POLITECNICO
MILANO 1863

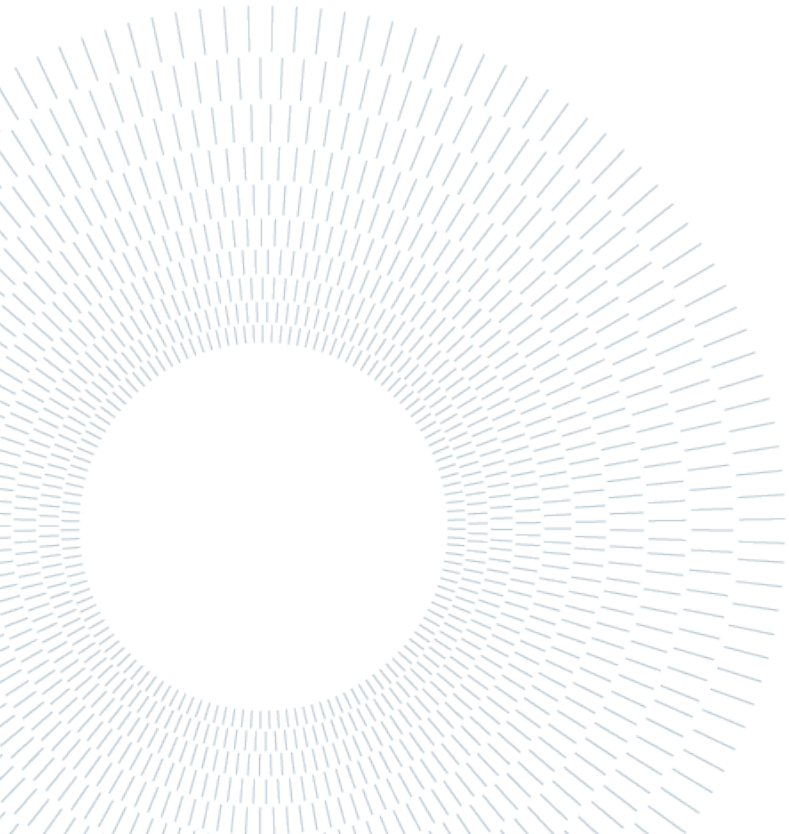
SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Real-time and high-quality video compression for telesurgery

TESI DI LAUREA MAGISTRALE IN
BIOMEDICAL ENGINEERING
INGEGNERIA BIOMEDICA

Author: **Martina Golini**

Student ID: 944114
Advisor: Elena De Momi
Co-advisor: Iuri Frosio, Aldo Marzullo
Academic Year: 2021-2022



Abstract

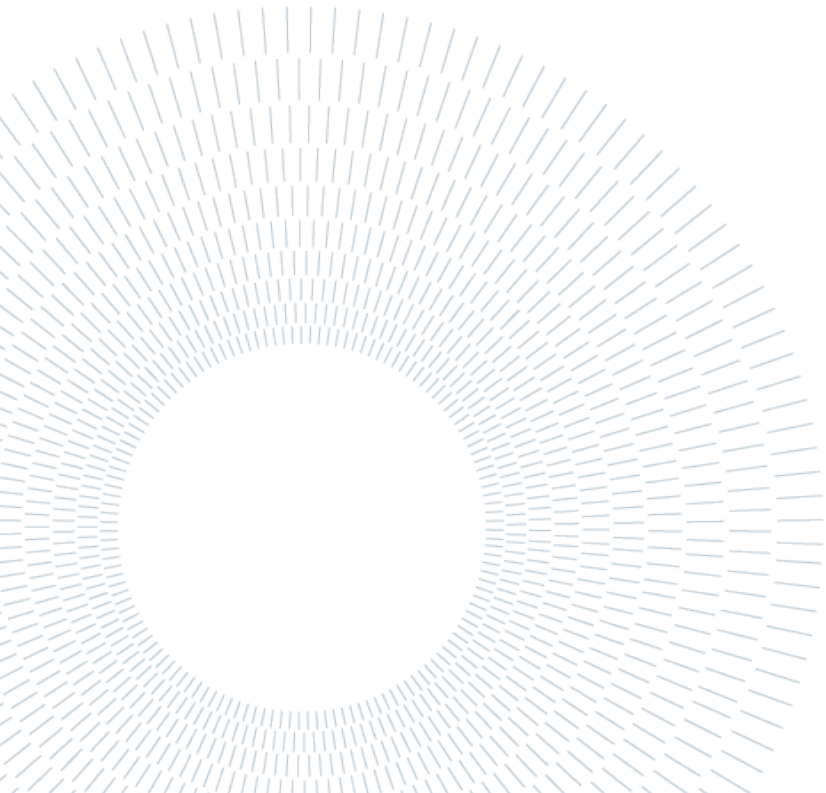
Nowadays deep learning-based solutions are widely spread among different fields. The employment in the surgical domain may result a useful tool to address the challenges proposed by the new frontiers of medicine. Indeed, telementoring, teleoperation and remote diagnosis, now realities thanks to advances in telecommunication technology and video coding system, require sophisticated system to storage and transmit big data, e.g., high-resolution videos. Focusing on video transmission, constrains are present in terms of latency and bandwidth to guarantee the real time application, without losing quality. In the specific case of remote surgery, low-latency and bandwidth are essential to ensure the stability of the system employed. Even though traditional approaches are highly performant, a further improvement would increase the efficiency, thus the employment, of these services. Since the leading standards for video compression, i.e., H.264/AVC and H.265/HEVC, have reached a turning point in terms of performance, alternative solutions for their optimizations and brand-new schemes needs to be explored. Deep Learning (DL) techniques may be well suited for the purpose, as they can overcome the limitations featured by the traditional video codecs. In this work, a deep learning-based method is proposed to enhance the performance of H.264/AVC in terms of quality, bandwidth and latency for Robot Assisted Minimally Invasive Surgery (RAMIS), namely for the Robotic Assisted Radical Prostatectomy (RARP). A binary autoencoder is proposed to compress the residual, thus the difference between the original and the compressed frame. The output of the network is summed to the one of H.264/AVC to obtain a better image reconstruction while saving compression time. The scheme proposed overcomes the traditional codec both in terms of quality and speed in a low bitrate scenario. Moreover, it is computational friendly and it could be further optimized to become a powerful tool for telemedicine applications.

Key-words: telesurgery, teleoperation, remote surgery, video compression, deep learning, real time, high quality, robotic assisted minimally invasive surgery

Abstract in italiano

Oggigiorno le soluzioni basate sul deep learning sono ampiamente diffuse in differenti contesti. Il loro utilizzo nel dominio chirurgico potrebbe risultare uno strumento utile per affrontare le sfide proposte dalle nuove frontiere della medicina. Infatti, le applicazioni di telemedicina sono divenute ormai realtà grazie ai progressi della tecnologia nel campo delle telecomunicazioni e del sistema di codifica video, e richiedono sistemi sofisticati per archiviare e trasmettere big-data, quali ad esempio video ad alta risoluzione. Nel caso specifico della trasmissione video, sono presenti vincoli in termini di latenza e larghezza di banda per garantire l'applicazione in tempo reale. La qualità deve essere comunque preservata. Per la chirurgia da remoto, bassa latenza e larghezza di banda sono essenziali per assicurare la stabilità del sistema impiegato. Anche se gli approcci tradizionali sono altamente performanti, un miglioramento ulteriore consentirebbe un aumento dell'efficienza con conseguente diffusione di questi servizi. Poiché gli standard correnti utilizzati per la compressione video, i.e., H.264/AVC e H.265/HEVC, hanno raggiunto altissimi livelli in termini di prestazioni, è necessario esplorare soluzioni alternative per la loro ottimizzazione, oppure sviluppare nuove tecniche di compressione. I metodi di Deep Learning (DL) possono considerarsi adatte allo scopo, poichè in grado di superare le limitazioni proprie dei codec tradizionali. In questa tesi si propone una rete neurale per migliorare le prestazioni di H.264/AVC in termini di qualità, larghezza di banda e latenza per la chirurgia mini-invasiva assistita da robot. Si propone un autoencoder binario per comprimere il residuo, ossia la differenza tra il frame originale e quello compresso. L'output prodotto dalla rete è sommato a quello di H.264/AVC al fine di ottenere una migliore ricostruzione dell'immagine, riducendo tempo di compressione. Lo schema proposto supera il codec tradizionale sia in termini di qualità che di velocità nello scenario dei bassi bitrate. Inoltre, è di facile implementazione e potrebbe essere ulteriormente ottimizzato, divenendo un potente strumento per la telemedicina.

Parole-chiave: telechirurgia, teleoperazione, chirurgia da remoto, compressione video, deep learning, tempo reale, alta qualità, chirurgia mini-invasiva.



Contents

Introduction

Video compression and transmission methods have changed among time; nowadays many Deep Learning algorithms are exploited to guarantee high performances in terms of quality while reducing bandwidth requirements at the same time. Deep Learning (DL) solutions are investigated in different fields and among companies, to ensure customers an increasingly efficient service. Disney can be mentioned as a clear example of the usage of DL methods [1].

In recent years, advances in telecommunication technology and video coding system have opened new perspectives also in the surgical area, where telementoring, teleoperation and remote diagnosis were at their infancy. Surgery in the 21st century is facing new challenges, as it plays an increasingly significant role in the treatment of acute and chronical diseases. However, the access to timely, affordable and safe surgical care is limited not only in middle and low-income countries, but also in the high-income ones, where centralization of cares concentrates specialist surgery in metropolitan hubs, with a consequent limitation of accessing surgery in rural areas [2], [3]. Telemedicine provides good alternatives to the emerging challenges. In particular, telementoring offers a solution which increases both quality and access to surgical care, allowing expert surgeons to guide their less experienced colleagues through the procedure from a remote location [2], [3]. In this context, high amount of data needs to be transmitted and the transmission error can affect the perceived video quality. This results in bandwidth constrains, which interfere with the achieving of real time performances. Furthermore, the usage of closed-loop-control mechanism and surgery robot in remote surgery requires a latency control [3], to guarantee the stability (no oscillations) of the system. Proper compression methods are necessary to satisfy the requests both on latency and bandwidth. Lossless algorithms cannot be used for real time application, for large bandwidth and high latency are required to maintain quality. In this context, a high compression level is not achieved. On the contrary, lossy algorithms can be well suited for the task. These techniques have been developed specially to reduce the data size for storage, handling, and transmitting content. H.264/AVC (Advanced Video Coding) and its successor H.265/HEVC (High Efficiency Video Coding) represent the most viable choice in many applications, for they can provide smaller files in higher quality with respect to the previous generations.

In fact, “they have a “smaller” algorithm that’s better at choosing the data to throw out” [5]. Since both H.264/AVC and H.264/HEVC are based on the hybrid prediction/transform coding method, proposed for the first time in 1979 by Netravali and Stuller [6], they can introduce block artefacts and other forms of quality compression degradation because of the quantization step applied before data transmission. Therefore, in recent times learning methods - Deep Learning techniques in particular - have been exploited for leveraging the potential of H.264/AVC and H.265/HEVC in the world of streaming and off-line video compression [1], [7]. Brand new codecs and solutions for improving one of the five main modules of the traditional codecs (intra-prediction, inter-prediction, quantization, entropy coding and loop filtering) have been developed through the years [6]. The analysis of the current state-of-art demonstrate that the employment of deep learning methods is still in its initial stages, but it shows good perspectives for the future.

Focusing on the medical field, it can be said that although H.265/HEVC overcomes its predecessor in terms of quality, it is computationally more demanding, hence it is not as widely spread as H.264/AVC [8]–[10]. As regards the usage of Deep Learning-based solutions in this area, its application is reported in very few documents. The thesis aims to explore a brand-new technique for surgical video transmission, with its focus on robot assisted minimally invasive procedures. In fact, in modern times, the usage of robotic systems to increase both precision and safety of surgical procedures is spread among the surgical area[10], especially in Minimally Invasive Surgery (MIS). MIS techniques seek to perform surgical procedures while avoiding the morbidity of the conventional surgical wounds by employing little tools and miniaturized video cameras to allow the visualization of the surgical area inside the body [11]. Nowadays, MIS procedures are broadly exploited among every specialty of surgical medicine, and their employment is continuing to expand [13]. The reason which stands behind the fully establishment of MIS can be found in improved immune responses, shorter hospital stays, reduced size of the instruments with better cosmetic results and a minimization of the area exposed [13]. The employment of a master-slave surgical system, i.e., the Da Vinci Robot, in laparoscopic procedures provides several advantages, such as 3D vision, motion scaling, intuitive movements, visual immersion and tremor filtration [13]. Therefore, Robotic Assisted Minimally Invasive Surgery (RAMIS) can optimize MIS techniques, increasing the quality of surgical cares. Nowadays, RAMIS techniques are mainly exploited by urologists, general surgeons, cardiothoracic, gynecologists and pediatric surgeons [13]. Robot Assisted Radical Prostatectomy (RARP) constitutes one of the most performed Robotic Assisted MIS operations [14].

The procedure aims to remove the prostate and nearby tissues with great precision, through several small keyhole incisions in the patient's abdomen [15].

The robotic assisted surgery procedure involves a system composed by a 3D endoscope and an equipment for image processing is employed, to deliver a superior view of the surgical area and its surrounding structures. The surgeon performs the operation guiding the small wristed instruments into the patient's body cavity through the console.

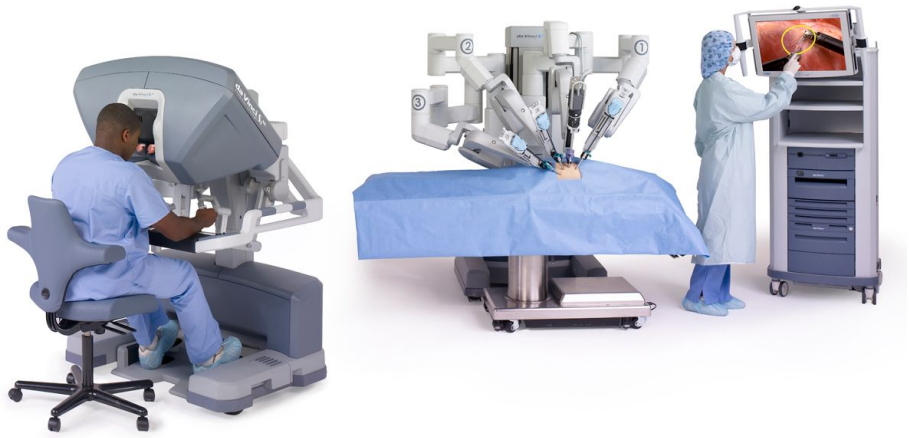


Figure 0.1 The Da Vinci robot system employed for many surgical procedures, included RARP. The surgeon performs the operation guiding the tiny wristed robotic arms through the console. Another surgeon is in charge of controlling the surgery through the video displayed, acquired by the 3D endoscope.

It is clear that in this context image processing is of great importance within the system, thus it requires advanced methods for compression, transmission and reconstruction of the image, to avoid loss of clinically relevant detail.

In this work a Neural Network has been trained on 1280x720 (**720p**) RARP videos - thus HD videos - collected from the Da Vinci robot for remote surgery, to achieve good performances both in terms of quality and time of transmission, to ensure a safe and precise execution by the surgeon, who operates from a remote position.

More precisely, the stability and the effectiveness of the system are guaranteed by:

- *Latency*: the stability of the closed-loop-system is strictly associated to the delay of the control signal going back to the robotic arm, as well as the one related to the transmission of the surgical field images acquired through the cameras [16]. Both delays need to be the smallest possible for real time application, hence average and maximum latency value have to be defined [17]–[19].

- *Bandwidth*: the bandwidth required to transmit the data does not have to exceed the bandwidth allowed by the transmission system.

- *Quality*: the quality needs to be good enough for the surgeon must be able to detect any clinically relevant detail during the procedure, e.g., small bleeding or unexpected tumor masses [8], [9].

The results achieved by the Neural Network implemented in the thesis have been compared to those of H.264/AVC. More precisely, PSNR and SSIM have been considered to assess quality; latency has been evaluated in terms of encoding and decoding time, considering **30 ms** (latency value for real time applications) as maximum acceptable value for both encoding and decoding time. Both quality and time have been calculated for different bitrate, ranging from 1 to 10 Mb/s, at different presets: ultrafast, medium, slow. The network shows to overcome H.264/AVC quality performances in a low bitrate scenario, featuring an encoding time lower than 30ms, thus suitable for real time application. Since the employment of deep learning structures in the surgical area is still at its infancy, there is a large space for improvement and optimization, allowing increasingly better services and opening new prospective toward unexplored horizons.

The thesis proposes a computational friendly solution for real time and high-quality endoscopic video transmission, to enhance the services offered by telemedicine and telementoring, increasing the possibility of their application, which can lead to an improvement in terms of life quality among different contexts. In particular, the work focuses on RARP.

The thesis is organized as follows: in Chapter I is proposed an overview of the existent methods which address the same challenge, both for video compression among different fields and for endoscopic video compression. Chapter II illustrates the proposed method and its application on a specific dataset; Chapter III presents the results achieved in terms of quality and time and a comparison between the performance associated to the H.264/AVC and to the scheme proposed. Chapter IV contains a discussion of the results, followed by the conclusions.

1 State of the art

The limitations in storing and transmitting picture data for video applications led to the development of techniques to overcome the constrains, reducing the large quantity of information used to represent the content without excessively reducing the quality of the original data. More in detail, video compression methods and tools aim to diminish size of a video by eliminating redundancies, which can be of four types: spatial, temporal, statistical and color space [20]. The reduction of the amount of data leads to lower bandwidth requirements and a smaller storing space.

Compression techniques can be categorized as *Lossy* or *Lossless*. Lossless algorithms allow a more efficient saving of data in their compression state, without losing any information, hence the original data can be perfectly reconstructed from the compressed ones. In other words, the process eliminates redundancies without affecting quality, hence guaranteeing data integrity and fully reversibility. Unfortunately, a high compression level cannot be reached and the file size is not significantly reduced. These methods are mainly used for text, programs, images and sounds [5]. Huffman coding can be cited as an example of lossless compression algorithm and it will be further deepened. Lossy techniques are based on the assumption that some data are irrelevant for human perception, thus they can be eliminated (perceptual coding). This results in a substantial data size reduction which leads t0 an approximate recovery of the original data [5], [16]. If compression is important the losses become noticeable, causing visible artifacts, such as blockness, blur, color bleeding, and banding in the reconstructed image or video [20]. Lossy algorithms, like Discrete Cosine Transform (DCT), are applied for images, audio and video [5].
Video can be compressed using both *intra and inter-frame strategies*. The first one refers to a compression of an individual frame. The inter-frame strategy instead utilizes redundancies between two successive video frames, thus the encoding scheme only keeps the information that changes. Various codecs, tools that perform compression, deploy these strategies [20]to guarantee quality without being highly computational demanding, both in terms of time and space.
As previously mentioned, video codecs are systems aim to compress video data, changing their format in a supported one by video player or decoders.

