Author:
Ma, Di

Title:
Deep Video Compression

Deep Video Compression

Di Ma

Department of Electrical and Electronic Engineering
University of Bristol

A dissertation submitted to the University of Bristol in accordance with the requirements for award of the degree of Doctor of Philosophy in the Faculty of Engineering.

July 2022

Abstract

With the increase in demand for improved viewing quality and more immersive experiences, the tension between the large amounts of video data consumed everyday and the available bandwidth is ever increasing. To address this issue, new video coding standards have been developed including Versatile Video Coding and Alliance for Open Media Video 1. Although these compression techniques have achieved evident coding gains when compared to current standards, they still employ a similar framework to that used in previous codecs, but with much more sophisticated modifications and enhancements. None of them however, exploits recent advances in artificial intelligence and machine learning.

In this context, this thesis describes novel CNN-based algorithms to further enhance video compression efficiency. It first presents a new extensive and representative video database (BVI-DVC) for training deep video compression algorithms, which can provide significantly improved training effectiveness compared to other commonly used image and video training databases. The overall additional coding improvements (based on the HEVC HM 16.20) by using the BVI-DVC for all tested coding modules and CNN architectures are up to 10.3% based on the assessment of PSNR and 8.1% based on VMAF.

Novel network architectures have also been investigated in the context of video coding, including MFRNet, which consists of new multi-level feature review residual dense blocks. This structure offers significant coding gains when integrated into various enhancement-based coding tools. When compared to the state-of-the-art networks, up to 11.6% (PSNR) and 11.7% (VMAF) of overall additional coding gains have been provided by MFRNet based on the HEVC HM 16.20.

The perceptual quality of CNN reconstructed content has been further improved through the utilisation of GAN-based networks and training methodologies. The new CVEGAN architecture is also presented in this thesis, which achieves superior compression performance over state-of-the-art architectures for different coding tools (an average additional coding gain up to 18.1% (VMAF) has been achieved based on the HEVC HM 16.20).

Finally, the complexity issue of these CNN-based coding tools is addressed through flexible complexity distribution between the encoder and decoder. By including a CNN-based

resolution down-sampling module, we have achieved both coding performance improvement (more than 10% (BD-rate PSNR) based on the HEVC HM 16.20) and computational complexity reduction (29% and 10% for encoder and decoder, respectively).

Acknowledgements

First and foremost, I would like to thank my supervisor Professor David R. Bull for giving me an opportunity to pursue a PhD at the EPSRC Centre for Doctoral Training in Communications, University of Bristol. I am grateful for his great help, support and guidance throughout my PhD. These always encouraged me to overcome setbacks and complete this project. I also would like to thank my co-supervisor, Professor Iain Gilchrist for the discussions and help in my first PhD research year. Moreover, I wish to express my appreciation to Professor Andrew Calway for his guidance and help in our annual progress review meetings.

A very special thanks to my research mentor, Dr. Fan Zhang. During my PhD research, Dr. Zhang and I have deeply discussed research problems and completed several important works. Because of his great help, support and guidance, I have gradually improved my research ability and become a better researcher.

Furthermore, I am grateful to my colleagues and friends in VI Lab, Dr. Pui Anantrasirichai, Dr. Paul Hill, Dr. Igor Rizaev, Dr. Oktay Karakus, Justin Worsey, Perla Jazmin Mayo Diaz de Leon, Alexandra Malyugina, Stephen Boyle, Sanat Nagaraju, Yanan Liu, Xinyu Yang, Yuhang Ming, Xingrui Yang, Hongbo Bo, Yao Lu, Pengcheng Hao, Jian Ma, Chen Feng, Duolikun Danier, Jing Gao, Xin Tian, Hanyuan Wang, Ruixiong Wang, Tianqi Yang and Mengjie Zhou. A big thanks to Dr. Alex Mackin, Dr. Mariana Fernandez Afonso, Dr. Rui Fan, Dr. Hao Song, Dr. Yiheng Chen, Dr. Mengwei Xu and Dr. Yang Zhang for their great help and support during my PhD research. Especial thanks go to Miss Suzanne Binding, Dr. Simon Armour, Professor Mark Beach and Professor Oliver Johnson for all the help and support throughout the entire PhD journey.

My PhD and the thesis would not have been possible without the support from my loving parents. I would like to express my deep appreciation to them for their persistent support and love in my study and my daily life.

Finally, I would like to thank my loving wife, for her love, constant support and for all the nights and early mornings. Without her tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my PhD.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: Di Ma DATE: May 2021

I would like to dedicate this thesis to my loving wife, parents, grandfather and grandmother.

Table of Contents

Abstract iii

Acknowledgements v

Declaration vii

List of Figures xv

List of Tables xxi

List of Abbreviations xxv

1 Introduction 1
- 1.1 Context . . . 1
- 1.2 Theme of the Thesis . . . 3
- 1.3 Aims and Objectives . . . 4
- 1.4 Approaches . . . 4
- 1.5 Thesis Organisation . . . 5

2 Deep Video Coding: A Review 7
- 2.1 Video Compression Standards . . . 7
- 2.2 Deep Learning . . . 11
  - 2.2.1 Basic Theory . . . 11
  - 2.2.2 Convolutional Neural Networks (CNNs) . . . 12
  - 2.2.3 Generative Adversarial Networks (GANs) . . . 15
  - 2.2.4 Applications and Approaches . . . 16
- 2.3 Deep Learning-based Video Compression . . . 17
  - 2.3.1 Training and Test Databases . . . 17
  - 2.3.2 Learning-based Coding Architectures . . . 20
  - 2.3.3 CNN-based Coding Tools . . . 21
  - 2.3.4 Typical Training Methodologies . . . 24
  - 2.3.5 Complexity Issue . . . 26

2.4 Video Quality Assessment 26
2.4.1 Objective Quality Metrics 27
2.4.2 Subjective Quality Experiment 29
2.5 Summary 31

3 BVI-DVC: A Training Database 33
3.1 Database Description 33
3.2 Experimental Configurations 35
3.2.1 Test Coding Modules 36
3.2.2 Evaluated CNN Models 36
3.2.3 Training Data Generation 37
3.2.4 Network Training and Evaluation 38
3.2.5 Experiment Settings 40
3.3 Results and Discussion 45
3.3.1 Comparison of Databases 45
3.3.2 Comparison of Networks 47
3.4 Summary 49

4 MFRNet: A New CNN Architecture for Deep Video Coding Enhancement 51
4.1 Network Architecture 51
4.1.1 Network Backbone Structure 52
4.1.2 Multi-level Feature Review Residual Dense Block (MFRB) 52
4.1.3 Feature Review Residual Dense Block (FRB) 53
4.2 Experimental Configurations 53
4.3 Results and Discussion 54
4.3.1 Comparisons with Popular CNN Architectures 54
4.3.2 Compression Performance 62
4.3.3 Comparison between CNN-based PP and ILF Approaches 67
4.4 Summary 68

5 Perceptually-inspired Deep Video Coding Enhancement 71
5.1 Network Architectures 72
5.1.1 Preliminary Work-I: MSRGAN for SRA 72
5.1.2 Preliminary Work-II: BDGAN for EBDA 73
5.1.3 A New GAN Architecture: CVEGAN 75
5.1.4 Relationship between MSRGAN, BDGAN and CVEGAN 80
5.2 Network Training and Evaluation 80
5.2.1 Training Methodology-MSRGAN and BDGAN 80
5.2.2 Training Methodology-CVEGAN 82
5.2.3 Training and Evaluation Configurations 87

5.3 Experimental Configuration 88
5.3.1 Experiment Settings 88
5.3.2 Benchmarked CNN and GAN Architectures: CVEGAN Comparisons 88
5.3.3 Ablation Study: CVEGAN Comparisons 89
5.3.4 Subjective Test Configuration: CVEGAN Comparisons 89
5.4 Results and Discussion 91
5.4.1 MSRGAN 91
5.4.2 BDGAN 92
5.4.3 CVEGAN 96
5.5 Summary 106

6 Complexity Analysis 115
6.1 Complexity Analysis for the Proposed Networks 115
6.1.1 Latency Analysis 116
6.1.2 Analysis of Network Complexity and Coding Performance 116
6.2 New Coding Framework with Reduced Computational Complexity 118
6.2.1 Different SRA Scenarios 118
6.2.2 SRA with CNN-based Down-sampling 119
6.2.3 Training Methodology 121
6.2.4 Training and Evaluation Configurations 122
6.2.5 Experimental Configurations 122
6.2.6 Results and Discussion 123
6.3 Summary 125

7 Conclusion 127
7.1 Contributions 127
7.2 Future Work 129

References 133

List of Figures

2.1 Basic CNN structure for image and video restoration. 13
2.2 Basic GAN structure. 15
2.3 Coding workflow with a CNN-based PP module. 22
2.4 Coding workflow with a CNN-based ILF module. 23
2.5 Coding workflow with a CNN-based SRA module. 23
2.6 Coding workflow with a CNN-based EBDA module. 24

3.1 Sample frames of 20 example sequences from the BVI-DVC database. . . . 35
3.2 Diagram of the frame reconstruction method used in the thesis. 40
3.3 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), RDN models trained using six databases for PP coding tool. The bit consumption in each example set is identical/similar for all tested versions. Rows 1, 2 and 3 correspond to the 250th frame of the *DaylightRoad2* sequence. 46
3.4 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), RDN models trained using six databases for SRA coding tool. The bit consumption in each example set is identical/similar for all tested versions. Rows 1, 2 and 3 correspond to the 104th frame of the *CatRobot1* sequence. 47
3.5 Average coding gains for four coding modules obtained using 10 commonly employed network architectures trained on six different databases. 48
3.6 Average BD-rate (based on PSNR and VMAF) of six tested training databases for all the evaluated coding modules and CNN architectures. 48
3.7 Average BD-rate (based on PSNR and VMAF) of 10 test network architectures for all coding modules and training databases. 49

4.1 Illustration of the proposed MFRNet architecture. 52
4.2 Illustration of an MFRB (B_i). 53

4.3 Illustration of an FRB (b_i^j). 54

4.4 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 170th frame of the *PartyScene* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast. 56

4.5 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 216th frame of the *CatRobot1* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast. 57

4.6 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 250th frame of the *DaylightRoad2* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast. 58

4.7 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based SRA. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 98th frame of the *CatRobot1* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast. 59

4.8 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based SRA. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 216th frame of the *CatRobot1* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast. 60

4.9 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based SRA. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 250th frame of the *DaylightRoad2* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast. 61

5.1 Network architecture of the MSRGAN's Generator (MSRResNet). 72

5.2 Comparison of the original and the modified residual blocks. 73

5.3 Network architecture of the Discriminator for MSRGAN. 73

5.4 Network architecture of the BDGAN's Generator (BDNet). 74

5.5 Residual Dense Block (RDB) used in BDNet. 74

5.6 Illustration of the CVEGAN's Generator (CVENet). 75

5.7 Illustration of an Mul^{2}Res Block. 77

5.8 Illustration of an ERNB. 78

5.9 Illustration of an ECBAM. 79

5.10 Illustration of the CVEGAN's Discriminator. 79

5.11 Illustration of the proposed ReSphereGAN. The yellow plane and sphere represent the 1024-dimensional Euclidean feature space and hypersphere respectively. Green and purple points represent the feature points of fake and real data respectively. 84

5.12 Perceptual comparisons between the HM 16.20 and the proposed approach using MSRResNet-$\ell 1$ and MSRGAN (patches extracted from the 26th, 17th and the 270th frames of *Campfire*, *Tango2* and *DaylightRoad2* reconstructed sequences respectively and amplified by 4 times). 95

5.13 Example blocks of the reconstructed frames for the anchor HM 16.20, EBD up-sampling with BDNet-$\ell 1$ and BDGAN (their bitstreams have similar bit rates). These are from the 175th and 162nd frames of *Campfire* and *PartyScene* sequences respectively and amplified by 4 times. 97

5.14 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2, 3 and 4 correspond to the 170th frame of the *PartyScene* sequence. 107

5.15 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2, 3 and 4 correspond to the 104th frame of the *CatRobot1* sequence. 108

5.16 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2, 3 and 4 correspond to the 250th frame of the *DaylightRoad2* sequence. 109

5.17 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2, 3 and 4 correspond to the 216th frame of the *CatRobot1* sequence. 110

5.18 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based SRA. The bit consumption in each example set is similar for all tested versions. Rows 1, 2, 3 and 4 correspond to the 104th frame of the *CatRobot1* sequence. 111

5.19 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based SRA. The bit consumption in each example set is similar for all tested versions. Rows 1, 2, 3 and 4 correspond to the 250th frame of the *DaylightRoad2* sequence. 112

5.20 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based SRA. The bit consumption in each example set is similar for all tested versions. Rows 1, 2, 3 and 4 correspond to the 216th frame of the *CatRobot1* sequence. 113
5.21 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based SRA. The bit consumption in each example set is similar for all tested versions. Rows 1, 2, 3 and 4 correspond to the 161st frame of the *RaceNight* sequence. 114

6.1 (Left) Relative complexity (decoding) for different number of residual blocks. (Right) PSNR gains for different number of residual blocks. 117
6.2 Diagram of the generic spatial resolution adaptation workflow. 119
6.3 Network architecture of the proposed DSNet. 120
6.4 Residual Dense Block (RDB) used in DSNet. 120

List of Tables

2.1 Relative average BD-rate results for JVET-CTC tested sequences under the corresponding configurations for four standard video codecs (codecs in column utilised as the anchor). 10

3.1 Key features of thirteen training databases including BVI-DVC. 34
3.2 Evaluation results for PP coding module for ten tested network architectures and six different training databases. Values indicate the average BD-rate (%) for all nineteen JVET-CTC tested sequences assessed by PSNR or VMAF. . 41
3.3 Evaluation results for ILF coding module for ten tested network architectures and six different databases. Each value indicates the average BD-rate (%) for all nineteen JVET-CTC tested sequences assessed by PSNR or VMAF. . . . 42
3.4 Evaluation results for SRA coding module for ten tested network architectures and six different databases. Each value indicates the average BD-rate (%) for all six UHD JVET-CTC tested sequences assessed by PSNR or VMAF. 43
3.5 Evaluation results for EBDA coding module for ten tested network architectures and six different databases. Each value indicates the average BD-rate (%) for all nineteen JVET-CTC tested sequences assessed by PSNR or VMAF. 44

4.1 Comparison between eleven popular CNN architectures and the proposed MFRNet in the context of ILF and PP for HM 16.20. 55
4.2 Comparison between eleven popular CNN architectures and the proposed MFRNet in the context of SRA and EBDA for HM 16.20. 55
4.3 Compression results of the MFRNet-based ILF and PP for HM 16.20. . . . 63
4.4 Compression results of the MFRNet-based ILF and PP for VTM 7.0. 64
4.5 Compression results of the MFRNet-based EBDA and SRA for HM 16.20. 65
4.6 Compression results of the MFRNet-based EBDA and SRA for VTM 7.0. . 66
4.7 Comparison between MFRNet-based PP and ILF and existing CNN-based PP and ILF approaches for HEVC. 67

4.8 Comparison between MFRNet-based PP and ILF and existing CNN-based PP and ILF approaches for VVC. 68

5.1 Cross-validation results over eight training-testing trails. 83
5.2 Compression results of the MSRResNet and MSRGAN-based SRA for HM 16.20 based on the JVET-CTC UHD tested sequences. 92
5.3 The compression performance of the MSRResNet and MSRGAN-based PP methods benchmarked on the original VVC VTM 4.0.1. Negative BD-rate values indicate coding gains. 93
5.4 The compression performance of the the MSRResNet and MSRGAN-based PP methods benchmarked on the original AV1 1.0.0. Negative BD-rate values indicate coding gains. 94
5.5 Compression results of the BDNet and BDGAN-based EBDA for HM 16.20 based on the JVET-CTC tested sequences. 96
5.6 Compression results of the CVEGAN-based PP for HM 16.20 and VTM 7.0. 98
5.7 Compression results of the CVEGAN-based EBDA and SRA for HM 16.20. 99
5.8 Compression results of the CVEGAN-based EBDA and SRA for VTM 7.0. 100
5.9 Comprehensive comparison results (in terms of BD-rate (%) based on both PSNR and VMAF) between the CVEGAN and 24 benchmark networks when they are integrated into the PP coding tool for HEVC compression. The result sets {i/j/k} in this table stands for the BD-rate values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test network is also provided for comparison. 101
5.10 Comprehensive comparison results (in terms of BD-rate (%) based on both PSNR and VMAF) between the CVEGAN and 24 benchmark networks when they are integrated into the SRA coding tool for HEVC compression. The result sets {i/j/k} in this table stands for the BD-rate values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test network is also provided for comparison. 102
5.11 Ablation study results for PP coding tool based on the HEVC HM 16.20. The result sets {i/j/k} in this table stands for the BD-rate (%) values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test variant is also provided for comparison. 103
5.12 Ablation study results for SRA coding tool based on the HEVC HM 16.20. The result sets {i/j/k} in this table stands for the BD-rate (%) values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test variant is also provided for comparison. 104

5.13 Comparison between CNN and GAN architectures presented in Chapters 4 and 5 in the context of PP, SRA and EBDA for HM 16.20. 105
5.14 Comparison between CNN and GAN architectures presented in Chapters 4 and 5 in the context of PP, SRA and EBDA for VTM 7.0. 105
5.15 Subjective results based on 12 UHD source sequences from the JVET-CTC and UVG test datasets. 106

6.1 Latency results (millisecond-ms) of different network architectures. 116
6.2 Compression performance comparison between various SRA scenarios and the original HEVC HM 16.20 (AI configuration) ("↓" and "↑" represent spatial down-sampling and up-sampling respectively). 123
6.3 Relative complexity for four SRA Scenarios. 125

List of Abbreviations

AV1	AOMedia Video 1
AV2	AOMedia Video 2
AVC	Advanced Video Coding
BVI	Bristol Vision Institute
CBD	Coding Bit Depth
CNN	Convolutional Neural Network
DMOS	Difference Mean Opinion Score
DNN	Depp Neural Network
EBD	Effective Bit Depth
EBDA	Effective Bit Depth Adaptation
GAN	Generative Adversarial Network
HEVC	High Efficiency Video Coding
HM	HEVC Test Model
HVS	Human Visual System
ILF	In-loop Filtering
JVET	Joint Video Exploration Team
MOS	Mean Opinion Score
PP	Post-processing
PSNR	Peak Signal-to-Noise Ratio
QP	Quantisation Parameter
SDR	Standard Dynamic Range
SI	Spatial Information
SRA	Spatial Resolution Adaptation
TI	Temporal Information

UHD	Ultra High Definition
VMAF	Video Multimethod Assessment Fusion
VTM	VVC Test Model
VVC	Versatile Video Coding

Chapter 1

Introduction

The importance of video compression has come to the fore over the past two decades driven by the tension between the huge quantities of video content consumed everyday and the bandwidth available for transmission. This challenge has been addressed through the development of new video coding standards with progressively improved performance through the adoption of numerous sophisticated coding tools. Recently, non-conventional techniques, especially deep learning-based algorithms have been increasingly applied in the context of image and video compression. These approaches, in particular for those learning-based coding tools can be flexibly integrated into the standard video codecs and provide evident coding gains over the current compression standards. This type of approach is one of the most active research fields in the image and video compression community demonstrating great potential for future multimedia delivery technologies, which however still needs to be carefully explored.

In this chapter, we firstly review the context of video compression, and describe the motivations and theme of the thesis. We then present the aims and objectives of the research and overview the approaches utilised. Finally, the thesis organisation is presented.

A subset of the content presented in this chapter has been published in [1–7].

1.1 Context

In recent years, the consumption of video content has increased dramatically. It was predicted that over 82% of all global Internet traffic will be video by 2022 [8]. This has been associated with demands for improved viewing quality, and for more immersive experiences (e.g. augmented and virtual reality, 360°, etc.) with multiple views, higher spatial and temporal resolutions and wider dynamic range [9]. These new formats typically consume much higher

bitrates which significantly challenge the limited network capacity and the current video compression techniques.

To address this issue, multiple generations of video coding standards have been developed by ITU-T and/or ISO/IEC for various application scenarios since the early 1980s. Among these, the most successful has been H.264/AVC (Advanced Video Coding) [10] which was released in 2004 targeting Internet streaming and HDTV. It is still widely used, although its successor HEVC/H.265 (High Efficiency Video Coding) [11] can provide nearly 50% overall coding efficiency improvement. In 2018, in order to support immersive video formats (e.g. high dynamic range and 360°) and offer further compression efficiency improvements, a new coding standard, Versatile Video Coding (VVC) [12], was initiated in 2018. It has been finalised in 2020 [13] and currently [14] shows an improvement of more than 35% overall coding gain over HEVC, but with a significant increase in encoder complexity.

In parallel with developments under ITU-T and ISO/IEC, the Alliance for Open Media (AOM, an industry consortium) has developed an open source and royalty-free coding solution, AOM Video 1 (AV1), targeting Internet streaming. AV1 was launched in 2018, and its most recent versions appear to offer evident and consistent performance improvements over HEVC [14, 15]. The development of its successor, AV2 (AOMedia Video 2), has started in 2020 to further enhance coding gains over AV1 [16]. Other recent advances in coding standards include the Essential Video Coding/MPEG-5 [17] and AVS standards [18] that target royalty-free solutions and complexity-performance trade-offs.

