Block Based Enhancement using Deep Learning for Conversion of Low Resolution AVS Video to High Resolution HEVC Video

Mehmood Nawaz¹, *Member, IEEE*, Mansoor Ali Teevno³, Rizwan Qureshi², Muhammad waqas⁴, Ali Raza Shahid³ *Member, IEEE*

¹Department of Diagnostic Radiology, The University of Hong Kong, Hong Kong

²National University of Computer and Emerging Sciences, Karachi Campus, Pakistan

³Department of Electrical Engineering, City University of Hong Kong, Hong Kong

⁴Department of Computer Sciences, University College of Zhob, Baluchistan University of IT, Engineering, and Management Sciences, Quetta, Pakistan

Abstract—Conversion of one video bitstream to another video bitstream is a challenging task in the heterogeneous transcoder due to different video formats. In this paper, a region of interest (ROI) based super resolution technique is used to convert the low-resolution AVS (audio video standard) video to high definition HEVC (high efficiency video coding) video. Firstly, we classify a low-resolution video frame into small blocks by using visual characteristics, transform coefficients, and motion vector (MV) of a video. These blocks are further classified as blocks of most interest (BOMI), blocks of less interest (BOLI) and blocks of non-interest (BONI). The BONI blocks are considered as background blocks due to less interest in video and remains unchanged during SR process. Secondly, we apply deep learning based super resolution method on low resolution BOMI, and BOLI blocks to enhance the visual quality. The BOMI and BOLI blocks have high attention due to ROI that include some motion and contrast of the objects. The proposed method saves 20% to 30% computational time and obtained appreciable results as compared with full frame based super resolution method. We have tested our method on different official video sequences with resolution of 1K, 2K, and 4K. Our proposed method has an efficient visual performance in contrast to the full frame-based super resolution method.

Index Terms—Video coding standard, High efficiency video coding, Super-resolution, Block based enhancement

I. INTRODUCTION

A. Background

Nowadays, the ultra-high definition videos have become the future demand in multimedia industry. To achieve this objective, joint collaborative team for video coding (JCT-VC) has introduced high efficiency video coding (HEVC) [1]. On another hand, audio video coding (AVC) standard working group in china has also developed the new audio video coding standard (AVS) to fulfil the requirement of multimedia industry [2], [3]. AVS has similar visual qualities to H.264/AVC with different computational cost. AVS presents a valuable substitute for H.264 within the video entertainment market

in China. Two coding groups are very famous among many videos coding groups due to their effortless contribution in video trans-coding industry. One of them is VCEG (video coding expert group) and second is MPEG (moving picture expert group). HEVC is a valuable effort of these two group teams. Changing the AVS format to the HEVC format is a valuable step in video coding industry. The requirement of HEVC based devices and applications are increasing due to the fast growing internet technology. As per CISCO data traffic prediction, 80% to 90% global internet circulation will contain high definition video contents in future. The fast progress in multimedia contents, the demand for better quality visualization and high-resolution videos is increasing.

Therefore, it is required to cut down large video data via different video compression coding standards like HEVC [17, 19, 20, 21, 22, 24 and 25] and H.264/AVS []. The hi0.gh resolution videos have many applications in different fields, such as healthcare [4], computer vision [5], and wireless multimedia transmissions [6].

The aim of HEVC is to minimize the bitrate up to 50% while giving comparable visual quality as compared to its equivalent video coding standard AVC/H264. HEVC has many new prominent features; one of them is macro block sizes extension. By extending these blocks, we can easily improve the visual quality of low-resolution video. Different types of transcoders have been used to change the format of one bitstream to another bitstream [7]–[9]. The main purpose of transcoding is to fill the gap caused by rapid change in the digital world and the increasing capacity of video contents in different formats such as AVI, VP8, MP4 and VP9, etc. In the past, the video transcoder was limited to convert the video files from one digital device to another digital device but now it has become popular and easy.

Usually two basic types of video transcoders are used in transcoding. One is called homogenous transcoder and the second is called heterogeneous transcoder. In homogenous transcoder, the same format of video bitstream is used to convert one video into another video. For example, low resolution HEVC video bitstream can be converted into high

Mehmood Nawaz is with the Department of Diagnostic Radiology, The University of Hong Kong, Hong Kong. Email: mnawaz-c@my.cityu.edu.hk. Corresponding author: Mehmood Nawaz

Fig. 1: Flow chart of different blocks detection.

resolution HEVC video bitstream and vice versa. In heterogeneous transcoding, the different format of video bitstreams are used. For example, low resolution HEVC video bitstream can be converted into target video bitstream (high resolution AVC/H.264) and vice versa [10], [11]. This digital-to-digital conversion of data is very important for electronic devices, when target devices have supported low capacity video than the original size. Transcoding is needed sometimes for resolution conversion from one coding standard to another coding standard. Transcoding may depend on different devices resolution. For example, tablets or mobile devices do not have enough capacity to display the high definition videos, for this purpose, high spatial videos are down sampled into low spatial videos and vice versa. For spatial downscaling, the discrete cosine transform (DCT) scheme has been shown significantly higher visual quality than the low-pass filtering schemes of pixel domain [12]. However, some applications need to show standard definition television (SDTV) bitstream on high definition mobile devices. This involves the upscaling of the original bitstream by using spatial enhancement techniques [7], [8].