Over the last four decades, MPEG - Motion Pictures Expert Group – have developed MPEG-1, MPEG-2, MPEG-4 and MPEG-H video compression standards and the collaboration with the International Standard Organization/Motion Picture Expert Group - ISO/MPEG and International Telecommunication Union - ITU have allowed their harmonization and standardization [20]. MPEG-1 was the first codec standard, and it was completed in 1993 and it was widely deployed within Video CDs.
Its successor, MPEG-2, made its official appearance in 1995, becoming of popular usage with DVDs and digital TV broadcasting. In 2003 H.264, also known as AVC or MPEG-4 part 10, was introduced and mostly deployed for HD-TV and Internet-based video services.
The codec has been jointly created by ISO-MPEG and ITU and it represents the leading standard for many applications among different fields, comprising the surgical one.
Since compression efficiency and speed guaranteed by H.264 can no longer be substantially improved efforts are made for extending today's standards rather than developing completely new video coding methods. Therefore, the research has been directed towards pre-processing operations or layer enhancement to give customers exceptional visual quality [20], [21]. The AVC standard has been upgraded with H.265/High-Efficiency Video Coding (HEVC) which has entered the market in 2013 and it has been widely adopted for 4K and 8K, high dynamic range (HDR) and wide color gamut (WCG) video applications [20]. In the surgical domain the codec standard currently in use for dealing with data transmission is H.264 [9], [10], [18], since it is less computationally demanding then HEVC [10], thus hardware-friendly, easily implemented and distributed in its accelerated version. As a result of the algorithm optimization, small latency can be reached and it is a strict requirement for real time applications.

1.1 H.264/AVC standard codec

Since the H.264/AVC codec is nowadays considered the leading standard for surgical video compression, an overview is proposed. More in particular, the focus is placed on the encoder side, since the central decoder is standardized, guaranteeing the possibility to optimize the encoding implementations for specific applications [23].

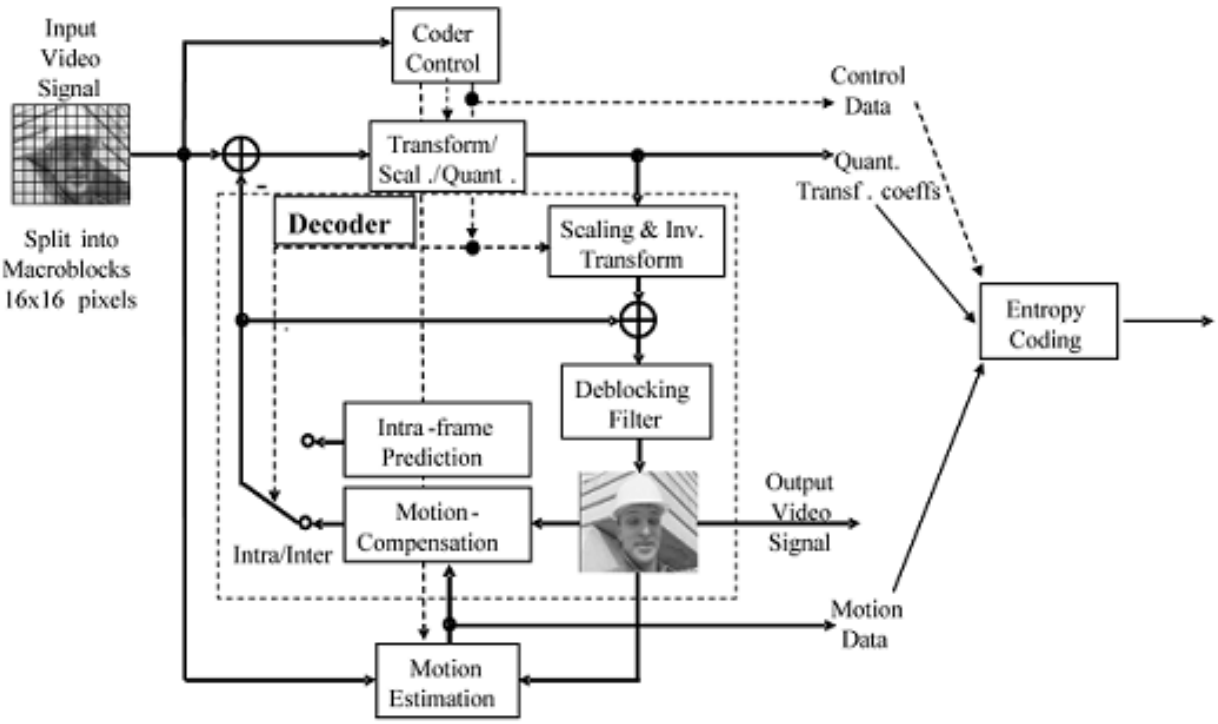


Figure 1.1 Structure of an H.264/AVC encoder

H.264, as its predecessor MPEG-2, is based on hybrid block-based scheme (Figure 1.1) which exploit both temporal and spatial prediction in combination with block-based transform coding [24].

A coded video sequences is composed by a sequence of coded pictures, which can represent either an entire frame or a single field [23], [24]. The frame can contain, in general, two interleaved fields, a top and a bottom one. Field-based coding is useful when the scene presents strong motion. It can happen that some scenarios require parts to be encoded in field mode, while others are more efficiently encoded in frame mode. Therefore, H.264 supports macroblock-adaptive switching between frame and field coding [24]. At the beginning of the encoding process, frames are partitioned in macroblocks of fixed size, which cover a rectangular picture area of 16x16 samples for the luma component and 8x8 samples each of the two chroma components, in the case of 4:2:0 chroma sampling format [24]. Luma component Y refers to brightness, while the two chroma components Cb and Cr represent the extent to which the color deviates from gray toward blue and red respectively [23]. In fact, the human visual system perceives scene contents in terms of brightness and color information separately, with more sensibility to brightness than colors.

To create a functional video transmission design that leverage this characteristic, the chroma component has one fourth of the number of samples in both vertical and horizontal direction than the luma component (4:2:0 configuration).
The previously mentioned macroblocks are grouped into slices, regions of a given picture that can be decoded independent of each other. A picture comprises the set of slices representing a complete frame or field [24].
Five slice coding types are supported by the H.264/AVC encoder [23], [24]:

- *I slice*: intra-prediction is used for the entire number of macroblocks of the slice and spatially neighboring samples of a given block already decoded are used as a reference for spatial prediction [24].
- *P slice*: macroblocks are coded in inter-prediction mode, hence prior coded images are used to form a prediction signal [24]. Each P-type macroblock is divided into sub-blocks, employed for motion compensation. The prediction signal is realized by displacing an area of the corresponding reference picture [23], [24]; the displacement is described by a translational motion vector and a picture reference index. The innovation apported by H.264 with respect to its predecessor lies in the possibility of using more than one coded picture as reference for motion-compensated prediction [24].
- *B slice:* B-types can be coded using inter-prediction employing a weighted average of two motion-compensated prediction values per prediction blocks for building the signal; in fact, B stands for bi-directional. As for P slices, H.264 allows to exploit any arbitrary pair of reference pictures for the prediction of each region [24].
- *SP slice:* the switching P slice allows efficient switching between different pre-coded pictures.
- *SI slice:* switching I slices permit to match with precision macroblock for random access and error recovery purpose [13].

In a typical encoding sequence in field mode, the first field is encoded with intra prediction, while inter-prediction is used for the second one [24]. As regarding the frame mode, the I-frame is found at the beginning and at the ending of the sequence, while the remaining part is composed by P and B-frames [25]. As for slices, I-frame is obtained by using intra-prediction, while P and B-frame are inter-coded frames. I-frames are characterized by a spatial compression made by exploiting only the information contained within that frame. P-frames and B-frames involve temporal compression; in particular, P-frame contains only the changes from the previous one, while B-frames uses differences between the current frame and both the preceding and following frames to determine its content [25].

The encoding process aim to select which samples need to be used and the way to combine them for a good prediction; the choice is finally communicated to the decoder. The *residual*, defined as the difference between the original sample and the predicted one, is transformed. The transform coefficients are then scaled and approximated through scalar quantization. A quantization parameter, which can take 52 values, is used for the purpose. The organization of these values has been established so that an increase of 1 in quantization parameter means an increase of quantization step size by approximately 12% [23]. Successively, the entropy coding process is applied and the new coefficients are transmitted together with the entropy-coded prediction information [24]. In H.264 two entropy coding configurations are supported: the exponential-Golomb code for all the syntax elements, excluding the quantized transform coefficients, for which the Context-Adaptive Variable Length Coding (CAVLC) is employed [23], [24]. A model of the decoding process is contained in the encoder, to allow it to compute the same prediction values calculated by the decoder. The decoder inverts the entropy coding process and performs the prediction using the information given by the encoder [24]. Moreover, the decoding phase involves the inverse-scaling and the inverse-transforming of the quantized transform coefficients to produce the approximated residual, which is added to the prediction. Since a peculiarity of block-based coding is the accidental production of visible discontinuities along block boundaries which can diffuse into the blocks, an adaptive deblocking filter is applied. The solution allows a reduction of the so-called *blocking artifact*, without changing the sharpness of the continent. The perceived quality is considerably improved, and the bitrate is diminished by 5%-10% [23], [24]. Finally, the decoded video is provided as output.

1.2 H.265/HEVC standard codec

The increasing demand for high compression efficiency due to a growing popularity of HD and Ultra HD videos together with the transmission needs for video-on-demand services and the necessity to handle the traffic caused by video applications for mobile devices, tablet and PCs, have pushed researchers toward the development of more performant codecs. H.265/HEVC has been designed to provide the same services of its predecessor, i.e., H.264/AVC, with an increased video resolution and an increased parallel processing architectures [26]. In HEVC, as well as in H.264, the standardization interests only the syntax and the bitstream structure, as well as the bitstream constrains and the mapping for the generation of the decoded output. In this way a certain freedom for optimization for specific applications is conferred [16]. HEVC scheme employs the block-based hybrid approach (Figure 1.2).

Although the video coding technique is similar to the one employed by its predecessor, HEVC shows brand new features aimed to improve the parallel processing capability or the ability to modify the structuring of slice data for packetization purposes [26].

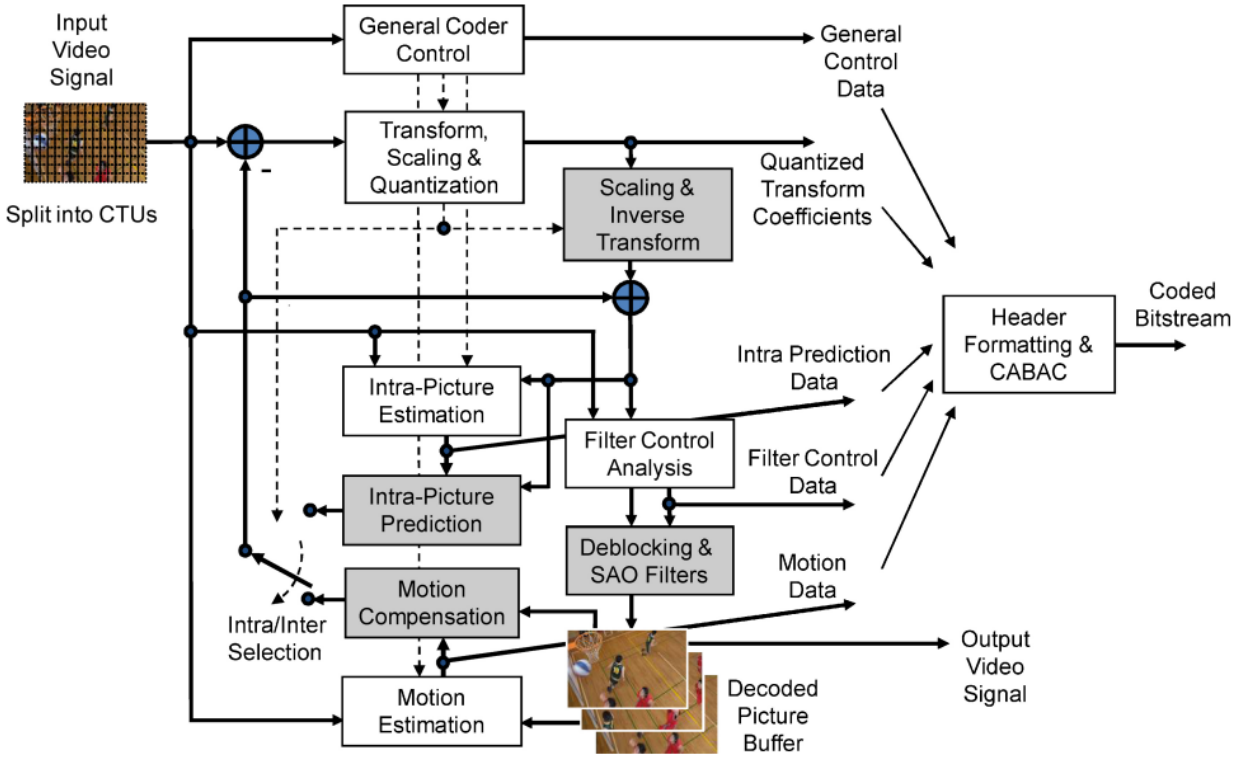


Figure 1.2 The HVEC scheme

The Coding Tree Units (CTU) constitute the principal processing units, which size is selected by the encoder according to memory and computational requirements; they substitute the macroblock used by the previous codecs. The variable size represents a novelty with respect to the other standards, which are characterized by 16 x 16 luma samples macroblocks, and it is beneficial for high-resolution video content. Each CTU is composed by a Luma Coding Tree Block (CTB), the corresponding chroma CTBs and syntax elements [26]. CTBs can be either used directly as CBs (Coding Block) or being splitter into various CBs. The partitioning is performed simultaneously for luma and chroma component and is achieved by using a tree structure. CBs are split through the quadtree splitting process, which exploit the quadtree syntax [26] contained in the CTU.

The coding unit (CU) consists of one luma Coding Block (CB) and two chroma CBs; each CTB can contain a CU or can be split in multiple CUs to which are associated partitioning into Prediction Units (PUs) and a tree of Transform Units (TUs) [26]. The prediction-type decision, i.e., inter or intra prediction, is decided at the CU level, where the PU partitioning structure has its root.
In compliance with the prediction strategy chosen, luma and chroma CBs are split and predicted from luma and chroma prediction blocks (PB). For both intra and inter-prediction, slices are a fundamental concept in HEVC; they contain a variable number of CTUs and they are introduced mainly for resynchronization after data losses [26]. In compliance with slices, HEVC presents tiles to increase the parallel processing. Tiles are independently decodable rectangular regions of a picture, encoded with some shared header information [26]. TU tree structure is designed for the transformation of prediction residual. The transformation can be computed by employing integer basis function similar to DCT or derived from the discrete sine transformation (DST), in case of luma intra-picture prediction [26]. The luma and the chroma CB residual can either coincide with the transform block (TB) or being split into multiple TBs. Intra-prediction strictly depends upon the TB size and the prediction signal is formed by previously decoded boundary samples from spatially neighboring TBs [16]. HEVC adopts three intra-coding methods, i.e., Intra-Angular, Intra-Planar and Intra-DC [26]. Inter-prediction coding follows the same strategy of H.264, thus uni-predictive and bi-predictive coding are employed. However, while AVC applies a two-stage interpolation process for fractional samples, HEVC employs a single stage procedure, increasing the precision and simplifying the architecture designed for the purpose. The interpolation of fractional sample positions is computed through 7-tap or 8-tap filters, longer than the one used in AVC. Moreover, an Advanced Motion Vector Prediction (AMVP) is introduced in HEVC. The quantization process is performed as in the previous standard, while only one entropy coding method, i.e., CABAC, is specified for HEVC. To reduce blocking artifacts a simplified blocking filter is utilized, in conjunction with a Sample Adaptive Offset (SAO), aimed to reconstruct the original signal amplitude exploiting a non-linear amplitude mapping.

1.3 Deep Learning for video compression

In recent years researchers have developed deep-learning based solutions either for implementing a brand-new end-to-end scheme to improve video quality or to increase the performance of one of the five main stages of the traditional codecs: intra-prediction, inter-prediction, quantization, entropy coding and in-loop filtering [6]. In fact, although the modules are well designed, the whole compression system is not end-to-end optimized [27].

Deep Neural Network can exploit large scale end-to-end training and highly non-linear transformations [27], which can lead to better compression performances. However, deep learning-based solutions are at their infancy since they are not trivially applied to develop an efficient compression system. The first problem lies on the generation and compression of motion information to reduce temporal redundancy.

Even though learning-based optical flow represents a possible solution, it is often not optimal for a particular video task [27], [28]. The second obstacle stands in the creation of a scheme able to optimize the rate-distortion for both motion information and residual [27].
Fukushima and LeCun have laid the foundation of research around Convolutional Neural Network (CNN) with their works [29], published respectively in 1980 [30] and 1989-1990 [31], [32]. The structure implemented have paved the way for the development of increasingly sophisticated structures to face challenges in many fields. In particular, CNN have been widely employed in computer vision and natural language processing for text classification. Moreover, they have become the state of art for many visual applications, such as image classification [33]. The structure is a multi-layered feed-forward neural network characterized by convolutional layers, which performs an operation called *convolution*. The convolution kernel slides along the input matrix for the layer, generating a feature map which provides the input of the next layer. Through convolution, the network can extract progressively more complex shapes. Convolution can be 2D or 3D, when the kernel slides in 2 or 3 dimensions respectively.

The output size of convolutional layer can be computed as:

$$\begin{cases} S_x = \frac{I_x + 2z_x - f_x}{L_x} + 1 \\ S_y = \frac{I_y + 2z_y - f_y}{L_y} + 1 \\ S_z = \frac{n_s + 2z_z - f_z}{L_z} + 1 \end{cases} \quad (1.1)$$

where: I is the input image, z is the zero padding size, f is the filter size, L is the stride length of the filter, ns is the number of slices of the image.
A typical CNN features also pooling layers and a final fully connected layer (Figure 1.3).
Nowadays, CNN have been extensively adopted for image/video compression tasks.

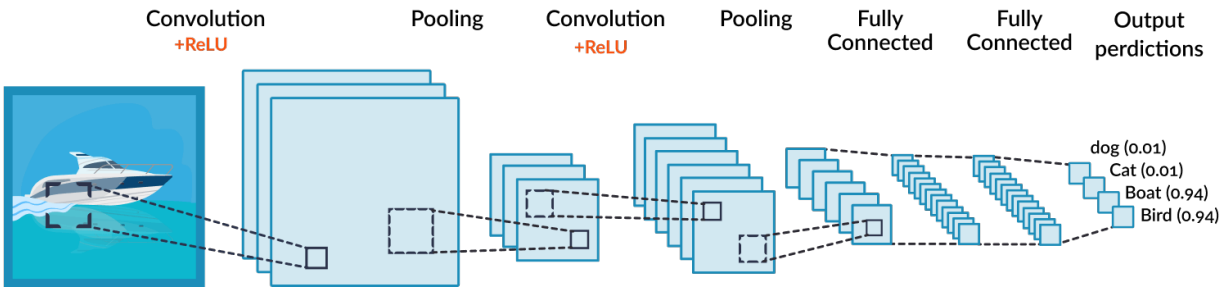


Figure 1.3 Typical Convolutional Neural Network framework [34]

The autoencoder is also considered one of the promising deep-learning structures, specifically a feed-forward non-recurrent neural network, aimed to efficiently compress and encode a series of input data (**x**) and to reconstruct the data back from the reduced and encoded ones to a representation which is as close as possible to the input (**x'**). The architecture features two functions: the encoder **E,** parametrized by ϕ, and the decoder **D,** expressed in function of θ. The first one learns intrinsic features of a dataset, which is represented by a vector. The output of the function results in a smaller vector than the one given as input. The coded input is referred $\mathbf{h=E_\phi(x)}$ is referred to as *code* or *latent variables***.** The layer which contains the ultimately compressed representation is named *bottleneck* The second function, i.e., the decoder, learns to reconstruct the compressed information, to return an output $\mathbf{x'=D_\theta(h)}$ similar to the input.