It is noted that all of these new coding standards mentioned above employ a similar framework to that used in previous coding standards such as H.264/AVC (Advanced Video Coding) [10], but with much more sophisticated modifications and enhancements. None of them however, exploits recent advances in (deep) machine learning.

The last decade has seen significant advances in the application of machine learning, especially using convolutional neural networks (CNNs), for various computer vision applications from low-level vision tasks (e.g. image and video restoration [19, 20], etc.) to high-level vision applications (e.g. image recognition and classification [21, 22], etc.). More recently, deep learning techniques have also been applied to the problem of image and video compression. Not only to enhance existing coding modules (intra or inter prediction, transform, post-processing, in-loop filtering, etc.), but also to provide new end-to-end solutions [23, 24].

Among learning-based video coding modules, there is a distinct class of methods (denoted as video compression enhancement tools) which provide superior performance by employing CNN processes to enhance the quality of the video reconstructed at the decoder or encoder. Typical examples include post-processing (PP), in-loop filtering (ILF) and video format adaptations (with CNN format restoration) [23, 24]. More recently, these compression

enhancement tools have further improved perceptual quality by using Generative Adversarial Networks (GANs) [25].

It is noted that the coding gains reported by the previous CNN-based video compression enhancement approaches are primarily for less effective coding configurations in HEVC and VVC (e.g. All Intra and Low Delay modes). They cannot provide evident and consistent bit rate savings for a more effective coding mode (i.e. Random Access configuration) [23, 24]. This is mainly due to the following reasons.

(1) The CNNs employed in these approaches are often trained on databases with relatively limited content coverage. To the best of our knowledge, there is no such public video training database which contains a wide range of video content types and scenes and is specifically developed for training CNN-based video coding algorithms;

(2) These approaches often employ simple network structures primarily developed for image/video restoration tasks. These networks have not utilised advanced architectures (such as residual dense blocks, cascading connections or feature review structures, etc.), which do not reflect the latest advances in the field;

(3) For those GAN-based video compression enhancement methods, the loss functions utilised in the training phase have typically combined pixel-wise distortions (e.g. $\ell 1$ or $\ell 2$ loss), low complexity quality metrics (e.g. SSIM [26] and MS-SSIM [27]) and feature map differences (e.g. VGG19-54 [28, 29]) with artificially configured weights, which do not offer optimal correlation with visual quality.

Additionally, it is noted that exiting deep learning-based video coding enhancement tools always suffer from high computational complexity due to the CNN operations. Especially, using trained CNN models at the decoder can significantly increase the decoding complexity. In order to reduce the complexity of this type of approach whilst maintaining high coding efficiency, lightweight and effective network architectures, and low complexity coding modules also need to be carefully addressed.

1.2 Theme of the Thesis

In this context, this thesis focuses on developing novel and effective deep learning methodologies for video compression enhancement modules, including creating a new extensive and representative video database, designing novel CNN and GAN architectures with advanced network structures, and developing new training methodologies. Moreover, we have demonstrated the potential of using CNN-based methods to reduce overall computational complexity while achieving enhanced coding performance compared to conventional codecs.

1.3 Aims and Objectives

This thesis explores advanced deep learning algorithms to effectively enhance coding efficiency and perceptual quality of compressed video content based on the typical video coding enhancement modules. Specifically, the aims include:

- Significant coding gains through the utilisation of CNN-based video coding enhancement tools to the current video coding standards - HEVC, VVC and AV1.
- Relatively low computational complexity compared to other deep learning-based methods.

In order to achieve these aims, the objectives of the thesis are summarised below:

- To develop a new and extensive video database for training deep video compression algorithms. This database should contain diverse content types and scenes relatively avoiding overfitting problems and effectively optimising the generalisation ability of the deep neural networks.
- To develop novel CNN and GAN architectures with carefully designed network structures which can provide optimal enhancement performance for video content with complex textures and compression artefacts.
- To design new training methodologies and perceptual loss functions to improve the perceptual quality of enhanced content.
- To design low complexity CNN-based coding framework in order to achieve the optimal trade-off between the compression performance and computational complexity.

1.4 Approaches

As discussed in Section 1.1, existing CNN-based video coding enhancement algorithms suffer from sub-optimal performance due to the use of training databases with limited content coverage and ineffective deep learning algorithms (including network architectures and training methodologies). In this context, the following approaches have been proposed to achieve the objectives of the thesis.

- To collect publicly available video content to cover various content types, different video textures (static and dynamic) and scenes as widely as possible, and carefully

select sequences among the collected content following specific strategies to develop an extensive and representative video database for training deep learning-based video coding methods.

- To design novel CNN and GAN architectures for video compression enhancement using recent advances in network structures (such as residual dense connections, feature review structures and attention mechanisms, etc.) to effectively improve overall network performance.

- To develop new perceptual loss functions (trained on existing video quality databases using the cross-validation method) and novel GAN training methodology to further enhance perceptual quality of compressed videos and stabilise adversarial training process in the context of video compression.

- To design new low complexity CNN-based coding framework to enable the flexible complexity allocation between the encoder and decoder, reducing complexity for both encoding and decoding processes whilst providing evident coding gains.

1.5 Thesis Organisation

The rest of the thesis is organised as follows:

In Chapter 2, a brief review of video coding standards and the background to the field of deep learning techniques is presented. We also summarise the recent advances in deep video compression, including the existing coding modules, end-to-end solutions, training and test databases, network architectures and training methodologies. Finally, the subjective and objective quality assessments are described.

In Chapter 3, a new extensive and representative video database, BVI-DVC is presented for training deep video compression algorithms. A comprehensive experiment and analysis of the performance of BVI-DVC in the context of four compression enhancement tools compared with five existing (commonly used) databases are conducted, demonstrating the effectiveness of the BVI-DVC database. Additionally, a performance comparison of ten popular CNN architectures based on the identical training materials and evaluation configurations is also presented, further showing the importance of the network architectures in deep video compression.

In Chapter 4, we present a novel CNN architecture-MFRNet which exploits new multi-level feature review residual dense blocks for typical coding enhancement tools: PP, ILF, SRA and EBDA. It has been integrated into both HEVC and VVC test models, demonstrating

significant coding gains for Random Access configurations. A comprehensive comparison is conducted between the MFRNet and thirteen existing state-of-the-art network architectures based on the same training and evaluation material. In addition, we also compared the proposed method with other notable CNN-based PP, ILF, SRA and EBDA approaches developed for the HEVC and/or VVC Random Access configuration. These all demonstrate the effectiveness of the MFRNet.

In Chapter 5, we firstly present two primary works MSRGAN and BDGAN for perceptually-inspired video coding enhancement based on the spatial resolution and effective bit depth adaptations respectively. In order to further improve coding performance, a novel GAN architecture (CVEGAN) is then presented employing new and advanced network structures to effectively improve coding efficiency and perceptual quality of compressed video content. Additionally, we also developed a new GAN training methodology together with the novel perceptual loss functions. The comprehensive performance comparisons between the CVEGAN, original HEVC and VVC test models, as well as several state-of-the-art network architectures are conducted respectively, demonstrating the effectiveness of the CVEGAN.

In Chapter 6, the computational complexity of the typical CNN-based compression enhancement tools (PP, ILF, SRA and EBDA) is comprehensively analysed. Then, a new low complexity coding framework has been presented for video compression based on the spatial resolution adaptation. The experimental results show that the proposed coding framework offers a trade-off solution between computational complexity and coding performance, and enables flexible complexity allocation between the encoder and decoder. This provides a reference to the low complexity CNN-based coding framework design.

Finally, Chapter 7 provides a conclusion of the works presented in this thesis and outlines the directions for future work.

Chapter 2

Deep Video Coding: A Review

This chapter presents a background review of the video compression standards, deep learning techniques and their applications in the field of computer vision, and an overview of deep learning-based video compression methods. Moreover, the typical objective quality metrics which are commonly used for compressed video quality assessment are also introduced in this chapter. A subset of the work presented in this chapter has been published in [1–7].

2.1 Video Compression Standards

Since the early 1980s, multiple generations of video coding standards have been developed by ITU-T and/or ISO/IEC for various application scenarios, including video conferencing, film, television, terrestrial and satellite transmission, surveillance and particularly Internet video [9, 4]. These video coding standards define the format of bitstream, syntax and the decoder [9]. In most cases, associated with each coding standard, a corresponding reference encoder is also provided, such as HEVC Test Model (HM) for HEVC and the VVC Test Model (VTM) for VVC. These reference codecs not only produce the standard-compliant bitstream, but also offer the benchmark encoding performance. Besides reference models, different encoder variants are also developed for various application scenarios and requirements. Some of them focus on high coding performance (but with high complexity), while others are designed to have a fast encoding process but with relatively low compression efficiency.

The development history and primary features of the major video compression standards are briefly described below.

- H.120: In 1980, the Study Group SB XV of the CCITT started to develop H.120 [30] and released the first international digital video coding standard in 1984. It was designed for videoconferencing applications with bit rate requirements of 2.048 Mb/s

and 1.544 Mb/s for 625/50 and 525/60 TV systems respectively. This standard was not successfully utilised for commercial use mainly due to the low compressed video quality, especially the unsatisfactory temporal quality.

- H.261: Based on the H.120, H.261 was proposed in 1989 [31]. H.261 was developed based on a $p \times 64$ kbit/s model ($p = 1, 2, ..., 30$) for ISDN conferencing applications. It was the first video coding algorithm which adopted a block-based hybrid compression framework with the combination of discrete cosine transform (DCT)-based transform, temporal differential pulse code modulation (DPCM), motion estimation and compensation techniques.

- H.262/MPEG–2: The Moving Picture Experts Group (MPEG) was founded in 1988 and produced a video compression standard H.262/MPEG–2 [32] in 1994 for digital video broadcasting applications. From 1994 to 2004, the H.262/MPEG–2-compliant encoder performance has been significantly improved by employing high cost and advanced coding methodologies. The development of this standard has a great impact on digital video communications technology [9].

- H.263: From 1993, the ITU-T SG15 commenced work on H.263 which was expected to achieve the goal of encoding at bit rate below 64 kbit/s [33]. It was primarily developed for videoconferencing, surveillance applications as well as early Internet streaming (e.g. utilised in YouTube and MySpace). In addition, H.263 was developed to be integrated into the H.324 framework for circuit-switched applications [9].

- H.264/AVC: The ITU-T/SG16/Q6 (Video Coding Experts Group - VCEG) and MPEG produced the H.264/AVC (Advanced Video Coding) [10, 34] in 2003 and finalised a scalable video coding (SVC) extension for this standard in 2007. H.264/AVC was the most successful and widely used coding standard, targeting Internet streaming (including Vimeo, YouTube and iTunes), mobile video services, OTT services, IPTV and HDTV, as well as satellite and terrestrial broadcasting applications [9, 34].

- H.265/HEVC: In 2013, ISO/IEC MPEG and ITU-T VCEG released a new video coding standard, H.265/HEVC (High Efficiency Video Coding) [11, 35] which has achieved up to 50% coding gain over the H.264/AVC. It was developed for mobile video, broadcasting and Internet streaming services providing higher visual quality and more immersive experiences for consumers [9]. Similar to its predecessor H.264/AVC, the H.265/HEVC also utilised a block-based hybrid compression framework but with new features and coding tools to reduce encoding bit rates. In the HEVC reference test model (HM), there are three coding configurations available for users to test based

on different application scenarios: All Intra, Low Delay and Random Access modes [9]. This reference test model aims to provide a standard-compliant bitstream and simultaneously achieve the best encoding performance compared to other codecs.

- H.266/VVC: Based on the H.265/HEVC, the Joint Video Experts Team (JVET) (a joint collaborative team with experts from both ITU-T and ISO/IEC) has initiated the latest video coding standard H.266/VVC (Versatile Video Coding) [12, 13] and finalised it in July 2020 with more than 35% overall coding gain over H.265/HEVC. It is developed to support more immersive video formats (including higher spatial resolution, dynamic range, 360°, etc.) and significantly enhance compression performance over the current coding standards. The H.266/VVC test model (VTM) also supports three coding configurations, All Intra, Low Delay and Random Access that are similar to those provided in H.265/HEVC HM. More recently, the JVET Ad-hoc Group 11 (AHG11) has studied the potential of extending H.266/VVC with Neural-network-based coding tools for further improving compression efficiency based on existing coding modules, such as intra or inter prediction, in-loop filtering and post-processing [36, 37].

- VP9: Google commenced work on an open source and royalty-free video coding format, VP9, in 2011 and then finalised it in 2017 with significantly enhanced encoding speed. It was primarily developed for Internet streaming (YouTube) and has supported various modern web browsers (such as the Android browser and Google Chrome) [9].

- AV1: The first video coding standard of the Alliance of Open Media (AOM), AOMedia Video 1 (AV1) [38] was released in 2018 and its performance has been progressively enhancing since then, offering significant coding gains over the H.265/HEVC [14]. AV1 is also developed based on the block-based hybrid video compression framework and provides open source and royalty-free solutions for video content delivery. There are several unique features and coding tools utilised in AV1 that are different with H.265/HEVC and H.266/VVC, including the new non-directional intra-prediction modes, motion compensation based on overlapped blocks, new asymmetric discrete sine transform and frame super-resolution technique, etc. It is noted that the next AOM codec version, AOMedia Video 2 (AV2) has already been developed [16] since 2020 based on the AV1 but with more advanced coding tools to further improve compression efficiency, and it will be the primary competitor of the H.266/VVC.

- EVC/MPEG–5 and AVS: The Essential Video Coding (EVC)/MPEG–5 [17] and Audio Video Coding Standard (AVS) [18] are another two notable video coding standards that also provide open source and royalty-free multimedia delivery solutions. It is noted

that their recent versions have achieved coding performance improvements over the H.265/HEVC.

Table 2.1 Relative average BD-rate results for JVET-CTC tested sequences under the corresponding configurations for four standard video codecs (codecs in column utilised as the anchor).

	JM 19	HM 16.20	VTM 7.0	AV1 libaom
Video Codec	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)
JM 19	–	+81.2%	+172.8%	+126.8%
HM 16.20	-42.8%	–	+49.4%	+27.2%
VTM 7.0	-61.0%	-32.5%	–	-14.7%
AV1 libaom	-53.8%	-20.7%	+17.9%	–

Table 2.1 summarises the results presented in [14], where four video coding standards are evaluated on the test sequences in the JVET Common Test Conditions (CTC) dataset [39] (this will be introduced in detail in Section 2.3.1). Tested standard video codecs include H.264/AVC JM 19, H.265/HEVC HM 16.20, H.266/VVC VTM 7.0 and AV1 libaom (version 1.0.0-5ec3e8c). They are compared using the Bjøntegaard Delta [40] measurement (BD-rate) based on the assessment of PSNR (Peak-Signal-to-Noise-Ratio, luminance channel only). Here, the BD-rate statistics indicate the overall bitrate savings across the tested QP (quantisation parameter) range achieved by the test algorithm for the same video quality compared to the anchor approach. The algorithm first takes logarithm to bitrates and separately fits two rate-distortion curves (including both anchor and tested codecs) as a function of the objective quality (e.g. PSNR) using the third order polynomial. Then, the difference between the integrals of these two fitted curves is calculated and further divided by the given quality interval to obtain the BD-rate result [40]. Negative BD-rate values indicate bitrate savings or coding gains. When using other perceptual quality metrics (e.g. SSIM or VMAF) to measure compressed video quality, the corresponding BD-rate can be also computed using a similar way discussed above [9]. It is noted that the BD-rate thresholds, at which the visual quality can be differentiated between two codecs, are highly content dependent. Generally, the perceptual quality improvement for video content with fewer textures is more likely to be identified for similar BD-rate values [9].

It can be observed in Table 2.1 that H.266/VVC VTM 7.0 has achieved significantly better coding performance compared to H.264/AVC JM 19 (61% of BD-rate), H.265/HEVC HM 16.20 (32.5% of BD-rate), and AOM AV1 codec (14.7% of BD-rate). In this thesis, the HEVC HM 16.20, VVC VTM 7.0 and AV1 will be used as host codecs for the developed CNN-based coding tools, and they will also be employed as anchors for performance comparison.

2.2 Deep Learning

Over the past decade, machine learning (ML) techniques, especially deep learning methods based on advanced deep neural networks (DNNs) have provided revolutionary advances across various computer vision applications, in particular for image/video processing and understanding [22, 21, 41]. The structures of DNNs were inspired by the information processing and neuron structures in the biological systems [42, 43]. Their performance is progressively improved through the iterative training process using volumes of training material. The DNNs have powerful non-linear modelling and data representation abilities and can deal with many complex vision tasks including classification, recognition, etc. [44, 21, 41]. With the fast development of high-performance computing techniques and the significantly increased data storage capacities, these data-driven deep learning methods have also been increasingly applied in other areas, including intelligent assistants, marketing and finance, and the decision-making process [9].

In this section, we first introduce the basic theories of the DNNs. Then, two typical DNNs-Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) are briefly reviewed. Finally, several popular CNN and GAN architectures are described, which are for applications of deep image and video restoration.

2.2.1 Basic Theory

The commonly used modern DNNs contain a large number of highly interconnected artificial 'neuron' units which comprise various trainable parameters to mimic human behaviours with abilities to deal with some complex tasks (e.g. object recognition, classification) that the human can do [42]. The parameters of artificially designed neurons can be progressively and adaptively updated through the network's 'learning/training' operations which use large amounts of input data (denoted as training data) [44, 42, 43]. There are several types of training methodologies, including supervised learning, semi-supervised learning, self-supervised learning, etc. In this thesis, we focus on the supervised learning strategy.

In supervised learning, the pre-defined differentiable loss functions (e.g. $\ell 1$ or $\ell 2$ loss) compute an error measurement based on the target/labelled data (i.e. ground truth) and output of DNNs. The gradients of this error signal are then calculated and back-propagated to update parameters of the network using a stochastic gradient descent algorithm [45, 46]. This algorithm attempts to minimise the error across the training data and iteratively updates the network parameters until an optimal model is obtained for testing [44, 43]. The notable stochastic gradient descent algorithms include Adam [47], ADADELTA [48], Nadam [49] and AMSGrad [50]. The Adam optimiser is the most commonly used method in CNN-based

image and video restoration approaches [41, 23], which was developed based on the two previous algorithms, Adaptive Gradient Algorithm (AdaGrad) [51] and Root Mean Square Propagation (RMSProp) [52]. It effectively increases the convergence speed of networks and stabilises training process [47]. In this thesis, it will also be utilised to train the CNN models for video compression enhancement.

2.2.2 Convolutional Neural Networks (CNNs)

The first class of the most commonly used deep neural networks is the Convolutional Neural Networks (CNNs), which were originally proposed for written zip code recognition [46]. The CNN architectures are comparable to the highly interconnected neurons' structures in the human brain and the organisation of the animal visual cortex [43], where, the individual cortical neuron's responses to external environment stimuli are processed within a restricted area of the visual field (animal or human) which is denoted as the receptive field. The final response of the entire visual area can be obtained by collecting and fusing the different responses from those individual receptive fields [43].

CNNs stimulate the 'responses to the external stimuli' within the human visual system by performing 2D or 3D convolutional operations on the input data (e.g. image blocks) to progressively extract spatial or spatio-temporal features. Kernels (e.g. 3×3) act as convolution filters and stimulate the receptive field of the convolutional layer [43, 9]. In each convolution filter, the kernel weights are important parameters which are trainable and their values can be iteratively updated in the CNN training process. In practice, the CNNs generally contain a number of concatenated convolutional layers, each of which comprises a set of convolution filters with the same kernel size but with different kernel weights. Each convolutional layer in CNNs processes and extracts features output from the previous layer. This allows later layers to have a greater field-of-view and their features become more informative.

Following each convolutional layer, other operations such as the pooling layers (e.g. average/max pooling), activation functions (e.g. ReLU, Leaky ReLU), batch normalisation layer (BN) can also be employed to further process feature maps from the previous layer. The pooling layers are designed to reduce the size of feature maps which can effectively reduce network complexity and mitigate overfitting problems [43]. The activation functions mainly introduce non-linearities in the networks which are beneficial for learning complex patterns in the data [53]. The batch normalisation technique, which was first proposed in 2015 [54], generally standardises the inputs of a convolutional layer to stabilise training process of the deep CNNs. However, it could lead to gradient explosion especially in the very deep CNN models [55]. In the end, fully connected layers can be adopted as the final stage

of the networks to map the extracted features to specific output requires in various learning tasks [43, 9].

Regarding the CNN training, the stochastic gradient descent-based error backpropagation methods are typically used, which update the weights of convolution filters employed in the networks based on the gradients of the local error surface (produced by the pre-defined loss functions) [43, 9].

