International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V70I1P223

Volume 70 Issue 1, 199-203, January, 2022
©2022 Seventh Sense Research Group®

A Neural Network-based Interframe Prediction for HEVC

Kanike Sreenivasulu[1], T V K Hanumantha Rao[2]

[1]*Scientist RCI/DRDO, Department of ECE, NIT, Warangal, Telangan, India.*

[2]*Associate Professor, ECE Department, NIT, Warangal, Telanagana India*

[1]skanike@rcilab.in, [2]tvkhrao75@nitw.ac.in

Abstract* — *Exploration of spatial and temporal redundancies in video data is one of the most important processes in video encoding procedures, contributing to the high compression capability of the H.265 architecture, one of the latest video codecs. The aim of this paper is to come up with a deep learning-based approach for the same and contrast it with the accuracy current motion vector-based prediction system.

Keywords — *HEVC, H.265, inter-frame, neural networks, Video Compression, LSTMs.*

I. INTRODUCTION

Video compression and decompression or codec algorithms have been around for four decades. They have become a necessity in today's era, owing to the ever-increasing resolution capabilities of video cameras and their increasingly high storage and transmission requirements. It can be seen that while an average 1080p video of length 1 minute would turn out to be approximately 130MB in size, the needed bandwidth to stream such videos is only between 8 to 16 Mbps. This can be attributed to advancements in video codec algorithms. The High-Efficiency Video Coding standard/ H.265 is one of the latest standards in video encoding designed by MPEG as a successor to the H.264/ AVC standard. It promises a 25 percent to 50 percent reduction in bit rate without compromise in video quality. Though similar in architecture, the improved features in the H.265 include the use of CTUs instead of macroblocks, better in-loop filters, etc., leading to greater accuracy in the encoding process with a reduction in bit rate[1].

A prominent characteristic of video data is redundancy. This basically refers to the similarities within video data. There are two types of redundancies: spatial, referring to the similarities within a specific image, and temporal, referring to the similarities between two consecutive frames. Using these redundancies to obtain residual frames, which are much lesser in size when compared to the original frames, is a pivotal part of video compression[2]. Lesser the error between a reconstructed frame and the original frame, the lesser the data that is to be encoded. In this paper, the aim is to use an LSTM or Long Short Term based deep learning approach to do inter-frame prediction to come up with a reconstructed frame, which has to be made as accurate as possible. This is done with a sequence of N previous frames to obtain a predicted (N+1) th frame, with which a residual frame is generated, occupying much less space when encoded. The paper is structured as follows: an overview of HEVC and LSTMs is provided, followed by the construction of the neural network architecture for inter-frame prediction, concluding the same with results and comparisons.

II. AN OVERVIEW OF THE HEVC ARCHITECTURE

The HEVC architecture is an improved architecture to the H.264, having the following basic building blocks for the encoding and decoding process.

A. Analysis block / Prediction block: This block, consisting of the interframe and intraframe prediction blocks, is a key block in generating residual frames. In intraframe prediction, the given frame is reconstructed by exploiting spatial similarities in images. This is done with the help of coding tree units (CTUs) and 35 intra prediction modes [3], which offers a wide range of flexibility when reconstructing a predicted frame. It also consists of inter-frame prediction modules, which use the previous and successive frames to reconstruct the present frame with the help of motion vectors. The difference between the original and reconstructed frame is taken as the residue frame, which is encoded.

B. The Discrete Cosine Transform: The discrete cosine transform is a type of linear spatial transformation. It is capable of separating the image into sub-bands of differing importance and is able to do so with lesser coefficients when compared to other transforms like the fast Fourier transform[4]. In video compression algorithms like the HEVC, DCT is done through Transformation Units (TUs) of sizes 8x8, 16x16, etc. In general, it is computed as follows:

$$\mathrm{DCT}(i, j) = \frac{1}{\sqrt{2N}} C(i)\, C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \mathrm{pixel}(x, y) \,\mathrm{COS}\left[\frac{(2x+1)i\pi}{2N}\right] \mathrm{COS}\left[\frac{(2y+1)j\pi}{2N}\right]$$

$$C(x) = \frac{1}{\sqrt{2}} \text{ if } x \text{ is } 0, \text{ else } 1 \text{ if } x > 0$$

○

C. Quantization: Before encoding the data into binary format, it is essential to quantize the DCT coefficients into input values. This is a lossy process and leads to distortion in image data. The quantizer is designed such that it maps quantization parameters between 0 to 51 for an 8-bit sequence, with a twofold increase in step size each time there is an increase of 6 in the quantization parameter[5].

