Machine learning based optimization for VVC low bitrate coding

Fatemeh Nasiri

DOCTORAT BRETAGNE LOIRE / MATHSTIC

INSA RENNES

THÈSE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Traitement du signal

Par

« Fatemeh Nasiri»

«Machine learning based optimization for VVC low bitrate coding»

Thèse présentée et soutenue à « Rennes », le « 19/05/2022 »
Unité de recherche : IETR, Equipe VAADER

Rapporteurs avant soutenance :

Marco Cagnazzo	Professeur, Telecom Paris, France / University of Padua, Italy
Christian Timmerer	Associate Professor, Alpen-Adria-Universität Klagenfurt, Austria

Composition du Jury :

Président :	Christine Guillemot	Directrice de Recherche, INRIA, France
Examinateur :	Giuseppe Valenzise	Chargé de recherche, Université Paris-Saclay, France
Dir. de thèse :	Luce Morin	Professeur, INSA Rennes, France
Co-dir. de thèse :	Wassim Hamidouche	Maître de Conférences, INSA Rennes, France

Invité(s) :

Encadrant industriel :	Nicolas Dhollande	Ingénieur de Recherche et Innovation, Aviwest, France
Encadrant industriel :	Jean-Yves Aubie	Directeur de lab, IRT-B-com, France

Résumé en français

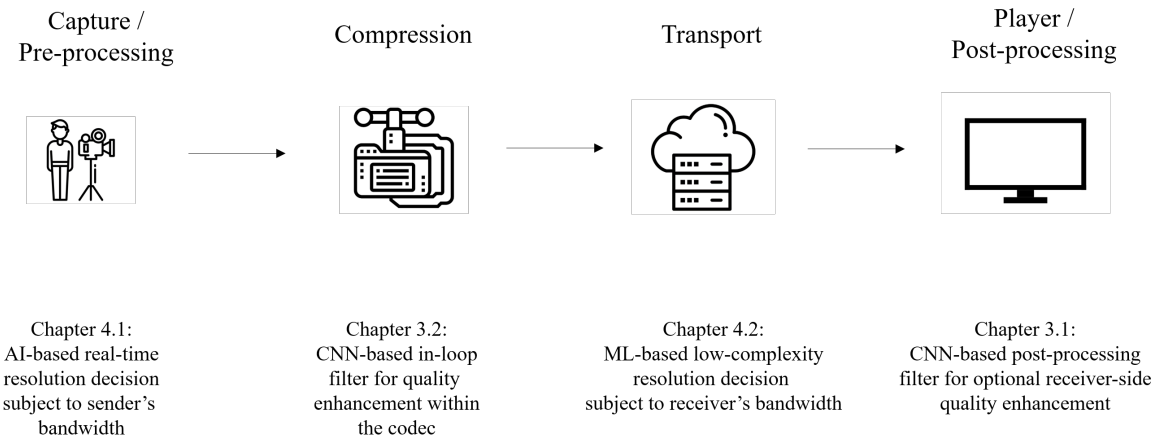
Contexe

Le développement de la vidéo numérique a eu une conséquence majeure : les utilisateurs dont la bande passante était considérée comme trop faible pour les communications vidéo envoient et reçoivent désormais des vidéos compressées. Ces applications à bas débit ou à très bas débit méritent une attention particulière, qu'il s'agisse d'applications de divertissement (ex : cloud gaming ou appel vidéo) ou critiques (ex : chirurgie à distance, surveillance ou partage d'écran). Pour identifier ce qui peut améliorer la qualité d'expérience dans de telles applications, il faut d'abord bien comprendre l'écosystème de diffusion vidéo, de la capture et du prétraitement, au transcodage, à la transmission, à la distribution et à la diffusion, au décodage, au post-traitement, et enfin la lecture, comme représenté ci-dessous.

Au cours de cette thèse, plusieurs pistes de recherche ont été menées, touchant différents éléments de l'écosystème de diffusion de contenu. L'objectif commun à toutes ces pistes était d'aider le système à fonctionner plus efficacement pour des applications de communication vidéo où la bande passante disponible est très contrainte. Précisément, cette thèse traite des trois aspects suivants d'un système de diffusion vidéo. Ainsi, elle tente de répondre aux questions suivantes :

1. Pré-traitement : quels processus peuvent être appliqués avant l'encodeur, sur le contenu non-compressé, pour optimiser les performances de l'encodage ? Lesquelles pourraient profiter aux applications vidéo à faible débit ? Ce processus doit-il être agnostique au processus d'encodage et des capacités du récepteur ?
2. Encodage : quels processus peuvent être appliqués dans un encodeur pour améliorer l'efficacité de la compression des applications vidéo à faible débit ? Doivent-ils également imposer des modifications côté décodeur (c'est-à-dire des changements normatifs) ? Ou doivent-ils uniquement être représentés comme une optimisation des décisions encodeur (changements non-normatifs) ?
3. Post-traitement : côté récepteur, quelles sont les options pour améliorer la qualité de l'expérience des applications vidéo à faible débit ? Cet effort devrait-il impliquer le processus de décodage ? Ou doit-il éventuellement être appliqué sur les pixels reconstruits (c'est-à-dire après décodage), en fonction de la capacité de l'affichage ?

En cherchant à répondre à toutes ces questions, nous nous sommes posé une question supplémentaire : comment l'IA peut-elle être exploitée à ces fins ? Les différents algorithmes d'IA



Un écosystème de diffusion vidéo simplifié et apports de cette thèse

ont montré leur potentiel dans la résolution de problèmes complexes, notamment en traitement du signal 2D. Par conséquent, nous avons activement continué à nous poser une question supplémentaire : comment l'intelligence artificielle (IA) peut-elle être exploitée ? Cela est dû aux potentiels éprouvés de différents algorithmes d'IA dans la résolution de problèmes complexes dans le contexte du traitement du signal bidimensionnel. En conséquence, nous avons intégré différents algorithmes d'IA tels que les réseaux de neurones convolutifs, les arbres de décision, la régression et l'apprentissage d'ensemble dans nos recherches.

L'amélioration de la qualité et le changement dynamique de résolution en fonction du contenu sont les deux thèmes principaux de cette thèse. Chacun de ces thèmes pourrait potentiellement impliquer l'un ou l'autre des aspects ci-dessus, à savoir le pré-traitement, l'encodage et le post-traitement.

Cette thèse a commencé fin 2018 alors que la normalisation VVC en était à ses dernières étapes et que les différentes industries avaient commencé à le considérer comme le codec de nouvelle génération. Par conséquent, nous nous sommes concentrés sur d'éventuels problèmes de faible débit où VVC pourrait potentiellement se démarquer, ouvrir de nouvelles perspectives ou améliorer les applications existantes..

Résumé par chapitre

Chapitre 1 : Une brêve introduction est fournie sur la compression vidéo hybride basée blocs. Pour cela, les principaux éléments d'un système de compression moderne sont abordés. De plus, les métriques et méthodologies utilisées pour évaluer la performance des codecs vidéos sont expliqués. À cette fin, les principaux éléments sont discutés à un niveau élevé. De plus, des métriques et des méthodologies pour évaluer les performances des codecs vidéo sont discutées.

Chapitre 2 : En se concentrant sur les deux thèmes de l'amélioration de la qualité et du changement dynamique de résolution, un état de l'art est fourni. Dans cette étude, l'accent a été mis sur la définition du problème ainsi que sur l'utilisation de l'IA pour résoudre les problèmes dans les travaux existants.

Chapitre 3 : Le premier thème de nos travaux : l'amélioration de la qualité basée sur l'IA, est abordé. Dans ce chapitre, plusieurs algorithmes basésIA sont présentés ,pouvant agir à différents niveaux. Plus précisément, nous verrons comment les outils de filtrage de VVC de Versatile Video Coding (VVC) peuvent coexister ou être entièrement remplacés par des méthodes basées sur CNN. De plus, les méthodes proposées sont également évaluées lorsqu'elles sont utilisées en post-traitement.

Chapitre 4 : Le deuxième thème de nos travaux, : la changement dynamique de résolution, a été présenté.Tout d'abord, un framework générique est présenté, où les écosystèmes de diffusion vidéo déterminent actuellement la résolution optimale en fonction du débit disponible.. Ensuite, en formulant ce problème dans le contexte de la classification et de la régression, deux algorithmes sont présentés pour les applications de diffusion en direct et de vidéo à la demande (VoD).

Conclusion : Enfin, une conclusion est présentée dans ce chapitre pour expliquer ce qui peut être apporté dans de futurs travaux sur les thèmes de l'amélioration de la qualité et de la résolution adaptative.

Contributions

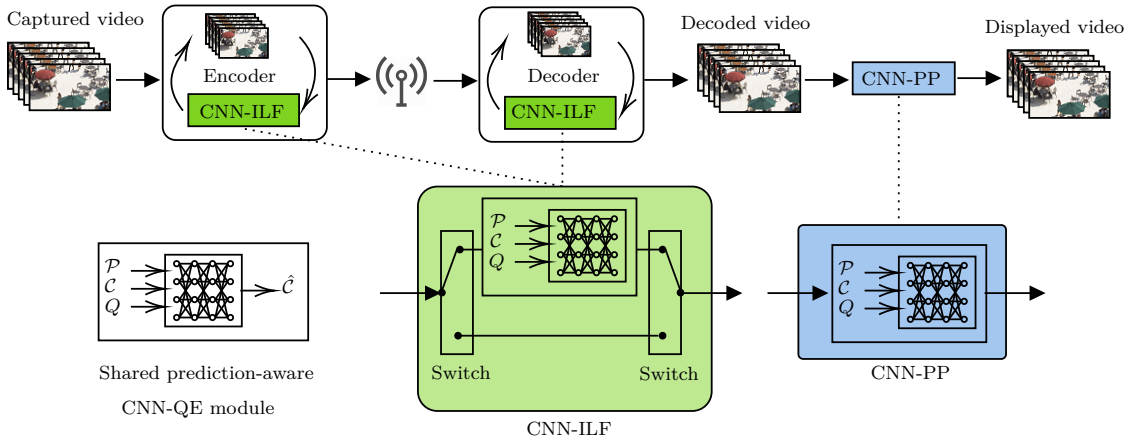


FIGURE 2 – Le cadre proposé prenant en compte la prédiction avec deux approches d'intégration de codec : Un module CNN pour l'amélioration de la qualité, deux approches prenant en compte la prédiction : filtrage in-loop (vert) et post-traitement (bleu).

Dans le chapitre 2 et le chapitre 3 de cette thèse, toutes les contributions sont élaborées en les intégrant dans leur framework dédié. Voici cependant la synthèse de ces contributions sorties

de leur contexte :

- Thème amélioration de la qualité (la Figure 2) :
  - Proposition d'une méthode de post-traitement basée sur CNN pour les images intra et inter.
  - Conception d'une méthode d'amélioration de la qualité multi-modèle avec une signalisation de haut niveau du modèle optimal au niveau blocs et trames
  - Intégration de la méthode en tant que filtre dans le logiciel de référence VVC.
  - Etude de l'impact itératif du filtre sur les différents types d'images du GoP et proposition d'une solution commutable afin de minimiser la propagation d'erreurs entre les images.
  - L'utilisation d'informations de codage telles que la prédiction et le partitionnement et la démonstration qu'il existe un potentiel important à le faire pour les filtres d'amélioration de la qualité basés sur CNN.
- Thème de changement de résolution adaptatif au contenu (la Figure 3) :
  - Étude sur la manière dont l'utilisation des informations de codage peut améliorer les performances de l'algorithme de super résolution basé sur CNN.
  - Prédiction de bitrate ladder pour les applications de streaming/diffusion vidéo en direct en minimisant le nombre d'encodages nécessaires pour construire l'enveloppe convexe optimale.
  - Prédiction de bitrate ladder pour des applications de vidéo à la demande (VOD) utilisant un encodeur multi-preset, en prédisant les seuils de débits du preset "slow" à partir de ceux du preset "fast".

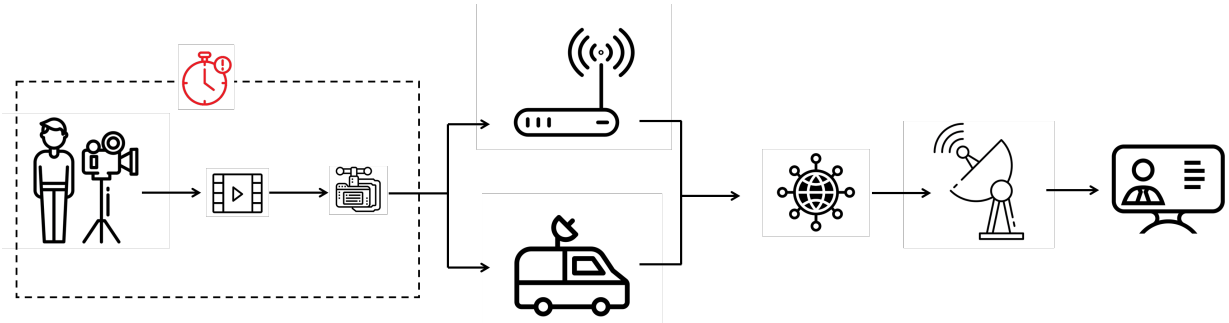


FIGURE 3 – L'écosystème typique des applications de diffusion vidéo en direct et les contraintes en termes de temps de traitement.

Conclusion

Les contributions de cette thèse sont classées et présentées en deux parties : l'amélioration de la qualité et la prédiction du bitrate ladder. Les deux problèmes sont basés sur lesur les mêmes problématiques soulevées dans le chapitre 1, tout en s'appliquant à des applications différentes. Les solutions présentées dans les deux parties ont deux points communs. Tout d'abord, ils utilisent VVC comme codec vidéo sous-jacent. Deuxièmement, les solutions proposées bénéficient des techniques basées sur apprentissage automatique pour résoudre le problème. Dans la première contribution de cette thèse cette thèse, le problème de l'amélioration de la qualité de la vidéo à très bas débit a été étudié. Les types de dégradation que nous avons ciblés étaient, en général, des artefacts de compression dus à un manque de bande passante adéquate. Ces artefacts incluent les effets de blocs, le blurring, le ringing, etc. Cependant, sans faire de distinction entre eux, nous avons cherché à les corriger, en utilisant une technique de ML, appelée CNN. a 2ème contribution de cette thèse tente de résoudre le problème d'optimisation de la qualité vidéo en préparant le signal vidéo non compressé de manière à maximiser l'efficacité des encodeurs. Précisément, dans cette contribution, nous supposons que l'encodeur vidéo est une boîte noire et nous n'avons aucun contrôle dessus, mais nous utilisons seulement des paramètres simples tels que le débit d'encodage et le preset. Cependant, cet encodeur est déployé dans un système de diffusion vidéo de bout en bout, dans lequel l'algorithme a la liberté de sélectionner au cours de temps la sous-résolution optimale qui doit être encodée d'une séquence nativement UHD doit être encodée et envoyée aux récepteurs.

Table of Contents

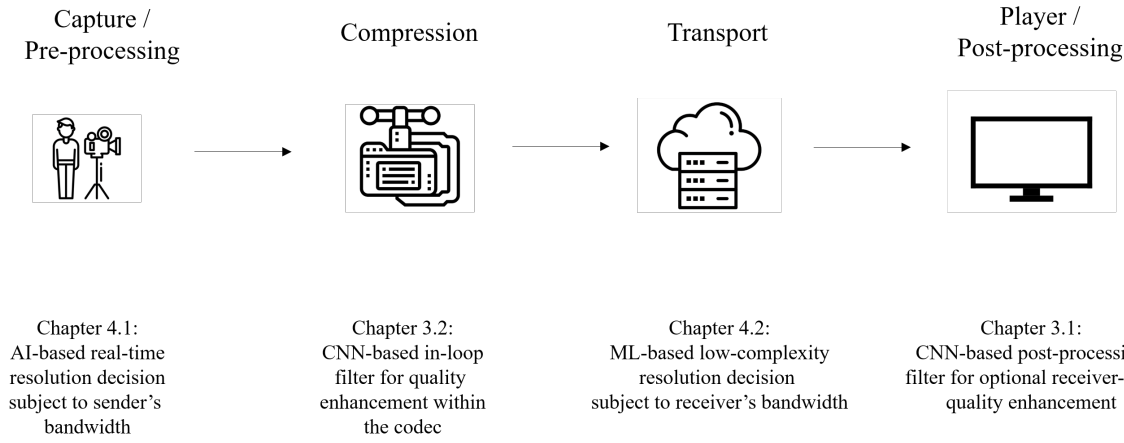
TABLE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION

Context

The prevalence of digital video has recently had one important effect : users whose bandwidth used to be considered too low for video communications, are now sending and receiving compressed videos. Given that historically, "conventionall" bitrate ranges have been in the center of attention (in terms of encoder design/optimization), one can argue that these low bitrate or very low bitrate users now deserve more dedicated attention ; Whether their applications concern entertainment (*e.g.* cloud gaming or video call) or pressing (*e.g.* remote surgery, surveillance, or screen sharing) applications. To identify what should be improved to impact the quality of experience in such applications, one might first fully understand the video delivery ecosystem, from capture and pre-processing, to compression, transport over network, decoding, post-processing and finally the playback, as visualized below.



A simplified video delivery ecosystem and contributions of this thesis

In the course of this thesis, different elements of the above ecosystem have been studied in different tracks. The common goal in all tracks was to help the system function more efficiently for low bitrate video communication applications where the available bandwidth imposes the most limiting constraint. Precisely, this thesis deals with the following three aspects of a video delivery system, and for each aspect, it attempts to answer the following questions :

1. Pre-processing : What processes are available to be applied on the uncompressed video, before compression ? Which ones might benefit low-bitrate video applications ? Should this process be aware of the encoding process and receiver's capacity ?
2. Encoding : What processes can be applied within an encoder to improve the compression efficiency of low-bitrate video applications ? Should they also impose additional processes at the decoder side (*i.e.* normative changes) ? Or should they only be represented as optimization of the decision-making process and impact only the encoder side (*i.e.* non-normative changes) ?
3. Post-Processing : What are the options as the last effort of improving the quality of experience of low bitrate video applications at the receiver side ? Should this effort involve the decoding process ? Or should it be optionally applied on the reconstructed pixels (*i.e.* after decoding), based on the capacity of the display ?

In the search for answers to all the above questions of all aspects, we actively kept asking ourselves one additional question : How can Artificial Intelligence (AI) can be leveraged ? This is due to the proven potentials of different AI algorithms in solving complex problems in the context of two-dimensional signal processing. As a result, we incorporated different AI algorithms such as CNN, decision trees, regression and ensemble learning in our research.

Two potential solutions were explored. First, we investigated how the quality of compressed videos can be enhanced to remove artifacts that might have been added due to the limitations on the bitrate. And second, we entered the domain of content-adaptive video resolution switching, in order to cope with the bitrate constraints by knowing when and how to down-sample a video before encoding. These two are the main themes of this thesis. Each of these themes could potentially involve either of the above aspects, namely pre-processing, encoding and post-processing.

Regarding the standardization context, this thesis started in late 2018 when the VVC standardization was in its last stages and the different industries had started considering it as the next generation codec. Consequently, we oriented the problem definition of low-bitrate video coding around VVC, assuming that it could potentially act as a game-changer and enable new applications or enhance current ones in this domain.

Chapters in a glance

Chapter 1 A concise introduction of a modern hybrid block-based video compression system is provided. To this end, the main elements are discussed at a high level. Moreover, metrics and methodologies of assessing the performance of video codecs are discussed.

Chapter 2 Focusing on the two themes of quality enhancement and content-adaptive resolution switching, a state-of-the-art study has been provided. In this study, the focus has been put on the problem definition as well as the use of AI for solving the problems in existing works.

Chapter 3 The first theme, AI-based quality enhancement, is discussed. In this chapter, several AI-based algorithms are presented to serve for different aspects of the workflow. Precisely, we will discuss how current loop filters of VVC can co-exist or entirely be replaced by CNN-based methods. Moreover, the proposed methods are also assessed when incorporated as optional post-processing modules.

Chapter 4 The second theme, content-adaptive resolution switching, has been presented. To this end, first, a generic framework is introduced, where video delivery ecosystems currently determine the optimal resolution based on receivers' bandwidth capacity. Then, by formulating this problem in the context of classification and regression, two algorithms are presented for live and Video-on-Demand applications.

Chapter 5 Finally a conclusion is presented in this chapter by presenting what can be done next in either quality enhancement or content-adaptive resolution switching themes.

Contributions in a glance

In Chapter 3 and Chapter 4 of this thesis, all contributions are elaborated by integrating in their dedicated framework. Here is the summary of these contributions taken out of their framework context :

- Quality enhancement theme :
  - Proposing a CNN-based post-processing method for intra and inter frames.
  - Designing a multi-model quality enhancement method with high-level signalization of the optimal model in Coding Tree Unit (CTU) and frame levels.

  - Integrating the above method as an in-loop filter within the VVC reference software.
  - Studying the impact of applying the above CNN-based in-loop filter iteratively on images of GoP and proposing a switchable method to minimize the negative effect due to multiple enhancement.
  - The use of coding information such as prediction and partitioning and demonstrating that there is significant potential in doing so for CNN-based quality enhancement filters.
- Content-adaptive resolution switching theme :
  - Study on how the use of coding information can benefit the performance of the CNN-based SR algorithm.
  - Bitrate ladder prediction for live video streaming/broadcast applications by minimizing the number of encodings needed to construct the convex hull.
  - Bitrate ladder prediction for Video on Demand (VoD) applications using a multi-preset encoder, by predicting the ladder of the slow presets from a fast preset.

Chapter 1

Introduction to hybrid block-based video coding

1.1 Introduction

The first step to investing in new video technologies is considering dedicated standards and specifications. Video codec standards are to guarantee interoperability and format compatibility between devices to enable playback of any video file conforming to the syntax of a given standard, using any device supporting it. The first advantage of such a mechanism is the facility of the interaction between different sectors of video communication, such as consumer electronics manufacturers, broadcasters, content providers, content delivery networks, etc. This virtuous circle significantly accelerates the progress of innovation and the wide adoption of new technologies. For instance, with properly defined and adopted video coding standards, operators will know that once they start distributing contents of a new format, there will be inexpensive equipment for their playback. Conversely, hardware manufacturers will also ensure that there will be operators distributing new format content to motivate viewers to buy their new products and watch them.

In the past decades, different communities have attempted to introduce new video coding standards. However, the long collaboration between the International Telecommunication Union (ITU) and the Moving Picture Experts Group (MPEG) working group of the International Standardization Organization (ISO) has been by far the most successful entity to provide video coding standards. The most widely adopted standard of this collaboration was accomplished in the late 90's, where a joint collaboration, called Joint Collaboration Team on Video Coding (JCT-VC), resulted in the most successful video coding standard, called H.264 Advanced Video Coding (AVC). Following this success, the same community of experts continued to expand existing standards or introduce new ones, notably H.265 High Efficiency Video Coding (HEVC) in 2013. The most recent collaboration between the MPEG and ITU-T, called Joint Video Experts Team (JEVT), finalized a standard in 2020. This team, consisting of numerous experts with different backgrounds (e.g. hardware, software, network etc.), aimed at investigating the video coding techniques for compression of a diversity of video formats, in a more efficient

manner. As a result, this standard adopted the name Versatile Video Coding (VVC) to reflect this primary goal.

As VVC is the standard that will most likely be used widely in the coming years, the analysis and contributions of this thesis were entirely based on it. However, without loss of generality, all proposed algorithms are applicable to most other standards, whether old ones such as AVC and HEVC, or other next-gen standards and codecs such as AV2, etc. In this chapter, a high-level overview of the video coding scheme is presented to provide a background for the rest of the chapters.

The term "block-based hybrid" video coding has been widely used for modern video standards for the past two decades. The exact meaning of this term lies under its two consisting elements, which are common among all modern standards. Precisely, from a high-level point of view, the process of compressing an image/video consists of first, splitting it into smaller units of pixels (*i.e.* the block-based aspect), then applying a combined scheme of prediction and error transformation (*i.e.* the hybrid aspect).

Figure 1.1 shows the diagram of a hybrid video codec. This generic structure only contains the main modules of such codecs and shows how their functionalities are ordered in the global view to turn the raw input video sequence into a compressed bitstream with the smallest possible memory size or bitrate. These modules, either individually, or in combination with each other, aim at decorrelation existing in different types of content. Such a scheme provides significant flexibility in terms of coding parameters choices, that must be adapted for different conditions of the signal. These content-dependent characteristics almost change constantly, either in the spatial domain (i.e. within a single frame) or temporally (i.e. through frames). As a result, an encoder has to take into account these changes and keep making the optimal decisions by using the provided coding tools and modules. In the rest of this chapter, details of some modules will be covered.

1.2 Block-based aspect: Partitioning

From the high level perspective, the solution to the video compression problem is a nested divide-and-conquer approach. The main tool to implement the "divide" step is block partitioning. The goal of this module is to identify regions of the image with correlated content and isolate them from neighboring regions that might have different content. The main motivation for such a content-based pixel isolation scheme is that different types of content require dedicated attention for decorrelation and compression. More precisely, the block partitioning module of a modern video codec identifies regions of the image that are more likely to be efficiently modeled and compressed with one of the integrated tools in that codec. The choice and tuning of such tools are then left to the rate-distortion optimization module which will be elaborated later in this

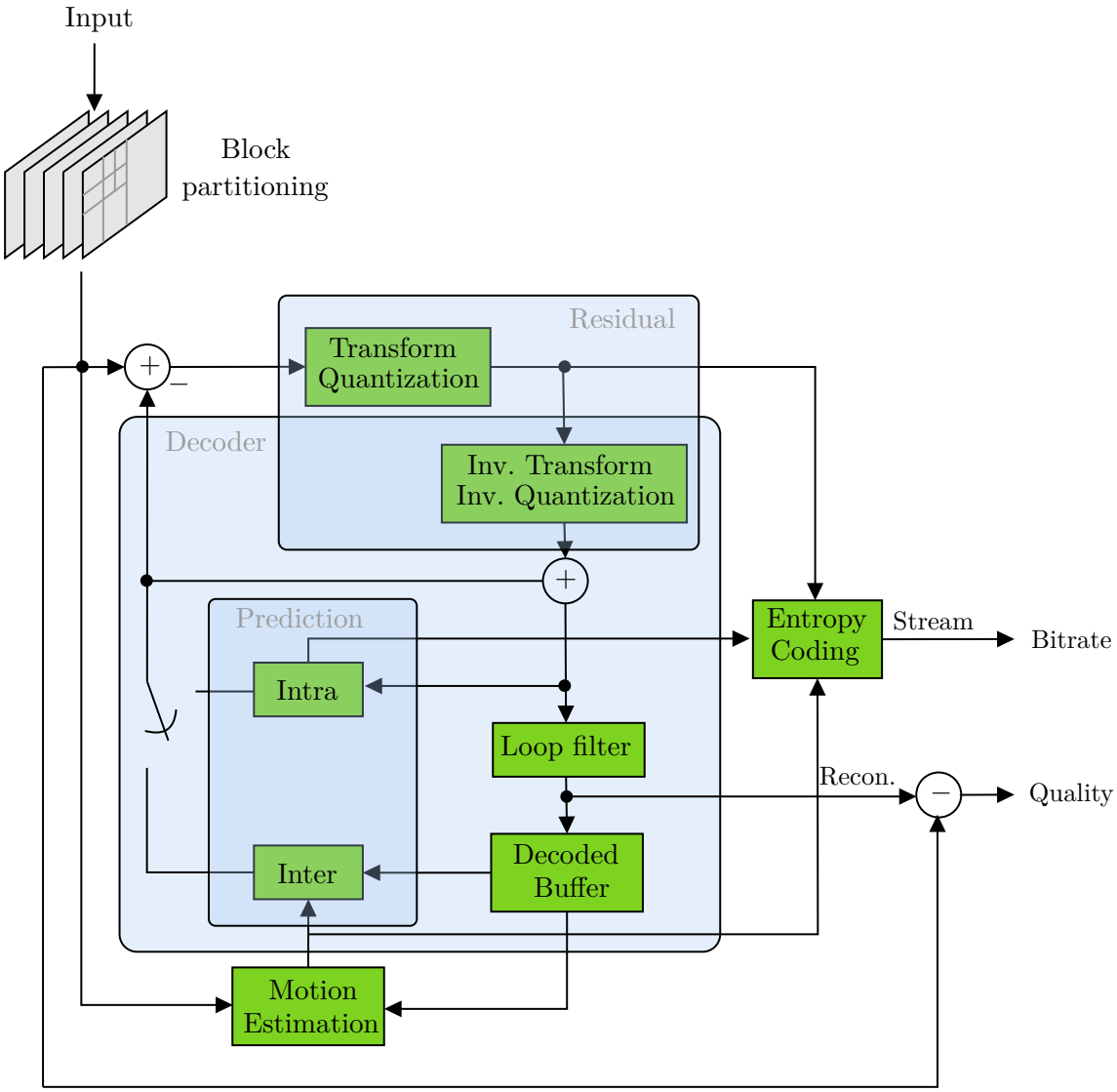


Figure 1.1 – High-level diagram of a block-based hybrid video encoder. [4].

chapter.

To elaborate on why different regions of an image must be isolated while coding, assume that we are given a video frame with a still background with a simple texture pattern. However, in the foreground, an object is moving. For such content, the encoder is better to use different partitioning strategies for the background and foreground, so that different decorrelation tools will be used on these regions. To implement such a strategy, non-overlapping rectangular blocks are offered by video compression standards, which depending on the given standard, the shapes and the sizes of these blocks are different. Figure 1.2 visualizes the main characteristics of block partitioning module in the most three recent video compression standards.

In AVC, Macro Blocks (MB) of size 16×16 are used for block partitioning. Using an MB, one can split the blocks as large as 16×16 and as small as 4×4. In between, using a concept called Sub-Macro Block (SMB), allows different rectangular blocks with an aspect ratio of 1:2. Figure 1.2 (a) shows all possible MB and SMB sizes in AVC. As the use of higher resolutions such as High Definition (HD) and Ultra High Definition (UHD) became more prevalent in years following the standardization of AVC, the next standard, HEVC was specified with a significantly more flexible partitioning scheme, called CTU and Coding Unit (CU). A CTU in HEVC is equivalent of a MB in AVC. The maximum CTU size is 64×64 and it performs a QuadTree (QT) block splitting scheme. This scheme takes the CTU as the root of a tree, where nodes are either leaves or roots for further tree branching. Regardless, each node in this scheme is a CU and is used as block partitioning unit, which can be as small as 8×8. Upon reaching the CU, a Prediction Unit (PU) scheme can further split the CU into more flexible rectangular shapes. An example of this scheme is shown in 1.2 (b). In VVC, the block partitioning is mostly based on that of HEVC, while adding further flexibility in terms of maximum and minimum CU size, compared to HEVC. First, a CTU in VVC can be as large as 128×128, allowing to more efficiently compress UHD content. Moreover, VVC applies three different splitting schemes, namely binary, ternary, and quad splits. Accordingly, the partitioning scheme in VVC is called Multi-Type Tree (MTT). Figure 1.2 (c) visualizes this scheme.

Depending on the deployed partitioning scheme, video codecs use pre-defined scan orders for processing partition blocks. Even though scan orders of all video codes are principally based on raster-order, advanced partitioning schemes, such as that of VVC, can result in complex orders when the block sizes are too different. Figure 1.3 shows an example of raster scan order in VVC. As can be seen, the high level scan order is simply based horizontal order of CTUs of the same size. However, inside each CTU, the MTT partitioning tree is recursively applied to traverse internal CUs. In this example, the top region (green overlay) indicates CTUs and CUs that have already been scanned. While the single yellow rectangle is the current CU under scanning. Finally, the third region (red overlay) indicates CTUs and CUs that have yet not been scanned.

1.3 Hybrid aspect: Prediction and residual coding

In the context of video compression standards, the term hybrid refers to the fact that two modules of prediction and residual transformation function dependently to decorrelate video signals. Particularly, there are two ways to predict pixels of a video signal: intra-picture for spatial redundancies within an image and inter-picture prediction for temporal redundancies between consecutive images. In each domain, several tools and algorithms are provided to exploit existing redundancies and compress the video signal. In either cases, even with the most accurate prediction, an error is typically introduced. In order to control the propagation of this error,

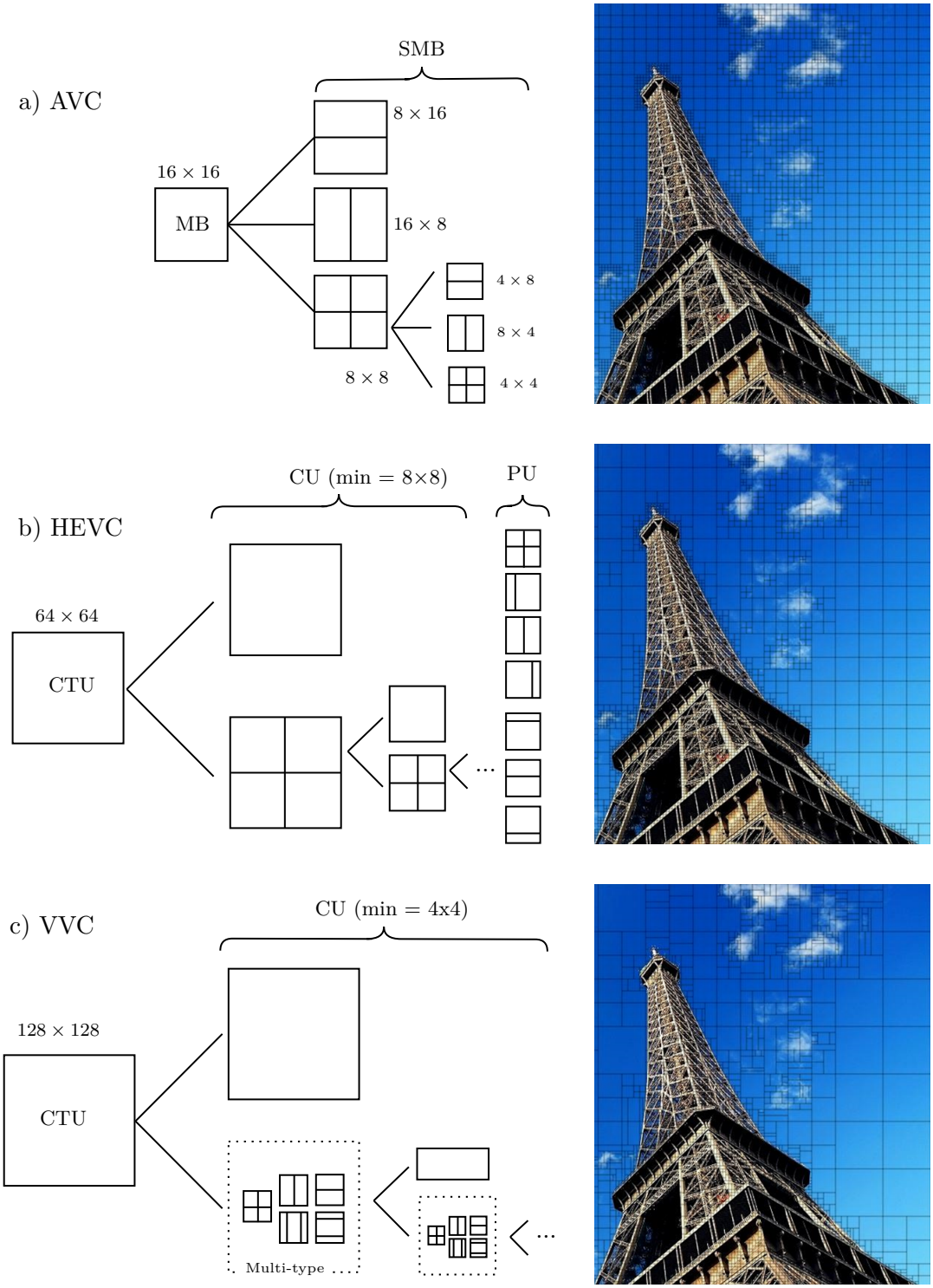


Figure 1.2 – Evolution of block partitioning used in three video coding standards of AVC, HEVC and VVC.

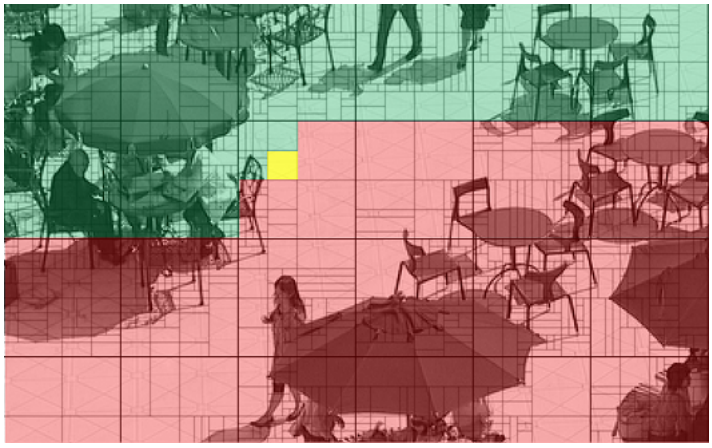


Figure 1.3 – Scan order of blocks in VVC includes a raster scan of CTUs, and then recursive raster scan of CUs within each CTU.

the encoder must transmit a signal called the residual, along with the prediction model. At the decoder side, the prediction model selected and transmitted by the encoder is reproduced and added to the decoded residual signal that is also transmitted. This process results in the final reconstructed signal. The fact that a video compression algorithm deals with both prediction and residual aspects, is commonly referred to as the hybrid aspect, which is in contrast with non-predictive compression schemes such as JPEG 2000.

1.3.1 Intra-picture prediction

When there is little or no temporal correlation between the content of the current frame and the available reference frame (*e.g.* in case of a scene change), a video encoder has no choice but to use spatial pixel prediction. To do so, intra coding benefits from similarities in texture patterns and models them by a set of geometric models. In VVC, a total of 67 IPMs are provided to model different basic texture patterns. This set consists of two modes of DC and Planar for modeling homogeneous patterns. While the remaining modes are responsible for covering the angular textures along with the range of 180 degrees. The angular modes are designed such that they provide a finer precision for angles that are more common in natural video contents (*e.g. horizontal, vertical, and diagonal*). Figure 1.4 visualizes how these IPMs are represented in four quarters of 45 degrees.

1.3.2 Inter-picture prediction

Inter-picture coding or in brief, inter coding exploits the temporal similarities between pixels form different frames of a video. This aspect that is specific to video signals, in contrast to

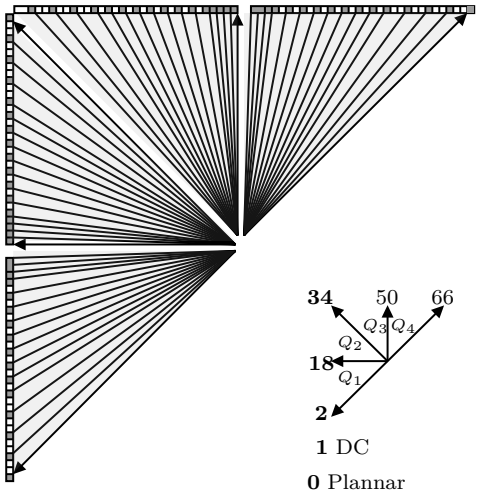


Figure 1.4 – Set of IPMs in VVC. This set consists of DC and Planar for modeling homogeneous textures, while the remaining 65 modes non-uniformly cover the angles, with an emphasis on vertical, horizontal, and diagonal angles.

still images, benefits from the fact that objects, whether they move or stay motionless, share several similar pixels that are relatively collocated in consecutive frames, hence, they can be predicted from one another. This type of correlation can largely be exploited for compression as long as the content does not drastically evolve through a scene change. The principle of inter-coding is simple. It models the relative displacement of blocks with similar pixel content, using a vector. An algorithm that performs such motion modeling is called Motion Estimation (ME) and it typically operates on two inputs: the original block to model and at least one reference frame which has a different temporal timestamp, selected whether from past or future frames, depending on the GoP structure.

The ME algorithm aims at finding a MV from a set of candidates, by minimizing an objective distortion metric between the original block and its displaced version from the reference. Figure 1.5 shows a simplified example of ME.

1.3.3 Residual transformation

As mentioned before, it is vital that a prediction step is followed by a residual coding step. To this end, residual transmission deploys a transform coding technique. The motivation is that the energy of important – with respect to Human Visual System (HV) – information in residual is typically concentrated in low-frequency regions. As a result, one can represent them with just a

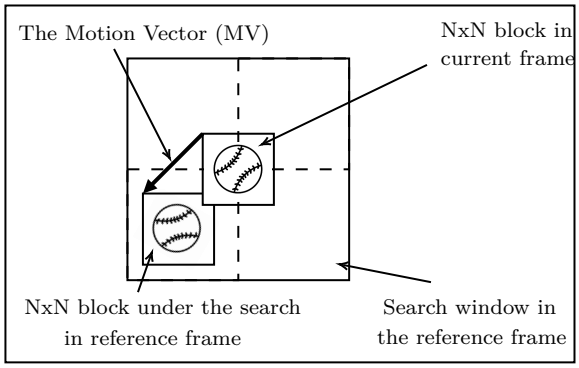


Figure 1.5 – A simplified ME algorithm deployed in inter coding

few non-zero transform coefficients. This interesting property plays a key role in the compression performance of video encoders. Another important property of transform coding, which is called energy compaction, is that it enables the lossy aspect of video compression. To do so, a lossy video encoder compresses the signal by eliminating its least informative parts. In other words, the transform domain allows us to easily identify parts of the signal that, if removed, the least amount of artifacts will be added to the decoded video. This is thanks to the modeling of human visual perception system that has less sensitivity to changes in high frequencies. As a result, lossy image and video coding schemes benefit from this feature by quantization of the transform coefficients.

Figure 1.6 simplifies different steps to demonstrate how lossy compression can benefit from the energy compaction of the transform domain. Assume that an encoder is given a simple 2-dimensional input signal as in Figure 1.6-I. This signal has a clear angular redundancy in the vw plane, as can be seen in Figure 1.6-II. However, the current xy plane is not perfectly appropriate to exploit this correlation. Therefore, a transformation step is applied to project the samples in the new vw plane, as shown in Figure 1.6-III. In this analogy, the projected samples are coefficients of the transform that was applied. Therefore, the final step applies the loss by quantizing the coefficients on the vw plane, as shown in Figure 1.6-IV. At the receiver side, the quantized coefficients are parsed one by one and arranged as in Fig 1.6-IV. Since the receiver supposedly knows the inverse transform, it projects the parsed coefficients back to the initial xy plane, as in Fig 1.6-V. Finally, the reconstructed samples are generated as in Figure 1.6-VI.

In video compression, the transform coding step operates on the residual error signal in a similar manner as described in Figure 1.6. However, there are a few differences between the two. Most importantly, the signal dimension in an actual video codec is as large as the number of pixels in the residual block, while in this simplified example of this figure, the dimension is just two. Figure 1.7 shows an actual example of transformation and quantization on 8×8 residual. In this figure, the original (lossless) signal is transformed (left column). Then the obtained

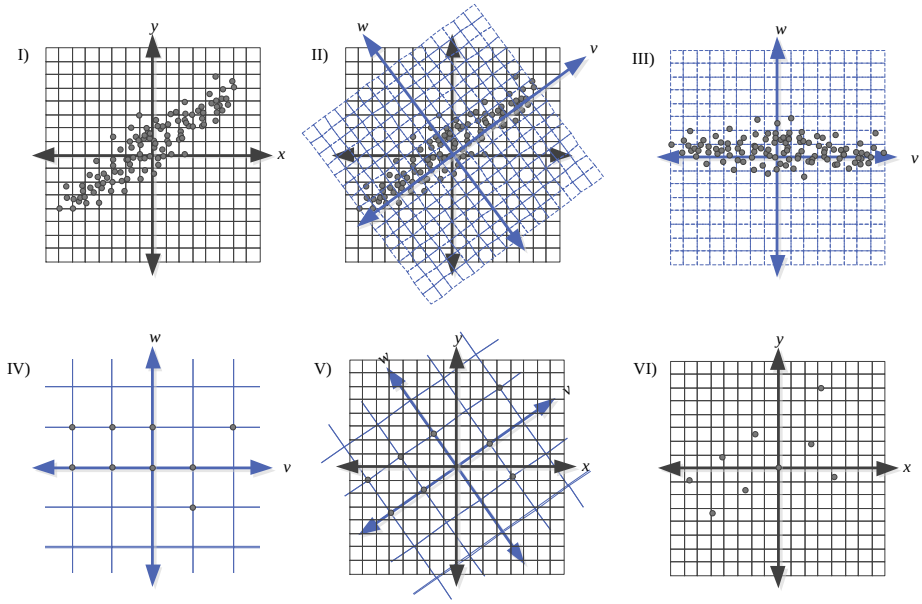


Figure 1.6 – A simplified example of end-to-end process of transformation, quantization, inverse quantization and inverse transformation [5]

coefficients are quantized with two different levels of loss. Precisely, a low QP for a small loss (hence, high rate) is represented in the middle column, and a high QP for big loss (hence, low rate) is represented in the right column. As the low QP scenario applies a fine quantizer, it results in several non-zero coefficients in the transform domain. Therefore, its impact on the amount of information loss is negligible in the reconstructed signal at the bottom-middle. On contrary, the high QP scenario applies a relative coarser quantizer. Hence, it results in a poorer reconstruction with the benefit of less non-zero coefficients in the transform domain (i.e. a lower bitrate).

