Object Tracking Based Surgical Incision Region Encoding using Scalable High Efficiency Video Coding for Surgical Telementoring Applications

Karthik Sairam SANAGAVARAPU, Muralidhar PULLAKANDAM

Dept. of ECE, NIT Warangal, India

karthik_sai_ram@yahoo.com, pmurali@nitw.ac.in

Submitted September 30, 2021 / Accepted April 19, 2022

Abstract. *Surgical telementoring is an advanced telemedicine concept where the expert surgeon guides the onsite novice present at the remote location. The efficient telementoring system requires the wireless transmission of high-quality surgical video with less bitrate in less time. The bit rate of the surgical video can be decreased by segmenting the surgical incision region and removing the background region. The High Efficiency Video Coding (HEVC) standard has provided promising results for surgical telementoring applications. But the Rate-Distortion Optimization (RDO) search process in HEVC increases the complexity that in turn increases the encoding time. We propose the method which involves the segmentation of the surgical incision region using the Kernelized Correlation Filter (KCF) object tracking technique. The segmented region is encoded by the complexity-efficient Scalable HEVC (SHVC) to meet the resolution of an end-user device. The complexity of SHVC is decreased by using the Convolutional Neural Network (CNN) and Long- and Short- Term Memory (LSTM) to predict the Coding Tree Unit (CTU) structure. The results show that the proposed method decreases the bitrate significantly for segmented surgical video sequences without degradation in Peak Signal-to-Noise Ratio (PSNR). These results are obtained for the surgical video sequences with slow-moving objects. Furthermore, the CNN+LSTM approach reduces the encoding time of standard SHVC by 51% with negligible Rate-Distortion (RD) performance loss.*

Keywords

Surgical telementoring, object tracking, KCF tracker, region of interest, High Efficiency Video Coding

1. Introduction

Surgical telementoring has acquired heaps of interest, particularly in rural areas. The problems that arise during the surgical procedures are complex for the inexperienced surgeon to handle. Telementoring is the process of transferring the knowledge from the experienced surgeon to the novice who is present at a distant location. However, the limited bandwidth resources present in the remote areas make the telementoring system difficult to implement. The efficient telementoring system requires transmission of Region of Interest (ROI) of video with high quality in a limited bandwidth. In this paper, the surgical incision region is referred to as ROI. The High Efficiency Video Coding (HEVC) [1], [2] helps to compress the video with 50% less bitrate compared to the H.264 Advanced Video Coding (AVC) [3], [4] standard without loss in video quality [5]. Approximately 5 Mbps bandwidth is required to transmit the high-quality videos for telementoring applications which is very difficult to achieve in disaster-affected areas. The Scalable extension of HEVC (SHVC) [6], [7] can be used in such a case that provides highly scalable coding efficiency and allows the transmission of a single video with different resolutions in a single bitstream. However, the complexity of SHVC makes it unsuitable for real-time applications. The complexity is increased mainly due to the Rate-Distortion Optimization (RDO) search process.

The RDO search process is shown in Fig. 1. In SHVC, each frame is divided into Coding Tree Units (CTUs). The CTU can be recursively divided into Coding Units (CUs) till it reaches the smallest size. The maximum size of the CU is 64×64, and the smallest size is 8×8. During the RDO process, each parent CU is subdivided into four child CUs based on the condition in (1).

$$split = \begin{cases} 1, & \text{if} \quad J^U \geq \sum_{i=1}^{4} J^{U_i} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where U is the parent CU, U_i represents the child CUs and $i \in \{1, 2, 3, 4\}$. If $split = 1$, the parent CU is divided into four child CUs which are of the same size. Otherwise, the parent CU remains the same. The RDO search process involves the Rate-Distortion (RD) cost J calculation starting at the top and successively moving towards the RDO tree's bottom. However, the checking process using (1) is performed in reverse order (bottom to top). Total 85 CUs need to be checked in each CTU, which significantly increases the encoding time.

DOI: 10.13164/re.2022.0231

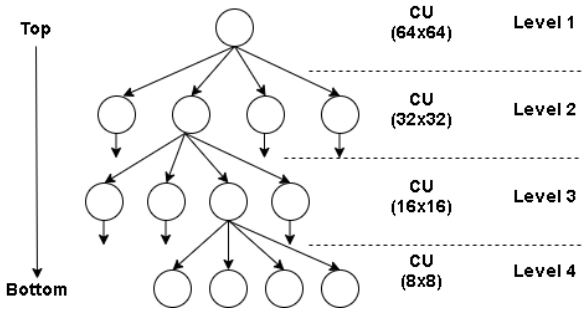


Fig. 1. Conventional RDO search process to determine the CU size.

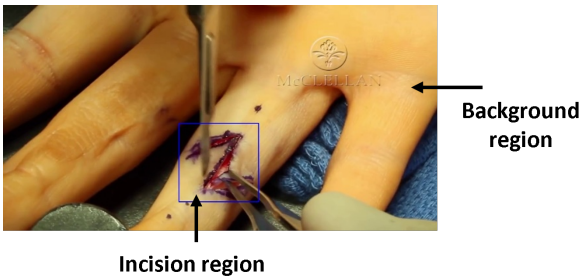


Fig. 2. Surgical video frame with surgical incision and background region.

SHVC consists of one Base Layer (BL) [8] and more than one Enhancement Layer (EL). The BL acts as a single layer HEVC with the lowest quality. The EL is coded using the reference of lower layers and provides better quality compared to BL. SHVC makes use of Inter-Layer Texture Prediction (ILTP) [9] and Inter-Layer Motion Prediction (ILMP) to exploit the correlation between the motion vector and pixel values of non-identical layers.

The surgical video frame is shown in Fig. 2 consists of the background region and the surgical incision region. The surgical telementoring system requires the transmission of the surgical incision region with high quality. Several segmentation techniques can be used to identify the incision region in the video. The Mean Shift (MST) [10] algorithm is used in the medical image field to identify the surgical region. However, the machine learning techniques are incapable of processing the large image data, which results in the segmentation of the ROI with less accuracy. The Convolutional Neural Network (CNN) helps segment the ROI in computer vision applications; however, it increases the computational complexity.

The above techniques help to encode the ROI with high quality and the background region with low quality. As the background region does not contain any valuable information, the pixel values of the background region can be made zeros, resulting in the decrease of bitrate. The ROI region in the surgical videos moves slowly, and less abrupt changes can be observed throughout the video sequence. Hence, the object tracking technique can be used to track and extract the ROI in the surgical video.

The main contributions of this paper are as follows:

1. The Kernelized Correlation Filter (KCF) object tracking technique is used to track the ROI in the surgical video sequence.
2. A large database with 397 video sequences is created to train the CNN.
3. The Long- and Short-Term Memory (LSTM) network in combination with CNN is designed to reduce the complexity of SHVC.

The proposed method extracts the ROI from the surgical video frames using the KCF object tracker and encodes the ROI using SHVC with less complexity using the deep learning CNN+LSTM technique. The rest of the paper is presented as follows. Section 2 explains the background work, and Section 3 discusses the KCF object tracking technique to extract the ROI and deep CNN+LSTM network to reduce the complexity of SHVC. Section 4 analyzes the experimental results, and Section 5 concludes the work.

2. Background Work

The surgical telementoring system requires the surgical incision region to be encoded with high quality. Several authors suggested different algorithms to code the Region of Interest (ROI) in a video with high quality and the remaining region with low quality. The authors in [11] extracted the facial features using MST algorithm and encoded them with high quality. The background region is encoded in lower quality. In [12], the authors used the 3D morphological technique to segment the ROI region in colon computed tomography (CT). The researchers in [13] used the H.264 encoder to encode the segmented part of echocardiogram and CT video sequences. The segmentation is done using the image processing techniques like squared gradient, Sobel operators, and thresholding techniques. The Nearest Neighbor (NN) classifier is used in [14] to extract the ROI region in ultrasound videos. The results show that the bitrate is reduced by an average of 13.52% at the cost of high computational complexity.

