. It initially checks for the rate distortion cost of the parent CTU [12] and the total cost of splitting it till end. Once this procedure is done comparison is done. In comparison it will the checking the RD [11] cost of parent and the cost after splitting. if the RD cost after split is more than the system will not split it further and if the RD cost of parent is more then it will proceed with the split. This calculation procedure in HEVC is tedious that make the system complex. This issue was addressed by many algorithms, some provides enhancement to the existing HEVC system while other set provides a totally new algorithm [3] providing a new architecture [13] to perform the procedure of compression. Deep learning based algorithms [13] [14] started working on this in recent years. So a depth decision algorithm with deep CNN [15][16] is modelled to solve this issue. Fig. 1(b) shows the level and depth of CTU. Understanding this depth concept [6] helps in designing deep CNN [13] algorithm to predict depth and thus to make intra prediction less complex.

The paper aims in complexity reduction in video compression (HEVC) by reducing encoding time. It is achieved by designing a deep learning-based system that predicts the depth of the CTU by making the intraprediction procedure less complex. The design is evaluated by pipelining it with the original HEVC and evaluates the complexity of the system. The overall design idea is shown in Fig. 2. The paper is divided mainly into two halves, 1) design of the deep depth decision algorithm, here the deep depth decision algorithm is designed tested and validated for datasets and 2) evaluation and experimental results of the model pipelined with original HEVC, were the model is pipelined with the original HEVC and the performance is evaluated for various resolution videos. The paper is concluded with the results showing encoding time reduction and future scopes.

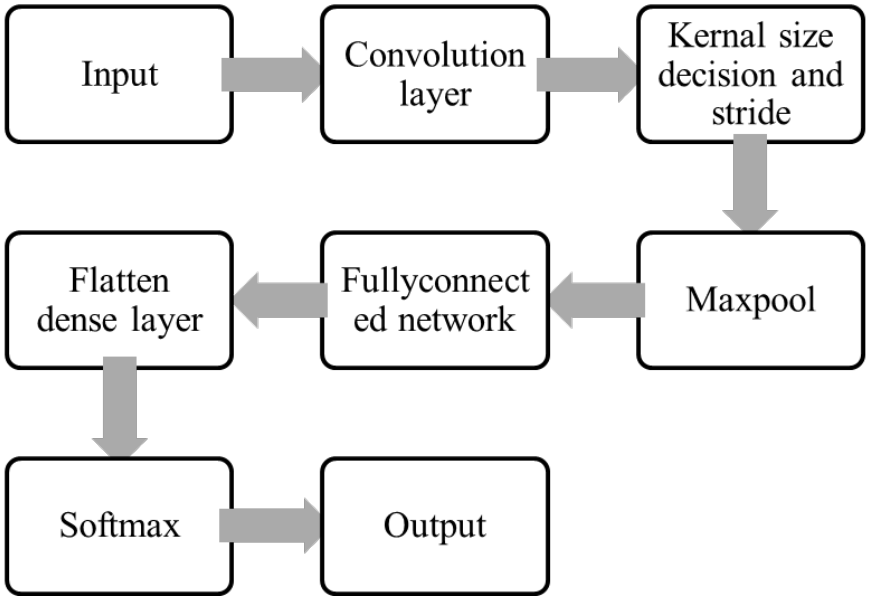


Fig. 2. Illustration of Steps in Order for the Modelling of Deep Depth Decision Algorithm with CNN.

II. Design of Deep CNN Depth Decision Algorithm for Inter Prediction

The inter prediction and its computational complexity was the issue took for analysis to model a new network. The design of this network should possess less computational complexity compared to the existing system and should be compatible to the existing codec. The design chosen should be compatible for faster transmission of frames while coding, so scalability and compatibility will also be the focus while designing, considering all this convolutional neural network (CNN) is chosen for this purpose so that all features are extracted correctly from the frames to produce better prediction as shown in Fig. 3.