In the proposed algorithm, we used heterogeneous transcoding. A heterogeneous transcoder provides conversion between different standards, for instance, H.264 to MPEG transcoder, H.264 to MPEG-4 transcoder, MPEG-4 to MPEG-2 transcoder, etc. Implementation of this technique has many challenges, because a heterogeneous transcoder needs a syntax conversion unit and may change the picture resolution, picture type, picture rate and directionality of motion vectors (MV) [13]. Due to different encoding format of output sequence, and spatial temporal subsampling, the encoder and decoder motion compensation loops are very complex in heterogeneous transcoder. Quadtree structure size is also a

challenge in heterogeneous transcoding. Quadtree structure in HEVC uses a larger coding unit size (64×64) for estimating the object movement in the video frame, while H.264 / AVC uses (16×16) unit size [14]. While upsampling of low resolution frame into high resolution frame is also a well-known challenge in video transcoder.

II. LITERATURE REVIEW

To overcome such type of problems, many algorithms have been proposed in last two decades [15]–[18]. Huang et al. [19] proposed a method, which adds high frequency information in videos frames. This technique did not perform well when noise ratio increased in the video frame. Chow-Sing et al. [12] proposed a fast intra transcoding from H.264/AVC to high efficiency video coding based on discrete cosine transform (DCT) coefficients and prediction modes. This method uses intra prediction information embedded in H.264 bitstream as well as DCT coefficients to predict the coding depth map for depth limitation. Linwei et al. [14] proposed a machine learning based fast H.264/AVC to HEVC transcoder by exploiting block partition similarity. This method uses a feature selection technique to predict the quadtree coding unit (CU) partition in HEVC. Johan et al. [20] proposed a spatially misaligned HEVC transcoder with computational scalability, which minimizes the complexity of composition process and reduce some information from original bitstream. Mamoon et al. [21] proposed a transparent encryption technique with scalable video communication at lower latency.

This method represents the gains in reducing delay and increasing distortion arising from a transparent encryption. Chia-Hung et al. [1] proposed a coding unit complexity based predictions of coding unit depth and prediction unit mode for efficient HEVC to SHVC transcoding with quality scalability. A survey of video compression is also given in [22].

Assuncao et al. [13] proposed an open loop transcoder to cascade the decoder and encoder directly. It contains no feedback loop in the transcoding architecture for the drift error. The aim of this transcoding is to minimize the complexity and modify the DCT coefficients to reduce the bit rate. Keesman et al. [23] proposed a close loop transcoder, which has the feedback loop in structure for minimizing the transcoding distortion by compensating the drift error in the transcoder. Wee-svsie et al. [8] proposed a spatial domain video transcoder to perform the dynamic bit rate adaption through rate control. Eleftheriads et al. [24] proposed a frequency domain transcoder which used variable length decoding (VLD) and inverse quantization to extract the DCT value of each block at the decoder end. In this technique, the motion compensated residue errors are encoded through re-quantization and variable length coding (VLC) [2], [25]. A scalable omni-directional video coding scheme for virtual reality applications is presented in [26]. Xie et al. [21] proposed a hybrid domain transcoder, which provides the trade-off between the computational cost and video quality. An end-to-end convolutional neural network is proposed for video compression in [27].

The above disused methods have limitations in terms of computational complexity and data visualization. In this paper,

Fig. 2: Example of block partition similarity between H.264/AVC and HEVC method. [14]

the downsample AVS bitstream is classified into many sub-blocks by using visual characteristics, transform coefficients, and motion vector (MV) features [28]. Blocks of most interest (BOMI) have large motions in the video, blocks of less interest (BOLI) have little motion and blocks of non-interest (BONI) are mostly stable in the video. After classification of these blocks, we apply deep learning based super-resolution approach on BOMI and BOLI blocks, which enhances the visual quality of the low resolution blocks. The BONI blocks have no interest in video and they are considered as background blocks. The enhancement of region of interest area in video, saves the transcoding time as well as super resolution complexity as compared to full frame based enhancement. The main contributions of this work are summarized as:

- We proposed a block detection and classification technique that use visual characteristics, transform coefficients, and motion vector (MV) features of a video. This technique classifies all blocks into three types: block of most interest, block of less interest, and block of non-interest.
- A deep learning based enhancement technique is used to convert low resolution to high resolution video. This technique enhances the region of interest area in a video instead of full frame and saves the computational time of the system.
- The multi scale super resolution technique is progressive in both design and training during the upsampling of frames. To make training process smoothly, we used curriculum learning in the proposed framework.
- We have carried out detailed experiments on six different benchmarks to test the efficiency of the proposed method.

The rest of the paper contains the following sections. Section II shows the classification of blocks. The super resolution framework on different blocks is presented in section III. Simulation results are presented in section IV and finally conclusion is discussed in section V.

III. DEEP LEARNING BASED SUPER RESOLUTION FRAMEWORK

A. Classification of Blocks

In fact, humans are more concerned about the area of motion in video content. The sensitivity with respect to eyes of the visual system is very important for classification of blocks in video frame. Mostly richer texture blocks or moving blocks get more attention of the viewers in the video. These blocks are known as the block of most interest (BOMI). The blocks,

which are very near to the block of most interest and have low motion in video, are called blocks of less interest (BOLI). Frequently, background objects have no motion and are less appealing to the viewers, known as blocks of non-interest (BONI). The video quality of BOMI, and BOLI can be well preserved by the super-resolution algorithm. Regions for newly incorporated objects and fast-moving objects or camera motion can be determined by transforming coefficients and motion vectors (MVs). In the video frame, regions of interest (ROI) can be detected using coding parameters.