In [15], a new method is proposed that allows the manual selection of the desired region in the video and encodes the selected area with high quality for surgical telementoring application. In [16], the authors designed a method that adaptively sets the ROI location and resolution based on the predefined settings. The desired ROI location is obtained by removing the background region, and then the inter-layer prediction operation is performed on the selected ROI region. This method saves the bitrate by 33.48%. The kernel-based MST method is used in [17] that requires the user interaction to select the desired ROI and the related resolution. The authors encode the selected ROI using the Huffman encoding technique. The authors in [18] use the non-parametric

segmentation to detect the surgical incision region by considering the physiological behavior of the visual system and encodes the ROI with high quality. The authors in [19] developed a method for surgical telementoring application that performs image compression, image denoising, and image segmentation operations on computed tomography images for the diagnosis of congenital heart disease. The researchers in [20] reports the augmented reality system that uses the 3D tracking module and the Microsoft HoloLens for training and the telementoring surgery. The authors in [21] developed the deep CNN, which is SegNet that uses 26 convolutional layers for image segmentation. This method is computationally expensive.

The authors in [22] use the probability of the human attention over the frames to allocate coding bits using the visual saliency map scheme. The experimental findings indicate 43% saving in encoding time and 23% reduction in bitrate. In [23], the smartphones are used to capture the wound image. The wound part is segmented using the mean shift algorithm, and the red-yellow-black color model analyzes the wound. Similarly, the authors in [24], and [25] use the mean shift algorithm to detect the boundary of the foot injury and for the classification of skin tissue. In [26], the authors use the CNN approach to segment and analyze the wound region. The researchers in [27] proposed the CNN method that uses the convolutions for the extraction of multiple-level features for Diabetic Foot Ulcer (DFU) classification.

The CNN technique efficiently separates the surgical incision region from the background region. However, high computational complexity makes them less suitable for real-time applications. The background region doesn't contain vital information. The encoding of the background region increases the bit rate. Some of the authors use the HEVC encoder to encode the surgical videos with high quality. But the RDO search process in the HEVC increases the complexity that in turn increases the encoding time. We proposed the efficient surgical telementoring system that encodes the ROI with high quality in less time using (CNN+LSTM) for real-time performance.

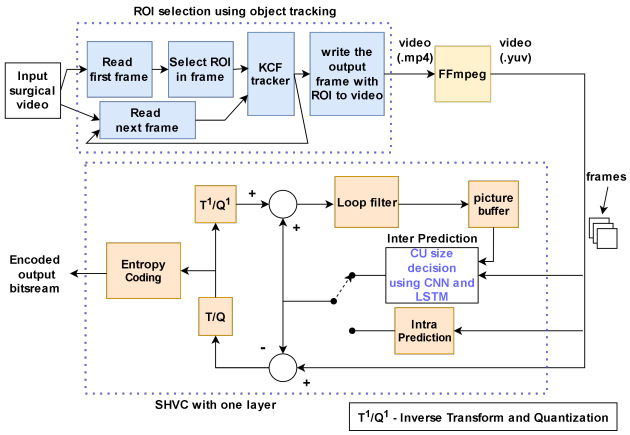


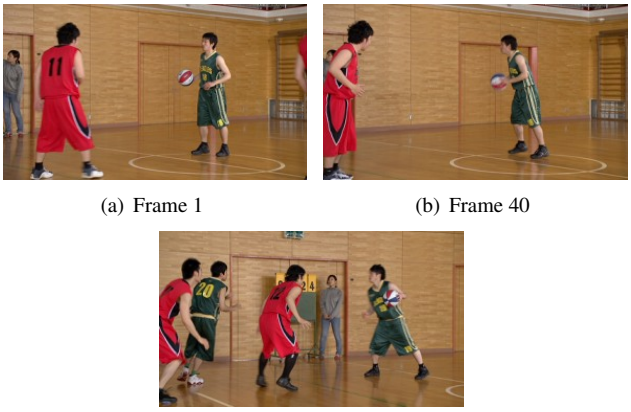
Fig. 3. Framework of the proposed method for surgical telementoring application.

3. Proposed Method

In this section, we first analyze the correlation between the frames for surgical and general video sequences. Then, the object tracking using the KCF tracker is used to detect the surgical incision region. Finally, the CNN and LSTM are trained using the database (refer to Sec. 4) to encode the surgical incision region with less complexity. The framework of the proposed method is shown in Fig. 3. The operation of each block in the proposed method is explained in the following subsections.

3.1 Analysis of Correlation between Frames

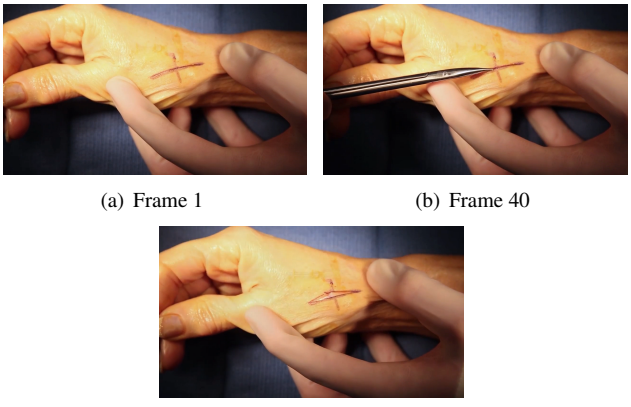
This section analyzes the correlation between the frames for the surgical and general video sequences. The surgical video sequence "NuGrip Arthroplasty" and the general video sequence "BasketballPass" with frames at a different distance are shown in Fig. 5 and Fig. 4. The frames in the surgical video show that the surgical incision region movement is very small throughout the sequence.



(a) Frame 1 (b) Frame 40

(c) Frame 200

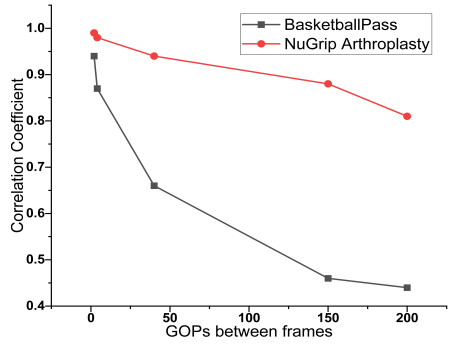
Fig. 4. Example frames of the fast motion Basketball video sequence.



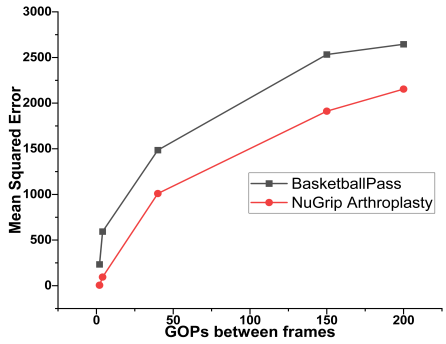
(a) Frame 1 (b) Frame 40

(c) Frame 200

Fig. 5. Example frames of the surgical NuGrip Arthroplasty video sequence.



(a) Correlation coefficient



(b) Mean squared error

Fig. 6. Correlation and mean squared error curves of general and surgical video sequence at different distance between the frames. Note that the BasketballPass represents the general video sequence and NuGrip Arthroplasty represents the surgical video sequence.

The conventional object tracking technique is sufficient to track and extract the surgical region. However, the objects in the general video sequence move rapidly, which requires deep learning segmentation techniques to detect the boundary of ROI. The deep learning techniques segment the ROI more accurately at the cost of high computational complexity. Figure 6 shows the correlation coefficient and Mean Squared Error (MSE) curves of BasketballPass and NuGrip Arthroplasty video sequences. The figures show that the correlation coefficient is high for NuGrip Arthroplasty compared to the BasketballPass sequence. The high correlation coefficient is observed due to the small movement of ROI in a surgical video sequence with the background region remaining almost constant. The correlation coefficient and MSE are inversely proportional to each other. The NUGrip Arthoplasty produces less MSE due to the high similarity between the frames.