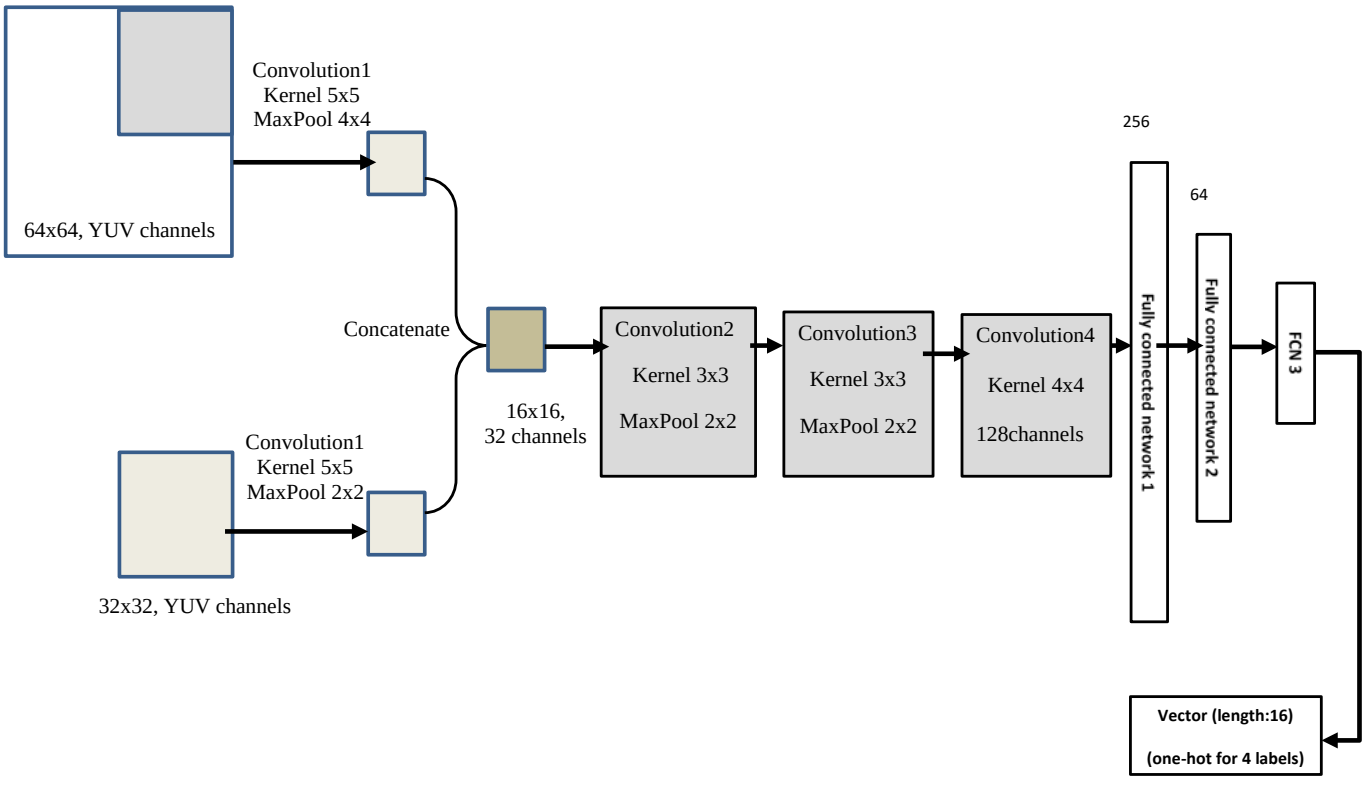


Fig. 3. The Representation of Deep Depth Decision Algorithm Model, with Input as 64X64 Patch followed by Convolution, MaxPool, Fully Connected Network followed by the 16 Length Output.

The CNN[17] used here is having multiple layer, the initial layer is the input layer. The input used here is video frames. The video frame can be of various properties, the YUV is the format chosen for this evaluation, other formats are also compatible in this model. The next layer in CNN model is convolutional layer, this is the layer that extracts the features of the frame based on the kernel used. The kernel size can be chosen based on the features that need to be extracted, if the kernel size is big it collects the global features or information from the frame whereas the small kernel [12] extracts the local features. Based on the need of the feature kernel can be chosen. In the design 5X5, 3X3 along with the 16X16, 4X4 [18] are also used, so model clearly extracts the global and local features from the frame. To cover all the inputs zero padding is used in this model. The stride used is same as the width of the kernel used in each case in the design. After extracting the feature its max pooled to reduce the size and converge the multiple values to a single value or less values. The activation function helps to decide the neuron is fired or not, so activation function is the node is kept in between and in the end of neural network. Here the activation function [19] used is ReLU rectified linear unit. ReLu maintains a value between (0- ∞) zero and infinity by avoiding negative values. It's a simple function that returns if input is negative else returns the same value in other cases. Both forward and backward propagation exist in CNN [20] network. Here in the model training uses backward propagation while validation uses forward propagation.