In AVS, video frames are encoded by macroblocks (MBs) and each macroblock has fixed dimension which is 16×16 . We will evaluate coding information of macroblocks for the classification of each coding tree unit (CTU). Although each macro block of frame is used to estimate either frame is intra or inter. In the blocks detection, each video frame is divided into three types of blocks as shown in Fig. 1. The number of MBs encoded with the intra mode in a frame is used to further decide the intra or inter mode. Let ha_i is used to indicate the i^{th} MB in one frame.

$$a_i = \begin{cases} 0, & \text{intra mode} \\ 1, & \text{inter mode} \end{cases} \quad (1)$$

$$\sum_{i=1}^{16} a_i \geq T, \quad (2)$$

Eq. (2) is used as a criterion for classification of blocks. It also classifies a CTU as an intra/inter coded, where T is used as a threshold, and its optimized value is 8. The motion vectors of current frame are calculated as follows:

$$FMV_{avg}(i) = \frac{1}{N} \sum_{k=1}^N \sum_{l=1}^M B \times MV_{MB_k}(k), \quad (3)$$

where FMV_{avg} represents the average motion vectors of a frame, i is the index of current frame and MB_k presents the k^{th} macro block in the current frame. $BMV_{MB_k}(k)$ represents the magnitude of motion vectors, which is defined as:

$$BMV_{MB_k}(k) = \sqrt{BMVa^2(k) + BMVb^2(k)}, \quad (4)$$

where $BMVa^2(k)$ and $BMVb^2(k)$ are the motion vector (MV) components. These motion vector components are used horizontally and vertically in a frame. The current macro blocks and average motion vectors are calculated as:

$$MBMV_{avg}(i) = \frac{1}{N} \sum_{k=1}^{NF} \sum_{l=1}^M B \times MV_{MB_k}(k), \quad (5)$$

where NF shows the total number of macro blocks and m shows the total number of motion vectors in the frame. The background macro blocks are determined by transform coefficient of each frame value. Therefore, the average value of transform coefficient and current macro block can be estimated by these equations:

$$FTC_{avg}(i) = \frac{1}{N} \sum_{k=1}^N \sum_{l=1}^M B \times l k T C_{MB_k}(k). \quad (6)$$

Fig. 3: Blocks detection and classification: (a) represents the original video sequences of *Cactus*, *Jockey* and *Bosphorus* video, (b) shows the AVS encoding information results: grey regions show skip macro blocks, purple color region shows intera MBs and blue region shows inter MBs, and (c) shows the region of interest: where white color shows the blocks of the most interest, gray color shows the blocks of less interest and black color shows the blocks of non-interest (Best viewed in color).

$$MBTC_{avg}(i) = \frac{1}{N} \sum_{k=1}^{NF} \sum_{l=1}^M B \times lkTC_{MB_k}(k). \quad (7)$$

where FTC_{avg} shows the average motion vectors, i is the index of current frame and MB_k presents the k^{th} macro block in the frame. $BTC_{MB_k}(k)$ shows transform coefficients of k^{th} macro block in the current frame. According to the given parameters, we can easily detect the required blocks, which are shown in Fig. 1.

1) Blocks of most interest (BOMI)

$$\sum_{i=1}^{16} a_i \geq T \parallel (FMV_{avg}(i) > MBMV_{avg}(i)) \quad (8)$$

$$\delta\delta FTC_{avg}(i) \leq MBTC_{avg}(i)$$

2) Blocks of less interest (BOLI)

$$\sum_{i=1}^{16} a_i < T \parallel (FMV_{avg}(i) > MBMV_{avg}(i)) \quad (9)$$

$$\delta\delta FTC_{avg}(i) \leq MBTC_{avg}(i)$$

Or

$$\sum_{i=1}^{16} a_i < T\delta\delta \parallel (FMV_{avg}(i) \leq MBMV_{avg}(i)) \quad (10)$$

$$\delta\delta FTC_{avg}(i) > MBTC_{avg}(i)$$

3) Blocks of non-interest (BONI)

$$\sum_{i=1}^{16} a_i < T\delta\delta \parallel (FMV_{avg}(i) \leq MBMV_{avg}(i)) \quad (11)$$

$$\delta\delta FTC_{avg}(i) \leq MBTC_{avg}(i)$$

These conditions are very useful to separate the region of interest block. The detection and classification of three video sequences, *Cactus*, *Jockey*, and *Bosphorus* is shown in Fig.

Fig. 4: The deep learning based super resolution method architecture where compact compression units (CCUs) are allocated in the lower pyramid level to reduce the memory consumption and improve the reconstruction accuracy of high quality blocks in video frame.

Fig. 5: Flow chart of the proposed video transcoder with deep learning based super resolution.

3. The resolution of each sequence is 1920×1080 . Fig. 3(a) shows the AVS encoding information, which is used to classify the blocks into three gray, purple and black colors. In Fig. 3(c), the white color blocks are region of most interest, gray color blocks are considered as less region of interest, and black colored blocks are mostly stable and have less interest to the viewers, which are known as blocks of non-interest.

IV. SUPER RESOLUTION OF DIFFERENT BLOCKS

We implemented multi scale super-resolution method to enhance the visual quality of low-resolution MBOI, and LBOI blocks of video frame. According to the visual features and motion vectors, the down sample frames of a video are divided into three blocks: MBOI, LBOI and BONI, as described in Section III. In Fig. 3, AVS decoded frame is divided into three types of blocks and then we applied multi-scale SR method to obtain the high quality blocks. The resolution of BONI blocks remains unchanged due to less interest as compared to MBOI and LBOI. The proposed technique mainly focuses on foreground area of the frame which obtained from blocks detection technique as shown in Fig. 3.