As discussed in Chapter 1, this thesis focuses on enhancing CNN-based video compression enhancement tools. This often employs the CNN models which are developed for image and video restoration (e.g. super-resolution, denoising). These existing networks have a similar backbone structure which includes three primary stages: shallow feature extraction (Stage 1), deep feature processing (Stage 2) and the final reconstruction (Stage 3). Figure. 2.1 shows the diagram of the basic structure of the CNN model in the context of image and video restoration applications. At Stage 1, one or two concatenated convolutional layers are generally employed to extract shallow (low-level) features from the input image block. Then, Stage 2 adopts several convolutional layers along with the advanced network structures (such as residual learning, residual dense connections, feature review, etc.) to progressively process and extract deep (high level) features. Finally, Stage 3 always utilises several concatenated convolutional layers to reconstruct final image block with higher quality based on the previously extracted high-level feature information. It is noted that most of the previous CNN models focus on optimising Stage 2 to improve the overall performance of the networks [7].

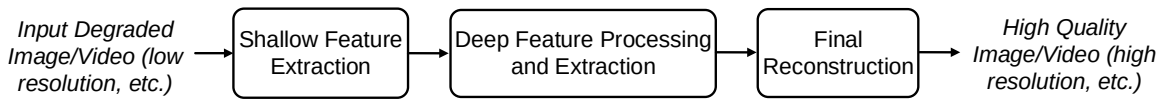


Figure. 2.1 Basic CNN structure for image and video restoration.

The common CNN networks used for image and video restoration primary include the following features: (i) concatenated convolutional layers [56] (ii) concatenated residual blocks [57]; (iii) residual dense connections [2]; (iv) cascading connections [4]; and (v) feature review structures [58]. More recently, new advanced structures have also been proposed including channel and spatial attention mechanisms [59] and non-local feature extraction [60]. Their primary implementations, which have been widely used in image super-resolution, restoration and video compression, have been summarised below and will also be used in this thesis for benchmarking. These include 14 typical CNN structures and 9 GAN architectures. In this section, we briefly describe these 14 popular CNN architectures. Their names, original literature and primary features are summarised below [3, 7]:

- **SRCNN** [61] is the first CNN model designed for single image super-resolution (SISR). It employs a simple network structure with only 3 convolutional layers.
- **FSRCNN** [62] was also developed for SISR, containing 8 convolutional layers with various kernel sizes.
- **VDSR** [63] contains 20 convolutional layers employing global residual learning to achieve enhanced performance. However, it does not employ residual blocks [55], which may lead to unstable training and evaluation performance [64].
- **SRResNet** [28] was the first network structure with residual blocks designed for SISR, improving the overall performance and stability of the network.
- **DRRN** [64] employs a recursive structure and also contains residual blocks for SISR.
- **EDSR** [65] significantly increases the number of feature maps (256) for the convolutional layers in each residual block. It offers improved overall performance but with much higher computational complexity.
- **RDN** [66] was the first network architecture to combine residual block and dense connections [67] for SISR.
- **ESRResNet** [29] enhances SRResNet by combining residual blocks with dense connections, and employs residual learning at multiple levels. It also removes the batch normalisation (BN) layer used in SRResNet to further stabilise training and reduce artefacts.
- **RCAN** [68] incorporates a channel attention (CA) scheme in the CNN, which better recovers high frequency texture details.
- **MSRResNet** [1] modified SRResNet by removing the BN layers for all the residual blocks. This network structure has been employed in several CNN-based coding algorithms and has been reported to offer significant coding gains [57].
- **CARN** [69] was the first network architecture to combine the cascading connections and residual blocks for SISR task.
- **UDSR** [70] integrated the U-shape structure with deep residual learning to achieve improved performance for SISR.
- **HR-EnhanceNet** [71] utilises the U-shape structure along with two modified HRNets [72].

- **RNAN** [73] was the first network architecture to combine non-local operation [60] with residual learning structures.

It is noted that the performance of a CNN is generally related to three primary factors: *depth* (i.e. the depth of networks), *width* (i.e. the number of feature maps), and *cardinality* (i.e. the size of transformation sets) [43, 74, 75]. All of the network architectures described above have been designed to increase the network *depth* [63, 28, 29, 66, 68, 73, 1, 76] and *width* [74, 67, 65, 66, 29, 77, 76], while only a few of them exploit the *cardinality* characteristic [78, 75, 79, 80]. However, *cardinality* is widely acknowledged to be a more effective way to improve overall performance and network capacity compared to the other two factors [75]. We also note that the kernel size of convolutional layers in these existing networks is usually set at a fixed value (3 in most cases), which may limit the receptive field size and hence the overall network performance [43].

2.2.3 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) were first proposed by Goodfellow *et al.* [81] in 2014, which consist of two modules (a generator and a discriminator) as shown in Figure. 2.2. In the first training stage, the generator is trained independently following a typical CNN training methodology. It is then trained again alongside the discriminator using the adversarial training strategy [28, 29], which effectively enhances the final performance of the generator. For image and video restoration tasks, GAN-based approaches can produce image and video content with higher visual quality and more realistic textural detail compared to methods based on conventional CNNs [76].

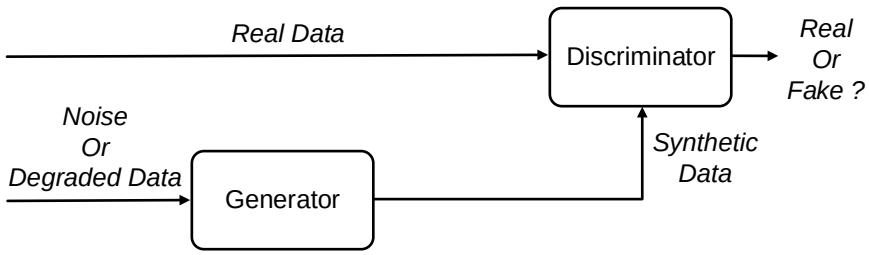


Figure. 2.2 Basic GAN structure.

Notable examples of GANs which have been developed for image and video restoration algorithms are summarised below:

- **ADGAN** [82] combines the U-shape structure with simple convolutional layers and trains the network using the standard GAN [28].

- **SRResCGAN** [83] employs multiple residual blocks to realise SISR based on the RaGAN training methodology [29].
- **SRGAN** [28] was the first network to combine the standard GAN training methodology with perceptual loss functions (VGG19 [84]) for photo-realistic SISR.
- **PCARNGAN** [85] employs cascading connections and residual learning blocks, and trains the network with the standard GAN methodology [28].
- **RCAGAN** [77] modified SRResNet by replacing the original residual blocks with residual channel attention blocks, and trained the network using the conditional GAN (cGAN) methodology [86].
- **ESRGAN** [29] employs the Relativistic GAN algorithm [87] to train the ESRResNet for SISR.
- **RCAN-GAN** [88] trained the RCAN [68] using both standard GAN [28] and RaGAN training methodology [29].
- **PatchESRGAN** [89] trained the ESRGAN [29] using the PatchGAN algorithm [90].
- **RFB-ESRGAN** [76] modifies the ESRGAN [29] by replacing multiple residual-in-residual dense blocks with receptive field dense blocks (RFBs) and trained the network using the RaGAN methodology [29].

It is noted that although GAN-based architectures can provide improved enhancement performance based on perceptual quality metrics comparing to non-GAN-based CNN algorithms, they also have the following disadvantages [81, 43]: (1) higher training complexity and a larger GPU memory size is required due to the joint training of both generator and discriminator; and (2) non-stable training process due to the adversarial training strategies.

2.2.4 Applications and Approaches

Due to the powerful non-linear modeling and representational abilities, DNNs especially CNNs and GANs have provided revolutionary advances across various computer vision tasks, in particular for image/video understanding and restoration [22, 41, 91]. Typical image/video understanding applications include: image/video segmentation [92, 93], classification [94, 95], recognition [55, 96], object detection [97, 98], tracking [99, 100], localisation [101, 102], pose estimation [103, 104], and optical flow estimation [105, 106]. For image/video restoration, the CNNs/GANs are utilised to restore the high quality images/videos from

the corresponding degraded/distorted low quality counterparts, with typical applications including: super-resolution [107, 108], denoising [109, 110], deblurring [73, 111] and video frame interpolation [112, 113].

2.3 Deep Learning-based Video Compression

Deep neural networks have played an important role in video compression, both in terms of enhancing individual coding tools within conventional codecs, and also in providing new end-to-end compression via auto-encoder architectures [9, 4, 7]. In this section, we firstly review the commonly used training and test databases and then introduce two types of deep video compression algorithms (i.e. learning-based coding frameworks and CNN-based coding tools). Finally, the review of typical training methodologies for deep video compression enhancement is further presented.

2.3.1 Training and Test Databases

Training Databases

Training databases are a critical component for optimising the performance of machine learning-based algorithms. A well-designed training database can ensure good model generalisation and avoid potential over-fitting problems [114, 115]. Currently, there are few publicly available databases, which are specifically designed for learning-based video coding. Researchers, to date, have typically employed training databases developed for other purposes (such as super-resolution, frame interpolation and classification) for training. Notable publicly available image and video training databases are summarised below.

- **BSDS** [116] is an image database originally developed for image segmentation. It contains 500 RGB images, and has been used to train CNN-based loop filters [117] for video coding. Comparing to DIV2K, BSDS has fewer source images and lower spatial resolution (481×321).

- **ImageNet** [118] is a large image database primarily designed for visual object recognition. It contains more than 14 million RGB images at various spatial resolutions (up to 2848p) covering a wide range of natural content. It has also been used as a training database for single image super-resolution [28].

- **DIV2K** [119] contains 1000 RGB source images with a variety of content types, which was firstly developed for super-resolution. It has currently been employed as training

material by several JVET proposals [120, 121] and many other CNN-based coding algorithms [122, 123].

- **UCF101** [124] is a large video training database initially designed for human action recognition, and has been frequently used for training CNN-based temporal frame interpolation and motion prediction approaches [112, 125, 126]. It contains 13320 videos collected from YouTube, which consist of 101 types of human actions. All the sequences in UCF-101 have a relatively low spatial resolution of 320×240.

- **Kinetics** [127–130] is a large video dataset combining all of the sequences from the Kinetics-400 [127], Kinetics-600 [128], Kinetics-700 [129] and Kinetics-700-2020 [130] datasets. It has been commonly used to train deep CNNs for human action recognition. It contains 650,000 video clips collected from YouTube covering 700 action classes. The sequences in this database have various spatial resolutions up to 480×360.

- **Vimeo** [131] is a video database originally developed for training CNN-based optical flow and temporal super-resolution approaches. It contains 89,800 sequences at spatial resolutions up to 448×256. A constraint is imposed on motion vector magnitudes between any two adjacent frames and content with dynamic textures has not been included in this database. Vimeo has not been frequently employed for deep learning-based coding approaches and in particular has not been used for those approaches that exhibit superior improvements over standard video codecs (e.g. HEVC and VVC) [24].

- **Moments in Time** [132] is a large-scale video dataset primarily used for training CNNs to recognise and understand actions in videos. This dataset contains over 1,000,000 labelled video clips in 339 action classes at the spatial resolution of 340×256.

- **YouTube UGC** [133] is also a large-scale video database containing user-generated content (UGC) collected from YouTube for video quality assessment. It contains 1500 UGC sequences at various spatial resolutions from 360p to 1080p.

- **Tencent Video Dataset** [134] is a video dataset developed by the Tencent Media Lab, which has been used for training and/or testing CNN-based video coding tools and other computer vision tasks such as object detection and tracking. At the current stage, this dataset contains 86 source sequences covering a variety of content types. All the video sequences are in YCbCr 4:2:0 format at a spatial resolution of 3840×2160, 65 frames and bit depth of 8 bit or 10 bit.

- **CD** (Combined Database) in [135] is a video database combining source content from the LIVE Video Quality Assessment Database [136], MCL-V Database [137] and TUM 1080p Database [138] and has been employed to train CNN-based super-resolution approaches [135]. It contains 29 sequences at two different spatial resolutions, 1920×1080 and 768×432.

- **VideoSet** [139] is a video database proposed for quality assessment, which contains 880 source videos at four different spatial resolutions from 360p to 1080p.

- **REDS** [140] is a video database developed for training video super-resolution algorithms [141], which contains 300 video clips with spatial resolution 1280×720.

- **HIF** [142] is a video database primarily developed for training CNN-based in-loop filtering and post-processing algorithms. It contains 182 source video sequences with various spatial resolutions up to 2048×1080.

Modern video coding algorithms are required to process content with diverse texture types at high spatial resolutions and bit depths. For example, the standard test sequences included in the JVET Common Test Conditions (CTC) dataset include video clips at UHD resolution (2160p) at a bit depth of 10 bits, with various static and dynamic textures[1]. However many training databases mentioned above do not contain image or video content with high spatial resolution and bit depth, and most do not include any dynamic texture content.

Test Databases

Compared to training databases, test datasets are also required to evaluate the performance of coding algorithms, which include much few video sequences but also with diverse video content types and scenes. Associated with the primary coding standards mentioned in Section 2.1, there are three commonly used standard test databases including JVET-CTC SDR, UVG and o-1-f.

- JVET-CTC SDR Test Database: The JVET Common Test Conditions (CTC) SDR (standard dynamic range) test database [39] contains nineteen video sequences with YCbCr 4:2:0 format (10 bit) at various spatial resolutions of 2160p (6 sequences), 1080p (5), 480p (4) and 240p (4).

[1] Here, we follow the definition of textures in [143–145]. Static textures are associated with rigid patterns undergoing simple movement or subject to camera movement, while dynamic textures have complex and irregular movements, e.g. water, fire or steam.

- UVG Dataset: The UVG dataset [146] contains sixteen 4K (3840×2160) 10 bit YCbCr 4:2:0 video sequences. It was developed for video codec analysis and development. These sixteen video sequences cover wider ranges of video features including SI (spatial information) and TI (temporal information) the JVET-CTC SDR test database [146].

- Objective-1-fast Dataset: The *objective-1-fast* (o-1-f) [15] is the AOM main test dataset which has been utilised to evaluate the performance of the AV1 codec. This dataset contains thirty 8 bit YCbCr 4:2:0 video sequences at various spatial resolutions (twelve natural 1080p sequences, four 1080p screen content sequences, seven 720p sequences and seven 360p sequences). All of these test clips have 60 frames.

In this thesis, we use the JVET-CTC SDR dataset as the main test set to evaluate the coding performance of the presented algorithms. The other two databases, UVG and o-1-f are also employed occasionally to offer further validation for our methods.

2.3.2 Learning-based Coding Architectures

One important class of learning-based coding approaches implement the whole image and video coding framework using CNNs, which enable end-to-end training and optimisation processes [147, 3]. In 2016, Ballé *et al.* [148] firstly proposed a general nonlinear transform-based framework for image coding. This coding framework consists of an encoder and a decoder, both of which are based on CNN architectures containing several concatenated convolutional layers, novel generalised divisive normalisation (GDN) and non-linear activation functions. All of the encoder and decoder network parameters are jointly trained based on a rate-distortion optimisation method. This approach has outperformed JPEG and JPEG 2000, offering evident coding gains based on PSNR and MS-SSIM (multi-scale structural similarity) [27]. It also provides coding performance improvement over the HEVC intra coding (Better Portable Graphics-BPG) [149]. More recently, this framework has been extended to video compression based on optical flow-based motion estimation between adjacent frames [150–152], which has achieved comparable coding performance to the HEVC fast implementations (HEVC x265 with very fast mode). Additionally, inspired by the conventional hybrid video codecs, several approaches follow similar coding frameworks, and employ multiple CNNs to implement each coding tool in original hybrid codecs (e.g. transform, motion estimation and compensation, etc.) [153, 147]. These approaches have been reported to provide coding gains over the HEVC fast implementation x265 (very fast mode). Other notable end-to-end image and video compression algorithms include [154–157].

It should be noted that although these end-to-end solutions demonstrate significant potential [7], they still cannot compete with the reference models of the latest coding standards, such as HEVC HM, VVC VTM and AV1.

2.3.3 CNN-based Coding Tools

The other class of deep learning-based video compression algorithms are designed to enhance individual coding tools integrated into the standard codec configuration. Such approaches have been used to optimise tools including: intra prediction [158, 159], motion estimation [160, 161], transforms [162, 163], quantisation [164], entropy coding [165, 166], post-processing [56, 167] and loop filtering [117, 4]. New coding tools such as format adaptation [168, 57, 2] and virtual reference frame optimisation [161] have also been proposed typically with even higher coding gains. Due to the high computational complexity and the large GPU memory requirements associated with CNN computation [5], none of these methods has been adopted in the first version of VVC. More recently, JVET has started to conduct research work (Ad-hoc group 11) on neural network-based video coding tools for VVC [37, 169, 36]. Some of these tools have been demonstrated great potential with evident coding improvement over the original VVC.

Among these CNN-based coding tools, there is one group of methods, which typically offer more significant coding gains compared to the rest [23, 24]. These approaches typically apply CNN operations at the decoder to enhance the quality of reconstructed video frames. Typical examples include post-processing (PP) [170, 5], in-loop filtering (ILF) [171, 117], spatial resolution adaptation (SRA) and effective bit depth adaptation (EBDA) [57]. In this thesis, we mainly focus on developing novel and effective deep learning techniques to enhance these four compression enhancement modules.

Coding Module 1 (Post Processing - PP)

Lossy compression often introduces various visible artefacts such as blocking mismatches, banding and blurring, especially when large quantisation steps are employed during the encoding process. These unpleasant distortions can be mitigated by filtering the reconstructed frames. When this enhancement process is performed outside of the encoding loop (generally after decoding), it is referred to as post-processing (PP) [5]. The coding workflow for PP is illustrated in Figure. 2.3. This approach is commonly applied at the decoder, on the reconstructed video frames, to reduce compression artefacts and enhance video quality. When a CNN-based approach is employed, the network takes each decoded frame as input and outputs the final reconstructed frame with the same format [3].

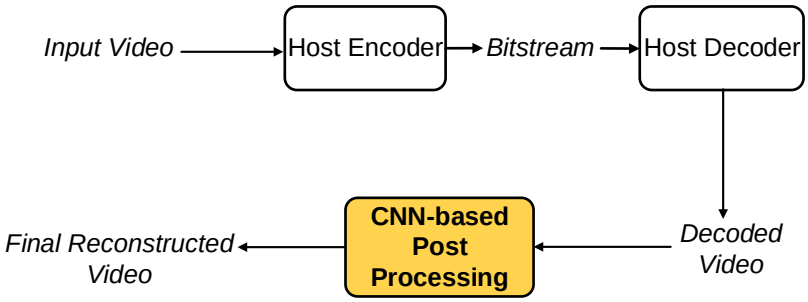


Figure. 2.3 Coding workflow with a CNN-based PP module.

Coding Module 2 (In-loop Filtering - ILF)

In-loop filtering applies processing at both the encoder and the decoder on the reconstructed frames, and the output can be used as reference for further encoding/decoding. As shown in Figure. 2.4 (an encoder architecture with a CNN-based ILF module), the CNN-based ILF module is located after the conventional in-loop filtering process, and has the same input and output format as for the PP case [142, 117]. This is similar to that for most of the previous contributions on CNN-based ILF filters [171, 172]. Compared to other possible designs, where the CNN operation is not performed as the last step in the whole coding workflow (e.g. before ALF, SAO or DBF), this implementation will not conflict with the existing conventional loop filters and will achieve better reconstruction performance due to its end-to-end optimisation in the training process [3, 4]. It is noted that the ALF in-loop filter is only available in VVC VTM. The basic principle of ILF is to utilise CNN model to further enhance the quality of the reconstructed frame before restoring it into the reference memory. This enhanced reference frame can effectively improve inter-prediction performance (e.g. improve motion estimation and compensation accuracy). This leads to the reduction of residual signals' strength and improvement of the overall coding performance of the codecs [117, 4].

Coding Module 3 (Spatial Resolution Adaptation - SRA)

CNN-based spatial resolution adaptation (SRA) down-samples the spatial resolution of the original video frames for encoding, and reconstructs the full resolution during decoding through CNN-based super-resolution. This approach can be applied at Coding Tree Unit level [168] or to the whole frame. In this thesis, only the frame-level SRA was implemented [173, 3], as shown in Figure. 2.5. In this case, the original video frames are spatially down-sampled by a fixed factor of 2, using the Lanczos3 filter. In order to obtain a similar bitrate as achieved by encoding full resolution video frames, the QP values utilised to encode

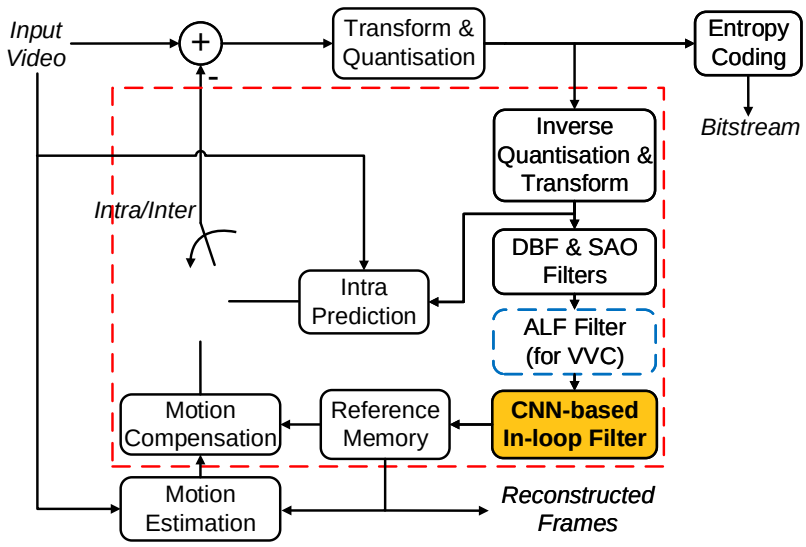


Figure. 2.4 Coding workflow with a CNN-based ILF module.

spatially-downsampled video frames need to be adjusted using a fixed QP offset (set to -6 in this thesis). The CNN-based super-resolution module processes the compressed and down-sampled video frames at the decoder to generate full resolution reconstructed frames. It is noted that a nearest neighbour filter is firstly applied to the reconstructed down-sampled video frame before CNN operation [57].