The model designed takes the input as YUV CTU of 64X 64, the first convolution layer users its kernel and convers as 32X32 coding unit. The both CU of 32x32 are concatenated to extract more features and its polled to 16X16 patch. The next stage of convolution with 3x3 kernel extracts its fine features, and a 4x4 for global features. After feature extraction in each stage the data are polled by 2x2. In final stage the fully connected later flatten the information and compress it using SoftMax to 256 to 64 to a 16-length vector holding all the information of the CTU depth. The model is trained with various resolution input varying from 240p to 4k. After training the model will be having a training loss factor of 3.1049. the loss function estimated in this model is the cross entropy. Cross-Entropy Loss can be evaluated for separate images and independently and finally added together to obtain the final cross entropy as each path are mutually independent. A 66.12% of accuracy is obtained by the trained model.

A. Dataset

The dataset is the collection of sample video frames used for testing training and validation of the design proposed. Multiple and verity in dataset helps in the improvement of accuracy in the model. Here the dataset contains the Coding Unit image file extracted from the YUV video files as set of input and their corresponding depths for HEVC intra-prediction as output to train the proposed system. The dataset chosen here has multiple resolution and are not of same pattern videos to maintain the quality and efficiency of the model.

In HEVC intra-prediction, each I-frame is divided into 64x64 (CTU). For each 64x64 CTU, there's a depth prediction represented by a 16x16 matrix. The elements in the matrix are 0, 1, 2 or 3, indicating depth 0/1/2/3 for a 4x4 block in the CT. The dataset contains images and corresponding labels. There're three folders: train, validation, test Image files: Each image may have different size based on the resolution of the video, and it is one frame extracted from a video. While using in the system, split the image into several 64x64 images or 32x32 and so on.

Labels: The labels are in separate folder called pkl folder. For one CTU, which is a 64x64 image file, the label will be a Python list with a length of 16. The length is 16 vectors instead of a 16x16 matrix, because there's redundant information for a 16x16 matrix, and it can be reduced to a 16x1 vector. So, for a 64x64 CTU, it has 16 labels, each label corresponds to a 16x16 image block in the CTU. If the frame is split into 64x64 CTUs, the size of the train dataset is around 110K images. The size of the validation dataset is around 40K images. The name of the image file will be like: v_0_42_104_.jpg, were v represents Video Number, followed by FrameNumber, CTU number and image extension. The Video Number is to find the corresponding .pkl file, like v_0.pkl. To get the label for a certain 64x64 CTU, index the dict by:

label_vector = video_dict [FrameNumber][CtuNumber], for example: label_vector = video_dict ["42"] ["104"]. The label_vector will be a length 16 Python list. Dataset loading in deep learning projects implemented in PyTorch can be done by in load_example.py.

In the final stage for evaluation and comparison CPIH data set is also used to know the performance of the proposed system verses the existing models. The CPIH data set is not used in any of the testing or training for proper quality check.

B. Input and Pre-Processing Layer

The input used here is the YUV image patch derived from video frames. Each of this will be saved in a folder with separate labels in a python dictionary. The raw inputs need to be pre-processed by down sampling and splitting into 64x64, 32x32 and so on.

C. Convolution Layer

This layer performs convolution operation between the input and the kernel. If i is the pre-processed input and k is the kernel of size varying from 5X5 ,4x4,3x3 etc the convolution block output can be formulated as equation 1 and * represent the convolution operation. The size of the kernel decides the nature of the feature extracted.

In the design both global and fine features are extracted with variant kernels.

$$output = \mathrm{i} * \mathrm{k} \quad (1)$$

D. Fully Connected Layer

The fully connected layer initially flattens the output of convolution layer to a large single dimensional vector. The SoftMax operation helps to compress it further to required size without losing the information in it. The FCN1, FCN2 and FC3 along with averaging help it to shrink to 16 length vectors changing from 256 to 64 to 16.

E. Other Layers

The system is having a loss of feature dropping as the stages are crossing so the activation function ReLU[21][22], rectified linear unit shown in equation 2 where z is the input and R(z) is the output of ReLu . It should be noted that the output is activated by sigmoid function represented by S(z) in equation 3.

$$R(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases} \quad (2)$$

$$S(z) = \frac{1}{1+e^{-z}} \quad (3)$$

In original HEVC the prediction process is complex and time consuming as it should predict the RDO cost, so here the CNN network [23],[24]with depth decision [25][26] helps to predict the depth of each patch of 64X64 to a 16-length vector whereas original HEVC needs a matrix of size 64X64 to store it. The model converts the input patch to a 16 vector which can predict all the characteristics of that CTU with depth information as 0/1/2/3. The model is designed to take the input as 64x64 but while processing its split into 32x 32. Predicting for 64x64 patch directly doesn't make sense so the actual input is 32x32. The depth is 0 when the patch is not split and encoded as it is. The 64x 64 patch represent 16 length vectors. So, for representing 32x32 the vector required is 4x4. So, in the output blocks of four 4x4 patches will b available as output for a CTU of 64x64. Each value in the vector indicate the depth of the CTU. if the first vector is 0 it says that It is a 64x64 patch and if depth is 1 means the 64x64 CTU is split once into four 32x32 CTUs and so on.