1.4 Entropy coding

In addition to the predictive and block-based aspects, modern video codecs are still highly dependent on the most classic lossless compression technique, namely entropy coding. The goal of an entropy coding engine is to represent a redundant signal such that more frequent symbols use shorter code-words while less frequent ones use longer code-words. To this end, Shannon's

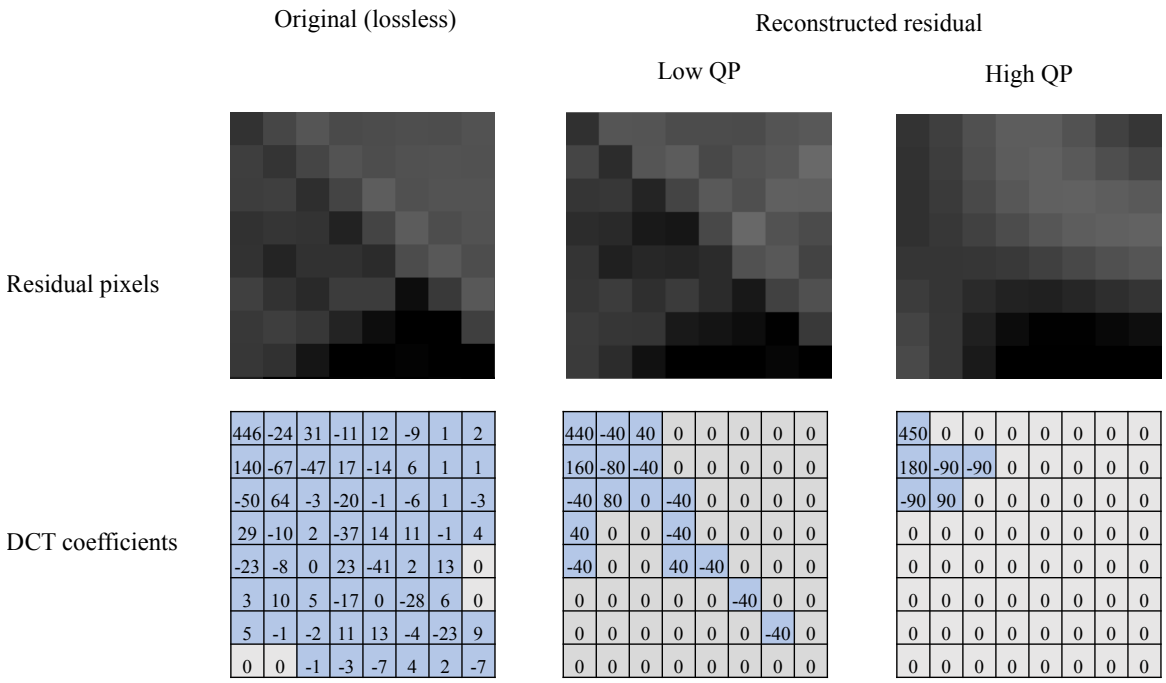


Figure 1.7 – Transform and quantization of a 8×8 block and their impact on the decoded signal.

source coding theorem says that the optimal code length of a symbol is $-log_b P$, with b being the number of symbols needed to make output codes and P the probability of the input symbol [6]. In other words, the closer probability of a symbol gets to 0, the longer its code-word will become.

Context Adaptive Binary Arithmetic Coding (CABAC) is one of the most common and efficient entropy coding engines nowadays used in video codecs. This method enables video codecs to write and read binary symbols with a rate close to their optimal rate according to Shannon's theorem. Therefore, the first step in using CABAC is to binarize the non-binary symbols.

The binarization step turns the symbol into a series of 0s and 1s, called bins. The CABAC engine associates a context to each bin (or to a group of bins). The mapping between bins and CABAC contexts is based on the symbols and their statistics. In other words, bins that are related to the same functionality of the codec and have relatively similar statistics, most likely share a CABAC context. The reason is that using an excessive number of CABAC contexts significantly increases both the implementation and execution complexity of its codec.

Once a bin is associated with a CABAC context, its statistical behavior will be followed and updated on-the-fly. This adaptiveness aspect of the CABAC engine ensures that videos with any spatial and temporal characteristics will be entropy-coded as efficiently as possible. While

without such a technique, statistics of bins would have been hard-coded with a high possibility of overfitting on certain statistical behavior.

1.5 In-loop filters

Once blocks are encoded, their reconstructed version is generated to put in the decoding picture buffer for further uses. Around this stage, normative in-loop filters are designed to enhance reconstructed pixels before putting them in the buffer. These filters are qualified as "in-loop" because they are applied inside the encoding and decoding loops, before storing the pictures in the decoded picture buffer. There are different motivations for such post-encoding pixel processing, mostly related to different types of compression artifacts that might subjectively or objectively impact the performance of the overall system. Here, we introduce the main filters in VVC, noting that the first contribution of this thesis will directly deal with them and in some cases, it competes with them to improve the same type of quality degradation.

1.5.1 Deblocking filter

As the name suggests, the DBF mainly deals with inevitable artefacts on the border of blocks. Here, the definition of a block might be vague, depending on the codec under study. But typically, most conceptual borders, including that of transform blocks, prediction blocks, CTUs, tiles, and slices are considered as borders to be treated differently during the DBF [7].

Other than the type of block border, the strength of a block edge has an important impact on the internal functionality of DBF. Intuitively, sharper block edges are filtered more strongly and vice versa. The sharpness of an edge is figured out by taking into account neighboring pixels. Figure 1.8 shows a simplified one-dimensional example of block border edge, where the relationship between pixels from the past p_i and pixels from the future q_i (where i=0, 1, 2, and 3) determine the strength of filtering in DBF.

In two-dimensional image signals, the above problem is solved similarly. Figure 1.9 demonstrates an actual image example, showing how the DBF is capable of removing the blockiness artefact.

1.5.2 Sample Adaptive Offset

The Sample Adaptive Offset (SAO) filter has been proposed initially in the HEVC standardization to reduce the ringing artefacts [1]. The key idea is to reduce distortion by classifying reconstructed samples into different categories. This classification leads to an offset for each category which is then added to all samples of the category. The offset of each category is properly calculated at the encoder and explicitly signaled to the decoder for reducing sample distortion effectively. However, the classification step is performed implicitly without needing to transmit

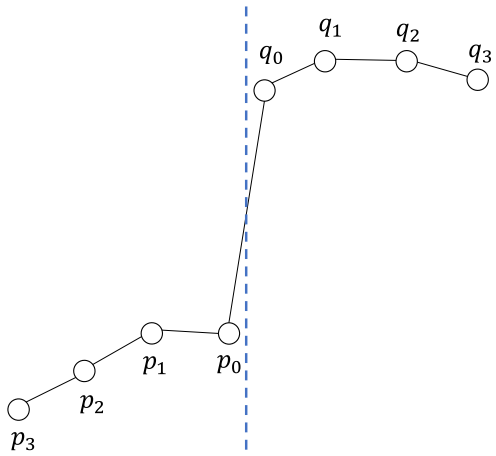
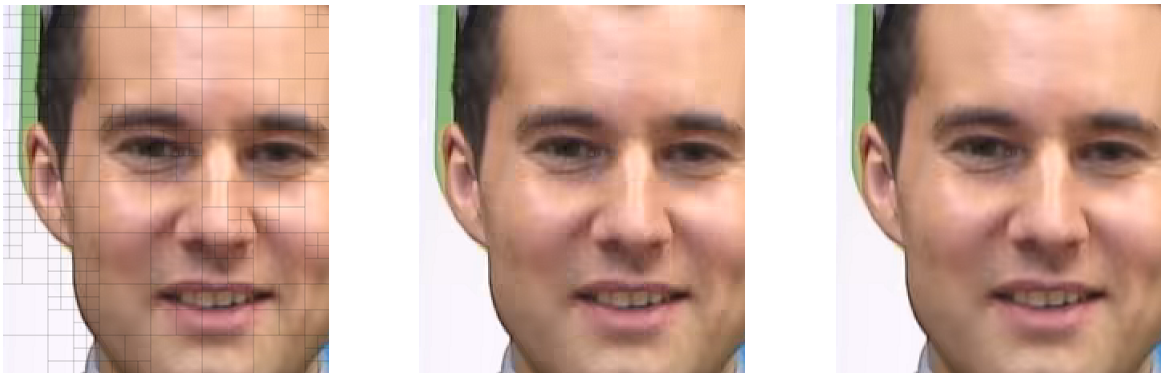


Figure 1.8 – A one-dimensional block border edge example.



a) Partitioning b) Distorted image with blockiness Deblocked image

Figure 1.9 – Impact of partitioning on block artefacts and the DBF as its efficient solution.

any side information. Figure 1.10 demonstrates how efficiently the SAO can remove the ringing artifact.

1.5.3 Adaptive Loop Filter

Adaptive Loop Filter (ALF) is one of the most advanced, efficient, and complex filters, integrated particularly in VVC. This filter is adaptive in the sense that the filtered coefficients are computed at the encoder side and are signaled in the bitstream. Moreover, its design is based on image content and distortion of the reconstructed picture.

The main idea of ALF is to apply a classification to divide sample locations into a set of pre-defined classes. Once classified, Wiener filters are calculated and applied for each class. To this end, two diamond filter shapes are used: 1) the 7×7 diamond shape for luma component and 2) the 5×5 diamond shape for chroma components.

a) SAO disabled (ringing artifact) b) SAO enabled

Figure 1.10 – SAO and its efficiency in removing the ringing artifact [1].

For luma, each 4×4 block is categorized into one out of 25 classes to determine its directionality and a quantized indicator of its activity. This includes the calculation of gradients in four directions for the reconstructed luma samples. Before filtering 4×4 luma blocks, geometric transformations such as rotation or diagonal and vertical flipping are applied. The idea is to differentiate between blocks based on their directionality for applying ALF. Finally, each sample location is classified into one of four classes and filtered by diamond-shaped filter. Figure 1.11 shows an example of how ALF chooses different parameters based on the content classification.

1.6 Encoder control

As codec specifications include more and more compression techniques, it will become more complicated for encoders to determine in which situations they should use each technique. Every time the encoder chooses a certain tuning of coding configuration for compression of a region of an image, it is typically said that it has made a “decision”. To make a decision, the encoder has to consider all or some available alternative decisions, that are generated through differently tuning of the same coding configuration. To do so, a cost function is used to be minimized on the set of considered alternative decisions.

The principal parameter impacting the encoder decision is called the QP. This user-defined parameter plays two key roles in video compression, by determining the trade-off between rate and distortion in two block-level domains: decision and residual coding.

1.6.1 Quantization Parameter (QP)

QP in a decision: Rate-distortion cost

In the heart of all nested loops of an encoder, there is always a cost function that turns the two-folded nature of the problem into a single metric. Precisely, the two-fold aspect means that neither minimizing the bitrate nor maximizing the reconstruction quality is merely the objective of the encoding optimization. While the goal is to jointly optimize them using a single metric.

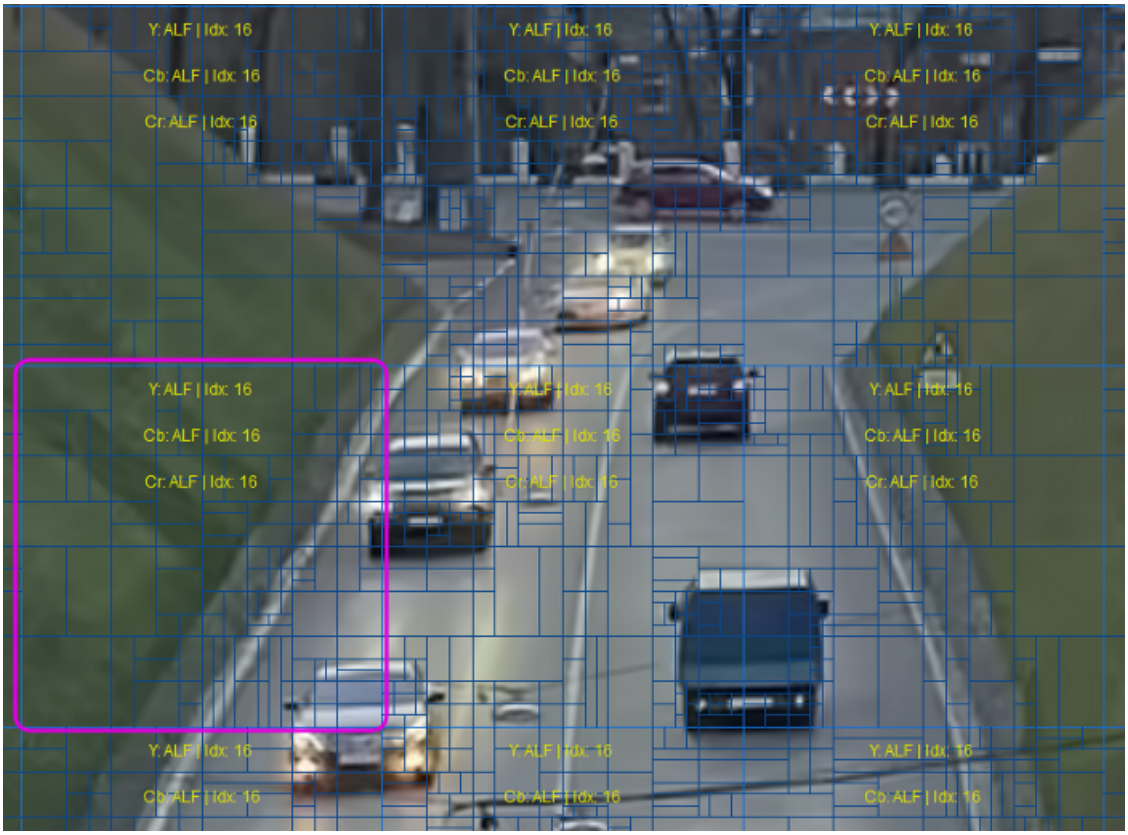


Figure 1.11 – ALF and how different filter indexes are used for luma and chroma of different regions of an image [2]

Even though neither the computation nor the utilization of this metric is not specified in any video coding standard, there is one typical method for it.

Let R^e and D^e be the bitrate and the distortion of encoding, respectively. Given R^c as the constraint in terms of available bitrate, the initial formulation of the Rate Distortion Optimization (RDO) can be expressed as:

$$\text{minimize } D^e \text{ , subject to: } R^e < R^c. \tag{1.1}$$

To solve the above optimization problem, a Lagrangian multiplier λ is used, as:

$$J = D^e + \lambda R^e, \tag{1.2}$$

where J is called the rate-distortion cost. In theory, given a fixed value of λ, minimizing this cost metric through out the encoding process would guarantee obtaining the optimal encoding configuration for a video. In practice, encoders deploy different heuristics in computation and utilization of the rate-distortion cost, mostly depending on their complexity constraints.

The Lagrangian parameter λ is computed based on the user-defined QP value. Particularly, the larger the QP becomes, the more constraints on bitrates must be applied, hence, the larger the weight of R in Eq. 1.2.

The process in which an encoder minimizes the rate-distortion cost is called the RDO. In this process, different possibilities for encoding image blocks are investigated. For each possibility which is represented by a set of coding parameters, the rate-distortion cost is computed by either precisely or approximately quantifying rate and distortion penalties due to the selected parameters. The final decision is the one that results in the smallest rate-distortion cost.

Rate penalty is simply caused by the fact that the chosen parameters, most importantly the residual signal, have to be written in the bitstream. Therefore, the metric of rate in this computation is simply the number of bits. However, the distortion penalty of a set of coding parameters is directly determined by the amount of information that is lost during the coefficient quantization [8].

QP in residual coding: Coefficient quantization

The quantization step is the core element of any lossy compression system. This typically non-linear process is responsible for mapping transform coefficient amplitudes to a predefined set of representative values [4]. Therefore, the compression process consists of important steps for controlling the quantization both within the coded frames and over the coded video sequence. The main goal of these steps is to optimize the proportional amount of irrelevant lost information compared to lost relevant information. Here, the relevance is typically defined with respect to the human visual system and what can be perceived and cannot be perceived in certain conditions.

1.6.2 Rate-distortion curve

To demonstrate the overall performance of a video codec in different ranges of bitrates, one can conduct several encodings to generate a so-called rate-distortion curve. To this end, different operational points of the given codec are plotted on a 2-dimensional axis, with typically x-axis being a bitrate metric (*e.g.* kilo-bit per second (kbps), megabit per second (mbps) *etc.*), and the other axis being a quality metric (*e.g.* Peak Signal-to-Noise Ratio (PSNR), Video Multi-Method Assessment Fusion (VMAF), Multi-Scale Structural SIMilarity (MS-SSIM) *etc.*). Figure 1.12 shows an example rate-distortion curve, with an arbitrary quality metric, where larger values indicate higher quality. A rate-distortion curve is convex, just as in the example.

There are different methods to variate the bitrate of a video codec to generate different operational points on the x-axis of the rate-distortion curve. These methods are known as rate-control modes. Here, we describe two rate-control modes that have been used throughout this thesis.

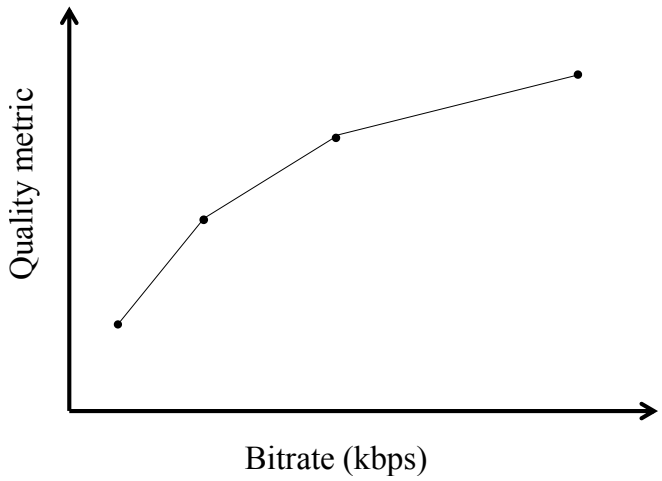


Figure 1.12 – An example of rate-distortion curve, where larger values of the arbitrary quality metric indicate better quality, that can be achieved by higher bitrates.

1.6.3 Rate-control: Constant QP (CQP)

The first mean to adjust the bitrate operational point of a video encoder is to change the QP value of the input parameters. This method is the most preferable when comparing the performance of essential tools of a codec as it provides rate-distortion decisions that are very close to optimal. As the QP value directly determines the levels of data loss during the transform quantization, variation of its value can serve for the purpose of generating different bitrate operational points.

Figure 1.13 shows an example of how two different video contents can result in entirely different rate-distortion curves when encoded with the same video encoder at the same values of QP. In this symbolic schema, since the statistical characteristics of the two videos were supposedly different, the resulting bitrates from each QP value are different between the two sequences.

Rate-control: Constant Bitrate (CBR)

The second mean to produce different bitrate operational points is to use the so-called Constant Bit-Rate (CBR) rate control mode. In general, the term CBR refers to transmitting any data at a constant rate, whereas for video transmission applications, this means that the encoder would output bitstream data at a constant rate, that is determined by the user. As a consequence, the CBR mode video encoding is not theoretically responsive to the size or content complexity of the input video that it processes. This rate control mode is commonly used in real-world video communication applications, as it guarantees that the target bitrate imposed

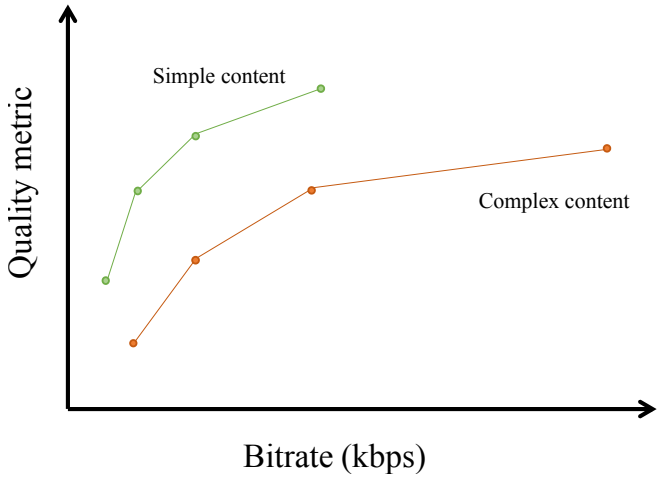


Figure 1.13 – Rate-distortion curve of two different types of content (simple and complex), generated by the CQP rate control mode..

by bandwidth limitation will be respected.

The principles of CBR is based on determining the QP value in finer granularity than in the frame or sequence level, as is the case in the CQP mode. Most typical granularity for determining the QP is in the block-level QP. To efficiently benefit from this granularity during encoding, first an initial frame-level (or sequence-level) QP is determined based on the target bitrate. Then, through a syntax element called the QP delta or the QP offset, each coding block is potentially coded with a different final QP value.

There are several methods to implement the CBR mode and decide block-level QPs in a video encoder. Moreover, since the rate-control module is an encoder-only (*i.e.* non-normative) functionality, most industrial encoders do not publicly disclose their optimized CBR rate-control algorithms. However, in the literature, most CBR algorithms perform a multi-pass (typically 2-pass) encoding, while the first encoding pass serves as a content analysis pass to provide information about spatial and temporal importance as well as rate consumption of each region of video frames. This information is then used to determine block-level QP values to attain the target bitrate in the second pass. Figure 1.14 shows an example of encoding the same sequences in Figure 1.14, but in the CBR mode. As can be seen, this time the operating points are almost perfectly aligned and the difference in the PSNR quality metric of the two sequences indicates their content complexities.

1.6.4 Codec performance metric

In this thesis, we often conduct codec performance comparisons as the first mean to assess the performance of our proposed methods. Codec performance comparison is a general practice

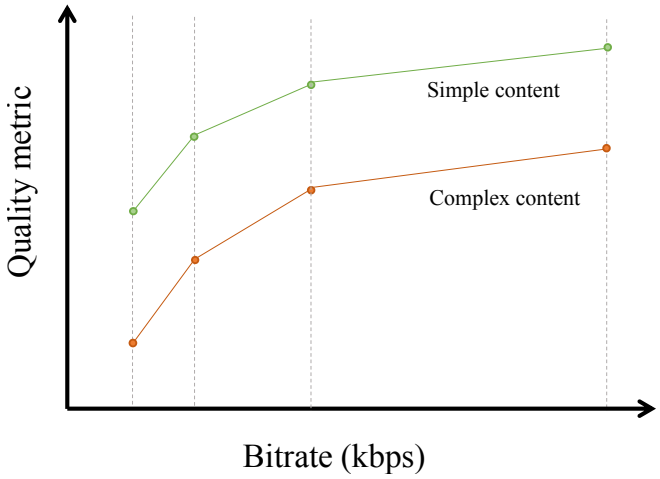


Figure 1.14 – Encoding two statistically different video sequences in the CBR mode, results in alignment of the bitrate operational points.

when developing novel video compression technologies and it is often based on rate-distortion curves. In such comparisons, the set of codecs under study could be either of the followings:

— Implementations of reference software of different video standards: *e.g.* JM of H.264/AVC, HM of H.265/HEVC, VTM of H.266/VVC, AOMEnc of AV1 *etc.*

— Different presets of an implementation of the same standards: *e.g.* x265-fast, x265-medium, x265-slow.

— Or even the very same preset of the same codec, but when activating and deactivating certain tool(s): *e.g.* VTM with MTT partitioning vs. VTM without MTT partitioning.

Using rate-distortion curves, there exist two common metrics, known as Bjøntegaard Delta (BD) metrics [9], to conduct codec performance comparison: BD-BR and BjøntegaardDelta-PSNR (BD-PSNR). These metrics are closely related, in the sense that they both involve bitrate and quality as the main two elements. Moreover, they both describe the average improvement of one element in the same level of the other element. Precisely, BD-bitrate indicates the average bitrate reduction in the same level of PSNR quality. Likewise, BD-PSNR indicates average improvement of PSNR at the same bitrate. Figure 1.15 visualizes the difference between the interpretations of BD-BR and BD-PSNR.

In this section, only the BD-BR computation will be described, as it is the main metric used in the experiments presented in the thesis studies. However, since the counterpart BD-PSNR metric is based on the same principle, one can obtain its computation simply by exchanging the two elements of bitrates and PSNR in the computation of BD-BR [10]:

To compute the BD-BR metric:

1. The bitrate and distortion points are calculated for the reference and experiment codecs.

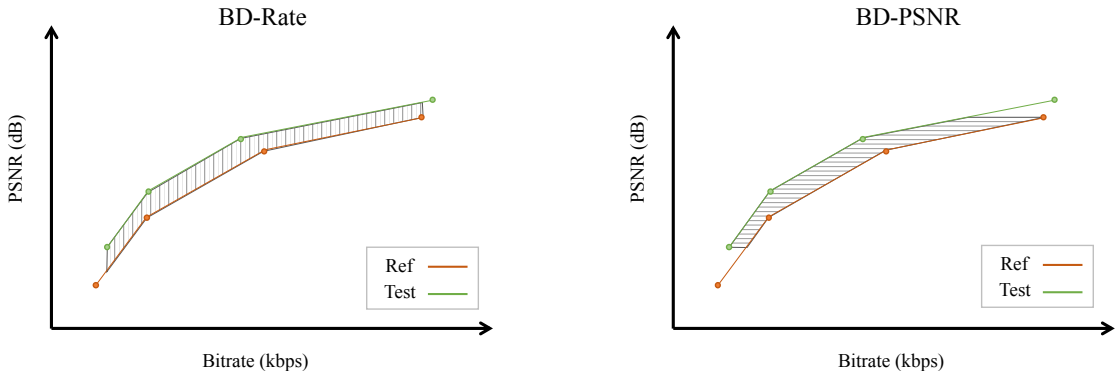


Figure 1.15 – Interpretation of BD-BR and BD-PSNR on the rate-distortion curve with the dashed area.

   - At least four points must be computed. These points should be obtained with the same quantizers when comparing two versions of the same codec.
   - Additional points outside of the range should be discarded.
2. The rates are converted into log-rates.
3. A piecewise cubic Hermite interpolating polynomial is fit to the points for each codec to produce functions of log-rate in terms of distortion.
4. Metric score ranges are computed:
   - If comparing two versions of the same codec, the overlap is the intersection of the two curves, bound by the chosen quantizer points.
   - If comparing dissimilar codecs, a third anchor codec's metric scores at fixed quantizers are used directly as the bounds.
5. The log-rate is numerically integrated over the metric range for each curve, using at least 1000 samples and trapezoidal integration.
6. The resulting integrated log-rates are converted back into linear rate, and then the percent difference is calculated from the reference to the test codec.

1.7 Conclusion

In this chapter, an overview of the principle of hybrid video coding was provided. VVC codec follows a similar structure as its ancestors, namely HEVC and AVC, in order to remove and minimize the signal correlation by taking into account various methods. Partitioning as the first starting process for a block-based compression algorithm plays a critical role in defining the proper split for effective compression. Thus, VVC has adopted more efficient partitioning method to further enhance compression performance. Moreover, it was discussed how spatial

and temporal correlations in video frames can be modeled and represented with appropriate parameters, and in each domain, which approaches are used in video codecs and how they are improved in VVC. As the next step, the prediction residual is transformed and quantized to compact the signal by efficiently throwing away the less significant information. Finally, entropy coding is performed to encode the quantized coefficients with a minimum number of bits.

Chapter 2

State of the art

2.1 Introduction

In this chapter, context information and state-of-the-art for two main contributions are presented, namely, CNN-based quality enhancement and Machine Learning (ML)-based bitrate ladder prediction. Given the background provided in the previous chapter, here for each contribution, we first present a problem definition and then describe some notable works in the stat-of-the-art that are somehow related to our research.

2.2 CNN-based quality enhancement

2.2.1 Problem definition

The first contribution of this work aims at improving the quality of reconstructed pixels with the help of Artificial Intelligence (AI). Since CNNs have proved miraculously efficient on 2-dimensional signals such as image and video, all the work developed in this part of this thesis is based on CNN. The main goal of CNN-based QE is to remove artifacts from coded videos. These artifacts are usually introduced to the compressed signal due to the limitation in their coded bitrate. Therefore, one can expect that in low bitrate and very low bitrate coding, which is the main domain of this thesis, they are more undesirable.

Figure 2.1 shows two opportunities in a video coding workflow to benefit from CNN-based QE methods: Post Processing (PP) and In-Loop Filtering (ILF). Even though the two applications have subtly different characteristics and challenges to overcome, the use of a CNN-based QE module has the same high-level formulation in terms of input, internal process, and output. In both cases, the input is a supposedly distorted decompressed image, with possibly additional information, and the expected output is a modified version of the image with better subjective and/or objective quality. What happens inside the CNN-based QE is typically a series of convolutional layers with optional non-convolutional layers.

In this section, some studies with significant contributions to the CNN-based video QE task are reviewed. As the proposed framework of our approach particularly focuses on the use of coding information, these studies are categorized and ordered to reflect how much they take

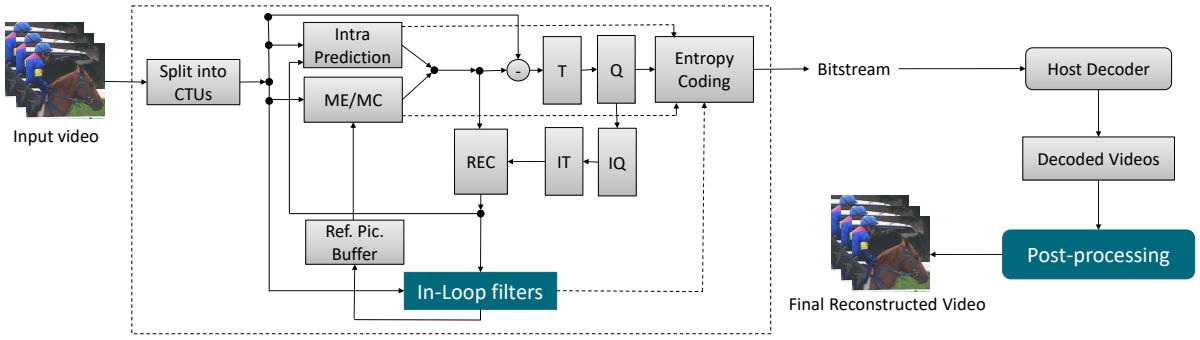


Figure 2.1 – Two opportunities where a CNN-based QE module can be integrated in a video compression workflow.

into account the nature of compressed video signal and how their method exploits spatial and temporal correlations for the QE task. Table 2.1 provides a list of the most relevant papers published in the past few years and summarizes their principle contributions, with a focus on the use of coding information..

2.2.2 Single-Frame Quality Enhancement

Most CNN-based QE methods enhance the quality of video in a frame-by-frame manner, where each frame is enhanced independently. These methods exploit spatial information of texture pixels of individual frames in order to enhance their quality and remove their artifacts. One of the early works in this category, proposed in [11], uses a network with three convolutional layers to learn the residual information. This method is implemented as ILF and replaces SAO filter of HEVC. Another method called Deep CNN-based Auto Encoder (DCAD), deploys a relatively deeper network with ten layers to be used as a PP filter after decoding [12]. Inspired by the diversity of block sizes in HEVC, an ILF named Variable-filter-size Residue-learning CNN (VRCNN) proposes a network with different filter sizes to replace both SAO and DBF filters of HEVC intra coding [13]. The method presented in [14] enhances the performance of VRCNN by introducing more non-linearity to the VRCNN network. The added ReLU [15] and batch normalization [16] layers in this method improve its performance, compared to VRCNN.

In another work, presented in [17], two different networks are trained for intra and inter frames. The intra network is a sub-net of the inter network which helps the method to capture artifacts of intra coded blocks in P and B frames more efficiently. Furthermore, in [17], the complexity of the QE filter is controlled by comparison of a Mean Squared Error (MSE)-based distortion metric in the CTU level at the encoder side. The Residual Highway CNN (RHCNN) network, presented in [18], is composed of several cascaded residual blocks, each of which has two convolutional layers, followed by a ReLU activation function. In RHCNN, inter and intra coded

frames are also enhanced with dedicated networks. In [19], residual blocks in the network are enhanced by splitting the input frame and processing each part with different CNN branches. The output of each branch is then concatenated and fed to the next block. As another contribution, a weighted normalization scheme is used instead of batch normalization which also improves the training process.

More recently, Multi-scale Mean value of CU-Progressive Rethinking Network (MM-CU-PRN) loop filter has been introduced [20] which uses progressive rethinking block and additional skip connections between blocks, helping the network to keep a longer memory during the training and be able to use low-level features in deeper layers. MM-CU-PRN is placed between DBF and SAO filters and benefits from coding information by using a multi-scale mean value of CUs. The network presented in [21], Multi-level Feature review Residual dense Network (MFRNet), deploys similar network architecture as in MM-CU-PRN. However, MFRNet utilizes multi-level residual learning while reviewing (reusing) high dimensional features in the input of each residual block which leads to a network with better performance compared to existing networks.

Multi-Reconstruction Recurrent Residual Network (MRRN) is a method based on recursive learning and is implemented as PP filter for decoded frames of HEVC [22]. In recursive learning, the same layers are repeatedly applied which reduces the probability of over-fitting during the training. Likewise, in [23], another recursive residual network is proposed as ILF for intra coded frames. The proposed network in [23] is applied on reconstructed frames before the DBF and SAO filters. Block Information Constrained Deep Recursive Residual Network (BDRRN) is another method based on recursive residual learning in which a block-based mean-mask, as well as the boundary-mask, are used as input to the network [24].

Furthermore, in some works, the focus is put on strategies for enhancing the quality of video frames. In [25], a blind quality enhancement approach is proposed where frames with different distortion levels are processed differently. An "easy-to-hard" QE strategy is used to determine which level of the CNN-based filtering shall be applied on a given frame. In the case that the quality is already satisfying based on a blind quality assessment metric, the QE process stops, otherwise, it continues. In Squeeze-and-Excitation Filtering CNN (SEFCNN) [26], an adaptive ILF is also proposed in which networks with various complexity levels are trained for different QPs. In Multi-level Progressive Refinement Network (MPRN) [27], a Generative Adversarial Network (GAN)-based post-processing filter for intra coded frames is presented to be used instead of SAO and DBF. The generator network utilizes a progressive refinement strategy to generate enhanced frames.

Table 2.1 – An overview of recently published CNN-based QE methods in the literature, with a summary of their contribution as well as the type of coding information they use (e.g. Transform Units Map (TM) , Coding Units Map (CM), Residual Map (RM)), Prediction Map (PM), Mean Map (MM), Intra Prediction Mode Map (IPMM) and Coding Type (CT).

Method	Published	Coding information	QE Func	Summary of contribution
IFCNN	IVMSP 16 [11]	QP	ILF	Applied after DB instead of SAO. 3 layers CNN with residual learning.
STResNet	VCIP 17 [28]	QP	ILF	After SAO, uses previous reconstructed block. 4 CNN layers with residual learning
DCAD	DCC 17 [12]	QP	PP	A 10 layers CNN network with residual learning.
DSCNN	ICME 17 [29]	QP	PP	Scalable network with separate branches for inter and intra frames
MMS-net	ICIP 17 [30]	QP TM	PP	Replaces all HEVC loop filters in intra coded frames. Scalable training.
VRCNN	MMM 17 [13]	QP	PP	Replaces HEVC in-loop filters in intra mode, variable CNN filter sizes.
MSDD	DCC 18 [31]	QP	PP	Multi-frame input (next and previous frames) with multi-scale training
-	ICIP 18 [32]	QP BM MM	PP	Mean and partitioning mask with reconstructed frames are fed to a residual-based net.
QECNN	IEEE-TCSVT 18 [17]	QP	PP	Two networks with different filter sizes for inter and intra frames, time constrained QE
-	IEEE access 18 [33]	QP	PP	Temporally adjacent similar patches are also fed to an inception-based net.
MFQE	CVPR 18 [34]	QP	PP	Current and motion compensated frames of high quality adjacent frames are fed to net.
CNNF	ICIP 18 [35]	QP	ILF	QP and reconstructed frame are fed to network, replacing SAO and DBF.
RHCNN	IEEE-TIP 18 [18]	QP	ILF	QP-specific training of a network based on several residual highway units
FECNN	ICIP 18 [36]	QP	PP	A residual based network with two skip connections proposed for intra frames.
R-VRN	BigMM 18 [37]	QP RM PM	PP	Prediction and quantized residual frame are fed to a residual based network as input.
MGANet	arXiv 18 [38]	QP TM	PP	TM is also fed to a multi-scale net. which exploits output of a temporal encoder.
ADCNN	IEEE access 19 [39]	QP TM	ILF	Network composed of attentions blocks, using also QP and TU map.
MM-CU-PRN	ICIP 19 [20]	QPCM	PP	Based on Progressive Rethinking Block which multi-scale CU maps are also used.
SDTS	ICIP 19 [40]	QP	PP	Multi frame QE scheme, using motion compensated frames and an improved network.
VRCNN-BN	IEEE access 19 [14]	QP	PP	Adds further non-linearity to VRCNN by adding batch normalization and Relu layers.
MIF	IEEE-TIP 19 [41]	QPCM TM	ILF	Selects high quality references to the current frame and exploits them in the QE.
-	APSIPA ASC 19 [42]	QPCM TM	PP	A network based on residual learning, exploiting TU and CU maps.
MRRN	IEEE SPL 19 [22]	QP	PP	Adopts a multi-reconstruction recurrent residual network for PP-QE task.
RRCNN	IEEE-TCSVT 19 [23]	QP	ILF	Intra , Recursive structure and Residual units with local skip connections
B-DRRN	PCS 19 [24]	QPCM MM	PP	Network based on recursive residual learning, exploiting mean and boundary mask.
CPHER	ICIP 19 [43]	QP PM	PP	Network based on residual blocks, exploiting unfiltered frame and prediction.
WARN	ICIP 19 [44]	QP	ILF	A wide activation residual network for ILF of AV1 codec.
ACRN	ICGIP 19 [45]	QP	ILF	Asymmetric residual network as ILF in AV1, with a more complex net. for higher QPs.
-	IEEE-TIP 19 [46]	QP	PP	Based on Kalman filters, using temporal information restored from previous frames.
SimNet	PCS 19 [47]	QP	PP	Depth of network is varied based on the distortion level.
LMVE	ICIP 19 [48]	QP	PP	Single and multi frame QE net. proposed, using FlowNet to generate high quality MC.
-	CVPR 19 [49]	QP	PP	Residual block based network which receives different scales of input frame.
SEFCNN	IEEE-TCSVT 19 [26]	QP CT	ILF	Optional ILF with adaptive net. selection for different CT and distortion levels.
DIANet	PCS 19 [50]	QP	ILF	Dense inception net. with different attention blocks, separating inter/intra frames.
-	IEEE-TIP 19 [51]	QP	ILF	Content-aware ILF with adaptive network selection depending on CTU content
MFRNet	arXiv 20 [21]	QP	ILF	An architecture based on multi-level dense residual blocks with feature review.
EDCNN	IEEE-TIP 20 [19]	QP	ILF	Network with enhanced residual blocks with weight normalization.
BSTN	MIPR 20 [52]	TM MM	PP	MC frames along with distorted frame and additional coding information are fed to net.
FGTSN	DCC 20 [53]	QP	PP	Flow-guided multi-scale net. using motion field extracted from neighboring frames.
-	IEEE access 20 [54]	QP	PP	Sparse coding based reconstruction frame fed to net. with MC and distorted frames.
-	ACM 20 [55]	QP	PP	Fine-tuned QE network transmitting modified weights via bitstream.
FQE-CNN	IEEE-TCSVT 20 [56]	QP IPM	PP	Image size patches used for training, using intra modes map.
-	ICME 20 [57]	QP	PP	Post-processing for VVC encoded frames with network based on residual blocks.
RBQE	arXiv 20 [25]	QP	PP	Blind QE with an easy-to-hard paradigm, based on dynamic neural net.
PQEN-ND	NC 20 [58]	QP	ILF	Noise characteristic extracted from frames for enhancing intra and inter frames.
QEVC	ICCCS 20 [59]	QP	ILF	Depending on motion, a selector network selects different networks for QE.
MSGDN	CVPRW 20 [60]	QP	PP	A multi-scale grouped dense network as a post-processing of VVC intra coding
PMVE	IEEE-TC 20 [61]	QP	PP	Frames are enhanced by contributing prediction info. from neighboring HQ frames.
MWGAN	arXiv 20 [62]	QP	PP	GAN multi-frame wavelet-based net., recovering high frequency sub-bands.
STEFCNN	DCC 20 [63]	QP	PP	Multi-frame QE method with a dense residual block based pre-denoising stage
MPRNET	NC 20 [27]	QP	PP	GAN multi-level progressive refinement, replacing the DBF and SAO in HEVC
RRDB	UCET 20 [64]	QP	PP	A GAN-based network for QE of Intra coded frames of HEVC

2.2.3 Multi-frame Quality Enhancement

Multiframe QE methods process a set of consecutive frames as input. The basic idea behind this category of methods is to remove the compression artifacts while considering the temporal correlation of video content. Moreover, the quality is propagated from frames encoded at high quality to adjacent frames encoded at lower quality.

One of the earliest implementations exploits temporal information simply by adding previously decoded frames to the input of the network, along with the current frame [28]. In another method, Peak Quality Frames (PQFs) are detected with an Support Vector Machine (SVM)-based classifier [34]. Using a network named Motion Compensated (MC) sub-net, the MC frames of previous and next PQFs are generated. The three frames are then fed to another network, called QE sub-net, in order to enhance the quality of the current frame, while the selected PQFs are kept to be enhanced by another dedicated network. An improved version of this method has been introduced in [65, 40], where the high quality frame detection, as well as the QE network itself, are improved.

In multiframe QE methods, finding the best similar frames to the current frame for the task of motion compensation is important. In [41], similar reference frames with higher quality than the current frame are detected with a dedicated network. Then they are used to generate the MC frames with respect to the content of the current frame. This frame with computed motions is then used as input to the QE network along with the reconstructed frame. In another research, a flow-guided network is proposed, where the motion field is extracted from previous and next frames using FlowNet [66, 53]. Once the motion compensation is completed, a multi-scale network is applied to extract spatial and temporal features from the input. Following the same principle, motion compensated frames of adjacent frames are fed to the network in [52]. A ConvLSTM-based network is then used to implicitly discover frame variations over time between the compensated adjacent frames and the current frame. Moreover, in order to capture the texture distortion in compressed frames, the Transform Unit (TU) mean map is also fed to the network. In [54], in addition to the motion compensated frame, a sparse coding based reconstruction frame is also fed to the network as input. The purpose of using sparse coding prediction is to simplify the process of texture learning by the network. Similarly, in [33], most similar patches in previous and next frames are extracted and fed to a network with three branches of stacked convolutional layers. The branches are then concatenated to reconstruct the final patch.

Inspired by the multi-frame QE methods, a bi-prediction approach is proposed in [61]. In this work, instead of computing the motion field, a prediction of the current low quality frame is generated from neighbouring high quality frames. Then the predicted frame and reconstructed frame are fed to the QE network. In [62], a GAN-based multi-frame method is presented in which adjacent frames and current frame are fed to a GAN, which is itself composed of two

parts: one for the motion compensation and the other for quality enhancement. Wavelet sub-bands of motion compensated frames and reconstructed frames are fed to the second part of the network as input. For evaluation of generated content, a wavelet-based discriminator that extracts features in the wavelet domain at several levels (*i.e.* sub-bands) is proposed. In [67] they propose a multi-frame method that uses neighboring compressed frames. As loss function, they use an FFT-based loss function in order to complete the missing high-frequency information.