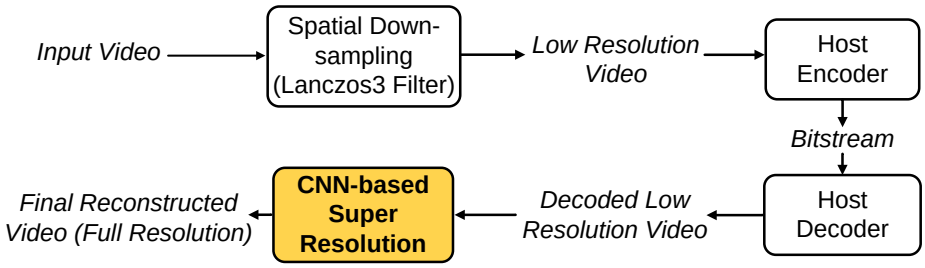


Figure. 2.5 Coding workflow with a CNN-based SRA module.

Coding Module 4 (Effective Bit Depth Adaptation - EBDA)

Similar to the case for spatial resolution, bit depth can also be adapted during encoding in order to achieve improved coding efficiency. Here Effective Bit Depth (EBD) is defined as the actual bit depth used to represent the video content, which may be different from the Coding Bit Depth (CBD) that represents the pixel bit depth, e.g. *InternalBitDepth* in HEVC reference encoders. This process is demonstrated in Figure. 2.6. In this thesis, we have fixed

CBD at 10 bits, the same as in the Main10 profile of HEVC and VVC. Before encoding, the EBD of the original video frames is down-sampled by 1 bit through bit-shifting. The host encoder then compresses the video frames with reduced EBD (the CBD remains the same) to produce the bitstream. When receiving the bitstream, the host decoder reconstructs the reduced EBD video frames and applies the CNN-based up-sampling to obtain the final reconstructed frames with full EBD [3, 2]. It is noted that a fixed QP offset of -6 is also applied to base QP values when encoding the low bit depth version of video frames in EBDA.

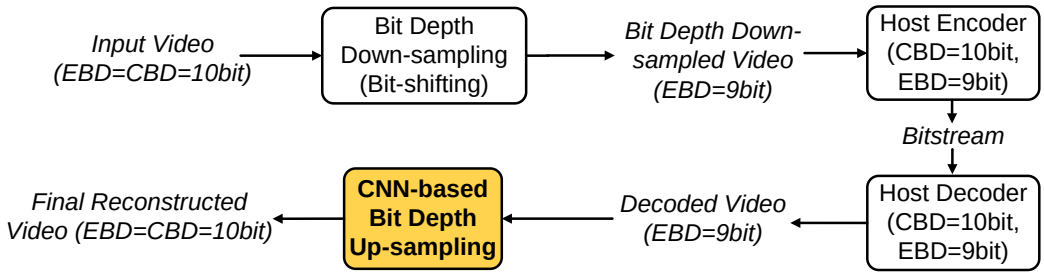


Figure. 2.6 Coding workflow with a CNN-based EBDA module.

In the context of CNN-based video coding enhancement, existing methods often employ existing or modified networks with relatively simple architectures (such as the VDSR [56, 173, 117]). These do not contain advanced structures (e.g. dense connections, feature review structures), and therefore cannot deal with complex video textures and high compression artefacts.

2.3.4 Typical Training Methodologies

In most deep learning-based coding enhancement tools, $\ell 1$ or $\ell 2$ loss is used to train the CNN models with the aim of minimising pixel-wise distortions. Alternative training strategies have been proposed to improve perceptual image quality, typically based on GAN architectures and loss functions combining $\ell 1/\ell 2$ loss, feature map differences (e.g. VGG19-54 [29]) and low-complexity quality metrics (e.g. MS-SSIM and SSIM). Notable examples include approaches using standard GANs [28, 82], Relativistic average GANs (RaGANs) [29, 85, 1, 2, 5, 83, 88, 76], conditional GANs (cGAN) [77], Patch GANs [89], and Wasserstein GAN-gradient penalty (WGAN-GP) [174].

It is important to note that the $\ell 1$ and $\ell 2$ losses do not correlate well with subjective video quality [175–177], and the combined loss functions employed in these GAN-based training strategies use artificially configured combining weights, which have never been fully evaluated in terms of their correlation with subjective video quality. These issues inevitably lead to sub-optimal training performance when the networks are utilised for compression

application. The primary features of the popular GAN training methodologies discussed above are summarised below.

- **Standard GANs** [28, 82]: In [28], researchers proposed SRGAN model which was the first standard GAN architecture designed for single image super-resolution. SRGAN's discriminator consists of several concatenated convolutional layers to produce a scalar which is the probability directly predicting the input image block is real or fake. Based on the discriminator's output, the adversarial losses (i.e. entropy-based losses) can be further computed to jointly train the generator and discriminator through the adversarial training methodology. This standard GAN method has been recently used in [82] for image denoising and quality enhancement purposes.

- **Relativistic average GANs (RaGANs)** [87]: RaGANs were primarily proposed for image generation purposes, which also utilise conventional entropy-based loss in adversarial training stage. Different from the standard GANs, the RaGANs' discriminator predicts the probability that a real image is relatively more realistic than a fake one. In this case, the generator of RaGANs can benefit gradients from both real and fake data samples. This is different from the standard GANs (e.g. SRGAN) where only the fake data takes effect in the generator training process. RaGANs have been widely utilised in various image/video restoration and deep video compression enhancement tasks [29, 85, 1, 2, 5, 83, 88, 76], and effectively improved visual quality of image/video content compared to the standard GANs [29].

- **conditional GANs (cGANs)** [178, 77]: cGANs also employ an adversarial training methodology with entropy-based losses. In the training stage, different from standard GANs which only accept one data sample (real or fake) in probability computation process, cGANs' discriminator receives an additional input (the input of the generator). This structure has been reported to offer performance improvement for both generator and discriminator [178], when it been utilised for super-resolution and noise reduction [77].

- **Patch GANs** [90, 89]: Patch GANs also use an entropy-based adversarial training methodology [90], where, the real or fake image is firstly segmented into several non-overlapped $N \times N$ local patches. The discriminator then accepts each $N \times N$ local patch as input and distinguishes if it is real or fake based on the entropy loss. The average entropy loss is also calculated among all the patches of an image block, and is considered as the adversarial loss in the training process.

- **Wasserstein GAN-gradient Penalty (WGAN-GP)** [179, 174]: WGAN-GP [179] was built on the original WGAN [180], which employs a Wasserstein distance-based adversarial training methodology. The discriminator of both WGAN and WGAN-GP was designed to distinguish real and fake data by measuring the difference between the probability distributions of the real and fake data based on their 1-Wasserstein distance. Different from the original WGAN, an additional gradient penalty is employed in WGAN-GP which can stabilise adversarial training process and further optimise network performance [179]. WGAN-GP has been utilised in [174] for single image restoration.

2.3.5 Complexity Issue

As discussed above, CNN-based video compression enhancement tools have demonstrated significant potential for improvements in coding efficiency compared to the original standard video codecs and other coding tools without using deep learning techniques. However, these methods suffer from high computational complexity due to the CNN operations, especially when CNN-based methods are employed at the decoder [6]. This is one of the most challenging issues for this type of approach and has been previously addressed from three main aspects [24, 6]:

- Development of fast and lightweight CNN architectures (maintaining high network performance) for deep video compression applications. An alternative approach is to simply modify the existing network architectures to reduce the relative complexity of the networks, for example by reducing the number of residual blocks utilised in the network for PP application to obtain the trade-off between the coding efficiency and relative decoding complexity [181, 5].

- Development of low complexity CNN-based coding modules supporting the flexible allocation of complexity between the encoder and decoder and achieving better trade-off between the computational costs and coding performance [6].

- Improvement of hardware systems with better high-performance computing equipment effectively supporting integration and use of deep CNNs in terminal devices.

2.4 Video Quality Assessment

In this section, we review the video quality assessment methodologies including the objective quality assessment algorithms and subjective quality experiment.

2.4.1 Objective Quality Metrics

Objective quality assessment algorithms played an important role in conventional video codecs as well as deep video compression. For learning-based video coding algorithms, quality metrics have been involved in both network training and evaluation processes. They can be (1) utilised as the loss functions in training process; and also be employed (2) as quality assessment methods for evaluating network performance.

Image and video quality metrics can be divided into three groups according to the availability of the reference sources [9]. (i) Full-reference (FR) metrics use both distorted and the corresponding original data for assessment. (ii) No-reference (NF) only evaluates the distorted content. (iii) Reduced-reference (RF) methods perform quality assessment based on partial information of the original data. In this thesis, we solely focus on FR metrics, as they are more commonly used for compression performance evaluation and in CNN training processes (as loss functions). Existing popular FR quality metrics are summarised as follows.

- MAD: Mean-absolute-difference (MAD) is a typical pixel-wise image quality metric which is widely used for motion vector estimation in standard codecs [9]. It can be easily obtained by calculating the average absolute pixel value differences between the reference and distorted images [9]. MAD has been widely utilised as the loss function ($\ell 1$ loss) in CNN training process for image and video restoration and most of CNN-based video compression enhancement algorithms [91, 23].

- MSE and PSNR: Mean-squared-error (MSE) is simply calculated by computing the average squared pixel differences between the reference and distorted images. The peak-signal-to-noise-ratio (PSNR) can be directly converted from MSE based on the peak signal value of an image and logarithm transformation [9]. MSE and PSNR are two typical quality metrics used in video compression. They have low computational complexity, but do not correlate well with subjective opinion scores [175, 182]. It is noted that MSE (denoted as $\ell 2$ loss) has been also used to train deep CNNs for image and video restoration, as well as the CNN-based video coding enhancement approaches [183, 7].

- SSIM [26]: Structure Similarity Image Metric (SSIM) is a popular full-reference image quality metric. It measures visual degradation of structure similarity based on a combination of three features extracted from both reference and distorted images. These include: (i) contrast, (ii) structure information, and (iii) luminance. SSIM has achieved better correlation performance over the PSNR with relatively low computational complexity.

- MS-SSIM: Based on the SSIM algorithm [26], Wang *et al.* [27] proposed a multi-scale SSIM (MS-SSIM) for image quality assessment. It combines three features (same with the SSIM) which are extracted from impaired and reference images at different scales (except luminance) to produce a quality index. MS-SSIM has effectively improved quality assessment performance compared to the SSIM [27].

- VSNR [184]: The visual signal-to-noise-ratio (VSNR) has been proposed for still image quality assessment. It exploits near and supra threshold properties of the human visual system (HVS). Specifically, the cortical decomposition of the HVS is stimulated using a wavelet filter followed by a two-stage approach to measure the detectability of distortions and produce the final measurement of visual SNR as the quality assessment result. VSNR has been reported to achieve competitive performance on LIVE image database [184].

- VIF [185]: Sheikh *et al.* proposed an advanced image quality metric, visual information fidelity (VIF), which utilises different models (e.g. Gaussian scale mixtures) to stimulate distortions and the HVS. Based on these models, two mutual information quantities can be calculated which are the amount of information that the HVS extracts from the reference and distorted images, respectively. Then, the ratio of these two visual information quantities is utilised as the quality assessment result. VIF has achieved improved performance on various published databases especially for those with supra-threshold distortions [185, 186].

- VQM: Pinson and Wolf proposed a quality metric (VQM) [187] for video quality assessment. It combines seven features (including blurring, jerkiness, global noise, block and colour distortions, and spatio-temporal features) extracted from the reference and impaired videos using the impairment filters to predict video quality. VQM has provided better correlation performance with mean-opinion-scores (MOS) on VQEG dataset compared to other metrics, and it has been utilised as an ANSI and ITU standard [9].

- MOVIE: Seshadrinathan *et al.* [188] proposed a video quality metric, motion-based video integrity evaluation (MOVIE). It employs spatio-temporal Gabor filters to decompose original and distorted videos. Then, both of the spatial and temporal quality components can be obtained based on the decomposition results. These spatio-temporal quality indices are finally combined to predict the quality of the distorted video. MOVIE has outperformed PSNR and SSIM on the VQEG FRTV Phase I database

[188]. However, it has higher computational complexity due to a large number of Gabor filters and temporal information required in computation process.

- PVM: Zhang and Bull [186] developed a perception-based video quality metric (PVM). This algorithm emulates the perceptual properties of the HVS by non-linearly fusing the noticeable distortions and blurring artefacts for video quality assessment. It has provided state-of-the-art correlation performance on both VQEG FRTV Phase I and LIVE databases compared to other popular quality metrics, such as VQM, SSIM, MOVIE and VSNR [186].

- VMAF: Video Multimethod Assessment Fusion (VMAF) [189] is a new learning-based assessment method, which combines multiple quality metrics and video features (including VIF [185], Detail Loss Metric (DLM) [190] and temporal frame difference [189]) using a Support Vector Machine (SVM) regressor. It has been reported to offer robust and state-of-the-art correlation performance with subjective opinion scores based on different video quality databases with various compressed content [175, 182].

Among these quality assessment methods, PSNR and VMAF were selected as the quality metrics for compression performance evaluation in this thesis. PSNR is the most commonly used metric for evaluating video compression performance [9], while VMAF has been demonstrated to achieve high correlation performance based on various subjective video quality databases with relatively low computational complexity [175, 182].

2.4.2 Subjective Quality Experiment

Alongside objective quality metrics, the perceptual quality of video content can also be measured through controlled psychophysical experiments. This type of approach is more accurate compared to objective methods, but is also more time and resource consuming.

In the literature, there are two primary test methodologies, which are commonly used for video quality assessment, double stimulus methods and single stimulus methods.

Double Stimulus Methods

Double stimulus methodologies are often used when both distorted and original versions are available [9]. There are two major double stimulus evaluation methods, including the Double Stimulus Continuous Quality Scale (DSCQS) and Double Stimulus Impairment Scale (DSIS).

- DSCQS is suitable for the tasks in which the qualities of both original and corresponding videos are similar, and it tests how well the evaluated algorithms/systems perform relative to the original. In each trial within the test, participants were shown Sequence A and Sequence B twice. One of them is a distorted sequence, while the other is the corresponding original version (impaired by the algorithms/systems). Their orders are randomly determined and unknown to the subjects. After viewing these two sequences, participants were asked to rate the perceived quality of both videos, based on a continuous quality scale (e.g. from 1 to 5 where, 1-Bad, 2-Poor, 3-Fair, 4- Good and 5-Excellent).

- DSIS is similar to the DSCQS methodology except that in each trial a pair of original and impaired content is presented to subjects only once, and viewers know the order of the original and distorted versions. DSIS is suitable for evaluating the robustness of the algorithms/systems in cases where artefacts are more noticeable [9]. Compared to DSCQS, DSIS commonly employs a labelled quality scale, which contains five grades: Imperceptible; Perceptible but not annoying; Slightly annoying; Annoying; and Very annoying [9].

Single Stimulus Methods

Single stimulus methods are typically used in scenarios where there are no explicit anchor clips or when distorted sequences are not expected to be directly compared to their original versions. In these methods, subjects are simply shown a number of test videos in random order. Two primary single stimulus methods are the absolute category rating (ACR) and the single stimulus continuous quality evaluation (SSCQE) method.

ACR is generally suitable for relatively short video durations (e.g. 10 s) and utilises a labelled grading scale (Excellent, Good, Fair, Poor, Bad) to collect subjective scores. SSCQE is used to evaluate quality changes due to temporal content variations in test sequences, which stimulates practical video delivery processes. Therefore, it is more relevant to longer clips with durations of more than 5 minutes.

It is noted that ITU recommends video sequence duration to be approximately 10 seconds [191] for most test methodologies (except SSCQE). However, the optimal video duration has been recently investigated by Mercer Moss *et al.* [192], who reported that sequences with 5 seconds can still offer similar evaluation accuracy as the recommended 10s. This can effectively reduce the overall length of subjective experiments.

2.5 Summary

This chapter presented an overview of video compression standards, basic deep learning techniques and learning-based video compression approaches. Moreover, primary video quality assessment methods have also been reviewed. Based on these, three issues will be addressed in the following chapters, which are related to training databases, network architectures and training methodologies.

Chapter 3

BVI-DVC: A Training Database

As discussed in Chapter 1 and Chapter 2, most deep learning-based coding methods have been trained on image or video databases [24], which were mainly designed for computer vision applications, e.g. super-resolution. Most of these databases do not provide sufficient content coverage and diversity. As a result, the generalisation of networks cannot be ensured in the context of video coding, and the optimum performance of employed CNN models has not been achieved when trained on these databases. Prior to BVI-DVC, there was no public dataset specifically developed for this purpose.

In this context, this chapter presents a new extensive and representative video database, denoted as BVI-DVC, for training CNN-based video coding algorithms, in particular those tools that enhance the performance of conventional compression algorithms. BVI-DVC contains 800 progressive-scanned video clips at a wide range of spatial resolutions from 270p to 2160p, with diverse and representative content. The experimental results show that this database produces significant improvements in terms of coding gains over five existing (commonly used) image/video training databases under the same training and evaluation configurations.

The work presented in this chapter has been published in [3].

3.1 Database Description

To develop a large and diverse video database for training deep learning-based coding algorithms, we first collected 280 UHD (3840×2160) sequences from publicly available video databases and public websites. A subset of these sequences was then selected based on the approach in [140] in order to provide optimal coverage for different scenes (e.g. nature and objects) and various texture types (e.g. static, dynamic, structural and luminance plain).

The selection operation is based on subjective observation and manual process, which is similar to that in [140, 142, 139].

Table 3.1 Key features of thirteen training databases including BVI-DVC.

Training Databases	Image or Video?	Seq. Number	Max Resolution	Bit depth	Various textures?
BSDS [116]	Image	500	321p	8	No
ImageNet [118]	Image	14M	2848p	8	No
DIV2K [119]	Image	1000	1152p	8	No
UCF101 [124]	Video	13,320	240p	8	No
Kinetics [127–130]	Video	650,000	360p	8	No
Vimeo [131]	Video	89,800	256p	8	No
Moments in Time [132]	Video	1M	256p	8	No
YouTube UGC [133]	Video	1500	1080p	8	No
CD [135]	Video	29	1080p	8	No
VideoSet [139]	Video	880	1080p	8	No
REDS [140]	Video	300	720p	8	No
HIF [142]	Video	182	1080p	8	No
BVI-DVC	Video	800	2160p	10	Yes

Finally, two hundred source sequences have been selected from different sources, including 69 sequences from the Videvo Free Stock Video Footage set [193], 37 from the IRIS32 Free 4K Footage set [194], 25 from the Harmonics database [195], 19 from BVI-Texture [196], 10 from the MCML 4K video quality database [197], 7 from BVI-HFR [198], 7 from the SJTU 4K video database [199], 6 from LIVE-Netflix [200, 201], 6 from the Mitch Martinez Free 4K Stock Footage set [202], 5 from the Dareful Free 4K Stock Video data set [203], 3 from MCL-V [137], 2 from MCL-JCV [204], 2 from Netflix Chimera [205], 1 from the TUM HD databases [206], and 1 from the Ultra Video Group-Tampere University database [207]. These sequences contain natural scenes and objects [140], e.g. mountains, oceans, animals, grass, trees, countryside, city streets, towns, buildings, institutes, facilities, parks, marketplaces, historical places, vehicles and colourful textured fabrics. Different texture types such as static texture, dynamic texture[1], structure content and luminance-plain content are also included. The BVI-DVC database also covers a large variety of motion types, including camera motion (e.g. pan, zoom, etc.), human actions (e.g. running, walking, etc.),

animal activity (e.g. tigers, horses, etc.), plant movements (e.g. trees, bushes, etc.), fluid motion (e.g. water, smoke, etc.) and other object actions (e.g. cars, cycles, etc.).