III. Evaluation and Experimental Result Analysis of Deep CNN Depth Decision Algorithm for Inter Prediction

The designed model is allowed to work with the HEVC codec as shown in Fig. 4. To simulate the original HEVC, HM software is used. The evaluation is done between the original HEVC and proposed model for intraprediction, pipelined to HEVC using CPIH dataset. Integrating neural network models in HEVC encoder helps to test the complexity reduction using deep-learning-based method in HEVC intra-prediction. Using neural networks, the system can directly predict the Coding Unit (CU) depths for each frame. The intention is to speed up the encoding process of HEVC encoder. Thus, after we have a trained model, another thing that needs to be done is to integrate the deep learning prediction process into the HEVC encoder.

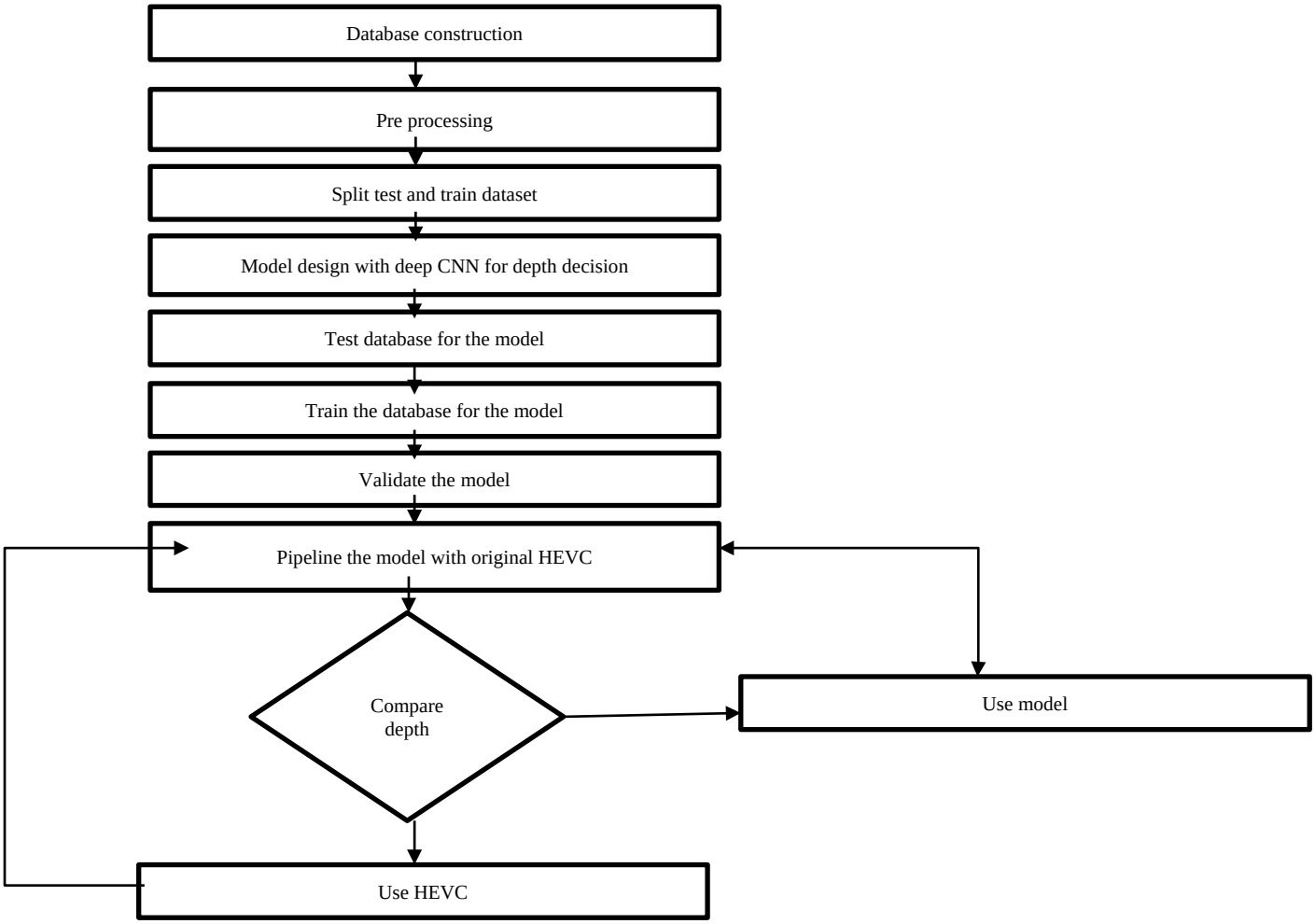


Fig. 4. Pipelining Structure with Deep Depth Decision Algorithm Model Added to the Original HEVC Model. It shows the Overall Steps for the Evaluation of the Designed Model.