A. Multi-Scale Super Resolution

we presented a GAN-SR approach that is progressive in both design and training. This network upsamples an image in intermediate phases, while the learning process is ordered from simple to difficult that is done by curriculum learning. We designed a generative adversarial network (GAN) called Gan-SR that uses the same progressive multi-scale design idea to achieve more credible outcomes. This not only able to measure well against large sampling factors, but also a multi-scale technique that simultaneously raises the quality of the reconstruction of each upsampling factor. This method can work on all image channels (chrominance and luminance, etc.). Here we use luminance channel of each block to increase the spatial resolution of the frame, but other channels remained unchanged.

Suppose we have a set of N low resolution (LR) input frames with corresponding to high resolution target frames $(x_1, y_1), \dots, (x_n, y_n)$. We consider an upscale function $U: X \rightarrow Y$, with X low resolution and Y are high resolution frames. It is difficult to find the parameters that are appropriate for the upscaling function U for high resolution ratios: the bigger the ratio, the more complicated the class of functions is needed. To this purpose, the upscaling function U is a progressive solution. We apply multi-scale super resolution pyramidal network on low resolution frames that are discussed below.

1) *Pyramidal Division*: A pyramidal division model is used which divides a upsampling factor U into a number of simpler functions $(U_0, \dots, U_n)$. The task of each function is to perform the upscaling and the refinement of the features of its own input. A cascade model of compact compression units (CCUs), followed by a sub-pixel convolution layer, is used at each pyramid level. The asymmetrical structure is achieved by assigning of additional CCUs in the lower pyramid level as shown in Fig. 5. The high computational power in lower pyramid decreases the memory consumption and rises the receptive field to the original frame, which exceeds the symmetrical variation in terms of quality and

execution. The upscale function U is divided across the pyramid levels. We additionally utilize two sub scaling-network which are represented by v_s and r_s . These networks translate the individual transformation between scale and standardized feature space of a frame. Figure 5 shows a schematic depiction of our progressive upsample network. This network uses a fixed upsampling of input (x), e.g. bicubic interpolation to make the learning process easy to get a residual as an output.

$$R_s(x) = (r_s \circ u_s \circ \dots \circ u_0 \circ v_s)(x) \quad (12)$$

For a certain scaling factor (s), the estimated HR frame may be calculated as:

$$\bar{Z} = R_s(x) + \phi_s(x) \quad (13)$$

In particular, the Laplacian pyramid [29] concept is not followed in this network. In the Laplacian pyramid based network, an intermediate sub-network outputs are neither supervised nor monitored as basic frame. This network perform well over the Laplacian alternative, which simplifies the back-pass and reduces the optimisation challenge. Moreover, we do not down sample the ground truth for interim supervision. This network is very helpful to prevent artifacts which may occur from sub-sampling.

2) *Compact Compression Units (CCUs)*: A previously proposed DenseNet architecture [30] is used to construct each pyramid level. Similarly to skip connections, we use the compact connections that increase the flow of gradients and reduce disappearance in gradients [31]. A compact compression unit (CCU) is the central component of each level of the pyramid and includes a modified compact-connected block by 1×1 convolution $CONV(1,1)$. The $BN - ReLU - CONV(1,1) - BN - ReLU - CONV(3,3)$ is used in the initial compact layer. We eliminate all batch normalizations in super-resolution due to recent practice in SR methods [9], [24]. However, as the characteristics of prior layers may differ, the first $ReLU$ to rescale $CONV(1,1)$ characteristics can also be removed. This result leads to composition of a dense layer: $CONV(1,1) - RELUCONV(3,3)$.

In Compact Net, the compact connection is broken at the end of each CCU with a compression layer $CONV(1,1)$ that quickly reassembles the information compact connections and increase the efficiency gain in results. A pyramid and local residual connections are used to improve the gradient propagation in this deep model, as illustrated in Fig. 5.

3) *Progressive GAN*: Generative adversarial networks [32] developed as an effective way to improve the perceptiveness of upsampled frames in SR. Training GANs is notoriously hard and the success of using GANs to SR is sparse to a single sample in very small targets. The proposed transcoder network is based on multi-scale SR network which is similar to the generator adversarial network suggested in the second part of Figure 5. The reverse pyramid structure u_2, u_1, u_0 are indicated in the second part of Fig. 6, where each level diminishes the input frame spatially using *AvgPOOLING* progressively.

A scale-specific frame transformation layers v_s is used before each pyramid that is similar to the generator. The network is entirely converted and produces a tiny patch of features

Fig. 6: The top part shows the blending procedure for the generator and bottom part shows the blending process for discriminator in curriculum training.

similar to PatchGAN [33] to handle multifaceted outputs from the generator. The discriminator works on residual of bicubic up sampled frame and the original frame is similar to the generator network. This permits both the generator and the discriminator to solely focus on the major variation source which is a upsampling procedure. This may also be seen as the subtraction from the discriminator of a database that contributes to a reduction of variance. We employ the less square loss rather than the original cross-entropy loss as our training target. The discriminator and the generator loss for the training of scales (s) may be described by means of the anticipated residual and real residual, which represented as $\tilde{r}$ and r :

$$\mathcal{L}_{D_s}^i = (D(\tilde{r}_i^s))^2 + (D(r_i^s) - 1)^2, \quad (14)$$

$$\mathcal{L}_{R_s}^i = (D(\tilde{r}_i^s) - 1)^2 + \sum_{k \in (2,4)} \|p_k(\bar{Z}_i) - p_k(Z_i)\|^2, \quad (15)$$

where p_k shows the k^{th} pooling layer of VGG-16 model [12].