D. CABAC (Context adaptive binary arithmetic coding): It is a form of lossless encoding that converts the information in the spatial domain to binary form. It uses statistical properties to compress data such that the number of bits used to represent the data is logarithmically proportional to the probability of the data. For instance, when compressing a string of characters, frequently used characters are each represented by a few bits, while infrequently used characters are each represented by many bits.[6]

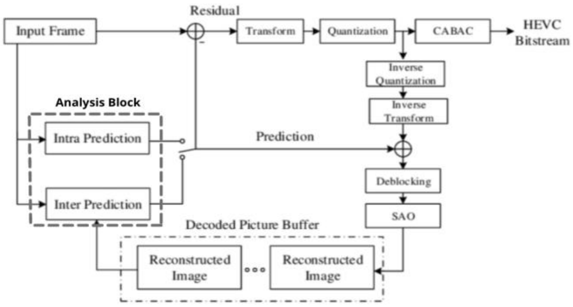

Fig. 1.A block diagram of the H.265 encoding.

With these components in mind, one can trace the full working of the HEVC algorithm with the block diagram shown in Fig. 1. Initially, an input frame is obtained. Assuming that this is not the first frame of the sequence, the interframe prediction module is selected, which is denoted by a switching mechanism. The reconstructed image of the previous frame is also fed into the interframe prediction module. A residual frame is generated from the difference of the two, which is then transformed and quantized, following which the residual frame is encoded. Now, one can also observe a feedback loop. Inverse quantization is done to the quantized residual frame, which then goes through an inverse transform block to get back the original residual frame. This is then added to the predicted frame to get back the original frame. This frame is now used as an input to generate the next residual frame in interframe prediction.

III. AN OVERVIEW OF THE ANALYSIS BLOCK IN HEVC

Since the aim of the paper will be based on modifications to the analysis block, it is worthwhile to take a closer look at its components and mechanism. As already seen, it has two modes interframe prediction and intraframe prediction.

The exploitation of spatial redundancies is done with the help of intraframe prediction modes. Modes 0 and 1 are named DC and planar modes, while modes 2 to 33 are named intra angular modes. The prediction of an N x N pixel square within the image is performed by taking 2N reference pixels to the left and top of the square to be predicted and performing computations with these reference pixels. The difference between this predicted frame and the original frame is encoded by the HEVC algorithm[7].

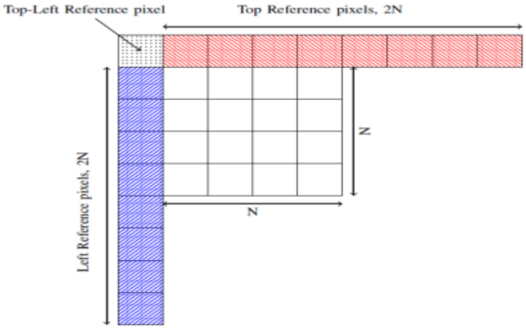

Fig. 2.A diagrammatic representation of reference cell selection.

Interframe prediction, on the other hand, uses temporal redundancies. When observing two successive frames, one can notice that the changes between the frames are minimal. Therefore, It only makes sense to encode the difference between the two frames. This is done through the interframe prediction module. In both interframe and intraframe prediction, to speed up the process, the entire image is divided into coding tree units/ CTUs [8].