3.2 ROI Tracking using KCF Tracker

The analysis in Sec. 3.1 shows that the object movement is very small, and the background region remains almost constant throughout the sequence. Hence, the object tracking techniques can be used to track the ROI effectively with less computational complexity. We use the Kernelized Correlation Filter (KCF) tracker to track the ROI throughout the video sequence. The KCF tracker has the advantage of high efficiency.

The flowchart of the object tracking algorithm using the KCF tracker shown in Fig. 7 is discussed in the following steps.

Step 1: Take the surgical video as an input.

Step 2: Read the first frame and select the ROI using the rectangular bounding box as shown in Fig. 2.

Step 3: Initialize the KCF tracker.

Step 4: Read the next frame of the surgical video sequence.

Step 5: Create the mask with the size of the surgical frame and make all the pixel values of the mask zero.

Step 6: Check whether the ROI is tracked by the KCF tracker using the bounding box coordinates. If tracked, go to Step 7. Otherwise, treat the mask as output and go to Step 9.

Step 7: Identify the tracked bounding box coordinates and change the pixel values in the bounding box coordinates of the mask to 255.

Step 8: Find the output by applying AND operation between mask and frame.

Step 9: Write the output to the video.

Step 10: If the frame count reaches the last frame of the surgical video sequence, Stop the operation. Otherwise, move to Step 4.

KCF Tracker

We base our methodology on KCF [28], which displays amazingly real-time performance and accuracy comparative with the new top-performing trackers. The purpose of the correlation filter is to estimate an optimal filter to produce the desired response for the image input. The desired response is of Gaussian shape at the ROI location. The samples for training are obtained by cyclically shifting the whole area around the object. During testing, the position where the maximum filter response is obtained represents the target location. The KCF tracker has the advantage of high computational efficiency, obtained by utilizing a Discrete Fourier Transform (DFT). The "kernel trick" is also deployed to improve the performance of the KCF tracker further. The KCF tracker is summarized below.

Consider the cyclic shift matrix $\mathbf{X}$ with the dimension of $M \times N$. M and N represent the total number of rows and columns of the matrix. Each row represents the one-dimensional data. Let the data in the first row is $\mathbf{x} = [x_1, x_2, x_3, \ldots, x_{n-1}, x_n]$ and the data in the remaining rows represents the cyclic shifted data of previous row. The cyclic shifted data of the first row is $[x_n, x_1, x_2, x_3, \ldots, x_{n-1}]$. All the cyclic shifted rows together form a cyclic shift matrix.

During training, the tracker learns an optimal filter w that can be found by minimizing the regression error as

$$\min_{w} \sum_{j} \left(w\psi(x_j) - y_j\right)^2 + \lambda \|w\|^2 \tag{2}$$

where $\psi(x_j)$ is training samples, y_j is regression labels and $\lambda \geq 0$ represents the regularization parameter.

As the circulant matrix can be diagonalized with the help of a Discrete Fourier Transfer (DFT) matrix, w in (2) can be calculated quickly using the Fourier domain operation as

$$\hat{w} = \frac{\hat{x} \odot \hat{y}}{\hat{x} \odot \hat{x}^* + \lambda} \tag{3}$$

where $\odot \rightarrow$ element-wise product, * and ˆindicates conjugate and DFT operation.

In KCF tracker, the 'Kernel trick' is applied to improve the performance of the filter in the non-linear regression. Now the w becomes

$$w = \sum_{j} \alpha_j \psi(x_j) \tag{4}$$

where α = dual parameter of w. For the circulant matrix, the solution of the regression $\hat{\alpha}$ can be obtained as shown in (5).

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{5}$$

where k^{xx} is the first row of the kernel matrix, ˆrepresents the DFT operation.

After training, the detection operation is applied on the image patch z in the upcoming frame within a $M \times N$ window. Then the response is obtained as :

$$f(z) = \mathrm{DFT}^{-1}(\hat{k}^{xz} \odot \hat{\alpha}) \tag{6}$$

where $\hat{k}^{xz}$ is kernel correlation. Hence, the location of the target can be determined in each frame based on the maximum response ($f(z)_{\max}$). Finally, to maintain the appearance of the target, the linear interpolation is used to update the sample template $\hat{x}$ and the dual coefficients $\hat{\alpha}$ with η as a fixed learning rate is given in (7) and (8):

$$\hat{x_t} = \hat{x}_{t-1}(1-\eta) + \eta\hat{x_t}, \tag{7}$$

$$\hat{\alpha_t} = \hat{\alpha}_{t-1}(1-\eta) + \eta\hat{\alpha_t}. \tag{8}$$

The KCF tracker works efficiently when the surgical video sequence contains slow-moving objects and a smaller number of scale changes.

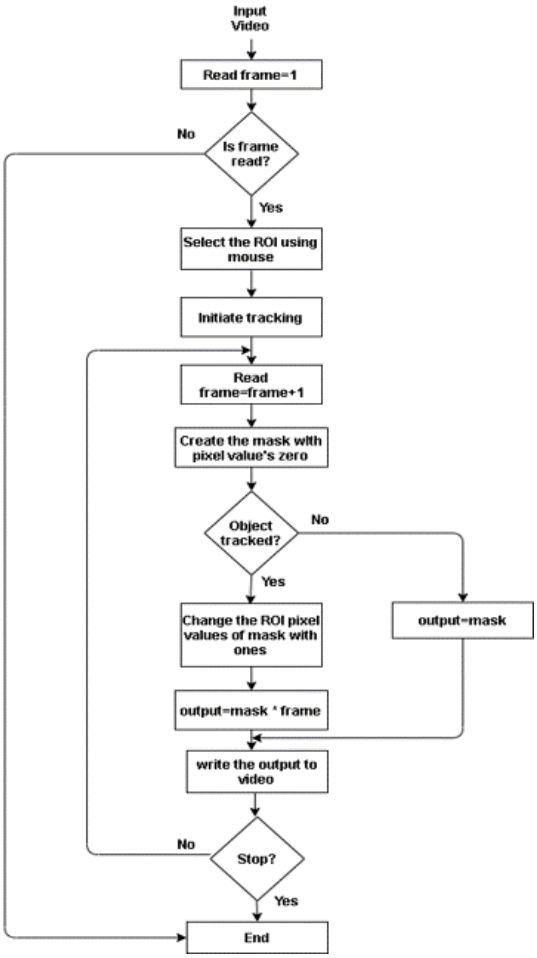


Fig. 7. Flowchart of ROI tracking using the KCF tracker.

3.3 FFmpeg

After object tracking, FFmpeg helps to convert the output video to the YUV video sequence. FFmpeg is the leading multimedia framework, able to encode, decode and transcode videos with different formats. FFmpeg can be obtained from the website https://www.ffmpeg.org/. In FFmpeg, the video sequence from 'mp4' format to 'yuv' format can be converted by using the command below:

```
ffmpeg -i input.mp4 -c:v rawvideo -pixel_format
yuv420p output.yuv
```

3.4 Hierarchical CNN and LSTM Structures for CU Size Prediction

The analysis in Sec. 3.1 shows that the correlation between the frames decreases with the distance. The CNN can use only the spatial correlation to determine the CU size. However, the LSTM can predict the CU size accurately using the temporal correlation between the frames. In this section, we will train the CNN using the residual CTU data. The features obtained after the first Fully Connected Layer (FCL) of CNN are input to the LSTM. The LSTM process the features using the LSTM gate and two FCLs to predict the CTU structure. The CNN and LSTM structures are explained below.