4) *Curriculum Learning*: Curriculum learning [34] is a training approach which increases the complexity of the learning task progressively. It is widely used to forecast sequence and to make decision sequentially, where huge speeds may be achieved in the training period and an improvement in performance. The pyramidal division of U makes it possible for us to naturally implement the curriculum learning. The loss for a scalable exemplar (x_i^s, z_i) can be set to:

$$\mathcal{L}_{R_s}^i = \|R_s(x_i^s) + v_s(x_i^s) - z_i\|_1, \quad (16)$$

where x_i^s corresponds to $s \times$ downsampled version of z_i and the goal at scale s is calculated as:

$$\theta_s = \underset{\theta_s}{\operatorname{argmin}} \sum_{s' \leq s} \sum_i \mathcal{L}_{R_{s'}}^i, \quad (17)$$

where θ_s parameterises all functions in and below the current scale ($u_0, v_0, r_0, \dots, u_s, v_s, r_s$) according to the pyramid network that is described in Fig. 5.

Our method merely starts with a $2\times$ network instruction. As we enter a new phase (e.g. to $4\times$), a new level of the pyramid is progressively included to decrease its influence on previously learned levels as shown in Fig. 6. The predicted $\tilde{r}_s$ of the generator at scale s is a linear combination of level s and $s-1$ output. For the discriminator, the new pyramid's output features are combined with the scale-specific input of the preceding level v_{scale1} before entering the trained pyramids $\{u_{s-1}, \dots, u_0\}$.

Bilinear interpolation and AvgPOOL are employed before merging to match spatial dimensions. In both situations, β determines the effect of the new pyramid, which changes from 0 to 1 during the mixing process. As a consequence, we add training pairs of the next scale that scale gradually. Finally, to construct the batches we have chosen randomly one of the scales (s) to prevent batch statistics from being mixed as recommended in [2].

This progressive training strategy considerably reduces overall training time in comparison to basic multi-scale training when training examples from multiple scales are provided to the network at the same time. In comparison to single scale and basic multi-scale training, it offers a further efficiency improvement on all included scales and relieves instabilities in GAN training.

V. EXPERIMENTS

The proposed SR technique is implemented by using Python and MATLAB 2018a. All simulations are performed at Intel (R) Core™ i5-CPU 4590 @ 3.3GHZ with 8 GB RAM with a spatial resolution of 3840×2160 for performance evaluation. All frames are encoded with the low delay-P-main configuration file. The performance of proposed method is evaluated based on image quality (Δ PSNR), bit rate (Δ Bit-rate), and computational complexity (Δ T). The parameters are calculated using the equations given below:

$$\Delta PSNR = PSNR_{(proposed)} - PSNR_{(anchor)} \quad (18)$$

$$\Delta Bitrate = \frac{Bitrate_{(proposed)} - Bitrate_{(anchor)}}{Bitrate_{(anchor)}} \times 100\% \quad (19)$$

$$\Delta T = \frac{T_{(proposed)} - T_{(anchor)}}{T_{(anchor)}} \times 100\% \quad (20)$$

where $PSNR_{(proposed)}$, $Bitrate_{(proposed)}$ and $T_{(proposed)}$ show the proposed algorithm PSNR, bitrate and encoding time, respectively and $PSNR_{(anchor)}$, $Bitrate_{(anchor)}$ and $T_{(anchor)}$ represent the PSNR, bitrate and encoding time of anchor software HM10.1, respectively.

To evaluate the performance of proposed method, we used six different classes of videos. Each video belongs to a specific class. For example, Class A has *PeopleOnStreet* and

Traffic video, Class B has *Cactus* and *Kimono* video, Class C has *BasketBallDrill* and *PartyScene* video, class-D has *BlowingBubbles* and *BQSquare* video, Class-E has *FourPeople* and *KristenAndSara* video, and Class-F has *ChinaSpeed* and *SlideEditing* video. Experimental results of block-based super-resolution method and frame based method are shown in Table I and Table II. The difference between the frame based and block based results is shown in Table III. *HoneyBee* up-sampling results show that our proposed method saves 37% time compared to the frame based method with similar picture quality PSNR (dB) and SSIM (index). In block based, *Bosphorus* saves 26% of the time during 1K to 4K up sampling. In 1K to 2K upsampling, *Kimono* takes 00.006% more SR time. The experimental results show that the proposed method has achieved good results in 2K and 4K video with low computational time. The negative sign in the Table-III indicates a decrease, and the positive sign indicates an increase in value. In Table III, the *HoneyBee* PSNR drops by 0.0091 dB compared to the frame based value and the *Cactus* PSNR decreases by 0.0022 dB over the frame based result. As to compare bitrate results, *BQTerrace* obtains a maximum bit rate of 1.77% over the frame based value, and the *Bosphorus* achieves a minimum bit rate reduction of 0.745% based on the frame based value. Fig. 7 shows the HEVC encoding time analysis for six different videos with different resolutions. In Fig. 7, it can be seen that *Bosphorus* video consumes more HEVC encoding time in 2K and 1K resolutions.