In summary, multi-frame solutions adopt different motion compensation methods mostly based on block-matching or CNN-based approaches. However, they all overlook the fact that the actual motion modeling is performed based on a normative process which takes into account complex factors such as bitrate restriction or the internal state of the encoder modules. In other words, one might consider the normative motion information available in the bitstream more useful than texture-based heuristic motion modeling. Furthermore, this signal is already available both at the encoder and decoder sides and can easily be used as side information for inference in the QE networks of PP and ILF methods.

2.2.4 Codec Specific

Roughly, most works in the domain of CNN-based QE are developed to enhance HEVC coded videos. However, significant works are also conducted to improve the quality of decoded frames in other codecs. For AV1 codec, in [44] an in-loop filter based on wide activation residual network is proposed that is applied on the specific temporal layers of one GoP in order to prevent the PSNR loss. In [68], the method named Asymmetric Convolutional Residual Network (ACRN) is proposed in which two different architectures are used for different ranges of QPs. Similarly, in [47] an in-loop filter that has different number of CNN layers for different QPs is proposed in which the QE filter is only applied on specific temporal layers in GoP to avoid double enhancing by QE filter.

More recently enhancing the quality of the latest codec of MPEG, VVC, has attracted a lot of attention. In [39] an attention-based in-loop filter for VVC is proposed where different regions of the frames are enhanced based on their distortion. The chroma and luma components are enhanced through one CNN where the QP and CU map of each component is fed to the network separately. Moreover, in [42] a network to be served as only in-loop filter instead of all other existing filter is proposed. The QP map and CU map are also fed to the network to improve the performance of the QE filter. Finally, in [55] a PP filter is proposed in which a pre-trained network is fine-tuned to enhance the quality of each video before transmission. During the fine-tuning, only the biases are changed and their difference with original biases are transmitted.

In order to verify the capacity of our proposed QE method in removing artifacts, we have integrated our PP and ILF into the VVC codec. The new and complex tools which are introduce

in the new codec affect the artifacts pattern.

2.2.5 Methods based on coding information

In the literature, there are diverse levels of involving coding information in the CNN-training of the QE task. Here, we categorize these methods from the most basic coding information to the most advanced ones.

Quantization Parameter (QP)

A common basic coding information and one of the most useful one is QP. Most of methods somehow involve the applied QP of the encoded signal in the training and the inference phases. There are mainly two approaches to use QP in CNN-based QE:

— QP-specific training: dedicating one model for each QP or a range of QPs [21, 69].

— QP-map training: providing QP as an input to the network [39, 42, 23, 55, 61].

Each approach has benefits and drawbacks. In the QP-specific training methods, the performance is usually higher as the artifacts of each QP have been particularly observed by their dedicated network during the training. However, they usually require storing several trained models at the decoder-side which is not hardware-friendly. On contrary, QP-map methods are usually lighter to implement, especially when the QP value varies in finer granularity such as frame-level or block-level.

Partitioning

Another common coding information is block partitioning and boundary information. Depending on the flexibility of the codec under study (*e.g.* HEVC, AV1, VVC), this aspect is used differently in the literature. The simplest form of partitioning information is the boundary mask [32]. More sophisticated methods, especially HEVC-based ones, differentiate between Coding Units (CUs), Prediction Units (PUs) and Transform Units (TUs) boundaries [30, 32, 39, 41, 42, 24, 52].

Prediction information

Spatial and temporal prediction information has also been used for the enhancement of coded videos. In intra coding mode, the simplest prediction representation is the mean-mask of intra blocks [32]. Other methods use actual intra prediction signal associated to intra blocks. In [32], coding information such as the partitioning map as well as a mean-mask have been used as input to their proposed CNN-based QE network. The mean-mask is computed based on average reconstructed pixel values in each partition. In this method, a network with several

CNN-based residual units is used which takes two signals as input: reconstructed frame, mean-mask. In another work, presented in [43], a QE network is proposed in which the unfiltered frame and prediction frame are used along with the reconstructed frame as the input of the network. Finally, a three-level network composed of Inception and Residual Learning based Block (IResLB) units is proposed. The Inception and Residual Learning based Block (IResLB) units have three branches, each one having one to two convolutional layers. The intra mode map is then fed to the network to enhance the intra coded frames [56].

Regarding inter coding mode, an ILF with a selector network has been proposed in [59]. In this method, the selector network determines the motion complexity of a set of selected CUs and then decides whether to increase or decrease the QP value of CU and also which network (large or small scale network) to be used.

Residual information

Finally, the residual information has also been used as additional input information for the task of encoded video enhancement. In [37], the coded residual information and prediction information are fed to a network to enhance intra coded frames in HEVC. The proposed network uses direct current (DC)-ReLU activation function in the first residual layer. The loss function in this work is a combination of MS-SSIM, L1 and L2 functions. In another work, presented in [70], the QE task is modeled as a Kalman filtering procedure and enhance the quality through a deep Kalman network. To further improve the performance of the network, it uses prediction residuals as prior information.

To best of our knowledge, our work is the first QE framework in which the spatial and temporal prediction information in frame and block levels are used for compressed video. In Chapter 3, the details of proposed framework are explained and integration in the VVC codec at both PP and ILF is presented.

2.3 Bitrate ladder construction

2.3.1 Problem definition

Video service providers dedicate considerable resources to optimizing video compression parameters before transmission. This optimization allows them to deliver the highest level of video quality possible while improving user satisfaction and meeting varying user constraints. Network bandwidth heterogeneity, varying users' display size, and various video contents with different spatio-temporal features are all factors that could impact the performance of live video streaming or VoD services. As a result, Dynamic Adaptive Streaming over HTTP (DASH) [71] and HTTP Live Streaming (HLS) [72] are two main industrial technologies that have been widely adopted in the media industry to incorporate heterogeneous network conditions. In both technologies, the input video is potentially down-sampled from its native resolution changes before encoding, in order to meet the available constraints such as bandwidth, complexity and latency.

The traditional approach to change the resolution is performed by employing the so-called "bitrate ladder". A bitrate ladder recommends the resolution for a given bitrate, by dividing the bitrate range into a set of predefined bitrate intervals and associating ascending resolutions to consecutive intervals. The simplest implementation of this idea is called static bitrate ladder or "one-size-fits-all", where one single ladder is optimized for all types of video content. The main drawback of a static bitrate ladder is that its recommendation scheme is the same for all video contents, regardless of their spatio-temporal features.

The above problem is addressed in the second contribution of this thesis. Precisely, we propose ML-based methods to determine the best resolution in which a given sequence should be encoded in a given bitrate. There are mainly two applications where such solutions could be useful, as depicted in Figure 2.2. In both applications, a high-resolution video, also known as "mezzanine", is either captured or stored, and the goal is to determine which encoded resolution should be transmitted. First, in live applications in which the sender up-link is under strict bandwidth constraints, the bitrate ladder can optimize the encoded resolution to ensure the full capacity of the network is exploited. It is noteworthy that this application which is particularly important for AVIWEST's products. Second, in recently emerged on-demand streaming services, where the main bandwidth limitation is imposed at the receiver side, the sender determines which resolution of the encoded video must be delivered to each user. In this application, several combinations of bitrate and resolution have to be encoded and stored. And the main task is to avoid redundant encodings in such combinations that are not optimal for either receiver-side bandwidth limitation.

In this section, a review of existing methods for addressing this problem is presented. These methods include both heuristic and ML-based methods.

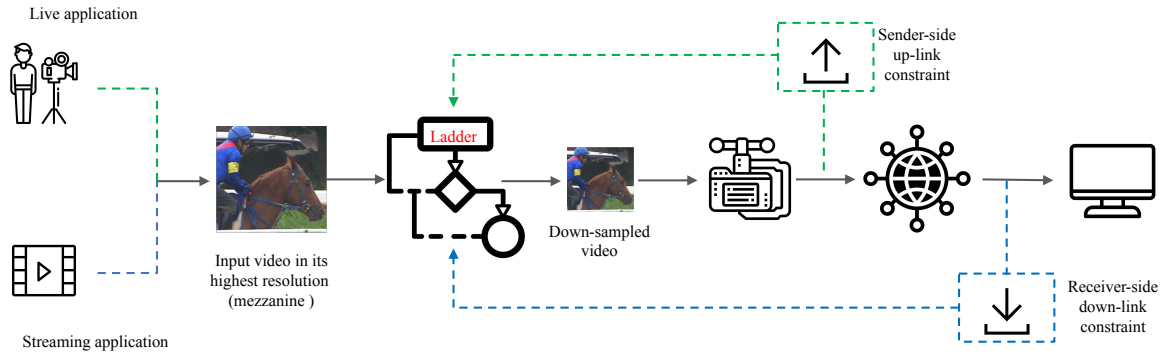


Figure 2.2 – Two video delivery ecosystems and how the bitrate ladder prediction serves in each one.

2.3.2 Heuristic methods

One solution to involve the content is to employ separate bitrate ladders for videos with different genres. For instance, the work presented in [3] provides three various ladders for three categories of content including animation, sport, and movie. In table 2.2 these three categories and their behavior in different bitrate ranges are shown. As can be seen, the different spatio-temporal features in each category of videos lead to different resolutions in the same bitrates.

Videos categorized in a specific genre have still a large variety of characteristics in terms of motion and spatial textures which directly affects the rungs of the bitrate ladder. Considering this behaviour, recently some advanced methods are proposed to overcome the issues with fixed bitrate ladder. First category of solutions rely on exhaustive encodings, while in some cases they propose to accelerate the encodings to make their complexity affordable. In the second category of solutions, the additional encodings are partially or entirely replaced by methods that directly or indirectly predict the ladder mostly with ML methods.

In per-title and per-chunk encoding methods proposed by Netflix [73], the R-D curve of different titles in several resolutions is calculated by running several encodings in each chunk of the video. These R-D curves are then used to form the convex hull to obtain the optimal parameters for a defined bitrate range. In [74], Netflix improve their per-title encoding scheme by using the VMAF as the quality metric instead of PSNR. They also perform complexity analysis before constructing the bitrate ladder and further tuning the predefined encoding parameters.

The work presented in [75] develops a novel Integer Linear Program (ILP) to find the optimal set of representations, which maximizes user satisfaction under network and system constraints. The level of customer satisfaction is determined by the encoding rate, the resolution, the characteristics of the video requested, and the bandwidth available for delivering the video. The optimal representation sets are obtained through solving the ILP generically on representative cases. Moreover, In [76], a method is proposed to identify the best-fit bitrate for all video segments

No	Animation	Sport	Movie
1	50 kbit/s, 320×240	100 kbit/s, 320×240	50 kbit/s, 320×240
2	100 kbit/s, 320×240	150 kbit/s, 320×240	100 kbit/s, 320×240
3	150 kbit/s, 320×240	200 kbit/s, 480×360	150 kbit/s, 320×240
4	200 kbit/s, 480×360	250 kbit/s, 480×360	200 kbit/s, 480×360
5	250 kbit/s, 480×360	300 kbit/s, 480×360	250 kbit/s, 480×360
6	300 kbit/s, 480×360	400 kbit/s, 480×360	300 kbit/s, 480×360
7	400 kbit/s, 480×360	500 kbit/s, 854×480	400 kbit/s, 480×360
8	500 kbit/s, 480×360	700 kbit/s, 854×480	500 kbit/s, 854×480
9	600 kbit/s, 854×480	900 kbit/s, 854×480	600 kbit/s, 854×480
10	700 kbit/s, 854×480	1,2 Mbit/s, 854×480	700 kbit/s, 854×480
11	900 kbit/s,1280×720	1,5 Mbit/s,1280×720	900 kbit/s,1280×720
12	1,2 Mbit/s,1280×720	2,0 Mbit/s,1280×720	1,2 Mbit/s,1280×720
13	1,5 Mbit/s,1280×720	2,5 Mbit/s,1280×720	1,5 Mbit/s,1280×720
14	2,0 Mbit/s,1280×720	3,0 Mbit/s,1920×1080	2,0 Mbit/s,1920×1080
15	2,5 Mbit/s,1920×1080	4,0 Mbit/s,1920×1080	2,5 Mbit/s,1920×1080
16	3,0 Mbit/s,1920×1080	5,0 Mbit/s,1920×1080	3,0 Mbit/s,1920×1080
17	4,0 Mbit/s,1920×1080	6,0 Mbit/s,1920×1080	4,0 Mbit/s,1920×1080
18	5,0 Mbit/s,1920×1080	5,0 Mbit/s,1920×1080	
19	6,0 Mbit/s,1920×1080	6,0 Mbit/s,1920×1080	
20	8,0 Mbit/s,1920×1080		

Table 2.2 – Bitrate ladder for different genres of the videos [3]

based on their complexity using a Constant Rate Factor (CRF) based multi-pass encoding.

Furthermore, in the work presented in [77], the probability distribution of player-estimated bandwidth and viewport size are modeled as two stationary random processes. These probability distributions are created based on the measurements on actual usage of the millions of video clips. Then, an optimization process is performed based on created probability distributions to preserve the best possible quality in a given bitrate.

Brightcove proposes a method in [78], that takes into account the R-D characteristics of the source, client, and network models used for delivery, and formulates the problem of optimal design of encoding profiles for Adaptive Bitrate Selection (ABR) streaming. Their results demonstrate that ladders designed for different networks and sources are not the same. In another work [79], a similar method is proposed to build a multi-codec bitrate ladder. In this work, two codecs of AVC and HEVC are considered to be used by clients.

In order to reduce the complexity of the exhaustive search encodings, in the work presented in [80], the coding information extracted from encodings in the lowest resolution is used to speed

up the encoding process at higher resolutions. This method derives the coding decisions including CU quad-tree structure and PU predictions, coding modes, and MVs information from low resolution video to reduce the overall number of RDO calculations in higher resolutions. Moreover, in [81], an Artificial Neural Network (ANN) based approach is used for a fast multi-resolution and multi-rate encoding. For multi-rate encoding, the lowest bitrate representation and for multi-resolution encoding, the highest bitrate from the lowest resolution representation is chosen as the reference, respectively. Then the CTU split decisions are predicted by using the pixel values from the target resolution and encoding information from the reference representation.

More recently, the Multi-Period Per-Scene Optimization (MiPSO) method presented in [82] proposed a per-scene optimization framework for VoD HAS applications to determine the maximum quality or minimum bitrate for various encoded representations. In their proposed method, the different encoded representations of video content are examined to obtain the best quality-bitrate combination. More precisely, first, a set of quality-bitrate points are extracted from some encodings they perform in *representative segments* for calculating complexity. Then, they construct the convex hull of R-D curves in different resolutions in each scene. To detect video scenes, they used a threshold-based algorithm implemented in an intelligent scene cut detection and video splitting tool called PySceneDetect[83].

In [84] an approach to perceptually optimize ABR ladder for the web streaming is proposed. In this method, the size of the video player window which is used to render the decoded video on the user's screen has also been taken into account for determining the optimal ladder. Moreover, the method in [85] computes content complexity and also uses historical network throughput data to construct the bitrate ladder. As a quality metric, they use methods introduced in [86].

Considering the temporal resolution, the study in [87] proposes to construct the bitrate ladder based on spatial and temporal dimension. They demonstrate how different videos have different behavior when they are encoded in different temporal resolutions. The other approach introduced in [88] aims to solve the optimal laddering problem that determines the optimal encoding ladder to maximize the client viewing quality in 360' videos.

2.3.3 Machine learning based methods

The second category of solutions involves directly or indirectly predicting the ladder in place of the additional encodings. In one of the simplest realizations of this category, the work presented in [3] provides separate bitrate ladders for different pre-defined categories of video content. As a result, each new video has to be first classified, then adopt one of the provided ladders. In another solution proposed by Bitmovin [89], first, a variety of features such as frame rate, resolution and resulting bitrate from multiple encodings are extracted from the source video. Then, an ML based method is used to predict the convex hull and adjust an optimized profile for encoding the video. In their method, the complexity of encoding has also been taken

into account for choosing the best profile for encoding. Likewise, Cambria [90] proposes a method named Source Adaptive Bitrate Ladder (SABL). They run a fast Constant Rate Factor (CRF) encoding to estimate the encoding complexity using a simple ML-based technique. The obtained results from this encoding are then used to adjust the encoding ladder up or down.

Moreover, MUX [91] proposes a neural network based solution for estimating the bitrate ladder in which the new videos loaded into the network are contributed back to the training set. Another approach, introduced in [92], uses neural networks to encode the sequences in an optimal resolution. The decision is made for each GoP of the video. Furthermore, the work presented in [93] proposes a method to predict the QPs of the cross-over points between the R-D curves of two consecutive resolutions. First, they extract the Pareto front (convex hull) of a large dataset in four different resolutions as well as the cross-over QPs. In the next step, they use this information to train a supervised regression method to predict the several QP values on the convex hull curve. At the final step, they compute the corresponding bitrate based on the predicted QP by performing several encodings. They compare the performance of their feature-based prediction ladder with other approaches including interpolation-based ladder and hybrid ladder.

In [94] based on VMAF, the per-title bitrate ladder is predicted using Random Forest Regression (RFR), multi-layer perceptron, and Support Vector Regression (SVR) without running test encodings. In addition, in [95] based on SVR, perceptual quality, and some test encoding, a method is proposed to optimize the bitrate ladder by generating the R-D points while retaining a constant Just-Noticeable Difference (JND) [96]. Moreover, in [97], authors propose a real-time resolution prediction for low-bitrate applications. First, they analyze the first few frames of a video sequence and then by using a binary classification, they find an optimal resolution for encoding the whole video. Finally, in [98], a fast per-scene encoding method is proposed based on using a neural network for predicting the quality metric of video segments. This method benefits from different sets of input features and a fast entropy-based video scene detection approach where uses TI of the video encoded at low resolution to split videos into scenes.

2.4 Conclusion

In this chapter, an overview of research works related to our contribution in this thesis was presented. The chapter is divided into two parts in which the inspiring works and algorithms are presented in different categories. In the first part, the two different approaches of PP and ILF that CNN-based quality enhancement methods can be served during encoding, are explained. The related works have been divided into several categories, based on their specific approaches for removing the coding artifacts. In single-Frame QE methods, the video is enhanced frame by frame while in multi-frame based methods, several consecutive frames are taken into account

for QE task. Moreover, the methods that target specific codecs have been distinguished and the focus was to explore the methods that have integrated their methods on top of VVC. In the last category, the methods that exploit the coding information such as QP, partitioning and prediction information have been described.

In the second part of the chapter, the focus was put on bitrate ladder construction methods. Similarly, the existing methods have been presented in two categories to demonstrate their contribution more clearly. First, in the heuristic category, the methods that rely on exhaustive encodings or other approaches to accelerate encodings have been explained. Then, in the second category, the ML-based methods that directly predict the bitrate ladder from specific input features are discussed.

Along with the contextual background information presented in Chapter 1, the information provided in this chapter is necessary for better understating of the novelties in our contributions. These contributions are presented in the following two chapters, categorized by their theme: quality enhancement and adaptive streaming.

Chapter 3

Compression-Aware Quality Enhancement

3.1 Introduction

Video codecs aim at reducing the bitrate of compressed videos to decrease the traffic pressure on the transmission networks. As this process directly affects the perceived quality of received videos, the importance of retaining high quality displayed video becomes more evident. In particular, the emergence of new video formats, such as immersive 360°, 8K and Virtual Reality (VR), has pushed more pressure on further bandwidth saving in order to guarantee an acceptable quality. To address this problem, in recent years, besides the great improvements in the domain of transmission network technologies, the development of new video codecs and standards has been initiated. Notably, VVC [99], AoM Video codecs (AV1 and AV2) [100] and Essential Video Coding (EVC) [101] are expected to bring a significant improvement in terms of bitrate saving over existing video coding standards such as HEVC [102].

Although the new video codecs benefit from more efficient algorithms and tools compared to the previous generation standards, reconstructed videos using these codecs still suffer from compression artifacts, especially at low and very low bitrates. The block-based aspect of the hybrid lossy video coding architecture, shared among all these codecs, is the main source of the blockiness artifact in reconstructed videos. To remove this type of artifact, as it was explained in chapter 1, a DBF has been used in most of existing codecs [99, 102, 100, 103]. DBF applies low pass filters in order to smooth out block borders and correct the discontinuous edges across them. Quantization of transform coefficients rather introduces other types of compression artifacts, such as blurriness and ringing. The larger the quantization step gets, the more visible the blurriness and the ringing become. The quantization step is controlled by QP, which varies from 1 to 63 in VVC. In low bitrate video coding, where higher QP values are used, the perceived quality is visibly degraded. SAO and ALF are additional filters that are mainly designed to overcome this problem. SAO categorizes reconstructed pixels into pre-trained classes and associates to them a set of optimized offsets to be transmitted for texture enhancement. ALF, that is applied after DBF and SAO in VVC, further improves reconstructed frames. In

ALF, parameters of a set of low pass filters are optimized at the encoder side and transmitted to the decoder. The common aspect between all these methods is the hand-crafted nature of their algorithms. Although these methods significantly remove undesirable artifacts, the task of enhancing reconstructed videos still has room for further quality improvement.

The promising advances in the domain of machine learning have recently encouraged the broadcast industry to explore it in the video compression domain. Particularly, deep CNNs have attracted more attention owing to their significant performance [104, 105]. Motivated by CNN-based approaches in other image processing tasks, such as Super Resolution (SR) and machine vision, several recent studies have been established in the domain of artifact removal from compressed videos. These approaches are categorized into two main groups: Post Processing (PP) and In-Loop Filtering (ILF). The PP approach improves reconstructed videos after the decoding step and is considered flexible in terms of implementation, as it is not normatively involved in the encoding and decoding processes. In other words, such a PP algorithm serves as an optional step to be used based on the hardware capacity of decoder/receiver device. On the contrary, ILF approach involves the normative aspect of encoding and decoding, by generating high quality reconstructed frames to be served as a reference to other frames in the prediction process. This aspect allows them to offer higher bitrate saving and higher quality, when applied on a smaller sub-set of frames.

In order to reduce artifacts and distortions in reconstructed videos, it is essential to take into account the source and the nature of the artifacts. Most studies have only used reconstructed video and corresponding original video as the ground truth for the training phase of their networks [21, 106]. Except for QP, which has a key influence on the distortion level, the use of other coding information is mostly overlooked in the existing studies. To further improve this aspect, in some more advanced works, coding information such as partitioning, prediction and residual information are also used [30, 32, 42]. However, these approaches are mainly applied to intra coded frames.

This work presents a CNN-based framework for quality enhancement of compressed video. The key element of the proposed method is the use of prediction information in intra and inter coded frames.

To this end, a prediction-aware QE method is proposed and used as the core module of two codecs integration approaches in VVC, corresponding to PP and ILF. In this method, separate models are trained for intra and inter coded frames to isolate the learning of their specific artifacts. The proposed framework emphasizes on the prediction type as critical coding information and offers frame-level as well as block-level granularity for enhancing the quality of reconstructed video pixels. The codec integration of the proposed framework has been carried out in the latest version of the VVC Test Model (VTM-10.0), resulting in coding efficiency gain in different coding configurations (*e.g.* All Intra, Random Access *etc.*). The main contributions

of this work are summarized as follows:

1. Design and implementation of a complete framework for CNN-based quality enhancement based on frame-level and block-level prediction types in VTM-10.
2. Use of normative prediction decisions for training and testing of both intra and inter coding modes. For intra coding mode, proposing an approach to take into account the normative decisions made by the encoder regarding spatial texture modelling via intra prediction modes. Likewise, for inter coding mode, exploiting normative decisions of motion modelling to inform the network about temporally correlated content, possibly with higher quality texture.
3. In inter frames, offering block-level granularity for distinguishing between enhancement task of intra blocks, inter blocks and skip blocks, using a block-type mask and explicit model selection.
4. In-loop implementation of the proposed framework, using a normative frame-level signalling to deactivate CNN-based enhancement in case of quality degradation.
5. Minimizing the memory requirement by sharing the QE models between all three colour components in all QP values.
6. Finally, presenting results of several experiments to analyse the impact of each significant design choice.

In chapter 2, we introduced the related works and categorizes them based on their contribution and relevance to this work. In this chapter, first, section 3.2 presents details of the proposed prediction-aware QE method. This method is then used in Section 3.3 as the core QE module in two codec integration approaches (PP and ILF). Experiment results are presented in Section 3.4, and finally the chapter is concluded in Section 3.5.

3.2 Proposed Quality Enhancement Neural Networks

In this section, fundamental elements of the proposed prediction-aware QE method are described. A common network architecture is adopted that takes into account prediction information associated with reconstructed image. This network is then trained separately for intra and inter images, and applied at the frame-level and block-level to both luma and chroma components, in order to enhance their content based on local coding types.

3.2.1 Prediction-aware QE

Normative decisions made by an encoder are results of extensive searches over possible values of parameters corresponding to its internal coding tools. To make optimal decisions, the spatial and temporal features of video as well as bandwidth constraints are taken into account.

Consequently, the information associated to these coding decisions provide rich and informative representation of signal characteristic. Therefore, in the proposed QE method, we exploit different coding information to help our CNN networks better remove compression artifacts.

QP-map

The quantization step, determined by QP, controls the balance between the level of distortion and the bitrate of a compressed video. Higher QP values apply coarser quantization step on transform coefficients which results in throwing out more high frequency information, hence less bitrate and higher distortion.

In our proposed QE method, we construct a normalized QP-map for each frame and feed it to the network at the same stage as the reconstructed frame. The normalized QP-map ($\mathcal{Q}$) for a frame with the width and height of W and H, respectively, is calculated as:

$$\mathcal{Q}_{i,j} = \frac{q_{i,j}}{q_{max}}, \tag{3.1}$$

where $q_{i,j}$ is the QP value of the block that contains the pixel at coordinates (i, j), with $0 \leq i < W; 0 \leq j < H$, and q_{max} is the maximum QP value (*e.g.* 63 in VVC). In constant QP mode, as in the JEVT CTC [107], the QP-map of a frame would contain a constant value. However, in the CBR mode, this value may change at the block level.

Intra Prediction

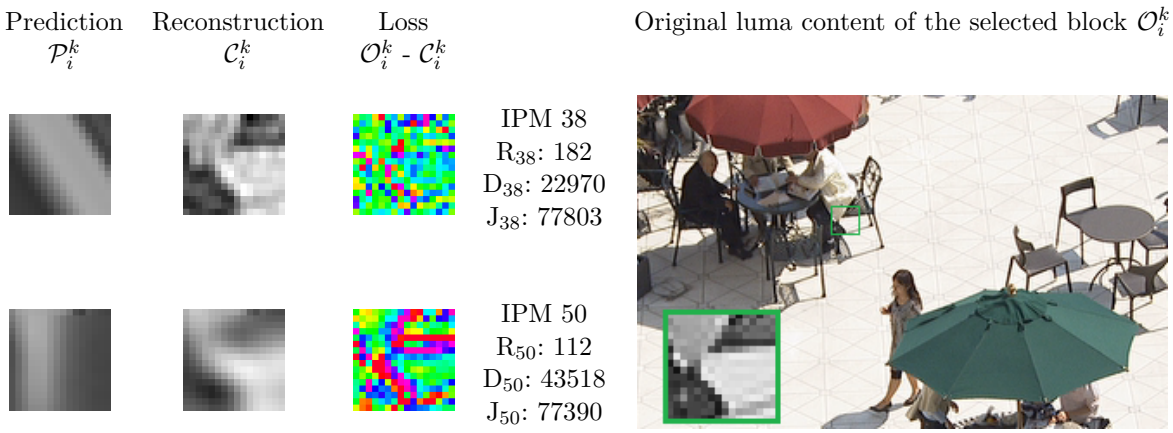


Figure 3.1 – An example of how two IPMs with similar R-D cost can result in different compression artifacts. The tested 16 × 16 block, k, is coded with IPMs 38 and 50 in QP 40 with λ=301.

Intra coding is based on exploiting the spatial redundancies existing in frame textures. In VVC, a set of 67 IPM, representing 65 angular IPMs, plus DC and planar modes are used for modelling texture of blocks. The selection of an IPM for a block is performed by optimizing the rate-distortion (R-D) cost, denoted as J_i:

$$J_i = D_i + \lambda R_i \quad i = 1, \ldots, 67, \tag{3.2}$$

where D_i and R_i are the distortion and the rate of using the i^{th} mode, respectively. The Lagrangian multiplier λ is computed based on the QP which determines the relative importance of the rate and the distortion during the decision making process. In lower bitrates (higher QP values), the value of λ is higher, meaning that minimization of the rate is relatively more important than minimization of the distortion. Similarly, the opposite principle is applicable to higher bitrates (lower QP values).

The best IPM, minimizing the R-D cost of a block, is not necessarily the IPM that represents the block texture most accurately [69]. An example of such a situation is presented in Fig. 3.1. In this figure, a 16×16 block, k, is selected and the prediction ($\mathcal{P}_i^k$) and reconstruction ($\mathcal{C}_i^k$) blocks corresponding to its two best IPMs in terms of R-D cost are shown. Precisely, these two best IPMs are angular modes 38 and 50. As can be seen, despite their similar R-D costs, these two IPMs result in very different reconstructed signals, with different types of compression loss patterns. On one hand, IPM 38 is able to model the block content more accurately (i.e. smaller distortion D_{38}) at the cost of a higher IPM/residual signaling rate (i.e. R_{38}). On the other hand, IPM 50 provides a less accurate texture modeling (i.e. high distortion D_{50}) with a smaller IPM/residual signaling rate (i.e. R_{50}). As a result, these two IPMs result in very different types of artifacts for a given block, as can be seen by comparing the corresponding reconstruction blocks (i.e. $\mathcal{C}_{38}^k$ and $\mathcal{C}_{50}^k$). This behavior is due to two different R-D trade-offs of the selected modes.

The above example proves that the task of QE for a block, frame or an entire sequence could be significantly impacted by different choices of coding modes (e.g. IPM) determined by the encoder. This assumption is the main motivation in our work to use the intra prediction information for the training of the quality enhancement networks.

The example in Fig. 3.1 proves that encoder decisions can have major impact on the QE task and its performance. Particularly, for intra blocks, we assume that the selected IPM for a block carries important information and shall be included in the training of the proposed CNN-based QE method [69]. Therefore, an intra prediction frame is constructed by concatenating the intra prediction signals in the block-level. This signal is then used as input to the network.

Inter Prediction

Inter coding is mainly based on taking advantage of temporal redundancy, existing in consecutive video frames. The prediction signal in the inter mode is a block, similar to the current one, selected from within the range of MVs search, based on a distortion metric. In modern video codecs, we are allowed to search for such similar blocks in multiple reference frames. A motion compensated signal of a given frame, defined as a composition of the most similar blocks to the blocks of the current frame, is used as the prediction information signal in the proposed method. Fig. 3.2 visualizes how the prediction information signal is concatenated from reference frames. In this figure, current frame at time t uses four reference pictures, two from the past ($t-1$, $t-2$) and two from the future ($t+1$, $t+1$).

The temporal prediction signal usually provides additional texture information which is displaced with respect to the texture in the current frame. Hence, there is a potential benefit in using the additional texture information in the temporal prediction for CNN-based quality enhancement [108]. Moreover, in the hierarchical GoP structure of the Random Access (RA) and Low Delay (LD) coding configurations, these references are usually encoded using lower QP values than that of the current frame (Fig. 3.2). Therefore, for some local textures of the current frame, there could occasionally be a version of the texture in a higher quality due to the lower QP of its corresponding frame. The assumption of using the temporal prediction-aware QE method is that feeding the prediction information to the network makes it easier to model the heavily quantized residual signal and retrieve the missing parts in the current frame.

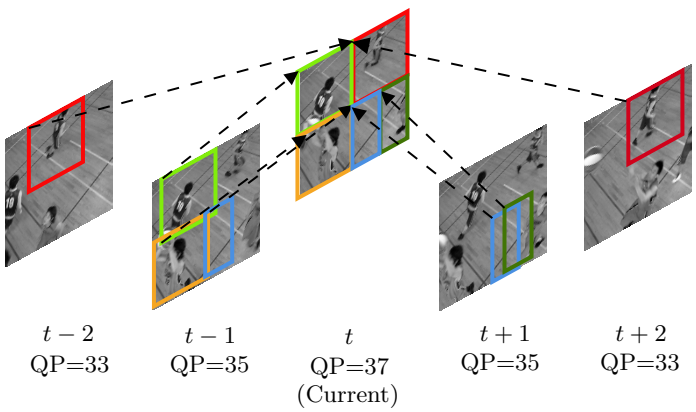


Figure 3.2 – An example of QP cascading in the hierarchical GoP structure, providing higher quality motion compensated blocks at frame t from past ($t-1$ and $t-2$) and future ($t+1$ and $t+2$) frames. Each block in frame t is predicted from at least one reference frame with lower QP (*i.e.* higher quality texture information).

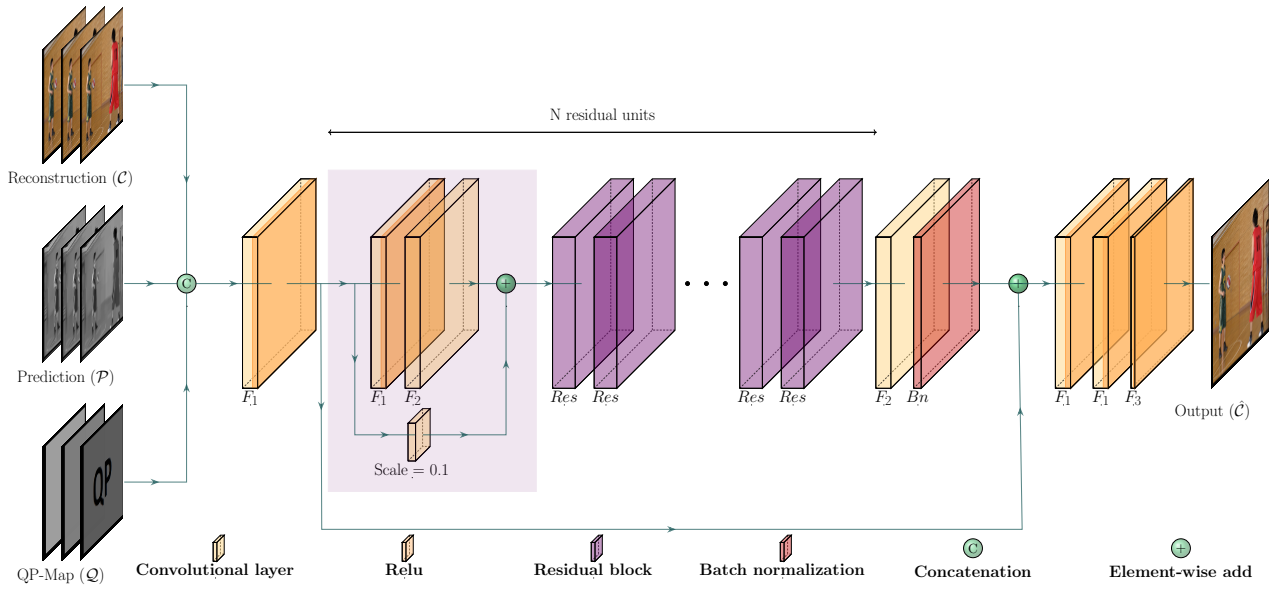


Figure 3.3 – Network architecture of the proposed method using the prediction, QP-map and reconstruction signal as the input.

3.2.2 Network architecture

Recently there have been numerous studies on CNN-based architectures, improving their performance and complexity. In the literature, the residual blocks [1] have been widely used for super resolution and quality enhancement tasks which results in better enhancement and detail retrieval [109, 110]. Inspired by those works, the network architecture of this work is based on residual blocks, combined with CNN layers, as shown in Fig 3.3. The first convolutional layer of the adopted network receives a single reconstructed signal as well as two associated coding information. In particular, a QP map and the prediction signal, both with the same size as the reconstructed signal, are concatenated with the reconstructed image as coding information. After one convolutional layer, N identical residual blocks, each composed of two convolutional layers and one ReLU activation layer in between, are used. The convolutional layers in the residual blocks have the same size as the feature maps and kernel size of the first convolutional layer. In order to normalize the feature maps, a convolutional layer with batch normalization is applied after the residual blocks. A skip connection between the input of the first and the last residual block is then used. Finally, three more convolutional layers after the residual blocks are used for reconstructing the enhanced reconstructed signal. Worthy of mention, the input size is arbitrary and not limited to one full frame. As will be explained later, depending on the granularity of the QE task, the inference might be applied in the frame-level or block-level.

1. In the remaining of this chapter, the term "residual" is occasionally used interchangeably in the context of video compression residual signal as well as neural network residual layer.

Table 3.1 – Summary of the three models trained for different coding types.

Name	Inputs			Frames type
	Reconstruction	Quantization	Prediction	
M_{cqp}^{intra}	✓	✓	✓	Intra
M_{cq}^{inter}	✓	✓	✗	Inter
M_{cqp}^{inter}	✓	✓	✓	Inter

Given $\mathcal{I}$ as the concatenation of the input signals, the process of producing the enhanced reconstructed signal $\hat{\mathcal{C}}$, by the proposed CNN-based QE method is summarized as:

$$\hat{\mathcal{C}} = F_3^1(F_1^2(Bn^1(F_2^1(Res^N(F_1^1(\mathcal{I})))) + F_1^1(\mathcal{I}))), \tag{3.3}$$

where $F_1(.)$ and $F_2(.)$ are $3 \times 3 \times 256$ convolutional layers, with and without the ReLU activation layer, respectively. Moreover, $F_3(.)$ is a $3 \times 3 \times 1$ convolutional layer with the ReLU activation layer. The superscript of each function indicates the number of times they are repeated sequentially in the network architecture. Finally, Res and Bn are the residual block and batch normalization layer, respectively.

Based on the above network architecture and the use of the prediction information, three models are trained with different sets of inputs. Each model is trained using frames that are encoded in its given coding mode and also used for the inference of the same coding mode. In the first two models, denoted as M_{intra}^{cqp} and M_{inter}^{cqp}, the input is the concatenation of the decoded image $\mathcal{C}$, the QP-map $\mathcal{Q}$ and the prediction signal $\mathcal{P}$:

$$\mathcal{I}_{cqp}^{m} = \mathcal{C}^m \oplus \mathcal{Q} \oplus \mathcal{P}^m, \tag{3.4}$$

where $\oplus$ is the concatenation operator and m is the coding mode which can be *intra* or *inter*. The other model, denoted as M_{inter}^{cq}, do not use the prediction signal as input:

$$\mathcal{I}_{cq}^{m} = \mathcal{C}^m \oplus \mathcal{Q}, \tag{3.5}$$

The normalized QP-map ($\mathcal{Q}$) for a frame (or a block) with the width and height of W and H, respectively, is calculated as:

$$\mathcal{Q}_{i,j} = \frac{q_{i,j}}{q_{max}}, \tag{3.6}$$

where $q_{i,j}$ is the QP value of the block that contains the pixel at coordinates (i, j), with $0 \leq i < W; 0 \leq j < H$, and q_{max} is the maximum QP value (*e.g.* 63 in VVC).

Table 3.1 summarizes the details of three proposed models. Regardless of the QE method, in all three models, the QE task can be formulated as:

$$\hat{\mathcal{C}} = f_{QE}(\mathcal{I}; \theta_{QE}), \tag{3.7}$$

where θ_{QE} is the set of parameters in the network architecture of Eq. (3.3). This parameter set is optimized in the training phase, using the L_1 norm as the loss function, computed with respect to the original signal $\mathcal{O}$:

$$L_1(\mathcal{O}, \hat{\mathcal{C}}) = |\mathcal{O} - \hat{\mathcal{C}}|. \tag{3.8}$$

3.2.3 Implicit model selection

Frame-level

In the two previous sections, we explained how the use of prediction information could improve the performance of the QE task. The two trained networks for intra and inter are applied differently at the frame-level. In intra frames, since all blocks have the same coding mode and prediction type, the whole frame is enhanced using the intra trained network (M^{cqp}_{intra}).

However, in inter frames, not all blocks have the same type of prediction. Depending on local texture and motion characteristics, the encoder has the choice between different types of predictions. More precisely, three main prediction types can be found in blocks of an inter coded frame: inter, intra and skip. Fig. 3.4 shows an example of different block types within an inter coded frame. As a result, the prediction-aware quality enhancement of inter frames is performed in the block-level.

Block-level

The choice of the coding type of blocks in an inter coded frame depends on motion and texture characteristics. The regular inter mode, where the motion information along with residual signal is transmitted, is usually the more common type in inter coded frames. However, when the local content becomes too simple or too complex to compress, the skip mode and the intra mode might be used instead, respectively. More precisely, when a part of the video is static or has homogeneous linear motion, reference frames usually have very similar co-located blocks. In this case, skip mode is useful, where the motion is derived from neighbouring blocks and residual transmission is skipped. On contrary, due to fast motion or occlusion, sometimes no similar block can be found in reference frames. In this case, intra coding can offer a better prediction signal based on the spatial correlation of texture.

In the proposed QE scheme, blocks within inter coded frames are enhanced based on their coding type. To do so, a block-type mask is formed using the type information extracted from the bitstream. This mask is then used to determine the proper QE model for each block. Precisely, intra blocks and inter blocks are enhanced with intra-trained (M^{cqp}_{intra}) and inter-trained models

(M_{inter}^{cqp}), respectively. On contrary, skip blocks are enhanced using the prediction-unaware model (M_{inter}^{cq}), which is trained without any prediction information. When skip mode is signalled for a block, the content of prediction signal for that block is identical to the reconstructed block. As a result, the network which is trained with inter prediction signal and has learnt the motion in the video, is unsuitable for its enhancement. Our experiments showed that if identical prediction and reconstruction signals of skip blocks are fed to the proposed prediction-aware QE network, the performance will degrade, compared to the enhancement with the prediction-unaware method.

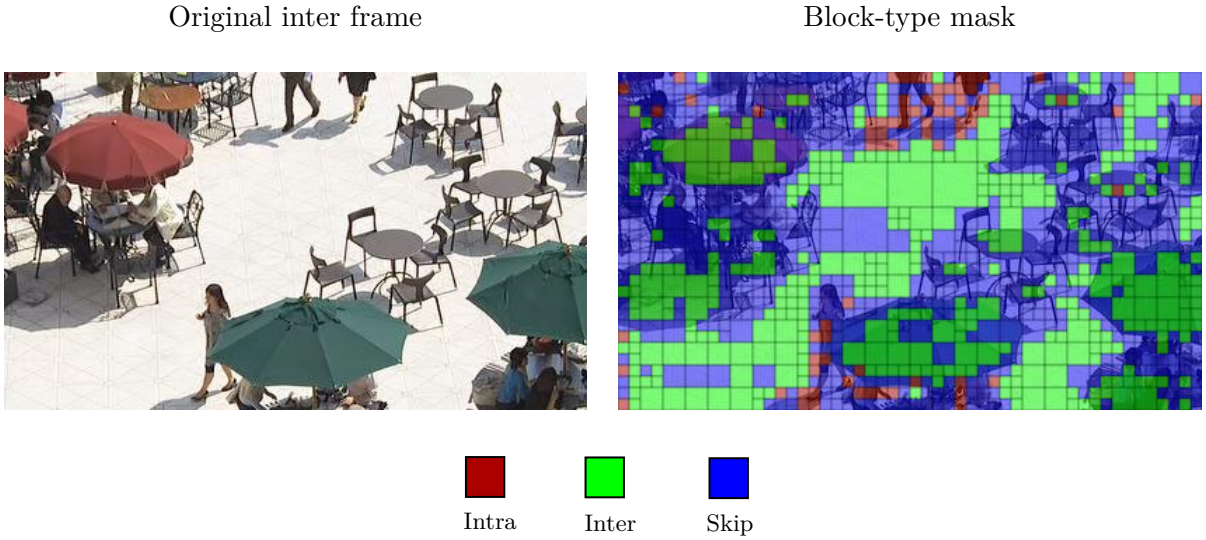


Figure 3.4 – Block type mask of an inter frame from the BQSquare sequence, with the three block types present.