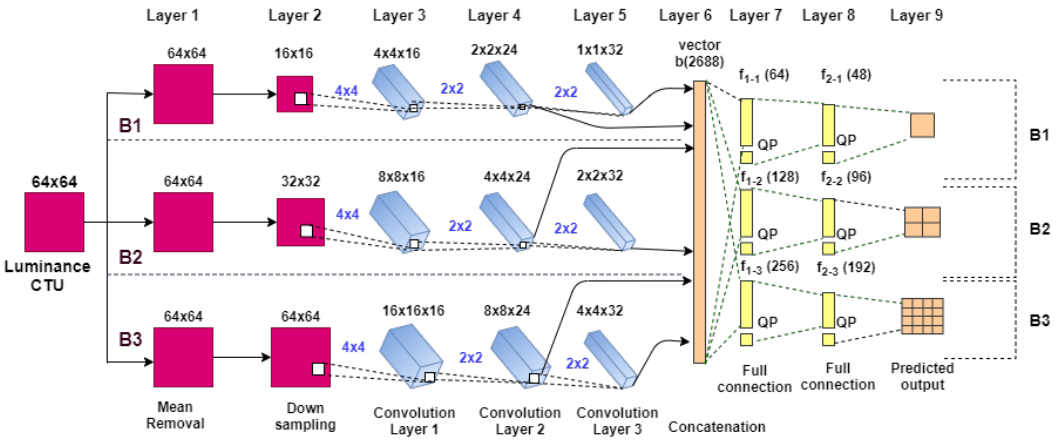


Fig. 8. Convolutional Neural Network (CNN) structure to extract the features from Coding Tree Unit.

CNN Structure

The CNN structure using the deep learning approach is shown in Fig. 8. The CNN consists of a Mean removal layer, a Downsampling layer, three convolution layers, and two fully connected layers. Each layer is discussed below.

1. **Preprocessing Layer**
   The preprocessing layer is the combination of the mean removal layer and the downsampling layer. This layer preprocesses the CTU to reduce the input sample variations. The mean removal layer removes the mean value in every CU to match the structures at the branches B1, B2, and B3, respectively. This helps to reduce the input sample variations of the CTU. The CU of size 64×64 remains the same at B3 and reduces to 32×32 and 16×16 at B2 and B1 using the downsampling process.

2. **Convolution Layer**
   After downsampling, the preprocessed data is passed through the three convolution layers. The convolution layer at layer 3 uses 4×4 kernel with 16 filters to convolve with subsampled data at three branches. The low-level features obtained from layer 3 are passed through the convolution layers at layer 4 and layer 5 to extract high-level features. In layer 4 and layer 5, 2×2 kernel with 24 and 32 filters are used for convolution with the output data of layer 3 at B1, B2, and B3 branches. In the convolution layers, the non-overlapping operations are performed by considering the width of the kernel as a stride length.

3. **Concatenating Layer**
   In this layer, the output features of layers 4 and 5 at three branches are concatenated to form a single vector b. The output features are a combination of local and global features.

4. **Fully Connected Layer (FCL)**
   The CNN structure consists of two FCLs at layer 7 and layer 8. The vector b is given as an input to FCL at three branches in layer 7. The FCL at layer 8 predicts the output based on the features of layer 7. In this paper, the accuracy of prediction is improved by using the LSTM, which takes the output features of FCL present in layer 7 of the CNN as input.

LSTM Structure

The LSTM structure shown in Fig. 9 learns the correlation between frames to predict the CTU structure. The output FCL features of CNN $f_{1-L}(t)(a)$ at layer 7 are given as an input to the LSTM cell. The LSTM cell consists of input gate $i_L(t)$, output gate $o_L(t)$ and forget gate $g_L(t)$. The three gates are trained by using (9), (10), and (11).

$$i_L(t) = \sigma(W_\mathrm{i} \cdot [f_{1-L}(t), f'_{1-L}(t-1)] + b_\mathrm{i}), \tag{9}$$
$$o_L(t) = \sigma(W_\mathrm{o} \cdot [f_{1-L}(t), f'_{1-L}(t-1)] + b_\mathrm{o}), \tag{10}$$
$$g_L(t) = \sigma(W_\mathrm{f} \cdot [f_{1-L}(t), f'_{1-L}(t-1)] + b_\mathrm{f}) \tag{11}$$

where $\sigma(\cdot)$ is sigmoid function, $W_\mathrm{i}, W_\mathrm{o}, W_\mathrm{f}$ are three gates trainable parameters and $b_\mathrm{i}, b_\mathrm{o}, b_\mathrm{f}$ are biases. The output $f'_{1-L}(t)$ of the LSTM cell is calculated by using (12).

$$f'_{1-L}(t) = o_L(t) \odot c_L(t). \tag{12}$$

The output of the LSTM is updated using three gates at frame t is given as

$$c_L(t) = i_L(t) \odot \tanh(W_\mathrm{c} \odot [f_{1-L}(t), f'_{1-L}(t-1)] + b_\mathrm{c} + g_L(t) \odot c_L(t-1) \tag{13}$$

where $W_\mathrm{c}, b_\mathrm{c}$ are parameters and biases of $c_L(t)$, and $\odot$ represents element-wise multiplication. $f_{1-L}(t), f'_{1-L}(t-1) \rightarrow$ CNN feature, LSTM cell feature output of last frame.

$f^1_{1-L}(t)(b)$ and $f^1_{1-L}(t)(c)$ represents the output of LSTM cell and first FCL with b and c features. L represents the levels in LSTM. There are three levels in LSTM, and the three levels $L = \{1, 2, 3\}$ corresponds to branches B1, B2, and B3 of CNN. The initial values of a, b, and c features are 64, 64, and 48, respectively, at $L = 1$. The output

$y^1(U,t)$ can be 0 or 1, which is obtained based on the features of the second FCL. The termination mechanism is employed to reduce the complexity of SHVC.

If $y^1(U,t) = 0$ at $L = 1$, the processing of FCLs at $L = 2, 3$ can be skipped out, which reduces the complexity. Otherwise, move to level 2 and obtain $2b$ and $2c$ output features of LSTM cell and first FCL. If the output $y^1(U,t) = 0$ at $L = 2$, skip the prediction operation. Otherwise, increment the level and repeat the operation at level 3. If the level reaches 4, terminate the prediction operation.

The output CU size at different levels when the $y^1(U,t) = 0$ is given below:

- at $L = 1$, the CU size is 64×64,
- at $L = 2$, the dimension of CU is 32×32,
- at $L = 3$, the CU size is 16×16.

The cross-entropy is used as a loss function to train the parameters. The LSTM cell at each level is trained by optimizing the loss as

$$L = \frac{1}{RT} \sum_{r=1}^{R} \sum_{t=1}^{T} L_r(t). \tag{14}$$

The parameters are trained by considering R training samples and T frames. Finally, the LSTM can predict the CU size by using the trained LSTM cells.

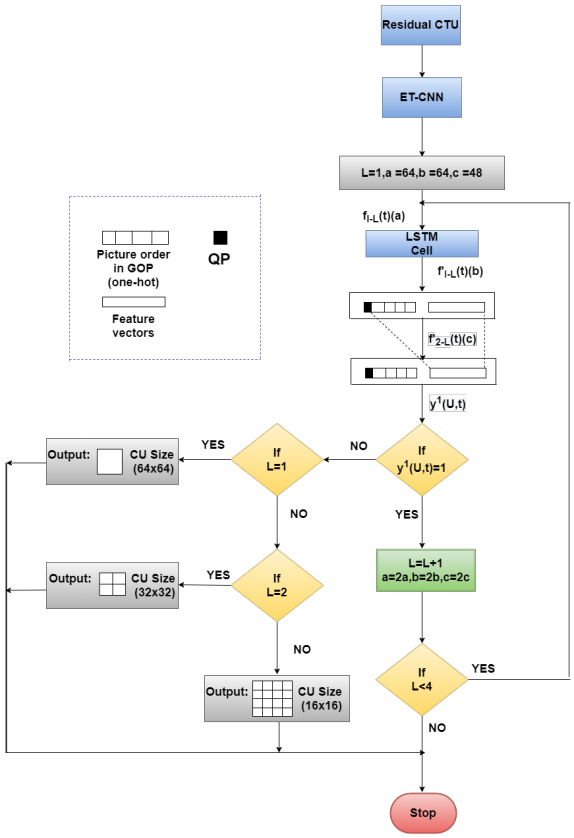


Fig. 9. Flowchart of LSTM structure to predict the CU size.