Fig. 7(a) shows the HEVC encoding time of six different videos. The experiments show that the all block based videos have high HEVC encoding time. On average, the block based results take 15.98% more HEVC encoding time as compared to frame based. The average difference between block based and frames based in *PeopleOnStreet* video frame is -45.80 seconds, which indicates that frame based transcoder has a high computational cost. High definition videos are more time consuming, due to high pixel density. Fig. 7(b) shows the super-resolution complexity between frame-based and block-based SR results. The SR time varies from video to video due to different resolution. Different types of up-sampling is also the part of experiments. For example, 1K low resolution (1280×800) images of *PeopleOnStreet*, and *Traffic* are up-sampled to 2K resolution (2560×1600) by using proposed SR method same as the low resolution (960×544) images of *Cactus*, and *Kimono* are up-sampled to 1K high resolution (1920×1080) frames using same SR technique.

Fig. 7(c) illustrates the visual quality of frame-based and block-based SR approaches. Here, we will contemplate the results of six different videos, which are used in the experiment. In most cases, the maximum signal to noise ratio (PSNR) of both frame based and block based videos is similar. The visual quality difference raises gradually in low resolution videos. In 4K videos, the visual quality of both block based and frame based videos is mostly comparable. During up-sampling 1K to 2K or 2K to 4K the number of pixels density decreases, which leads to decrease in the visual quality of all video frames. Fig. 7(d) shows a comparison of bitrates between frame-based and block-based transcoding methods. The bitrate is the size of video file per second of data. It is

TABLE I: Block Based Result of Proposed Method

Video Sequence	Initial Resolution	Final Resolution	PSNR (dB)	SSIM (index)	SR Time(sec)	Bitrate (kb)	HEVC Encoding Time
HoneyBee	1920×1080	3840×2160	38.7043	0.9345	4171.90	1665.63	7157.30
Bosphorus			39.2649	0.9438	4790.18	2690.54	10778.62
Jockey			39.2649	0.9438	4790.18	2690.54	10778.62
HoneyBee	960×540	3840×2160	37.4774	0.9294	1548.80	1593.00	7111.07
Bosphorus			37.7081	0.9492	1017.67	3767.36	11492.01
Jockey			37.9340	0.9383	1137.79	3756.54	12360.79
Cactus	960×540	1920×1080	32.8406	0.8801	1228.84	3831.21	8049.17
kimono			37.8653	0.9469	1313.90	3021.42	8802.40
BQTerrace			29.4451	0.8695	1020.25	7271.20	4791.19

TABLE II: Frame Based Result of Proposed Method

Video Sequence	Initial Resolution	Final Resolution	PSNR (dB)	SSIM (index)	SR Time(sec)	Bitrate (kb)	HEVC Encoding Time
HoneyBee	1920×1080	3840×2160	38.7134	0.9346	7456.26	1659.34	10130.64
Bosphorus			39.6489	0.9593	4850.50	3202.68	14652.55
Jockey			39.2736	0.9436	5520.03	2675.47	9324.05
HoneyBee	960×540	3840×2160	37.4784	0.9293	1843.21	1600.73	11288.62
Bosphorus			37.7084	0.9492	1312.07	3795.72	15616.37
Jockey			37.9478	0.9382	1624.99	3726.01	10924.37
Cactus	960×540	1920×1080	32.8428	0.8800	1565.66	3880.83	4500.52
kimono			37.8657	0.9471	1860.14	3018.84	4553.62
BQTerrace			29.4715	0.8699	1361.43	7142.20	4783.99

TABLE III: Difference between Block Based and Frame Based Result

Video Sequence	Initial Resolution	Final Resolution	PSNR (dB)	SR Time(sec) %	Bitrate (kb)%	HEVC Encoding Time (sec)%
HoneyBee	1920×1080	3840×2160	-0.0091	-44.04	0.3790	-29.3499
Bosphorus			-0.0079	-24.75	0.2785	-24.3625
Jockey			-0.0087	-13.22	0.5601	15.6001
HoneyBee	960×540	3840×2160	-0.0010	-15.97	-0.4829	-37.0067
Bosphorus			-0.0030	-22.43	-0.7471	-26.4104
Jockey			-0.0138	-29.98	0.8127	13.1487
Cactus	960×540	1920×1080	3-0.0022	-21.51	-1.2780	35.4865
kimono			-0.0040	-29.36	0.0854	00.0015
BQTerrace			-0.0264	-25.06	1.7741	42.4878

TABLE IV: Image quality (PSNR) and Bitrate (kb) comparison among proposed method and other methods

Video Sequence	TMM [35] PSNR(dB)	TMM [35] Bitrate (kb)	CSVt [36] PSNR(dB)	CSVt [36] Bitrate (kb)	Proposed PSNR(dB)	Proposed Bitrate (kb)
HoneyBee	—	—	—	—	-0.0091	0.3790
Bosphorus	—	—	—	—	-0.0079	0.2785
Jockey	-0.1180	2.646	-0.0115	2.4980	-0.0087	0.5601
HoneyBee	—	—	—	—	-0.0010	-0.4829
Bosphorus	—	—	—	—	-0.0030	-0.7471
Jockey	-0.0790	1.5180	-0.1450	2.8940	-0.0138	0.8127
Cactus	-0.1250	3.0780	-0.0660	1.6430	-0.0022	-1.2780
kimono	-0.024	0.5520	-0.0570	1.3080	-0.0040	0.0854
BQTerrace	-0.1260	3.4610	-0.1130	3.1000	-0.0264	1.7741

usually represented by kilobits (kbs) or megabits per second (mbs). The bitrate in video streaming quality is a vital factor. Streaming media codec uses “lossy” compression to compress the video file. The high compression rate on video affects the video quality in transcoder. Overall, the results of proposed method are efficient and reliable in terms of SR time, bitrate, HEVC encoding time and PSNR. The visual efficiency of ROI based super resolution and full frame based super resolution is shown in Fig. 8.