All these sequences are progressively scanned at a spatial resolution of 3840×2160, with frame rates ranging from 24 fps to 120 fps, a bit depth of 10 bit, and in YCbCr 4:2:0 format. All are truncated to 64 frames without scene cuts, using the segmentation method described in [192]. To further increase data diversity and provide data augmentation, the 200 video clips were spatially down-sample to 1920×1080, 960×540 and 480×270 using a Lanczos filter of order 3. This results in 800 sequences at four different resolutions. Figure. 3.1 shows the sample frames of twenty example sequences. The primary features of this database are summarised in Table 3.1 alongside those for the other twelve databases [118, 119, 116, 131, 127–130, 132, 133, 139, 142, 135, 140, 124] mentioned in Section 2.3.1.

Figure. 3.1 Sample frames of 20 example sequences from the BVI-DVC database.

3.2 Experimental Configurations

In order to evaluate the training effectiveness of the BVI-DVC database in the context of video compression, ten network architectures [61–65, 28, 29, 68, 66, 1, 208] were employed in conjunction with four CNN-based coding modules: post-processing (PP), in-loop filtering (ILF), spatial resolution adaptation (SRA) and effective bit depth adaptation (EBDA).

In terms of benchmarking databases, one image database (DIV2K) and four video databases (REDS, CD, VideoSet, and HIF) have been selected for comparison with BVI-DVC. DIV2K is selected because it has been used for training CNN models in multiple JVET contributions and many other CNN-based coding algorithms. REDS, CD, VideoSet and HIF were selected as they contain relatively diverse content at higher spatial resolutions compared to other datasets such as BSDS, Vimeo, Kinetics, Moments in Time and UCF101. YouTube UGC dataset has not been included which contains source content with imperfect quality. This is different from the high quality videos commonly used for evaluating the performance of coding algorithms. It is noted that the Tencent Video Dataset (described in Section 2.3.1) was published after BVI-DVC. Therefore, it has not been included for benchmarking the BVI-DVC database.

3.2.1 Test Coding Modules

Four typical CNN-based video compression enhancement tools, PP, ILF, SRA and EBDA have been utilised to evaluate training effectiveness of the BVI-DVC database. Their coding frameworks and primary features have been described in Section 2.3.3. These four coding modules were selected since they have been demonstrated to offer significant coding gains over standardised video codecs compared to other tools (e.g. inter prediction, entropy coding, etc.) and also outperform existing end-to-end solutions. In contrast to end-to-end deep image coding architectures, they are amenable to integration into standard codecs for practical applications. In this context, our evaluation is restricted to CNN-based coding tools for standard video codecs rather than new end-to-end solutions. The latter will be included in our future work.

3.2.2 Evaluated CNN Models

To evaluate our four coding modules, ten popular network architectures have been implemented. These were selected to reflect a range of CNN architecture types, including: (i) simple concatenated convolutional layers (SRCNN [61], FSRCNN [62] and VDSR [63]); (ii) deep residual blocks (SRResNet [28], EDSR [65], DRRN [64], and MSRResNet [1]); (iii) dense connections (RDN [66] and ESRResNet [29]); (iv) feature review structures (RDN [66]); and (v) channel attention mechanism (RCAN [68]). Most of these network structures were initially designed for super-resolution processing or image enhancement, and some have been employed in CNN-based coding approaches as described in Section 2.3.3. Their primary features have been summarised in Section 2.2.2.

In this experiment, we have employed architectures for these ten networks, that are identical to those reported in their original publications; we only modify the input and output interfaces in order to process content in the appropriate format. The input of all CNNs employed is a 96×96 YCbCr 4:4:4 colour image, while the output targets the corresponding original image block with the same size. The choice of block size was driven by the fact that larger sizes lead to higher computational complexity and increased GPU memory for some complex CNN models (e.g. ESRResNet, RCAN and RDN). In contrast, it has been reported [29] that larger input block sizes can slightly improve network performance. Given the GPU size used (NVIDIA P100 GPU with 16 GB memory) in this work, 96×96 was selected as a trade-off between complexity and performance. Similar sizes have also been employed in previous work such as [28].

The input block can be either compressed (for PP and ILF), compressed and EBD down-sampled (for EBDA) or compressed and spatial resolution re-sampled (for SRA - a nearest neighbour filter is applied before CNN processing). The same loss functions have been used as in the corresponding literature. All these ten networks have been re-implemented using the TensorFlow framework (version 1.8.0).

3.2.3 Training Data Generation

Five existing image and video databases are selected to benchmark the training effectiveness of BVI-DVC, including DIV2K [119], REDS [140], CD [135], VideoSet [139], and HIF [142]. The training data generation of four typical CNN-based video coding enhancement modules is described in detail below.

All the original images or videos in each database were first spatially down-sampled by a factor of 2 using a Lanczos3 filter or down-sampled by 1 bit through bit-shifting. The original content (for training PP and ILF CNNs), together with spatially down-sampled clips (for training SRA CNNs) and bit depth reduced sequences (for training EBDA CNNs) were then compressed by the HEVC Test Model (HM 16.20) based on the JVET Common Test Conditions (CTC) [39] using the Random Access configuration (Main10 profile) with four base QP (quantisation parameter) values: 22, 27, 32 and 37[1] (a fixed QP offset of -6 is applied for both spatially and bit depth down-sampled cases as in [209]). This results in three training input content groups for every database, each of which contains four QP sub-groups. For the input content group with reduced spatial resolution, a nearest neighbour filter was applied to obtain video frames with the same size as the original content.

[1]Here results with four QP values are generated due to the limited time and resource given. During the evaluation, if the base QP is different from these four, the CNN model for the closest QP value will be used.

For each input group and QP sub-group, the video frames of all reconstructed sequences and their original counterparts were randomly selected (with the same spatial and temporal sampling rates) and split into 96×96 image blocks, which were then converted to YCbCr 4:4:4 format. Block rotation was also applied here for data augmentation.

It is noted that similar training data generation methodologies described in this section for four typical CNN-based coding enhancement tools will also be used in the following chapters when new CNN architectures and training methodologies are trained.

3.2.4 Network Training and Evaluation

The training process was conducted using the following parameters: Adam optimisation [47] with the following hyper-parameters: β_1=0.9 and β_2=0.999; batch size of 16; 200 training epochs; learning rate (0.0001); weight decay of 0.1 for every 100 epochs. Based on the generated training content, four CNN models aligned with four QP groups were separately trained for each tested coding module and network architecture. This generates 480 CNN models for 4 training databases, 3 input content groups (PP and ILF use the same CNN models), 4 QP sub-groups and 10 tested network architectures. The reason for employing QP sub-grouping in network training process is mainly due to the fact that the different base QP values lead to different compression artefacts levels. Hence, separately training CNN models for different QP groups is beneficial for effectively optimising the reconstruction performance of networks [57]. Based on our experiments, QP sub-grouping achieves an average additional BD-PSNR gain of approximately 0.05 dB (content dependent) against using a single model in the context of PP coding module for VVC [5]. It is also noticed that although QP sub-grouping increases the training complexity, the computational complexity at the evaluation stage remains the same compared to using a single model. The only drawback is that more storage space is needed when multiple CNN models are used [5].

Based on the discussions above, the CNN models which are subsequently used in the evaluation stage for different base QP values (before applying QP offset for SRA and EBDA coding tools) are described as follows:

$$\text{CNN Models} = \begin{cases} \text{Model}_1, & \text{QP}_{\text{base}} \leq 24.5 \\ \text{Model}_2, & 24.5 < \text{QP}_{\text{base}} \leq 29.5 \\ \text{Model}_3, & 29.5 < \text{QP}_{\text{base}} \leq 34.5 \\ \text{Model}_4, & \text{QP}_{\text{base}} > 34.5 \end{cases} \tag{3.1}$$

During the evaluation stage, for a specific coding module, the decoded video frames (already up-sampled to the same spatial resolution if needed) are firstly segmented into

96×96 overlapping blocks as CNN input (YCbCr 4:4:4 conversion). The output blocks are then aggregated following the same pattern and then converted to YCbCr 4:2:0 format to form the final reconstructed frame.

Figure. 3.2 shows the diagram of the frame reconstruction method which was commonly used in CNN-based image and video restoration algorithms [28, 29, 210, 173] to avoid boundary effects caused by the convolutional layers. Here, four main block aggregation scenarios (the ways of removing overlapped pixel) are introduced below to illustrate this frame reconstruction algorithm based on five overlapped blocks, $B_{1,1}$, $B_{1,2}$, $B_{1,3}$, $B_{2,1}$ and $B_{2,2}$ as shown in Figure. 3.2. The rest of blocks within the input frame can be processed and aggregated in the similar ways that are introduced below to obtain the final reconstructed frame.

As shown in Figure. 3.2, the input decoded frame is first segmented into a total number of $M \times N$ overlapped blocks. After feeding each block into the CNN for enhancement, there are four main aggregation scenarios in this approach according to different block locations.

- For $B_{1,1}$ which is located at the first row and first column of the input frame, the 8 pixels individually from the rightmost and bottom boundaries of the processed block (shown as the blue area in Figure. 3.2) are removed. Then, the rest of content within the filtered block (shown as the white area in Figure. 3.2) is utilised as the first block ($B'_{1,1}$) of the final reconstructed frame.

- $B_{1,2}$ (located at the first row and second column of the input frame) is overlapped with $B_{1,1}$ along the width axis with an overlap size of 12 pixels. In order to avoid boundary effects and continuously joint frame content, 4 pixels individually from the leftmost and rightmost boundaries alongside 8 pixels from the bottom boundary of the CNN processed block are removed. The rest of content is then used as the second block ($B'_{1,2}$) of the final aggregated frame. As shown in Figure. 3.2, from $B_{1,3}$ (except blocks which are located at the boundaries of the decoded frame, e.g. $B_{1,N}$, $B_{2,1}$, $B_{2,2}$, $B_{M,N-1}$ and $B_{M,N}$), the segmented blocks have an overlap size of 8 pixels with adjacent blocks along the width and/or height axes/axis and will be assembled in the reconstructed frame using the similar method introduced above.

- For $B_{2,1}$ (located at the second row and first column of the input frame), 4 pixels individually from the top and bottom boundaries along with 8 pixels from the rightmost boundary of the filtered block are removed. Then, the rest of content ($B'_{2,1}$) is further aggregated in the final reconstructed frame.

- For $B_{2,2}$ (located at the second row and second column of the input frame), it is overlapped with both $B_{2,1}$ and $B_{1,2}$ along the width axis (12 pixels) and the height

axis (12 pixels) respectively. After CNN processing, 4 pixels from each boundary of the processed block are removed simultaneously and the rest of region ($B'_{2,2}$) is then aggregated in the final reconstructed frame.

Based on the discussions above, it can be observed that during the evaluation, the input decoded frame is first segmented into several overlapped blocks with 12 and/or 8 overlap pixels along the width and/or height axes/axis according to their locations within the input frame. Then, the trained CNN model is employed to process these overlapped blocks separately. After that, several pixels (8 or 4 pixels) will be further removed from the filtered block to avoid boundary effects and ensure that the content from different blocks can be continuously assembled together. Finally, the rest of content is aggregated into the final reconstructed frame. These steps are repeated until all overlapped blocks within the input decoded frame have been processed and assembled in the final reconstructed frame.

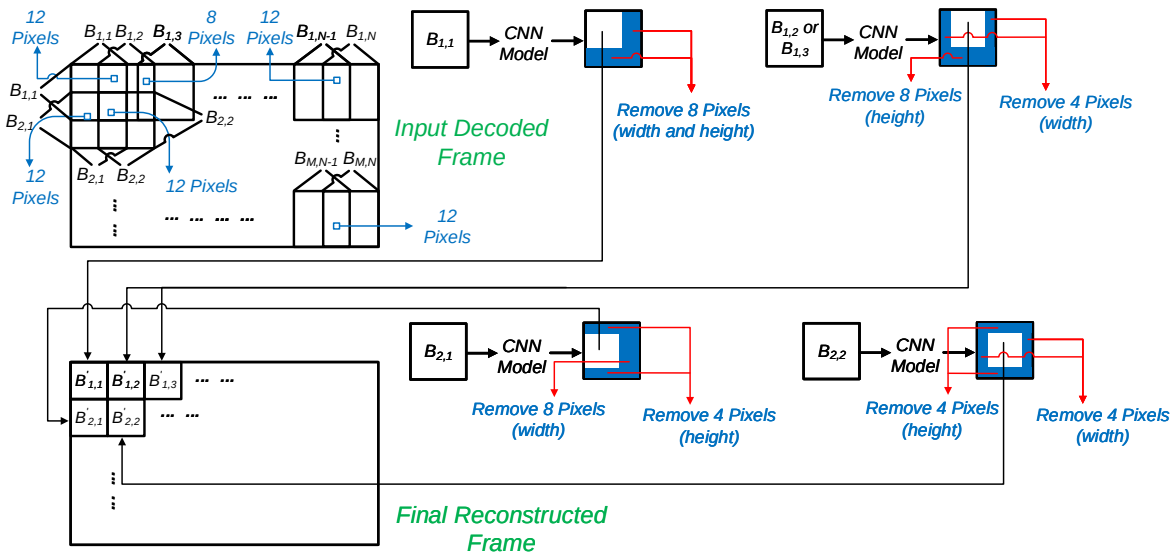


Figure. 3.2 Diagram of the frame reconstruction method used in the thesis.

It is noted that the following chapters will also employ the similar network training and evaluation procedures described in this section.

3.2.5 Experiment Settings

Four different coding modules have been integrated into the HEVC (HM 16.20) reference software, and have been fully tested under JVET-CTC [39] using the Random Access configuration (Main10 profile). Nineteen JVET-CTC SDR (as introduced in Section 2.3.1) video sequences from resolution classes A1, A2, B, C and D were employed as test content,

Table 3.2 Evaluation results for PP coding module for ten tested network architectures and six different training databases. Values indicate the average BD-rate (%) for all nineteen JVET-CTC tested sequences assessed by PSNR or VMAF.

CNN Model	DIV2K [119]		REDS [140]		CD [135]	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
SRCNN	0.4	-2.8	2.5	-3.8	11.0	-6.2
FSRCNN	0.7	-2.6	3.2	-1.2	24.4	2.6
VDSR	0.3	-2.9	2.3	-4.0	3.1	-2.2
DRRN	-5.0	-6.1	-4.7	-8.2	0.4	-1.1
EDSR	-5.4	-4.9	-3.1	-6.1	-0.8	-6.5
SRResNet	-5.3	-5.4	-4.0	-9.0	4.0	-3.8
ESRResNet	-6.9	-6.7	-6.1	-9.4	-3.5	-9.1
RCAN	-6.6	-7.3	-6.3	-9.7	-4.5	-11.0
RDN	-7.0	-7.2	-6.9	-10.6	-4.6	-10.9
MSRResNet	-6.4	-6.5	-5.3	-9.2	-2.6	-8.7

Continue

CNN Model	VideoSet [139]		HIF [142]		BVI-DVC	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
SRCNN	-0.4	-2.8	-0.8	-3.4	**-1.9**	**-7.4**
FSRCNN	-0.2	-2.5	-0.5	-3.2	**-1.6**	**-7.3**
VDSR	-0.3	-2.9	-0.7	-3.5	**-1.9**	**-7.6**
DRRN	-7.3	-10.4	-8.7	-12.0	**-10.8**	**-14.9**
EDSR	-6.6	-10.1	-7.8	-11.6	**-10.0**	**-14.6**
SRResNet	-6.5	-8.3	-7.4	-9.5	**-9.8**	**-12.7**
ESRResNet	-8.4	-13.2	-9.5	-14.7	**-11.8**	**-17.7**
RCAN	-8.6	-14.1	-9.7	-15.3	**-12.1**	**-18.5**
RDN	-8.8	-12.4	-9.8	-13.9	**-12.2**	**-17.0**
MSRResNet	-7.0	-9.8	-8.1	-11.2	**-10.4**	**-14.2**

none of which were included in any of the six training databases. It is noted that only class A1 and A2 (2160p) were used to evaluate SRA coding module, as it has been previously

Table 3.3 Evaluation results for ILF coding module for ten tested network architectures and six different databases. Each value indicates the average BD-rate (%) for all nineteen JVET-CTC tested sequences assessed by PSNR or VMAF.

CNN Model	DIV2K [119]		REDS [140]		CD [135]	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
SRCNN	-0.2	-2.6	-0.6	-2.5	-0.1	-1.4
FSRCNN	-0.1	-2.2	-0.2	-1.1	0.0	-0.5
VDSR	-1.0	-1.1	-0.7	-2.7	0.0	-0.5
DRRN	-4.0	-5.6	-3.1	-4.6	0.4	-1.1
EDSR	-4.5	-6.1	-2.7	-3.1	-1.3	-4.0
SRResNet	-5.1	-8.6	-2.9	-4.0	-1.1	-2.8
ESRResNet	-5.8	-8.6	-3.3	-5.1	-2.5	-6.8
RCAN	-5.4	-8.5	-6.3	-9.7	-3.0	-8.5
RDN	-5.8	-8.8	-3.7	-5.6	-3.0	-8.9
MSRResNet	-5.6	-9.4	-4.6	-6.7	-2.0	-6.5

Continue

CNN Model	VideoSet [139]		HIF [142]		BVI-DVC	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
SRCNN	-0.5	-5.3	-0.7	-5.7	**-1.4**	**-8.5**
FSRCNN	-0.6	-4.8	-0.5	-5.2	**-1.3**	**-8.1**
VDSR	-1.2	-3.9	-1.4	-4.1	**-2.2**	**-6.5**
DRRN	-5.9	-8.5	-6.2	-8.7	**-6.8**	**-11.0**
EDSR	-4.5	-7.3	-5.3	-7.6	**-5.9**	**-9.9**
SRResNet	-4.9	-8.2	-5.7	-8.4	**-6.4**	**-10.6**
ESRResNet	-6.4	-9.0	-6.6	-9.3	**-7.3**	**-12.0**
RCAN	-6.6	-9.1	-6.9	-9.4	**-7.4**	**-11.4**
RDN	-6.7	-9.2	-6.9	-9.6	**-7.5**	**-11.8**
MSRResNet	-5.8	-8.4	-5.9	-9.0	**-6.4**	**-11.3**

reported [57] that for lower resolutions SRA may provide limited and inconsistent coding gains.

Table 3.4 Evaluation results for SRA coding module for ten tested network architectures and six different databases. Each value indicates the average BD-rate (%) for all six UHD JVET-CTC tested sequences assessed by PSNR or VMAF.

CNN Model	DIV2K [119]		REDS [140]		CD [135]	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
SRCNN	3.9	-11.9	8.6	-19.6	6.6	-12.4
FSRCNN	1.1	-12.3	-0.3	-18.0	9.9	-9.8
VDSR	4.4	-11.5	4.3	-15.9	25.9	7.2
DRRN	-8.5	-17.6	-7.8	-26.1	-7.2	-22.1
EDSR	-6.4	-16.3	-6.9	-26.1	-3.2	-20.4
SRResNet	-6.7	-11.5	-7.0	-28.1	-5.5	-19.8
ESRResNet	-9.9	-19.4	-9.9	-31.7	-7.8	-23.5
RCAN	-10.2	-19.3	-10.9	-32.2	-8.4	-23.2
RDN	-10.0	-19.1	-9.7	-31.4	-8.4	-22.7
MSRResNet	-9.2	-18.9	-8.5	-29.9	-7.1	-22.8

Continue

CNN Model	VideoSet [139]		HIF [142]		BVI-DVC	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
SRCNN	-1.4	-19.2	-1.8	-19.9	**-3.1**	**-21.1**
FSRCNN	-1.7	-17.9	-2.2	-18.5	**-4.5**	**-20.9**
VDSR	-0.4	-16.2	-0.8	-16.1	**-6.6**	**-18.3**
DRRN	-11.2	-27.3	-12.1	-27.6	**-15.0**	**-33.2**
EDSR	-9.8	-26.8	-10.9	-27.1	**-13.4**	**-30.1**
SRResNet	-9.3	-26.5	-10.6	-26.7	**-13.2**	**-30.0**
ESRResNet	-12.3	-29.7	-13.3	-30.0	**-16.1**	**-33.6**
RCAN	-13.4	-31.2	-14.2	-31.7	**-17.1**	**-35.1**
RDN	-12.6	-30.9	-13.7	-31.1	**-16.6**	**-34.5**
MSRResNet	-10.7	-27.8	-11.9	-28.5	**-14.6**	**-32.7**

The rate quality performance (coding performance) is benchmarked against the original HEVC HM 16.20, using Bjøntegaard Delta measurement (BD-rate) [40] based on two quality

Table 3.5 Evaluation results for EBDA coding module for ten tested network architectures and six different databases. Each value indicates the average BD-rate (%) for all nineteen JVET-CTC tested sequences assessed by PSNR or VMAF.