4. Experimental Results

This section presents the experimental findings to analyze the performance of the proposed method. The surgical telementoring system requires the wireless transmission of high-quality video with less bit rate. However, it is extremely difficult to encode the entire frame with high quality and less bit rate. We use the KCF tracker to track the ROI in the frames of the video, extract it and encode it with SHVC by maintaining high quality and less bitrate.

Configuration of Experiment:

The scalable HEVC reference software SHM-12.1 [29] is used to simulate the proposed method. The experiment is performed on Intel Core i7 CPU using experimental parameters shown in Tab. 1. The complexity of the CNN+LSTM approach is tested using eighteen Joint Collaborative Team on Video Coding (JCT-VC) sequences that belongs to five different classes, which are shown in Tab. 2. The proposed method is analyzed in terms of bit rate (BR) saving, time saving (TS) and change in peak signal-to-noise ratio ($\Delta PSNR$), which can be calculated using (15), (16) and (17).

$$BR_{\text{Saving}}\ [\%] = \frac{BR_{\text{orig}} - BR_{\text{prop}}}{BR_{\text{orig}}} \times 100, \tag{15}$$

$$TS\ [\%] = \frac{T_{\text{orig}} - T_{\text{prop}}}{T_{\text{orig}}} \times 100, \tag{16}$$

$$\Delta PSNR\ [\text{dB}] = PSNR_{\text{orig}} - PSNR_{\text{prop}}. \tag{17}$$

The PSNR and BR can be measured using (18) and (19).

$$PSNR = 10 \log_{10} \frac{(2^{bitdepth} - 1)^2 \times W \times H}{\sum_i (O_i - D_i)^2} \tag{18}$$

where $bitdepth$ is each pixel bit depth, W is width, H is height, O_i is reference frame pixel value, D_i is decoded frame pixel value, i is pixel address.

$$BR\ [\text{kbps}] = \frac{W \times H \times T_{\text{F}}}{bpp \times fps \times 1000} \tag{19}$$

where bpp represents bits per pixel, fps represents frames per second, W is frame width, H is frame height, and T_{F} is total number of frames. Bitrate is measured in kbps.

In addition, the average saving in bitrate (BD-BR) and average PSNR gain (BD-$PSNR$) [30], [31] quantifies the RD performance loss.

Configuration	encoder_lowdelay_P_scalable
Codec version	SHM-12.1
Number of layers in SHVC	2
QP	22, 27, 32, 37
CU size (Max)	64×64
CU depth (Max)	4
Search range and GOP Size	64 and 8

Tab. 1. Experimental conditions to simulate the proposed method in SHM-12.1.

Database for Training CNN+LSTM:

The database contains 397 video files. Out of which, 300 ultrasound video files of (112×112) size are taken from Stanford University [32], 18 sequences from the JCT-VC standard test set, and 79 video sequences of different resolutions from Xiph.org [33]. The video sequences of the database belongs to different video resolutions: SIF (352×240), CIF (352×288) , 240p (416×240), 480p (832×480), 720p (1280×720), 1080p and WQXGA (2560×1600). The above sequences are randomly divided into validation (42 sequences), testing (30 sequences), and training (325 sequences). The above sequences are encoded at four QPs by HEVC reference software to generate the CU depth data. Besides, 19,607,566 samples were collected for the LDP configuration.

Training Settings:

In this paper, the CNN is trained using the database for inter-mode. During the training process, the hyperparameters were used to tune the validation datasets of the database. The batch size for training is 32, and the momentum of the gradient descent algorithm is set to 0.8 for training the CNN. In addition, the learning rate is set to 0.01, and there are a total of 1,000,000 iterations. Similarly, to train the LSTM, the training batch size is 32, and the momentum of the gradient descent algorithm is set to 0.9. The total number of iterations is 200,000, and the initial learning rate is set to 0.01 to train the LSTM.

Test Settings:

The bi-threshold scheme is chosen by following [34] to set the upper and lower threshold levels. In addition, the threshold is set by assuming that the upper and lower thresholds are symmetrical, i.e., the upper threshold is equal to (1-lower threshold). The bi-threshold decision scheme is used at three different CU depth levels. At level 1, the upper and lower thresholds are 0.6 and 0.4, respectively. Similarly, 0.7 and 0.3 for level 2 and 0.8 and 0.2 for level 3 are chosen as the upper and lower thresholds. The CU splits if the output probability is greater than the lower threshold and less than the upper threshold value. The bi-threshold scheme is chosen such that the RD performance increases and the complexity of the SHVC decrease.

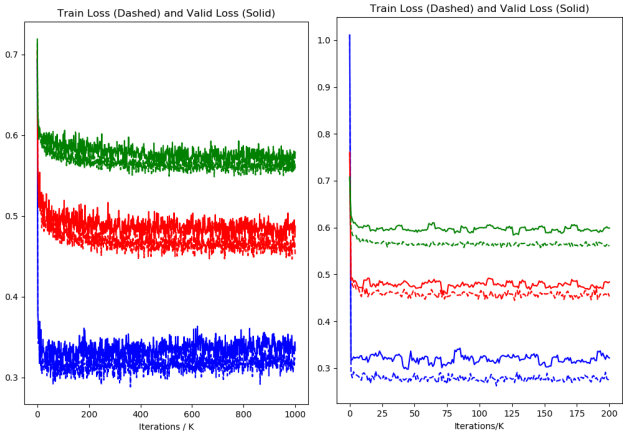
Evaluation on Training Performance and Prediction Accuracy:

The training and validation loss for CNN and LSTM alongside the iterations are shown in Fig. 10. The training loss is calculated using (14) at each iteration. The figure shows that the loss converges after 3×10^4 iteration. The average accuracy of 88%, 83%, and 78% were obtained for CU partitions at levels $L = \{1, 2, 3\}$ while training the LSTM.

Analysis of Experimental Results:

Table 2 presents the simulation results of SHVC using hierarchical CNN+LSTM (SHM+DL) approach and compared with the state-of-the-art methods: [6] and [35]. The results are generated by treating the SHM-12.1 as an anchor. The findings show that 51% of saving in encoding time (TS) can be observed with a 3.76% rise in BD-BR and 0.18 dB loss in quality. [6] and [35] approaches save the coding time by 44% and 38%, which is less compared to the proposed method. The proposed method outperforms the [6] in terms of both TS and RD-performance. However, the [35] approach provides better RD-performance than the CNN+LSTM approach at the cost of more encoding time. The CNN+LSTM approach helps to reduce the complexity of SHVC by predicting the CU size using a deep learning approach. The complexity reduction, in turn, reduces the encoding time, which is highly required for the real-time surgical telementoring system.

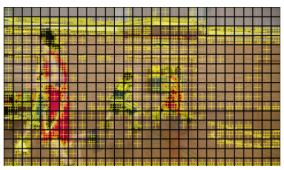
Figure 11 shows the BasketballDrive original frame, reconstructed frame, and reconstructed frame with CU partitions. From Fig. 11(c), we can observe that many CTUs are present in the frame with no partitions. The CTU with zero CU partitions represents the output $y^1(U, t)$ is zero at $L = 1$. If the output is zero, the early termination process is invoked, and the prediction operation can be skipped out at other levels, which saves the encoding time. The CTU with four CU partitions represents the output $y^1(U, t)$ is zero at $L = 2$. If the prediction is performed at level 3, then the output CTU contains more than four CU partitions.



(a) CNN (b) LSTM

Fig. 10. Training and validation loss at levels 1, 2 and 3.