VI. CONCLUSION

This paper introduces a novel approach, where a deep learning based enhancement technique is used to convert low

resolution to high resolution video. This technique enhances the region of interest area in a video instead of full frame and reduces the 20% to 30% SR complexity of a system. The results validate the efficiency of this approach. The proposed block based transcoder performs better than frame based super-resolution method with comparable visual quality in PSNR (dB) and SSIM (index).

In future, we can extend this transcoder for other video formats like H.264 to MPEG transcoder, H.264 to MPEG-4 transcoder, MPEG-4 to MPEG-2 transcoder, etc. This field is growing rapidly and continuous effort will bring significant improvement in video compression and quality.

Fig. 7: (a) shows the ROI based and full frame-based HEVC-encoding Time (sec), (b) shows the ROI based and full frame-based SR Time (sec), (c) shows the ROI based and full frame-based PSNR (dB), and (d) shows the ROI based and full frame-based Bitrate (kb).

REFERENCES

- [1] C.-H. Yeh, W.-Y. Tseng, L.-W. Kang, C.-W. Lee, K. Muchtar, and M.-J. Chen, "Coding unit complexity-based predictions of coding unit depth and prediction unit mode for efficient hevco-to-shvc transcoding with quality scalability," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 342–351, 2018.
- [2] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video transcoding: an overview of various techniques and research issues," *IEEE Transactions on multimedia*, vol. 7, no. 5, pp. 793–804, 2005.
- [3] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal processing magazine*, vol. 20, no. 2, pp. 18–29, 2003.
- [4] G. Abbas, M. J. Khan, R. Qureshi, and K. Khurshid, "Scope of video magnification in human pulse rate estimation," in *2017 International Conference on Machine Vision and Information Technology (CMVIT)*, pp. 69–75, IEEE, 2017.
- [5] A. R. Shahid, S. Khan, and H. Yan, "Contour and region harmonic features for sub-local facial expression recognition," *Journal of Visual Communication and Image Representation*, vol. 73, p. 102949, 2020.
- [6] H. Hassan, M. N. Khan, S. O. Gilani, M. Jamil, H. Maqbool, A. W. Malik, and I. Ahmad, "H. 264 encoder parameter optimization for encoded wireless multimedia transmissions," *IEEE Access*, vol. 6, pp. 22046–22053, 2018.
- [7] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "Hvc complexity and implementation analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, 2012.
- [8] S. J. Wee, J. G. Apostolopoulos, and N. Feamster, "Field-to-frame transcoding with spatial and temporal downsampling," in *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 4, pp. 271–275, IEEE, 1999.
- [9] M. Pantoja and N. Ling, "Transcoding with resolution conversion using super-resolution and irregular sampling," *Journal of Signal Processing Systems*, vol. 60, no. 3, pp. 305–313, 2010.
- [10] D. Zhang, B. Li, J. Xu, and H. Li, "Fast transcoding from h. 264 avc to high efficiency video coding," in *2012 IEEE International Conference on Multimedia and Expo*, pp. 651–656, IEEE, 2012.
- [11] E. Peixoto, T. Shanableh, and E. Izquierdo, "H. 264/avc to hevco video transcoder based on dynamic thresholding and content modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 1, pp. 99–112, 2013.
- [12] C.-S. Lin, W.-j. Yang, and C.-W. Su, "Fitd: Fast intra transcoding from h. 264/avc to high efficiency video coding based on dct coefficients and prediction modes," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 130–140, 2016.
- [13] P. Assuncao and M. Ghanbari, "Transcoding of single-layer mpeg video into lower rates," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 144, no. 6, pp. 377–383, 1997.
- [14] L. Zhu, Y. Zhang, N. Li, G. Jiang, and S. Kwong, "Machine learning based fast h. 264/avc to hevco transcoding exploiting block partition similarity," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 824–837, 2016.
- [15] L. Pengyu and J. Kebin, "Fast extraction method for video motion region of interest combining visual characteristics," in *2011 International Conference on Computer Science and Service System (CSSS)*, pp. 3939–3944, IEEE, 2011.
- [16] S.-F. Huang, M.-J. Chen, and M.-S. Li, "Region-of-interest segmentation based on bayesian theorem for h. 264 video transcoding," in *2011 Visual Communications and Image Processing (VCIP)*, pp. 1–4, IEEE, 2011.
- [17] P. Xing, Y. Tian, X. Zhang, Y. Wang, and T. Huang, "A coding unit classification based avc-to-hevc transcoding with background modeling for surveillance videos," in *2013 Visual Communications and Image Processing (VCIP)*, pp. 1–6, IEEE, 2013.
- [18] N. Bjork and C. Christopoulos, "Transcoder architectures for video coding," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 1, pp. 88–98, 1998.
- [19] X. Huang, J. Søgaard, and S. Forchhammer, "No-reference pixel based video quality assessment for hevco decoded video," *Journal of Visual Communication and Image Representation*, vol. 43, pp. 173–184, 2017.
- [20] J. De Praeter, G. Van Wallendael, T. Vermeir, J. Slowack, and P. Lambert, "Spatially misaligned hevco transcoding with computational-complexity scalability," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 149–158, 2016.
- [21] R. Xie, J. Liu, and X. Wang, "Efficient mpeg-2 to mpeg-4 compressed video transcoding," in *Visual Communications and Image Processing 2002*, vol. 4671, pp. 192–201, International Society for Optics and Photonics, 2002.
- [22] O. Stankiewicz, K. Wegner, and M. Domański, "Study of 3d video compression using nonlinear depth representation," *IEEE Access*, vol. 7, pp. 31110–31122, 2019.
- [23] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman,