The implementation of the block-type mask can be performed at two levels: frame-level and block-level. In the frame-level application of the block-type mask, each inter frame is enhanced three times, using the trained networks for intra, inter and skip (prediction-unaware) coding types. Then, using the block-type mask of the frame, the three outputs are combined and one enhanced frame is produced. However, in the block-level application of the mask, the CNN-based QE is applied in the block-level, where each block is enhanced only once by using its appropriate model. Then all enhanced blocks are concatenated to form the final enhanced frame. Our experiments show that these two implementations have a negligible difference in terms of performance. Therefore, we chose to use the block-level approach, since it is significantly less complex in terms of the number of operations than the frame-level implementation.

3.2.4 Explicit model selection

As discussed, based on frame type, the prediction signal used for blocks of a frame can be different. In intra frames, all blocks are encoded using the intra coding mode. However, in inter

frames, blocks can be either inter coded or intra coded, depending on the local motion and texture complexity. Moreover, in all frame types, there are often blocks whose residual signal is zero, which makes the prediction signal identical to the reconstructed block. As a result, different types of artifacts can be found in the same encoded frame, that might need different networks for enhancement. Using the three models presented in previous section, a MS strategy is proposed in two levels: frame and Coding Tree Block (CTB).

At the CTB level, each CTB is enhanced by all three models at encoder side. For a given CTB, with the original content $\mathcal{O}$, the MS at the CTB level is performed by minimizing the MSE as:

$$M_{in^*}^{m^*} : (in^*, m^*) = \underset{m,in}{\operatorname{argmin}} \ \mathrm{MSE}(\hat{\mathcal{C}}_{in}^{m}, \mathcal{O}), \tag{3.9}$$

where $\hat{\mathcal{C}}_{in}^{m}$ is the enhanced signal using the model M_{in}^{m}. Moreover, the distortion of the decoded signal with no enhancement is also calculated. At this point, four distortion values are calculated – three corresponding to the enhanced signals and one corresponding to the decoded signal. At the end, the setting that gives the lowest distortion among the above four settings is selected.

In order to inform the decoder about the enhancement setting chosen by encoder, the corresponding information should be transferred in the bitstream. To this end, a signaling scheme is implemented at both CTB and frame levels. The frame level signaling is performed with a flag f_1 to indicate whether or not the CTB level signaling is used. In the case that this flag is zero, decoder will use a default model, either M_{cqp}^{intra} or M_{cqp}^{inter}, depending on the frame type. Otherwise, the selected model is determined in the CTB level, using two flags f_2 and f_3. More precisely, four bits are transmittable with two f_2 and f_3 flags. In table, possible combination of f_2 and f_3 flags and their interpretations are shown. Moreover, the encoder side decision for the MS is presented in Algorithm 1. The inputs to this algorithm are the rate and distortion of the encoded frame, enhanced by the default model, denoted as R^{def} and D^{def}, respectively.

$f_2 f_3$	Interpretation
00	No enhancement
01	Enhancement with M_{cqp}^{intra}
10	Enhancement with M_{cqp}^{inter}
11	Enhancement with M_{cq}^{inter}

Table 3.2 – CTB level signalling interpretation

Algorithm 1 Frame level of MS

input: R^{def}, D^{def} as rate and dist. of frame, respectively.
$R_{f_1:0} \leftarrow R^{def} + 1$
$R_{f_1:1} \leftarrow R^{def} + 1 + 2 \times$ (number of CTBs in frame)
$D_{f_1:0} \leftarrow D^{def}$
$D_{f_1:1} \leftarrow 0$
for each CTB $u \in$ frame **do**
  Get $M_{in^*}^{m^*}$ for u using Eq. (3.9)
  Set f_2 and f_3 based on $M_{in^*}^{m^*}$or decoded distortion based on table 3.2
  Enhance u with $M_{in^*}^{m^*}$ using Eq. (3.7) if needed
  Compute D_u as the distortion of u after enhancement
  $D_{f_1:1} \leftarrow D_u$
end for
$J_{f_1:1} \leftarrow D_{f_1:1} + \lambda R_{f_1:1}$
$J_{f_1:0} \leftarrow D_{f_1:0} + \lambda R_{f_1:0}$
if $J_{f_1:1} < J_{f_1:0}$ **then**
  $f_1 \leftarrow 1$
else
  $f_1 \leftarrow 0$
end if

3.3 Codec integration

The proposed QE method is integrated in the VVC codec with two different approaches: Post Processing (PP) and In-Loop Filtering (ILF). While their core QE modules share the same principles, described in the previous section, they possess unique characteristics and impose different challenges.

Figure . 3.5 shows where each codec integration method is placed and how it impacts the end-to-end system. In this figure, the green and blue modules represent the ILF and PP approaches, respectively, where only one of them can be activated in an end-to-end system. Also, removing both of them results in the reference system where no CNN-based QE is integrated. As can be seen, a common prediction-aware network is shared between the ILF and PP approaches. However, it is used differently, which will be explained later in this section.

3.3.1 QE as Post Processing

QE as PP module is placed after decoding the bitstream and before displaying the reconstructed image. Therefore, it is applicable only on the decoder side. From another point of view, pixel modifications of an image impact only the quality of that image with no temporal propagation.

In this approach, the encoder side is not aware of the fact that the displayed image will go

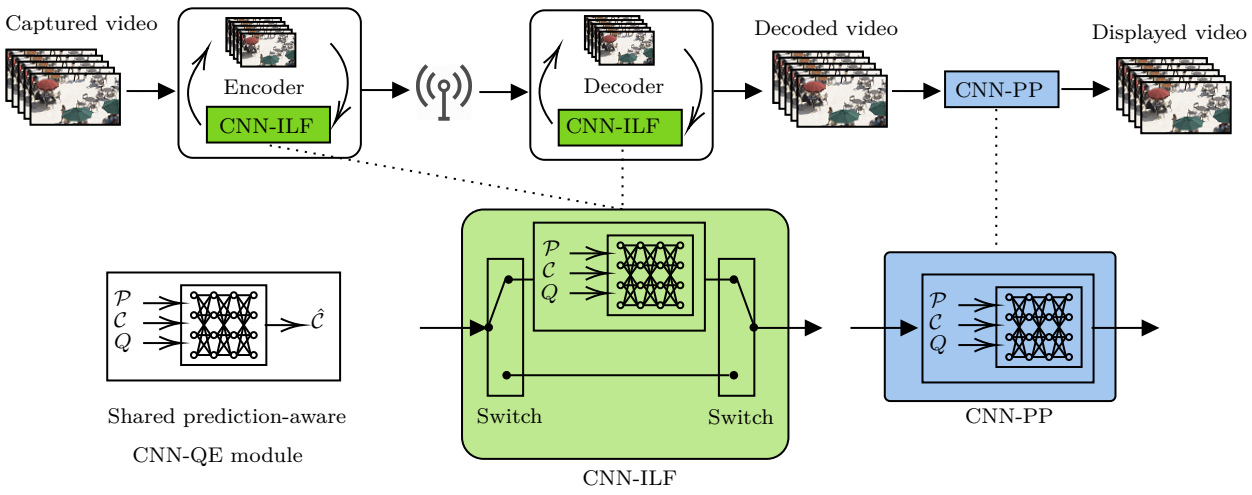


Figure 3.5 – The proposed prediction-aware framework with two codec integration approaches: ILF (green) and PP (blue), sharing the same CNN-based QE module.

through a QE step. Therefore, no complexity is added to the encoder and the normative aspect of the generated bitstream remains unchanged.

At the decoder side, the PP step is considered as optional. Usually, this choice depends on the processing capacity of the display device. For instance, if the device is equipped with dedicated Graphic Processing Unit (GPU) or other neural network inference hardware, then the post-processing can be applied and bring quality improvement at no bitrate cost. Another advantage of the PP approach is that it can be applied on already encoded videos without needing for their re-compression.

Activating the blue box in Figure . 3.5 represents the scheme of the PP codec integration. As the QE module requires the necessary coding information, namely the QP map, the prediction signal and the coding type mask, which are extracted from bitstream during the decoding phase.

Moreover, the explicit model selection strategy, has only been employed as PP. Figure . 3.6 demonstrates the overall workflow of proposed QE with explicit MS.

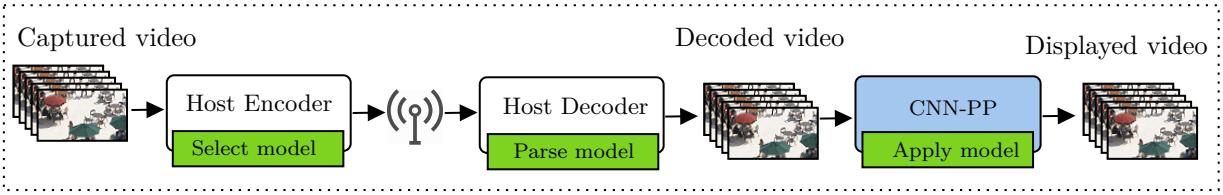


Figure 3.6 – The workflow scheme of the proposed CNN-based PP with explicit MS

3.3.2 QE as In-Loop Filter (ILF)

The main idea of an ILF is based on the propagation of improvements. More precisely, an ILF locally improves pixels of the current frame and then temporally propagates the improvement through frames which use the current frame as their reference.

QE as ILF module is placed after existing in-loop filters in VVC (*i.e.* DBF, SAO and ALF). Since the framework is shared with the PP approach, same coding information is required which is accessed during the encoding process of a frame.

Unlike the PP approach and similar to existing VVC in-loop filters, the ILF approach is normative. In other words, if activated, both encoder and decoder are forced to apply it on their reconstructed samples. Therefore, one main difference of ILF compared to the PP is the mandatory complexity at both encoder and decoder sides.

On contrary, ILF approaches have an interesting advantage of propagating the quality enhancement through the frames. Figure . 3.7 visualizes this aspect. In this figure, the propagation of quality enhancement in a GoP of size 8, with four temporal layers (Tid_i, i=0,1,2,3) is shown, where only the intra frames at POC0 and POC8 are enhanced. The offsets $\{+1, \ldots, +4\}$ approximately represents how far a frame is placed from the enhanced frames. Moreover, the spectrum of greens indicates the benefit of each frame from the enhancement propagation, based on their distance order from the enhanced intra frames. Therefore, one can see that the propagation benefit gradually diminishes as the frame gets further from the enhanced frames.

Multiple-enhancement

The potential downside of enhancement propagation is a phenomenon called multiple enhancement in this work. In common GoP structures with inter frames, the effect of processing one frame usually propagates through other frames that refer to it in the motion compensation. In particular, by applying in-loop quality enhancement, either CNN-based or standard methods (e.g. SAO, ALF, DBF, etc.), when the quality of a frame in lower temporal layers is enhanced (Figure . 3.7), the effective enhancement will also impact frames in higher temporal layers. For instance, a reconstructed inter frame, may contain blocks from its reference frames which are already enhanced by the applied method.

To better understand the multiple-enhancement effect, imagine a simplified low-delay GoP structure of length 2, with one intra frame and one inter frame (P-frame), as shown in Figure . 3.8. The blue and green dashed lines from the inter frame to the intra frame represent the reference frame used for its motion compensation, without and with a CNN-based ILF module, respectively. The ILF module is represented with a simplified version of Eq. (3.7), where the prediction signal and the QP map are not used in the enhancement, hence, $\mathcal{I} = \mathcal{C}$. Moreover,

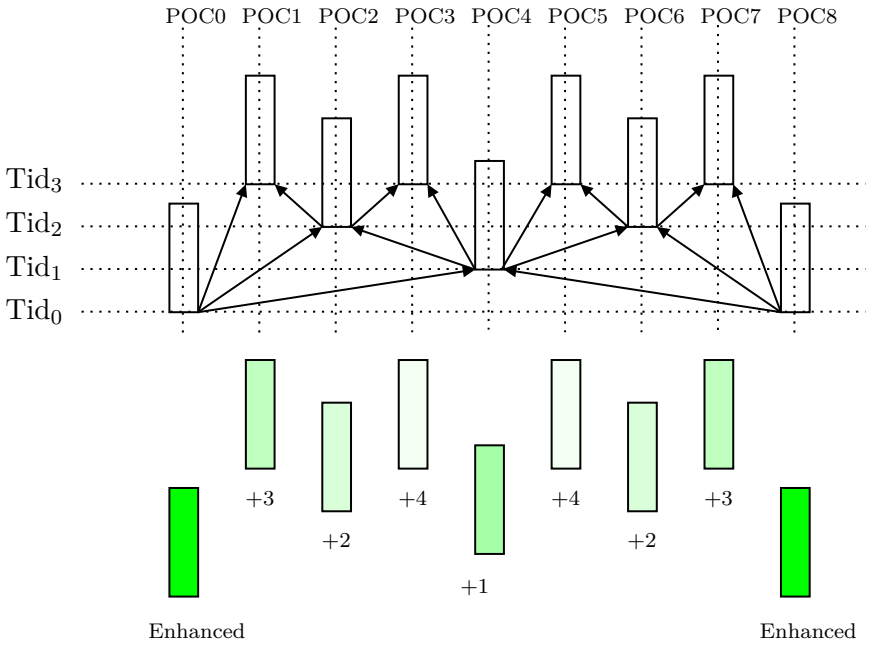


Figure 3.7 – Propagation of quality enhancement in ILF approach in GoP of size 8. The spectrum of greens approximately shows the benefit of each frame from the enhancement propagation, based on the distance order from the enhanced intra frames, which is approximated based on the number of steps required to reach frames from both enhanced frames.

the reconstructed signal $\mathcal{C}$ is composed as:

$$\mathcal{C} = \mathcal{P} + \hat{\mathcal{R}}, \tag{3.10}$$

where $\mathcal{P}$ and $\hat{\mathcal{R}}$ are the prediction signal and reconstructed residual signal (i.e. after quantization and de-quantization), respectively. Accordingly, the enhanced inter frames in Figure . 3.7 can be expressed as:

$$\hat{\mathcal{C}}_2 = f_{QE}(\mathcal{C}_2; \theta) = f_{QE}(\mathcal{P}_2 + \hat{\mathcal{R}}_2; \theta). \tag{3.11}$$

As often happens in content with no or limited linear motion, the residual transmission of an inter block could be skipped (i.e. the skip mode):

$$\hat{\mathcal{R}}_2 = 0. \tag{3.12}$$

In such circumstances, the reconstructed inter frame associated to a skip block can locally be expressed as:

$$\mathcal{C}_2 = \mathcal{P}_2. \tag{3.13}$$

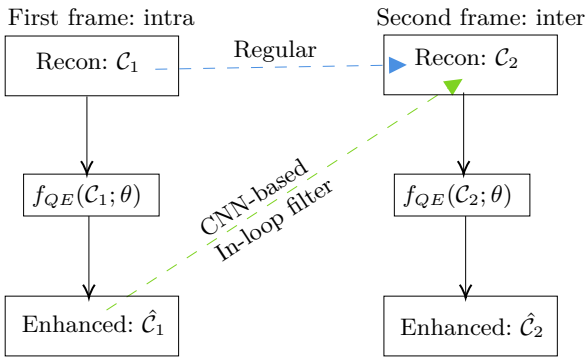


Figure 3.8 – Multiple enhancement example in a simplified GoP of size 2 (one intra frame at left and one inter frame at right). The dashed lines show the use of reference picture for the inter frame, with (green) and without (blue) a CNN-based in-loop filter.

Since the prediction signal of the skip block is motion compensated from its enhanced reference, we also have:

$$\mathcal{P}_2 = \hat{\mathcal{C}}_1, \tag{3.14}$$

which according to Eq. (3.11), it would result in multiple enhancement of the inter frame:

$$\hat{\mathcal{C}}_2 = f_{QE}(\hat{\mathcal{C}}_1;\theta) = f_{QE}(f_{QE}(\mathcal{C}_1;\theta);\theta) \tag{3.15}$$

The multiple-enhancement effect is not an issue by nature. For instance, standard ILF also deal with a similar situation, where the reference frame has gone through the same enhancement process, and this multiple enhancement effect seems not to impact their performance.

However, in CNN-based QE methods, the tuning process is automatic through a complex offline training process. This aspect makes the multiple-enhancement effect a potential hazard for CNN-based ILF algorithms. More precisely, one of the main challenges is that a CNN-based QE network which is trained for the PP task, would not perform well for the ILF task, since it has not observed enhanced references during the PP training. In other words, such network has observed frames whose references were not enhanced by any CNN-based QE. While, during the ILF inference, this network will have to deal with reconstructed frames whose reference have also been enhanced by a CNN-based QE. As shown in previous studies, the multiple enhancement effect negatively impacts CNN-based ILF methods.

End-to-end training solution

One solution to the multiple-enhancement issue is to avoid the mismatch between the training set and test set. This solution, called the end-to-end training in the literature [21], guarantees that frames whose references have been through CNN-based QE are present in the training set. However, since it is important that the references of such frames are also enhanced by the same

CNN-based QE, the end-to-end training solution has a potential chicken-and-egg problem.

One way to overcome the above problem is to run the dataset generation step and the training step in multiple iterations. Starting from the first temporal layer, in each iteration, one temporal layer is used for training and a network is trained for it. Then, all frames in that layer are enhanced in the dataset generation step, to be used in the training step of the next temporal layer. This solution is extremely time-consuming, therefore not practical for the current problem.

Adaptive ILF method

A greedy approach to avoid the multiple-enhancement is to determine at the encoder side whether or not a frame should be enhanced. For this purpose, an adaptive ILF mechanism for inter frames is used in the proposed method. The main idea is to enhance only frames where applying the CNN-based QE filter results in increasing the quality. More precisely, each reconstructed frame is processed by the proposed CNN-based QE method at the encoder side. Then, using the original frame as reference, an MSE comparison is performed between the unprocessed reconstructed frame (before enhancement) and the processed reconstructed frame (after enhancement). The switch in the adaptive ILF is then set based on the smaller MSE value.

The adaptive ILF solution requires an encoder-side signalling. Since in the proposed method, the encoder decides about the switch flag at the frame-level, the signalling is performed in the Picture Parameter Set (PPS). Signaling in the frame-level adds only one bit per frame, therefore, its impact on the coding efficiency is negligible. However, alternative implementations might apply the switch in finer granularity, such as CTU-level or even CU-level. This latter aspect is left as future work.

3.4 Experimental Results

3.4.1 Experimental setup

Dataset

The training phase has been carried out under the recent Deep Neural Network Video Coding (DNNVC) CTCs, released by JVET [107, 111]. The recommended dataset in these CTCs is BVI-DVC [112], which consists of 800 videos of 10-bit pixel representation, in different resolutions covering formats from CIF to 4K. We also used two image databases, namely DIV2K and Flickr2K for the training of intra-based networks. These datasets are composed of 900 and 2650 high quality images, respectively. The videos and images in the training dataset were converted to 10-bit YCbCr 4:2:0 and only the luma component has been used for training.

To train the models for inter frames, the native RA configuration of VTM10.0 reference software was used with input and internal pixel depth of 10-bit. Moreover, all in-loop filters

were kept activated. The video dataset was encoded in five base QPs, {22, 27, 32, 37, 42}. For each QP, four out of 64 frames of each reconstructed video were randomly selected. Finally, a total of 3200 reconstructed frames obtained for each QP base, resulting in 16000 frames for all five QPs. The equivalent ground truth and prediction signal, as well as QP map for these reconstructed frames, were also extracted.

The networks for intra frames were trained separately by encoding the DIV2K and Flickr2K datasets in the AI configuration. In total, 3550 images were generated, from which we randomly selected 1200 for each QP, resulting in 5800 images. Moreover, we added the intra frames of the RA dataset to the AI dataset. To sum up, a total of 7400 intra frames were used. Finally, a patch-based strategy was employed which will be explained in Section 3.4.1.

For the test phase, nineteen sequences from the JVET CTCs (classes A1, A2, B, C, D and E) were used [113]. It is important to note that none of these sequences were included in the training dataset. The test sequences were finally encoded in RA and AI configurations, using the same encoder settings as for the training.

Training Settings

The networks were implemented in PyTorch platform and the training was performed on NVIDIA GeForce GTX 1080Ti GPU. The parameter N (number of residual blocks of the network) was set to 16. All networks were trained offline before encoding. The initial learning rate was set to 10^{-5} with a decay of 0.5 for every 100 epochs. The Adam optimizer [114] was used for back propagation during the training and each network was trained for 500 epochs. The validation dataset was extracted from the training dataset and was composed of 50 cropped reconstructed frames and their corresponding prediction and original frames. During the training, the best network parameters were chosen based on the evaluation performed on the evaluation dataset.

The training has been performed on 64×64 patches, randomly chosen from the training dataset. These patches are fed to the network on batches with a size of 16. Block rotation and flip were also applied randomly to selected patches to achieve data augmentation.

It is important to note that one single model is shared between the three colour components (Y, U and V) in all QP values. Given that the proposed method requires different models to apply the block-level coding type mask (see Section 3.2.3), the following models should be stored at the encoder and decoder sides:

— Intra-trained model, for enhancing intra frames as well as intra blocks in inter frames (M_{cqp}^{intra}).

— Inter-trained model, for enhancing inter blocks in inter frames (M_{cqp}^{inter}).

— Prediction-unaware model, for enhancing skip blocks in inter frames (M_{cq}^{inter}).

Evaluation metrics

The main performance metric used for comparison is the BD-BR [9]. This metric is formally interpreted as the amount of bitrate saving in the same level of PSNR and VMAF based quality [115]. Based on this metric, the performance of the VVC reference software VTM-10.0 integrated with different configurations of the proposed CNN-based QE method is presented. For this purpose, the VTM-10.0 with no modification is used as the anchor. The BD-BR saving is calculated in two ranges of QPs, QP $\in \{22, 27, 32, 37\}$ and QP $\in \{27, 32, 37, 42\}$ naming CTCs QP range and high QP range, respectively.

The outputs of the tested methods are also compared in terms of PSNR. To do so, this metric is computed for each tested CNN-based method and is noted as PSNR_{Prop}. Likewise, the metric is computed for the output of the reference anchor VTM-10, noted as PSNR_{VTM}. The original input signal before compression is used for computation of both PSNR values. The average difference between the two PSNR values, on a given set of sequences S and a set of QP values Q, is measured as the ΔPSNR and is computed as:

$$\Delta\text{PSNR} = \frac{\sum_{s\in S}\sum_{q\in Q}\left(\text{PSNR}^{s,q}_{Prop} - \text{PSNR}^{s,q}_{VTM}\right)}{|S||Q|}, \tag{3.16}$$

where $|S|$ and $|Q|$ denote the number of sequences and QP values tested, respectively. Positive values of above equation indicate compression gain. In our experiments, the presented ΔPSNR values are averaged over four QP values of CTCs.

Finally, the relative complexity of tested methods is computed as the ratio of the Run-Time (RT) with respect to the reference. This metric, which is applicable to both encoder and decoder sides, is computed as:

$$RT = \frac{1}{|S||Q|}\sum_{s\in S}\sum_{q\in Q}\frac{\text{RT}^{s,q}_{Prop}}{\text{RT}^{s,q}_{VTM}} \tag{3.17}$$

3.4.2 Ablation Study

In this section, the impact of the following elements of the proposed method are analysed: QP map training, network depth, prediction-awareness, and model selection strategy. It should be noted that for this ablation study, results and discussions are limited only to the Post Processing (PP) integration of the proposed method. This way, different elements of the proposed method can be evaluated without entering into the complexity of multiple-enhancement and temporal aspect of the In-Loop Filtering (ILF) integration.

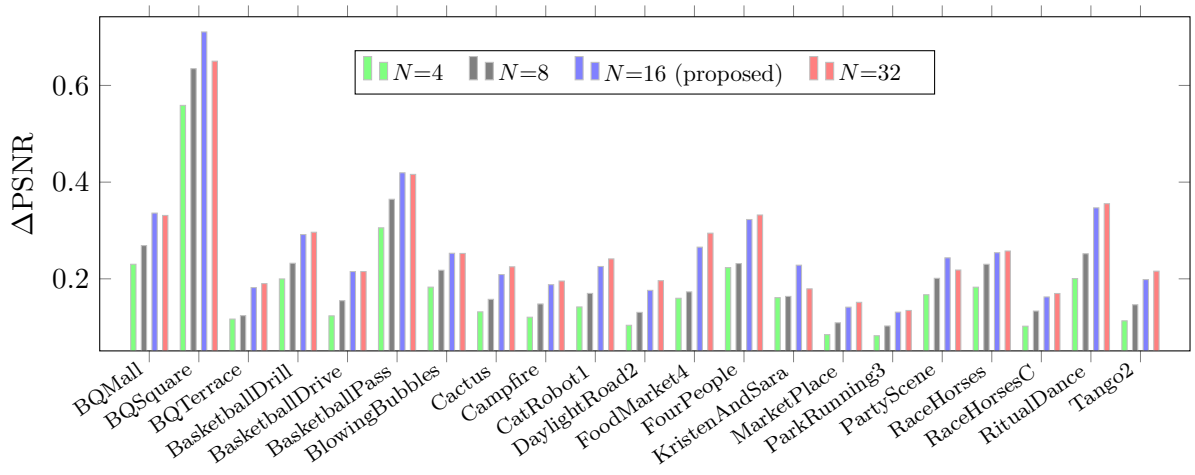


Figure 3.9 – Impact of the number of residual layers (N) on the performance of the network in terms of ΔPSNR. The reference for this test is VTM10, hence positive ΔPSNR values indicate higher quality enhancement. All test are carried out in the RA mode.

Network Architecture

Generally, more complex network architectures have a larger capacity of learning complicated tasks, such as coding artifacts. In the first ablation study, we modify the network architecture, presented in Fig. 3.3, in terms of the number of residual blocks N. More precisely, instead of using N=16, we try alternative values 4, 8 and 32. Fig. 3.9 shows that decreasing the parameter N impacts the QE performance (RA configuration) in terms of ΔPSNR. As can be seen, the global trend is a decrease in the performance. However, the difference between N=16 and N=32 is negligible. Therefore, in this work we chose N=16 for the network architecture.

QP-map training

QP has been used as an input to all QE networks presented in this work. However, a noticeable number of studies in the literature take the QP-specific training approach, assuming that several trained networks can be stored at the encoder and/or decoder side.

A set of ablation studies have been conducted to understand differences between the two approaches in the All Intra (AI) coding mode. To this end, in addition to performance comparison of the two approaches, the potential damage due to the use of the incorrectly trained network in the QP-specific approach is also studied. More precisely, each network which has been trained on a particular QP was used for the QE task of other QP values.

The result is presented in Fig. 3.10, where the average ΔPSNR with respect to VTM-10 is used as metric (computed on all CTCs sequences). In this figure, the proposed method based on

the QP-map (shown in green) is compared to five other configurations. Four of them, expressed as q_L^i with i =22, 27, 32 and 32, are the configurations where one single model trained on the QP value i is used for enhancement of all other QPs. As the fifth one, each of above models are used for their exact QP value, which results in the QP-specific methods in the literature (shown with a dashed black curve).

As can be seen, the use of QP-specific training approach is slightly better than the QP-map approach. However, the cost of storing several models makes it less useful from the implementation point of view. Especially, this additional cost is problematic at the decoder side which is supposed to be implemented various devices with different range of capacities, including mobile devices with considerably limited hardware resources. Moreover, it can be seen that the QP-map training approach has a robust performance when applied on different QP values. On the contrary, the models that are trained for a particular QP usually have a poor performance on any other QP value, therefore, sharing them for a range of QP values can also damage the performance.

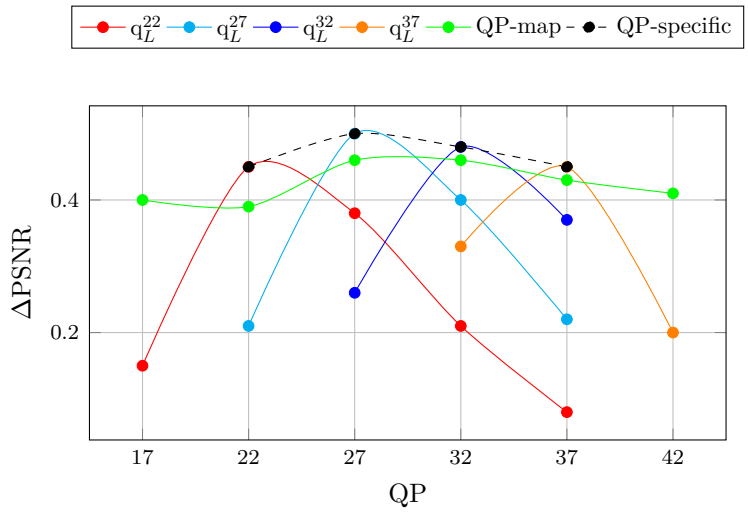


Figure 3.10 – PSNR performance of the QP-specific and QP-map training. The reference for the ΔPSNR computation is VTM-10 and all test are carried out in the AI mode.

Prediction-awareness

As the main contribution, the prediction-awareness of the proposed method is evaluated in Table 3.3 for CTC QPs and in Table 3.4 for high QPs, in terms of BD-BR gain compared to VTM-10. For these tables, the proposed method is integrated as the PP module in the RA coding configuration. This means that both intra and inter frames have been enhanced using their dedicated networks (implicit MS). Moreover, the BD-BR metric is measured in two ranges of CTCs and high QPs. Finally, two configurations of the proposed QE framework are evaluated on

luma and chroma components: prediction-unaware (described in Eq. (3.5)) and prediction-aware (described in Eq. (3.4)). The benchmark for the BD-BR comparison is the prediction-unaware version of the proposed CNN-based QE method.

As can be seen, the proposed prediction-aware algorithm consistently outperforms the prediction-unaware one, in both QP ranges. In the CTCs QP range, the coding gains (BD-BR(PSNR)) of prediction-awareness on Y, U and V are -7.31% -8.90% -11.22%, respectively. Compared to the prediction-unaware setting with coding gains of -5.79%, -8.11%, -9.53%, it can be noticed that adding prediction-awareness brings -1.52%, -0.79% and -1.69% more bitrate savings in the three components, respectively. Moreover, the VMAF-based BD-BR results show even higher performance improvement when prediction information is used. Precisely, the gains for prediction-aware and prediction-unaware setting are -9.2% and -5.5%, respectively, which indicates -3.7% more bitrate saving in terms of VMAF-based BD-BR.

Likewise, a consistent luma BD-BR gain of 1.03% by the prediction-aware compared to the prediction-unaware method can be observed in the high QP range (table 3.4). The relative gain in the high QP range is about 0.5% smaller than in the CTCs range. This could be explained by the fact that the absolute gains of both prediction-unaware and prediction-aware methods are larger in this QP range than in the CTCs range, possibly causing a saturation of performance gain.

Table 3.3 – BD-BR metric for performance comparison of the proposed CNN-based QE method as Post-Processing in the RA coding configuration on top of VTM-10.0 (CTCs QP).

Class	Sequence	Prediction-unaware				Prediction-aware			
		BD-BR(PSNR)			BD-BR	BD-BR(PSNR)			BD-BR
		Y	U	V	(VMAF)	Y	U	V	(VMAF)
A1	Tango	-5.6%	-14.4%	-12.5%	-6.4%	-8.5%	-15.7%	-14.5%	-11.7%
	FoodMarket	-3.6%	-11.4%	-8.4%	-11.9%	-7.2%	-11.5%	-10.5%	-15.5%
	CampFire	-4.3%	-4.2%	-10.8%	-9.9%	-5.9%	-6.9%	-14.2%	-14.2%
	Average	-4.5%	-10.0%	-10.6%	-9.4%	-7.2%	-11.4%	-13.0%	-13.8%
A2	CatRobot	-6.9%	-13.3%	-12.8%	-5.4%	-8.5%	-13.6%	-13.5%	-11.1%
	Daylight	-9.3%	-11.3%	-6.1%	-9.4%	-10.8%	-11.5%	-8.6%	-15.4%
	ParkRunning	-3.0%	-2.6%	-3.5%	-1.4%	-4.1%	-2.4%	-4.2%	-5.5%
	Average	-6.4%	-9.1%	-7.5%	-5.4%	-7.8%	-9.2%	-8.8%	-10.7%
B	MarketPlace	-4.8%	-7.2%	-8.5%	-5.0%	-5.6%	-8.0%	-9.8%	-7.5%
	RitualDance	-6.0%	-9.0%	-11.3%	-8.5%	-8.0%	-10.6%	-13.4%	-11.7%
	Cactus	-4.2%	-6.3%	-8.5%	-6.3%	-6.0%	-6.3%	-10.2%	-8.6%
	BasketballDrive	-5.4%	-6.8%	-13.4%	-3.9%	-7.0%	-10.1%	-15.3%	-7.3%
	BQTerrace	-4.9%	-11.8%	-9.9%	1.3%	-5.9%	-13.3%	-12.8%	-2.9%
	Average	-5.1%	-8.2%	-10.3%	-4.5%	-6.5%	-9.7%	-12.3%	-7.6%
C	BasketballDrill	-6.5%	-12.0%	-15.3%	-6.4%	-8.3%	-12.3%	-16.3%	-9.4%
	BQMall	-5.2%	-5.2%	-6.4%	-4.9%	-6.7%	-5.5%	-7.4%	-8.7%
	PartyScene	-5.3%	-4.7%	-7.3%	-3.9%	-6.1%	-4.8%	-8.2%	-7.5%
	RaceHorses	-2.8%	-8.4%	-8.6%	-4.0%	-4.2%	-9.4%	-11.6%	-7.4%
	Average	-5.0%	-7.6%	-9.4%	-4.8%	-6.3%	-8.0%	-10.9%	-8.2%
D	BasketballPass	-8.0%	-7.5%	-16.6%	-7.2%	-8.9%	-7.9%	-17.2%	-10.0%
	BQSquare	-12.4%	-3.8%	-5.9%	1.1%	-12.8%	-4.3%	-6.8%	-2.4%
	BlowingBubble	-6.2%	-5.3%	-6.6%	-7.3%	-7.0%	-5.9%	-8.4%	-9.2%
	RaceHorses	-5.6%	-8.8%	-8.6%	-6.0%	-7.4%	-9.0%	-10.4%	-8.3%
	Average	-8.0%	-6.4%	-9.4%	-4.9%	-9.0%	-6.8%	-10.7%	-7.5%
All		**-5.8**	**-8.1**	**-9.5**	**-5.5**	**-7.3**	**-8.9**	**-11.2**	**-9.2**

Table 3.4 – BD-BR metric for performance comparison of the proposed CNN-based QE method as Post-Processing in the RA coding configuration on top of VTM-10.0 (High QP).

Class	Sequence	Prediction-unaware				Prediction-aware			
		BD-BR (PSNR)			BD-BR	BD-BR (PSNR)			BD-BR
		Y	U	V	(VMAF)	Y	U	V	(VMAF)
A1	Tango	-6.1%	-12.7%	-11.7%	-6.8 %	-7.9%	-13.6%	-12.8%	-11.8 %
	FoodMarket	-5.8%	-11.6%	-10.0%	-12.0%	-7.7%	-11.7%	-11.0%	-15.5 %
	CampFire	-5.9%	-4.3%	-10.6%	-9.1%	-7.4%	-6.8%	-13.6%	-12.7 %
	Average	-5.9%	-9.5%	-10.8%	-9.3%	-7.7%	-10.7%	-12.4%	-13.3 %
A2	CatRobot	-6.8%	-11.0%	-11.5%	-5.7%	-7.8%	-11.2%	-11.2%	-10.9 %
	Daylight	-8.1%	-8.8%	-3.9%	-9.7%	-9.0%	-8.8%	-5.1%	-15.4 %
	ParkRunning	-3.2%	-2.7%	-3.4%	-1.8%	-4.2%	-2.9%	-3.9%	-5.3 %
	Average	-6.1%	-7.5%	-6.3%	-5.7%	-7.0%	-7.6%	-6.7%	-10.6 %
B	MarketPlace	-4.6%	-5.8%	-9.2%	-5.2%	-5.2%	-6.2%	-10.0%	-7.3 %
	RitualDance	-6.2%	-8.0%	-12.5%	-8.3%	-7.6%	-9.0%	-13.8%	-11.2 %
	Cactus	-5.5%	-5.9%	-9.3%	-6.4%	-6.7%	-6.0%	-10.3%	-8.5 %
	BasketballDrive	-5.8%	-2.2%	-13.4%	-4.7%	-7.0%	-9.4%	-14.4%	-7.7 %
	BQTerrace	-6.7%	-7.7%	-6.6%	0.1 %	-7.4%	-8.4%	-7.6%	-3.3 %
	Average	-5.8%	-5.9%	-10.2%	-4.9%	-6.8%	-7.8%	-11.2%	-7.6 %
C	BasketballDrill	-6.6%	-10.2%	-16.3%	-6.4%	-7.9%	-10.8%	-15.8%	-9.2 %
	BQMall	-6.4%	-3.7%	-6.6%	-5.0%	-7.2%	-4.0%	-6.9%	-8.4 %
	PartyScene	-5.9%	-3.6%	-6.6%	-4.0%	-6.3%	-3.8%	-6.7%	-7.0 %
	RaceHorses	-3.7%	-9.0%	-9.4%	-4.1%	-4.9%	-9.7%	-11.7%	-7.1 %
	Average	-5.6%	-6.6%	-9.7%	-4.9%	-6.6%	-7.1%	-10.3%	-7.9 %
D	BasketballPass	-9.0%	-6.3%	-16.8%	-6.7%	-9.4%	-6.8%	-16.8%	-9.2 %
	BQSquare	-12.8%	-0.5%	-3.7%	0.1 %	-12.9%	-1.2%	-4.2%	-2.8 %
	BlowingBubble	-6.6%	-3.6%	-4.7%	-6.7%	-7.5%	-4.3%	-6.2%	-8.4 %
	RaceHorses	-5.7%	-7.9%	-8.8%	-5.3%	-7.1%	-8.3%	-9.3%	-7.3 %
	Average	-8.5%	-4.6%	-8.5%	-4.7%	-9.2%	-5.2%	-9.1%	-6.9 %
	All	**-6.4%**	**-6.6%**	**-9.2%**	**-5.7%**	**-7.4%**	**-7.5%**	**-10.1%**	**-8.9%**

Explicit Model Selection

In Table 3.5 the compression performance of proposed method in four settings including pred-unaware, pred-aware (frame-level), pred-aware + Implicit Model Selection (IMS) and pred-aware + Explicit Model Selection (EMS) in terms of BD-BR are presented. As can be seen, the proposed methods in all four settings provide a significant improvement in coding gain on all test sequences with average of -5.8%, -7.1%, -7.3% and -7.6% in pred-unaware, pred-aware, pred-aware + IMS, and pred-aware + EMS settings, respectively. It can be observed that adding prediction information brings -1.33% more bitrate saving to the base QE setting. Moreover, using MS strategy adds -0.46% more bitrate saving, where in some sequence like CampFire, BQTerrace and PartyScene the gain is more than -1.3% in average.

Table 3.5 – BD-BR metric for performance comparison of the three versions of the proposed CNN-based QE method as PP in the RA coding configuration of VTM-10

Class	Sequences	Pred-unaware	Pred-aware	Pred-aware + IMS	Pred-aware + EMS
A1	Tango	-5.6 %	-8.4%	-8.5 %	-8.5%
	FoodMarket	-3.6 %	-7.0%	-7.2 %	-7.4%
	CampFire	-4.3 %	-5.2%	-5.9 %	-6.7%
	Average	-4.5 %	-6.8%	-7.2 %	-7.5%
A2	CatRobot	-6.9 %	-8.4%	-8.5 %	-8.6%
	Daylight	-9.3 %	-10.6%	-10.8 %	-10.9%
	ParkRunning	-3.0 %	-4.0%	-4.1 %	-4.3%
	Average	-6.4 %	-7.7%	-7.8 %	-7.9%
B	MarketPlace	-4.8 %	-5.5%	-5.6 %	-5.6%
	RitualDance	-6.0 %	-7.7%	-8.0 %	-8.1%
	Cactus	-4.2 %	-5.9%	-6.0 %	-6.6%
	BasketballDrive	-5.4 %	-6.7%	-7.0 %	-7.5%
	BQTerrace	-4.9 %	-5.8%	-5.9 %	-7.0%
	Average	-5.1 %	-6.3%	-6.5 %	-7.0%
C	BasketballDrill	-6.5 %	-8.2%	-8.3 %	-8.6%
	BQMall	-5.2 %	-6.6%	-6.7 %	-7.3%
	PartyScene	-5.3 %	-5.7%	-6.1 %	-7.1%
	RaceHorses	-2.8 %	-4.1%	-4.2 %	-4.3%
	Average	-5.0 %	-6.1%	-6.3 %	-6.8%
D	BasketballPass	-8.0 %	-8.7%	-8.9 %	-9.0%
	BQSquare	-12.4%	-12.5%	-12.8 %	-13.1%
	BlowingBubble	-6.2 %	-6.9%	-7.0 %	-7.1%
	RaceHorses	-5.6 %	-7.3%	-7.4 %	-7.4%
	Average	-8.0 %	-8.9%	-9.0 %	-9.1%
	All	-5.8 %	-7.1%	-7.3 %	-7.6%

3.4.3 Performance evaluation of PP

In this section, performance of a set of recent PP methods developed for RA of VVC are compared to our proposed method. For this purpose, two academic papers [21, 27] and three JVET contributions [116, 117, 118], have been selected. The coding gain of these works is extracted from its corresponding literature. It is important to note that a fair comparison of CNN-based QE methods in the literature is difficult since they use a network with different architecture and complexity levels.

The performance of five above-mentioned methods, as well as our proposed QE, in terms of BD-BR are summarized in Table 3.6. The average BD-BR of each class is shown for comparison. First, it can be observed that when our proposed QE is integrated as PP to the VVC, it outperforms all the competing methods. The performance improvement is consistent over the average of all classes. Secondly, the coding gain of our prediction unaware setting is less than the MFRNet [21]. It can be concluded that, by adding the same strategies to the network of MFRNet, we can even get higher coding gain. This subject will be studied in future. Finally, the work in JVET-T0079 [116] also benefits from intra prediction for enhancing the quality of intra coded frames in RA configuration. The higher BD-BR in our method compared to this work is likely due to the inter prediction and coding type mask that is employed in the proposed QE for enhancing the inter coded frames.

Table 3.6 – BD-BR comparison of the proposed method against state-of-the-art PP methods. All tests have been carried out in the RA mode and under JVET-CTCs.

Method		Class					
		A1	A2	B	C	D	**All**
State-of-the-art	JVET-O0132 [117]	-0.15%	-0.28%	-0.22%	-0.59%	-0.80%	**-0.40%**
	JVET-O0079 [118]	-0.87%	-1.68%	-1.47%	-3.34%	-4.97%	**-2.47%**
	Zhang *et al.* [57]	-2.41%	-4.22%	-2.57%	-3.89%	-5.80%	**-3.76%**
	JVET-T0079 [116]	-2.86%	-2.98%	-2.92%	-2.96%	-3.48%	**-3.04%**
	MFRNet [21]	-6.73%	-7.16%	-6.30%	-6.00%	-7.60%	**-6.70%**
Proposed	Pred-unaware	-4.47%	-6.41%	-5.06%	-4.97%	-8.04%	**-5.79%**
	Pred-aware	-7.20%	-7.79%	-6.50%	-6.35%	-9.01%	**-7.31%**

3.4.4 Performance evaluation of ILF

In terms of complexity-performance trade-off, the main benefit of the ILF approach is that, due to the propagation of improvement, one can achieve higher compression gain by enhancing a few frames which are referred to the most in the given GoP structure. In this section, we evaluate this aspect. For this reason, here a set of ILF QE configurations are defined for presenting the performance in different conditions.

ILF QE configurations

The use of sixteen frames in one GoP, as recommended in the native RA configuration of VTM, results in five temporal layers Tid_i, with $0 \leq i \leq 4$. For the experiments of this section, we define six ILF QE configurations, to progressively increase the number of enhanced frames in the ILF approach. In the first configuration, noted as C_I, only the intra frame in the GoP is enhanced. In each of the other five configurations, represented as C_j, with $0 \leq j \leq 4$, all frames in the temporal layers Tid_i (with $i < j$) are enhanced (Table 3.7).

Table 3.7 – Description of the tested ILF QE configurations used for evaluation of the ILF approach. In each configuration, frames in some temporal layer of the GoP are enhanced (✓) and some are not enhanced (✗). All tests are carried out in the RA mode.