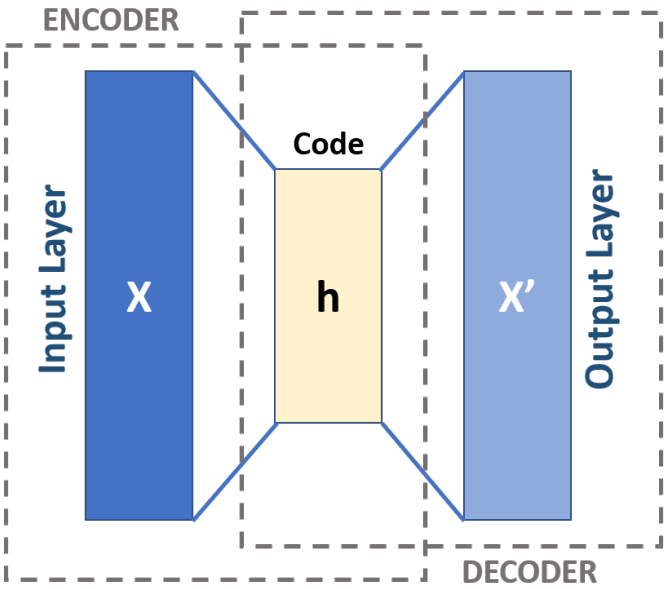


Figure 1.4 The structure of a simple autoencoder

As shown in Figure 1.4, the input $\boldsymbol{x} \in \boldsymbol{R^d}$ is mapped to $\boldsymbol{h} \in \boldsymbol{R^p}$ such that $\mathbf{h} = \sigma(\mathbf{Wx} + \mathbf{b})$. σ is the activation function, **W** is the matrix of weights and **b** is the bias. The last two elements are randomly initialized and then iteratively updated during training through backpropagation. The decoder takes as input **h** and reconstruct a x same-shaped vector $\mathbf{x'} = \sigma'(\mathbf{W'h} + \mathbf{b'})$. Notice that σ',**W'** and **b'** can be unrelated to the corresponding encoding values. The autoencoder is trained to minimize the *reconstruction error* or *loss*, which can be computed employing different functions, e.g., Mean Square Error (MSE) and Cross Entropy.

Going into detail, many authors have been interested in developing deep-learning structures to address the challenges of video compression.
Lu et al. [27] have proposed an end-to-end deep video compression (DVC) scheme (Figure 1.5) which jointly optimizes the key components of video compression - *i.e.*, residual compression, motion compression and estimation, quantization and bitrate estimation. The implemented scheme minimizes the reconstruction error through a single loss function while reducing the bits required for compression, and it is one-to-one mapped with the components of traditional video compression techniques [27].

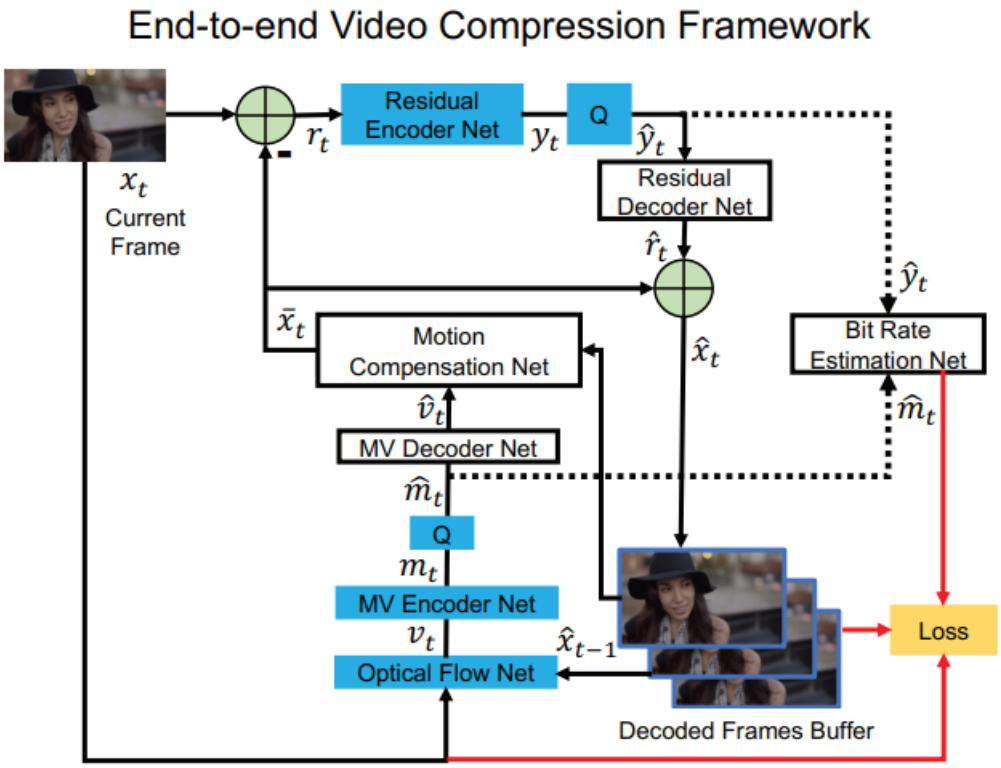


Figure 1.5 The video compression pipeline proposed by Lu et. all [27]

More in detail, motion compression and estimation are performed by compressing and decoding the optical flow values through a MV encoder-decoder network, instead of directly encoding the raw ones. Motion compensation is achieved through another Convolutional Neural Network, exploiting a pixel-wise method which provides more accurate temporal information and avoids blocking artifacts [27] , meaning that no deblocking filter is required.

The residual is transformed by a high non-linear residual encoder-decoder network, which substitutes the linear transform.
Quantization operation is not differential, hence it cannot be applied to an end-to-end training; to address the problem quantization has been replaced by adding uniform noise during the training phase. For the bitrate estimation the CNNs of [35] have been employed. The frame reconstruction is obtained by adding the predicted frame to the reconstructed residual and by passing the result through the decoder. The proposed scheme has been demonstrated to outperform the H.264/AVC video compression standard [27].
The method proposed by *Lu et. al* aims to exploit deep learning techniques for the whole compression scheme, i.e., the five main modules. Other authors, instead, have concentrated their efforts on the implementation of a single stage of the system; in particular, solutions have been proposed for the intra/inter prediction and for the post-processing filtering techniques.
Li et. al [36] have developed CNN-based block up-sampling scheme for intra-frame coding. In detail, the learning-based approach performs an up-sampling of different regions of the frame, instead of compressing the entire sample, since pictures are equipped with locally variant features, hence different parameters or coding methods are required [36]. The basic unit for down/up-sampling is the Coding Tree Unit (CTU), in compliance with HEVC. The up-sampling strategy is based on the employment of both CNN and Discrete Cosine Transform Based Interpolation Filters (DCTIF), for dealing respectively with complex image regions (e.g., structures) and fractional pixel interpolation for motion compensation. Two different five layers CNN are developed for luma and chroma up-sampling.

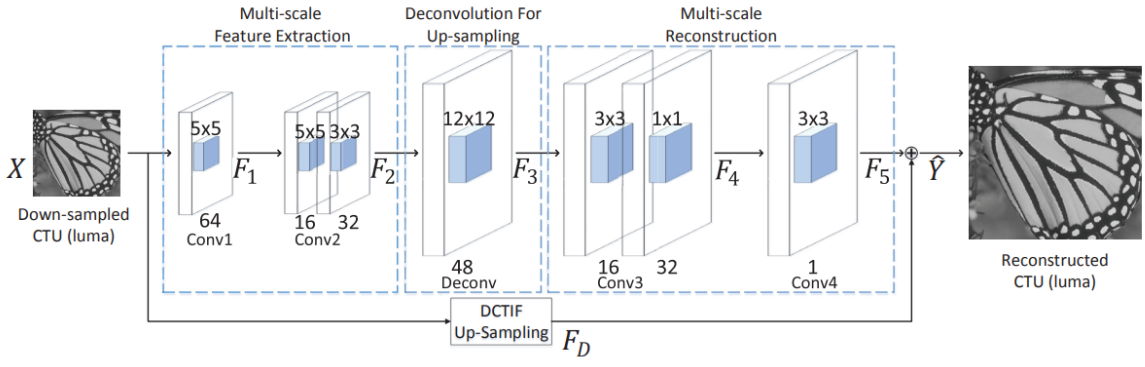


Figure 1.6 The luma up-sampling scheme. The kernel size is indicated by the numbers at the top of the picture, while the number of the output channels are reported at the bottom [36].

The luma up-sampling system (Figure 1.6) features four stages: multi-scale feature extraction, deconvolution, multi-scale reconstruction and residue learning [36].

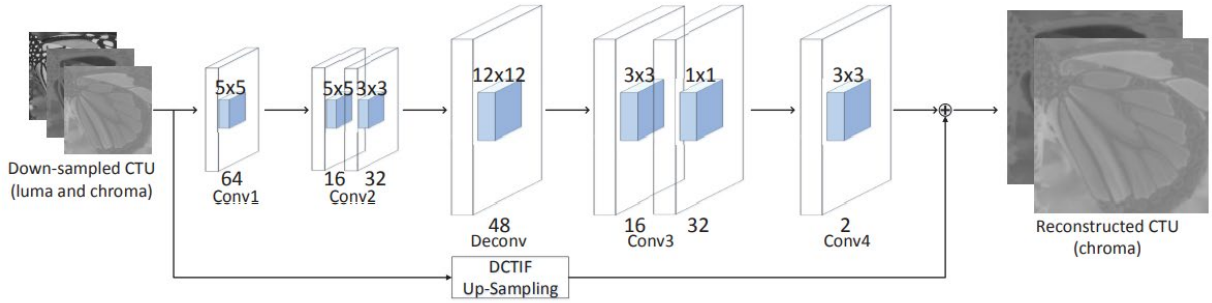


Figure 1.7 The chroma up-sampling scheme [36].

The chroma up-sampling scheme (Figure 1.7) is similar, but presents two more features, i.e., the prediction of chroma from luma and a joint training of the chroma components, to improve the reconstruction quality. The designed structure is implemented based on the HEVC reference software and is shown to achieve a significant bit saving at low bitrates when compared to the traditional encoding system [36]. On the other hand, compression noise due to the dependency of the CNN from the Quantization Parameters (QPs) used in compressed training videos has been highlighted in some cases. Moreover, the CNN encoding/decoding time was significantly higher when compared to HEVC (although without any optimization for speed on the CNN side [36]). In 2019 the same authors have proposed an extension of the scheme for inter-frame prediction [37]. Here, reference frames are employed by a trained CNN model to improve the up-sampling of the current frame, exploiting the temporal correlation. As in their previous work, the scheme has been implemented referring to HEVC software and it has shown better performances than the traditional codec for high-definition videos compressed at low bitrates [37].

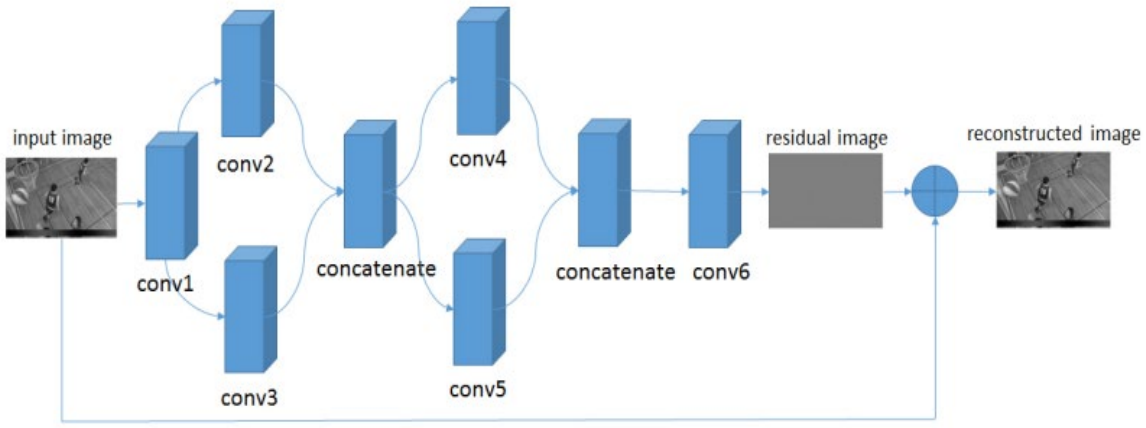


Figure 1.8 The VRCNN pipeline

To reduce artifacts in HEVC intra coding, *Dai et. al* have built a Variable-filter-size Residue-learning CNN (VRCNN) [21], aimed to replace the two post-processing systems in the traditional codec (Figure 1.8). It presents a 4 layers configuration, which features a combination of 5 x 5 and 3 x 3 filters in the second layer, to better reduce the noise given by the quantization. The third layer, which handle the feature restoration, also presents 3 x 3 and 1 x 1 filters. Fixed filters are used for the first and the last layer, because they perform respectively feature extraction and final reconstruction, which are not affected by variable block size transform in HEVC [38]. The processed image by the presented scheme suppresses both blocking and ringing, offering a better visual quality with respect to HEVC; moreover, a significant bitrate reduction is achieved. However, the decoding time does not satisfy the requirement for real time applications and, besides, is highly memory demanding.

Wang et. al [39] have designed a neural network-based enhancement to inter prediction (NNIP) to improve the coding performance at the inter-prediction level. The scheme (Figure 1.9) is composed by a residue estimation network, a combination network and a deep refine network.

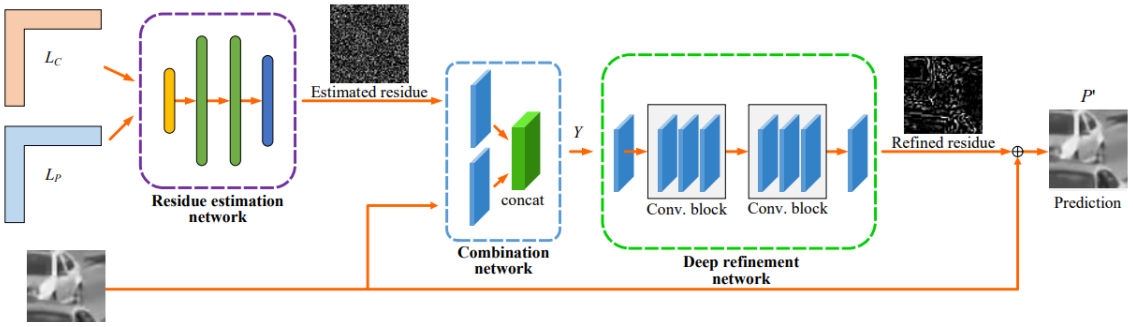


Figure 1.9 The framework proposed by Wang et. all to improve the inter-prediction [39]

The first one aims to estimate the residue between current block and its predicted block using available spatial neighbors [39]. The combination network concatenates the feature maps of the estimated residue and the predicted block together, once extracted. The last network is designed to derive the refine residue, which is added to the predicted block. Although the system overcomes HEVC in terms of quality, the computational complexity leads to a significant increasing in the encoding and decoding time. To refine motion compensation in video coding, *Huo et. al* [40]have studied a CNN-based motion compensation refinement (Simple CNNMCR) scheme to enhance the prediction signal directly, based only on temporal features. Furthermore, a CNN which exploits both temporal and spatial correlation to improve the prediction accuracy is proposed (CNNMCR). More in details, the CNNMCR is structured as the VRCNN [38] previously described, but it is trained to leverage the information extracted both from motion compensation and the already reconstructed region of the current frame to refine the prediction signal [40]. The implemented scheme has shown better quality performances then HEVC, with 1.8-2.3% bitrate saving.

Fractional interpolation for inter-prediction has been widely employed among the traditional codecs [41]–[43] to remove the temporal redundancy in consecutive frames. The method adopts filters, either fixed or adaptive, to generate fractional samples from integer pixel values in a reference picture. In recent years, various CNN models have been presented to further improve the fractional interpolation efficiency [26]–[28]. *Zhang et. al* [41] have designed a Compression Priors assisted Convolutional Neural Network (CPCNN).

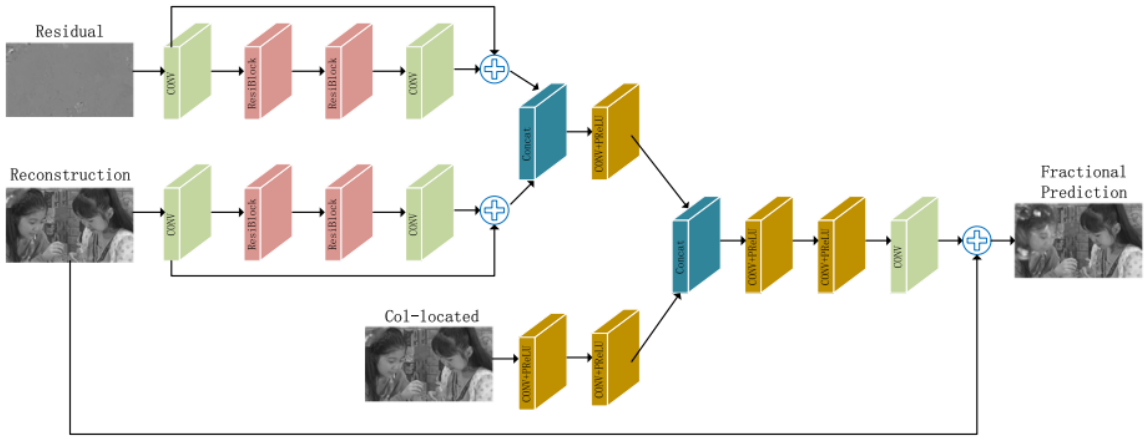


Figure 1.10 The CPCNN framework: a unique architecture with three branches. The feature maps are extracted from these components separately and then combined together to derive the final output fractional prediction [41].

Three sources of information have been inputted into the scheme (Figure 1.10): the reconstruction component of reference block, used as the integer reference to generate the interpolated samples, the corresponding residual component, that indicates the prediction efficiency and contains effective texture information, and the co-located high quality component, employed to reduce quality fluctuations. The first source is used as the only input in traditional methods. The second component has been added as a complementary tool to handle the hard-predictable areas, e.g., sharp edges zones and complexity textures, which result in non-zero residual. The residual prior allows a better detection of the non-zero residual areas and further reduces the noise in the reconstructed components. The third prior has been designed to deal with quality instability derived from quantization, providing high quality information to improve the real reference reconstruction and introducing no-local information for better interpolation performances. Compared to HEVC, the scheme has been demonstrated to achieve superior performances in terms of bitrate saving. *Yan et. al* have depicted the fractional-pixel motion compensation as an inter-picture regression problem [42], which can be well handled by CNN. In fact, they have proposed fractional-pixel reference generation CNN (FRCNN) for both uni-directional and bi-directional motion compensation. A first CNN set, corresponding to different fractional pixel location, has been trained for uni-directional motion compensation. Following, another CNN set has been introduced to enhance the first set for bi-directional motion compensation.

Zang et. al [43] have considered the interpolation problem as a generation task and have designed a dual-input CNN-based structure (Figure 1.11) to exploit the real integer position samples at the reference block to predict and generate fractional samples [43].