- “Transcoding of mpeg bitstreams,” *Signal processing: Image communication*, vol. 8, no. 6, pp. 481–500, 1996.
- [24] A. Eleftheriadis and D. Anastassiou, *Dynamic rate shaping of compressed digital video*. PhD thesis, Columbia University New York, NY, 1995.
- [25] W. Jiang, Y. Chen, and X. Tian, “Fast transcoding from h. 264 to hevcc based on region feature analysis,” *Multimedia tools and applications*, vol. 73, no. 3, pp. 2179–2200, 2014.
- [26] D. Liu, P. An, R. Ma, W. Zhan, and L. Ai, “Scalable omnidirectional video coding for real-time virtual reality applications,” *IEEE Access*, vol. 6, pp. 56323–56332, 2018.
- [27] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, “Dvc: An end-to-end deep video compression framework,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11006–11015, 2019.
- [28] R. Luo, R. Xie, and L. Zhang, “Fast avs to hevcc transcoding based on roi detection using visual characteristics,” in *2014 IEEE International*

- Symposium on Broadband Multimedia Systems and Broadcasting*, pp. 1–6, IEEE, 2014.
- [29] P. J. Burt and E. H. Adelson, “The laplacian pyramid as a compact image code,” in *Readings in computer vision*, pp. 671–679, Elsevier, 1987.
- [30] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, “Densenet: Implementing efficient convnet descriptor pyramids,” *arXiv preprint arXiv:1404.1869*, 2014.
- [31] X. Zhang, “Melanoma segmentation based on deep learning,” *Computer assisted surgery*, vol. 22, no. suppl, pp. 267–277, 2017.
- [32] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” *arXiv preprint arXiv:1606.03498*, 2016.
- [33] U. Demir and G. Unal, “Patch-based image inpainting with generative adversarial networks,” *arXiv preprint arXiv:1803.07422*, 2018.
- [34] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Fig. 8: shows the qualitative results of 1K, 2K, and 4K videos. (a) shows the results of 4K *Bosphorus*, *Jockey*, *HoneyBee* and *ShakeNDry* video. There is no any significant difference between picture quality of both frame based and block based method, (b) shows the results of class A videos. In this class, the visual quality of *PeopleOnStreet* and *Traffic* videos is similar, (c) illustrates the results of class B videos (*BasketballDrive*, *BQTerrace*, *Cactus*, and *Kimono*), (d) shows the visual quality of *PartyScene* and *RaceHorses* videos, which belong to class C, (e) shows the visual quality of *BlowingBubbles* and *BQSquare* video of class D, (f) shows the visual quality of *FourPeople* and *KristenAndSara* videos, which belong to class E, and (g) shows the results of *ChinaSpeed* and *SlideEditing* video of class F.

- [35] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An effective cu size decision method for hevc encoders," *IEEE transactions on multimedia*, vol. 15, no. 2, pp. 465–470, 2012.
- [36] E. Peixoto, B. Macchiavello, E. M. Hung, and R. L. de Queiroz, "A fast hevc transcoder based on content modeling and early termination," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 2532–2536, IEEE, 2014.

Mehmood Nawaz received the PhD degree from City University of Hong Kong and MSc Degree from Shanghai Jiaotong University, China. He is currently a Postdoctoral fellow in the Department of Diagnostic Radiology, University of Hong Kong. His research interests include graph matching, image segmentation, shape matching, and pattern recognition.

MANSOOR ALI TEEVNO received his BEng in electronic engineering and Master of Engineering in Electronic System Engineering from Mehran University of Engineering Technology, Jamshoro Pakistan in 2012 and 2017 respectively. Currently, he is working towards his PhD degree in electrical engineering at City University of Hong Kong. Before joining City University, he was working as lecturer in electronic engineering department at Mehran University of Engineering Technology, Jamshoro Pakistan. His research interests include signal and

image processing, machine learning and Embedded Systems.

Rizwan Qureshi is a professor of Computer Science, at School of Computing, National University of Computer and Emerging Sciences, Karachi Campus, Pakistan. He received his PhD from the department of Electrical Engineering, City University of Hong Kong, Hong Kong in 2021. Before joining City University, he was a lecturer in electrical engineering department at COMSATS University Islamabad, Wah Campus, Pakistan. His research interests include Bioinformatics, Signal and Image Processing, Machine Learning, Energy Management

and Spectral Imaging.

Muhammad Waqas is currently working towards his PhD degree in Computer Science at National University of Computer and Emerging Sciences, Karachi, Pakistan. He is also a lecturer at the department of Computer Sciences, University College of Zhob, Baluchistan University of Information Technology, Engineering, and Management Sciences, Quetta, Pakistan. His research interests include multiple instance learning, image processing, and computer vision.

Ali Raza Shahid (Graduate student member, IEEE) received an MSc degree in electrical engineering from COMSATS University Islamabad, Pakistan, in 2015. He is currently pursuing the Ph.D. degree in Electrical Engineering at City University of Hong Kong. He also served as lecturer in COMSATS University Islamabad, Pakistan. His research interests include magnetic resonance imaging and computer vision.