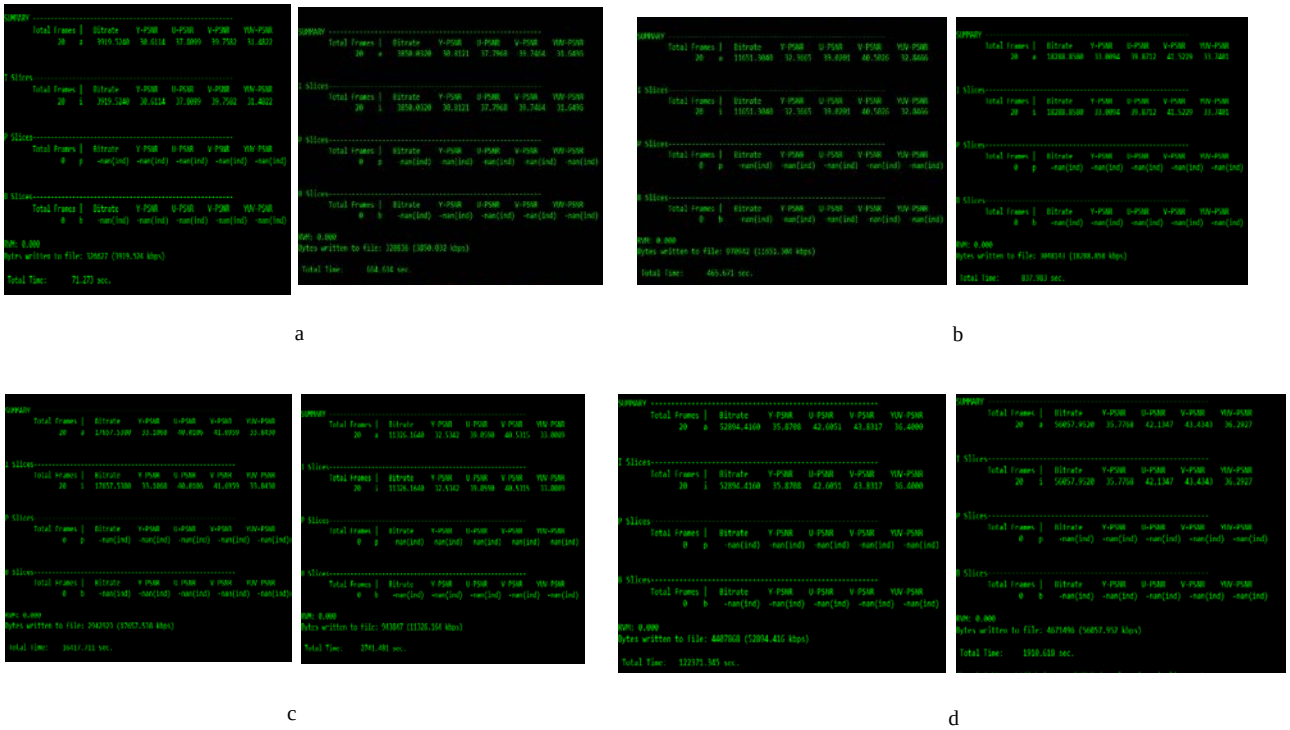
This is to check the compatibility of the model with existing HEVC and also it makes the evaluation of the performance of our neural network model easier. This pipeline is used for evaluating the performance of a neural network model in HEVC intra-prediction process. Comparing the difference in encoding time, Y-PSNR, U-PSNR, V-PSNR, YUV-PSNR with the original HEVC encoder helps to know the efficiency of the model. FFmpeg, Python3, PyTorch are the requirements to perform this.

The frames are send for encoding to HM software for original HEVC encoding and calculates encoding time, Y-PSNR, U-PSNR, V-PSNR, YUV-PSNR and the same set is send to the pipelined model or the proposed model and calculates the encoding time, Y-PSNR, U-PSNR, V-PSNR, YUV-PSNR. Both are evaluated and made as graphical representation to check the performance comparison.