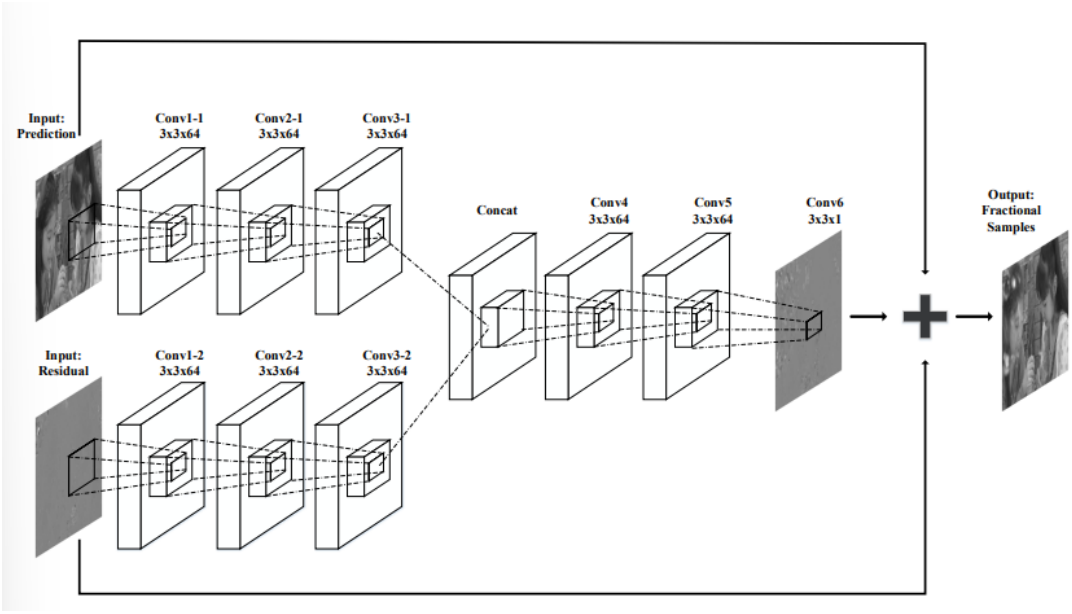


Figure 1.11 The dual-input-scheme pipeline

More in detail, the proposed approach involves a six-layer fully convolutional network which receives prediction and residual input. The sources are initially handled by means of separate convolutional layers. The extracted features are eventually concatenated to form the input of the following layers, interested by non-linear operations. Moreover, the residual learning strategy has been adopted to get the final output. Therefore, the final result is derived by adding the two inputs (prediction and residual) together with the output of the 6th layer.
In the inter-prediction module bi-prediction represents a fundamental step in the state-of-art video coding standard, i.e., H.264 and H.265, aimed to perform motion compensation exploiting two predictive signals. The assumption of a linear pixel-to-pixel correspondence which lies at the basis of the traditional codecs is limiting, for complicated motion scenarios cannot be well handled [44]. Therefore, a non-linear fusion between prediction blocks is recommended. Deep learning provides various non-linear activation functions, which allow a better prediction accuracy. In [44] the authors have designed a six-layer CNN to deal with the prediction error caused by irregular motion. The first layer takes as input a tensor with two channels, each corresponding to one of the prediction blocks. The last layer performs the prediction of the current block exploiting a non-linear combination. Each convolutional layer is characterized by a 3 x 3 spatial shape. ReLu is adopted as activation function for the first five layers.

As the network becomes deeper, the long-term memory is required to generate a reasonable output. However, it can lead to exploding or vanishing gradient, thus a skip connection is added to deal with the problem. The proposed method has been demonstrated to enhance the regions that cannot be well handled with traditional bi-prediction, such as the boundaries of a moving object [24]. Moreover, an important bitrate reduction has been achieved.

Bi-prediction improvement through deep-learning methods has been also presented by *Mao et. al* [45]. The authors have created a CNN model fed with spatial neighboring pixels of both current block and two reference blocks, other than the two reference blocks themselves. The first input provided aims to estimate the similarity between current block and reference blocks [45]for a more accurate prediction and a reduction of blocking artifacts. To deal with blocking artifacts specifically post processing techniques, i.e., performant filtering systems have been introduced in traditional codecs to. Although the performances have been boosted, the quality degradation still remains a problem to be handled. For this reason, many deep-learning solutions have been introduced in recent years.

In [46] a deep learning-based design has been proposed for the post processing, aimed to enhance quality of the reconstructed frames without conflicting with the deblocking filter and the sample adaptive offset typically found in HEVC. In fact, the residual highway convolutional neural network (RHCNN) is added to the filtering systems of the codec. To achieve better performances and robustness of the network, the entire quantization parameters range is divided into bands for which a RHCNN each is trained following a progressive training scheme [46].

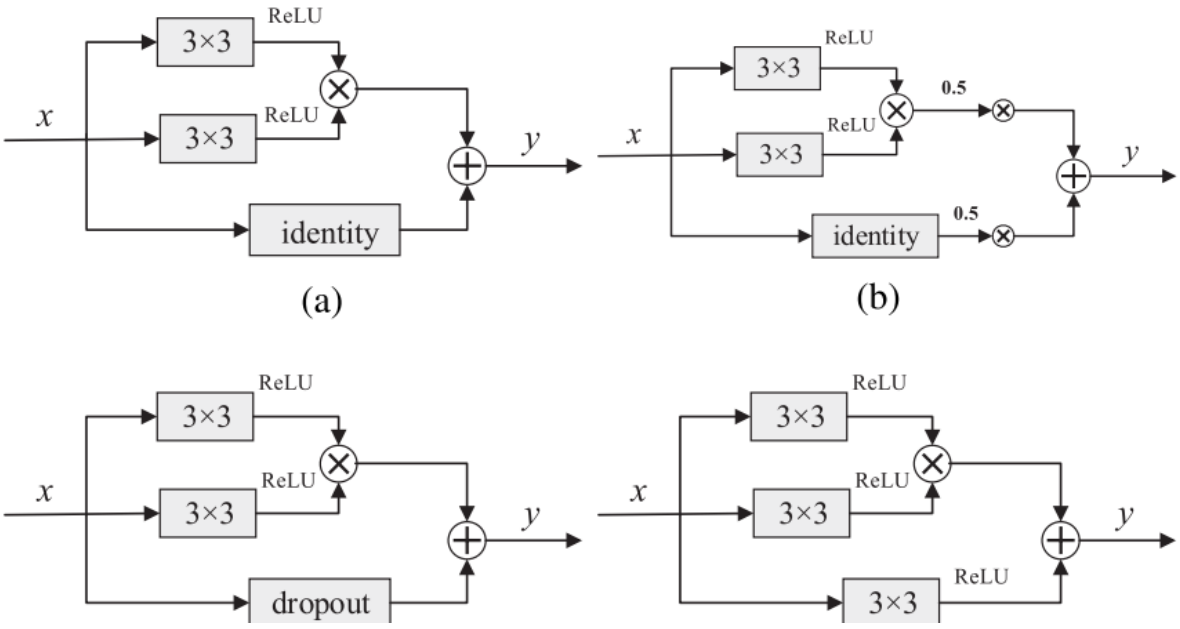


Figure 1.12 The highway units present in the scheme

The network designed is composed by several cascade highway units, convolutional layers and identity skip connections (Figure 1.12), to build a very accurate mapping between the distorted reconstructed frames and their corresponding original distortion-free ones [46]. The highway units constitute the main components of the RHCNN; each one of them is preceded by a convolutional layer, to guarantee that unit state and transform feature maps are the same size as input [21]. The network is composed by 12 layers with a spatial size of 3 x 3, followed by a closing 1 x 1 convolutional layer, which is fundamental for improving accuracy. Among the structure identity skip connections (shortcuts) are placed, for better handling the gradient vanishing and recovering clean images. It is noteworthy that shortcuts do not introduce computational complexity nor extra parameters. This framework achieved substantial coding gains, especially for low bit rates, but the encoding time heavily increased [6].
Improving one of the modules of traditional codecs with deep learning-based techniques is a valid solution, but it remains the doubt around the possibility of further significantly improving the compression performances. For that reason, many researchers have focused on the implementation of brand-new schemes.
Feng et. al's work aims to reduce compression artifacts in a low-bitrates and Super Resolution (SR) scenario by placing an enhancement network [47] ahead the CNN-based Super Resolution, in combination with a purpose-modified geometric self-ensemble strategy [48] for the improvement of the SR performances (Figure 1.15).

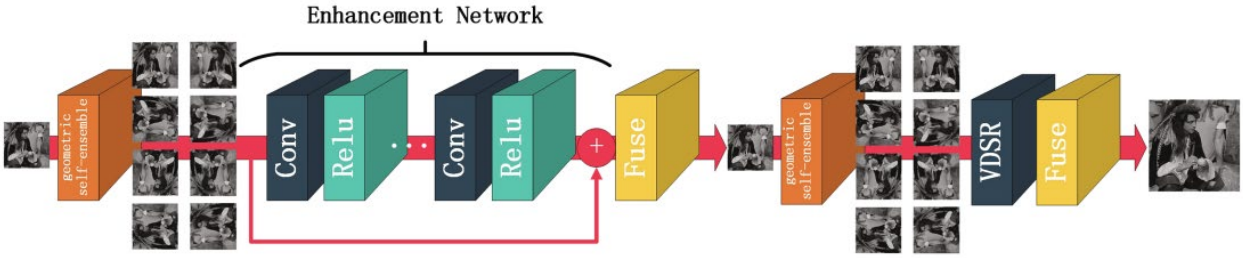


Figure 1.13 The pipeline of the dual network structure: the enhancement network is introduced before the VDSR, employed to manage the Super Resolution [47].

Both enhancement and Super Resolution can be regarded as a regression problem [47]; therefore, to handle the compression degradation a CNN with 20 layers which takes the rectified linear unit (ReLu) as activation function is employed. To cope with the Super Resolution problem, the VSDR structure is introduced right after [49].

Experimental results have shown a 31.5% bitrate saving for 4K video compression with respect to HEVC, when applied in a SR-based video coding scheme [47].
Han et. al have designed an end-to-end deep generative structure, built upon the Variational Autoencoder model [1], [50] to parametrize the transformation to and from the encoder and the decoder. In particular, a sequence of frames is transformed into a group of latent states and a global state. In addition, an entropy coding has been applied to remove the redundancy in latent space variables. The model learns has been implemented to learn simultaneously the predictive model required for entropy coding and the optimal lossy transformation into the latent space [1]. The scheme has shown to outperform the traditional codecs, achieving superior image quality at a significantly lower bitrate [1]. Although in this regime blurry video tend to be generated, block artifacts are not present.
In [51] a 3D Rate-Distortion Autoencoder has been created for lossy video compression. The latent variable model has been trained to capture the important information to be transmitted, allowing the reconstruction of the input. Both encoder and decoder are fully convolutional structure, with residual connections, batch-norm and ReLu activation function. Moreover, an autoregressive prior is added ahead. Despite the simplicity of the structure, the scheme has achieved comparable performances with respect to HEVC [51].
An interesting framework has been proposed by *Tsai et. al* [52], that employs both H.264/AVC and a deep learning-based technique for domain-specific video streaming, to achieve a better video quality and low-latency transmission (Figure 1.16).

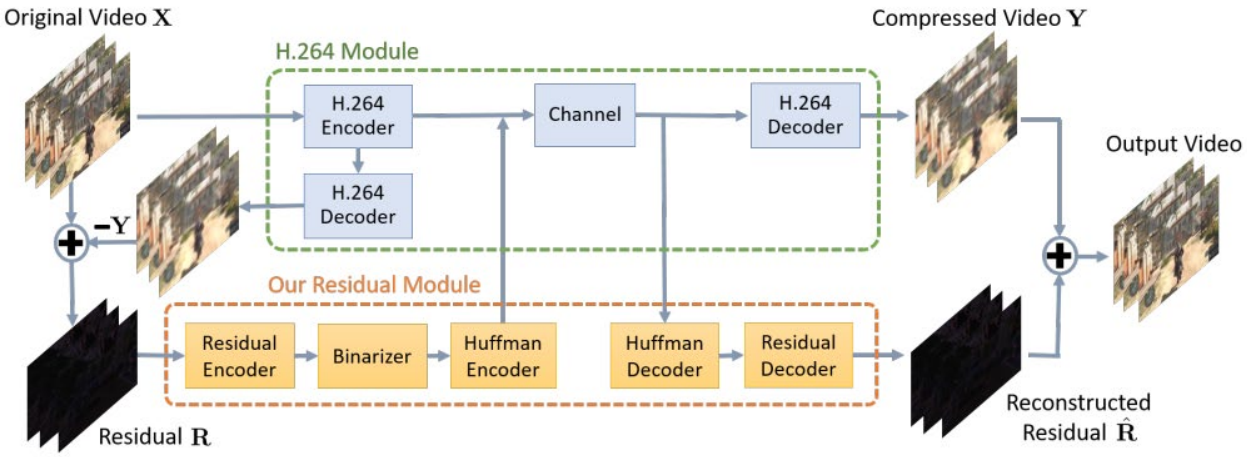


Figure 1.14 The pipeline proposed for domain-specific video streaming consists of two modules: the H.264 scheme and the Residual Autoencoder.

In particular, domain-specific videos are compressed by the AVC standard, while the leftover residual information is encoded into a binary representation by a specifically trained autoencoder. Moreover, the Huffman Coding performs a further lossless compression on the residual. These representations are eventually delivered to the client together with the AVC stream [52]. The solution shows better performances both in terms of quality and runtime in comparison with H.264.

1.4 Deep Learning for endoscopic video compression

Even though many studies have been conducted over the employment of deep learning-based architectures for video compression among different fields, their use in the surgical domain is still at its infancy. In fact, as the survey conducted by *Unzer et. al* [53] demonstrates, the major efforts of researchers have been directed toward the implementation of deep learning strategies for surgical image analysis, surgical task analysis, surgical skill assessment, and automation of surgical tasks in Minimally Invasive Surgery (MIS) [43]. Surgical video compression for real-time applications through DL architectures remains a challenge faced by few.
It has been proved that the detection of clinically relevant spatio-temporal information is crucial for saving compression time, while maintaining quality during the transmission process. To the purpose, deep learning-based solutions have been implemented for the input frame segmentation and the recognition of the Regions Of Interest (ROI), which requires the quality preservation. A flexible and interactive ROI-based video coding scheme for low-bandwidth scenario has been proposed by *Khire et. al*[54] In this work, more bits are assigned to the ROI decreasing the Quantization Parameters (QP), since they are inversely proportional to the bitrate. *Munzer et. al* [55] have highlighted relevant features and irrelevant contents, i.e., dark frames, out-of-patient frames and blurry frames, in endoscopic videos. In [56] the authors have proposed a relevance-based compression approach using two different CNN integrated to HEVC to compress cataract surgery videos. In detail, the first CNN, namely static frame-based CNN, has been developed to detect temporal regions where no instrument is visible [56] , i.e., idle phases. The second network, specifically a region-based CNN (Mask R-CNN), has been implemented to automatically locate the relevant spatial regions. In [57] a Shallow Convolutional Neural Network (S-CNN) based segmentation approach has been tailored to compress in high quality surgical incision regions, while the background has been handled by using a lossy technique. The scheme has been demonstrated useful for real time applications in a limited bandwidth scenario [57] Surgical regions are identified by the S-CNN and then transmitted by employing low QPs, that correspond to high quality output.
On the contrary, high QPs values have been used for the background. A significant bitrate reduction of 88.8% has been achieved in comparison with HEVC.

2 Materials and methods

In this section is presented an overview of the framework proposed to deal with the need of high compression quality under latency constrains in surgical domain. In particular, the work is focused on endoscopic video compression for remote surgery.

It is decided to employ the network implemented by *Tsai et. all* [52] as it can well compresses domain-specific videos in a low bandwidth scenario, thus shows suitable features for the task. The section describes the technicalities of the scheme, as well as the dataset used, the training of the residual neural network and the experiments conducted to analyze its performances both in terms of quality and speed.

2.1 Binary Residual Neural Network for RARP video compression

Telemedicine applications require advanced compression system to deliver contents both in high quality and real time. Thus, such schemes need to guarantee superior performances in a low-bitrate and low-latency scenario. Deep learning-based techniques have been shown to achieve high-level performances, overcoming the ones of the traditional codecs. Therefore, the research consisted in developing and training a neural network for real time stream of data in the surgical domain, guaranteeing the preservation of high quality. For the aim, the scheme proposed by *Tsai et. al* [52] has been employed. The framework features two modules: a video compression module, i.e., H.264/AVC, and a deep learning-based autoencoder. The traditional scheme compresses surgical videos, while the neural network is employed to recover the lost information during compression on the client side, exploiting the fact that videos belong to the same domain (Figure 1.16). More in detail, H.264 compressed the input **X** and the output **Y** is obtained. The residual **R** is the result of the difference between the input and the output, thus **R = X – Y**. The residual cannot be compressed employing a traditional method because is highly nonlinear, thus an autoencoder has been developed for the purpose. The deep learning scheme consists in three functions: the encoder **E**, which performs down-sampling, the binarizer **B** that maps the residual into binary values, and the decoder **D** that recover the residual information from the binary map. The compressed residual **R'** is eventually combined with **Y**, thus the final output **Y + R'** is obtained. The transmission of the residual bit stream necessitates additional bandwidth, but the autoencoder can be trained to reduce the bandwidth requirement.

The choice to employ what *de facto* can be defined as a Binary Neural Network lies upon the capability of the structure to largely saving storage and computation [46]. Over the years, researchers have proposed a variety of algorithms to optimize the promising technique. Binarization consists in transforming both weights and activation in 1-bit values at run time. At the training time these binary quantities are employed for the gradient computation [59].

In the scheme (Figure 2.1) proposed by *Tsai et. al* [52] the autoencoder features three components: the encoder **E**, the binarizer **B** and the decoder **D.**

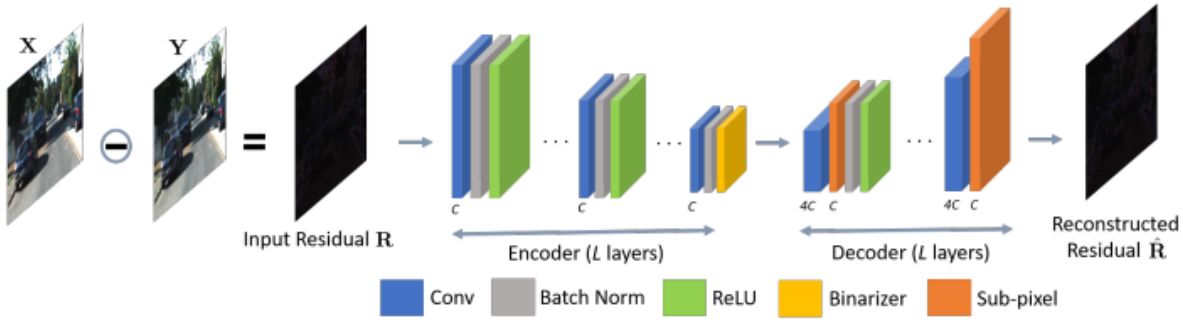


Figure 2.1 The pipeline of the proposed autoencoder for the residual compression: the number of layers is equal to 3 for both encoder and decoder. The number of channels corresponds to C = 32 for the encoder and to 4xC for the first two layers of the decoder [52].

The encoder is composed by L 2D-convolutional layers, with equal number of channels C, characterized by a stride of 2, which performs the down-sampling. The decoder consists of L 2D-convolutional layers, each one of them followed by a SubPixel layer [60], [61] featuring an upscaling factor of 2; the convolution process and subpixeling are jointly employed for up-sampling. In this case, the number of channels used in the first two convolutional layers is equal to 4xC, due to the presence of the SubPixel layer. For the same reason a stride of 1 is employed. The last layer of the decoder, instead, presents 12 output channels C Both the encoder and the decoder present ReLu as activation function:

$$f(x) = \max(0, x) \; (2.1)$$

An additional operation, i.e., batch-normalization [62] with a momentum of 0.999 is also included to facilitate the learning process.