Temporal	ILF QE configuration						
layer ID	Ref	C_I	C_0	C_1	C_2	C_3	C_4
Intra	✗	✓	✓	✓	✓	✓	✓
0	✗	✗	✓	✓	✓	✓	✓
1	✗	✗	✗	✓	✓	✓	✓
2	✗	✗	✗	✗	✓	✓	✓
3	✗	✗	✗	✗	✗	✓	✓
4	✗	✗	✗	✗	✗	✗	✓

Additionally, we also present a configuration as C_{ref} which is equivalent to the VTM-10 encoder with no CNN-based QE and is included to be used as a reference. Using the defined ILF QE configurations, we present the results of the proposed ILF method in a progressive manner so that the impact of multiple-enhancement is reflected. For this purpose, four settings of the VTM encoder are evaluated:

— Prediction-aware ILF.

— Prediction-unaware ILF.

— Adaptive ILF (prediction-aware only).

— Prediction-aware PP.

Performance Evaluation of ILF QE configurations

Fig. 3.11 shows the evolution of ILF methods in different ILF QE configurations. These results are obtained by averaging over class C and D of test sequences. A constant dashed line at BD-BR=0% is shown to indicate the border between having compression gain and compression loss.

The first comparison is between the two proposed ILF methods presented with red and blue lines, for prediction-unaware and prediction-aware versions, respectively. In these versions the frame-level switch mechanism of the adaptive ILF method is not used. As can be seen, the proposed prediction-aware ILF method outperforms the prediction-unaware method in almost all configurations.

The effect of the multiple-enhancement can be seen in the shape of both prediction-unaware and prediction-aware ILF methods. More precisely, the BD-BR gain of the ILF methods progressively decreases around C_0 and C_1, and eventually becomes a BD-BR loss around C_3 and C_4.

As the adaptive ILF algorithm has the flexibility to apply the QE step on any arbitrary frame in the GoP, the defined ILF QE configurations are not applicable to it and its results are presented as a constant green dashed line. The only fair comparison is between the prediction-aware ILF (blue line) and the adaptive version. As can be seen, with the adaptive version we can guarantee the highest performance of the non-adaptive version. The reason behind this behaviour is that the adaptive version performs the MSE-based comparison at each frame ensures that the multiple-enhancement is not going to negatively impact the performance. Therefore, the QE task usually stops at the optimum ILF QE configuration.

The PP algorithm is adapted to the ILF QE configurations in order to make a comparison. It is important to note that such comparison with PP might not be entirely fair, as the subset of enhanced frames corresponding to given ILF QE configurations are not necessarily optimal subsets for PP. However, our experiments show that the difference is small enough for drawing a conclusion. The interesting comparison between the PP method and the two ILF methods is their crossing point. In other words, until around C_2, both ILF methods are better than the PP. However, after this configuration, the PP becomes better. The reason is that, in the first part (until C_2, the enhancement propagation is causing the ILF methods to be better than the PP. However, in the second part (C_3 and C_4), the negative impact of the multiple-enhancement effect entirely compensates the enhancement propagation effect. Therefore, the crossing point of the PP method and ILF methods shows how the balance between the enhancement propagation and the multiple-enhancement can impact the performance of ILF method.

The final observation from Fig. 3.11 is the fact that best performance of the PP (at C_4) is better than the best performance of the ILF (shown with ILF-Adaptive line). This is most likely due to the fact that current ILF methods are not optimized with end-to-end training to be able

to enhance frames in higher temporal layers (*e.g.* at C_3 and C_4). Improvement of this aspect is left as future work.

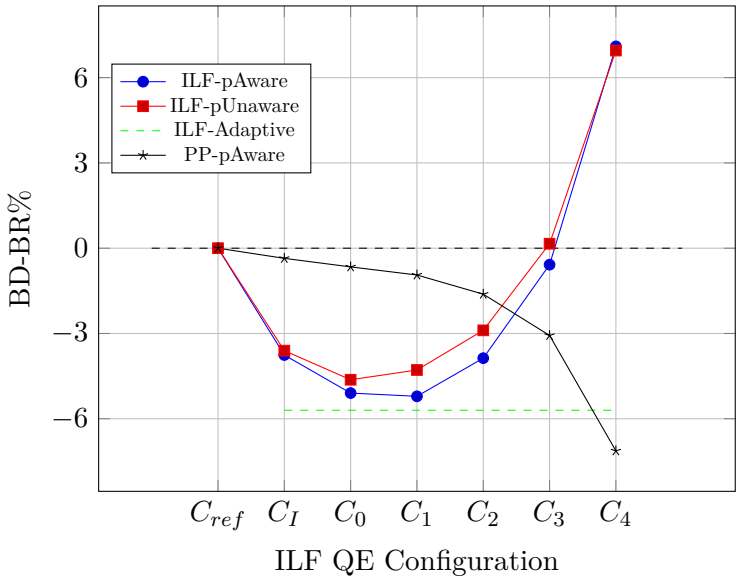


Figure 3.11 – Performance evaluation of different ILF QE configurations of the ILF approach, in terms of BD-BR. The proposed prediction-aware PP performance is also presented (black line) in order to magnify the performance drop due to the multiple enhancement effect in the last configurations of the ILF approach. All tests are carried out in the RA mode (Class C and D, CTC).

Performance vs Complexity Trade-off

In general, by applying the ILF approach to the frames in lower temporal layers, the side effect of multiple enhancement can be controlled. For instance, if only first I frame is enhanced and other frames remain intact, a noticeable gain will be obtained compared to the fact that only one frame is enhanced. In other words, by imposing the complexity of enhancing only one frame, we obtain a good portion of performance if we apply QE to all frames. To test this effect, we have calculated in table 3.8 the relative complexity of the configurations introduced in section 3.4.4. As can be seen, the encoding and decoding time is increasing as we apply QE filter to more temporal levels.

We can also see from Table 3.8 that, the decoder complexity of ILF in the adaptive configuration (C_{ad}) varies depending on how many frames have been enhanced at the encoder side. On the other hand, using PP does not impose any complexity at the encoder side, however at the decoder side, it dramatically increases the complexity.

Table 3.8 – Relative Complexity of the proposed PP and ILF, averaged on CTCs QP range, as in Eq. (3.17). Here, C_i refers to the ILF configurations presented in section 3.4.4. All tests have been carried out with the native RA coding mode of VTM-10.

Platform	Class	PP		$ILF - C_{ad}$		$ILF - C_I$		$ILF - C_0$		$ILF - C_1$		$ILF - C_2$		$ILF - C_3$	
		ET	DT	ET	DT	ET	DT	ET	DT	ET	DT	ET	DT	ET	DT
CPU	B	-	96.8	1.1	37.2	1.0	3.9	1.0	8.4	1.0	14.3	1.0	26.1	1.0	48.5
	C	-	329.9	1.2	78.8	1.0	12.3	1.0	26.3	1.1	46.5	1.1	87.0	1.1	167.9
	D	-	312.7	1.2	90.2	1.0	11.6	1.0	24.9	1.1	44.2	1.1	82.5	1.1	159.2
GPU	B	-	25.1	1.0	4.15	1.0	1.7	1.0	2.8	1.0	4.3	1.0	7.3	1.0	12.9
	C	-	14.4	1.0	2.6	1.0	1.4	1.0	2.0	1.0	2.8	1.0	4.5	1.0	7.8
	D	-	13.6	1.1	3.5	1.0	1.4	1.0	1.9	1.0	2.7	1.0	4.2	1.0	7.4

Performance comparison with state-of-the-art of ILF

Table 3.9 compares the performance of proposed QE methods, when integrated as ILF, against some state-of-the-art methods in literature. For this purpose, we chose MFRNet [21] and ADCNN [39] methods from academic papers in addition to three recent JVET contributions. As the source code of most of these works is not publicly available, the performance metric have been directly extracted from corresponding papers and documents. For representing both prediction-unaware and prediction-aware methods, we used the adaptive ILF implementation, described in Section 3.3.2, since it provides the highest performance.

It can be observed that our proposed prediction-aware ILF outperforms the proposed prediction-unaware method, by the coding gain of -5.85% compared to -5.12%, showing a consistent average BD-BR gain of about -0.73%. This result shows once more how the use of the prediction information can further improve the performance of a given CNN-based QE method.

Furthermore, the proposed prediction-aware method also significantly outperforms the selected papers from the state-of-the-art. It is worth to mention that since the benchmark methods use different network architecture and sometimes different training and test settings, this comparison might not entirely be fair. As future work, more efficient network architectures can be adopted from some of the benchmarks methods and integrated into the proposed prediction-aware framework of this work. Or inversely, one can implement the prediction-aware aspect of the proposed method on top of the benchmark methods and measure its performance changes.

In another analysis, the ILF results of Table 3.9 can be compared against the PP results in Table 3.6. By doing so, it can be observed that the PP approach is -1.4% better than the ILF one. This is mainly due to the multiple enhancement effect and the fact the CNN-based enhancement in some high temporal layers in the GoP have been avoided by the adaptive mechanism of the proposed method in order to control the quality degradation.

Table 3.9 – BD-BR comparison of the proposed ILF method against state-of-the-art methods, computed on the RA mode.

Method		Class			
		B	C	D	**All**
State-of-the-art	JVET-O0079 [118]	0.64%	-1.17%	-3.13%	**-1.22%**
	JVET-T0088 [119]	-3.44%	-3.38%	-3.48%	**-3.43%**
	JVET-U0054 [120]	-4.04%	-4.69%	-6.20%	**-4.98%**
	MFRNet [21]	-4.30%	-3.30%	-5.50%	**-4.37%**
	ADCNN [39]	-1.53%	-3.06%	-3.83%	**-2.81%**
	JVET-T0079 [116]	-3.25%	-2.85%	-3.13%	**-3.08%**
Proposed	Pred-unaware	-5.12%	-4.36%	-5.87%	**-5.12%**
	Pred-aware	-5.85%	-5.13%	-6.58%	**-5.85%**

3.5 Conclusion

In this chapter, we have proposed a CNN-based QE method to address the Post Processing (PP) and In-Loop Filtering (ILF) problems in VVC. Precisely, a filter which exploits the coding information such as prediction and QP is proposed in order to better enhance the quality. These coding information is fed to a proposed QE network based on the frame coding type (intra-frame or inter-frame), resulting in several trained models. Depending on the coding type used for a block (e.g inter mode, intra mode or skip mode), a model is selected among three models for the QE task. Moreover, in the ILF integration, in order to avoid the multiple enhancement issue, we adopt an adaptive framework to skip enhancement of frames posing this problem. Experimental results showed that the proposed PP, as well as ILF methods, outperform the state-of-art methods in terms of BD-BR.

Chapter 4

ML-based Dynamic Bitrate ladder construction

4.1 Introduction

Under strict bandwidth limitations, down-sampling of video before encoding, followed by an up-sampling to native resolution at the receiver side has proven benefits, in terms of overall resource usage optimization. Here, the sampling domain could be either spatial or temporal. In this chapter, we merely focus on the spatial domain sub-sampling, which is the most effective practice in low bitrate video transmission to adapt the encoding resolution to existing bandwidth bottlenecks. Depending on the video delivery application, the bandwidth bottleneck can occur in different places in a video delivery ecosystem. In this study, we particularly target the two most common cases:

1. Sender-side limitation: This case usually happens in live video applications, where the content is acquired in a location where physical and geographical constraints impose bandwidth limitations. Examples of such scenarios are news-gathering, on-site journalism, tour-based sport-event, etc. In the context of this thesis, several AVIWEST products fall into this category [1]. In this situation, the sender must decide about the resolution at which its real-time content should be encoded.
2. Receiver-side limitation: In contrast to the first category, the receiver-side bandwidth limitation attracts attention in offline applications, where receivers request video content in an on-demand manner. Examples of such applications are Subscribed Video on Demand (SVoD) services such as Netflix, Hulu, Prime Video, etc. In these applications, sender or intermediate cloud-based Content Delivery Network (CDN) service have to decide for each content, which encoded resolution should be delivered at a given bitrate.

In either case, the objective is to somehow decide the optimal resolution for a given content at a given bitrate. While the challenging aspect is that a bad or sub-optimal decision of resolution would result in a waste of resources, particularly compression efficiency or computational complexity.

1. https://www.aviwest.com/products/

Recently, in the literature and industry, adaptive methods for determining the best parameters for encoding have been developed. In such methods, each video is divided into several chunks, usually based on duration and scene changes in the video. Then, each chunk is encoded with variations of different pre-defined parameters such as QP, temporal and spatial resolution, intra period, codec presets etc. The convex hull of all R-D curves of encodings obtained from different parameter sets is then formed in order to find the highest quality in a given bitrate. This pre-processing step should be done on all video chunks of all video titles to determine the best parameters of encoding for each title at a specific bitrate.

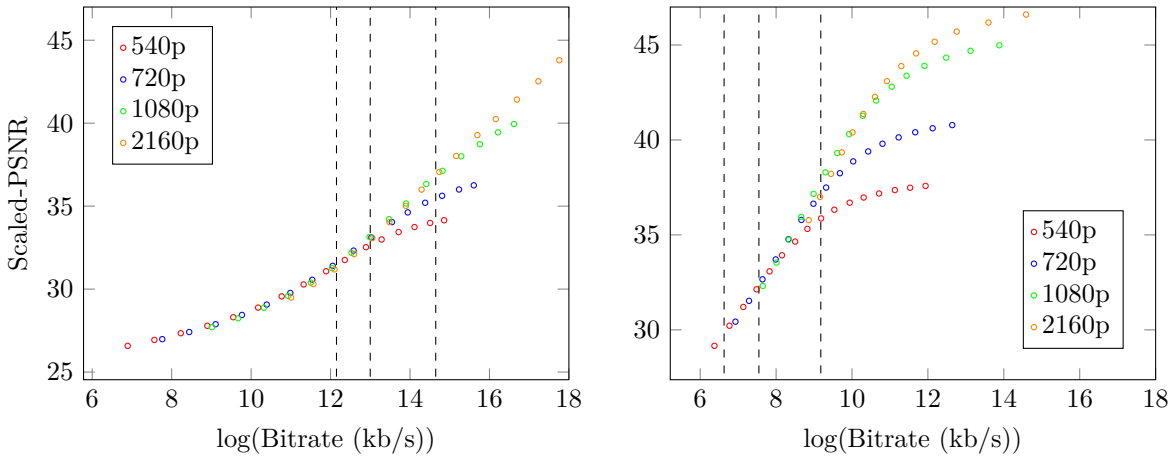
In Figure 4.1, the R-D curves of one second of two video sequences, encoded with VVC (VVenC 1.0.0), are shown. The parameter that has been varied to generate these curves was the encoded down-sampled resolution, while the native resolution of both videos is 4K/UHD. For constructing the R-D curves, the scaled-PSNR is calculated on the down-scaled videos. In this figure, the bitrate points where the R-D curves of two resolutions are crossing each other are shown by vertical dashed lines. These lines indicate at which bitrate one should switch from one resolution to another, in order to maximize the performance.

The important remark when comparing the two R-D curves in Figure 4.1 is that the cross-point bitrates of the two sequences are significantly different. Notably, in the first sequence (Figure 4.1-(a): Sparscut15) the cross-points happen around [12, 13, 15] log (kbps), respectively, while the cross-points of the same resolution switch in the second sequence (4.1-(b): Quad) happen around [6.5, 7.5, 9]log (kbps).

The above comparison proves that a content-independent fixed strategy for determining the encoding resolution would result in sub-optimal decision, hence waste of bandwidth resources. This motivates the second track of this thesis to design ML-based methods to take into account content characteristics and predict the optimal encoding resolution of videos.

The intuitive solution to address the adaptive resolution change problem is to obtain the full convex hull of the input sequence and determine the best resolutions for any given bitrate. Even though this simple approach is actually deployed in some streaming businesses where the number of video titles are small enough to be affordable, usually existing constraints keep conventional applications from doing so. Typically, two main constraints keep us from construction of the full convex-hull: time and resources. The time constraint often concerns senders in live encoding applications, where the decision about the best resolution has to be made in real-time and with low delay of encoding. While the resource constraint is mostly related to VoD and streaming applications, where even though encodings are run offline, encoding the full video for each of the possible resolutions would result in long and expansive processing on potentially costly cloud servers.

Considering the above behavior of different sequences as well as constraints in their traditional solutions, in this chapter, we propose two algorithms. Both of these algorithms are based



(a) Sparscut15 (b) Quad

Figure 4.1 – R-D curves of two video samples with different behavior in terms of resolution switching cross-point bitrates.

on supervised ML methods to predict at which bitrate the encoded resolution should change. This problem is typically known as bitrate ladder construction. The two proposed methods for the bitrate ladder construction are trained with low-level spatio-temporal features, extracted from the video sequences in their native resolution. Moreover, the proposed algorithms are designed to be used in CBR rate control mode, which is more realistic with respect to industrial applications.

4.2 Common aspects

4.2.1 Environment formulation

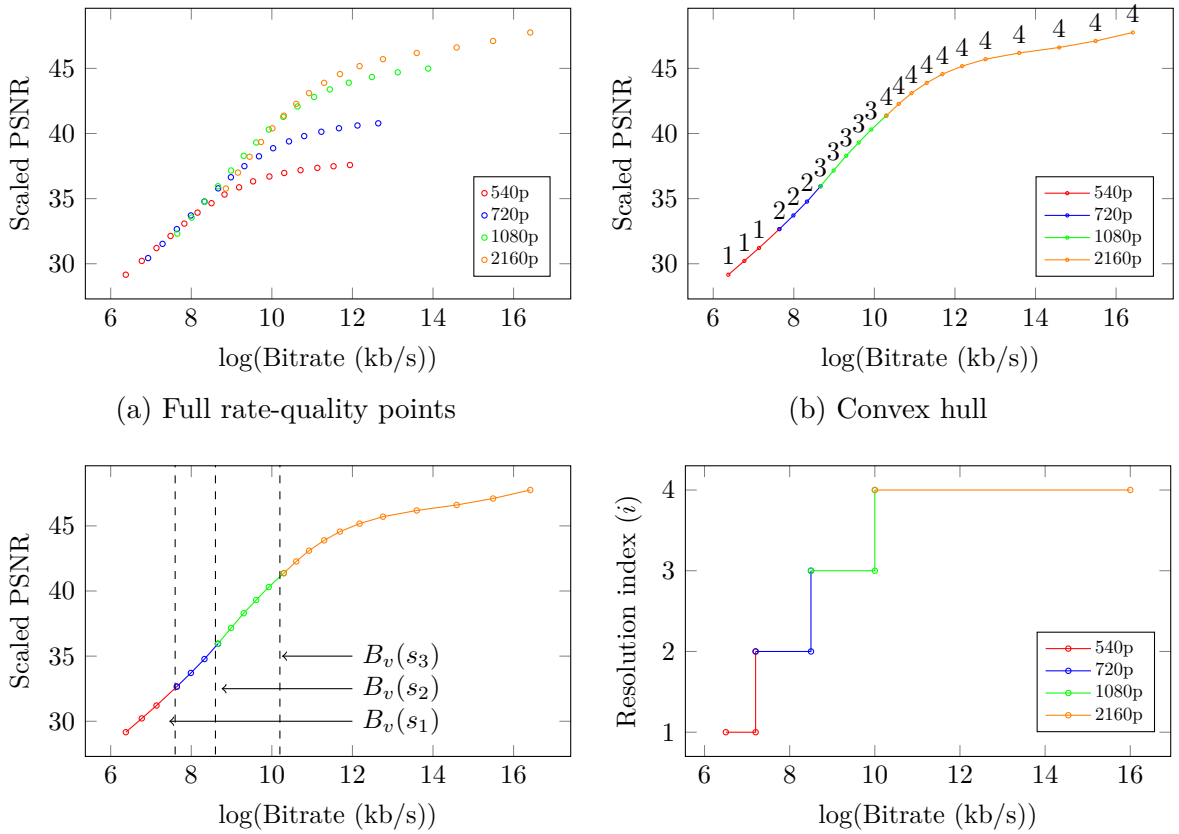
Let v be an input video sequence and $S = \{s_1, s_2, ..., s_{|S|}\}$ a set of resolutions in which v can be encoded. An encoder is also given whose task can be simplified in a function, denoted as E, which receives v and a resolution $s_i \in S$, as well as a target bitrate r.

The simplified output of the above encoder is a quality index q. Without loss of generality, we assume that the quality metric can potentially be any of the common objective metrics such as PSNR, VMAF or MS-SSIM. It should be mentioned that as the encodings are done in different resolutions the quality metric is calculated with scaled output signal in its native resolution.

Given the above specifications, the encoding of sequence v at resolution s_i, at bitrate r, which results in a quality q, can be expressed as:

$$q = E(v, r, s_i), \qquad \text{where } s_i \in S. \tag{4.1}$$

One can vary the two parameters r and s_i and output quality indexes of encoder E on video sequence v. These points generate a diagram of full rate-quality operating points, as shown in Fig. 4.2-(a). This diagram is used as the starting point for the task of bitrate ladder prediction.



(a) Full rate-quality points

(b) Convex hull

(c) Cross-points

(d) Bitrate ladder

Figure 4.2 – Four stages of constructing the bitrate ladder (d) from the full rate-quality points (a), through the convex-hull (b) and cross-point bitrate computations (c).

Given a full rate-quality operating point diagram as above, an indicator function called the convex hull can be defined for the encoder E on video v. This function denoted as $C^v(r)$, takes rate r as input and outputs the highest quality q^* that can be achieved at rate r among the different resolutions s_i, using the encoder E. This is expressed as follows:

$$q^* = C^v(r) \text{ where } E(v, r, s_i) \leq q^*, \text{ and for all } s_i \in S \tag{4.2}$$

In other words, the convex hull function $C^v(r)$ stores information about the upper-bound performance of the encoder E over the total range of bitrates. This information is visualized in Fig. 4.2-(b), where labels and colorization at given bitrate points indicate the resolution that is

resulting in the optimal quality q^*.

In this work, we assume that convex hulls are monotonic, and moreover, that each resolution switch is imperatively from resolution s_i (where $1 \leq i < |s|$) to resolution s_{i+1}, which is the immediate next larger available resolution. It is important to note that, since modern encoders are complex systems with numerous internal content-dependent functionalities, it is possible that in practice the above simplifications are violated.

Considering the above constraints, for each resolution s_i, a bitrate point r_i^* can be calculated in which the resolution switch must be applied. This bitrate is called the cross-point bitrate of resolution s_i in the rest of this section and is symbolically computed with a function $B^v(s_i)$:

$$\begin{aligned} r_i^* =& B^v(s_i), \\ & \text{where } C^v(r_i^*) = E(v, r_i^*, s_i) \\ & \text{and } C^v(r_i^* + \epsilon) = E(v, r_i^* + \epsilon, s_{i+1}). \end{aligned} \tag{4.3}$$

Eq. (4.3) computes for a given resolution s_i, the largest bitrate point as r_i^*, where the highest quality q^* is obtained by encoding in resolution s_i. This is such that after the computed bitrate point (*i.e.* addition of ϵ, where $\epsilon > 0$), a resolution switch to s_{i+1} is needed, as the highest quality obtained by the next resolution according to the convex hull function. Vertical dashed lines in Figure 4.2-(c) illustrates an example location of cross-point bitrates.

The bitrate ladder of a video sequence v is defined as a function that determines the optimal resolution for any given bitrate. A trivial approach to compute the bitrate ladder of a sequence is to actually encode it in all available resolutions and a sufficient number of bitrates. By doing so, one can obtain the full rate-quality operating points needed for Eq. (4.2) and Eq. (4.3). At this stage, the reference bitrate ladder of video v in resolutions defined in S, can be expressed as in Eq. (4.4). In this equation, i^* is the the index of the optimal resolution (*i.e.* ground-truth) and $L_{v,S}^*(r)$ is the reference ladder function that computes and returns this optimal index for a given rate r. Figure 4.2-(d) visualizes an example of a reference bitrate ladder computed from all operational rate-quality points.

$$i^* = L_{v,S}^*(r) \text{ where } B^v(s_{i-1}) < r \leq B^v(s_i). \tag{4.4}$$

It is noteworthy that in the rest of this chapter, depending on the context, reference bitrate ladder might alternatively be referred to as the "ground truth" bitrate ladder.

The problem that this chapter addresses is to obtain a bitrate ladder for a given sequence v, without actually having to encode it in all available resolutions in S. In other words, we aim at finding a function F such that:

$$\hat{L}_{v,S} = F(v, S), \tag{4.5}$$

To this end, two ML-based methods are proposed. Each of these methods particularly focus on one of above-mentioned applications where existing constraints (*i.e.* time and/or resources) keep us from constructing the reference bitrate ladder.

4.2.2 Encoder

Both algorithms proposed in this chapter are codec-agnostic and can be applied as a pre-processing step before virtually any codec. This is due to the fact that the features required for building the trained models are either based on spatio-temporal signal characteristics or output performance metrics of the codec, when used as black-box. In this study, the latest video standard, VVC has been chosen to be used for our experiments. However, unlike the previous chapter, instead of the reference VTM software, here VVenC, an optimized semi-industrial implementation of VVC is used. There are mainly two reasons for this choice:

1. While being able to perform almost at the same level as VTM, the VVenC codec is significantly faster. This helped us to enlarge our dataset, specially since the nature of the problem requires several encodings at 4K resolution.
2. VVenC is a multi-preset implementation. This aspect is essential for the second proposed algorithm of this chapter.

4.2.3 Dataset

One of the most important aspects in any ML-based method is to have a large and diverse training dataset. In this work, we collected a dataset of 300 videos from internal and public sources, including BVI SR [121], Derf collection [122], MCML [123], SJTU [124], and UGV [125]. All sequences have the native resolution of 3480×2160p with a frame rate of 60 fps. Some example frames of these sequences are shown in Figure. 4.3. These sequences are first converted to 8-bit and color formats of 4:2:0, as it was the only possible common format among them. Moreover, since the sequences had different durations, they were all split into chunks of one second (64 frames). It is worth mentioning that an additional scene change detection has been applied in order to ensure content homogeneity within each chunk and content diversity between different chunks.

In order to show the diversity of the dataset we have computed four descriptors, namely Spatial Index (SI), Temporal Index (TI), Colorfulness (CF) and Motion Vector (MV) [126]. The SI determines the edge energy in a frame based on Sobel filter and is calculated as follows:

$$SI = \max_{n}\{std[Sobel(I_n)]\};\ n = 1, 2, ..., N \tag{4.6}$$

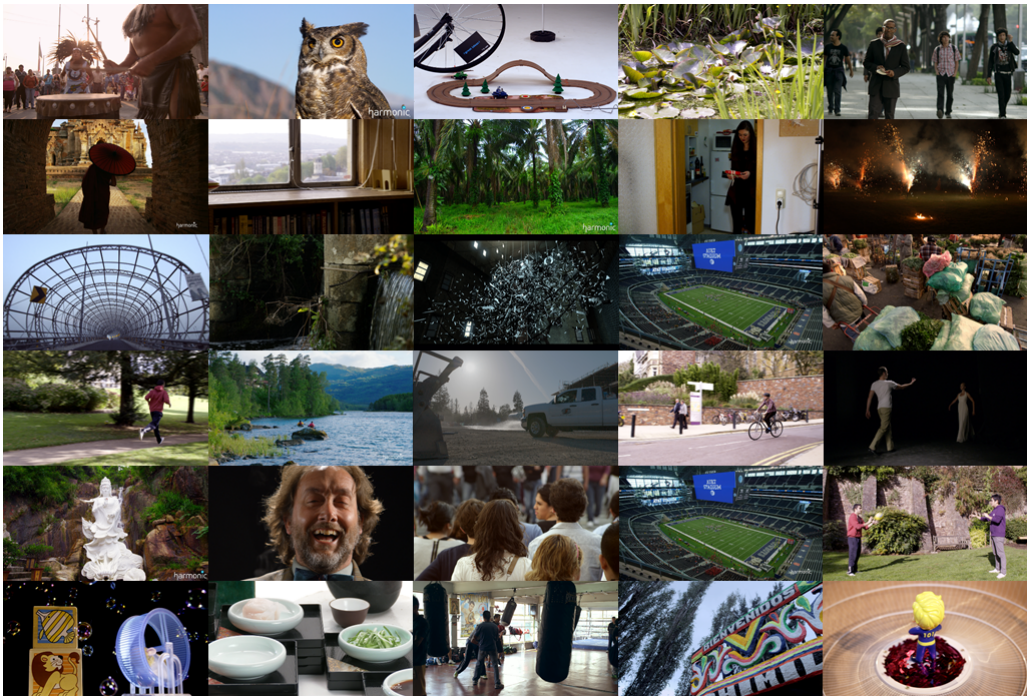


Figure 4.3 – Frame samples from the dataset

where I_n is the pixel values of the frame n and N is the total number of video frames. In other words, based on the above equation, the maximum values of the standard deviation of filtered frames over all the frames determine the SI values.

TI is based on the pixels difference between two consecutive frames, where higher values indicate higher movement in the video. This descriptor is calculated as:

$$TI = \max_{n}\{std[M_n(i,j)]\};\ n = 1, 2, ..., N \tag{4.7}$$

where M_n is computed based on pixel values $I_n(i,j)$ at pixel position (i,j):

$$M_n(i,j) = I_n(i,j) - I_{n-1}(i,j). \tag{4.8}$$

CF is an indicator for the colour distribution in a video, where higher values of the CF indicate that the video is more colored. In a video frame presented with three rgb color components, the CF is computed as follows:

$$C = \max_{time}\{\sigma_{rgyb} + 0.3 * \mu_{rgyb}\}, \tag{4.9}$$

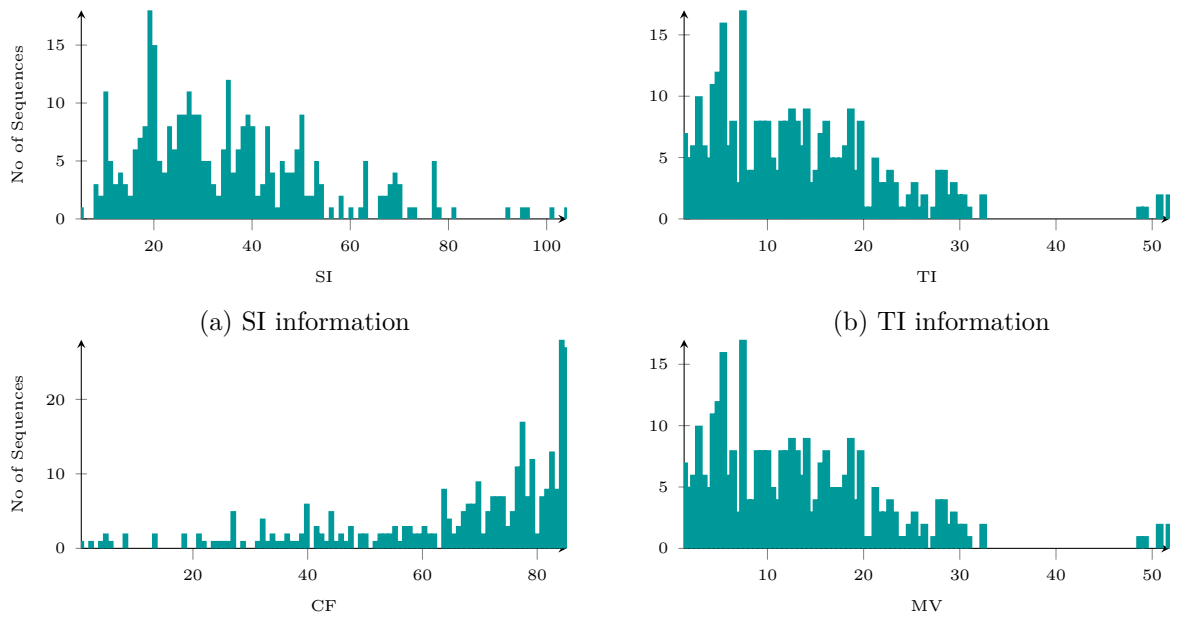
where σ_{rgyb} and μ_{rgyb} are computed as:

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \tag{4.10}$$

$$\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \tag{4.11}$$

In above equations, μ_{rg} and μ_{yb} are the mean and σ_{rg} and σ_{yb} are the standard deviation of signals $rg = R - G$ and $yb = \frac{1}{2}(R + G) - b$, respectively.

Finally, MV determines how fast the movements are in two consecutive frames. As simple block matching is used for MV calculation and the value of the MV indicator is the maximum motion element in the x-axis and y-axis. Hence, larger MV values indicate that movements of objects in the video are fast.



(a) SI information

(b) TI information

(c) CF information

(d) MV information (update it)

Figure 4.4 – Distribution of SI, TI, CF and MV descriptors of training dataset

In Figure 4.4, the distribution of the above four spatio-temporal descriptors are shown. As can be seen, the selected dataset covers a wide range of content characteristics. Most of the videos are in the range of 0-80 for SI and 0-35 for TI. We can also observe that there are some outliers that have higher values like the sequence Myanmar (row 2, column 1 of Figure 4.3) with the SI of 105 and TunnelFlag (row 3, column 1) with the TI of 50.

4.2.4 Features

In order to train the ML methods, it is important to extract a variety of features that collectively describe the characteristics of each video. To this end, we have extracted several spatial and temporal features from each sequence, represented in this section.

Spatial features

The videos with complex spatial characteristics are likely to have larger differences between neighboring pixels. Thus, in this work, we use Gray Level Co-occurrence Matrix (GLCM) [127] which is a traditional spatial descriptor which has been used in many studies for demonstrating spatial complexity. GLCM is composed of the intensity contrast of neighboring pixels in a video frame. Therefore, one can capture the level of coarseness as well as directional information of the video texture. GLCM has five main descriptors: contrast, correlation, energy, homogeneity, and entropy. For the frame I_n, let the GLCM be noted as G whose G_{ij} element is the number of occurrence for pairs (i, j) with the intensity value of Y_i and Y_j, i and j are defined depending on the dynamic range of image (*e.g.* between 0 and 255 for bitdepth of 8). Moreover, $p_{ij} = G_{ij}/K$ is the probability of pixel (i, j) assumes Y_i, Y_j values, where K is the number of occurrence. The five descriptors of GLCM are defined as below:

$$
\begin{aligned}
G_{contrast} &= \sum_{i=1}^{M}\sum_{j=1}^{N}(i-j)^2 p_{ij} \\
G_{correlation} &= \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{(i-m_r)(j-m_c)p_{ij}}{\sigma_r \sigma_c} \\
G_{energy} &= \sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}^2 \\
G_{homogeneity} &= \frac{p_{ij}}{1+|i-j|} \\
G_{entropy} &= -\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij} log_2 pij
\end{aligned}
\tag{4.12}
$$

In Eq. 4.12, M and N are the number of columns and rows of the frame, respectively. Also, m_r and m_c are the mean and σ_r, σ_c are the standard deviation values along rows and columns of G, respectively. The statistics of GLCM descriptors along the frames in a sequence including mean, standard deviation, skewness, and kurtosis are computed as features.

Temporal features

In addition to spatial features, to capture the temporal characteristics of the video, we have extracted the Temporal Coherency (TC) from two consecutive frames through the frames of the video. TC determines how easy a frame can be predicted from its previous frame and is computed using Fast Fourier Transform (FFT) as follows:

$$TC = \frac{|P_{I_{t-1}I_t}|}{P_{I_{t-1}I_{t-1}}P_{I_tI_t}}, \tag{4.13}$$

where $P_{I_{t-1}I_t}$ and $P_{I_tI_t}$ are the cross-spectral density of frames I_n and I_{n-1} and auto-spectral density of I_n, respectively. The value of TC is between [0, 1] and the higher values indicate that the video has high-frequency content and low motions. In this work, TC is computed over all pairs of two consecutive frames and the basic statistics of mean, standard deviation, skewness, and kurtosis are calculated at the sequence level.

In table 4.1, the list of all spatio-temporal features that are extracted for training the ML models is reported.

4.2.5 Reference convex hull construction

The ground truth of the training steps of this chapter is the reference convex hull of all given video sequences. Therefore, the first step is to construct these convex hulls for all video clips in our dataset. In the experiments, four resolutions are employed, namely $S = \{2160p, 1080p, 720p, 540p\}$, where the native resolution is 2160p and lower resolutions are obtained through down-sampling. Moreover, the employed quality metric was scaled PSNR, where the low-resolution samples are up-sampled into their native resolution, before the PSNR computation. For both down-scaling and up-sampling of the video sequences, the FFMPEG(3.2.10) [128] implementation of the Lanczos filter [129] has been used.

The process of constructing the convex hull from a sequence in the dataset is outlined in Algorithm 2.

The VVC codec that has been used is the VVenC (v.1.0.0), with Random Access (RA) coding configuration and with the GoP size of 32 frames. We also used the intra period of 64 which covers two full GoPs per video clip. The encodings are done in fixed QP mode for all resolutions ranging in $\{15, ..., 45\}$.

Figure 4.5 and Figure 4.6 show R-D curves and convex hull of some samples from the training dataset. In Figure 4.5, R-D curves of the video sequence in all resolutions are presented in different colors. While in Figure 4.6, only the convex hulls of these videos are shown. As can be seen, both the R-D curve sets and the convex hulls are highly diverse both in terms of the bitrate-PSNR range and their cross-points (*i.e.* change of color). This confirms that one bitrate

Table 4.1 – List of extracted features and their notation.

Feature	Notation	Feature	Notation
$F_1 = \frac{\sum_{i=0}^{N} G^i_{hom}}{N}$	$mean.G_{hom}$	$F_2 = \sqrt{\frac{\sum_{i=0}^{N}(G^i_{hom}-F_1)^2}{N}}$	$std.G_{hom}$
$F_3 = \frac{\sum_{i=0}^{N} G^i_{cor}}{N}$	$mean.G_{cor}$	$F_4 = \sqrt{\frac{\sum_{i=0}^{N}(G^i_{cor}-F_3)^2}{N}}$	$std.G_{cor}$
$F_5 = \frac{\sum_{i=0}^{N} G^i_{con}}{N}$	$mean.G_{con}$	$F_4 = \sqrt{\frac{\sum_{i=0}^{N}(G^i_{con}-F_5)^2}{N}}$	$std.G_{con}$
$F_7 = \frac{\sum_{i=0}^{N} G^i_{ent}}{N}$	$mean.G_{ent}$	$F_8 = \sqrt{\frac{\sum_{i=0}^{N}(G^i_{ent}-F_7)^2}{N}}$	$std.G_{ent}$
$F_9 = \frac{\sum_{i=0}^{N} G^i_{eng}}{N}$	$mean.G_{eng}$	$F_{10} = \sqrt{\frac{\sum_{i=0}^{N}(G^i_{eng}-F_9)^2}{N}}$	$std.G_{eng}$
$F_{11} = \frac{\sum_{i=0}^{N} G^i_{AMS}}{N}$	$mean.G_{AMS}$	$F_{12} = \sqrt{\frac{\sum_{i=0}^{N}(G^i_{AMS}-F_{11})^2}{N}}$	$std.G_{AMS}$
$F_{13} = \frac{\sum_{i=0}^{N} G^i_{dis}}{N}$	$mean.G_{dis}$	$F_{14} = \sqrt{\frac{\sum_{i=0}^{N}(G^i_{dis}-F_{13})^2}{N}}$	$std.G_{dis}$
$F_{15} = \frac{\sum_{i=0}^{N} TCmean^i}{N}$	$mean.TC_{mean}$	$F_{16} = \sqrt{\frac{\sum_{i=0}^{N}(TC^i_{mean}-F_{15})^2}{N}}$	$std.G_{mean}$
$F_{17} = \frac{\sum_{i=0}^{N} TCstd^i}{N}$	$mean.TC_{std}$	$F_{18} = \sqrt{\frac{\sum_{i=0}^{N}(TC^i_{std}-F_{17})^2}{N}}$	$std.G_{std}$
$F_{19} = \frac{\sum_{i=0}^{N} TCskew^i}{N}$	$mean.TC_{skew}$	$F_{20} = \sqrt{\frac{\sum_{i=0}^{N}(TC^i_{skew}-F_{19})^2}{N}}$	$std.G_{skew}$
$F_{21} = \frac{\sum_{i=0}^{N} TCkur^i}{N}$	$mean.TC_{kur}$	$F_{22} = \sqrt{\frac{\sum_{i=0}^{N}(TC^i_{kur}-F_{21})^2}{N}}$	$std.G_{kur}$
$F_{23} = \frac{\sum_{i=0}^{N} TCent^i}{N}$	$mean.TC_{ent}$	$F_{24} = \sqrt{\frac{\sum_{i=0}^{N}(TC^i_{ent}-F_{23})^2}{N}}$	$std.G_{ent}$
$F_{25} = \frac{\sum_{i=0}^{N} SI_i}{N}$	$mean.SI$	$F_{26} = \sqrt{\frac{\sum_{i=0}^{N}(SI_i-F_{25})^2}{N}}$	$std.SI$
$F_{27} = \max_{i=1,..,N} SI_i$	$max.SI$	$F_{28} = \min_{i=1,..,N} SI_i$	$min.SI$
$F_{25} = \frac{\sum_{i=0}^{N} TI_i}{N}$	$mean.TI$	$F_{26} = \sqrt{\frac{\sum_{i=0}^{N}(TI_i-F_{25})^2}{N}}$	$std.TI$
$F_{27} = \max_{i=1,..,N} TI_i$	$max.TI$	$F_{28} = \min_{i=1,..,N} TI_i$	$min.TI$

ladder would not be appropriate and that content should be taken into account.

4.2.6 Anchor bitrate ladders

As VVC has not been widely used as the compression codec in any sector of the streaming/broadcast ecosystem, there is neither officially nor unofficially no defined static VVC bitrate ladder in the literature/industry. In order to address this issue and provide a reference point to our performance measurements, we calculated the average bitrate ladder through our training dataset and considered it as the static VVC bitrate ladder in the experiments. In addition to the static ladder, the fully specialized bitrate ladders computed from exhaustive encoding in different resolutions for each sequence in the dataset have also been used as benchmarks. This

Algorithm 2 Convex Hull construction

input: v, S
output: $C_v(r)$
for s in S **do**
  Downscale v to resolution s
  for qp in QP **do**
    $q = E(v, qp, s)$ Encode to obtain scaled $PSNR(q)$
    Compute bitrate r
  end for
end for
Compute the convex hull: $q^* = C_v(r)$ where $E(v, r, s_i) \leq q^*$ for all $s_i \in S$
return r^*, q^* point on convex hull

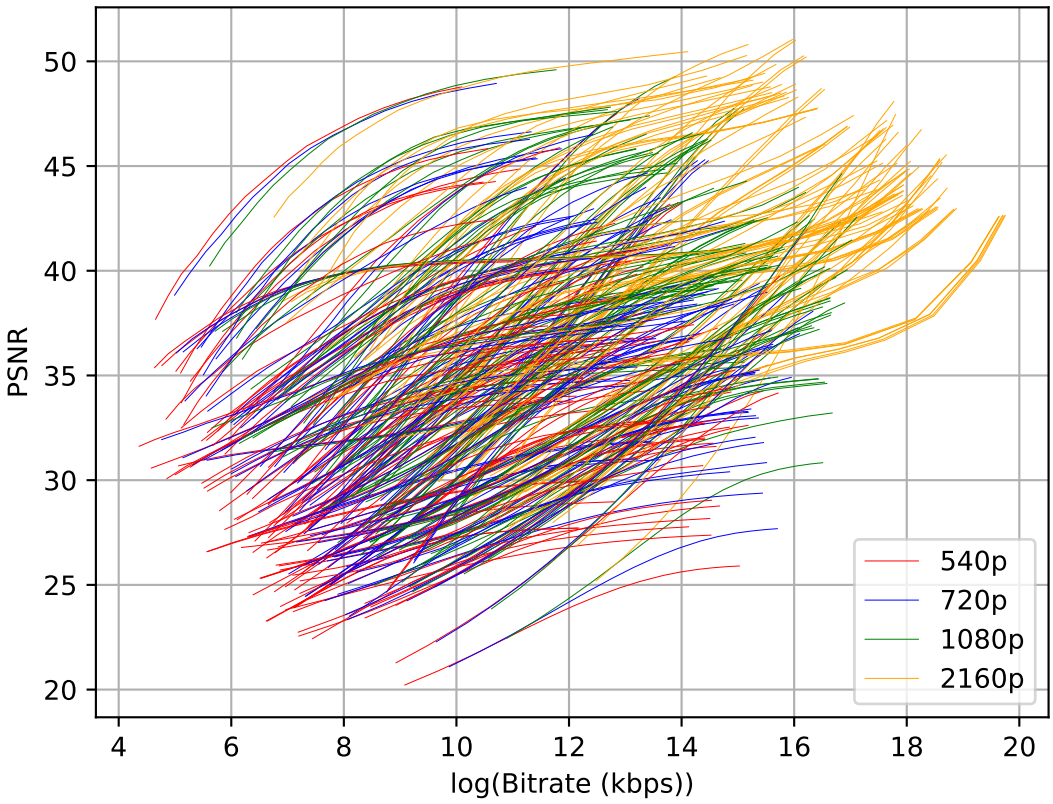


Figure 4.5 – R-D curves of training dataset in four resolutions.

ladder is referred to as the GT ladder in the results section.