The results show that the time of encoding with and without pipelining the deep CNN network is shown in the Table I for some sample input. The input chosen for the test is mutually independent from the training set to maintain the accuracy and the wide range of resolution is also considered to check the performance of the system foe different resolution video frames. The results clearly show that there is a change in encoding time and thus the system proves it can reduce the and bit rate in each case, it supports the encoder with a better performance. Computational complexity of the original HEVC is high due to the RDO cost calculation. The experimental results show the time of encoding is drastically changed to low values for the proposed method. The PSNR curve is slightly low here compared to original model but the system performance is not affected by this. The total process is done in python environment, when it's done, it will output with all information on the command line, like the encoding time, YUV-PSNR and so on. A sample output is shown in Fig. 5 and the comparison graphs are shown in Fig. 6. The proof of reduction in complexity is shown in Fig. 7 with the change in encoding time.

TABLE I. ENCODING TIME COMPARISON

Image source	**Resolution**	**Time of encoding with HEVC pipelined with Deep CNN network(T1)**	**Time of encoding with original HEVC(T2)**	**T1-T2**	**Δ T proposed**
CPIH dataset	768x512	71.273	664.634	-593.361	-89.27
	1536x1024	467.671	2741.481	-2273.81	-82.94
	2880x1920	2741.481	16417.71	-13676.2	-83.30
	4928x3264	1910.618	122731.3	-120821	-98.44
Average %Δ T improvement					88.49



a b c d

Fig. 5. Output Window showing the Bitrate-PSNR, U-PSNR, V-PSNR, YUV-PSNR, of Video Frame for Resolution (a) 768x512, (b) 1536x1024, (c) 2880x1920,(d) 4928x3264 with the State-of-Art Method and by HEVC-HM Software Simulation.