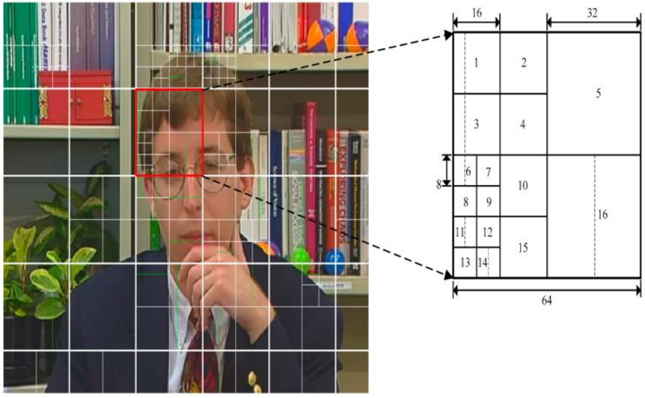

Fig. 3.The Coding tree Unit partitioning mechanism

IV. A BRIEF OVERVIEW OF LSTM

LSTMs (Long Short Term Memory) come under a class of neural networks called recurrent neural networks or RNNs. The various neurons of a recurrent neural network have an internal neuron state, which serves as a memory. This neuron state is used to process incoming information to the neural network. However, this internal memory is not stored for very long in a simple RNN, which in turn led to the formation of LSTMs.

LSTMs are capable of learning short-term as well as long-term correlations in the incoming data. This is done through the help of three different internal layers. Each of these layers helps in obtaining data, selective learning or forgetting data, and providing an output to the next layer. The sigmoid and tanh activation functions are crucial in an LSTM. The input layer obtains information from the previous time step, the forget layer decides which information has to be retained and which information to be forgotten, and the output layer decides which information should go to the next layer of the LSTM network. This is very useful when dealing with temporal data and time series [9].

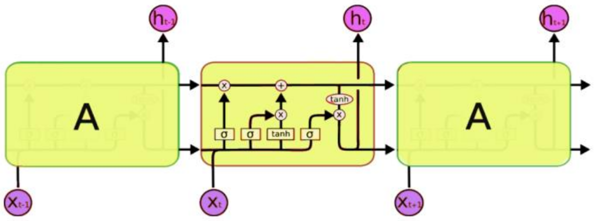

Fig. 4. An overview of an LSTM network.

While dealing with image data, instead of normal matrix multiplications in neural networks, convolution operations are performed for better feature learning by the neural network. Hence, instead of a normal LSTM, a convolutional LSTM is used for the task of inter-frame prediction[10].

V. EXISTING METHODS FOR INTER-FRAME PREDICTION

Interframe prediction in today's codecs like the HEVC uses the concept of motion estimation and compensation. It accounts for over 70 percent of the compression time in the generation of a compressed data stream[11]. In a motion estimation-based algorithm, an optimal motion vector has to be calculated for every block based on the previous video frames. A searching algorithm has to be implemented to find the block with the closest similarity in the current frame when compared to the previous frame. Hence, along with the residual frame, the computed motion vectors have to be sent to the decoder side, which incurs a cost in terms of memory. Also, the algorithm applied for motion prediction and compensation is computationally intensive. There is also the added disadvantage of generating blocking artifacts, which needs an added deblocking filter to increase image accuracy.[12]

VI. CONSTRUCTION OF THE NEURAL NETWORK

The paper uses an LSTM based convolutional neural network to get an accurate prediction of the next frame. Supervised learning methods are used, with a sequence of frames as the input and the predicted frame as the output. The model was trained using a wide variety of videos present in the UCF101 provided by the Centre for Research in Computer Vision. The network consists of two parallel Children networks that work together to create an ensemble of neural networks that can give highly accurate predictions of the next frame. The video frames are resized to 96 x 96 for easier training.

- The neural network will initially be using a generator module to take a sequence of 5 images to generate an output of (96,96,3*5) for the R, G, and B channels of the (96,96) images
- The input layer to the neural network of dimension (96,96,3,5) to obtain 5 images of dimensions (96,96,3) for each of the images. An input sequence of 5 frames is used to predict the 6th frame.
- Gaussian noise is added to these frames. This is shown to sometimes work more effectively than dropout when it comes to vision-based neural networks[13].
- From here, the neural network splits the incoming data into two subnetworks. The first one is a very simple layer that simply returns the last layer of the image sequence.
- The second network consists of a conv_lstm2D layer, which is an LSTM layer performing convolutional operations instead of regular matrix multiplications.
- This is then followed by a 2D convolution with 3 filters to get a (96,96,3) output.