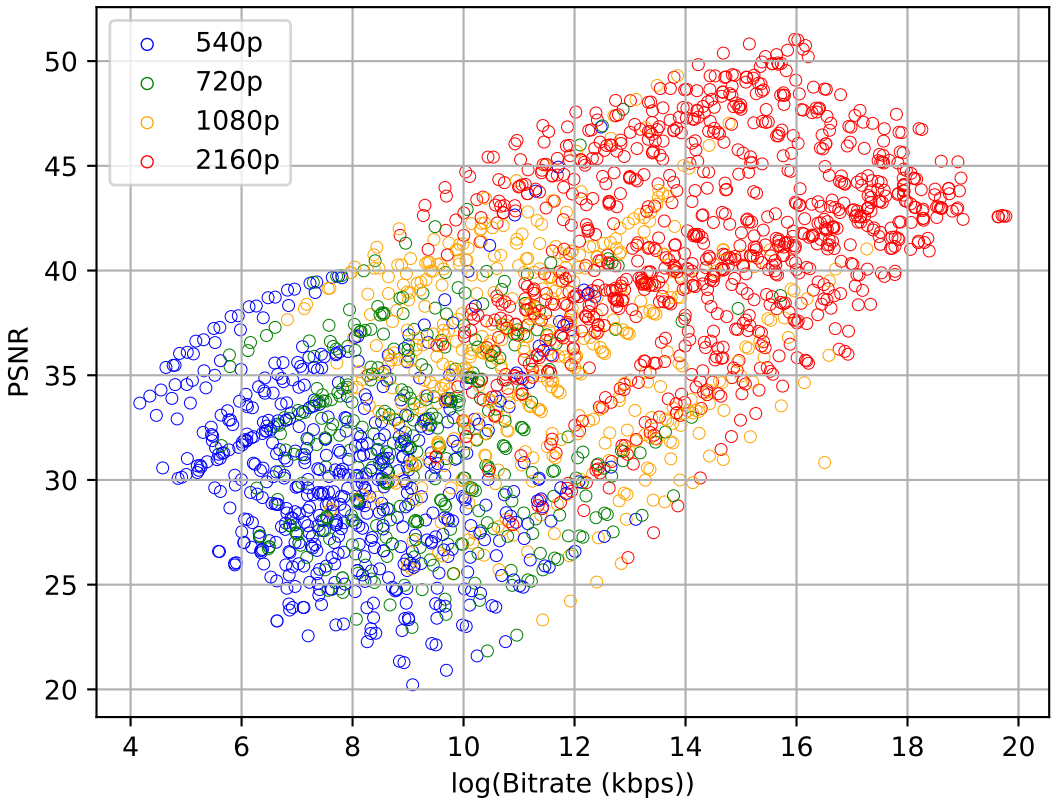


Figure 4.6 – Upper band convex hull of R-D curves of training dataset in four resolutions

4.2.7 Content-adaptiveness of resolution switch

Now that the different concepts related to the bitrate ladder construction are described, it is worth a few words for elucidating why do we even need to switch between different resolutions in the first place. In other words, what is the explanation behind the fact that in certain bitrates (typically low bitrates), one must go from the original resolution to lower resolutions in order to optimize the coding efficiency performance.

Two assumptions are the basis of the the current problematic. First, R-D curves of different resolutions of the same video always cross each other. Second, the location of points where the R-D curves cross each other varies from one sequence to another. In this work, we make a hypothesis that the above content-based variations are correlated with the selected spatio-temporal features and can be learned by an ML-based method.

Before moving to the next section, it is worth discussing in further detail why the R-D curves of different resolutions tend to cross each other in the first place. In other words, why

in certain bitrate ranges it is more efficient to change the resolution of the video in order to obtain better compression efficiency. To explain this phenomenon, one must take into account the decision making engine within an encoder, which is often used as a black box in this context. Precisely, this argument is related to how much bitrate is spent on two main elements of the bitstream: residual and motion vector. And how the video resolution impacts their balance. From the hybrid block-based compression point of view, the typical R-D behavior of encoding in different resolutions and bitrates is schematically shown in Figure 4.7. As can be seen, low resolution outperforms high resolution before certain bitrate cross-point.

The justification is that in the low bitrate ranges, typically the number bits used to transmit the motion vectors that are used to predict the motions in the videos is comparable with the number of bits used to transmit residual information of the blocks. Therefore, in the video with higher resolution in this bitrate range, the encoder allocates fewer bits to the residual information and the decoded video usually suffers from severe quantization distortion. As a result, the gap between the R-D curves of two encoded videos in two resolutions at low bitrate is mostly due to the temporal complexity of the video. As the bandwidth increases, the encoder can allocate more bits to the residual information. Therefore, in both resolutions, enough bits are allocated to the residual information compared to the coding signals. In this bitrate range, the distortion imposed by up-sampling filter is the source of the quality gap between encoded videos in two resolutions. Such distortion increases when the spatial complexity of video increases. Thus, in this higher range, the gap is highly due to the spatial complexity of the video.

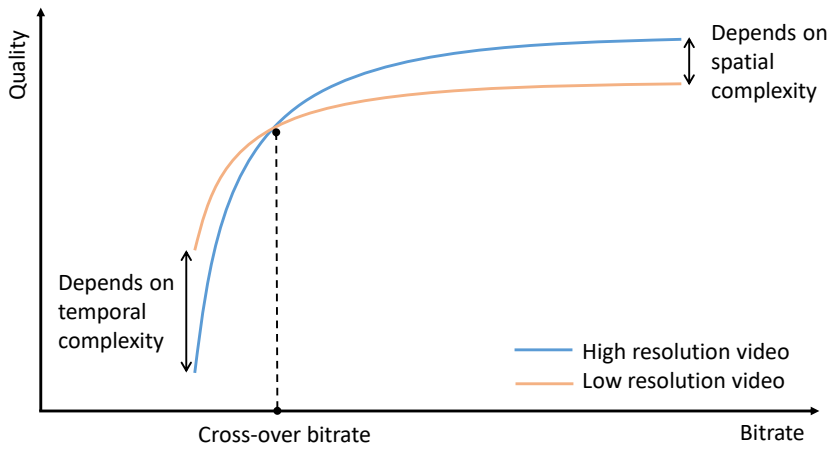


Figure 4.7 – Two R-D curve samples in two different resolutions showing how temporal and spatial complexity can change the gap between two curves and the position of the cross-point bitrate

Based on the above observations, the temporal and spatial characteristics of a video directly affect the points where the R-D curves of this video encoded in two resolutions cross each other.

It is also valid when we encode videos in more than two resolutions.

4.3 Live application: Ensemble bitrate ladder prediction

In this section, the first problem is introduced for which an ML-based solution is proposed and described in details.

4.3.1 Problem definition

As the name suggests, the encoding process has to take place in real-time for live video transmission applications. However, this does not change the fact that optimal video resolution in a given bandwidth is content-dependent and shall be obtained through content-adaptive bitrate ladders. Hence, solutions to the ladder prediction problem must be simple enough to cope with the real-time aspect of these applications. In other words, the algorithm does not have much time for processing the content – whether through encoding passes or complex ML-based processes – in order to determine the ladder. Figure 4.8 schematically shows how and where the time constraint impacts the bitrate ladder prediction in a live video delivery ecosystem.

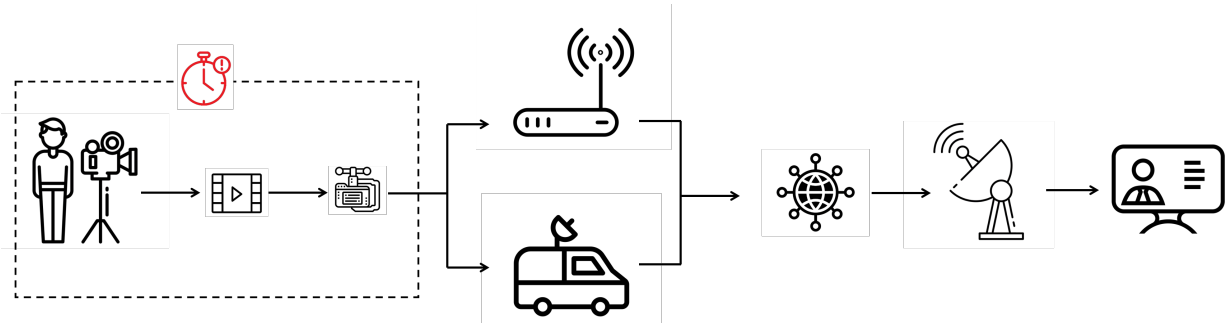


Figure 4.8 – The typical ecosystem of live video delivery applications and its constraint in terms of processing time.

In order to better simulate the live conditions of such a system, we chose to use the fastest possible preset of the VVenC encoder, namely the preset “faster”. Even though this preset is still far from real-time encoding for certain resolutions (*e.g.* 60 frames per second in 1080p), still it provides a realistic simulation of actual live video transmission. This above video encoder is used both as an actual video compressor (generating the bitstream to transmit through a network) and as a pre-processor for the construction of the bitrate ladder. The main objective of the algorithm is to minimize the number of encodings in the pre-processing phase while obtaining a ladder that maximizes the performance in the actual compression phase.

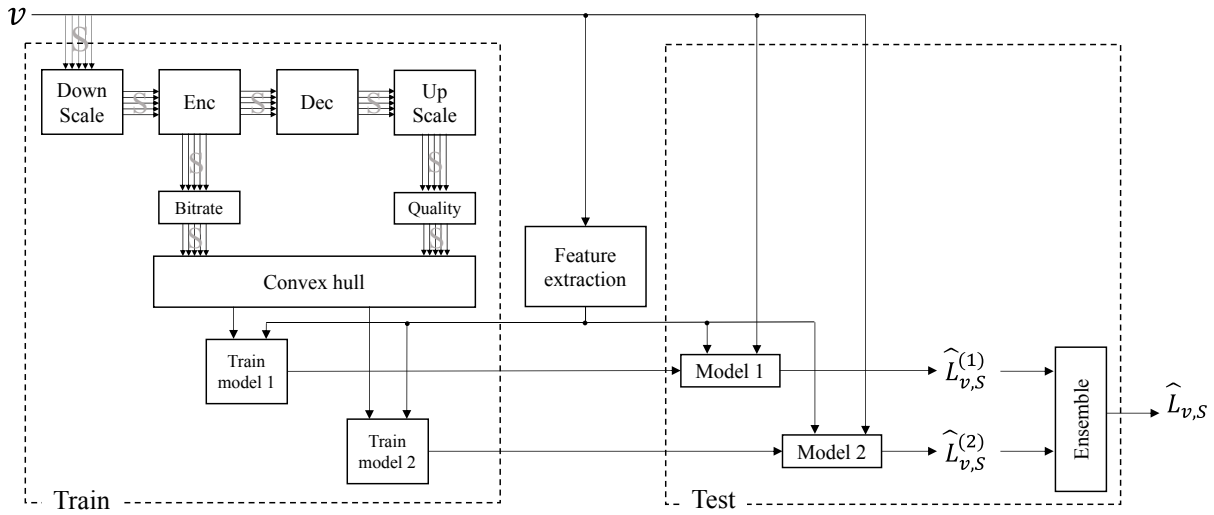


Figure 4.9 – Framework of the proposed method, including the "train" and "test" phases. The parallel arrows indicate the process has been carried out in all available resolutions of S.

4.3.2 Ensemble framework

The main contribution of this algorithm is the deployment of an ensemble machine learning model, which is a mechanism that allows combining multiple predictions coming from its constituent learning algorithms. The number of constituent methods can vary from two to several methods depending on the performance of the methods. The prediction process and inputs can be different in each ML method, however, eventually, the best resolution for a given bitrate is the output. In the proposed framework, we use an ensemble aggregator method to collect the output of all constituent methods and provide the final bitrate ladder.

Figure 4.9 shows the overall framework of our proposed method, including two main phases of “train” and “test”. These two phases, share a feature extraction step, which serves for the training and testing of the two constituent bitrate ladder prediction methods. The input video(s) v is to be represented in the highest possible resolution, specified by S. In the training phase, the goal is to independently train the two constituent methods, such that they can individually predict the bitrate ladder for any given video in the test phase. To do so, the high resolution input is down-sampled, encoded, decoded, and finally up-sampled, in order to provide the bitrate-quality points needed to construct the ground truth bitrate ladder. In the test phase, the two constituent methods are used to predict two potentially different ladders, which are then used as inputs to the ensemble aggregator for producing the final bitrate ladder prediction.

4.3.3 Proposed algorithm

Classifier constituent predictor

As the first constituent bitrate ladder prediction method, a multi-class classifier is used. At the core of this method, model M^{Cl} is trained that receives as input, the video sequence v and the target bitrate r, while the output is the index of predicted optimal resolution, defined in S:

$$\hat{i} = M_S^{Cl}(v, r), \text{ where } 1 \leq \hat{i} < |S|. \tag{4.14}$$

In other words, the first method directly predicts the value i in Eq. (4.4), without having to compute the cross-point bitrate P_v, denoted in Eq. (4.3). Therefore, by applying the core model M_S^{Cl} to all bitrate values, one can express the global operation of the classifier constituent predictor as:

$$\hat{L}_{v,S}^{Cl} = F^{Cl}(v, S). \tag{4.15}$$

Regressor constituent predictor

In the second method, a regressor is used to predict the cross-point bitrates. Given a resolution s_i (where $1 \leq i < |S|$) of the video sequence v, the regressor model that has learned the operation in Eq. (4.3), predicts at which bitrate the resolution should be switched to s_{i+1}:

$$\hat{r}_i = M_S^{Rg}(v, s_i). \tag{4.16}$$

By applying the regressor model in Eq. (4.4), instead of function B^v that identifies the cross-point bitrates, one can express the second constituent predictor as:

$$\hat{L}_{v,S}^{Rg} = F^{Rg}(v, S) \tag{4.17}$$

Ensemble aggregator

Once the two predictions of the bitrate ladder are computed by the constituent methods, the ensemble aggregator combines the two ladders and produces the final output, as:

$$\hat{L}_{v,S} = Agr(F^{Cl}, F^{Rg}) = F(v, S). \tag{4.18}$$

Algorithm 3 describes how the function Agr in Eq. (4.18) computes the final predicted bitrate ladder. The goal of this function is to take into account the two predictions made by the two constituents and determine the final resolution for each bitrate point. In case the two constituent predictions are the same, the aggregation is simply done by choosing the common prediction.

Algorithm 3 Ensemble aggregator *Agr*

input: $\hat{L}^{Cl}_{v,S}, \hat{L}^{Rg}_{v,S}$, isFast, MinRate, MaxRate
output: $\hat{L}_{v,S}$
for $r \coloneqq$ MinRate **to** MaxRate **do**
  $\hat{i}^{Cl} \leftarrow L^{Cl}_{v,S}(r)$
  $\hat{i}^{Rg} \leftarrow L^{Rg}_{v,S}(r)$
  if $\hat{i}^{Cl} = \hat{i}^{Rg}$ **then**
    $i^* \leftarrow \hat{i}^{Cl}$
  else
    if isFast **then**
      $i^* \leftarrow \arg\max_i E(v; r, s_i)$ where $i \in \{\hat{i}^{Cl}, \hat{i}^{Rg}\}$
    else
      $i^* \leftarrow \arg\max_i E(v; r, s_i)$ where $1 \leq i \leq S$
    end if
  end if
  $\hat{L}_{v,S}(r) \leftarrow i^*$
end for

However, in the case of different predictions, additional encodings by E are carried out to make the final decision. The number of encodings depends on a parameter, denoted as $isFast$ in Algorithm 3. If the fast mode is used, encoding is carried out only with the two resolutions predicted by the constituent methods. Otherwise, all possible resolutions are tested. In contrast with the "fast" mode, this mode is called the "full" mode in the rest of this section. In either mode, the resolution that provides the highest quality among the tested encodings is selected.

4.3.4 Training process

The two constituent methods are supervised ML methods that must be trained with appropriate features and ground truth data. The features extracted from video sequences (section 4.2.4) and the ground truth convex hull (section 4.2.4) are processed before using them to train the ML kernels. First, a reduction in the number of input variables can both reduce the computational cost of the modeling process and, in some cases, enhance its performance. To this end, before training the models a feature selection step is employed. Second, in order to find the best kernel to fit a model to our training data, different kernels should be evaluated. Thus, several ML methods for regression and classification have been tested with different parameters.

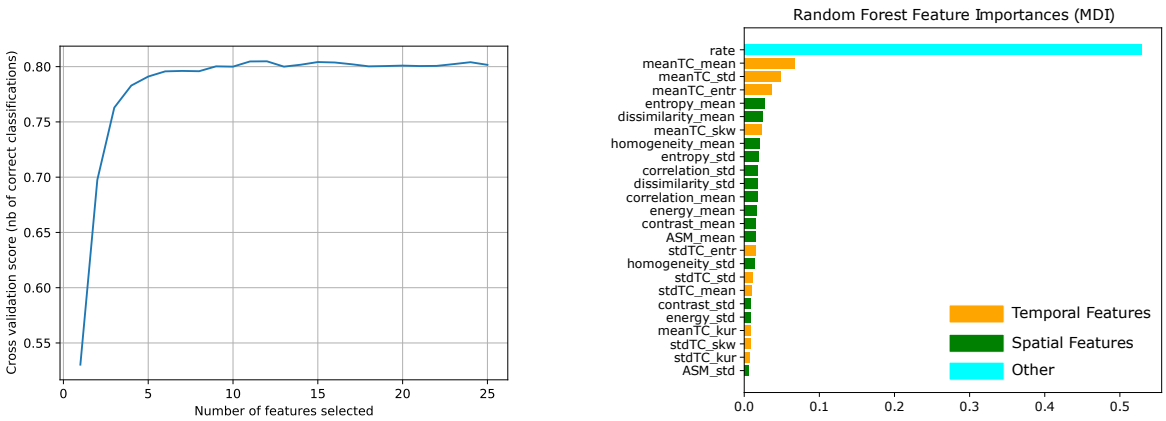
Features Selection

During the feature selection phase, the number of input variables is reduced to the minimum necessary in order to calculate a prediction of the target variable. Developing and training

predictive models with a large number of variables can be slow, requiring a large amount of memory in some cases and more importantly increases the risk of overfitting. In addition, some models may perform worse when input variables that are irrelevant to the target variable are included.

Thus, prior to using the extracted features to predict the bitrate ladder, we have used the RFE method [130] to select the most effective features. RFE involves recursively considering smaller and smaller sets of features, based on an external estimator that assigns weights to features. In our experiments, we have used the random forest as the estimator to compute the importance of the features.

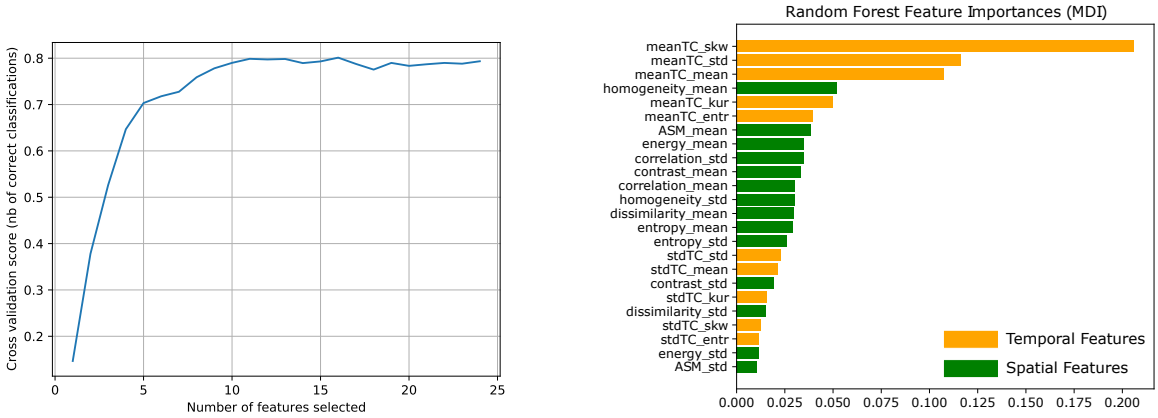
In Fig. 4.10(a) the ranking of the features for the classification problem is illustrated. As can be observed, the rate has been ranked as the first feature with the highest impact. The order of ranking shows that both spatial and temporal features (which are shown in different colors) are among high ranked features. Moreover, in Fig. 4.10(b), we have used RFE with automatic tuning of the number of features selected with cross-validation. Based on this figure the optimal number of features for the classification problem is about ten features.



(a) (b)

Figure 4.10 – Feature selection with RFE for classification constituent. (a): The ranking of all features. (b) Selection of optimal number of features with cross-validation

The same procedure is applied to the regression constituent for feature selection. In Fig 4.11.(a), the ranking of the features with RFE is shown. As can be seen, both temporal and spatial features are highly ranked. In addition, in Fig 4.11.(b) the selection of an optimal number of features with cross-validation is shown. As can be observed, the optimal number for regression constituent is ten features. It is worthy of mention that in the regression constituent, rate is not used as a feature, since it is considered as output and is to be predicted by the constituent model.



(a) (b)

Figure 4.11 – Feature selection with RFE for regression constituent. (a): The ranking of all features. (b) Selection of optimal number of features with cross-validation

ML methods

In order to find the proper ML methods for regression and classification, we trained and tested several methods. For classification, the decision tree classifier with gradient boost methods provided the best result compared to other kernels. Similarly, for the regressor models, after testing several methods, Gaussian Process (GP) provided the best results compared to other methods. Thus, we used the GP as the regressor for predicting the three cross-point bitrates.

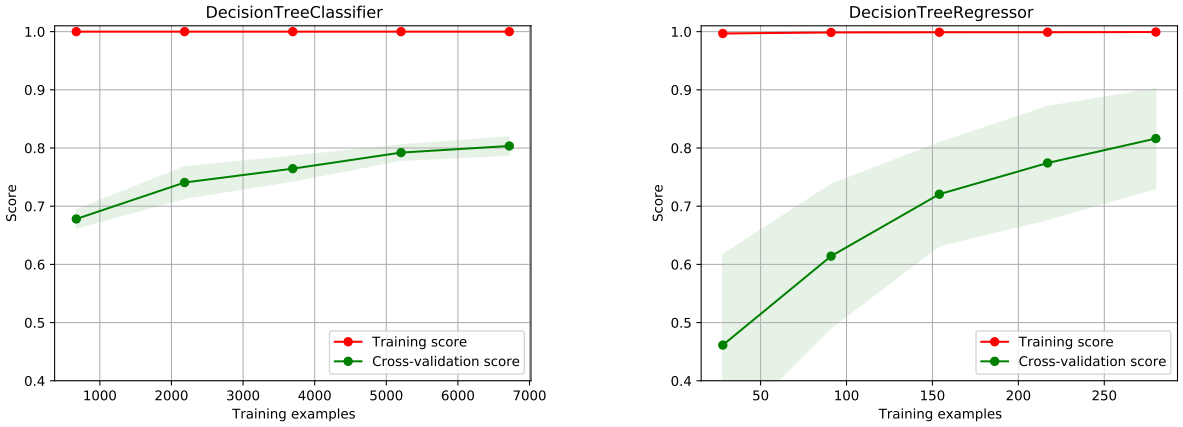
4.3.5 Experimental results

Learning curves

In Fig. 4.12 the learning curves of two constituent methods, classification, and regression, are illustrated. In the classification constituent, the cross-validation score reaches the score of 0.8, if we used all videos in our dataset. It can be seen that the training score is always around the maximum (1.0). More importantly, unlike the classifier model, the cross-validation score in the regressor method seems not to be saturated. In other words, it can be expected that its score would further increase by adding more video sequences to the dataset. As accessing a larger video dataset was not possible during this thesis, this aspects was left as a future work.

Prediction of cross-point bitrate

The results in Table 4.2 report the outcome of the ten-fold cross-validation with the accuracy of prediction metrics averaged over the ten folds in the regression constituent. In this table, each row represents the cross-point bitrates of two consecutive resolutions i and j, noted as B_{ij}. As



(a) Classification (b) Regression

Figure 4.12 – The learning curve of proposed classification constituent in training and cross-validation phases

can be seen, this table reports high values of R^2 and "Explained variance" of around 0.8 for all the three cross-point bitrates. Also, the Mean Absolute Error (MAE) and MSE are considerably low and comparable for all predicted cross-point bitrates.

In addition, in Fig 4.13, several samples of predicted (target) versus ground truth cross-point bitrates in cross-validation, and their scores are presented. Each row in this figure represents one resolution change, notably from 540p to 720p (first row), from 720p to 1080p (second row) and from 1080p to 2160p (third row). Also, the three diagrams in each row are selected randomly among ten available diagrams of the ten-fold cross-validation.

In this figure, the closer the points are to the blue line, the more accurately they are predicted. It can be observed that, in some samples, the predicted values are higher than what they should be and in some cases lower than the ground truth values. It is important to point out that the effectiveness of the method cannot be fully assessed by these results; the predicted cross-point bitrates will be utilized to estimate the bitrate ladder. Thus, the comparison of the predicted bitrate ladder to the reference will provide the full assessment of this framework.

Table 4.2 – Validation metrics of predicted cross-point bitrates.

Cross-over bitrate	R^2	Explained variance	MAE	MSE
B_{01}	0.79	0.79	0.54	0.72
B_{12}	0.84	0.80	0.53	0.67
B_{23}	0.78	0.80	0.66	1.00

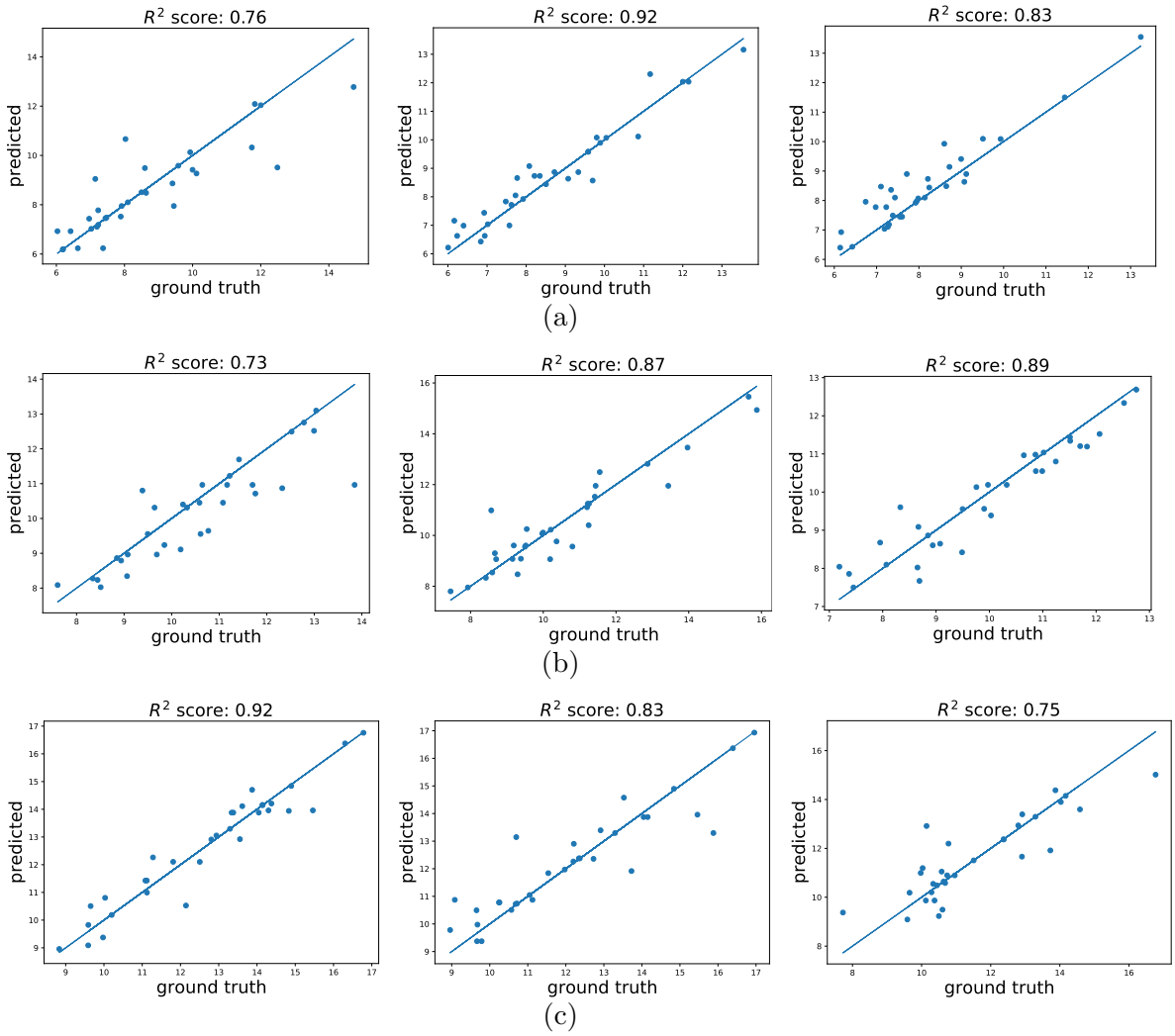


Figure 4.13 – Several samples of predicted (target) versus ground truth cross-point bitrates in cross-validation and their score (a) the cross-point bitrates between 540p to 720p (b) the cross-point bitrates between 720p to 1080p (c) the cross-point bitrates between 1080p to 2160p

Performance evaluation in terms of bitrate saving

For the performance evaluation of the proposed bitrate ladder prediction method, the BD-BR [131] metric has been used. To this end, we constructed R-D curves of available rate and distortions values and compared them with BD-BR metrics. To compute the BD-BR metric given two bitrate ladders, one ladder is used as "reference", while the other one is used as "test". Video sequences are then encoded in several bitrates, while their resolution is determined once by the "reference" ladder and once by the "test" ladder. The bitrate and scaled PSNR values are then collected and used with a mildly modified BD-BR computation in order to enable it

with more than four operational bitrate-quality points. It is important to note that in order to avoid overfitting, all results presented in this section are the output of tenfold cross-validation, and all the metrics are averaged over the ten folds.

Table 4.3 summarizes the performance evaluation of different settings of the proposed method. Notably, the first two rows present the performance of the two constituent predictors, when used outside the proposed ensemble framework. The last two rows are consequently the proposed ensemble method, when the "fast" and "full" modes are used, respectively.

The first metric demonstrates the accuracy of each method in the exact prediction of the optimal resolution over all tested bitrates. While the second and third metrics indicate the BD-BR performance versus the GT and static bitrate ladders, respectively. It is noteworthy that the negative values of the BD-BR metric indicate bitrate saving in the same level of quality, hence, should be considered as an improvement of performance.

The first observation is that the regressor method globally has a better performance than the classification method. However, both ensemble methods (with fast and full encoding) outperform the regressor method, in all three metrics. This proves that the ensemble approach is indeed helping to grasp the best out of each constituent predictor. This can be observed with all three comparisons. Particularly, in terms of bitrate saving (*i.e.* the last two columns), the use of ensemble methods improve the performance. Precisely, in terms of bitrate loss compared to the GT as well as bitrate gain compared to the static method.

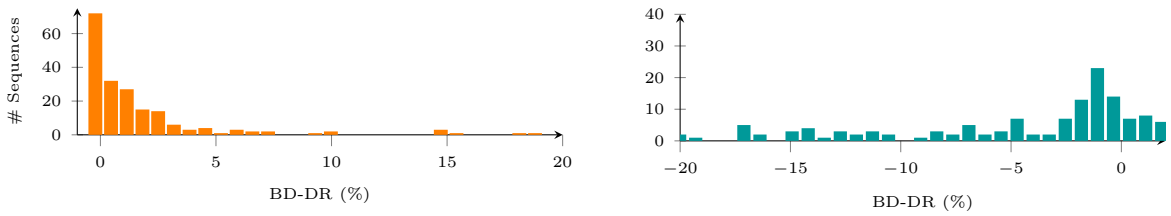
Table 4.3 – Average performance metrics of four different versions of the proposed method.

Method	Accuracy	BD-BR vs. GT	BD-BR vs. static
Classification	76%	2.97%	-11.45%
Regressor	83%	1.37%	-12.63%
Ensemble (fast)	90%	0.89%	-13.05%
Ensemble (full)	92%	0.77%	-13.14%

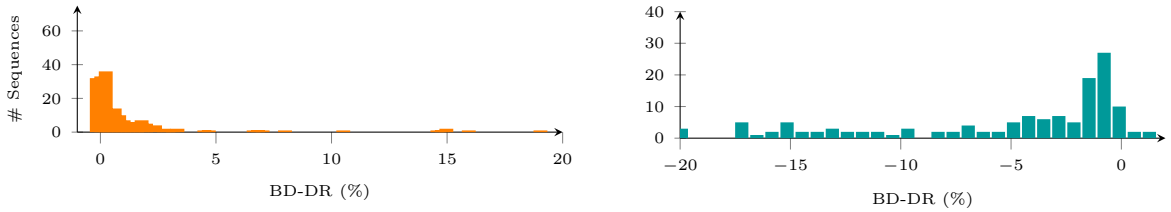
Fig. 4.14 provides a more detailed view of the BD-BR performance of the proposed methods on different sequences. Each diagram in this figure presents a histogram BD-BR metric on the test sequences. In the left column, the GT bitrate ladder has been used as a reference and positive BD-BR values indicate bitrate increase. Hence, being smaller is better.

In this sense, both ensemble methods significantly outperform the classification and regressor methods. Inversely, the results presented in the right column are obtained by using the static bitrate ladder as a reference. Hence, more negative values mean more gain.

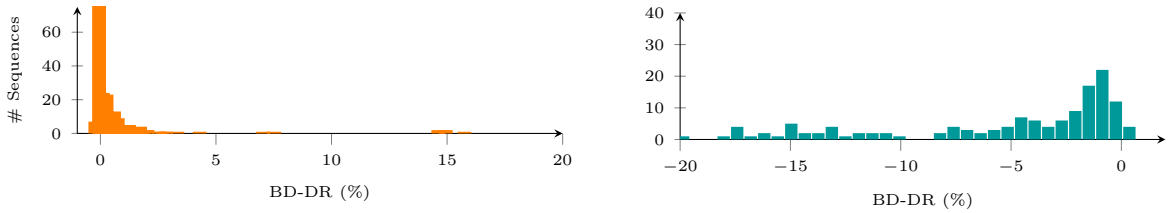
Additionally, in Fig 4.15 the bitrate ladder of two samples against the static bitrate ladder is illustrated. In this figure, the blue curve is the R-D curve, corresponding the static bitrate ladder, while the red curve is the R-D curve of the predicted bitrate ladder, using the proposed



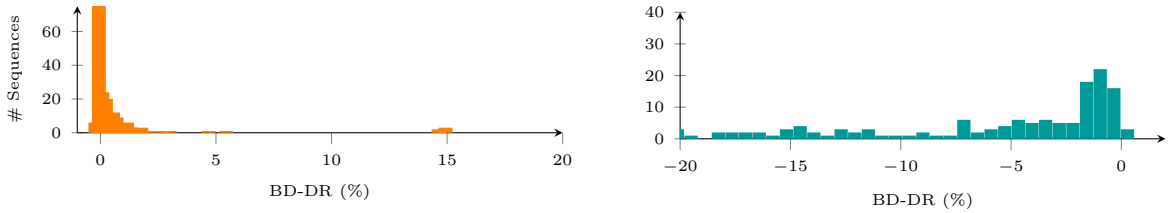
(a) Classification



(b) Regressor



(c) Ensemble (fast)

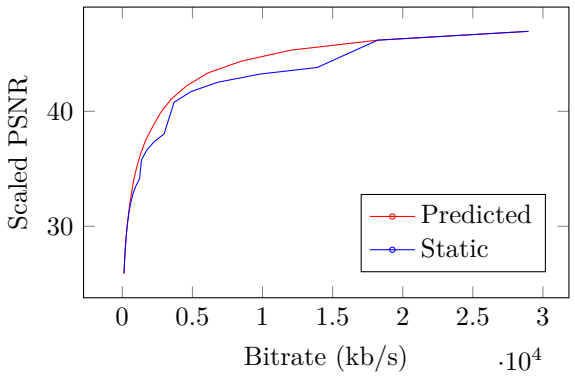


(d) Ensemble (full)

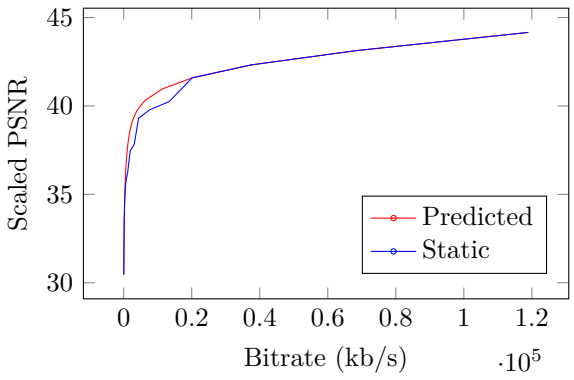
Figure 4.14 – Distribution of the BD-BR metrics on the test sequences. The left column presents the BD-BR metric versus the GT ladder, while the right column uses the static ladder as reference.

method. As can be seen, the different switching points that are not adapted to the content of the video in a static bitrate ladder, generate an inconsistent curve. However, the predicted bitrate ladder generates a rather smooth curve that provides optimal R-D values in different resolutions.

It is important to note that the red curves in Figure 4.15 are not necessarily convex, since they are also formed as concatenation of R-D curves corresponding to different resolutions (similar to the static ladder). The main reason that these curves look smooth and convex is that they are very close to the ground truth convex hull, hence possible discontinuities are not visible.



(a) Pond (row 1, column 4 of Fig 4.3)



(b) Sparkscut13 (row 4, column 4 of Fig 4.3)

Figure 4.15 – Comparison between bitrate ladder generated with static ladder (blue) versus predicted with ensemble method

Complexity

The additional gain brought by the ensemble methods is at the cost of encodings needed to aggregate decisions. To understand this impact, Fig. 4.16 demonstrates the average bitrate gain compared to the static bitrate ladder of different methods with respect to their complexity. The complexity metric of this experiment was the total encoding time spent for generating necessary bitrate-quality points of each method. As can be seen at the high quality extreme, the GT bitrate ladder method is highly complexity-intensive, while a significant portion of its BD-BR gain can be achieved by the proposed methods at much lower complexity. Conversely, on the low complexity extreme of the diagram, the two methods of classification and regressor impose no complexity overhead. However, their performance can be noticeably improved with a limited number of additional encodings by either of the ensemble methods.

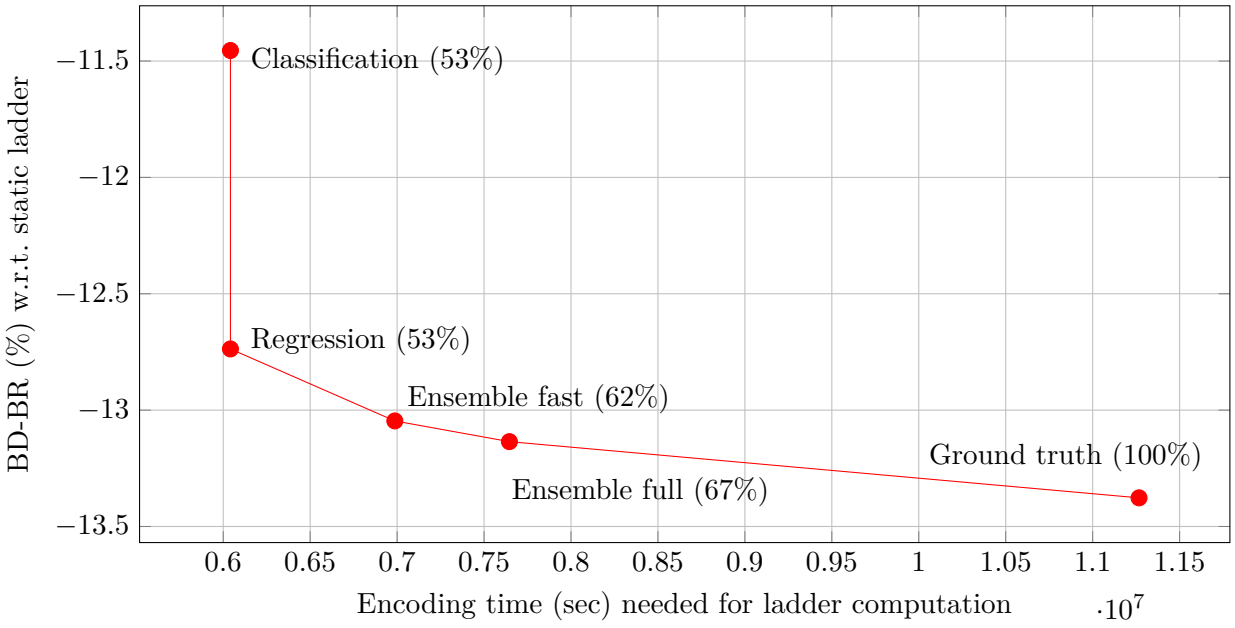


Figure 4.16 – BD-BR vs. complexity evolution of different methods. The numbers in parenthesis indicate the overhead in terms of encoding time with respect to the GT method as a reference.

4.4 VoD application: Fast-pass bitrate ladder prediction

4.4.1 Problem definition

Unlike live applications, the encoding does not have to take place in real-time for applications such as streaming and VoD services. As a result, transmitted video bitstreams in such applications are usually encoded with a more exhaustive rate-distortion optimization algorithm, allowing exploring for better rate-quality trade-offs, compared to real-time encoding. The means to this approach are multi-preset encoders which allow them choosing a preset that meets the global requirements of the pipeline, in terms of computational cost and quality fidelity. Given such encoder, the aforementioned applications would select a preset that is slow enough to provide high-quality encoded bitstreams, while it is fast enough to be affordable on the entire dataset of the streaming of VoD service

Multi-preset encoders provide a range of complexity-performance trade-off. Slower presets in such encoders spend more time on the RDO, resulting in a lower bitrate in a given quality, when compared to faster presets of the same encoder. Therefore, it is reasonable to use such presets for encoding of video titles in the VoD services, since the videos are encoded once and transmitted several times.

Often, the off-line encoding process in VoD applications is carried out on specialized cloud platforms that optimized hardware for video compression tasks. However, these services are not free of cost for professional usage. In other words, the more encodings VoD services launch on these platforms, the higher their encoding cost will become, and consequently, the lower their profit from their business. Therefore, even VoD services tend to limit their encoding-based computations.

There are typically two phases of encoding in VoD services. In the first phase, a number of encoding jobs are carried out to determine for each video title and each available bitrate, which resolution should be used for compression. It is important to note that the compressed bitstreams generated in this phase are likely to be discarded, since they serve only for statistical analysis purposes. In the second phase, the video title that was analyzed in the first phase is actually encoded at the given bitrate and using the determined resolution. In contrast to the first phase, the bitstreams generated in this phase are actually stored in a database and are transmitted to users on-demand. Figure 4.17 visualizes such a cloud-based VoD service environment and shows how easily the number of encodings in the first phase can become unfeasible for deploying the reference bitrate ladder approach.