CNN Model	DIV2K [119]		REDS [140]		CD [135]	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
SRCNN	-0.1	-7.6	2.6	-7.0	11.9	-10.0
FSRCNN	0.1	-6.7	3.7	-5.2	28.0	-0.4
VDSR	0.93	-6.8	19.6	-4.5	23.0	-0.13
DRRN	-6.0	-10.8	-3.7	-9.8	-1.1	-11.3
EDSR	-6.1	-11.5	-3.6	-11.1	-0.2	-9.6
SRResNet	-5.9	-10.4	0.5	-9.2	2.1	-7.0
ESRResNet	-7.1	-11.3	-4.1	-11.0	-2.0	-13.8
RCAN	-7.6	-11.0	-5.2	-11.7	-1.4	-12.3
RDN	-7.7	-11.7	-5.6	-10.6	-1.5	-13.7
MSRResNet	-7.0	-11.2	-4.1	-11.7	-2.5	-13.9

Continue

CNN Model	VideoSet [139]		HIF [142]		BVI-DVC	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
SRCNN	-1.7	-10.2	-1.8	-10.3	**-2.1**	**-11.0**
FSRCNN	-1.6	-7.7	-2.0	-8.6	**-2.9**	**-11.5**
VDSR	-1.1	-4.8	-1.4	-5.9	**-5.6**	**-9.1**
DRRN	-6.9	-13.9	-8.3	-15.4	**-11.8**	**-18.4**
EDSR	-7.1	-13.1	-7.8	-14.5	**-10.3**	**-17.7**
SRResNet	-6.7	-11.3	-7.6	-12.6	**-10.5**	**-15.9**
ESRResNet	-8.0	-14.6	-8.9	-15.6	**-12.0**	**-19.0**
RCAN	-8.3	-15.2	-9.8	-16.3	**-12.5**	**-19.8**
RDN	-7.8	-14.9	-9.1	-15.7	**-12.1**	**-19.1**
MSRResNet	-7.1	-13.6	-8.4	-14.8	**-11.1**	**-17.9**

metrics, PSNR (luminance channel only) and VMAF (version 0.6.1) [189] which have been discussed in Section 2.4.1. Here, BD-rate indicates the overall percentage bitrate reduction

(when negative) or increase (when positive) for the same video quality (PSNR or VMAF) [40]. The training and evaluation processes were both executed on a shared cluster, BlueCrystal Phase 4 (BC4) based in the University of Bristol [211], in which each node contains two 14 core 2.4 GHz Intel E5-2680 V4 (Broadwell) CPUs, 128 GB of RAM, and NVIDIA P100 GPU devices.

3.3 Results and Discussion

3.3.1 Comparison of Databases

The average BD-rate values are reported in Tables 3.2-3.5 for each evaluated training database, network architecture and coding module. It can be observed from Tables 3.2-3.5 and Figure. 3.5 that, for all tested network architectures and coding modules, the coding gains (in terms of average BD-rates for all tested sequences) achieved after training on the proposed BVI-DVC database are significantly greater than for the other five benchmark databases (DIV2K, REDS, CD, VideoSet, and HIF) for both PSNR and VMAF quality metrics. This is reinforced by considering the mean (among ten networks and four coding modules) of all the average BD-rates for each database; Figure. 3.6 shows in excess of 2.0% and 3.0% additional bitrate savings obtained by using BVI-DVC compared to the other five databases based on the PSNR and VMAF quality metrics respectively. CD offers the worse overall performance, especially for results based on the assessment of PSNR.

The effectiveness of the proposed database can be further demonstrated by comparing the perceptual quality of reconstructed frames when different databases are employed to train the same CNN model. Figures. 3.3 and 3.4 show examples of reconstructed frames generated by RDN [66] when it has been trained using DIV2K, REDS, CD, VideoSet, HIF and BVI-DVC databases for both PP and SRA coding tools. The HEVC anchors are also used for benchmarking. RDN [66] is selected as it has achieved evident coding gains (based on both PSNR and VMAF) with relatively lower computational complexity compared to other complex networks, such as ESRResNet [29] and RCAN [68] (the relative complexities of these networks are shown in Tables 4.1-4.2). From Figures. 3.3 and 3.4, it can be observed that when the CNN model is trained on the proposed BVI-DVC database, the reconstructed content exhibits improved visual quality, with fewer blocking artefacts, better high frequency detail and higher contrast compared to those trained using other databases. It is noted that each database was independently utilised to train different networks for coding performance evaluation in this thesis. Combining BVI-DVC with other databases which contain unique content types that BVI-DVC does not have (e.g. dark content, high dynamic range sequences)

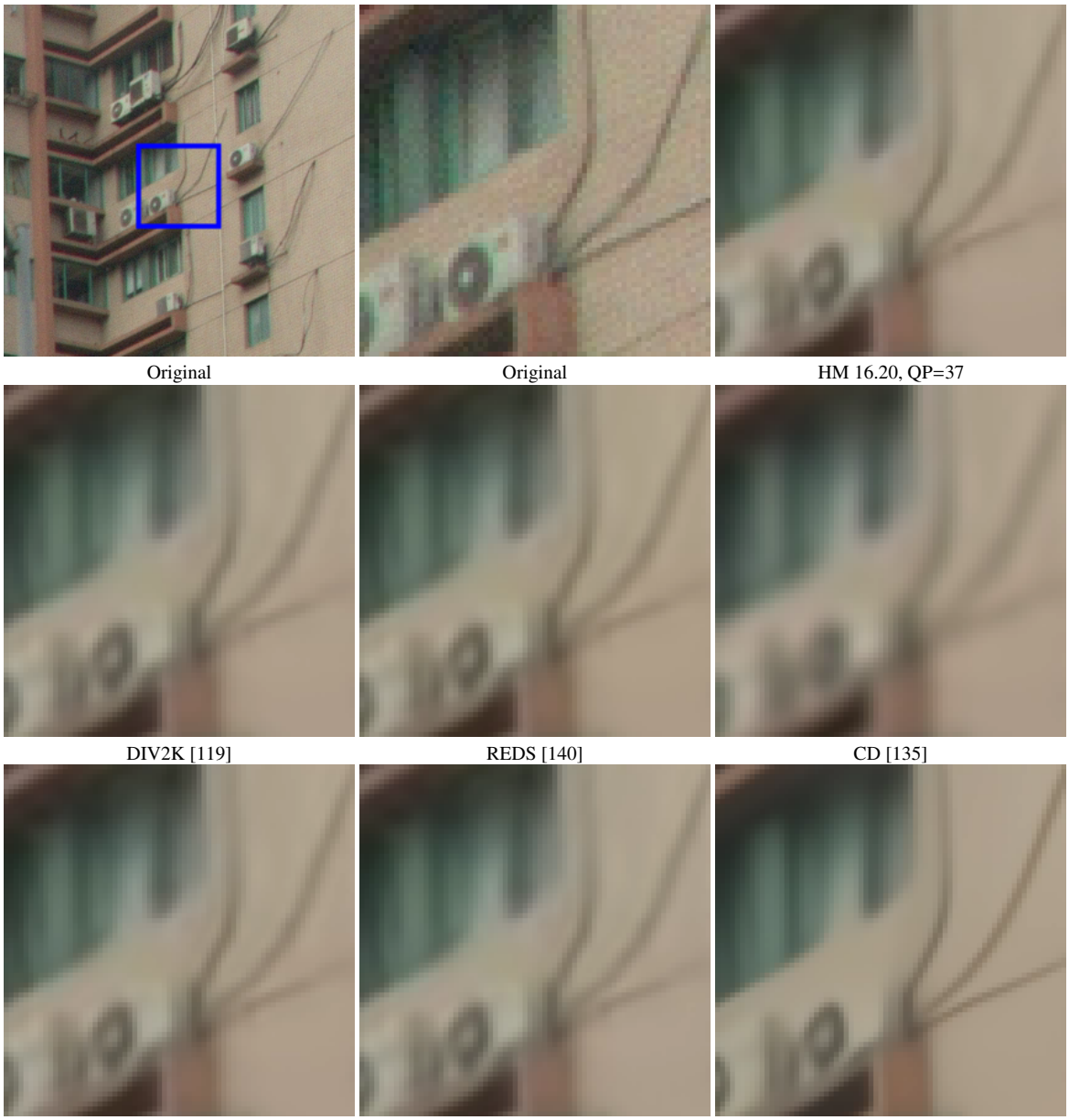


Figure. 3.3 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), RDN models trained using six databases for PP coding tool. The bit consumption in each example set is identical/similar for all tested versions. Rows 1, 2 and 3 correspond to the 250th frame of the *DaylightRoad2* sequence.

may lead to the improvement of the training performance and generalisation ability of the networks [114]. This will be further explored in the future work.

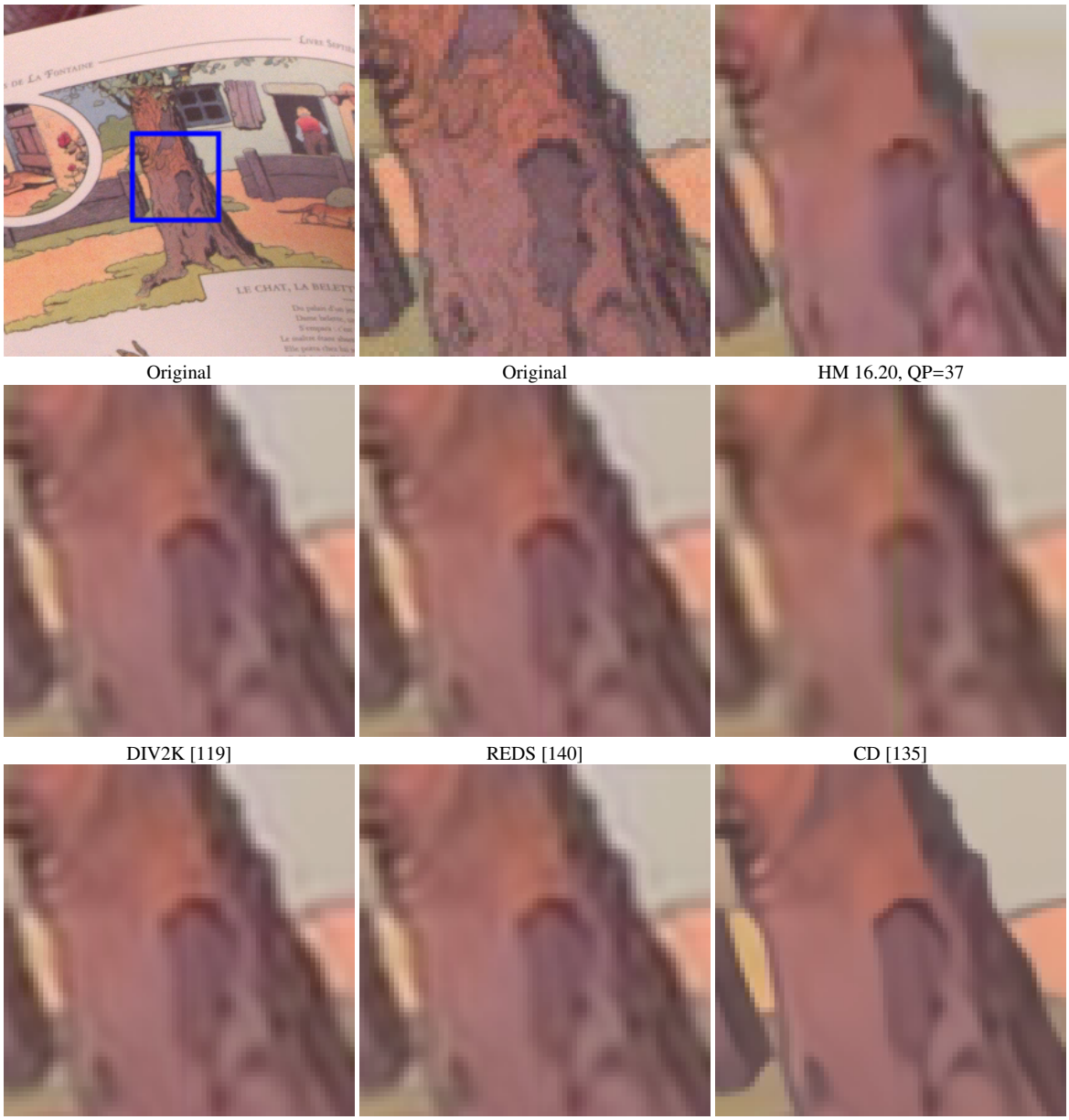


Figure. 3.4 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), RDN models trained using six databases for SRA coding tool. The bit consumption in each example set is identical/similar for all tested versions. Rows 1, 2 and 3 correspond to the 104th frame of the *CatRobot1* sequence.

3.3.2 Comparison of Networks

Ten evaluated network architectures have further been compared under fair configurations (identical training and evaluation databases). The results in Tables 3.2-3.5 are summarised, by taking the mean (among six training databases and four coding modules) of average

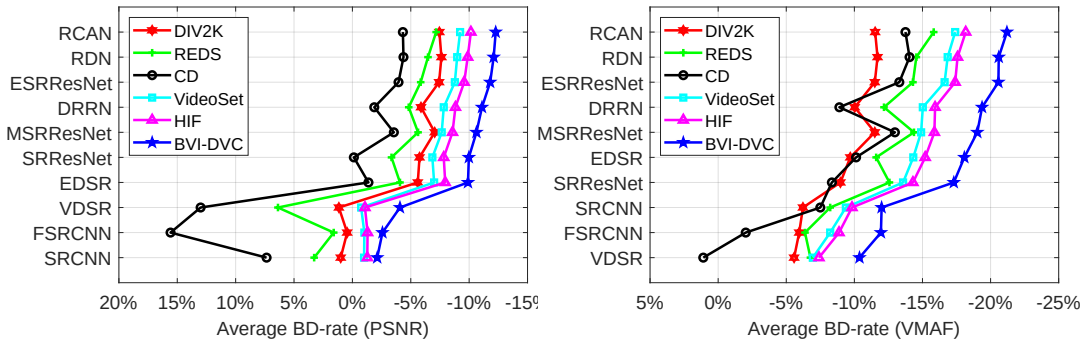


Figure. 3.5 Average coding gains for four coding modules obtained using 10 commonly employed network architectures trained on six different databases: BVI-DVC, DIV2K [119], REDS [140], CD [135], VideoSet [139], and HIF [142]. All methods are integrated into HEVC Test Model (HM 16.20).

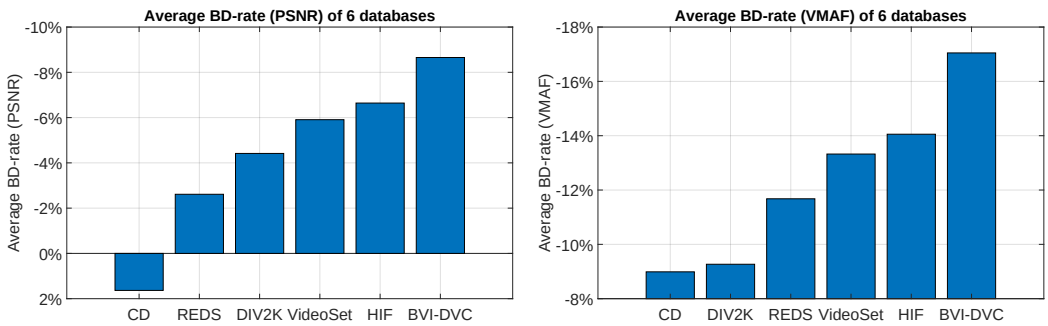


Figure. 3.6 Average BD-rate (based on PSNR and VMAF) of six tested training databases for all the evaluated coding modules and CNN architectures.

BD-rate values for each network architecture, in Figure. 3.7. It can be observed that RCAN, RDN, ESRResNet and MSRResNet offer better coding performance (for both PSNR and VMAF) than the other six evaluated network architectures. This is likely to be because of the residual block structure employed. The coding gains for VDSR, FSRCNN and SRCNN are relatively low compared to other networks, exhibiting coding loss when PSNR is used to assess video quality especially when they are trained on the CD database (refer to Figure. 3.5). This may be due to their simple network architecture (FSRCNN and SRCNN) and a large number of convolutional layers without residual learning structure (VDSR), which lead to less stable training and evaluation [66, 68].

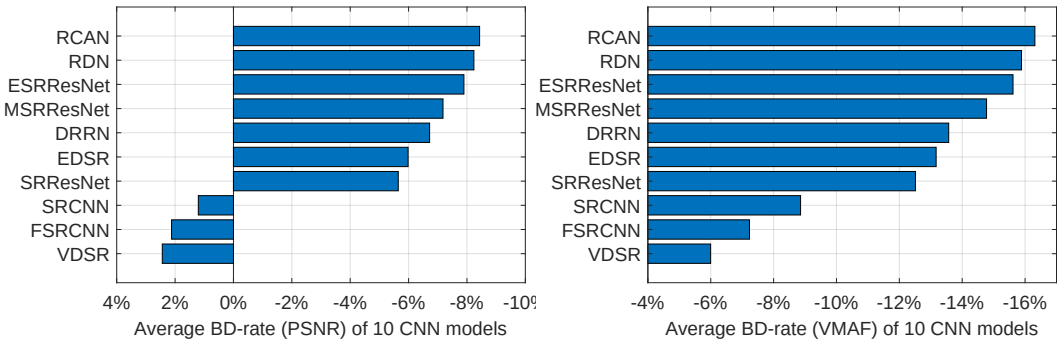


Figure. 3.7 Average BD-rate (based on PSNR and VMAF) of 10 test network architectures for all coding modules and training databases.

3.4 Summary

In this chapter, a new extensive and representative video database (BVI-DVC) is presented which is specifically designed for training CNN-based video compression algorithms. With carefully selected sequences including diverse content (including various static and dynamic video textures), this database offers significantly improved training effectiveness compared to other commonly used image and video training databases (overall additional coding gains are up to 10.3% based on PSNR and 8.1% based on VMAF). Additionally, the coding performance of different network architectures has also been compared. The results show that the networks with complex and advanced architectures generally achieved much better compression performance than those only with simple structures (e.g. simply concatenated convolutional layers without residual learning-SRCNN, FSRCNN and VDSR). This further demonstrates that the networks with carefully designed architectures are also important for deep video compression. The BVI-DVC database is available online[2] for public testing and it has recently been used by JVET Ad-hoc Group 11 [37, 169] for optimising neural network-based video coding tools. Its content diversity makes it a reliable training database, not just for CNN-based compression approaches, but also for other computer vision and image processing related tasks, such as image/video de-noising, video frame interpolation and super-resolution.

[2] The BVI-DVC database can be downloaded from: https://fan-aaron-zhang.github.io/BVI-DVC/.

Chapter 4

MFRNet: A New CNN Architecture for Deep Video Coding Enhancement

As discussed in Chapter 2, one of the important aspects in the development of deep video compression algorithms is the network architecture. Current deep video coding tools commonly employ existing or modified CNN structures which were originally designed for image and video restoration. Many of them do not contain advanced features, such as dense connections, feature review structures, and thus cannot produce optimal overall coding performance.

In this chapter, a new CNN architecture (MFRNet) is presented for video compression enhancement based on the four coding modules - post-processing (PP), in-loop filtering (ILF), spatial resolution adaptation (SRA) and effective bit depth adaptation (EBDA). MFRNet exploits novel network structures including cascading connections and multi-level feature review residual dense blocks, which offer evident performance enhancement over state-of-the-art network architectures.

The work presented in this chapter has been published in [4].

4.1 Network Architecture

This section presents the new CNN architecture, MFRNet. The backbone structure of the MFRNet is first introduced and then the two main components of the MFRNet, the Multi-level Feature Review Residual Dense Block (MFRB) and Feature Review Residual Dense Block (FRB) are respectively described in detail.

4.1.1 Network Backbone Structure

The proposed CNN architecture proposed is illustrated in Figure. 4.1. This network accepts a 96×96 YCbCr 4:4:4 image block as input, and outputs a filtered image block in the same format. It first employs a convolutional layer alongside a Leaky ReLU (LReLU) activation function to extract shallow features (SFs) from the input image block. This SF extraction layer is followed by four Multi-level Feature review Residual dense Blocks (MFRBs, $B1$-$B4$), which are designed for deep dense feature extraction. Ten cascading connections, shown as black curves in Figure. 4.1, are utilised to feed the initial SFs and the output from the first three MFRBs (G_1, G_2 and G_3) into following MFRBs or into the first reconstruction layer (shown as RL1 in Figure. 4.1) through a 1×1 convolutional layer (with l LReLU). This structure is designed to effectively improve information flow while reducing the number of residual dense blocks in the network [69, 85]. Moreover, each of the first three MFRBs also feeds its high dimensional feature outputs, F_1, F_2, F_3, into the next MFRB, as shown in Figure. 4.2, in order to reuse previous HDFs (high dimensional features) [66]. After four MFRBs and the first reconstruction layer (RL1), a skip connection is employed to connect the output of this reconstruction layer and the output of the shallow feature extraction layer. Finally, an additional reconstruction layer (RL2) and an output layer are employed to output a residual signal, which is then combined with the input through a long skip connection to obtain the final image block. The kernel sizes, feature map numbers and stride values for each convolution layer can be found in Figure. 4.1.