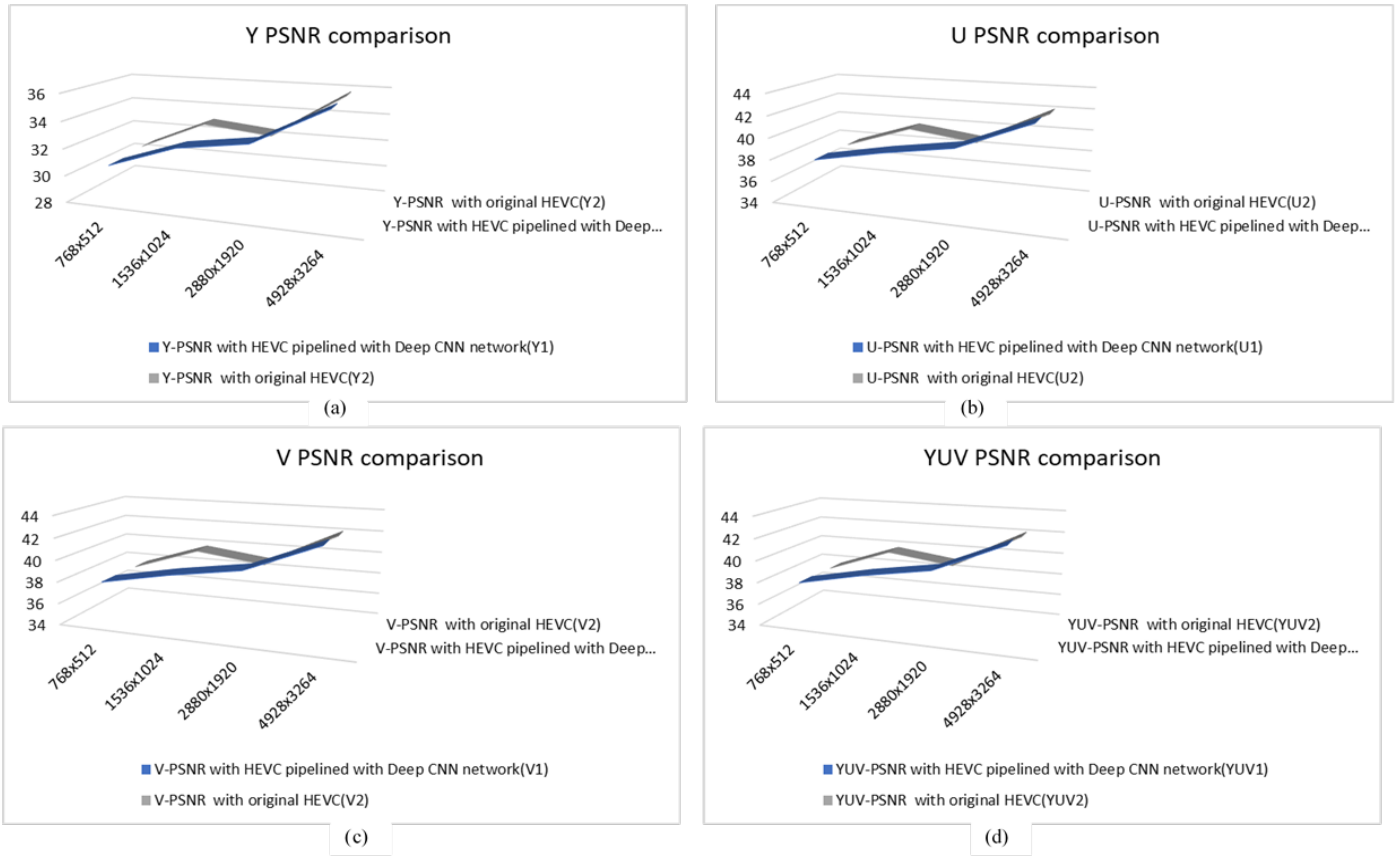


Fig. 6. Comparison Chart Showing (a) Y-PSNR, (b) U-PSNR (c) V-PSNR (d) YUV-PSNR, with the State-of-Art Method and by HEVC-HM Software Simulation for Video Frames of Different Resolution.

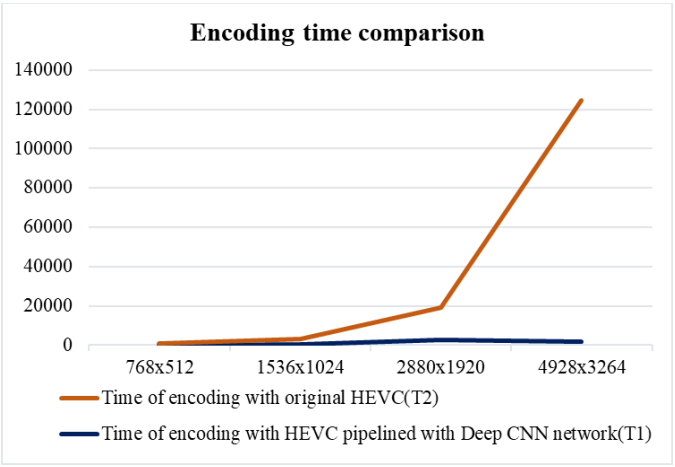


Fig. 7. Encoding Time Comparison.

IV. Conclusion

In this paper, a deep learning based inter-prediction is proposed to avoid the computational complexity issue in HEVC which predict the depth of the CTU in a 16-length vector than calculating the RDO cost by traditional signal processing method. The modelling adopted CNN network to perform this model with deep layers to predict the depth. The data set used for training was YUV and its tested on CPIH dataset to maintain the accuracy of the system and to avoid transfer or copied learning. The trained model is converted to system and pipelined to the original HEVC system to check the performance. The system evaluated the time of encoding with and without pipelining and calculated ΔT. The results and simulation clearly show that the design suits for the HEVC to work with less encoding time thus by reducing the complexity of the HEVC. The results prove it, the future enhancement on this can focus on the extension of this to inter prediction that improve the HEVC more.

References

[1] D. A. and N. G., “Combined spatial temporal based In-loop filter for scalable extension of HEVC,” ICT Express, vol. 6, no. 4, pp. 306–311, 2020.

[2] H. K. Joy and M. R. Kounte, "An Overview of Traditional and Recent Trends in Video Processing," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), 2019, pp. 848-851, doi: 10.1109/ICSSIT46314.2019.8987896.

[3] X. Li and N. Gong, “Run-Time Deep Learning Enhanced Fast Coding Unit Decision for High Efficiency Video Coding,” J. Circuits, Syst. Comput., vol. 29, no. 3, pp. 1–19, 2020.

[4] G. J. Sullivan, S. Member, and T. Wiegand, “Video Compression—From Concepts to the H.264 AVC Standard.pdf,” vol. 93, no. 1, 2005.

[5] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, 2012. Joy.

[6] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wanga, “Image and Video Compression with Neural Networks: A Review,” IEEE Trans. Circuits Syst. Video Technol., vol. 8215, no. SEPTEMBER 2018, pp. 1–1, 2019.

[7] J. L. Lin, Y. W. Chen, Y. W. Huang, and S. M. Lei, “Motion vector coding in the HEVC Standard,” IEEE J. Sel. Top. Signal Process., vol. 7, no. 6, pp. 957–968, 2013.

[8] Eirina Bourtsoulatze, Aaron Chadha, Ilya Fadeev, Vasileios Giotsas, and Yiannis Andreopoulos. "Deep Video Precoding", IEEE Trans. Cir. and Sys. for Video Technol. 30, 12 (Dec. 2020), 4913–4928. DOI:https://doi.org/10.1109/TCSVT.2019.2960084.