The problem definition of this section is exactly the same as that of the live application, in terms of inputs (*i.e.* sequences, features etc.) and outputs (*i.e.* predicted bitrate ladder). However, its other environmental aspects require specific algorithms for the task of bitrate ladder prediction. On one hand, as the real-time aspect is alleviated in the VoD applications,

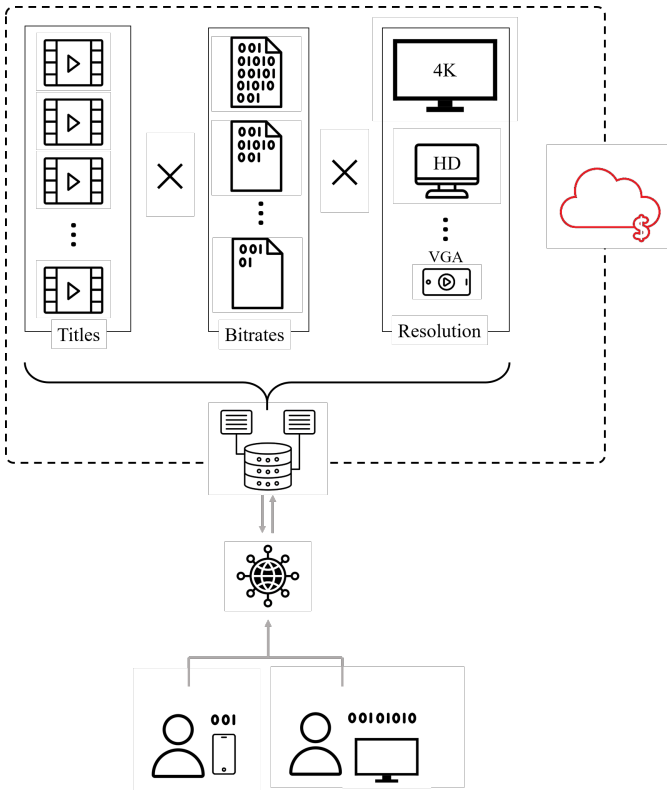


Figure 4.17 – A VoD pipeline, where the video titles are analyzed and encoded on a cloud-based server and users demand different versions of the stored titles, depending to their bandwidth constraints.

we might be allowed to conduct more encodings in the first phase in order to obtain the bitrate ladder. On the other hand, the encoding presets used in the second phase are usually way too slow to be affordable in the exhaustive analyze of the first phase.

This section proposes a new method called fast-pass bitrate ladder prediction. In this method, the reference bitrate ladder of a fast preset is used to estimate the bitrate ladder of a slow preset of the same encoder. There are two main challenges to overcome when designing an algorithm:

1. The bitrate ladders of a given sequence can be significantly different, between a fast and slow preset. This requires a transfer function that takes the ladder of the fast preset and outputs the ladder of the slow preset.
2. The transfer function in the first aspect can highly be content-dependent. In other words, a transfer that is obtained on sequence A, might terribly work on sequence B.

Figure 4.18 demonstrates how the reference bitrate ladder of the “faster” and “slow” presets of VVenC encoder can differ. This figure only focuses on the cross-point bitrates between the two resolutions 540p and 720p. To visualize the intended behaviour, in this figure, the difference between the cross-points in the two ladders of “faster” and “slow” is expressed as an arrow. As

can be seen, both amplitudes and directions of the arrows are strongly variable, which proves the content adaptivity of the problem under study.

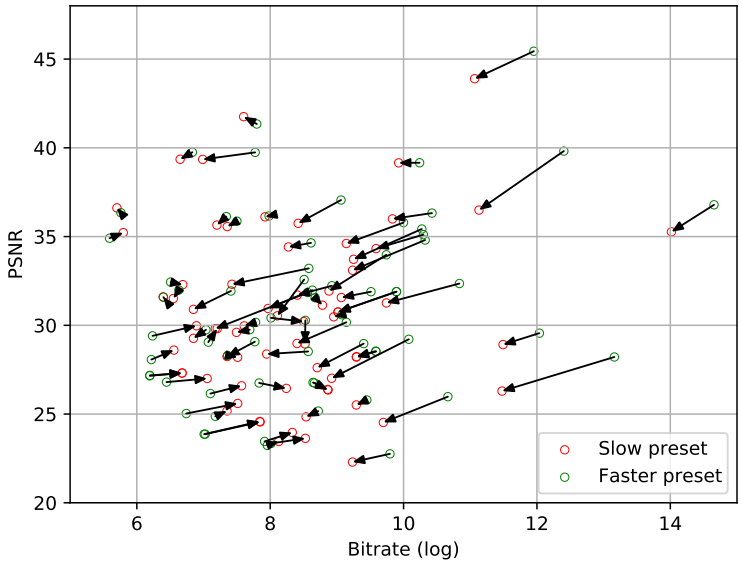


Figure 4.18 – The relative position of cross-point bitrates of the "slow" preset of VVenC, with respect to its "faster" preset. Note that only the resolution change from 540p to 720p in corresponding reference bitrate ladders are considered.

4.4.2 Fast-pass framework

Notation

The notation presented in 4.2.1 has to be slightly modified to incorporate the multi-preset aspect of the problem. To this end, the encoder E is considered to be equipped with a set of internal presets $p \in P$, allowing it to tune its global trade-off between coding efficiency performance and encoding complexity. For the sake of notation simplicity, here we suppose that this set only includes two presets as $P = \{\text{fast}, \text{slow}\}$. Therefore, the convex hull function in Eq. 4.2 can be re-written as in Eq. 4.19, where the function $C^v(r, p)$ indicates that, for the video sequence v, the best quality $(q^*(p))$ that can be obtained by encoder E at bitrate r and its preset p, when operated on available resolutions in S.

$$q^*(p) = C^v(r, p) \text{ where } E(v, r, s_i; p) \leq q^* \quad \text{for all } s_i \in S. \tag{4.19}$$

Consequently, the cross-point bitrate function in Eq. 4.3 can be modified to Eq. 4.20. Precisely, the function $B^v(s_i, p)$ returns the optimal bitrate as $r^*(p)$, in which the resolution switch from s_i to s_{i+1} should occur on sequence v, when encoded with encoder E in its preset p.

$$\begin{aligned} r_i^*(p) &= B^v(s_i, p) \text{ where} \\ C^v(r_i^*, p) &= E(v, r_i^*, s_i; p) \text{ and} \\ C^v(r_i^* + \epsilon, p) &= E(v, r_i^* + \epsilon, s_{i+1}; p). \end{aligned} \tag{4.20}$$

Finally, the bitrate ladder function in Eq. 4.4 is modified into Eq. 4.21, such that it differentiates between reference bitrate ladders of different presets. As reminder, the output $i^*(r, p)$ in this equation indicates the optimal resolution index (among possible resolutions in S) in which the video v must be encoded by E at bitrate r and in preset p.

$$i^*(r, p) = L^*_{v,S,p}(r) \text{ where } B^v(s_{i-1}, p) < r \leq B^v(s_i, p). \tag{4.21}$$

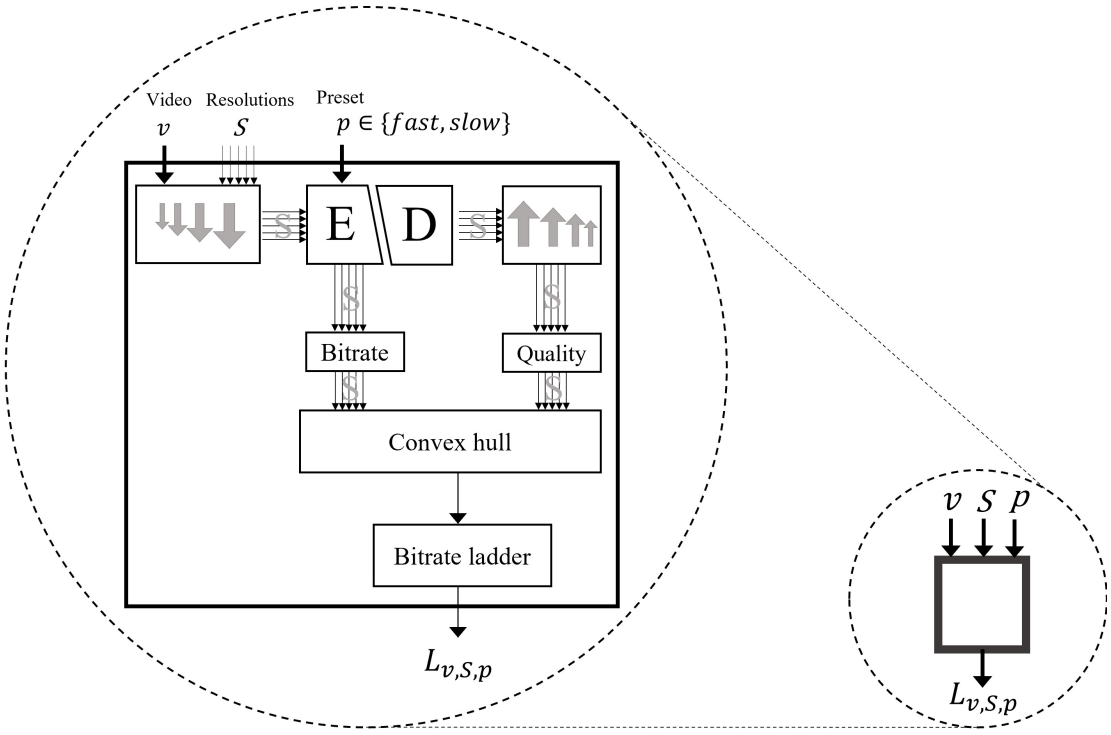


Figure 4.19 – Global functionality of a multi-preset encoder for constructing the reference bitrate ladder of a video sequence.

Figure 4.19 summarizes the overall process to calculate the reference bitrate ladder corre-

sponding to a given preset p of a multi-preset codec, where the encoder and decoder parts are denoted as E and D, respectively. In this figure, the video sequence v and available resolutions S are used as input and the output is the reference bitrate ladder $L^*_{v,S,p}$.

4.4.3 Proposed algorithm

Given the above modified notation, the high-level process of the proposed fast-pass algorithm can be seen as a function F, where the inputs are the video signal v and the reference bitrate ladder of the "fast" preset $L^*_{v,S,\text{fast}}$. While the output is a *prediction* of the the reference bitrate ladder of the "slow" preset, noted as $\hat{L}_{v,S,\text{slow}}$. This prediction process is expressed as:

$$\hat{L}_{v,S,\text{slow}} = F(v, L^*_{v,S,fast}). \tag{4.22}$$

Inside the function F, the bitrate ladder prediction problem is divided into a set of cross-point prediction sub-problems and solved independently. Each sub-problem corresponds to determining the cross-point bitrate between resolution i and $i+1$. In particular, an ML-based scheme is used to carry out a prediction from the cross-point bitrates of the "fast" preset to the equivalent cross-point bitrate in the "slow" preset. In addition to the cross-point bitrate in the "fast" preset, this ML-based method is designed such that it also receives a vector of spatio-temporal features, extracted from the signal v.

As a result, the goal is to train a model M that receives v and the reference cross-point bitrate $r^*_i(\text{fast})$, while outputting a prediction of $\hat{r}_i(\text{slow})$:

$$\hat{r}_i(\text{slow}) = M(v, r^*_i(\text{fast})) \tag{4.23}$$

Once $\hat{r}_i(\text{slow})$ values are predicted, Eq. 4.20 is estimated for $p = \text{slow}$ to obtain $\hat{B}^v(s_i, \text{slow})$. Finally, this estimated function is replaced in Eq. 4.21 to provide the predicted bitrate ladder in the preset "slow". Given the module defined in Figure 4.19, the proposed fast-pass bitrate ladder prediction can be defined as in Figure 4.20.

The key requirement of being multi-preset is met with almost all industrial encoders, while it is not supported with almost any reference software codecs. Among several available codecs, the most interesting codec for this algorithm was VVenC. Firstly, it is an implementation of the best video compression standard, with incredibly high performance in low complexity, thanks to its enormous optimized codes. Secondly, VVenC provides a wide range of presets, namely "faster", "fast", "medium", and "slow" offering a diverse range of trade-offs. Later in this section, we will also study the performance of these presets in the same proposed framework.

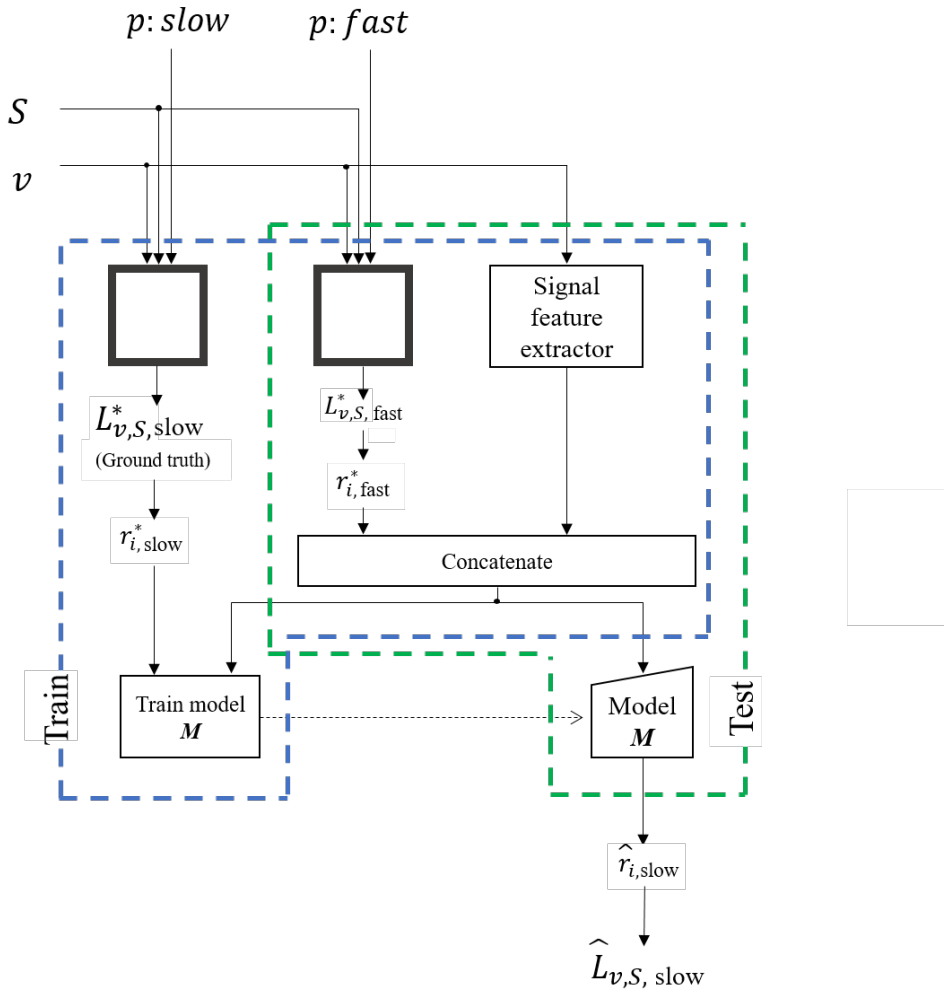


Figure 4.20 – Fast-pass bitrate ladder prediction framework.

4.4.4 Training process

The input the training process is the extracted features as described earlier. And the output is a model that can predict the bitrate ladder for the test samples. Similar to the previous algorithm, a supervised regression method is employed in order to predict the information required to construct each bitrate ladder. Initially, processing the input variables such as extracted temporal and spatial features and cross-point bitrate values of fast-pass encodings can lead to reducing the computational cost of modeling the predictor. In addition, choosing the proper kernel for modeling the predictor is essential and different kernels with different parameters should be tested. To this end, prior to the prediction of each cross-point bitrate, we applied feature selection, using RFE [130], on the set of input features. The optimal number of features for effective modeling equals twelve features including the cross-point bitrate of "faster" preset

and a selection of temporal and spatial features (table 4.1).

Three regression models are trained to predict the three cross-point bitrates. corresponding to resolution switches with four resolutions of 540p, 720p, 1080p and 2160p. For constructing the bitrate ladder, different regression models and kernels were trained and tested, such as linear model with two parameters, GP, SVM, random forest and decision tree. The GP and Decision tree had a similar performance where the Decision tree marginally outperformed when is combined with the gradient boost method. Thus, for predicting the three cross-points in the bitrate ladder, we used the decision tree with gradient boost.

4.4.5 Experimental Results

In this section, the performance results of the proposed method are presented. As the problem is very similar to that of the previous section, we have used the same measurement metrics as well as anchor methods to compare the performance. First group of metrics is related to the performance of training and prediction. In particular, R^2, MAE, explained variance, and MSE are used. The main used metric is BD-BR, since the ultimate goal in a streaming service – like any other system based on video compression – is to reduce the bitrate in the same level of quality. Similar to the method provided in the previous section, here we measure the BD-BR metric as a global variable, computed over the test dataset, given that the underlying system is using the proposed bitrate ladder prediction. And as a reference of the BD-BR computation, the same system is considered when it uses one of the anchor methods. To this end, we have used two anchor methods for determining the bitrate ladder, namely the static or "one-size-fit-all" ladder and the reference ground truth ladder. In fact, the former is an indicator of the lower-bound of the overall system performance, while the latter is considered as its upper-bound counterpart. In terms of complexity-performance trade-off, the static ladder requires no additional computation and offers relatively poor performance. On the other hand, the reference ground truth ladder is the most complex solution due to its brute-force search, however, it provides the highest possible overall performance that can be achieved.

As the VVenC codec offers several quality presets, we have presented the results accordingly [2]. In other words, each preset (except the "faster" preset that is used for the fast-pass) is associated with one of the possible settings of the proposed method, where the ladder of the given target preset is to be predicted from another preset. Namely, the target presets of the proposed settings are "fast", "medium", and "slow". In all these settings, the "faster" preset has been used as the preset from which the ladder of the target preset is predicted.

The results in Table 4.4 report the outcome of the ten-fold cross-validation with the accuracy of prediction metrics averaged over the ten folds for three examined presets. The table reports high values of R^2 and explained variance, around 0.9 for most of the cross-point bitrates. Also,

2. https://github.com/fraunhoferhhi/vvenc

the MAE and MSE are considerably low and comparable for all predicted cross-point bitrates. As it was expected, the accuracy of prediction in the "slower" preset is less than the "fast" and "medium" presets.

Table 4.4 – Average performance metrics of predicted cross-point bitrates in three presets.

Setting	Cross-over bitrate	R^2	Explained variance	MAE	MSE
Fast	B_{01}	0.97	0.96	0.22	0.08
	B_{12}	0.97	0.96	0.21	0.09
	B_{23}	0.92	0.83	0.30	0.27
Medium	B_{01}	0.95	0.97	0.19	0.06
	B_{12}	0.91	0.90	0.22	0.07
	B_{23}	0.85	0.87	0.37	0.33
Slow	B_{01}	0.96	0.97	0.23	0.12
	B_{12}	0.90	0.86	0.29	0.30
	B_{23}	0.89	0.86	0.44	0.38

Table 4.5 presents the overall BD-BR performance of the proposed method against the two anchors. In each anchor comparison, one of the two aspects of complexity and bitrate saving is important. In fact, when the proposed method is compared to the static ladder, the goal is to improve the BD-BR performance (*i.e.* smaller values) at the cost of a certain level of additional computation. As can be seen, in this sense the proposed method is significantly better than the static method in all settings. It can also be observed that the closer the target preset gets to "faster" preset, the more this performance improvement becomes. This can be justified by the fact that the similarities of ladders of the target preset is more when they are closer to the "faster" preset, hence, the prediction becomes more accurate.

Table 4.5 – Overall BD-BR performance of the proposed algorithm with respect to the ground truth and static anchors.

Setting	BD-BR vs. GT	BD-BR vs. static
Fast	0.79%	-9.44%
Medium	0.67%	-8.60%
Slow	0.88%	-8.32%

On the other hand, when the proposed method is compared to the reference ground truth method (GT in Table 4.5), the interest is to minimize the positive BD-BR value, which represents the bitrate loss in the same level of quality. In fact, the proposed method replaces the brute-force search on the target preset with another brute-force search in the "faster" preset. Hence, a loss of performance is traded with the acceleration of the ladder prediction. As it can be seen,

the proposed method introduces a bitrate loss of about 1%, which is consistent in the three settings. Similar to the previous comparison, it is observed here, too, that the proposed method slightly performs better when the target preset is closer to the "faster" preset. To complete this comparison, particularly the acceleration versus BD-BR loss, Table 4.6 shows how much acceleration the proposed method offers compared to the reference ground truth method. Each value is computed as the ratio of the total spent for the encoding jobs of the ladder computation in each method. As can be seen, the acceleration of the proposed method is 91% for the "slow" setting, which is the preset in which the VoD mezzanine contents are most likely encoded. While the BD-BR loss due to relative imprecision of the learning-based method is only 0.88%.

Table 4.6 – Run-time required to construct the bitrate ladder in different presets.

Setting	Ladder computation time of the proposed method, relative to the GT
Fast	75%
Medium	25%
Slow	9%

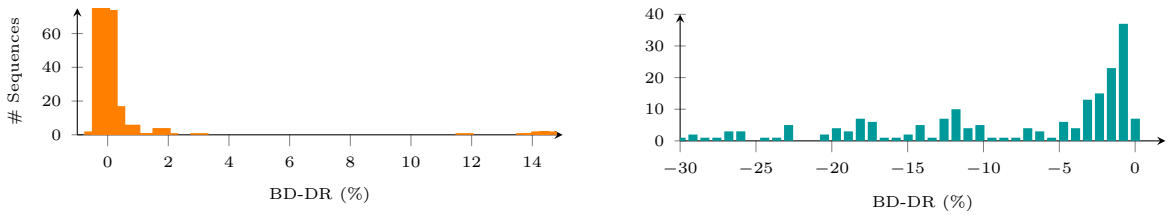
In addition, Figure 4.21 represents the same bitrate performance measurements of the proposed method in finer granularity. In this figure, the histogram of the obtained BD-BR values is provided for the three settings and against the two anchors. Each row in this figure corresponds to one of the three settings. In each row, the left histogram represents the histogram of BD-BR loss against the reference ground truth method, while the right histogram represents the BD-BR gain against the static bitrate ladder.

Finally, the proposed fast-pass method is compared against two benchmark methods:

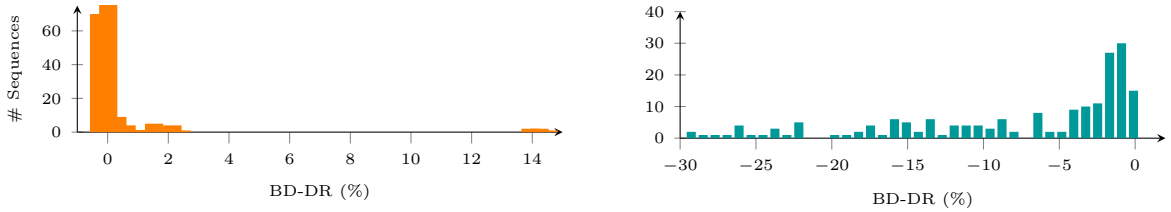
— "Faster" reference ladder: In this benchmark, we use the reference bitrate ladder of the fast-pass using the "faster" preset of the VVenC for encoding in a slower preset. The purpose of this comparison is to test how much can be lost if the bitrate ladder of an encoder is constructed with a simplified version of the same encoder. Moreover, this reference ladder information is available even with the proposed fast-pass algorithm. Therefore, one might wonder about its performance.

— Regression: This benchmark is exactly the same method as the regressor constituent method of the previous proposed method. The purpose of comparing against this benchmark is to show how much the proposed system is improving the global performance of the system, at the cost of additional fast-pass reference ladder computation as well as the ML-based inference.

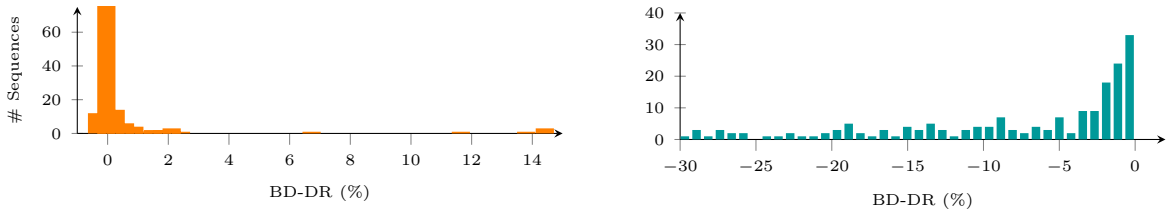
In table 4.7, we compare the performance of the proposed fast-pass bitrate ladder prediction to two above benchmark methods. These comparisons are carried out for the three settings cor-



(a) Fast



(b) Medium



(b) Slow

Figure 4.21 – Distribution of the BD-BR metrics on the test sequences. The left column presents the BD-BR metric versus the GT ladder, while the right column uses the static ladder as reference.

responding to the “Fast”, “Medium” and “Slow” presets. Moreover, the anchor ladder for this test is the ground truth (GT) ladder of the given setting. Therefore, positive numbers represent BD-BR loss. As can be seen, the proposed fast-pass method outperforms both benchmark methods, by resulting smaller BD-BR loss compared to the GT ladder. This performance also verifies the initial assumption of this section, that using a ladder that is constructed based on another preset (typically a faster preset) is not a good idea and it results in performance drop.

4.5 Conclusion

This chapter proposes two ML-based method for predicting the bitrate ladder in adaptive streaming use-cases. The first proposed method targets live video delivery applications, where the constraint is on the complexity of the sender side ladder computation, while the second method targets VoD applications where the constraint is on the number of cloud-based encodings with a slow preset of the given codec.

Table 4.7 – The performance of the proposed fast-pass algorithm compared to two benchmark methods. The numbers are in terms of BD-BR loss compared to the reference GT ladder of given settings.

Setting	Fast-pass	"Faster" reference ladder	Regression
Fast	0.79%	2.03%	1.06%
Medium	0.67%	1.88%	1.20%
Slow	0.88%	2.03%	1.06%

In both methods, a set of spatio-temporal feature is extracted from each sequence, in order to learn a ladder as close as possible to their ground truth bitrate ladder. The first proposed method fits two supervised ML-based methods on the extracted features, and applies an ensemble aggregation method to improve the performance of the two constituent ML-based methods. In the second proposed method, a fast-pass method is designed that runs a pre-encoding pass using the same codec, but in a faster preset than the actual encoding pass. From this pass, a ground truth bitrate ladder is produced and used for prediction of the ground truth bitrate ladder in the slow preset of the given codec in which the actual encoding jobs are run. In both methods, the performance of the proposed solution is assessed using a static and fully customized ground truth bitrate ladder as the benchmark method.

Conclusion and perspectives

The problem of compressed video delivery under strict bitrate constraints was studied. In this journey, the problem have been taken into account from different point of views for addressing different aspects. This manuscript reports most interesting contributions that were made during this thesis. Before providing technical details of proposed solution, we first presented adequate context elements in the first two chapters. In particular, a high-level, yet thorough introduction has been provided in Chapter 1. In this chapter, the basic building blocks of video compression system are discussed. This introduction gives insights to readers both about what happens inside a black-box video encoder, as well as how it is used in an actual video delivery system. Once the basics are covered, Chapter 2 presents a state-of-the-art of the problems that we have addressed in the rest of the paper. In each part, the goal is to first give an insight about the problem definition and then present existing solutions that currently can be found in the literature.

The contributions of this thesis are categorized and presented into two parts: quality enhancement and bitrate ladder prediction. Both problems are based on the same contextual background presented in Chapter 1, while dealing with different challenges. The solutions presented in both parts, share two aspects. First, they use VVC as the underlying video codec. Second, the proposed solution somehow benefits from ML-based techniques to address the problem.

Quality enhancement

In the first track of this thesis, the problem of enhancing quality of degraded video was studied. The specific types of degradation that we targeted were, in general, compression artifacts due to lack of adequate bandwidth. These artifacts include, blockiness, blurriness, ringing, banding and so on. However, without discriminating between them, we aimed at improving them at the same time, using a particular ML-based technique, called CNN.

The main contribution in this track was the use of coding information in the process of quality enhancement. The motivation behind this contribution was the fact that all compression artifacts are product of decisions that are made within the underlying video encoder. Therefore, one can imagine that if an automatic ML-based mechanism – such as CNN – is provided with this decision information during its training and as its input, it might result in a better understanding of the source of artifacts, as well as their proper enhancement scheme. To realize this idea, we tried and used different types of coding information, most notably prediction information, partitioning and quantization parameter.

The above idea of bewaring CNN with certain coding information was implemented in two domains in this thesis: post-processing and in-loop filtering. In the post-processing domain, the whole CNN-based QE step is carried out merely at the decoder side, leaving the encoder complexity unchanged. This is in contrast with the in-loop filtering domain, where the encoder side also implements the CNN-based QE algorithm, in order to use the enhanced video frames as reference for temporal prediction, hence further improving the performance. In both domains, the experiments show that all of these coding information are helpful, in terms of the overall quality improvement that their algorithm can offer on low bitrate coded videos.

Bitrate ladder prediction

The second track of this thesis attempts to address the low bitrate video coding problem in a higher level and by maximizing the efficiency of an entire video delivery ecosystem. Precisely, in this track we assume that the video encoder is a black-box and we have no control over it, but simple parameters such as encoding bitrate and operating preset. However, this encoder is deployed in an end-to-end video delivery system, on which an algorithm have freedom to determine which resolution of a naively 4K sequence should be encoded and sent to receivers.

The main contribution of this track is again the use of ML-based methods to address the problem. Unlike the first track where the entire learning was left to a set of convolutional layers, here we tend to use simpler ML-based methods such as regression and classification with certain handcrafted feature vectors. These features are analyzed and selected from a larger set of spatio-temporal descriptors.

Two main use-cases have been targeted by the bitrate ladder prediction track: live video delivery and VoD/streaming services. Each use-case has its own challenges. The live video delivery use-case has constraints on the senders' bandwidth, while the VoD/streaming use-cases are more limited by the receivers' bandwidth. In this thesis, one algorithm is proposed for each use-case, taking into account its special constraints and limitations. In particular, the proposed algorithm for the live use-case uses an method based on ensemble learning for predicting the bitrate ladder of the real-time video sender. In this algorithm, the processing time is kept under control in order to cope with the live aspect of the use-case. Moreover, the method for the VoD/streaming use-cases proposes a framework which allows running a series of low-complexity encoding jobs, on which a regression method is applied and predicts the bitrate ladder of the final high-complexity encoding jobs.

What is next?

In both tracks, the industry is advancing fast and the topics are becoming more and more interesting. However, due to the limitation of time, a handful of ideas in both domains have been left as future work. In this section, we briefly discuss some of these ideas.

Quality enhancement

Multitask network with pixel revert map

One of the interesting observations in the experiments of this chapter was that pixels are not equally improved by any CNN-based QE method. In other words, once a trained network is applied for post-processing of a distorted image, it will change the intensity values of almost all pixels. For a changed pixel, there are two possibilities; either the change is in the direction of improvement, or it is in the direction of becoming even more distorted. Our observation showed that the number of pixels which change in undesired direction are usually significant.

Given the above context, one future work can be somehow identifying such pixels and revert the impact of CNN-based post-processing on them. For instance, a multi-task CNN could be trained that in addition to overally enhances the input, it also generates a binary map, called pixel revert map. This map has the same size as the image that indicates which pixels should revert the post-processing impact.

The only potential hazard in such idea is the prediction error in the revert map. Precisely, if the multi-task network makes a mistake in identifying revert pixels, it will revert post-processing of pixels that have actually been improved by the post-processing. And this will negatively impact the overal PSNR.

End-to-end training

As discussed in Section 3.4.4, the in-loop implementation of CNN-based QE suffers from multi-enhancement phenomenon. And as was discussed, the intuitive solution to this problem is the so-called end-to-end training. To do so, one can categorize frames in terms of number of consecutive CNN-based enhancement their references have gone through.

Let image A be an inter-P image that is referring only to the intra image I of the GoP, where image I has also been enhanced by the CNN-based method. As the image I is not referring to any other enhanced image, then image A can go to category-1. Moreover, let image B be an inter-B image that is referring both image I and image A. However, in this case, since the reference A is enhanced and also refers to another enhanced reference (*i.e.* image I), there would be two consecutive CNN-based enhancement in the path to this image. Therefore, image B can go to category-2. Likewise, if we find an image in the GoP that is referring to image B, it would go

into at least category-3, unless it has higher category references. One can continue this process until all images in the GoP are categorized.

Once categorized, the end-to-end training process can be implemented in an iterative loop. First, one can train a network model, say M_0, for category-0 (*i.e.* intra images). Once trained, a new dataset of compressed videos (distorted) can be generated where the intra images have been enhanced with the M_0 as the CNN-based in-loop filter. This dataset can then serve for training M_1 that will enhance only images in category-1. By continuing these iterations, one can specialize the same network architecture for different categories and potentially address the multi-enhancement issue.

Residual enhancement

Residual information is the most interesting candidate to be tested in a residual-aware QE framework. As the pixel-domain residual signal is the subtraction of the prediction signal from the reconstructed signal, it will not likely improve the performance compared to the prediction-aware QE framework, proposed in this thesis. However, one can test transform-domain residual information in the form of quantized coefficients.

Precisely, the idea of transform-domain residual-aware QE is to use the dequantized coefficients as input, and train a network that produces an enhanced version of the dequantized coefficients which are closer to the non-quantized coefficients. The realization of this idea is better to in the block-level, as the boundary information of blocks in terms of transform coefficients is not necessarily informative.

The main known challenge for this idea is to apply the convolution step in a meaningful manner. In particular, the 2D sliding window of a CNN might not efficiently capture the texture information, since the signal is represented in the transform domain. Therefore, one can re-order the coefficients in a array (using the diagonal scan order) and apply a 1D convolution.

Spatial in-loop filtering

If the above idea works, one can integrate it directly within the decision loop of an encoder. For instance, once the best decision of a block is made (*i.e.* size, prediction mode, reference *etc.*), the final residual can be enhanced using the trained model. By doing so, neighboring blocks of such enhanced block will benefit from a higher quality reference and this might further improve the overall rate-distortion performance of the codec.

ILF implementation and existing filters

The interaction with existing filters, notably ALF, glssao and DBF can be further studied to answer certain questions. There are mainly two questions to answer in this regard: 1) where

is the best relative position of a CNN-based QE module with respect to these filters? 2) Should a CNN-based ILF co-exist with all above filters or it can replace them? At the time of writing this manuscript, these aspects are currently under study in the JEVT group.

Subjective assessment and saliency-aware QE

It is left as future work to study how the objective gains of CNN-based QE methods correlate to their subjective gain. However, in the experiments of this thesis, it was assessed that despite significant BD-BR performance improvement, there have been cases in which visually important regions of image a distorted. The most common example is the compression of human face. What has been observed was that in some very low bitrates scenarios, the impact of CNN-based QE on the faces in the video was unwatchable and highly artificial.

It is important to note that these faces are also highly distorted in the non-enhanced video. However, probably since viewers' eyes are used to such types of distortions in low bitrate, they do not find it unwatchable. However, this is just a theory and in order to find the actual answer, one can conduct subjective quality assessment viewings.

Bitrate ladder construction

Short look-ahead for low-latency

In the first part (live application), one of the tests one can do is to figure out whether it is possible to reduce the latency while maintaining the same level of performance. Precisely, the proposed method for this part extracts features from the whole GoP in order to determine its bitrate ladder. In certain scenarios, this might impose an additional latency of up to one GoP to the end-to-end live latency. Therefore, it might be interesting to test how a subset of frames from the beginning of the GoP performs, when used in the same ML-based pipeline.

Study on the impact of codecs

The research work conducted in this thesis was merely based on the state-of-the-art VVC standard. However, other codecs currently co-exist in the video transport ecosystem. Therefore, one might wonder whether the proposed bitrate ladder prediction methods of this thesis are codec-agnostic or they must re-train on different codecs. Moreover, if the problem requires per-codec implementation, is there any solution to train multi-codec ML-based bitrate ladder prediction methods?

Appendix A

Impact of Training Data in SR-based Video Coding

Introduction

With evergrowing applications of video transmission, the task of retaining a high quality displayed video under the network limitations has recently become more trendy. On the one hand, compressed videos are sent under stricter bandwidth constraints which limits the amount of transmitted information and makes the compression artifacts such as blurriness or blockiness more evident in received videos. On the other hand, receiver devices are usually powerful enough to afford complex post-processing steps to perform texture restoration. Therefore, there is a chance to achieve video quality levels that are currently inaccessible in specific low bitrate applications, such as telesurgery, monitoring systems [132, 133], *etc.*

A key to this goal is to adopt artificial intelligence techniques to learn compression loss patterns. In particular, the CNN based SR algorithms properly fit the requirements of the compression artifact restoration task [134]. The SR algorithms aim at generating a high-resolution signal from a given low-resolution one. In its basic form, the large amount of missing information in the low-resolution images makes generating the high-resolution image challenging. However, a variety of advanced SR methods are proposed to overcome this problem [135, 136, 137, 138, 139]. Particularly, CNN-based algorithms have shown impressive performance compared to traditional methods [140, 141, 142, 143].

Despite the potentials, the use of SR methods on compressed videos is sparsely studied in the literature [144, 145, 146]. Particularly that, recently, video transmission methods involving sub-sampling input signal have become very popular. For instance, a standardization activity is currently ongoing to release a video codec, called Low Complexity Essential Video Coding (LCEVC), which addresses the same issue by down-sampling, coding and transmitting metadata [147]. Also, AV1 codec also adopts resolution adaptation at both encoder and decoder with pre-defined up-sampling filters [148]. Finally, the concept of Reference Picture Re-sampling (RPR), which has recently been adopted for VVC, benefits from a similar methodology [149].

In this study, first a general framework for integrating SR methods within a coding system is described. Then, for a set of selected SR methods, the impact of training with compressed dataset is compared. In particular, the objective of this chapter is to demonstrate how differently CNN-based SR methods perform on reconstructed video signals when they are trained with compressed or uncompressed datasets. The importance of this work is perceiving the image super resolution subject rather from a video coding point of view for exploring its potentials.

In this chapter, first, an SR-based video transmission framework is described to integrate the selected SR methods along with the VVC codec. Next, we describe the characteristics of the experiments and we present the details of performance evaluation with discussions and finally, we conclude this chapter.

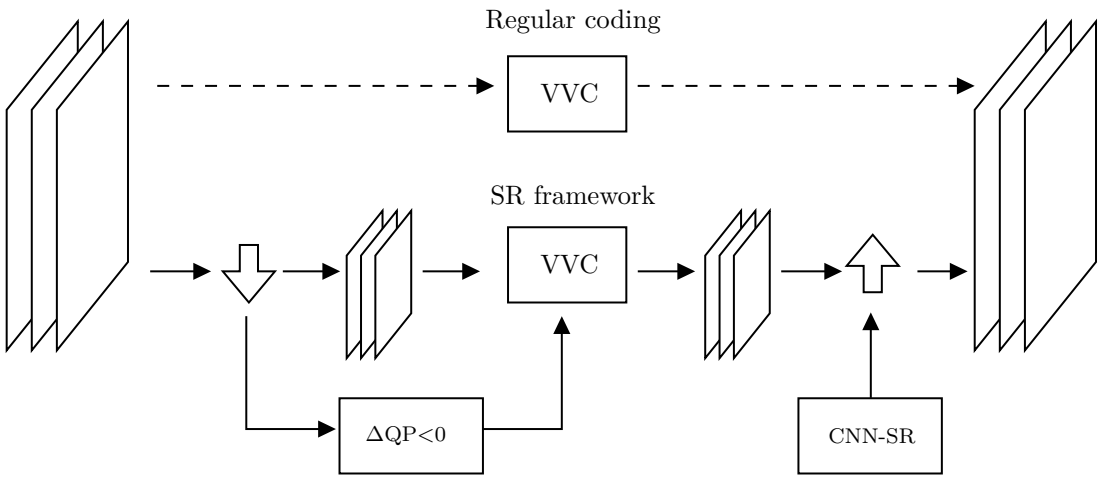


Figure A.1 – Comparison of the regular coding scheme and CNN-based SR framework

SR-based Video Coding Framework

A general framework for low bitrate video transmission using SR is described. This configurable framework is used by some broadcasters to adapt their content to low bandwidth and/or low complexity constraints. There are three main steps in the SR framework: first, the input sequence is down-sampled prior to the coding. Second, the down-sampled sequence is coded with VVC, using an adjusted QP. Third, an up-sampling step is performed on the reconstructed image using an SR method. Once all the above three steps are performed, the output is comparable to the regular coding scheme, where no down/up-sampling steps are used. Fig. A.1 compares the two coding schemes. In addition to possible coding efficiency gains in the low bitrate range, experiments of this study show that the use of the SR framework saves between 40% to 80% encoding time.

QP adjustment with respect to down-sampling factor

The QP is a mean to apply user-specified level of distortion to the compression and has two main functionalities in a codec: 1) determining the quantization step size of residual coefficients, and 2) making a trade-off between rate and distortion of different coding decisions. The combination of these two roles guarantees that under a given rate constraint, the distortion per pixel will remain below a threshold [150]. The fact that the SR framework reduces the resolution of the input sequence should not impact this distortion. Therefore, to accommodate a lower resolution, a QP adjustment parameter ΔQP<0 is added to the input QP value to apply a finer quantization. The principle of computing ΔQP as a function of the scale factor, described in [151], is adopted in the current work. Based on this method, a QP adjustment of ΔQP = -6 is applied for the used scale factor 2 on width and height.

Scope of the performance

The SR framework of Fig. A.1 and its internal modules are flexible in terms of functionality. More precisely, one can adjust the following settings depending on target application:

— Scale factor

— SR method for up-sampling

— Training dataset in case of CNN-based SR

The use of the SR framework becomes justifiable when properly tuned. For more efficient deployment, one should first understand when this framework can be beneficial. In terms of rate-distortion-complexity measurement, the SR framework can potentially have the following impacts:

— Rate:

  — Rate per sample may increase, since the QP adjustment causes finer residual quantization.

  — Rate per frame may decrease, since the down-sampling step reduces the number of coded samples in each frame.

— Distortion:

  — Distortion per frame may increase, since the down-sampling step throws away majority of samples, and this information loss may not be fully retrieved by the up-sampling step.

  — Distortion per frame may decrease, since the up-sampling step, in particular the CNN-based ones, are supposedly smart and able to retrieve a high amount of the lost information.

  — Distortion per sample may decrease, since a finer quantizer is applied on the down-sampled input.

— Complexity:

  — May decrease at the encoder side, since the number of coded samples is reduced due to the down-sampling.

  — May increase at the decoder side, since the up sampling modules are added.

The overall trade-off between all above impacts determines whether or not a specific setting of the SR framework provides desired bandwidth saving and/or complexity reduction over the regular coding scheme. In other words, the use of the SR framework is preferable when the combination of the rate-distortion-complexity results in a better global performance.

Problem statement

In this study, we investigate the impact of alternative training for SR methods. As only the training phase is impacted, the procedure of the SR framework will be identical until the up-sampling step. In other words, the only factor determining the performance of the SR framework will be the efficiency of the trained network applied for up-sampling. More precisely, we test a hypothesis: given that a CNN-based SR method is to be used for up-sampling the decoded sequences, involving compression artifacts in the training process of the SR method will improve its up-sampling performance.

The assumption is that observing compression artifacts during the training phase helps SR methods differentiate those artifacts from actual texture information during inference. Fig. A.2 presents examples of actual texture information and compression artifacts.

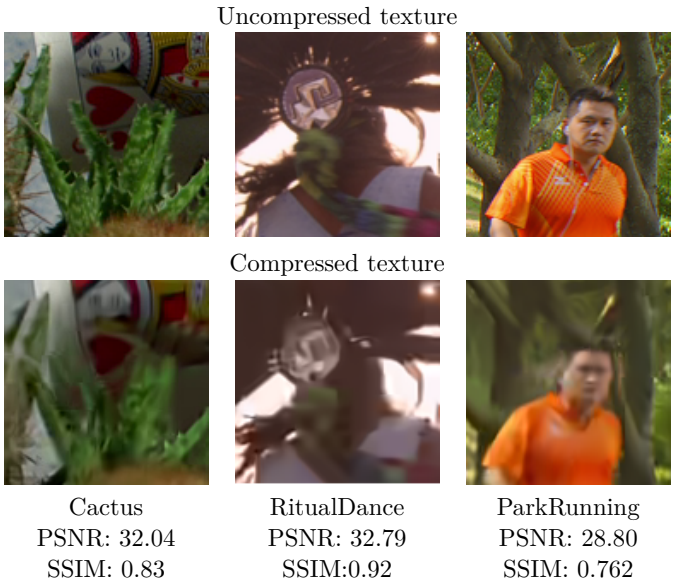


Figure A.2 – Compression artifacts (*e.g.* blockiness, blurriness) in textures coded at very low bitrate (*i.e.* 100-500kbps for 1920×1080p sequence).

Selected CNN-based SR methods

Two CNN-based SR methods are selected. These methods represent relatively simple and complex CNN architectures for SR. It is important to note that the selected methods are not supposed to be compared to each other in terms of performance. Instead, the goal is to compare each method with itself, under different training conditions.