(a) Original frame (b) Reconstructed frame



(c) Reconstructed frame with CU partitions

Fig. 11. BasketballDrive video sequence.

Class	Size	Video Sequence	Proposed(SHM+DL)			[6]			[35]		
			BD-*PSNR*	BD-*BR*	*TS* [%]	BD-*PSNR*	BD-*BR*	*TS* [%]	BD-*PSNR*	BD-*BR*	*TS* [%]
A	2560×1600	PeopleOnStreet	–0.18	2.09	47	–0.30	6.86	42	–0.11	2.97	40
		Traffic	–0.10	1.55	48	–0.24	8.66	45	–0.15	3.64	41
B	1920×1080	Kimono	–0.04	0.72	41	–0.13	4.54	45	–0.09	1.18	35
		ParkScene	–0.05	1.35	33	–0.23	7.77	44	–0.11	3.72	42
		Cactus	–0.16	2.87	43	–0.14	7.76	45	–0.06	3.27	41
		BasketballDrive	–0.04	0.68	46	–0.09	5.89	44	–0.05	1.21	34
		BQTerrace	–0.16	2.17	50	–0.12	5.81	46	–0.03	0.85	33
C	832×480	BQMall	–0.19	4.94	39	–0.26	6.68	43	–0.18	4.54	39
		BasketballDrill	–0.24	6.37	53	–0.32	8.23	44	–0.07	1.31	33
		RaceHorses	–0.21	4.96	54	–0.25	5.48	43	–0.10	2.32	37
		PartyScene	–0.33	5.71	52	–0.27	5.29	44	–0.17	3.32	38
D	416×240	BasketballPass	–0.41	5.76	67	–0.34	6.54	42	–0.23	4.54	39
		BlowingBubbles	–0.17	4.84	73	–0.30	6.85	42	–0.16	3.64	36
		RacingHorses	–0.37	4.99	63	–0.44	7.63	41	–0.22	3.92	35
		BQSquare	–0.25	5.38	63	–0.26	5.99	43	–0.09	1.08	30
E	1280×720	FourPeople	–0.15	4.80	40	–0.17	7.02	45	–0.12	4.82	44
		KristenAndSara	–0.13	4.87	50	–0.15	6.68	44	–0.17	7.58	45
		Johnny	–0.23	3.78	56	–0.14	8.19	45	–0.06	2.49	39
Average			**–0.18**	**3.76**	**51**	**–0.23**	**6.77**	**44**	**–0.12**	**3.13**	**38**

Tab. 2. Comparison of deep learning SHVC and state-of-the-art methods in terms of BD-BR, BD-PSNR and encoding time saving.

Video	SFF				PFROI				PROI			
	Size	*BR*	*PSNR*	Enc. time	Size	*BR*	*PSNR*	Enc. time	Size	*BR*	*PSNR*	Enc. time
Z-Plasty [37]	1280×720	5335.61	47.46	33480.15	1280×720	1489.99	56.99	8015.82	280×254	1333.52	48.22	1694.59
Digital nerve [38]		6754.70	46.25	34496.92		481.04	62.22	4933.73	128×116	410.42	46.96	343.85
Flexor [39]		11257.54	44.73	40585.82		3493.40	51.70	13786.04	410×458	3121.35	45.08	7257.96
Finger [40]		7692.73	45.60	38648.90		703.30	60.33	5483.99	148×118	547.90	45.78	425.39
Flexor Tendom [41]		3321.21	47.83	31556.20		1345.57	56.51	7641.21	416×196	1131.67	48.53	1498.82
Volar wrist [42]		48983.94	45.39	67907.25		6736.62	56.74	13099.15	410×300	782.22	48.05	1642.18
Arthroplasty [43]		5291.82	48.30	31542.49		997.25	59.01	6276.50	264×154	907.21	48.90	1017.26
Tendon saw injury [44]		74974.65	44.61	77555.37		21481.50	51.32	24137.70	526×574	3370.66	47.09	7495.33
Subcuticular [45]		121738.53	43.04	77670.56		19603.82	52.03	22111.30	454×456	3810.55	45.74	5425.51

Tab. 3. Experimental results of Default SHM 12.1 and proposed method for surgical video sequences.

(a) (b) (c) (d) (e)

Fig. 12. ROI extraction process using object tracking involves. (a) Original frame, (b) ROI selction in original frame, (c) Mask, (d) Output frame with tracked ROI, (e) Output ROI cropped frame.

Table 3 shows the experimental results of the full-frame coding using SHVC (SFF), proposed Frame with only ROI coding using SHVC (PFROI), and proposed ROI coding using SHVC (PROI). The PFROI coding is done by selecting and tracking the ROI using the KCF tracker, making the pixels other than ROI zero, and coding using SHVC. The PROI coding is performed by tracking the ROI from the surgical video using the KCF tracker, cropping the ROI, and coding the ROI using SHVC. Figure 12 shows the Full frame, mask, Frame with ROI, and cropped ROI. During the simulation process, only one layer is chosen in SHVC, which acts as a single layer HEVC to encode surgical videos. Based on the analysis present in [36], we choose $QP = 20$ for simplicity to encode surgical videos with high quality.

Video	Prop	SegNet	S-CNN	MST	Prop	SegNet	S-CNN	MST
	Pixel accuracy				*fIOU*			
Z-Plasty	97.40	96.7	98.6	94.3	98.21	98.1	97.5	92.0
Digital Nerve	98.21	97.4	98.3	96.4	97.48	97.6	97.3	94.4
Flexor	97.85	96.9	97.1	93.0	96.78	95.4	94.4	86.3
Finger	97.18	98.0	98.2	94.4	98.55	97.7	98.3	94.3
Flexor Tendom	96.42	96.3	98.1	86.9	96.80	94.3	96.5	84.1
Volar Wrist	96.68	96.8	97.4	93.2	97.27	95.4	96.3	91.2
Arthroplasty	97.97	97.3	98.6	95.1	96.23	97.4	97.0	93.6
Tendon Saw Injury	98.57	97.3	98.4	94.6	96.96	97.0	97.5	92.4
Subcuticular	96.85	98.0	97.3	95.1	97.93	98.3	96.9	92.9
Average	**97.45**	**97.18**	**98.0**	**93.66**	**97.35**	**96.80**	**96.85**	**91.14**

Tab. 4. Comparison of proposed method and state-of-the-art methods segmentation accuracy for surgical videos.

Video	PROI		MST		SegNet		S-CNN	
	BR_{Saving} [%]	*PSNR*	BR_{Saving} [%]	*PSNR*	BR_{Saving} [%]	*PSNR*	BR_{Saving} [%]	*PSNR*
Z–Plasty	75	0.76	90.05	–7.89	70.18	–0.12	74.86	–0.05
Digital nerve	93.92	0.71	96.61	–15.91	77.41	–0.09	76.53	–0.06
Flexor	72.27	0.35	88.41	–10.54	65.32	–0.16	63.70	–0.01
Finger	92.87	0.18	88.15	–10.99	79.36	–0.13	82.84	–0.11
Flexor Tendom	75.92	0.70	89.78	–8.65	71.15	–0.08	75.02	–0.04
Volar wrist	98.40	2.66	95.26	–10.98	76.93	–0.14	80.02	–0.04
Arthroplasty	82.85	1.60	95.93	–14.83	74.83	–0.07	74.71	–0.05
Tendon saw injury	95.50	2.48	93.78	–7.14	82.04	–0.10	86.05	–0.02
Subcuticular	96.87	2.70	98.40	–16.56	76.31	–0.19	80.55	–0.02
Average	**87.06**	**1.34**	**92.93**	**–11.49**	**74.83**	**–0.12**	**77.14**	**–0.04**

Tab. 5. Comparison of the proposed method and state-of-the-art methods in terms of bit rate saving and PSNR for surgical videos.