The kernel size is set to 2 for the convolution operations during encoding, while is set to 1 for each convolutional layer of the decoder. The binarization phase features two stages: a mapping of each element of the encoder output e_i to the interval [-1, 1] and a following discretization to {-1,1} through the activation function σ and the discretization function b:

$$B(e_i) = b\big(\sigma(e_i)\big) \quad (2.2)$$

The binarization function selected is *hardtanh*:

$$b(z) = \begin{cases} 1, & if\ z > 1 \\ z, & if\ -1 \leq z \leq 1 \\ -1, & if\ z < 1 \end{cases} \quad (2.3)$$

With $z = \sigma(e_i)$.

The size of the binary map strictly depends upon the width W and the height H of the input image as well as on the number of channels C and the layers L which characterized the neural network. Specifically, the size is given by:

$$S = \frac{C \times W \times H}{2^{2L}} \quad (2.4)$$

A deeper neural network corresponds to a smaller the binary map; intuitively, the compression task would be easier, while the training would be harder. Therefore, C and L have to be carefully chosen to guarantee a good trade-off between these two processes. Based upon the studies developed by *Tsai et.al* [52], the number of channels C is set to 32, while the number of layers is set to 3.

2.2 Dataset

For the aim of the work, five high-quality videos with the endoscopic view captured during RARP (1280 x 720) are downloaded from YouTube. The video duration ranges from 72 up to 100 minutes. The first video (*Video A*) is used for *testing*, while the other three (*Video B, C, D, E*) are used for *training*.

RARP includes three main phases. The first one is the pelvic lymphadenectomy, where the focus is concentrated at the level of the iliac vessels; in this phase, the most delicate structures in the centre of the field are the blood vessels that must be freed from the lymph nodes; the surgical field is small, the surgical movements are slow and more delicate. In the following step, called "demolition phase", the prostate is isolated posteriorly from the bladder, from the nerve bands laterally and anteriorly from the urethra; here the surgical field is wider, movements are faster and the organ of interest, the prostate, is in the centre of the visual field; the peripheral area is occupied by the iliac vessels laterally and by the pubic bone over. In the last reconstructive phase, the bladder neck is sutured to the urethra; the surgical field is tight since the anastomosis between bladder and urethra is performed in the small pelvis; movements are small and mostly in the centre of the surgical field.

To highlight the different phases of the procedure from each video, ten 40 seconds clips are selected. They include different anatomical sections, surgery instruments, levels of illumination and degrees of action performed in the surgery field (Figure 2.2).

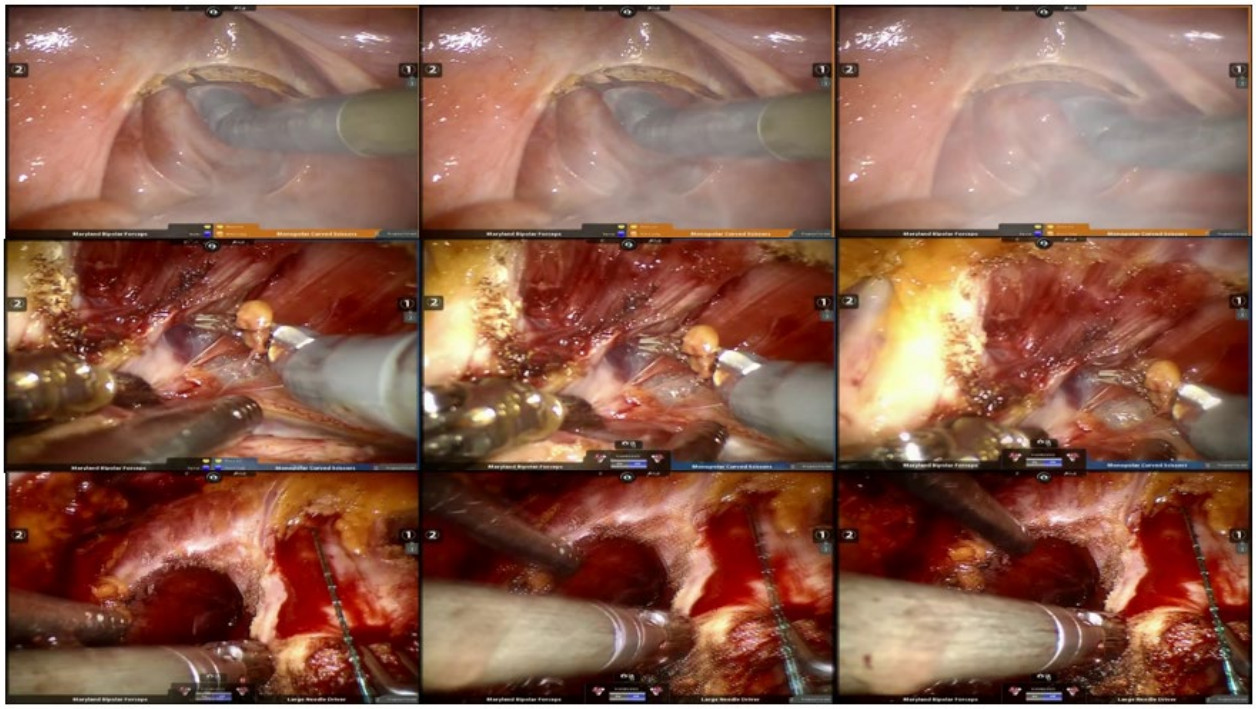


Figure 2.2 The three sequences extracted from different clips of the testing video (Video A) show three distinct events during RARP: smoking, rapid movements, bleeding.

From each videoclip is extracted one frame every ten (Figure 2.3), thus each testing dataset – one for each bitrate/preset configuration (see the following section) – is composed by 1216 frames, while the training/validation dataset– one for each bitrate/preset pair (see the following section) - is formed by 4802 frames (70% training set/30% validation set).

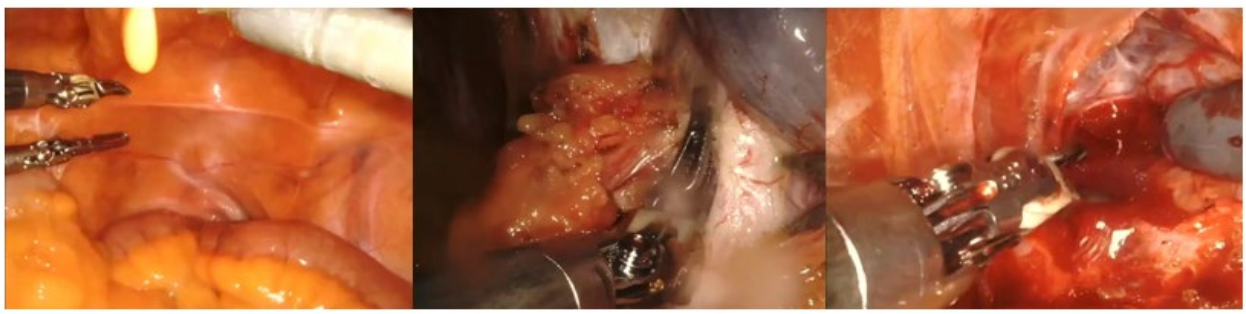


Figure 2.3 An example of extracted frames

2.3 Training the residual neural network

The autoencoder is implemented in Python, using the PyTorch library [63]. The training hyperparameters, i.e., learning rate η, batch size and number of epochs, are set respectively to 0.01, 5 and 50. Moreover, the learning rate is reduced by a factor of 0.5 every 5 epochs; in fact, decreasing the learning rate during training can lead to improved accuracy and reduced overfitting of the model [64]. Mean Square Error (MSE) [65] is utilized as loss function, while Adam is employed as optimizer [64]. The training is performed by using the NVIDIA GeForce GTX 850M GP.

2.4 Performance evaluation

The clips are extracted and eventually compressed and decompressed by using the H.264 implementation provided by FFmpeg [67], with a particular focus on bandwidth and latency, both dependent on the bitrate and the preset selected. The *FFmpeg preset* represents the coding speed value [68], thus returns a certain compression ratio / frame quality / compression time. More specifically, some preset is designed to compress the frame in a short amount of time (low latency) at the cost of decreased quality and larger bandwidth, while others achieve the highest compression rate and frame quality but require more processing time. For the aim of the research, it has been employed *Ultrafast, Medium* and *Slow* presets. The first one allows for a very fast compression, but it results in a low-quality profile. The medium preset represents a good trade-off between speed and quality, while the last preset results in better quality at the cost of high computational time. Compressing at different bitrate allows to investigate the codec performance as a function of the transmission bandwidth. In this work the first evaluation is conducted employing three bitrate values, i.e., 1,2,5 Mb. Each bitrate/preset pair – thus 9 configurations - is analyzed. A further assessment is proposed based on the results obtained from the first analysis. In particular, the 10 Mb-*Ultrafast* configuration is eventually explored. For each clip and each bitrate/preset pair of the video A the Peak-to-Noise-Ratio (PSNR) [69]–[71]and the Structural Similarity (SSIM) [70]–[72] are computed to determine the quality of the compression performed by H.264, employing the original, uncompressed data as ground truth. More in detail, one frame every ten is extracted from each clip of the testing video, either the original and the compressed one, and the metrics are computed for each original-compressed couple of images. Furthermore, the mean and the standard deviation are calculated.

Moreover, the encoding and the decoding time are measured for a single frame which indicates latency, once summed to the transmission time. Successively, the residual **R** is calculated for each original-compressed paired frame of the testing video and encoded/decoded by the trained autoencoder. The training is performed for each bitrate-preset pair, employing the images obtained from the compressed video B, C, D and E. Each compressed residual **R'** is eventually summed to the correspondent H.264 compressed frame. Finally, SSIM and PSNR are calculated for each paired original-reconstructed image of the testing dataset and both mean and standard deviation are measured for the two metrics. The mean encoding/decoding time normalized on the number of frames is also computed and summed to the encoding/decoding time featured by H.264 only.

PSNR, SSIM as well as encoding and decoding time for each configuration are reported on a plot in function of the bitrate, expressed in Bit-Per-Pixel (BPP), computed as:

$$BPP = \frac{80000 \times Bitrate\ (Kbs)}{H \times W \times fps} \quad (2.5)$$

Where *fps* corresponds to *frame per second*, thus it indicates the framerate. In this work, it is set to 29.9. H and W are respectively the height and the width of the video. The bitrate needs to be expressed in Kb.

3 Results

In this chapter are reported the results both in terms of quality and time obtained with the method proposed; the findings are compared with the performances of H.264/AVC.

3.1 Quality

The following plots and tables report the PSNR and SSIM values, employed as a measure of quality. First the performance of H.264/AVC and the deep learning-based scheme (AE + H.264) are presented separately, to allow a better visualization of the standard deviation associated to each value, the variability of results is highlighted (Figure 3.1).

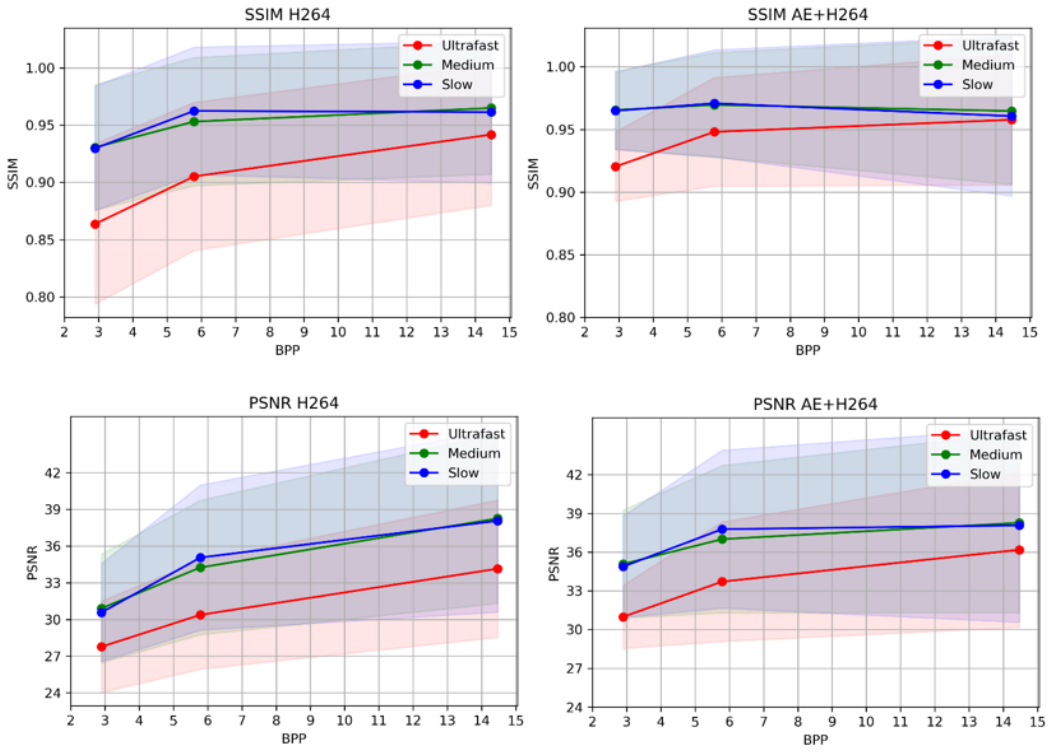


Figure 3.1 The plots represent quality in terms of SSIM (top) and PSNR (bottom) in function of the bitrate both for H.264 (left) and for the scheme proposed (right). For each configuration also the deviation standard is computed and represented.

Successively, a comparison between the quality achieved by the leading standard H.264/AVC and the scheme proposed is presented (Figure 3.2, Table 3.1, Table 3.2).

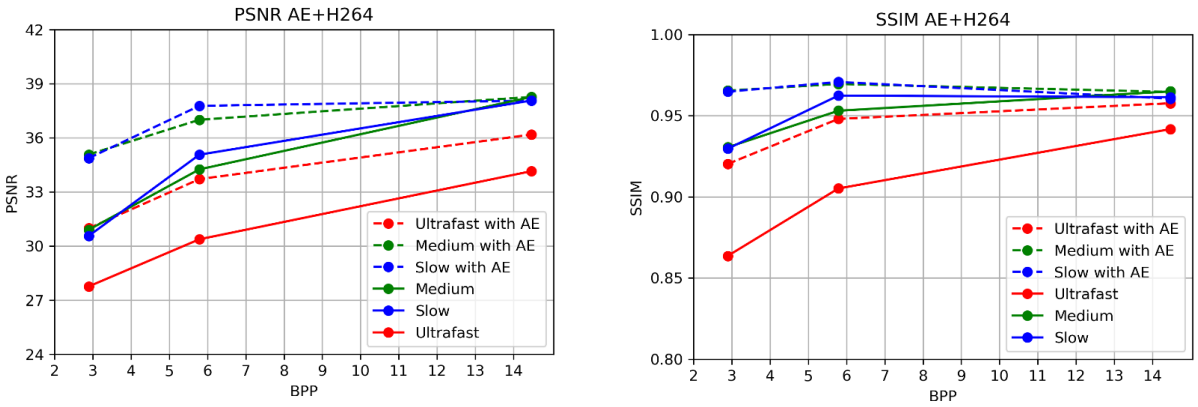


Figure 3.2 The plots show a comparison between the codec standard H.264 and the proposed scheme in terms of quality. Both PSNR (right) and SSIM (left) are expressed in function of the bitrate.

PSNR

Table 3.1 The table shows PSNR values for both H.264 and the proposed method

Preset/Bitrate (Mb)	**1**		**2**		**5**	
	H.264	H.264 + AE	H.264	H.264 + AE	H.264	H.264 + AE
Ultrafast	27,760	30,984	30,375	33,710	34,152	36,181
Medium	30,900	35,073	34,251	37,001	38,267	38,270
Slow	30,560	34,872	35,064	37,772	38,068	38,064

SSIM

Table 3.2 The table shows SSIM values for both H.264 and the proposed method

Preset/Bitrate (Mb)	**1**		**2**		**5**	
	H.264	H.264 + AE	H.264	H.264 + AE	H.264	H.264 + AE
Ultrafast	0,864	0,920	0,905	0,948	0,942	0,958
Medium	0,931	0,965	0,953	0,970	0,965	0,965
Slow	0,930	0,965	0,962	0,971	0,961	0,961

In terms of quality the developed structure outperforms the traditional standard H.264/AVC in a low bitrate scenario. More in detail, the quality shows an increasing trend within 1 and 5 Mb for videos compressed using the medium and the slow preset, thus H.264/AVC performs better at bitrates higher than 5Mb. Moreover, Figure 3.2 indicates that the results obtained exploiting the proposed scheme for videos compressed with ultrafast preset are always better than the ones given by the traditional codec, thus it can achieve better quality than H.264/AVC for bitrate values greater than 5Mb. Therefore, the 10Mb-Ultrafast configuration is additionally analyzed. The following plots show a comparison between the performances of both techniques only for videos compressed with the Ultrafast preset.

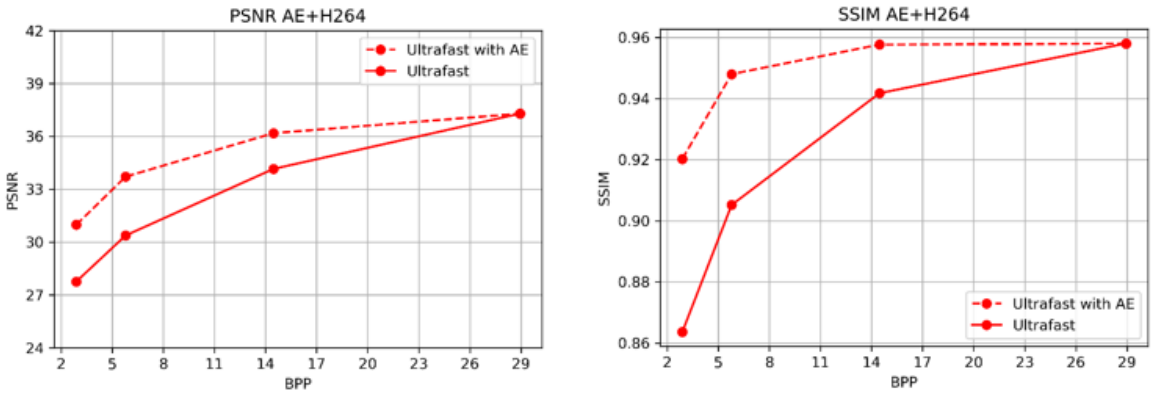


Figure 3.3 The plots show a comparison between the codec standard H.264 and the proposed scheme in terms of quality. Both PSNR (right) and SSIM (left) are expressed in function of the bitrate.

The plots in Figure 3.3 report the trend of the PSNR and SSIM only for the ultrafast preset for bitrate values which range from 1 to 10 Mb. It is found that the 10Mb bitrate value represents the point in which H.264/AVC achieves the same performance in terms of quality of the deep learning-based scheme.

A further analysis to state the reliability of the results is conducted by employing the Mann-Whitney U test [69]. It is demonstrated that there is a difference between the traditional and the proposed method for almost the entire bitrate/preset set, with an exception for the 5Mb-Medium/5Mb-Slow/10Mb-Ultrafast pair, for which the p value is respectively 0.86, 0.90 and 0.79, thus the null hypothesis is not rejected.
This result offers a further proof that H.264/AVC is capable of reconstructing frames with the same quality featured by the deep learning-based solution for those configurations.

3.2 Time

Time presents a superior limit, since low latency is requested to guarantee real time applications. More in detail, the threshold is set to *33,3ms* (30 Hz) per frame, for both encoding and decoding time.

As for quality, also for encoding and decoding time it is first proposed a highlighting of the values variability for both H.264/AVC and the DLL-based scheme; thus, each point of the plot is associated with its own standard deviation (Figure 3.4). The purple line represents the 30ms threshold.