Efficient Sub-Pel Convolutional Neural Network (ESPCN)

ESPCN is composed of three CNN layers [141]. The two first layers are used for the feature maps extraction and the last layer, which is a sub-pixel convolutional layer, is responsible for aggregating the feature maps from low-resolution space and constructing the high-resolution image. Using the sub-pixel convolutional layer, as an up-sampler in the last layer, decreases the computational time and increases the network flexibility in learning different down-sampling kernels. For the training, ESPCN uses L2 loss which maximizes PSNR. The design of this method is considered as a relatively simple network architecture.

Enhanced Deep Super Resolution Network (EDSR)

EDSR uses a residual network architecture for solving the SR problem [142]. In this network, the original architecture of the ResNet [152, 153] has been modified to increase the performance for this specific task. The modifications make the network lighter to be trained and to capture the proper features for constructing the best super resolved image. Experimental results show that EDSR outperforms most state-of-the-art SR networks. Moreover, the use of residual-based building blocks enables EDSR to learn missing high frequency information in different scaling factors. EDSR uses L1 loss for training, which gives better convergence than L2 loss. Compared to the ESPCN network, the architecture of the EDSR network is noticeably more complex.

Experiments description

SR framework setting

Test-train sequences: The experiments of this study are focused on full HD video sequences with sample resolution of 1920×1080. As training sequence set, we used the DIV2K[154] and Flickr2K datasets. The test sequence set is composed of 10 sequences from the CTC of JVET and JCT-VC [155]. In order to further extend the list of test sequences, five UHD sequences from the CTC are also down-sampled into HD and used. Fig. A.3 quantifies the motion and texture characteristics of these test sequences. For this purpose, SI and TI are used [156], where higher values indicates more complex texture and motion characteristics, respectively.

All sets have been carefully selected so that they represent adequate level of diversity in terms of motion and texture properties.

Down-sampling: The down-sampling step of the SR framework is shared between the training and test phases. In both cases, the bicubic filter, implemented in FFMPEG, has been used. As we only focus on 1920× 1080 resolution in the conducted experiments, only the scaling factor of 2 is used for down-sampling.

Coding schemes: Four coding schemes have been compared. The two CNN-based SR methods

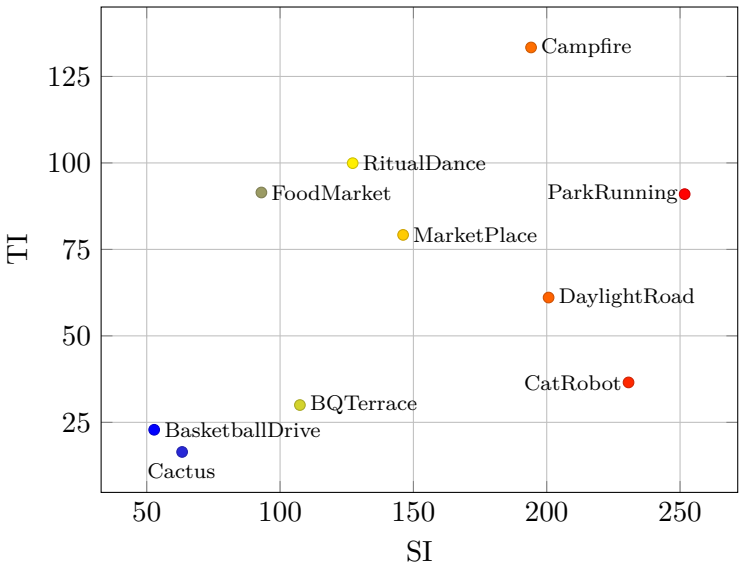


Figure A.3 – Spatial Index (SI) and Temporal Index (TI) of test sequences.

presented in Section "Selected CNN-based SR methods"; the non-CNN bicubic SR scheme; and the VTM coding scheme in the context of regular coding of Fig. A.1. The two latter schemes, the bicubic and VTM, are served as anchors for the former CNN-based SR methods.

Training of CNN-based SR methods

Ground truth: In the SR framework, the ultimate goal is to be as similar as possible to the original high-resolution sequence. Therefore, this signal is used as the ground-truth for both training settings.

Down-sampled dataset: Two down-sampled training datasets are used: uncompressed and compressed. In the uncompressed setting, the training data is simply provided by down-sampling the original sequences. While, in the compressed setting, the same down-sampled sequences are compressed prior to the training.

Coding artifacts: To produce the training dataset of the compressed setting, the VTM-5 has been used. It was assessed that the QP-independent training of the CNN-based SR methods results in a poor performance. This is due to the fact that coding artifacts have various characteristics in different bitrates. Therefore, we divided the QP range of 22-63 into 6 equal intervals, as a compromise between the performance and number of networks. For each interval, the QP value in the middle has been selected to learn the compression artifacts of that interval. This design choice results in six trained networks for each CNN-based SR scheme.

It is critical to emphasize that uncompressed sequences are also used for training in the compressed setting. In other words, the training data of the compressed setting is a super-set of training data in the uncompressed setting. The assumption is that by combining the compressed

and uncompressed, networks can learn both true textures and artifact edges.

Results

Coding efficiency performance

Table A.1 and A.2 presents the performance of various coding schemes and their different settings. In order to conduct performance comparisons, two metrics are used:

1) BD-BR: This metric is computed between different SR methods. The negative BD-BR value is interpreted as the percentage of bitrate saving in the same level of quality based on PSNR [157]. (Table A.1)

2) Critical bitrate: This term is used to denote the maximum bitrate of a sequence where the use of the SR framework still outperforms the regular coding [158]. Obviously, the larger values of critical bitrate indicate that the use of the SR framework can be justified in a wider range of applications (Table A.2). For a better comparison, Fig. A.4 shows the R-D curves of the EDSR method for a selection of test sequences, with their critical bitrate.

Table A.1 – Performance of ESPCN and EDSR methods trained with compressed and uncompressed datasets. Bitrate saving values of the compressed setting, presented in terms of BD-BR (%), are calculated against the bicubic SR method and the uncompressed setting.

	CNN-based SR method					
	ESPCN			EDSR		
Sequence	Uncom. vs. Bicubic	Com. vs. Bicubic	Com. vs. Uncom.	Uncom. vs. Bicubic	Com. vs. Bicubic	Com. vs. Uncom.
BasketballD.	-43%	-46%	-3%	-50%	-59%	-9%
BQTerrace	-19%	-24%	-5%	-26%	-33%	-7%
Cactus	-11%	-14%	-3%	-14%	-22%	-8%
CampFire	-2%	-5%	-3%	-19%	-23%	-4%
CatRobot	-8%	-11%	-3%	-16%	-24%	-8%
DayLight	-5%	-8%	-3%	-9%	-15%	-6%
FoodMarket	+6%	+4%	-2%	-2%	-6%	-4%
MarketPlace	0%	-3%	-3%	-7%	-12%	-5%
ParkRunning	-5%	-8%	-3%	-12%	-16%	-4%
RitualDance	+2%	-1%	-3%	-12%	-17%	-5%
Average	**-8.5%**	**-11.6%**	**-3.1%**	**-16.7%**	**-22.7%**	**-6%**

Table A.2 – Performance of ESPCN and EDSR methods trained with compressed and uncompressed datasets.The critical bitrates of compressed and uncompressed settings are computed against the VTM and presented in terms of "kbps".

	CNN-based SR method			
	ESPCN		EDSR	
Sequence	Uncompressed	Compressed	Uncompressed	Compressed
BasketballDrive	150 kbps	175 kbps	240 kbps	380 kbps
BQTerrace	90 kbps	102 kbps	110 kbps	165 kbps
Cactus	180 kbps	200 kbps	180 kbps	320 kbps
CampFire	490 kbps	630 kbps	2650 kbps	3700 kbps
CatRobot	170 kbps	190 kbps	200 kbps	455 kbps
DayLight	240 kbps	290 kbps	260 kbps	371 kbps
FoodMarket	570 kbps	730 kbps	1850 kbps	2000 kbps
MarketPlace	350 kbps	430 kbps	540 kbps	963 kbps
ParkRunning	1000 kbps	1250 kbps	3200 kbps	4550 kbps
RitualDance	450 kbps	600 kbps	1400 kbps	2471 kbps
Average	**293 kbps**	**460 kbps**	**1063 kbps**	**1538 kbps**

Observations and discussions

Training set

With no exception, the use of compressed training set outperforms the uncompressed one. This is reflected in three aspects. First, there are coherent BD-BR gains with the compressed setting compared to the uncompressed setting, which are -3.1% and -6% for the ESPCN and EDSR methods, respectively. Second, the critical bitrate of the SR framework significantly moves towards the higher bitrates, when the compressed setting is used instead. Third, it was assessed that the amount of BD-BR gain due to the compressed setting seems to be consistent and content-independent.

The QP-dependent network training in the compressed setting is critical. As mentioned earlier, the preliminary experiments of this study showed that when the compressed setting was trained with a dataset composed of all range of QPs, the results are significantly worse than the QP specific compressed setting. This means that the statistics of coding artifacts vary in different ranges of low bitrate. Therefore, in order to be able to restore low bitrate artifacts, one should expose the CNN learning to the appropriate training samples, representing the right type of artifacts. In conclusion, all these evidences show that the proper use of compressed training set significantly improves the performance of the SR framework.

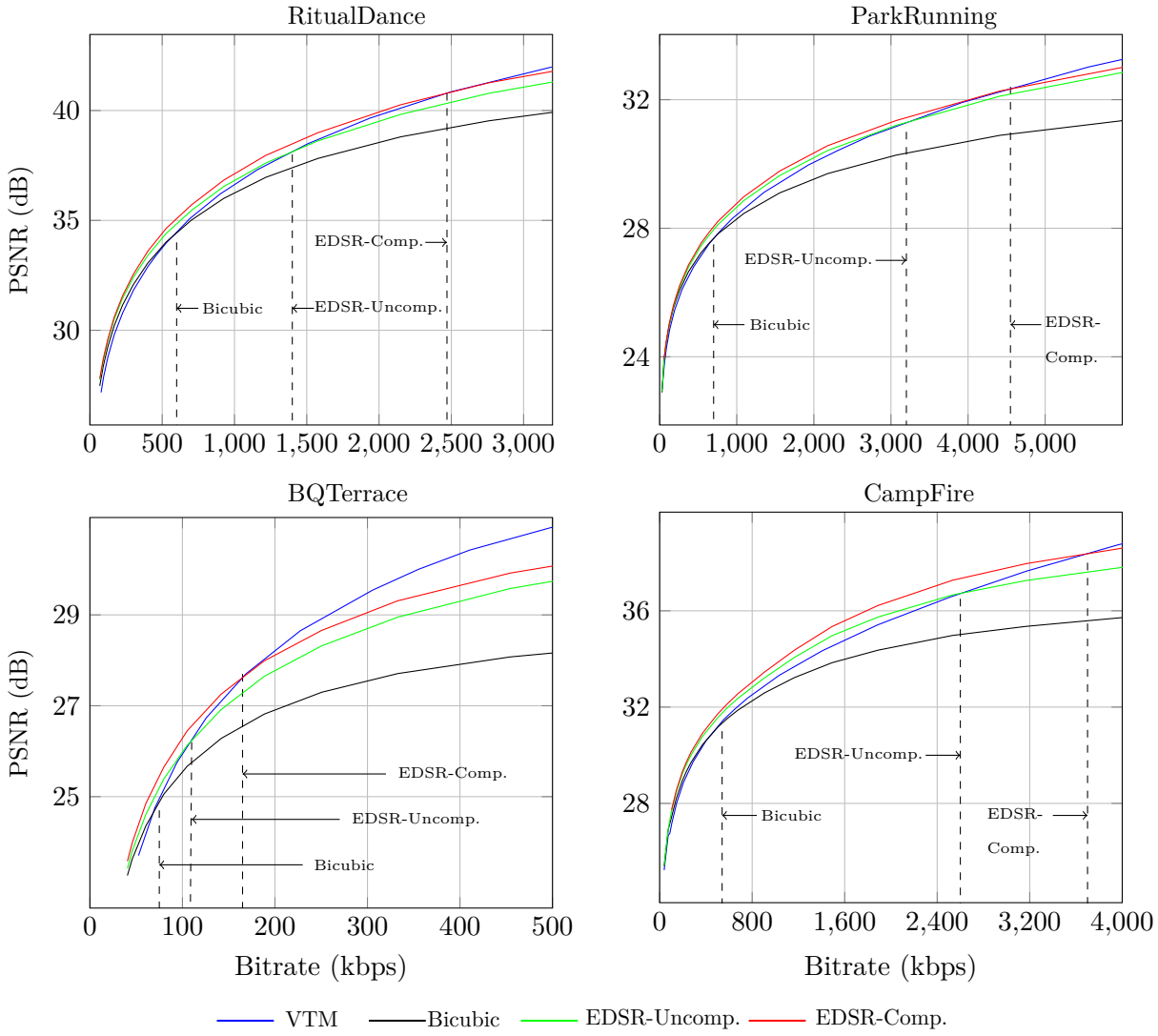


Figure A.4 – R-D curves corresponding to different coding schemes: EDSR in the uncompressed and compressed settings, the bicubic and the VTM. In order to clarify the improvement due to the use of compressed training set, the critical bitrates with respect to the VTM are shown with dashed lines.

The SR framework

The performance of the SR framework degrades in higher bitrates. The critical bitrates metrics of Fig. A.4 properly demonstrate this fact. This figure shows that after certain bitrate, the SR framework becomes significantly poorer than the regular coding with VTM. One possible reason for this behavior is the nature of artifacts that are specific for very low bitrates, (*e.g.* blockiness and blurriness). More precisely, restoring these artifacts might be suitable for neural-network based solutions, while avoiding them at these bitrates is very difficult for the VTM. It

is also asserted that the performance of the SR framework is highly content-dependent. As can be seen in Table A.2, in some sequences such as ParkRunning, CampFire and RitualDance, the use of compressed train data moves the critical bitrate about 1Mbps. According to Fig. A.3, all these sequences have relatively complex spatial and temporal characteristics.

SR methods

The BD-BR improvement of using compressed training set with EDSR is significantly larger than that of ESPCN . As mentioned earlier, the ESPCN architecture is relatively simpler than EDSR. Therefore, this result loosely concludes that simple network structures might not be powerful enough to differentiate between compression artifacts and actual texture information during training. Testing this hypothesis with more network examples is left as future work. Another observation is that the sequences with the highest improvement due to the use of compressed dataset, are BQTerrace, Cactus and CatRobot. According to Fig. A.3, all these sequences have relatively low temporal complexity. Interestingly, in all three sequences, the performance of the SR framework against the VTM anchor is among the poorest ones. The interpretation can be that when the SR framework performs poorly compared to the VTM, the use of compressed dataset can make a bigger change. Finally, the results show that the CNN-based SR methods do not necessarily perform better than simple SR methods, in all sequences. Examples like FoodMarket, RitualDance and MarketPlace, where the bicubic method outperforms ESPCN, prove that a bad choice of CNN-based SR method can easily deteriorate the SR framework.

Conclusion

In this chapter, the impact of adding compressed videos to the training set for CNN-based SR methods has been investigated. A coding framework is introduced in which different SR methods can serve for up-sampling. It was assessed that training CNN-based SR methods compressed training set significantly outperforms uncompressed training sets. This impact improves the global coding efficiency of the SR framework and justifies its use in a wider range of bitrates. Furthermore, it was observed that to boost the performance increase of using compressed training set, complex network architectures are preferred over simple ones, since they are more capable of learning common coding artifacts in low bitrates.

Appendix B

List of Publications

Conference papers

1. **F. Nasiri**, W. Hamidouche, L. Morin, G. Cocherel, N. Dhollande, "A study on the impact of training data in CNN-based super-resolution for low bitrate end-to-end video coding", Tenth IEEE International Conference on Image Processing Theory, Tools and Applications (IEEE IPTA), Paris, France, 2020.
2. **F. Nasiri**, W. Hamidouche, L. Morin, N. Dhollande, G. Cocherel, "Prediction-aware quality enhancement of VVC using CNN", International Conference on Visual Communications and Image Processing (IEEE VCIP), Hong Kong, 2020.
3. **F. Nasiri**, W. Hamidouche, L. Morin, N. Dhollande, G. Cocherel, "Model Selection CNN-based VVC Quality Enhancement [1]", The 35th Picture Coding Symposium (IEEE-PCS), Bristol, UK, 2021.
4. **F. Nasiri**, W. Hamidouche, L. Morin, N. Dhollande, J.Y. Aubie, "Ensemble Learning for Efficient VVC Bitrate Ladder Prediction", *submitted to* 10th European Workshop on Visual Information Processing (EUVIP), Lisbon, Portugal, 2022.
5. **F. Nasiri**, W. Hamidouche, L. Morin, N. Dhollande, J.Y. Aubie, "Multi-Preset Video Encoder Bitrate Ladder Prediction", *submitted to* 1st ACM Multimedia Workshop on Artificial Intelligence for Live Video Streaming, Lisbon, Portugal, 2022.

Journal papers

1. **F. Nasiri**, W. Hamidouche, L. Morin, N. Dhollande, G. Cocherel, "A CNN-based Prediction-Aware Quality Enhancement Framework for VVC", IEEE Open Journal of Signal Processing. 2021.

1. Awarded as Top-10 paper

BIBLIOGRAPHY

[1] C.-M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsai, C.-W. Hsu, S.-M. Lei, J.-H. Park, and W.-J. Han, "Sample adaptive offset in the hevc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1755–1764, 2012.

[2] Adaptive Loop Filter (ALF). https://medium.com/vicuesoft-techblog/adaptive-loop-filter-a49fe3d3f733/.

[3] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over http dataset," in *Proceedings of the 3rd multimedia systems conference*, 2012, pp. 89–94.

[4] M. Wien, *High efficiency video coding, Coding Tools and specification.* Springer, 2015.

[5] A. Arrufat, "Multiple transforms for video coding," Ph.D. dissertation, PhD Thesis, INSA Rennes, 2015.

[6] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec*, vol. 4, no. 142-163, p. 1, 1959.

[7] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Van der Auwera, "Hevc deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1746–1754, 2012.

[8] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers (speech coding)," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, 1988.

[9] G. Bjontegaard, "Improvement of BD-PSNR model," *Document VCEG-AI11*, Berlin, Germany, July 2008.

[10] T. Daede, A. Norkin, and I. Brailovskiy, "Video codec testing and quality measurement," *draft-ietf-netvc-testing-08 (work in progress)*, p. 23, 2019.

[11] W.-S. Park and M. Kim, "CNN-based in-loop filtering for coding efficiency improvement," in *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*. IEEE, 2016, pp. 1–5.

[12] T. Wang, M. Chen, and H. Chao, "A novel deep learning-based method of improving coding efficiency from the decoder-end for HEVC," in *2017 Data Compression Conference (DCC)*. IEEE, 2017, pp. 410–419.

[13] Y. Dai, D. Liu, and F. Wu, "A convolutional neural network approach for post-processing in HEVC intra coding," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 28–39.

[14] H. Zhao, M. He, G. Teng, X. Shang, G. Wang, and Y. Feng, "A CNN-based post-processing algorithm for video coding efficiency improvement," *IEEE Access*, vol. 8, pp. 920–929, 2019.

[15] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[17] R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, "Enhancing quality for HEVC compressed videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 7, pp. 2039–2054, 2018.

[18] 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, 2018.

[19] Z. Pan, X. Yi, Y. Zhang, B. Jeon, and S. Kwong, "Efficient in-loop filtering based on enhanced deep convolutional neural networks for HEVC," *IEEE Transactions on Image Processing*, vol. 29, pp. 5352–5366, 2020.

[20] D. Wang, S. Xia, W. Yang, Y. Hu, and J. Liu, "Partition tree guided progressive rethinking network for in-loop filtering of HEVC," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2671–2675.

[21] D. Ma, F. Zhang, and D. Bull, "MFRNet: a new CNN architecture for post-processing and in-loop filtering," *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[22] L. Yu, L. Shen, H. Yang, L. Wang, and P. An, "Quality enhancement network via multi-reconstruction recursive residual learning for video coding," *IEEE Signal Processing Letters*, vol. 26, no. 4, pp. 557–561, 2019.

[23] S. Zhang, Z. Fan, N. Ling, and M. Jiang, "Recursive residual convolutional neural network-based in-loop filtering for intra frames," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[24] T. M. Hoang and J. Zhou, "B-DRRN: A block information constrained deep recursive residual network for video compression artifacts reduction," in *2019 Picture Coding Symposium (PCS)*. IEEE, 2019, pp. 1–5.

[25] Q. Xing, M. Xu, T. Li, and Z. Guan, "Early Exit Or Not: Resource-efficient blind quality enhancement for compressed images," *arXiv preprint arXiv:2006.16581*, 2020.

[26] D. Ding, L. Kong, G. Chen, Z. Liu, and Y. Fang, "A switchable deep learning approach for in-loop filtering in video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[27] Z. Jin, P. An, C. Yang, and L. Shen, "Post-processing for intra coding through perceptual adversarial learning and progressive refinement," *Neurocomputing*, vol. 394, pp. 158–167, 2020.

[28] C. Jia, S. Wang, X. Zhang, S. Wang, and S. Ma, "Spatial-temporal residue network based in-loop filter for video coding," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.

[29] R. Yang, M. Xu, and Z. Wang, "Decoder-side HEVC quality enhancement with scalable convolutional neural network," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 817–822.

[30] J. Kang, S. Kim, and K. M. Lee, "Multi-modal/multi-scale convolutional neural network based in-loop filter design for next generation video codec," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 26–30.

[31] T. Wang, W. Xiao, M. Chen, and H. Chao, "The multi-scale deep decoder for the standard HEVC bitstreams," in *2018 Data Compression Conference*. IEEE, 2018, pp. 197–206.

[32] X. He, Q. Hu, X. Zhang, C. Zhang, W. Lin, and X. Han, "Enhancing HEVC compressed videos with a partition-masked convolutional neural network," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 216–220.

[33] J. W. Soh, J. Park, Y. Kim, B. Ahn, H.-S. Lee, Y.-S. Moon, and N. I. Cho, "Reduction of video compression artifacts based on deep temporal networks," *IEEE Access*, vol. 6, pp. 63 094–63 106, 2018.

[34] R. Yang, M. Xu, Z. Wang, and T. Li, "Multi-frame quality enhancement for compressed video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6664–6673.

[35] X. Song, J. Yao, L. Zhou, L. Wang, X. Wu, D. Xie, and S. Pu, "A practical convolutional neural network as loop filter for intra frame," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1133–1137.

[36] F. Li, W. Tan, and B. Yan, "Deep residual network for enhancing quality of the decoded intra frames of HEVC," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3918–3922.

[37] L. Ma, Y. Tian, and T. Huang, "Residual-based video restoration for HEVC intra coding," in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. IEEE, 2018, pp. 1–7.

[38] X. Meng, X. Deng, S. Zhu, S. Liu, C. Wang, C. Chen, and B. Zeng, "Mganet: A robust model for quality enhancement of compressed video," *arXiv preprint arXiv:1811.09150*, 2018.

[39] M.-Z. Wang, S. Wan, H. Gong, and M.-Y. Ma, "Attention-based dual-scale CNN in-loop filter for versatile video coding," *IEEE Access*, vol. 7, pp. 145 214–145 226, 2019.

[40] X. Meng, X. Deng, S. Zhu, and B. Zeng, "Enhancing quality for VVC compressed videos by jointly exploiting spatial details and temporal structure," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1193–1197.

[41] T. Li, M. Xu, C. Zhu, R. Yang, Z. Wang, and Z. Guan, "A deep learning approach for multi-frame in-loop filter of HEVC," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5663–5678, 2019.

[42] M. Wang, S. Wan, H. Gong, Y. Yu, and Y. Liu, "An integrated CNN-based post processing filter for intra frame in versatile video coding," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1573–1577.

[43] L. Feng, X. Zhang, S. Wang, Y. Wang, and S. Ma, "Coding prior based high efficiency restoration for compressed video," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 769–773.

[44] G. Chen, D. Ding, D. Mukherjee, U. Joshi, and Y. Chen, "AV1 in-loop filtering using a wide-activation structured residual network," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1725–1729.

[45] Y. Bei, Q. Wang, Z. Cheng, X. Pan, J. Lei, L. Wang, and D. Ding, "A CU-level adaptive decision method for CNN-based in-loop filtering," in *Eleventh International Conference*

on Graphics and Image Processing (ICGIP 2019), vol. 11373. International Society for Optics and Photonics, 2020, p. 113731G.

[46] G. Lu, X. Zhang, W. Ouyang, D. Xu, L. Chen, and Z. Gao, "Deep non-local kalman network for video compression artifact reduction," *IEEE Transactions on Image Processing*, vol. 29, pp. 1725–1737, 2019.

[47] D. Ding, G. Chen, D. Mukherjee, U. Joshi, and Y. Chen, "A CNN-based in-loop filtering approach for av1 video codec," in *2019 Picture Coding Symposium (PCS)*. IEEE, 2019, pp. 1–5.

[48] J. Tong, X. Wu, D. Ding, Z. Zhu, and Z. Liu, "Learning-based multi-frame video quality enhancement," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 929–933.

[49] M. Lu, T. Chen, H. Liu, and Z. Ma, "Learned image restoration for VVC intra coding." in *CVPR Workshops*, 2019, p. 0.

[50] X. Xu, J. Qian, L. Yu, H. Wang, X. Zeng, Z. Li, and N. Wang, "Dense inception attention neural network for in-loop filter," in *2019 Picture Coding Symposium (PCS)*. IEEE, 2019, pp. 1–5.

[51] C. Jia, S. Wang, X. Zhang, S. Wang, J. Liu, S. Pu, and S. Ma, "Content-aware convolutional neural network for in-loop filtering in high efficiency video coding," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3343–3356, 2019.

[52] X. Meng, X. Deng, S. Zhu, and B. Zeng, "Bstn: An effective framework for compressed video quality enhancement," in *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2020, pp. 320–325.

[53] X. Meng, X. Deng, S. Zhu, S. Liu, and B. Zeng, "Flow-guided temporal-spatial network for HEVC compressed video quality enhancement," in *2020 Data Compression Conference (DCC)*. IEEE, 2020, pp. 384–384.

[54] W.-G. Chen, R. Yu, and X. Wang, "Neural network-based video compression artifact reduction using temporal correlation and sparsity prior predictions," *IEEE Access*, 2020.

[55] Y.-H. Lam, A. Zare, F. Cricri, J. Lainema, and M. Hannuksela, "Efficient adaptation of neural network filter for video compression," *arXiv preprint arXiv:2007.14267*, 2020.

[56] H. Huang, I. Schiopu, and A. Munteanu, "Frame-wise CNN-based filtering for intra-frame quality enhancement of HEVC videos," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[57] F. Zhang, C. Feng, and D. R. Bull, "Enhancing VVC through CNN-based post-processing," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[58] W. Sun, X. He, H. Chen, R. E. Sheriff, and S. Xiong, "A quality enhancement framework with noise distribution characteristics for high efficiency video coding," *Neurocomputing*, vol. 411, pp. 428–441, 2020.

[59] H. Li, W. Lei, and W. Zhang, "QEVC: Quality enhancement-oriented video coding," in *2020 5th International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2020, pp. 296–300.

[60] X. Li, S. Sun, Z. Zhang, and Z. Chen, "Multi-scale grouped dense network for VVC intra coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 158–159.

[61] D. Ding, W. Wang, J. Tong, X. Gao, Z. Liu, and Y. Fang, "Biprediction-based video quality enhancement via learning," *IEEE transactions on cybernetics*, 2020.

[62] J. Wang, X. Deng, M. Xu, C. Chen, and Y. Song, "Multi-level wavelet-based generative adversarial network for perceptual quality enhancement of compressed video," *arXiv preprint arXiv:2008.00499*, 2020.

[63] X. Xu, J. Qian, L. Yu, H. Wang, H. Tao, and S. Yu, "Spatial-temporal fusion convolutional neural network for compressed video enhancement in HEVC," in *2020 Data Compression Conference (DCC)*. IEEE, 2020, pp. 402–402.

[64] T. Wang, J. He, S. Xiong, P. Karn, and X. He, "Visual perception enhancement for HEVC compressed video using a generative adversarial network," in *2020 International Conference on UK-China Emerging Technologies (UCET)*. IEEE, 2020, pp. 1–4.

[65] Z. Guan, Q. Xing, M. Xu, R. Yang, T. Liu, and Z. Wang, "MFQE 2.0: A new approach for multi-frame quality enhancement on compressed video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[66] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[67] Y. Xu, M. Zhao, J. Liu, X. Zhang, L. Gao, S. Zhou, and H. Sun, "Boosting the performance of video compression artifact reduction with reference frame proposals and frequency domain information," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 213–222.

[68] J. Xia and J. Wen, "Asymmetric convolutional residual network for av1 intra in-loop filtering," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1291–1295.

[69] F. Nasiri, W. Hamidouche, L. Morin, N. Dhollande, and G. Cocherel, "Prediction-aware quality enhancement of VVC using CNN," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2020, pp. 310–313.

[70] G. Lu, W. Ouyang, D. Xu, X. Zhang, Z. Gao, and M.-T. Sun, "Deep kalman filtering network for video compression artifact reduction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 568–584.

[71] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE multimedia*, vol. 18, no. 4, pp. 62–67, 2011.

[72] Apple, "HTTP live streaming," https://bitmovin.com/whitepapers/Bitmovin-Per-Title.pdf.

[73] A. Aaron, Z. Li, M. Manohara, J. De Cock, and D. Ronca, "Per-title encode optimization," *The Netflix Techblog*, 2015.

[74] M. Afonso, A. Moorthy, L. Guo, L. Zhu, and A. Aaron, "Improving our video encodes for legacy devices."

[75] L. Toni, R. Aparicio-Pardo, K. Pires, G. Simon, A. Blanc, and P. Frossard, "Optimal selection of adaptive streaming representations," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 11, no. 2s, pp. 1–26, 2015.

[76] J. De Cock, Z. Li, M. Manohara, and A. Aaron, "Complexity-based consistent-quality encoding in the cloud," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1484–1488.

[77] C. Chen, Y.-C. Lin, S. Benting, and A. Kokaram, "Optimized transcoding for large scale adaptive streaming using playback statistics," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3269–3273.

[78] Y. A. Reznik, K. O. Lillevold, A. Jagannath, J. Greer, and J. Corley, "Optimal design of encoding profiles for abr streaming," in *Proceedings of the 23rd Packet Video Workshop*, 2018, pp. 43–47.

[79] Y. A. Reznik, X. Li, K. O. Lillevold, A. Jagannath, and J. Greer, "Optimal multi-codec adaptive bitrate streaming," in *2019 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*. IEEE, 2019, pp. 348–353.

[80] K. Goswami, B. Hariharan, P. Ramachandran, A. Giladi, D. Grois, K. Sampath, A. Matheswaran, A. K. Mishra, and K. Pikus, "Adaptive multi-resolution encoding for abr streaming," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1008–1012.

[81] E. Çetinkaya, H. Amirpour, C. Timmerer, and M. Ghanbari, "Fast multi-resolution and multi-rate encoding for http adaptive streaming using machine learning," *IEEE Open Journal of Signal Processing*, vol. 2, pp. 484–495, 2021.

[82] C. Timmerer, H. Hellwagner *et al.*, "Mipso: Multi-period per-scene optimization for http adaptive streaming," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[83] PySceneDetect, "Intelligent scene cut detection and video splitting tool."

[84] Y. A. Reznik, K. O. Lillevold, and R. Vanam, "Perceptually optimized abr ladder generation for web streaming," *Electronic Imaging*, vol. 2021, no. 3, pp. 75–1, 2021.

[85] P. Lebreton and K. Yamagishi, "Network and content-dependent bitrate ladder estimation for adaptive bitrate video streaming," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4205–4209.

[86] K. Yamagishi and T. Hayashi, "Parametric quality-estimation model for adaptive-bitrate-streaming services," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1545–1557, 2017.

[87] H. Amirpour, C. Timmerer, and M. Ghanbari, "Pstr: Per-title encoding using spatio-temporal resolutions," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[88] C.-L. Fan, S.-C. Yen, C.-Y. Huang, and C.-H. Hsu, "On the optimal encoding ladder of tiled 360 videos for head-mounted virtual reality," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1632–1647, 2020.

[89] Bitmovin, "White paper: Per title encoding," https://bitmovin.com/whitepapers/Bitmovin-Per-Title.pdf.

[90] Cambria, "Feature: Source adaptive bitrate ladder (SABL)," https://www:capellasystems:net/capella_wp/wp-content/uploads/2018/01/CambriaFTC_SABL:pdf.

[91] MUX, "Instant per-title encoding," https://mux.com/blog/instant-per-title-encoding/.

[92] E. Bourtsoulatze, A. Chadha, I. Fadeev, V. Giotsas, and Y. Andreopoulos, "Deep video precoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4913–4928, 2019.

[93] A. Katsenou, J. Sole, and D. Bull, "Efficient bitrate ladder construction for content-optimised adaptive video streaming," *IEEE Open Journal of Signal Processing*, 2021.

[94] D. S. et al., "Machine learning for per-title encoding„" *NAB Broadcast Engineering and Information Technology (BEIT)*, 2020.

[95] M. Takeuchi, S. Saika, Y. Sakamoto, T. Nagashima, Z. Cheng, K. Kanai, J. Katto, K. Wei, J. Zengwei, and X. Wei, "Perceptual quality driven adaptive video coding using jnd estimation," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 179–183.

[96] J. Lubin and D. Fibushi, "Objective perceptual video quality measurement using a jnd-based full reference technique," *Alliance for Telecommunications Industry Solutions*, 2001.

[97] M. Bhat, J.-M. Thiesse, and P. Le Callet, "A case study of machine learning classifiers for real-time adaptive resolution prediction in video coding," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[98] A. Zabrovskiy, P. Agrawal, C. Timmerer, and R. Prodan, "Faust: Fast per-scene encoding using entropy-based scene detection and machine learning," in *2021 30th Conference of Open Innovations Association FRUCT*. IEEE, 2021, pp. 292–302.

[99] B. Bross, J. Chen, S. Liu, and Y.-K. Wang, "Versatile video coding (draft 7)," in *JVET-P2001, Geneva, Switzerland*, October 2019.

[100] AOMedia Video 1 (AV1). https://aomedia.googlesource.com/.

[101] K. Choi, J. Chen, D. Rusanovskyy, K.-P. Choi, and E. S. Jang, "An overview of the MPEG-5 essential video coding standard [standards in a nutshell]," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 160–167, 2020.

[102] 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.

[103] M. Karczewicz, N. Hu, J. Taquet, C.-Y. Chen, K. Misra, K. Andersson, P. Yin, T. Lu, E. François, and J. Chen, "VVC in-loop filters," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[104] D. Liu, Y. Li, J. Lin, H. Li, and F. Wu, "Deep learning-based video coding: A review and a case study," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–35, 2020.

[105] 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, 2019.

[106] J. Yao and L. Wang, "Convolutional neural network filter (CNNF) for intra frame," in *JVET-N0169, Geneva, Switzerland*, 2019.

[107] S. Liu, A. Segall, E. Alshina, and R.-L. Liao, "JVET common test conditions and evaluation procedures for neural network-based video coding technology," in *JVET-T2006, Teleconference*, 2020.

[108] F. Nasiri, W. Hamidouche, L. Morin, G. Cocherel, and N. Dhollande, "Model selection CNN-based VVC quality enhancement," in *2021 Picture Coding Symposium (PCS)*, 2021, pp. 1–5.

[109] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[110] F. Nasiri, W. Hamidouche, L. Morin, G. Cocherel, and N. Dhollande, "A study on the impact of training data in CNN-based super-resolution for low bitrate end-to-end video coding," in *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2020, pp. 1–5.

[111] E. Alshina, A. Segall, R.-L. Liao, and T. Solovyev, "[DNNVC] comments on common test conditions and reporting template," in *JVET-T0129, Teleconference*, 2020.

[112] D. Ma, F. Zhang, and D. R. Bull, "BVI-DVC: a training database for deep video compression," *arXiv preprint arXiv:2003.13552*, 2020.

[113] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, "JVET common test conditions and software reference configurations for SDR video," in *JVET-N1010, Geneva, Switzerland*, 2019.

[114] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[115] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. M. Manohara, "Toward a practical perceptual video quality metric," in *The Netflix Tech Blog, vol. 6*. Netflix, 2016, pp. 149–152.

[116] H. Wang, M. Karczewicz, J. Chen, and A. Meher Kotra, "AHG11: Neural network-based in-loop filter," in *JVET-T0079, Teleconference*, 2020.

[117] Y. Kidani, K. Kawamura, K. Unno, and S. Naito, "Evaluation results of CNN-based filtering with off-line learning model," in *JVET-O0132, Gothenburg, Sweden*, 2019.

[118] S. Wan, M. Wang, Y. Ma, J. Huo, H. Gong, C. Zou, Y. Yu, and Y. Liu, "Integrated in-loop filter based on CNN," in *JVET-O0079, Gothenburg, Sweden*, 2019.

[119] Y. Li, L. Zhang, K. Zhang, Y. He, and J. Xu, "AHG11: Convolutional neural networks-based in-loop filter," in *JVET-T0088, Teleconference*, 2020.

[120] Z. Wang, R.-L. Liao, C. Ma, and Y. Ye, "EE-1.6: Neural network based in-loop filtering," in *JVET-U0054, Teleconference*, 2021.

[121] A. Mackin, M. Afonso, F. Zhang, and D. Bull, "A study of subjective video quality at various spatial resolutions," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 2830–2834.

[122] D. collection. https://media.xiph.org/video/derf/. [Online]. Available: https://media.xiph.org/video/derf/

[123] M. Cheon and J.-S. Lee, "Subjective and objective quality assessment of compressed 4k uhd videos for immersive experience," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 7, pp. 1467–1480, 2017.

[124] L. Song, X. Tang, W. Zhang, X. Yang, and P. Xia, "The sjtu 4k video sequence dataset," in *2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2013, pp. 34–35.

[125] A. Mercat, M. Viitanen, and J. Vanne, "Uvg dataset: 50/120fps 4k sequences for video codec analysis and development," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 297–302.

[126] S. Winkler, "Analysis of public image and video databases for quality assessment," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 616–625, 2012.

[127] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[128] FFMPEG, "https://www.ffmpeg.org/."

[129] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of Applied Meteorology and Climatology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[130] M. Kuhn, K. Johnson *et al.*, *Applied predictive modeling*. Springer, 2013, vol. 26.

[131] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," *VCEG-M33*, 2001.

[132] L. Lévêque, W. Zhang, C. Cavaro-Ménard, P. Le Callet, and H. Liu, "Study of video quality assessment for telesurgery," *IEEE Access*, vol. 5, pp. 9990–9999, 2017.

[133] I. Mitrica, E. Mercier, C. Ruellan, A. Fiandrotti, M. Cagnazzo, and B. Pesquet-Popescu, "Very low bitrate semantic compression of airplane cockpit screen content," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2157–2170, 2019.

[134] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE signal processing magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[135] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos, "Softcuts: a soft edge smoothness prior for color image super-resolution," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 969–981, 2009.

[136] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[137] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer graphics and Applications*, no. 2, pp. 56–65, 2002.

[138] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[139] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[140] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[141] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[142] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[143] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3867–3876.

[144] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Super-resolution of compressed videos using convolutional neural networks," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1150–1154.

[145] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Super-resolution reconstruction of compressed video using transform-domain statistics," *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 33–43, 2004.

[146] H. Lin, X. He, L. Qing, Q. Teng, and S. Yang, "Improved low-bitrate HEVC video coding using deep learning based super-resolution and adaptive block patching," *IEEE Transactions on Multimedia*, 2019.

[147] B. A. et al., "Text of ISO/IEC CD 23094-2, low complexity enhancement video coding (LCEVC)," *Motion Picture Expert Group (MPEG) meeting 129*, 2019.

[148] U. Joshi, D. Mukherjee, Y. Chen, S. Parker, and A. Grange, "In-loop frame super-resolution in av1," in *2019 Picture Coding Symposium (PCS)*. IEEE, 2019, pp. 1–5.

[149] B. Bross and Y.-K. W. Jianle Chen, Shan Liu, "Versatile video coding draft text," *JVET-Q2001*, 2020.

[150] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE signal processing magazine*, vol. 15, no. 6, pp. 74–90, 1998.

[151] Y. Li, D. Liu, H. Li, L. Li, F. Wu, H. Zhang, and H. Yang, "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, 2018.

[152] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[153] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[154] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135.

[155] K. Suehring and X. Li, "JVET common test conditions and software reference configurations," *JVET-B1010*, 2016.

[156] T.-T. P.910, "Subjective video quality assessment methods for multi- media applications„" *ITU Telecom. Standardization Sector of ITU*, April 2008.

[157] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, 2001.

[158] W. Lin and L. Dong, "Adaptive downsampling to improve image compression at low bit rates," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2513–2521, 2006.

Titre : Optimisation basée sur l'apprentissage automatique pour le codage à faible débit avec VVC

Mot clés : faible débit, VVC, amélioration de la qualité, filtrage en boucle

Résumé : Cette thèse porte sur des méthodologies efficaces pour optimiser le codage et la transmission vidéo à bas débit, ou les limitations de bande passante entraînent souvent des vidéos avec des artefacts de compression notables (par exemple, flou, bloc et sonnerie). De tels artefacts peuvent réduire considérablement la qualité perçue du côté de l'utilisateur. En divisant le pipeline de diffusion vidéo en trois étapes principales de prétraitement, d'encodage et de post-traitement dans cette thèse, nous avons essayé de relever les défis du codage vidéo à faible débit à chaque étape. Tout d'abord, nous proposons une méthode d'amélioration de la qualité basée sur CNN en tant que post-traitement pour améliorer la qualité des vidéos fortement déformées avant l'affichage. Pour améliorer encore les performances de cet algorithme, nous tirons parti des informations disponibles dans le flux binaire reçu, telles que la prédiction, le partitionnement, le type de codage de bloc et le paramètre de quantification. De plus, pour réduire le débit binaire tout en améliorant la qualité, nous intégrons la méthode QE proposée comme filtre en boucle après tous les filtres de boucle existants dans VVC. Ensuite, comme la réduction d'échelle de la vidéo avant l'encodage peut être bénéfique à bas débit, nous avons mené une étude pour vérifier le potentiel des méthodes de super-résolution basées sur CNN à bas débit. De plus, pour déterminer la meilleure résolution vidéo avant l'encodage pour les cas d'utilisation spécifiques, nous avons développé une méthode basée sur le ML sensible au contenu pour construire l'échelle de débit binaire proche de l'optimum. Nous introduisons également une approche pour prédire l'échelle de débit d'un préréglage d'encodage spécifique à partir des autres préréglages. En résumé, nous avons proposé plusieurs méthodes et stratégies dans différentes parties des processus d'encodage et de décodage pour améliorer les performances du codage vidéo à faible débit.

Title: Machine learning based optimization for VVC low bitrate coding

Keywords: low bitrate, VVC, quality enhancement, in-loop filtering, bitrate ladder

Abstract: This thesis focuses on improving low-bitrate video coding and transmission, where the bandwidth limitations often result in videos with noticeable compression artifacts (*e.g.* blurriness, blockiness and ringing). Such artifacts can dramatically decrease the perceived quality. By breaking the video delivering pipeline into three main steps of pre-processing, encoding and post-processing in this thesis, we have tried to address the challenges of low bit-rate video coding in each step. First, we propose a CNN-based quality enhancement method as post-processing to enhance the quality of heavily distorted videos before display. To further improve the performance of this algorithm, we take advantage of the information available in the received bitstream. Moreover, to reduce the bitrate while enhancing the quality, we integrate the proposed QE method as in-loop filter after all existing loop filters in VVC. Next, as down-scaling the video before encoding can be beneficial at the low-bit rate, we conducted a study to verify the potential of CNN-based super-resolution methods in low-bitrate. In addition, to determine the best video resolution before the encoding for the specific use cases, we developed a content-aware ML-based method to construct the close to optimal bitrate ladder. We also introduce an approach to predict the bitrate ladder of a specific encoding preset from the other presets. In summary, several methods and strategies in different parts of the video ecosystem have been proposed to improve the overall performance in low-bitrate.