The output matrix is combined with the output from the simpler first branch to get the final predicted frame. [14]

Perceptual distance is used as a metric here instead of the usual root mean square error as a metric for minimization.

$$\Delta C = \sqrt{\left(2 + \frac{\bar{r}}{256}\right) \times \Delta R^2 + 4 \times \Delta G^2 + \left(2 + \frac{255 - \bar{r}}{256}\right) \times \Delta B^2}$$

Since the next frame has to be reconstructed as accurately as possible, RMSE (Root Mean Square Error) or other standard loss functions may not be able to quantify perceptual accuracy. For this, the network uses perceptual distance as a function to be minimized while training. [15]

VII. SIMULATIONS AND RESULTS

A variety of videos with varying ranges of motion was used to perform inter-frame prediction. The video sequence, ground truth, and predictions are shown below.

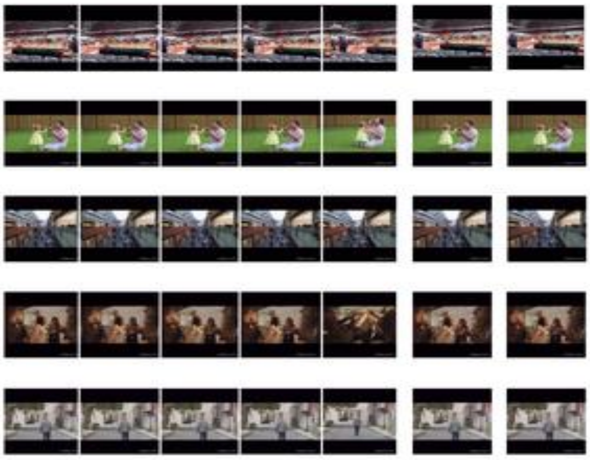

Fig. 5. Simulation results in the order of image sequence, ground truth, and the predicted frame using Basketball Blowing Bubbles, Market, party, and Man Walking (from top to bottom).

The residual frames generated for each of the sequences are given below.

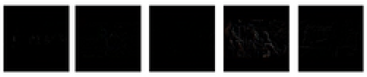

Fig. 6. Residual frames are generated for the 6th image of each sequence, each frame corresponding to the image sequence from top to bottom.

A. PSNR Calculations

The PSNR of a standard H.265 motion vector-based interframe prediction and the neural network-based prediction is compared.

IV. TABLEI
Comparing PSNRs of a Standard motion compensation-based interframe prediction with our neural network-based prediction

Image Sequence	PSNR of Motion vector-based inter-frame prediction	PSNR of Neural Network-based inter-frame prediction.
Basketball	33.15	38.914
Blowing Bubbles	34.23	39.931
Market	33.15	44.809
Party	33.20	25.869
Man Walking	33.18	38.678

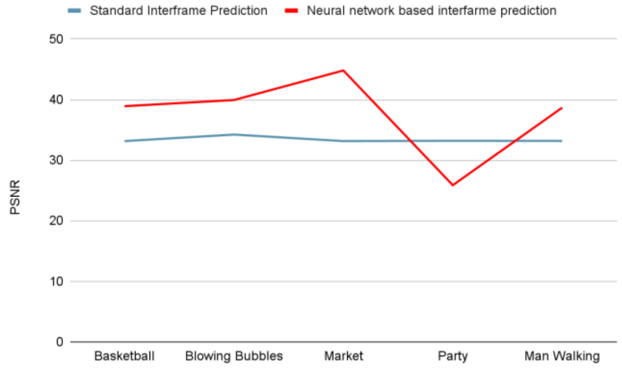

Fig. 7. A graphical representation of PSNR comparison of the two methods

B. SSIM Calculations

SSIM, unlike the PSNR, involves perceptual similarity between two images. It quantifies the visual similarity between the two images [16].