Table 4 presents the segmented accuracy results of the proposed method, SegNet, S-CNN, and the MST techniques. The segmented accuracy is calculated using pixel accuracy, and frequency weighted IoU (*fIoU*). The pixel accuracy and fIoU can be measured using (20) and (21).

$$Pixel\ accuracy = \frac{TN + TP}{TP + TN + FP + FN} \tag{20}$$

where $TP \rightarrow$ true positive, $TN \rightarrow$ true negative, $FP \rightarrow$ false positive and $FN \rightarrow$ false negative.

$$fIoU = \left(\sum_k T_k\right)^{-1} \left(\frac{\sum_n T_n p_{nn}}{T_n + \sum_m p_{nm} - p_{nn}}\right) \tag{21}$$

where p_{nn} is the number of correctly identified pixels, p_{nm} is the number of pixels rejected incorrectly for class m and T_n is the total number of pixels in class n.

From Tab. 4, the results show that the proposed approach achieved higher pixel accuracy compared to the SegNet and MST techniques. Even though the pixel accuracy is slightly less than the S-CNN, the proposed method can obtain higher *fIoU* than the S-CNN and other two state-of-the-art methods. We have also used the Mean Opinion Score (MOS) as a metric to evaluate the quality of the video for subjective quality assessment. The MOS is the average score of the expert on video quality. The score can be 1 to 5. '1' represents the lowest video quality, and '5' represents the highest. In this assessment, we have taken five surgical videos, and then each video is encoded by the SHVC. The output of the SHVC is a bitstream. The output video is reconstructed from the bitstream, and the input video is used as a reference. There are ten expert viewers, and each expert has given a score by observing the reference and output video sequences. The video sequences with average MOS scores are given in Tab. 6. From Tab. 6, all the MOS values are in the range of 3.8 to 4.4, indicating that the proposed method encodes the surgical video sequences with good quality. The MOS confidence lower and upper intervals for Z-Plasty, Digital Nerve, Flexor, Flexor Tendon, and Tendon saw injury video sequences are (3.64, 4.76), (3.64, 4.76), (3.42, 4.58), (3.24, 4.36) and (3.9, 4.9) respectively.

In Tab. 5, the PROI approach is compared with the state-of-the-art methods [10], [21] and [36], which uses the MST, SegNet and S-CNN surgical ROI segmentation techniques for surgical telementoring systems. The results indicate that the proposed method achieves 87% less bit rate with 1.34 dB improvement in PSNR using PROI. This improvement is achieved for surgical video sequences with slow-moving objects.

Video sequence	Z-Plasty	Digital Nerve	Flexor	Flexor Tendom	Tendon Saw Injury
Average MOS	4.2	4.2	4.0	3.8	4.4

Tab. 6. Average mean opinion scores for the subjective quality assessment of surgical video sequences.

The MST technique achieves high BR savings of 92.93% which is high compared to our proposed technique. However, 11.49 dB of PSNR loss is observed, making it less suitable for telementoring applications. The authors in [36] use the HEVC for encoding the ROI region. The HEVC uses the RDO search process that increases the complexity. The complexity increases the coding time, which makes it unsuitable for real-time telementoring applications. The SegNet approach saved the bitrate by 75%, which is less compared to the remaining approaches. In addition, SegNet uses 26 convolutional layers in the CNN model that increases the computational complexity. We use the CNN+LSTM approach to reduce the complexity, which decreases approximately 53% of encoding time required to encode using SHVC.

5. Conclusions

This paper proposed an efficient surgical telementoring system that transmits the surgical incision region at high quality with less bit rate. The surgical video consists of the surgical incision region and the background region. The background region can be removed to reduce the bit rate. The Kernelized Correlation Filter (KCF) tracker tracks the surgical incision region, crop, and writes to the video sequence. The resultant video is encoded using the SHVC video coder. SHVC uses the CNN+LSTM approach to predict the CTU structure in less time. On average, the Deep learning CNN+LSTM method helps in reducing the encoding time by 51% with a 3.76% rise in BD-BR and 0.18 dB loss in BD-PSNR compared to SHM-12.1 standard. Furthermore, the proposed method encodes the ROI surgical video using SHM software that saves the bit rate by 87% with a 1.34 dB improvement in video quality (PSNR).

References

[1] SULLIVAN, G. J., OHM, J. R., HAN, W. J., et al. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits, Systems for Video Technology*, 2012, vol. 22, no. 12, p. 1649–1668. DOI: 10.1109/TCSVT.2012.2221191

[2] WIEGAND, T., OHM, J. R., SULLIVAN, G. J., et al. Special section on the joint call for proposals on high efficiency video coding (HEVC) standardization. *IEEE Transactions on Circuits, Systems for Video Technology*, 2010, vol. 20, no. 12, p. 1661–1666. DOI: 10.1109/TCSVT.2010.2095692

[3] WIEGAND, T., SULLIVAN, G. J., BJONTEGAARD, G., et al. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits, Systems for Video Technology*, 2003, vol. 13, no. 7, p. 560–576. DOI: 10.1109/TCSVT.2003.815165

[4] SCHWARZ, H., MARPE, D., WIEGAND, T. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits, Systems for Video Technology*, 2007, vol. 17, no. 9, p. 1103–1120. DOI: 10.1109/TCSVT.2007.905532

[5] OHM, J. R., SULLIVAN, G. J., SCHWARZ, H., et al. Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC). *IEEE Transactions on Circuits, Systems for Video Technology*, 2012, vol. 22, no. 12, p. 1669–1684. DOI: 10.1109/TCSVT.2012.2221192

[6] YE, Y., ANDRIVON, P. The scalable extensions of HEVC for ultra-highdefinition video delivery. *IEEE Transactions on Multimedia*, 2014, vol. 21, no. 3, p. 58–64. DOI: 10.1109/MMUL.2014.47

[7] BOYCE, J. M., YE, Y., CHEN, J., et al. Overview of SHVC: Scalable extensions of the high efficiency video coding standard. *IEEE Transactions on Circuits, Systems for Video Technology*, 2016, vol. 26, no. 1, p. 20–34. DOI: 10.1109/TCSVT.2015.2461951

[8] CHEN, C., BOYCE, J., YE, Y., et al. Scalable HEVC (SHVC) Test Model 6 (SHM 6). In *Meeting MPEG 108 - Valencia*. Valencia (Spain), 2014, JCTVC-Q1007, p. 1–9.

[9] YIN, P., XIU, X., YE, Y. Inter-layer reference picture placement. In *Meeting of the Joint Collaborative Team on Video Coding (JCT-VC)*. Geneva (Switzerland), 2013, JCTVC-L0174.

[10] COMANICIU, D., MEER, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, vol. 24, no. 5, p. 603–619. DOI: 10.1109/34.1000236

[11] XU, M., DENG, X., LI, S., et al. Region-of-interest based conversational HEVC coding with hierarchical perception model of face. *IEEE Journal of Selected Topics in Signal Processing*, 2014, vol. 8, no. 3, p. 475–489. DOI: 10.1109/JSTSP.2014.2314864

[12] GOKTURK, S. B., TOMASI, C., GIROD, B., et al. Medical image compression based on region of interest, with application to colon CT images. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine, Biology Society (EMBC)*. Istanbul (Turkey), 2001, p. 2453–2456. DOI: 10.1109/IEMBS.2001.1017274

[13] YU, H., LIN, Z., PAN, F. Applications, improvement of H.264 in medical video compression. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2005, vol. 52, no. 12, p. 2707–2716. DOI: 10.1109/TCSI.2005.857869

[14] WU, Y., LIU, P., GAO, Y., et al. Medical ultrasound video coding with H. 265/HEVC based on ROI extraction. *PLoS One*, 2016, vol. 11, no. 11, p. 1–13. https://doi.org/10.1371/journal.pone.0165698

[15] KHIRE, S., ROBERTSON, S., JAYANT, N., et al. Region-of-interest video coding for enabling surgical telementoring in low-bandwidth scenarios. In *Military Communications Conference (MILCON)*. Orlando (USA), 2012, p. 1–6. DOI: 10.1109/MILCOM.2012.6415792