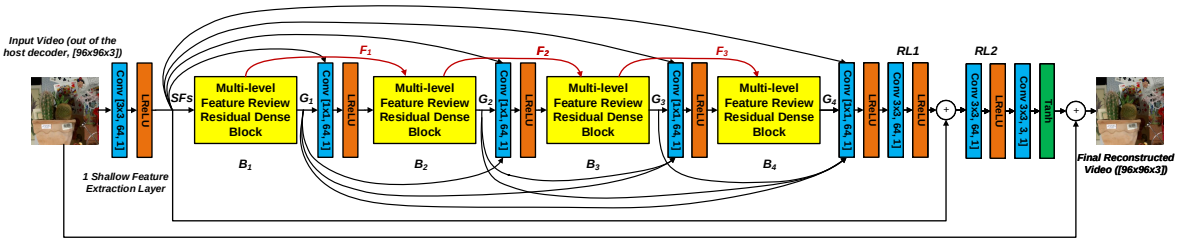


Figure. 4.1 Illustration of the proposed MFRNet architecture.

4.1.2 Multi-level Feature Review Residual Dense Block (MFRB)

Figure. 4.2 shows the structure of each MFRB (B_i, i=1, 2, 3 and 4), which contains three Feature review Residual dense Blocks (FRBs), b_i^1, b_i^2 and b_i^3. In many existing CNN architectures, which employ residual (or residual dense) blocks [64, 28, 29, 66, 1], there is only a single information flow, which prevents high-level blocks from fully accessing previously generated features. This leads to a problem of diminishing feature reuse, which in

turn affects the overall performance of the network [58]. To address this issue, in the proposed architecture, each FRB (b_i^j), except the first one in B_1 and the last one in B_4, is designed to have two inputs and two outputs [58], as shown in Figures. 4.2 and 4.3. Each FRB not only receives the main branch output from the previous MFRB (G_{i-1}) or FRB (g_i^{j-1}), but also accepts the side branch output from the previous MFRB (F_{i-1}) or FRB (f_i^{j-1}), which contains high dimensional features. Respectively, in addition to its main branch output (γ_i^j), each FRB also feeds its side branch output (f_i^j) into the subsequent FRB block in this or the next MFRB (if applicable). This new structure allows each FRB to review the high dimensional features generated in its previous block, which effectively enhances the information flow between blocks. Finally, a multi-level residual learning structure is designed to apply skip connection between the input of the first FRB and the output of each FRB. This enables bypassing of redundant information and stabilises training and evaluation processes [55, 212].

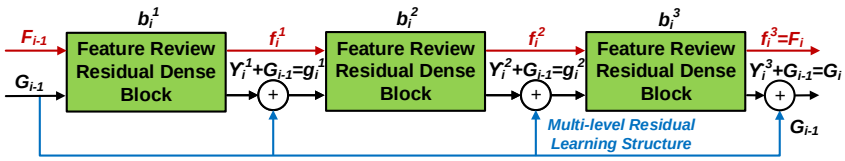


Figure. 4.2 Illustration of an MFRB (B_i).

4.1.3 Feature Review Residual Dense Block (FRB)

Figure. 4.3 shows the FRB structure (b_i^j), which contains a main branch and a side branch. The former first accepts the output from the previous MFRB (G_{i-1}) or FRB (g_i^{j-1}) if it is available, and extracts dense features through four convolutional layers with dense connections [67, 66]. Each of these layers contains one convolutional layer and a LReLU function. The output of these four dense convolutional layers is then concatenated together with the side branch output from the previous MFRB (F_{i-1}) or FRB (f_i^{j-1}) and fed into the last convolutional layer. The output of this layer is combined with the input (G_{i-1} or g_i^{j-1}) of this FRB through a skip connection to obtain the final FRB output (g_i^j). The concatenated HDFs are further fed into two modified residual blocks and one convolutional layer with a 1×1 kernel size to obtain the output of this side branch (f_i^j) in this FRB. This is also sent to the subsequent FRB block (if applicable) to realise HDF reviewing.

4.2 Experimental Configurations

Here we follow the same procedures for training data generation, network training and evaluation, and the identical experiment settings described in Sections 3.2.3-3.2.5. Four

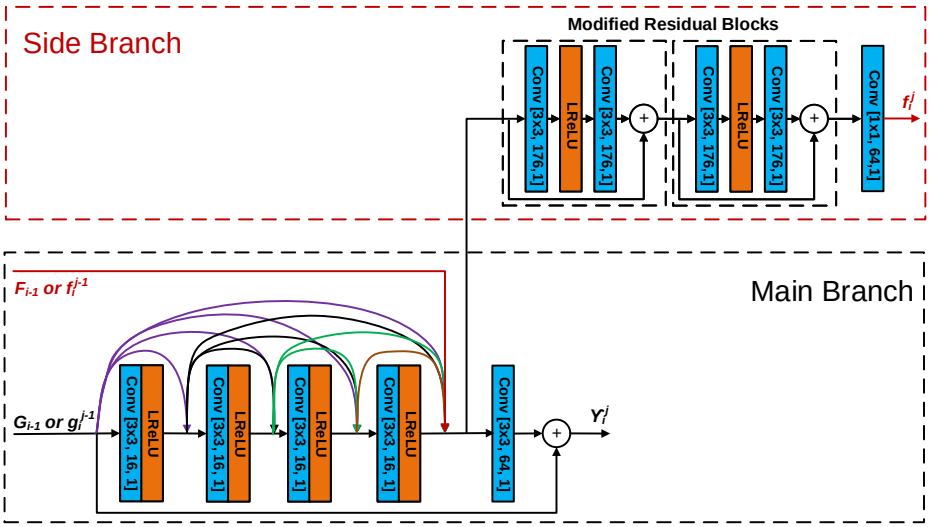


Figure. 4.3 Illustration of an FRB (b_i^j).

typical CNN-based video coding enhancement modules have been employed to evaluate the effectiveness of the MFRNet. These include post-processing (PP), in-loop filtering (ILF), spatial resolution adaptation (SRA) and effective bit depth adaptation (EBDA). All of these coding tools have been discussed in detail in Section 2.3.3. In this experiment, both HEVC HM 16.20 and VVC VTM 7.0 are utilised as the host codecs (and anchor codecs).

4.3 Results and Discussion

This section presents the experimental results, including the comprehensive comparisons between the MFRNet and other popular state-of-the-art network architectures for PP, ILF, SRA and EBDA tools based on the HEVC HM 16.20; compression results of the MFRNet for PP, ILF, SRA and EBDA coding modules based on the HEVC HM 16.20 and VVC VTM 7.0; and the comparisons between other notable CNN-based PP and ILF approaches based on the HEVC HM and VVC VTM.

4.3.1 Comparisons with Popular CNN Architectures

To further demonstrate the effectiveness of the MFRNet CNN structure, it has also been compared with eleven popular CNN architectures in the context of PP, ILF, SRA and EBDA for HEVC HM 16.20. These include SRCNN [61], FSRCNN [62], VDSR [63], DRRN [64], EDSR [65], SRResNet [28], CARN [69], ESRResNet [29], RCAN [68], RDN [66], and MSRResNet [1]. All of these models have been widely used in image super-resolution

Table 4.1 Comparison between eleven popular CNN architectures and the proposed MFRNet in the context of ILF and PP for HM 16.20.

CNN Model	CNN-based ILF (*Short Test*)			CNN-based PP		
	BD-rate (PSNR)	BD-rate (VMAF)	Relative Complexity (Encoding)	BD-rate (PSNR)	BD-rate (VMAF)	Relative Complexity (Decoding)
SRCNN [61]	-1.4%	-8.5%	1.2×	-1.9%	-7.4%	26.5×
FSRCNN [62]	-1.3%	-8.1%	1.5×	-1.6%	-7.3%	36.2×
VDSR [63]	-2.2%	-6.5%	1.8×	-1.9%	-7.6%	54.3×
DRRN [64]	-6.8%	-11.0%	2.4×	-10.8%	-14.9%	71.6×
EDSR [65]	-5.9%	-9.9%	9.4×	-10.0%	-14.6%	119.3×
SRResNet [28]	-6.4%	-10.6%	2.0×	-9.8%	-12.7%	64.9×
MSRResNet [1]	-6.4%	-11.3%	2.1×	-10.4%	-14.2%	65.1×
CARN [69]	-6.9%	-11.1%	1.9×	-11.2%	-15.4%	59.2×
ESRResNet [29]	-7.3%	-12.0%	7.8×	-11.8%	-17.7%	101.1×
RCAN [68]	-7.4%	-11.4%	12.4×	-12.1%	-18.5%	127.6×
RDN [66]	-7.5%	-11.8%	5.7×	-12.2%	-17.0%	91.8×
MFRNet	**-9.9%**	**-15.6%**	5.3×	**-14.1%**	**-21.0%**	81.2×

Table 4.2 Comparison between eleven popular CNN architectures and the proposed MFRNet in the context of SRA and EBDA for HM 16.20.

CNN Model	CNN-based SRA			CNN-based EBDA		
	BD-rate (PSNR)	BD-rate (VMAF)	Relative Complexity (Decoding)	BD-rate (PSNR)	BD-rate (VMAF)	Relative Complexity (Decoding)
SRCNN [61]	-3.1%	-21.1%	11.2×	-2.1%	-11.0%	26.6×
FSRCNN [62]	-4.5%	-20.9%	15.5×	-2.9%	-11.5%	36.2×
VDSR [63]	-6.6%	-18.3%	23.3×	-5.6%	-9.1%	54.1×
DRRN [64]	-15.0%	-33.2%	30.6×	-11.8%	-18.4%	71.8×
EDSR [65]	-13.4%	-30.1%	50.8×	-10.3%	-17.7%	119.7×
SRResNet [28]	-13.2%	-30.0%	27.6×	-10.5%	-15.9%	64.7×
MSRResNet [1]	-14.6%	-32.7%	27.7×	-11.1%	-17.9%	65.0×
CARN [69]	-14.7%	-32.6%	25.5×	-10.9%	-18.0%	59.4×
ESRResNet [29]	-16.1%	-33.6%	43.5×	-12.0%	-19.0%	101.5×
RCAN [68]	-17.1%	-35.1%	54.4×	-12.5%	-19.8%	127.4×
RDN [66]	-16.6%	-34.5%	39.3×	-12.1%	-19.1%	91.9×
MFRNet	**-17.5%**	-31.2%	34.4×	**-13.2%**	**-20.0%**	81.1×

and restoration, and some (VDSR and MSRResNet) have also been utilised in CNN-based

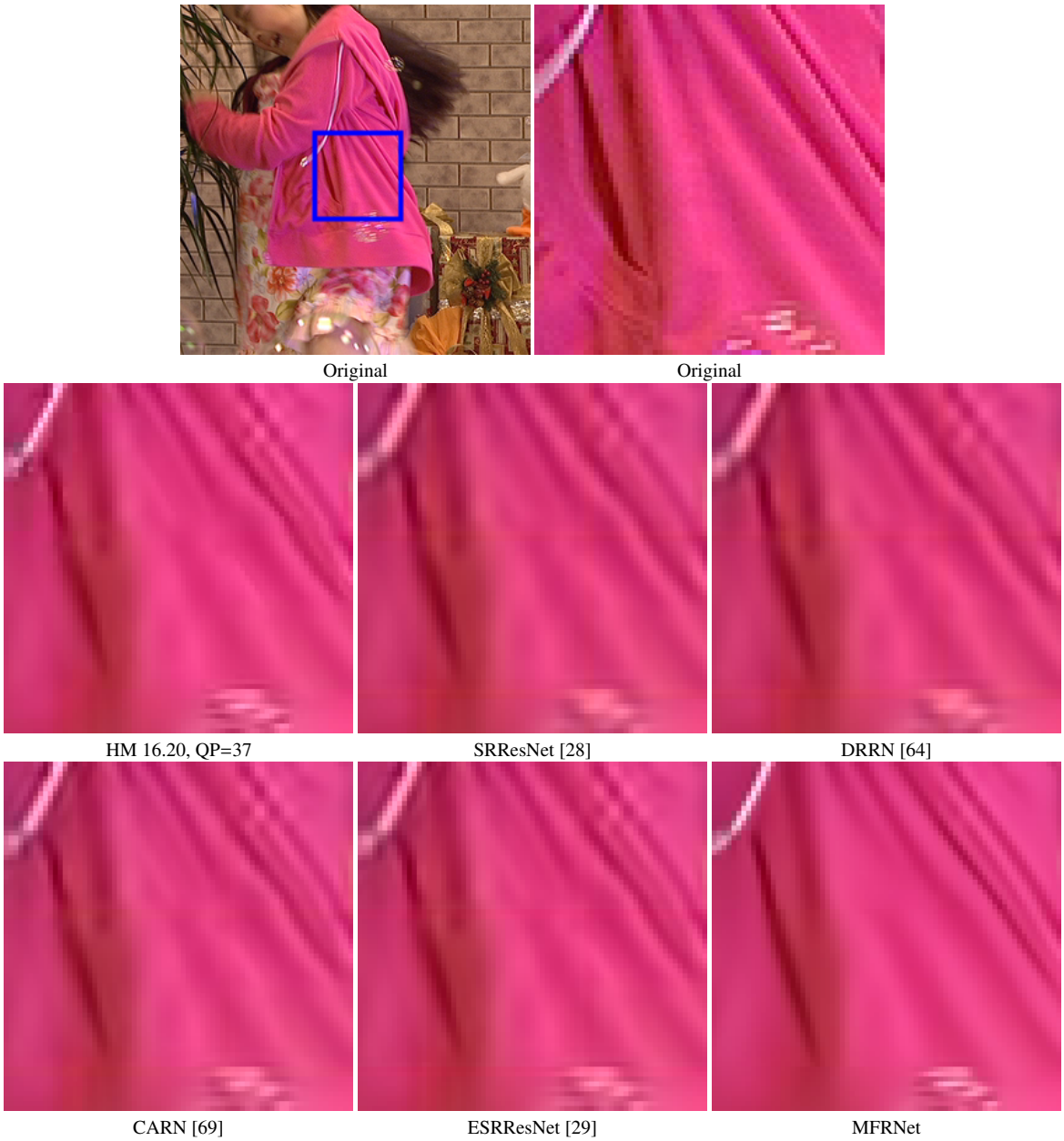


Figure. 4.4 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 170th frame of the *PartyScene* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast.

video compression tools [173, 1, 213, 57]. Most of these approaches provided superior performance to the state of the art in their application domain when they were first proposed.

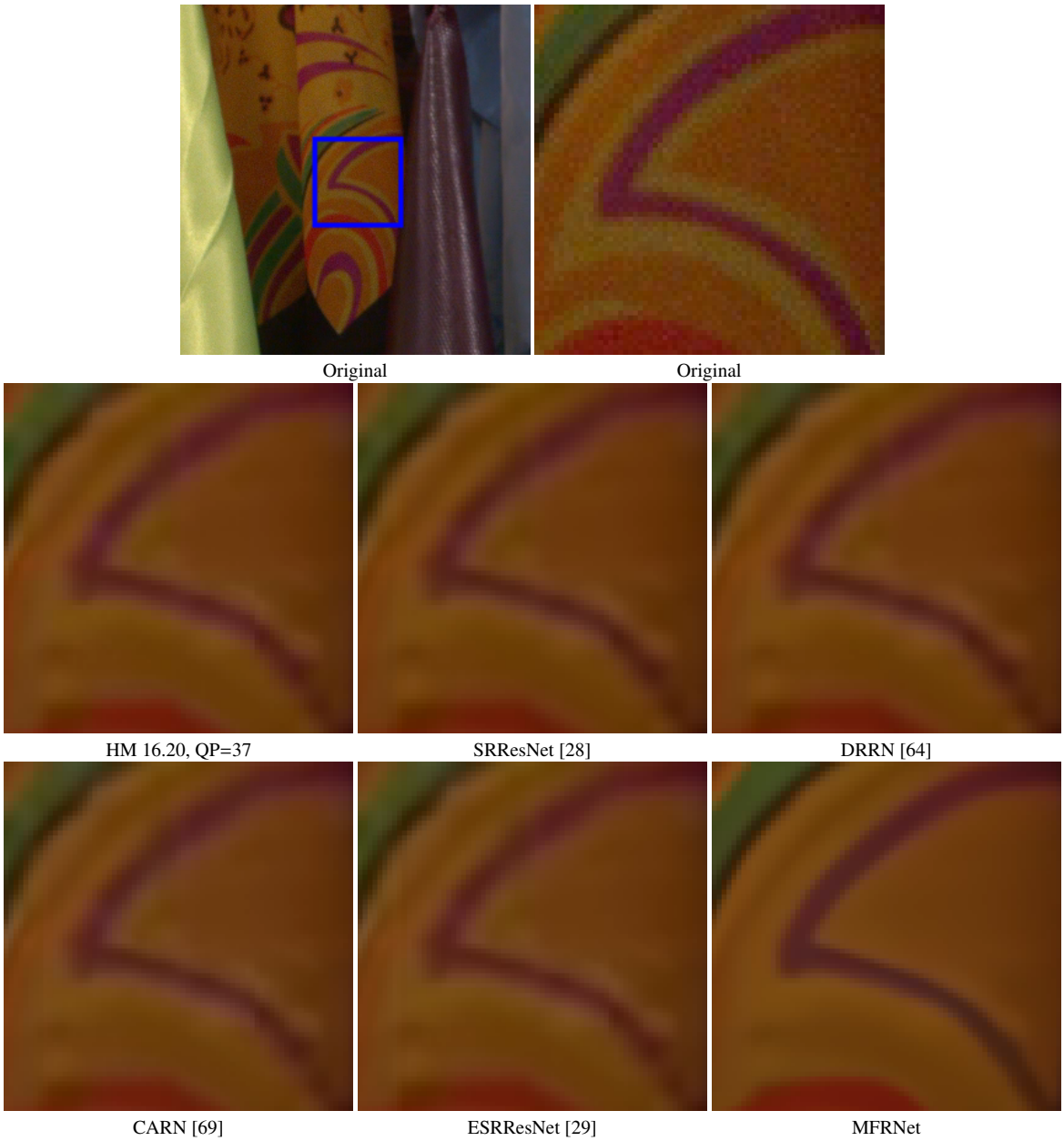


Figure. 4.5 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 216th frame of the *CatRobot1* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast.

All eleven models have been re-implemented using the same framework (TensorFlow 1.8.0) and were integrated into PP, ILF, SRA and EBDA coding modules for HEVC HM

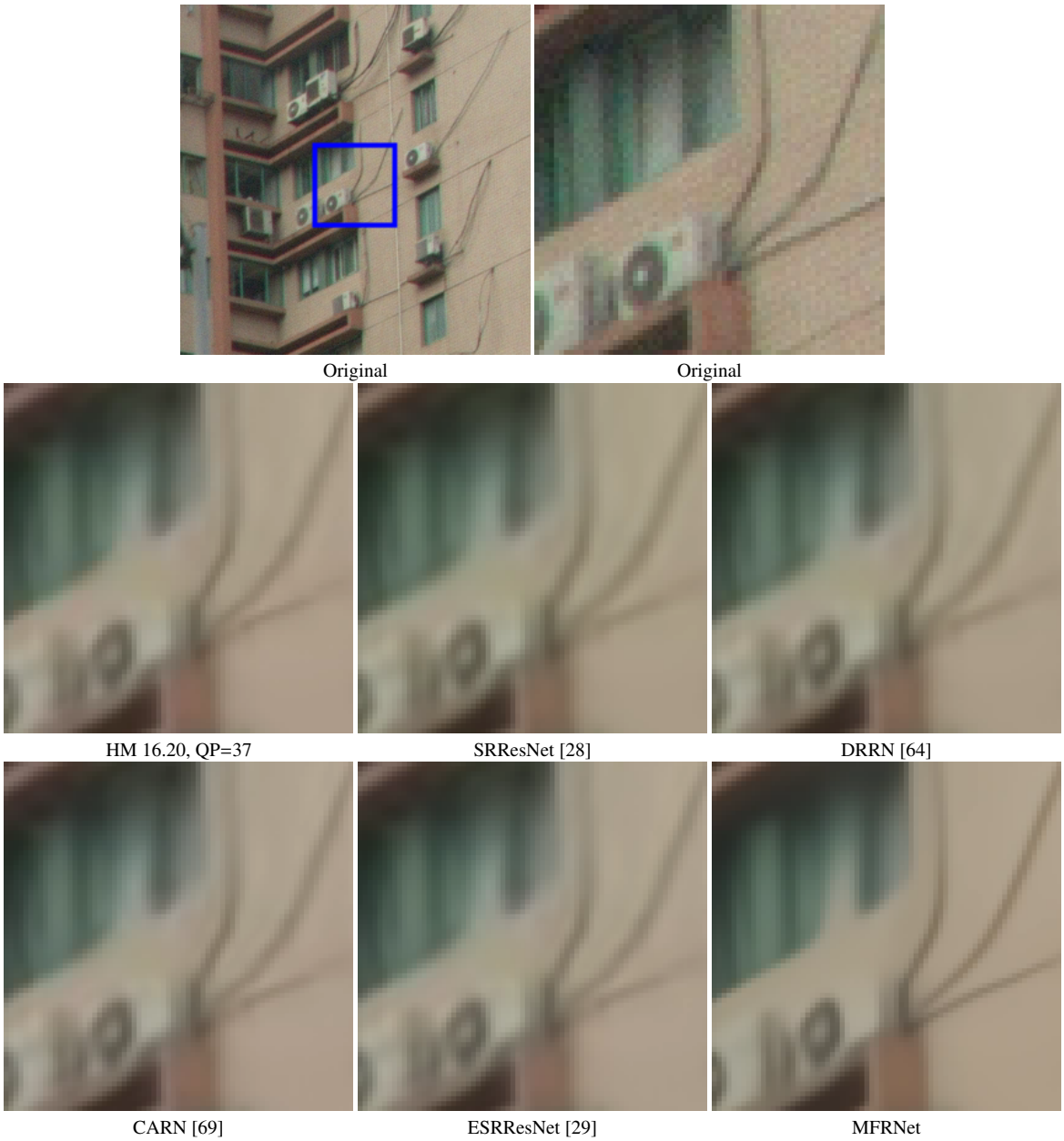


Figure. 4.6 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 250th frame of the *DaylightRoad2* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast.