[9] D. Liu, Z. Chen, S. Liu, and F. Wu, “Deep Learning-Based Technology in Responses to the Joint Call for Proposals on Video Compression with Capability beyond HEVC,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 5, pp. 1267–1280, 2020.

[10] J. Lainema, F. Bossen, W. J. Han, J. Min, and K. Ugur, “Intra coding of the HEVC standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1792–1801, 2012.

[11] S. Bouaafia, R. Khemiri, F. E. Sayadi, and M. Atri, “Fast CU partition-based machine learning approach for reducing HEVC complexity,” J. Real-Time Image Process., vol. 17, no. 1, pp. 185–196, 2020.

[12] C. Ma, D. Liu, X. Peng, L. Li, and F. Wu, “Convolutional Neural Network-Based Arithmetic Coding for HEVC Intra-Predicted Residues,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 7, pp. 1901–1916, 2020.

[13] J. K. Lee, N. Kim, S. Cho, and J. W. Kang, “Deep video prediction network-ased inter-frame coding in HEVC,” IEEE Access, vol. 8, pp. 95906–95917, 2020.

[14] B. S. Kumar and V. U. Shree, "An End-To-End Video Compression Using Deep Neural Netowrk,” JAC : A Journal of Composition Theory ISSN : 0731-6755 vol. XIII, no. Xi, pp. 209–215, 2020.

[15] O. Alharbi, “A Deep Learning Approach Combining CNN and Bi-LSTM with SVM Classifier for Arabic Sentiment Analysis,” Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, pp. 165–172, 2021.

[16] P. R. Lai and J. S. Wang, “Multi-stage Attention Convolutional Neural Networks for HEVC In-Loop Filtering,” Proc. - 2020 IEEE Int. Conf. Artif. Intell. Circuits Syst. AICAS 2020, pp. 173–177, 2020.

[17] S. Katiyar and S. K. Borgohain, “Comparative evaluation of CNN architectures for image caption generation,” Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 12, pp. 793–801, 2020.

[18] L. Zhao et al., “Enhanced Ctu-Level Inter Prediction With Deep Frame Rate Up-Conversion For High Efficiency Video Coding" Institute of Digital Media & Cooperative Medianet Innovation Center , Peking University , Beijing , China Department of Computer Science , City Univers,” 2018 25th IEEE Int. Conf. Image Process., no. Mv, pp. 206–210, 2018.

[19] G. Sreenu and M. A. Saleem Durai, “Intelligent video surveillance: a review through deep learning techniques for crowd analysis,” J. Big Data, vol. 6, no. 1, pp. 1–27, 2019.

[20] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, “Learning Convolutional Networks for Content-Weighted Image Compression,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 3214–3223, 2018.

[21] H. K. Joy, M. R. Kounte and A. K. Joy, "Deep Learning Approach in Intra -Prediction of High Efficiency Video Coding," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 2020, pp. 134-138, doi: 10.1109/ICSTCEE49637.2020.9277189.

[22] Joy H.K., Kounte M.R. (2022) Deep CNN Depth Decision in Intra Prediction. In: Subramani C., Vijayakumar K., Dakyo B., Dash S.S. (eds) Proceedings of International Conference on Power Electronics and Renewable Energy Systems. Lecture Notes in Electrical Engineering, vol 795. Springer, Singapore. https://doi.org/10.1007/978-981-16-4943-1_1.

[23] Amitha I C, N S Sreekanth and N K Narayanan, “Collaborative Multi-Resolution MSER and Faster RCNN (MRMSER-FRCNN) Model for Improved Object Retrieval of Poor Resolution Images” International Journal of Advanced Computer Science and Applications(IJACSA), 12(12), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0121270.

[24] Naga Deepti Ponnaganti and Raju Anitha, “Feature Extraction based Breast Cancer Detection using WPSO with CNN” International Journal of Advanced Computer Science and Applications(IJACSA), 12(12), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0121250.

[25] Sigit Widiyanto, Dheo Prasetyo Nugroho, Ady Daryanto, Moh Yunus and Dini Tri Wardani, “Monitoring the Growth of Tomatoes in Real Time with Deep Learning-based Image Segmentation” International Journal of Advanced Computer Science and Applications(IJACSA), 12(12), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0121247.

[26] Shridevi Jeevan Kamble, Manjunath R Kounte, “SG-TSE: Segment-based Geographic Routing and Traffic Light Scheduling for EV Preemption based Negative Impact Reduction on Normal Traffic”, International Journal of Advanced Computer Science and Applications, Vol. 12, No. 12, 2021, pp. 274-283.