[16] GROIS, D., KAMINSKY, E., HADAR, O. ROI adaptive scalable video coding for limited bandwidth wireless networks. In *IFIP Wireless Days (WD)*. Venice (Italy), 2010, p. 1–5. DOI: 10.1109/WD.2010.5657709

[17] BARSAKAR, T., MANKAR, V. A novel approach for medical video compression using kernel based meanshift ROI coding techniques. In *Conference on Advances in Signal Processing (CASP)*. Pune (India), 2016, p. 212–216. DOI: 10.1109/CASP.2016.7746167

[18] MUBEEN, G., ALI, T., BAKR, M., et al. Perceptually lossless surgical telementoring system based on non-parametric segmentation. *Journal of Medical Imaging and Health Informatics*, 2019, vol. 9, no. 3, p. 464–473. DOI: 10.1166/jmihi.2019.2512

[19] XIE, W., YAO, Z., JI, E., et al. Artificial intelligence based computed tomography processing framework for surgical telementoring of congenital heart disease. *ACM Journal on Emerging Technologies in Computing Systems*, 2021, vol. 17, no. 4, p. 1–24. DOI: 10.1145/3457613

[20] PENG, L., CHENMENG, L., CHANGLIN, X., et al. A Wearable augmented reality navigation system for surgical telementoring based on Microsoft HoloLens. *Annals of Biomedical Engineering*, 2020, vol. 49, no. 1, p. 287–298. DOI: 10.1007/s10439-020-02538-5

[21] BADRINARAYANAN, V., KENDALL, A., CIPOLLA, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, vol. 39, no. 12, p. 2481–2495. DOI: 10.1109/TPAMI.2016.2644615

[22] WEI, H., ZHOU, X., ZHOU, W., et al. Visual saliency based perceptual video coding in HEVC. In *IEEE International Symposium on Circuits and Systems (ISCAS)*. Montreal (Canada), 2016, p. 2547–2550. DOI: 10.1109/ISCAS.2016.7539112

[23] WANG, L., PEDERSEN, P. C., STRONG, D. M., et al. Smartphone-based wound assessment system for patients with diabetes. *IEEE Transactions on Biomedical Engineering*, 2015, vol. 62, no. 2, p. 477–488. DOI: 10.1109/TBME.2014.2358632

[24] RAMYA, R., JENITTA, A. Foot injury detection using K-means clustering, mean shift segmentation algorithm. *International Journal of Advanced Research in Basic Engineering Sciences and Technology*, 2017, vol. 3, no. 24, p. 323–329. ISSN: 2395-695X

[25] WANNOUS, H., TREUILLET, S., LUCAS, Y. Robust tissue classification for reproducible wound assessment in telemedicine environments. *Journal of Electronic Imaging*, 2010, vol. 19, no. 2, p. 1–9. DOI: 10.1117/1.3378149

[26] WANG, C., YAN, X., SMITH, M., et al. A unified framework for automatic wound segmentation, analysis with deep convolutional neural networks. In *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Milan (Italy), 2015, p. 2415–2418. DOI: 10.1109/EMBC.2015.7318881

[27] GOYAL, M., REEVES, N. D., DAVISON, A. K., et al. DFUNet: Convolutional neural networks for diabetic foot ulcer classification. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020, vol. 4, no. 5, p. 728–739. DOI: 10.1109/TETCI.2018.2866254

[28] HENRIQUES, J. F., CASEIRO, R., MARTINS, P., et al. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, vol. 37, no. 3, p. 583–596. DOI: 10.1109/TPAMI.2014.2345390

[29] FRAUNHOFER HEINRICH HERTZ INSTITUTE. *SHVC Reference Software Version.* [Online] Cited 04-July-2021. Available at: hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-12.1

[30] BJONTEGAARD, G. *Calculation of Average PSNR Differences Between RD Curves.* ITU-T SG16/Q6 Document, VCEG-M33, Austin, 2001.

[31] BJONTEGAARD, G. *Improvements of the BD-PSNR Model*. ITU-T SG16/Q6 Document, VCEG-AI11, Berlin, 2008.

[32] STANFORD UNIVERSITY. *Echocardiogram Ultrasound Dataset*. [Online] Cited 21-July-2021. Available at: https://echonet.github.io/dynamic/

[33] XIPH.ORG FOUNDATION. *Xiph.org Video Test Media*. [Online] Cited 21-July-2021. Available at: https://media.xiph.org/video/derf/

[34] ZHANG, Y., KWONG, S., WANG, X., et al. Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding. *IEEE Transactions on Image Processing*, 2015, vol. 24, no. 7, p. 2225–2238. DOI: 10.1109/TIP.2015.2417498

[35] SHI, N., MA, R., LI, P., et al. Efficient mode decision algorithm for scalable high efficiency video coding. *Proceedings of SPIE Optoelectronic Imaging, Multimedia Technology*, 2014, vol. 9273, p. 1–8. DOI: 10.1117/12.2071709

[36] HASSAN, A., GHAFOOR, M., TARIQ, S. A., et al. High efficiency video coding (HEVC)-based surgical telementoring system using shallow convolutional neural network. *Journal of Digital Imaging*, 2019, vol. 32, p. 1027–1043. DOI: 10.1007/s10278-019-00206-2

[37] MCCLELLAN, T. *Z-Plasty of Scar Contracture (Finger)*. [Online] Cited 07-July-2021. Available at: https://youtu.be/wdseg3UvXrI

[38] MCCLELLAN, T. *The Digital Nerve Was Cut*. [Online] Cited 05-July-2021. Available at: https://youtu.be/CY1HYIBrAwQ

[39] MCCLELLAN, T. *Flexor Digitorum Profundus (FDP) Finger Tendon Repair*. [Online] Cited 06-July-2021. Available at: https://youtu.be/boMlEa3P43g

[40] MCCLELLAN, T. *Foreign Body (BB) Removal from Finger*. [Online] Cited 07-July-2021. Available at: https://youtu.be/DWQ6WX3ImBU

[41] MCCLELLAN, T. *Ganglion Cyst: Flexor Tendon Sheath (Finger)*. [Online] Cited 04-July-2021. Available at: https://youtu.be/hDZBE8tcctE

[42] MCCLELLAN, T. *Ganglion Cyst Volar Wrist*. [Online] Cited 04-July-2021. Available at: https://youtu.be/ZgNJ8YDA7dY

[43] VANGELISTI. *NuGrip Arthroplasty (Thumb Arthritis Joint Replacement Surgery)*. [Online] Cited 06-July-2021. Available at: https://youtu.be/YZgDQl5kWFs

[44] MCCLELLAN, T. *Small Finger Extensor Tendon Saw Injury Cut Repair*. [Online] Cited 04-July-2021. Available at: https://youtu.be/3o7cgZsd3bs

[45] MCCLELLAN, T. *Running Subcuticular Suture*. [Online] Cited 05-July-2021. Available at: https://youtu.be/CiW93U-3XcQ

About the Authors ...

Karthik Sairam SANAGAVARAPU received his B.Tech degree in Electronics and Communication Engineering from DVR college of Engineering and Technology in 2012 and his M.Tech degree from Vardhaman college of Engineering, Hyderabad in 2015. He is a research scholar at National Institute of Technology, Warangal since 2018. His research interests are VLSI architectures for video compression algorithms.

Muralidhar PULLAKANDAM received B.Tech (ECE) and M.Tech degree in Electronic Instrumentation from National Institute of Technology, Warangal, India, in 1993 and 2004 respectively. Then he has received his Ph.D. degree from NIT Warangal. He joined Apollo Computing Labs Hyderabad in 1994 where he was engaged in design and development of high speed digital circuits. He joined NIT Warangal in 1997. Since then he has been working in the ECE Department, NIT Warangal. His research interests include design of embedded systems and VLSI architectures for video processing systems.