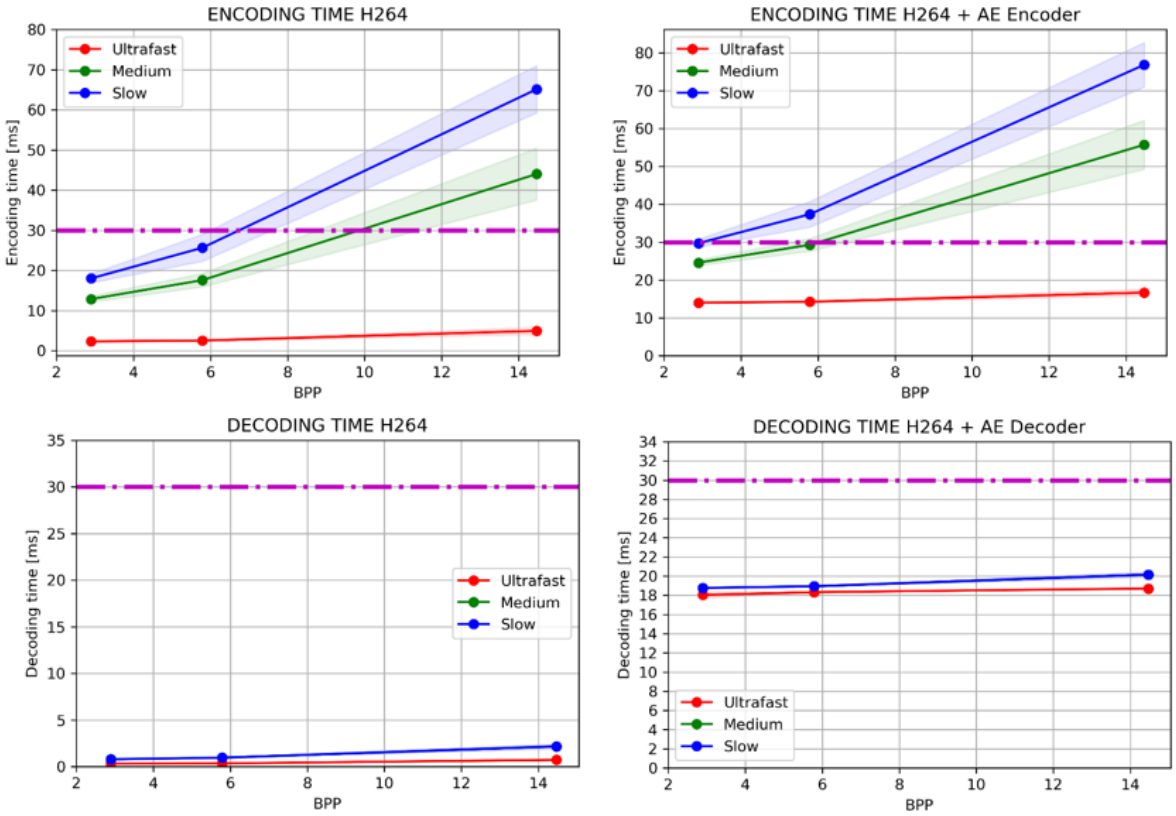


Figure 3.4 The plots show the trend of both encoding and decoding time for H.264/AVC and the scheme proposed (H.264 + AE) for each bitrate/preset pair. Each plot reports also the standard deviation. The purple line represents the threshold for real time application.

Traditionally, the time requested for the encoding process is significantly higher than the one addressed to the decoding one, as it is shown in the plots which presents the encoding/decoding time for H.264/AVC only (Figure 3.4). However, the method implemented shows opposite results, since the values associated to the decoder are higher than the ones of the encoder (Figure 3.4). The reason lies on the fact that the first one is more computationally demanding than the last one.

Successively, it is shown a comparison between the results achieved by the traditional method and the one implemented (Figure 3.5). As in Figure 3.4, also in Figure 3.5 the purple line represents the 30ms threshold.

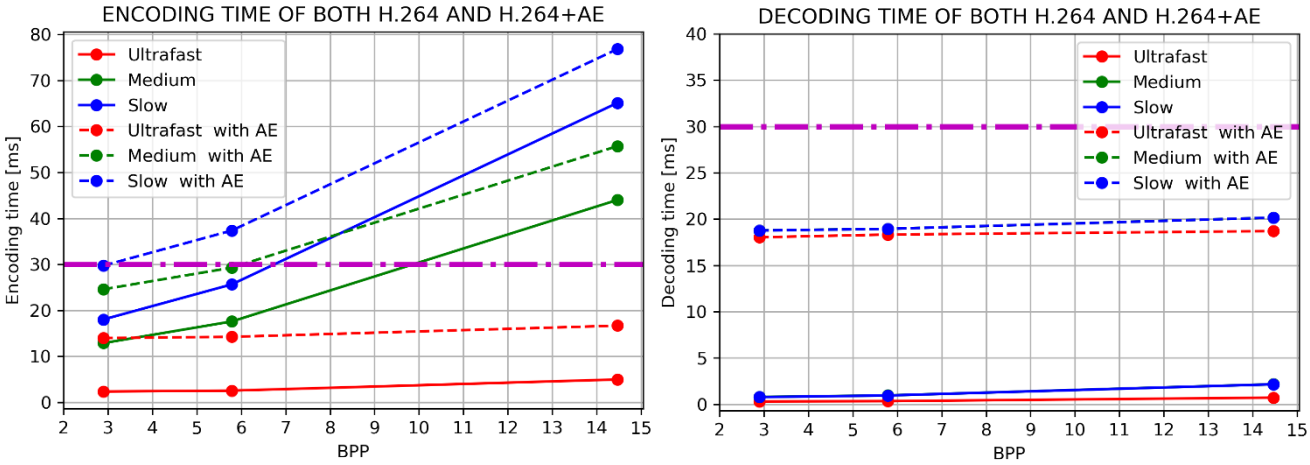


Figure 3.5 The plots report a comparison between the performances in terms of encoding/decoding time of H.264 and of the scheme implemented, for each configuration. The purple line represents the threshold for real time applications.

It can be noticed that the decoding time never overpasses the real time threshold, differently from the encoding time.

Since the 10Mb-Ultrafast configuration is additionally investigated, it is reported also the encoding/decoding time trend for the Ultrafast preset for bitrate values which range from 1Mb to 10Mb.

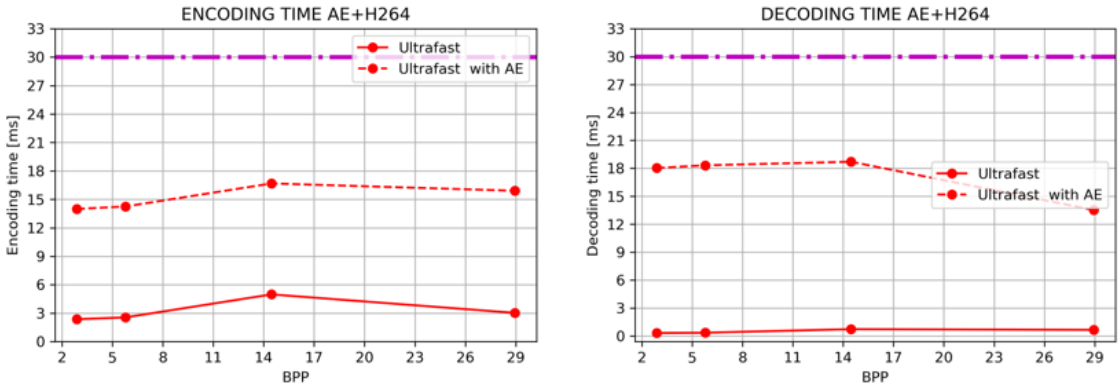


Figure 3.6 The plots report a comparison between the performances in terms of encoding/decoding time of H.264 and of the scheme implemented for each configuration for the specific Ultrafast preset.

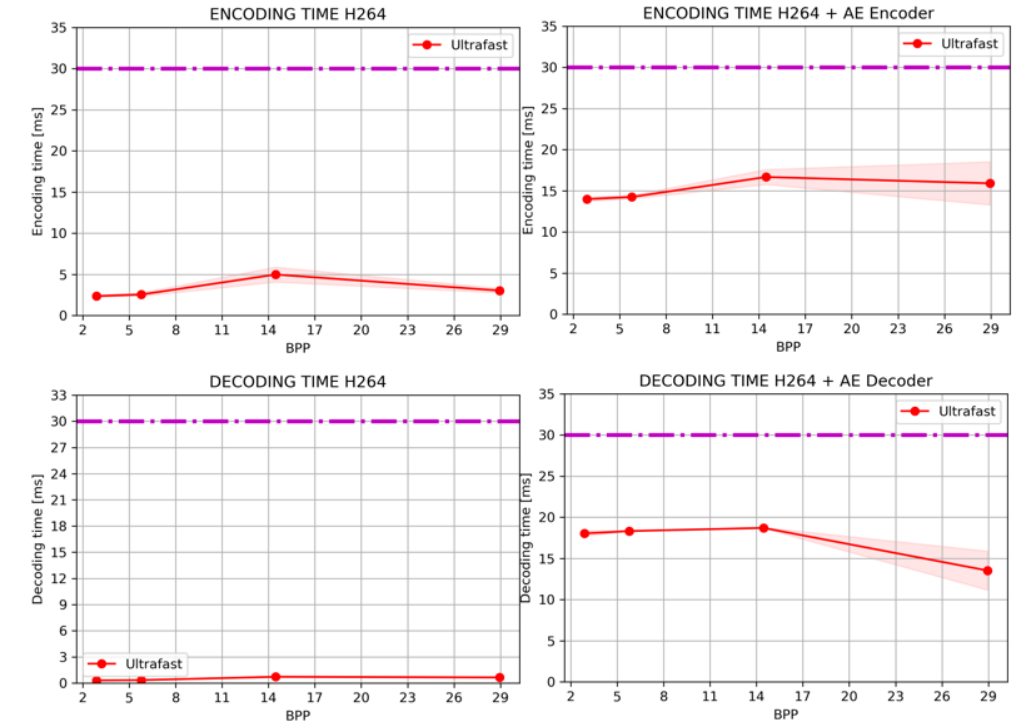


Figure 3.7 The plots show the trend of both encoding and decoding time for H.264/AVC and the scheme proposed (H.264 + AE) for each bitrate/preset pair for the specific Ultrafast preset. Each plot reports also the standard deviation. The purple line represents the threshold for real time applications.

The plots indicate that it real time applications are still possible for higher bitrate exploiting the ultrafast preset for compression, since both encoding and decoding time are highly below the 30 ms threshold.

4 Discussion

In this chapter it is reported a discussion to highlight strengths and limits of the method proposed. Furthermore, it is presented the conclusions, which wants to address further improvement to the algorithm proposed and brand-new challenges for future research.

To assess the quality of the reconstructed frames PSNR and SSIM have been employed. The first one is the most used metric to evaluate the reconstruction quality of lossy compression codecs [74]; typical PSNR values for 8-bit data range from 30 dB to 50 dB, where the higher the better. SSIM predict the perceived quality of digital images and videos [75], thus can be useful to further assess the performance achieved by both the traditional and the deep learning-based methods. SSIM values range from 0 to 1, where 1 indicates the perfect structural similarity. As Table 3.1 shows, the mean PSNR value of each bitrate/preset pair is comprised between 30 dB and 38 dB for both H.264/AVC and the method proposed, except the one associated to the 1Mb-Ultrafast configuration for H.264/AVC. It is worth noticed that the standard deviation computed for each point (Figure 3.1) indicates a variability which increases with the bitrate and also passing from ultrafast to medium to slow preset. However, as concerns the deep learning scheme, the minimum PSNR value of each bitrate/preset pair remains higher than 30 dB, except for the 1Mb-Ultrafast/2Mb-Ultrafast pair, whose value lies around 28 dB. As regards H.264/AVC, the minimum PSNR value it is not lower than 30 dB only for the 5Mb-Medium/5Mb-Slow pair. The comparison between the minimum PSNR values of the implemented scheme and the mean PSNR values of H.264/AVC shows that the deep learning-based solution performs always better than the traditional codec only for a bitrate equal to 1 Mb. Summing up the information obtained by analysing the PSNR values, it can be stated that the quality of the reconstruction performed by the scheme proposed is on average good (PSNR values > 30 dB) and better than the one achieved with H.264/AVC, except for the 5Mb-Medium/5Mb-Slow pair, as it is demonstrated in the first plot of Figure 3.2 . The Table 3.2, together with the plots in Figure 3.1, demonstrates that the percieved quality is on average meaningly better for the frames reconstructed by the DL-based scheme for a bitrate equal to 1 Mb and for the 2Mb-Ultrafast configuration, while is almost unnoticeable for higher bitrate, i.e., 2 Mb and 5 Mb. Moreover, the SSIM values related to 5Mb-Medium/5Mb-Slow pair indicates a slighetly higher percieved quality for images compressed by the traditional codec.

Considering the error associated to each mean value computed, i.e. the standard deviation, it can be noticed that the percived quality is always superior for frames obtained by employing the scheme proposed only for a bitrate equal to 1 Mb. In fact, the minimum SSIM value associated to 2 Mb and 5 Mb for the DL method is lower than the mean SSIM value computed for H.264/AVC. The percieved quality is high for frames reconstructed both by H.264/AVC and the scheme proposed; therefore, it can be visually noticed a difference in the images compressed by using the ultrafast preset.

Since the Ultrafast preset shows the possibility to employ higher bitrate values, a further evaluation for the configuration 10Mb-Ultrafast is conducted. As Figure 3.3 indicates, the quality reached by employing the traditional method is equal to the one achieved by the proposed scheme. Thus, it can be stated that the deep learning solution overcomes the performance of H.264/AVC for bitrate values that range from 1Mb to 10Mb.

The following images allow a visual comparison between the quality of reconstructed frames achieved by H.264/AVC and the method proposed, for a better comprehension of the results.

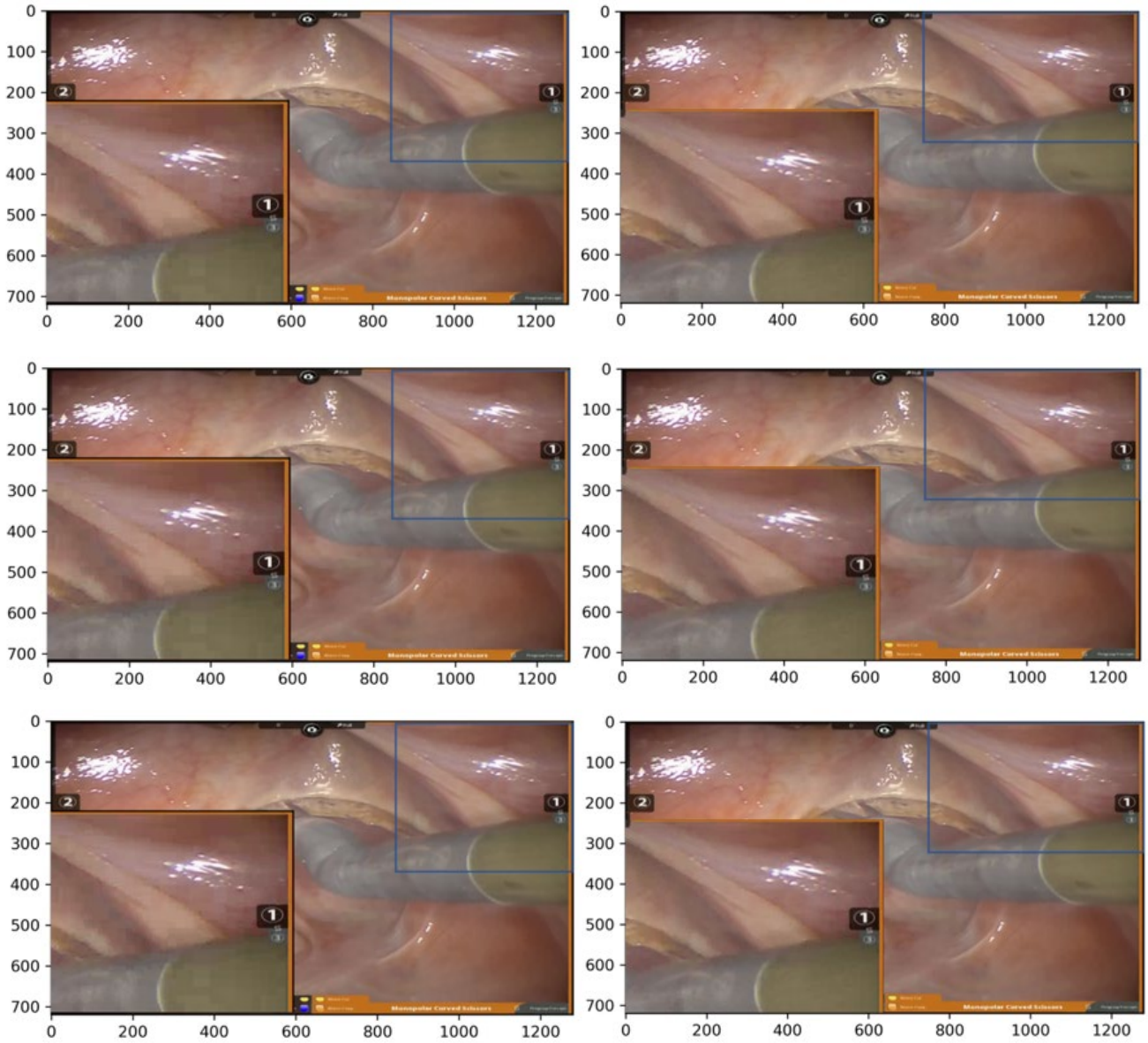


Figure 4.1 The figures show the difference between the reconstructed frame and the original one for both H.264/AVC (left) and the proposed method (right). The compression is performed by using the ultrafast preset associated with (from top to bottom) 1Mb, 2Mb, 5Mb bitrate value.