IV. TABLEII
Comparing SSIMs of a standard motion compensation-based interframe prediction with our neural network-based prediction

Image Sequence	SSIM of Motion vector-based inter-frame prediction	SSIM of Neural Network-based inter-frame prediction.
Basketball	97.38%	99.96 %
Blowing bubbles	98.89%	99.99%
Market	99.91%	99.99%
Party	98.59 %	97.63%
Man Walking	99.72%	99.99%

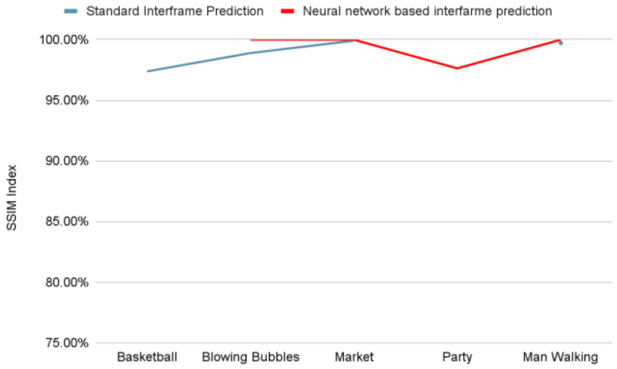

Fig. 8. Comparing SSIMs of a standard motion compensation-based interframe predict

From the above columns, one can find out that the average PSNR is 37.53 dB, and the average SSIM is 99.518%. One can also find an improvement by 11 percent in PSNR and percent in 0.6 percent SSIM.

VIII. CONCLUSION

With the advent of powerful processors, training and implementation of complex neural networks for such information-intensive processes are no longer out of our hands. With a neural network-based implementation, one can see the better accuracy of the inter-frame prediction module when compared to a more primitive algorithmic approach. Further improvements in compression technology will enable us to transmit and store very high-quality video (8K) in much lower spaces than required.

REFERENCES

[1] Gary J. Sullivan, High-Efficiency Video Coding (HEVC), Algorithms and Architectures, (2014).
[2] Kusuma. H.R, Dr.Mahesh Rao, Video Compression Using Spatial and Temporal Redundancy –A Comparative Study
[3] Vishal Deep, Realization of State of the Art Intra Prediction in High-Efficiency Video Coding.
[4] Watson, Andrew. Image Compression Using the Discrete Cosine Transform. Mathematica Journal (1994).

[5] B. S. Nanda and N. Kaulgud, Effect of quantization on video compression, IEEE International Conference on Industrial Technology, IEEE ICIT '02 2 (2002) 764-768

[6] Sze, Vivienne, and DetlevMarpe, Entropy Coding in HEVC, High-Efficiency Video Coding (HEVC) (2014) 209–274.

[7] Zhang, Hao& Ma, Zhan, Fast Intra Prediction for High-Efficiency Video Coding (2012).

[8] Z. Pan, S. Kwong, Y. Zhang, J. Lei, and H. Yuan, Fast Coding Tree Unit depth decision for high-efficiency video coding, 2014 IEEE International Conference on Image Processing (ICIP), (2014).

[9] Alex Shirstinsky, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network.

[10] An introduction to ConvLSTM, available: https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7”

[11] SerkanSulun, Deep Learned Frame prediction for video compression.

[12] Dhanalakshmi, A., Nagarajan, G. Convolutional Neural Network-based deblocking filter for SHVC in H.265. SIViP 14

[13] How to Improve Deep Learning Model Robustness by Adding Noise, available: machinelearningmastery.com

[14] Sandra Aigner and Marco Körner. Future Gen: Anticipating the Future Frames of Video Sequences using Spatio-Temporal 3D Convolutions in Progressively Growing GANs.

[15] Color metric article: https://www.compuphase.com/cmetric.htm

[16] Jim Nilsson, Tomas Akenine-Möller, Understanding SSIM.