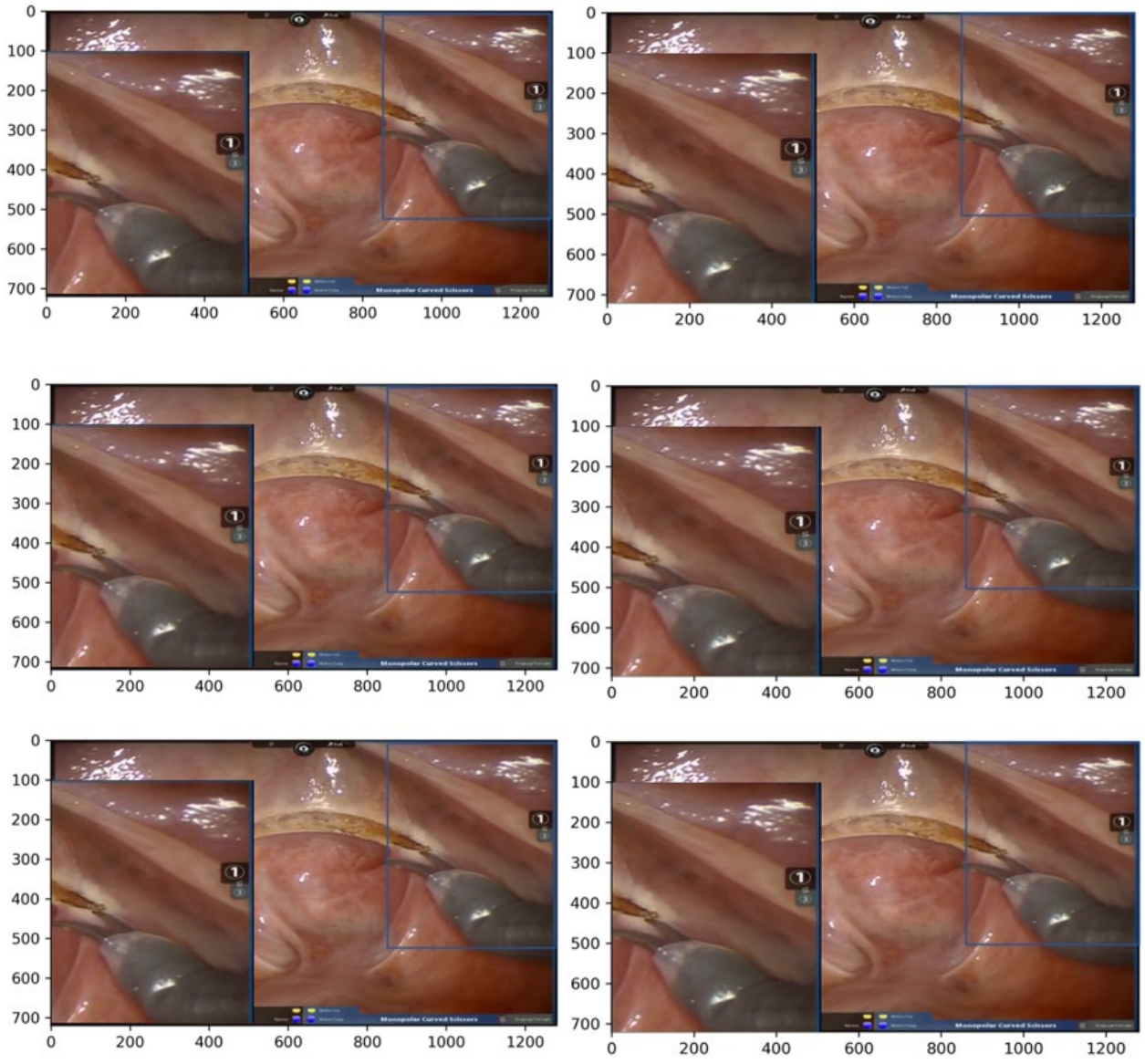


Figure 4.2 The figures show the difference between the reconstructed frame and the original one for both H.264/AVC (left) and the proposed method (right). The compression is performed by using the medium preset associated with (from top to bottom) 1Mb, 2Mb, 5Mb.

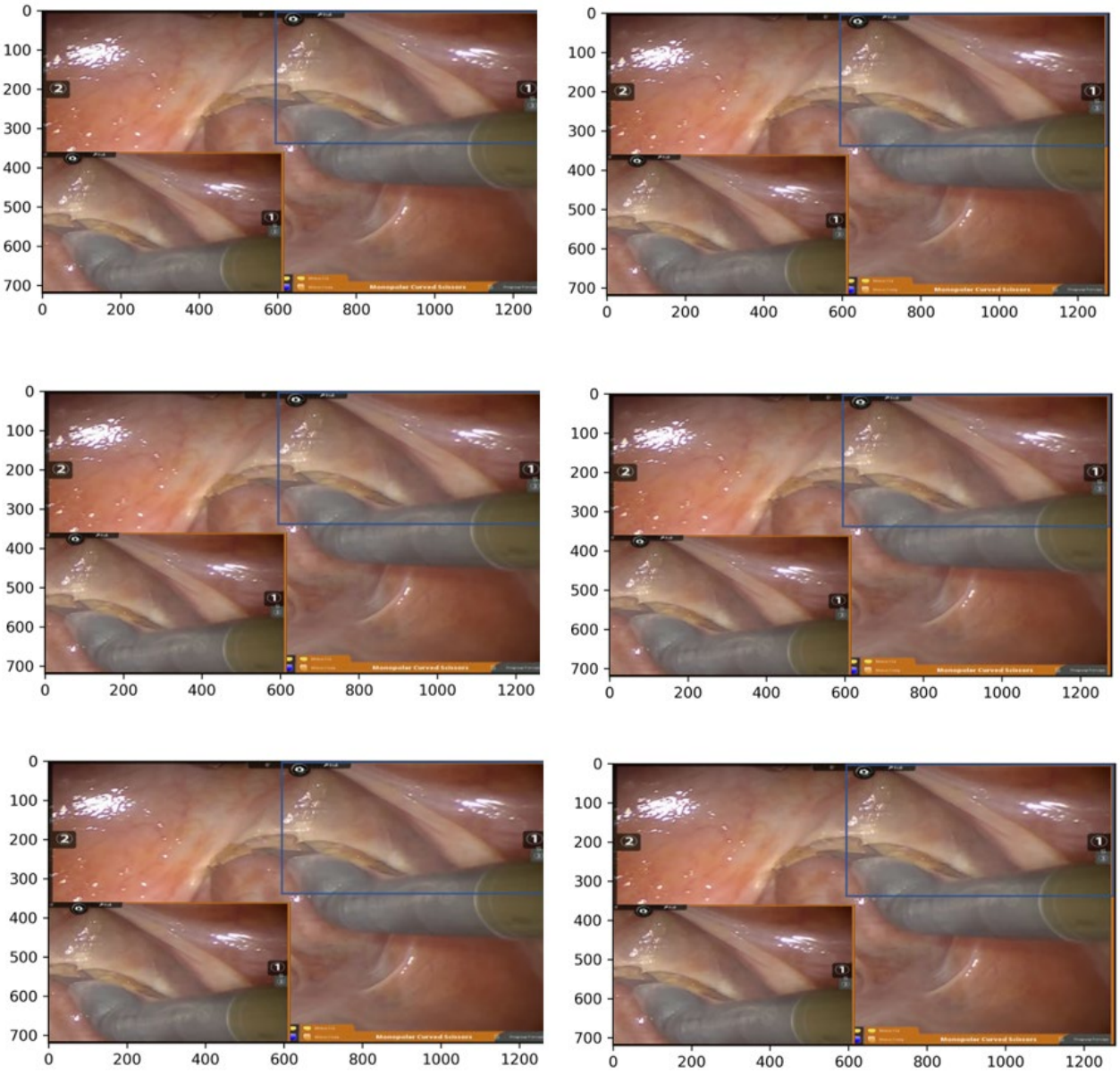


Figure 4.3 The figures show the difference between the reconstructed frame and the original one for both H.264/AVC (left) and the proposed method (right). The compression is performed by using the slow preset associated with (from top to bottom) 1Mb, 2Mb, 5Mb.

Analyzing the mean values obtained both for PSNR and SSIM for each clip, which present a common trend, it is observed that significantly lower values are obtained for clip 5 and clip 7 for the entire bitrate/preset configurations set, thus a frame-by-frame evaluation is conducted.

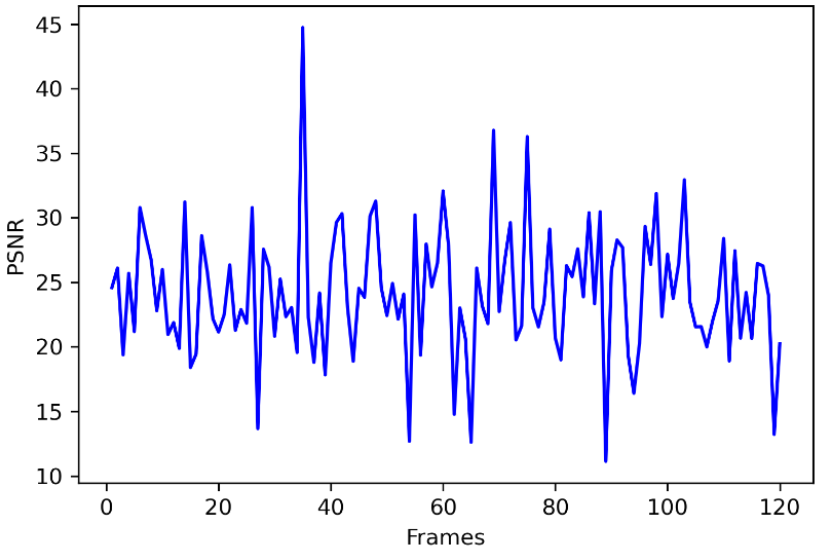


Figure 4.4 The plot shows the PSNR for each frame of the clip 5 for the 5Mb-Slow pair

Figure 4.4 indicates considerably low PSNR values among the entire video, characterized by very fast movements (Figure 4.5).

Figure 4.5 A frame extracted by clip 5 that well shows the fast movement

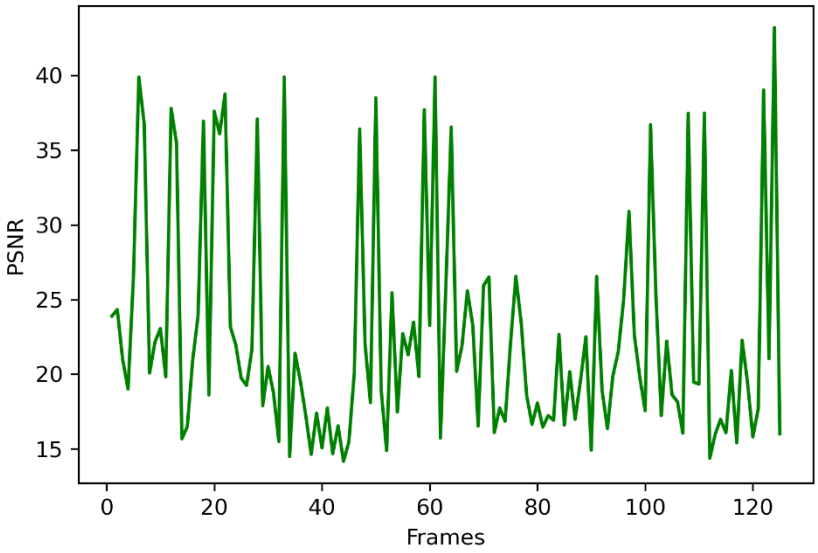


Figure 4.6 The plot shows the PSNR for each frame of the clip 7 for the 5Mb-Slow pair

The plot in Figure 4.6 presents minima around frame 40, then for frames ranging from 70 to 90 and again around frame 110. Therefore, the algorithm does not perform a good compression over the entire video, which features considerably fast movements (Figure 4.7).

Figure 4.7 A frame extracted by clip 7 that well shows the fast movement

As the remaining clips do not contain high-speed actions and feature good compression performance, it can be stated that the deep learning-based scheme performs lower quality compression where significantly fast movements are present.

Since the perceptive quality is shown to be really good for all the configurations, as the SSIM values are close to 1, the best bitrate/preset pair choice is made by considering the encoding/decoding time. Indeed, the latency of the video feedback highly limits telesurgery applications. It needs to be highlighted that the delay between the movement performed by the surgeon through the master console and its visualization on the video screen is composed by the sum of latency due to the video codec and the one associated to the transmission signal which allows the motion (Figure 4.8).

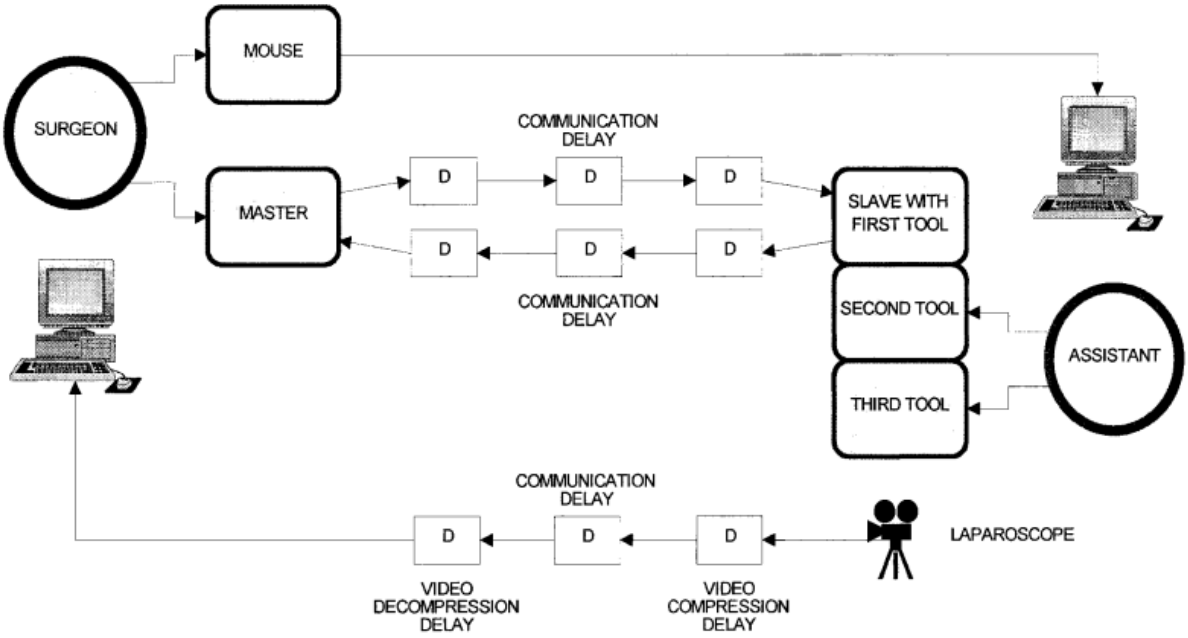


Figure 4.8 The telesurgery system [85]

In literature various experiments are proposed to assess the critical value of latency in this field. It needs to be considered that the impact of latency (transmission + compression) on surgeon's performances is strictly related to the task to be completed[76]: the higher is the difficulty of the operation, the greater the delay negatively affects the success of the surgery. Moreover, the achievement of the purpose is influenced by the experience of the surgeon. Thus, more capable doctors are less sensitive to the delay.

As can be observed, the effects of latency are subjective, since they depend upon both surgeon's experience and difficulty of the task. Anyway, a significant delay in the transmission of the videos results dangerous even for high-skilled medics, since unwanted events, e.g, bleeding and presence of mass, can occur. The studies on latency effects are based on the perception of the surgeons composing the experimental group, thus the evaluation is always subjective, thus it can vary among different assessments. In [76] is demonstrated that three distinct studies have conducted to different results: the first one has found 200ms to be critical for difficult tasks, while the other two noticed that 300ms can be accepted for the robot impose less physical and mental demands on the operator [76] In [77] a total amount of 150ms, comprising the compression and decompression process performed by MPEG-2, is found to be critical for completing the surgical procedure employed for the experiment. During a well conducted transcontinental Robot Assisted Remote Telesurgery a time delay of 150ms [77] is calculated. Also, in [78] 150ms is considered a "comfort zone" for such applications. Moreover, it is found 330ms to be the maximum value recommended for telesurgery, where the latency associated to the video codec is 70ms (encoding + transmission + decoding) [80]. Ideally, delay time is considered ideal for values less than 100ms [17].

Considering the previous data, it can be stated that the delay associated to the video transmission is in general lower than the one due to the transmission of the motion signal and it may not be greater than 70ms.

In view of the results obtained by the scheme proposed, only seven of the ten configurations are suitable:

- 1Mb-Ultrafast
- 2Mb-Ultrafast
- 5Mb-Ultrafast
- 10 Mb-Ultrafast
- 1Mb-Medium
- 2Mb-Medium
- 1Mb-Slow

From the six configuration *1Mb-Slow*, *2Mb-Medium* and *10Mb-Ultrafast* are selected since they represent the best quality-time trade-off for that bitrate value. The most suitable bitrate/preset pair results *1Mb-Slow* since it achieves the highest perceived quality, while not overcoming the time threshold. The last configuration shows an even lower encoding/decoding time than the 5Mb-Ultrafast pair. This result lies on the fact that the computation of the encoding/decoding time associated to the traditional codec is performed by FFmpeg, which can be affected by other processes that were running on the computer by the time the measures were taken. Since the ultrafast preset shows the possibility to perform real time coding, it should be further explored for higher bitrate, even if the quality achieved results lower than the one obtained by the traditional codec.

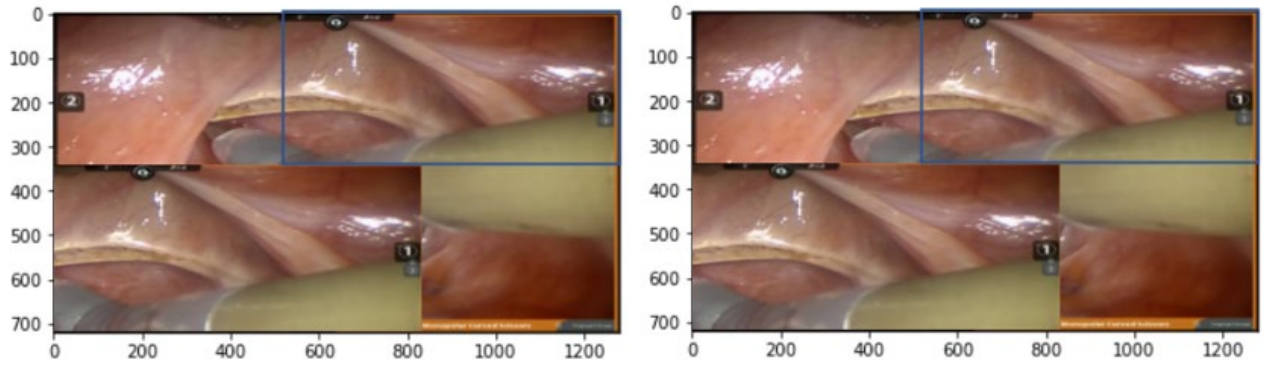


Figure 4.9 The figures show the difference between the reconstructed frame and the original one for both H.264/AVC (left) and the proposed method (right). The compression is performed by using the 10Mb-Ultrafast configuration.

The SSIM and PSNR values associated to this configuration are equal to the one obtained by employing the 5Mb-Slow pair, thus perception and the reconstruction are of good quality (Figure 4.9).

The configurations chosen are able to transmit 30 frame per second, even if in some cases the sum between encoding and decoding time overcomes the threshold. Indeed, the real time application remains possible assuming that both encoder and decoder are working at maximum 33,3ms each - without considering the transmission time - since encoder and decoder run on different devices.

It is worth noticing that the deep learning solution proposed is not developed specifically for high-speed compression, while H.264/AVC is highly optimized for the purpose. Moreover, a low-performing GPU is used, thus the computation could be accelerated by employing a better one.

Besides, the compression of the residual is performed frame-by-frame, hence it can be studied better solutions for saving time. It needs to be also considered that the videos employed for the analysis are downloaded from YouTube, therefore they have been previously compressed during the uploading on the site. It is clear that the implemented method may be widely optimized to achieve better performances both in terms of quality and speed.

5 Conclusions

It has been analyzed the problem of the transmission in the surgery domain, which requires peculiar constrains in terms of quality and latency. In fact, the stability of the system is guaranteed under low latency and limited bandwidth conditions, while high quality needs to be preserved to avoid the loss of relevant clinical information during the whole process, which includes compression, transmission and reconstruction of the frames. Although much research has been conducted around increasing quality by leveraging the properties of deep learning structures, such strategies in the surgical domain, which joints high quality and low-latency requirements, are still scarcely widespread. Even though AVC and HEVC codecs are largely optimized for speed and quality, deep learning techniques have a much larger margin of improved not completely explored yet. This work presents a computational-friendly solution which is capable to jointly enhance the compression quality and work under low-latency constrains in a low bitrate scenario. In other words, this scheme offers the possibility to obtain good compression quality of high-resolution videos in a low-bandwidth domain, which is useful in all those contexts that feature a non-fast internet connection, e.g., developing Countries and rural areas. The quality guaranteed is high, thus it allows for the detection of every detail in the surgical area in different situations, e.g., bleeding and smoking. Even though the reconstruction of really fast movement is more difficult, the quality perceived do not compromise the result of the surgery. The solution proposed allows for remote surgery in which the distance between the surgeon and the patient could be of more than 14 000 km, since latency can remain considerably under 70ms. The method implemented can be widely modified to become a powerful tool for telemedicine, telementoring and remote surgery applications. In fact, further optimizations could make the network more performant, especially in terms of speed. There are, indeed, several methods to accelerate DNN that can be easily exploited, ranging from hardware-aware optimizations of the network implementation [81], up to lightweight learning-based solutions for image compression [6] and software-based solutions that prune the DNN to make them more efficient [82], or even the adoption of GPU accelerators for DNN, e.g., Tensor cores [83], and ad-hoc hardware DNN implementations which can be compared to hardware-accelerated AVC encoders and decoders [84]. In the end, it is worth noticing that many surgical procedures enable the 3D perception, exploiting stereo-images.

Although it results in a more complex transmission, as in a higher quantity of data to handle, it can be leveraged the redundancy between left and right images for the implementation of brand-new solutions to guarantee increasingly performances not only in the medical domain, but also among a large variety of fields, such as virtual reality and videogames.

The progress in compression systems may lead to the spreading of tele-health, which can have a strong impact on the quality of life, allowing to perform surgical procedures in different context directly by highly experienced surgeons or less-skilled doctors guided by experts. Moreover, an evolution in the teaching method can improve the learning process of medicine students.

Even though the feasibility of telemedicine applications in terms of tools is demonstrated, their usage may be still difficult since these advanced technologies are expansive, thus such solutions could not be afforded by everyone. However, the cost-benefit should make the telematic system a worth investment to guarantee superior healthcare services to a higher number of people.

Bibliography

[1] S. Lombardo, J. U. N. HAN, C. Schroers, and S. Mandt, "Deep Generative Video Compression," in *Advances in Neural Information Processing Systems*, 2019, vol. 32. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/f1ea154c843f7cf3677db7ce922a2d17-Paper.pdf

[2] S. Erridge, D. K. T. Yeung, H. R. H. Patel, and S. Purkayastha, "Telementoring of Surgeons: A Systematic Review," *Surgical Innovation*, vol. 26, no. 1, pp. 95–111, 2019, doi: 10.1177/1553350618813250.

[3] A. J. Hung, J. Chen, A. Shah, and I. S. Gill, "Telementoring and Telesurgery for Minimally Invasive Procedures," *Journal of Urology*, vol. 199, no. 2, pp. 355–369, Feb. 2018, doi: 10.1016/j.juro.2017.06.082.

[4] P. Kavitha, "A Survey on Lossless and Lossy Data Compression Methods," 2016.

[5] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and Video Compression With Neural Networks: A Review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1683–1698, 2020, doi: 10.1109/TCSVT.2019.2910119.

[6] L. H. Chen, C. G. Bampis, Z. Li, A. Norkin, and A. C. Bovik, "ProxIQA: A Proxy Approach to Perceptual Optimization of Learned Image Compression," *IEEE Transactions on Image Processing*, vol. 30, pp. 360–373, 2021, doi: 10.1109/TIP.2020.3036752.

[7] A. Chaabouni, Y. Gaudeau, J. Lambert, J. M. Moureaux, and P. Gallet, "H.264 medical video compression for telemedicine: A performance analysis," *IRBM*, vol. 37, no. 1, pp. 40–48, Feb. 2016, doi: 10.1016/J.IRBM.2015.09.007.

[8] A. E. Kumcu *et al.*, "Visual quality assessment of H.264/AVC compressed laparoscopic video," *Medical Imaging 2014: Image Perception, Observer Performance, and Technology Assessment*, vol. 9037, p. 90370A, Mar. 2014, doi: 10.1117/12.2044336.

[9] L. Lévêque, H. Liu, Y. Cheng, C. Cavaro-Ménard, and P. le Callet, "Video quality perception in telesurgery," *2017 IEEE 19th International Workshop on Multimedia Signal Processing, MMSP 2017*, vol. 2017-January, pp. 1–5, Nov. 2017, doi: 10.1109/MMSP.2017.8122219.

[10] J. Shah, A. Vyas, and D. Vyas, "The History of Robotics in Surgical Specialties," *American Journal of Robotic Surgery*, vol. 1, no. 1, pp. 12–20, Apr. 2015, doi: 10.1166/AJRS.2014.1006.

[11] B. Jaffray, "Minimally invasive surgery," *Archives of Disease in Childhood*, vol. 90, no. 5, pp. 537–542, May 2005, doi: 10.1136/ADC.2004.062760.

[12] K. H. Fuchs, "Minimally invasive surgery," *Endoscopy*, vol. 34, no. 2, pp. 154–159, 2002, doi: 10.1055/S-2002-19857.

[13] J. H. Palep, "Robotic assisted minimally invasive surgery," *Journal of Minimal Access Surgery*, vol. 5, no. 1, p. 1, Jan. 2009, doi: 10.4103/0972-9941.51313.

[14] P. Dasgupta and R. S. Kirby, "The current status of robot-assisted radical prostatectomy," *Asian Journal of Andrology*, vol. 11, no. 1, pp. 90–93, 2009, doi: 10.1038/AJA.2008.11.

[15] "Robotic Prostatectomy | Johns Hopkins Medicine." https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/robotic-prostatectomy (accessed Mar. 08, 2022).

[16] A. Kumcu *et al.*, "Effect of video lag on laparoscopic surgery: correlation between performance and usability at low latencies," *Int J Med Robot*, vol. 13, no. 2, Jun. 2017, doi: 10.1002/RCS.1758.

[17] "Telesurgery prospects in delivering healthcare in remote areas - PubMed." https://pubmed.ncbi.nlm.nih.gov/30697023/ (accessed Mar. 21, 2022).

[18] C. Korte, S. Sudhakaran Nair, V. Nistor, T. P. Low, C. R. Doarn, and G. Schaffner, "Determining the threshold of time-delay for teleoperation accuracy and efficiency in relation to telesurgery," *Telemedicine and e-Health*, vol. 20, no. 12, pp. 1078–1086, Dec. 2014, doi: 10.1089/TMJ.2013.0367.

[19] S. Xu, M. Perez, K. Yang, C. Perrenot, J. Felblinger, and J. Hubert, "Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dV-Trainer(®) simulator," *Surg Endosc*, vol. 28, no. 9, pp. 2569–2576, 2014, doi: 10.1007/S00464-014-3504-Z.

[20] A. Punchihewa and D. Bailey, "A Review of Emerging Video Codecs: Challenges and Opportunities," *International Conference Image and Vision Computing New Zealand*, vol. 2020-November, Nov. 2020, doi: 10.1109/IVCNZ51579.2020.9290536.

[21] "(PDF) A Brief History of Video Coding." https://www.researchgate.net/publication/228745838_A_Brief_History_of_Video_Coding (accessed Feb. 22, 2022).

[22] B. Munzer, K. Schoeffmann, L. Boszormenyi, J. F. Smulders, and J. J. Jakimowicz, "Investigation of the impact of compression on the perceptional quality of laparoscopic videos," *Proceedings - IEEE Symposium on Computer-Based Medical Systems*, pp. 153–158, 2014, doi: 10.1109/CBMS.2014.58.

[23] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003, doi: 10.1109/TCSVT.2003.815165.

[24] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–142, Aug. 2006, doi: 10.1109/MCOM.2006.1678121.

[25] "Real words or Buzzwords?: H.264 and I-frames, P-frames and B-frames – Part 2 | Security Info Watch." https://www.securityinfowatch.com/video-surveillance/article/21124160/real-words-or-buzzwords-h264-and-iframes-pframes-and-bframes-part-2 (accessed Feb. 24, 2022).

[26] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012, doi: 10.1109/TCSVT.2012.2221191.

[27] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An End-to-end Deep Video Compression Framework", Accessed: Feb. 25, 2022. [Online]. Available: https://github.com/GuoLusjtu/DVC.

[28] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video Enhancement with Task-Oriented Flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019, doi: 10.1007/s11263-018-01144-2.

[29] "What are Convolutional Neural Networks? | IBM." https://www.ibm.com/cloud/learn/convolutional-neural-networks (accessed Mar. 08, 2022).

[30] K. Fukushima, "Biological Cybernetics Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biol. Cybernetics*, vol. 36, p. 202, 1980.

[31] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989, doi: 10.1162/NECO.1989.1.4.541.

[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[33] "Convolutional Neural Network Definition | DeepAI." https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network (accessed Mar. 08, 2022).

[34] "Konsep Convolutional Neural Network (CNN) - IGLab." https://iglab.tech/konsep-convolutional-neural-network-cnn/ (accessed Mar. 08, 2022).

[35] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, Feb. 2018, Accessed: Feb. 25, 2022. [Online]. Available: https://arxiv.org/abs/1802.01436v2

[36] Y. Li *et al.*, "Convolutional Neural Network-Based Block Up-Sampling for Intra Frame Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2316–2330, Sep. 2018, doi: 10.1109/TCSVT.2017.2727682.

[37] J. Lin, D. Liu, H. Yang, H. Li, and F. Wu, "Convolutional Neural Network-Based Block Up-Sampling for HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 12, pp. 3701–3715, Dec. 2019, doi: 10.1109/TCSVT.2018.2884203.

[38] Y. Dai, D. Liu, and F. Wu, "A Convolutional Neural Network Approach for Post-Processing in HEVC Intra Coding," *Lecture Notes in Computer Science*

(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10132 LNCS, pp. 28–39, Aug. 2016, doi: 10.1007/978-3-319-51811-4_3.

[39] Y. Wang, X. Fan, R. Xiong, D. Zhao, and W. Gao, "Neural Network-Based Enhancement to Inter Prediction for Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 826–838, Feb. 2022, doi: 10.1109/TCSVT.2021.3063165.

[40] S. Huo, D. Liu, F. Wu, and H. Li, "Convolutional Neural Network-Based Motion Compensation Refinement for Video Coding," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2018-May, Apr. 2018, doi: 10.1109/ISCAS.2018.8351609.

[41] H. Zhang, L. Song, L. Li, Z. Li, and X. Yang, "Compression Priors Assisted Convolutional Neural Network for Fractional Interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1953–1967, May 2021, doi: 10.1109/TCSVT.2020.3011197.

[42] N. Yan, D. Liu, H. Li, B. Li, L. Li, and F. Wu, "Convolutional Neural Network-Based Fractional-Pixel Motion Compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 840–853, Mar. 2019, doi: 10.1109/TCSVT.2018.2816932.

[43] H. Zhang, L. Li, L. Song, X. Yang, and Z. Li, "Advanced CNN Based Motion Compensation Fractional Interpolation," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2019-September, pp. 709–713, Sep. 2019, doi: 10.1109/ICIP.2019.8804199.

[44] Z. Zhao, S. Wang, S. Wang, X. Zhang, S. Ma, and J. Yang, "Enhanced Bi-Prediction with Convolutional Neural Network for High-Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3291–3301, Nov. 2019, doi: 10.1109/TCSVT.2018.2876399.

[45] J. Mao, H. Yu, X. Gao, and L. Yu, "CNN-based bi-prediction utilizing spatial information for video coding," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2019-May, 2019, doi: 10.1109/ISCAS.2019.8702552.

[46] Y. Zhang, T. Shen, X. Ji, Y. Zhang, R. Xiong, and Q. Dai, "Residual Highway Convolutional Neural Networks for in-loop Filtering in HEVC," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3827–3841, Aug. 2018, doi: 10.1109/TIP.2018.2815841.

[47] L. Feng, X. Zhang, X. Zhang, S. Wang, R. Wang, and S. Ma, "A Dual-Network Based Super-Resolution for Compressed High Definition Video," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11164 LNCS, pp. 600–610, Sep. 2018, doi: 10.1007/978-3-030-00776-8_55.

[48] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution".

[49] J. Kim, J. K. Lee, and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks".

[50] N. Sachdeva, G. Manco, E. Ritacco, and V. Pudi, "Sequential Variational Autoencoders for Collaborative Filtering," Nov. 2018, doi: 10.48550/arxiv.1811.09975.

[51] A. Habibian, J. M. Tomczak, and T. S. Cohen, "Video Compression With Rate-Distortion Autoencoders".

[52] Y.-H. Tsai, M.-Y. Liu, D. Sun, M.-H. Yang, and J. Kautz, "Learning Binary Residual Representations for Domain-specific Video Streaming", Accessed: Mar. 04, 2022. [Online]. Available: www.aaai.org

[53] B. M. ¨ Unzer, K. Schoeffmann, · Laszlo Böszörmenyi, B. Münzer, and L. Böszörmenyi, "Content-based processing and analysis of endoscopic images and videos: A survey," *Multimedia Tools and Applications 2017 77:1*, vol. 77, no. 1, pp. 1323–1362, Jan. 2017, doi: 10.1007/S11042-016-4219-Z.

[54] S. Khire, S. Robertson, N. Jayant, E. A. Wood, M. E. Stachura, and T. Goksel, "Region-of-interest video coding for enabling surgical telementoring in low-bandwidth scenarios," *Proceedings - IEEE Military Communications Conference MILCOM*, 2012, doi: 10.1109/MILCOM.2012.6415792.

[55] B. Munzer, K. Schoeffmann, and L. Boszormenyi, "Domain-specific video compression for long-term archiving of endoscopic surgery videos," *Proceedings - IEEE Symposium on Computer-Based Medical Systems*, vol. 2016-August, pp. 312–317, Aug. 2016, doi: 10.1109/CBMS.2016.28.

[56] N. Ghamsarian, H. Amirpourazarian, C. Timmerer, M. Taschwer, and K. Schöffmann, "Relevance-Based Compression of Cataract Surgery Videos Using Convolutional Neural Networks," *MM 2020 - Proceedings of the 28th*

ACM International Conference on Multimedia, pp. 3577–3582, Oct. 2020, doi: 10.1145/3394171.3413658.

[57] A. Hassan, M. Ghafoor, S. A. Tariq, T. Zia, and W. Ahmad, "High Efficiency Video Coding (HEVC)-Based Surgical Telementoring System Using Shallow Convolutional Neural Network.," *Journal of Digital Imaging*, vol. 32, no. 6, pp. 1027–1043, Dec. 2019, doi: 10.1007/S10278-019-00206-2.

[58] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, "Binary neural networks: A survey," *Pattern Recognition*, vol. 105, p. 107281, Sep. 2020, doi: 10.1016/J.PATCOG.2020.107281.

[59] "Binary Neural Networks." https://www.intel.com/content/www/us/en/developer/articles/technical/binary-neural-networks.html (accessed Mar. 08, 2022).

[60] "PixelShuffle — PyTorch 1.10 documentation." https://pytorch.org/docs/stable/generated/torch.nn.PixelShuffle.html (accessed Mar. 09, 2022).

[61] W. Shi *et al.*, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network".

[62] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift".

[63] "PyTorch." https://pytorch.org/ (accessed Mar. 09, 2022).

[64] "11.11. Learning Rate Scheduling — Dive into Deep Learning 0.17.4 documentation." https://www.d2l.ai/chapter_optimization/lr-scheduler.html (accessed Mar. 09, 2022).

[65] M. D. Schluchter, "Mean Square Error," *Encyclopedia of Biostatistics*, Jul. 2005, doi: 10.1002/0470011815.B2A15087.

[66] D. P. Kingma and J. Lei Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION".

[67] "Converting Video Formats with FFmpeg | Linux Journal." https://www.linuxjournal.com/article/8517 (accessed Mar. 08, 2022).

[68] "FFmpeg preset comparison x264 2019; Encode speed and file size." https://write.corbpie.com/ffmpeg-preset-comparison-x264-2019-encode-speed-and-file-size/ (accessed Mar. 09, 2022).

[69] Q. Huynh-Thu, M. Ghanbari, Q. Huynh-Thu, and M. Ghanbari, "The accuracy of PSNR in predicting video quality for different video scenes and frame rates," *Telecommunication Systems 2010 49:1*, vol. 49, no. 1, pp. 35–48, Jun. 2010, doi: 10.1007/S11235-010-9351-X.

[70] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," *Proceedings - International Conference on Pattern Recognition*, pp. 2366–2369, 2010, doi: 10.1109/ICPR.2010.579.

[71] U. Sara, M. Akter, M. S. Uddin, U. Sara, M. Akter, and M. S. Uddin, "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study," *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, Mar. 2019, doi: 10.4236/JCC.2019.73002.

[72] R. Dosselmann and X. D. Yang, "A comprehensive assessment of the structural similarity index," *Signal, Image and Video Processing 2009 5:1*, vol. 5, no. 1, pp. 81–91, Nov. 2009, doi: 10.1007/S11760-009-0144-1.

[73] P. E. McKnight and J. Najab, "Mann-Whitney U Test," *The Corsini Encyclopedia of Psychology*, pp. 1–1, Jan. 2010, doi: 10.1002/9780470479216.CORPSY0524.

[74] "Peak signal-to-noise ratio - Wikipedia." https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio?msclkid=ca971673a60811ec91ec7efbf435358e (accessed Mar. 17, 2022).

[75] "Structural similarity - Wikipedia." https://en.wikipedia.org/wiki/Structural_similarity?msclkid=6c451622a60911ec943a88f3947075b5 (accessed Mar. 17, 2022).

[76] A. Kumcu *et al.*, "Effect of video lag on laparoscopic surgery: correlation between performance and usability at low latencies," *Int J Med Robot*, vol. 13, no. 2, Jun. 2017, doi: 10.1002/RCS.1758.

[77] M. Perez *et al.*, "Paradigms and experimental set-up for the determination of the acceptable delay in telesurgery," *Annu Int Conf IEEE Eng Med Biol Soc*, vol. 2007, pp. 453–456, 2007, doi: 10.1109/IEMBS.2007.4352321.

[78] "(38) Transcontinental robot-assisted remote telesurgery: feasibility and potential applications | Michele Simone - Academia.edu." https://www.academia.edu/13660584/Transcontinental_robot_assisted_remo

te_telesurgery_feasibility_and_potential_applications?msclkid=2b75f010a93211ec997045537845d281 (accessed Mar. 21, 2022).

[79] M. Perez *et al.*, "Paradigms and experimental set-up for the determination of the acceptable delay in telesurgery," *Annu Int Conf IEEE Eng Med Biol Soc*, vol. 2007, pp. 453–456, 2007, doi: 10.1109/IEMBS.2007.4352321.

[80] S. E. Butner and M. Ghodoussi, "Transforming a Surgical Robot for Human Telesurgery," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 818–824, Oct. 2003, doi: 10.1109/TRA.2003.817214.

[81] P. M. Nvidia, J. Hall, M. Research, H. Yin, J. Kautz, and N. Fusi, "HANT: Hardware-Aware Network Transformation," Jul. 2021, Accessed: Mar. 11, 2022. [Online]. Available: https://arxiv.org/abs/2107.10624v1

[82] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 11256–11264, Jun. 2019, doi: 10.1109/CVPR.2019.01152.

[83] S. Markidis, S. W. der Chien, E. Laure, I. B. Peng, and J. S. Vetter, "NVIDIA tensor core programmability, performance & precision," *Proceedings - 2018 IEEE 32nd International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2018*, pp. 522–531, Aug. 2018, doi: 10.1109/IPDPSW.2018.00091.

[84] S. Han *et al.*, "EIE: Efficient Inference Engine on Compressed Deep Neural Network," *Proceedings - 2016 43rd International Symposium on Computer Architecture, ISCA 2016*, pp. 243–254, Aug. 2016, doi: 10.1109/ISCA.2016.30.

[85] J. M. Thompson, M. P. Ottensmeyer, and T. B. Sheridan, "Human factors in telesurgery: effects of time delay and asynchrony in video and control feedback with local manipulative assistance," *Telemed J*, vol. 5, no. 2, pp. 129–137, 1999, doi: 10.1089/107830299312096.

List of Figures

List of Tables