16.20. During re-implementation, the input and output interfaces of these networks have been modified to satisfy the data format requirements. All networks were also trained on the

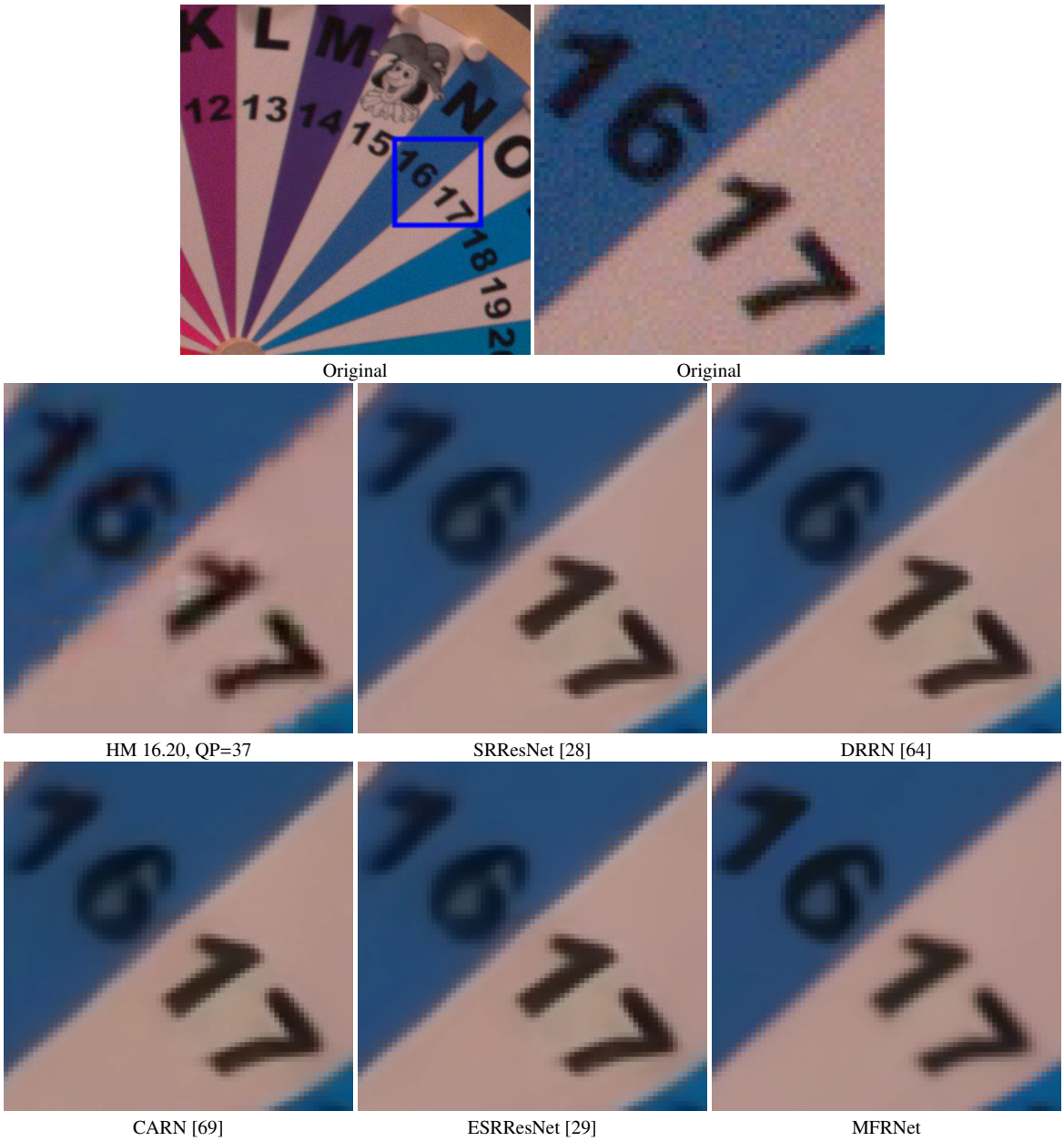


Figure. 4.7 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based SRA. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 98th frame of the *CatRobot1* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast.

BVI-DVC database following the same methodology as for the proposed network, using loss functions as described in their original literature.

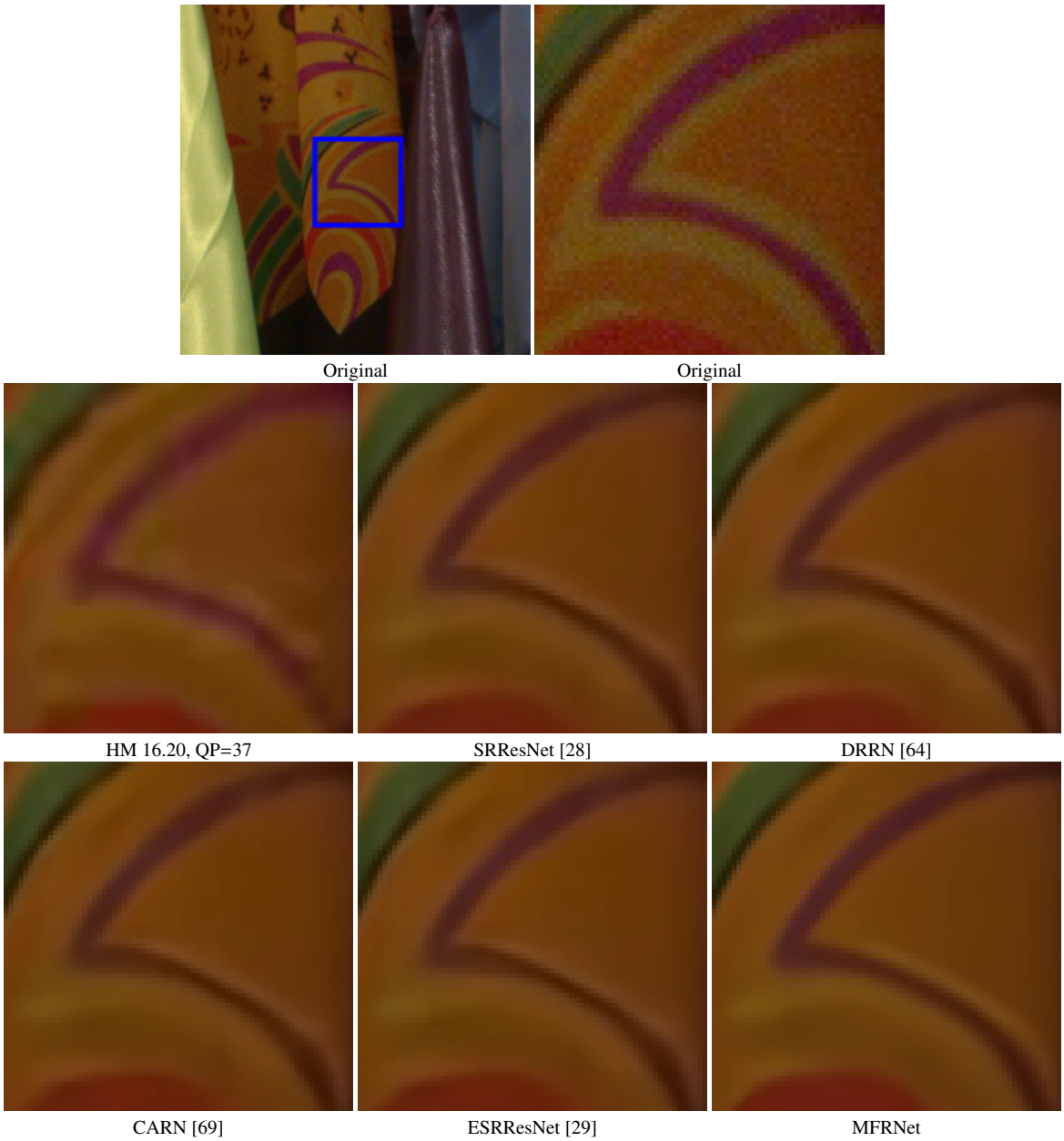


Figure. 4.8 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based SRA. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 216th frame of the *CatRobot1* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast.

Performance Comparisons for PP and ILF Coding Modules

Evaluation results on all 19 JVET test sequences are summarised in Table 4.1 and compared to those for MFRNet. The original HEVC HM 16.20 is employed as a benchmark. It should

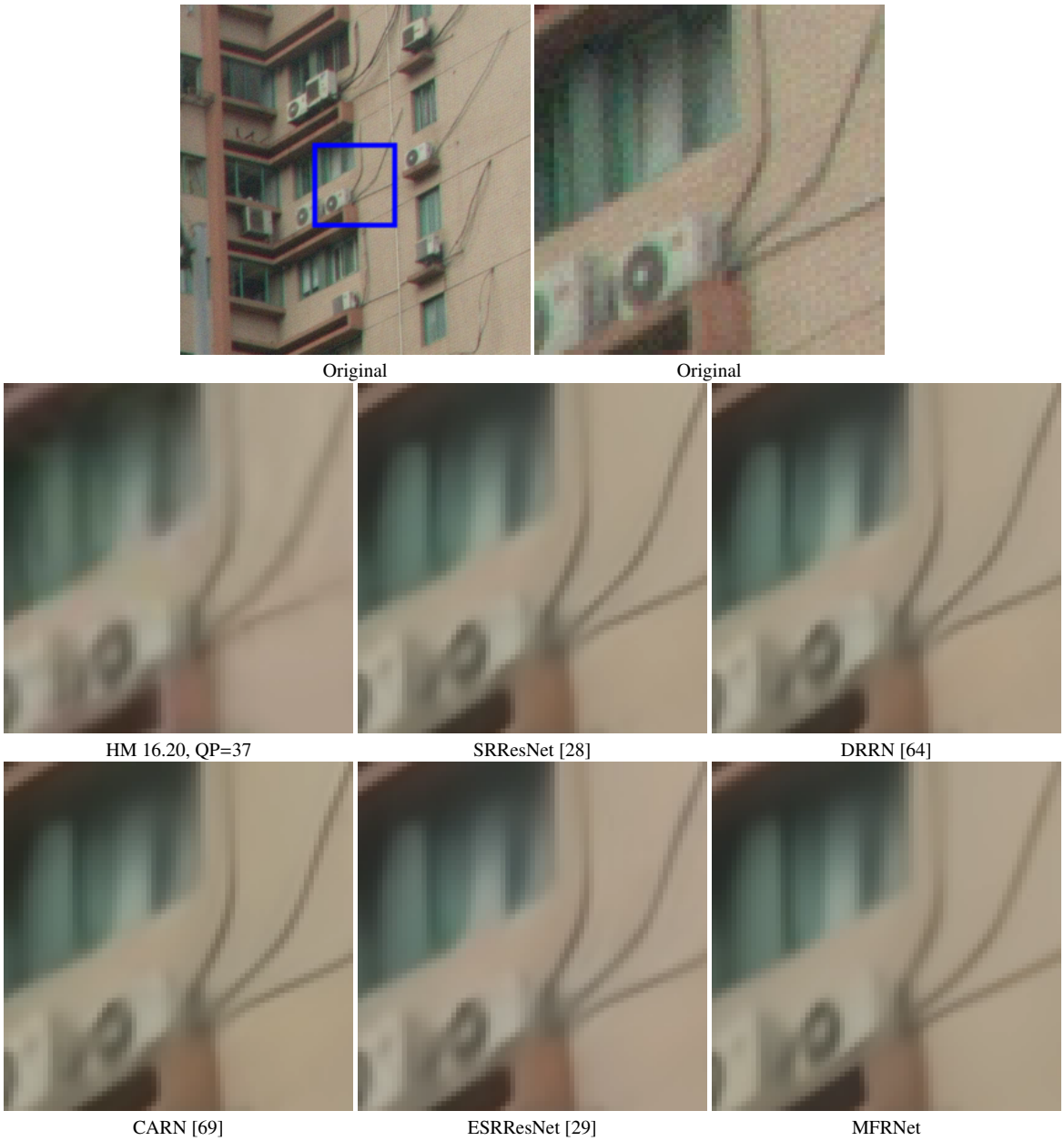


Figure. 4.9 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), four state-of-the-art network architectures and the MFRNet for CNN-based SRA. The bit consumption in each example set is identical for all tested versions. Rows 1, 2 and 3 correspond to the 250th frame of the *DaylightRoad2* sequence. It can be observed that the output of MFRNet exhibits improved perceptual quality compared to the anchor HEVC HM 16.20 and other compared networks, with fewer blocking artefacts, more textural detail and higher contrast.

be noted that a *Short Test* was conducted for evaluating different ILF coding modules as described in JVET proposal M0904 [214], in which only the first intra period of each test

sequence was encoded, while a *Full Test* (processing all frames in the sequence) was applied for PP. The relative computational complexity for each approach has also been calculated and benchmarked against the original HEVC HM 16.20 encoder (for ILF) and decoder (for PP).

It can be observed that MFRNet offers the best performance for both PP and ILF when compared to the other eleven architectures, with average coding gains of 14.1% for PP and 9.9% for ILF based on PSNR, and 21.0% for PP and 15.6% for ILF according to VMAF. These figures are consistently greater than those for other networks. In contrast, the computational complexity of the proposed architecture is lower than that of RCAN, EDSR, ESRResNet and RDN.

Performance Comparisons for SRA and EBDA Coding Modules

Table 4.2 summarises the coding performance comparisons between the MFRNet and eleven state-of-the-art CNN architectures in the context of SRA and EBDA for HEVC HM 16.20. It can be observed that the MFRNet provided the best coding performance compared to other networks based on the PSNR for both SRA and EBDA modules, with average coding gains of 17.5% for SRA and 13.2% for EBDA. When the VMAF is utilised as the quality metric, the average coding gains of MFRNet is slightly lower than some complex networks for SRA, such as the ESRResNet [29] and RCAN [68], while for EBDA, MFRNet can still provide the best performance over all the eleven network architectures.

Perceptual Comparisons

In order to further validate the effectiveness of the MFRNet, the subjective comparison results between the MFRNet, the anchor HEVC HM 16.20 and four popular networks (including SRResNet [28], DRRN [64], CARN [69] and ESRResNet [29]) are presented in Figures. 4.4-4.6 and Figures. 4.7-4.9 for PP and SRA coding modules respectively. It can be observed that for both PP and SRA modules, the output of the MFRNet exhibits clearly improved perceptual quality with fewer compression artefacts, more texture details and higher contrast compared to other networks. These effectively show the effectiveness of the MFRNet.

4.3.2 Compression Performance

Compression Results for PP and ILF Coding Modules

Tables 4.3 and 4.4 summarise the compression performance of the PP and ILF coding modules (with MFRNet) when integrated into HEVC HM 16.20 and VVC VTM 7.0. It can be observed that our proposed approach achieves significant and consistent coding gains

Table 4.3 Compression results of the MFRNet-based ILF and PP for HM 16.20.

Class-Sequence	CNN-based ILF (*Full Test*)		CNN-based PP	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
A1-Campfire	-6.2%	-16.5%	-12.4%	-19.2%
A1-FoodMarket4	-8.6%	-17.1%	-11.1%	-20.6%
A1-Tango2	-11.9%	-21.6%	-15.1%	-25.2%
A2-CatRobot1	-12.6%	-21.5%	-17.5%	-29.0%
A2-DaylightRoad2	-17.1%	-25.3%	-21.4%	-31.8%
A2-ParkRunning3	-6.8%	-11.9%	-8.9%	-12.1%
Class A (2160p)	-10.5%	-19.0%	-14.4%	-23.0%
B-BasketballDrive	-12.4%	-11.7%	-14.8%	-19.5%
B-BQTerrace	-16.5%	-25.5%	-20.7%	-29.9%
B-Cactus	-13.2%	-16.0%	-14.6%	-21.7%
B-MarketPlace	-7.3%	-14.9%	-9.6%	-19.7%
B-RitualDance	-6.0%	-13.8%	-10.7%	-18.3%
Class B (1080p)	-11.1%	-16.4%	-14.1%	-21.8%
C-BasketballDrill	-9.4%	-8.1%	-14.4%	-16.1%
C-BQMall	-10.8%	-12.9%	-13.6%	-20.9%
C-PartyScene	-7.3%	-14.7%	-13.6%	-19.9%
C-RaceHorses	-7.8%	-12.5%	-10.2%	-15.8%
Class C (480p)	-8.8%	-12.1%	-13.0%	-18.2%
D-BasketballPass	-8.8%	-11.9%	-12.3%	-13.9%
D-BlowingBubbles	-7.4%	-12.5%	-11.6%	-17.9%
D-BQSquare	-14.9%	-23.6%	-24.1%	-32.5%
D-RaceHorses	-9.0%	-13.1%	-10.7%	-15.0%
Class D (240p)	-10.0%	-15.3%	-14.7%	-19.8%
Overall	-10.2%	-16.0%	-14.1%	-21.0%

on all test sequences when integrated into HEVC, with average BD-rates of -10.2% and -14.1% for PP and ILF respectively. The coding gains are reduced for VTM, but are still significant with average BD-rates of -4.6% and -6.7% for ILF and PP respectively based on the assessment of PSNR. It can also be seen that, for both host codecs and both tested coding modules, the bitrate savings according to VMAF are generally higher than those for PSNR.

As shown in Tables 4.3 and 4.4, the coding gains for PP are consistently higher than those for ILF, by approximately 2% for VTM and 4-5% for HM (in terms of BD-rate). This may at first appear surprising but it should be remembered that, unlike conventional post-processing,

Table 4.4 Compression results of the MFRNet-based ILF and PP for VTM 7.0.

Class-Sequence	CNN-based ILF (*Full Test*)		CNN-based PP	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
A1-Campfire	-4.5%	-8.3%	-6.8%	-10.3%
A1-FoodMarket4	-3.5%	-7.3%	-5.0%	-9.0%
A1-Tango2	-5.9%	-6.2%	-8.4%	-7.1%
A2-CatRobot1	-6.7%	-6.5%	-9.3%	-9.0%
A2-DaylightRoad2	-7.6%	-8.3%	-9.5%	-10.0%
A2-ParkRunning3	-1.5%	-3.1%	-2.7%	-3.5%
Class A (2160p)	-5.0%	-6.6%	-7.0%	-8.2%
B-BasketballDrive	-4.3%	-5.5%	-7.2%	-6.2%
B-BQTerrace	-6.9%	-5.2%	-8.1%	-7.2%
B-Cactus	-4.4%	-5.4%	-6.7%	-7.5%
B-MarketPlace	-3.3%	-4.4%	-4.4%	-5.9%
B-RitualDance	-2.8%	-3.9%	-5.2%	-6.3%
Class B (1080p)	-4.3%	-4.9%	-6.3%	-6.6%
C-BasketballDrill	-4.4%	-1.7%	-6.7%	-4.8%
C-BQMall	-4.2%	-5.5%	-7.4%	-7.8%
C-PartyScene	-2.0%	-3.4%	-6.1%	-5.4%
C-RaceHorses	-2.5%	-5.4%	-3.7%	-5.7%
Class C (480p)	-3.3%	-4.0%	-6.0%	-5.9%
D-BasketballPass	-6.2%	-4.6%	-7.4%	-5.5%
D-BlowingBubbles	-4.6%	-3.6%	-5.9%	-5.2%
D-BQSquare	-6.6%	-3.1%	-11.5%	-12.4%
D-RaceHorses	-4.6%	-5.6%	-5.7%	-5.9%
Class D (240p)	-5.5%	-4.2%	-7.6%	-7.3%
Overall	-4.6%	-5.1%	-6.7%	-7.1%

CNN-based PP does employ end-to-end training. In addition, when CNN-processed frames are employed as a reference (after in-loop filtering), they are used to predict subsequently encoded frames through motion estimation and compensation. This process has not been reflected in the current CNN training (i.e. with CNN-processed content as network input),

Table 4.5 Compression results of the MFRNet-based EBDA and SRA for HM 16.20.

Class-Sequence	CNN-based EBDA		CNN-based SRA	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
A1-Campfire	-20.5%	-29.3%	-30.1%	-44.1%
A1-FoodMarket4	-8.4%	-17.6%	-14.5%	-26.9%
A1-Tango2	-11.6%	-22.7%	-20.8%	-28.4%
A2-CatRobot1	-15.4%	-26.1%	-11.3%	-25.9%
A2-DaylightRoad2	-17.6%	-28.1%	+0.5%	-26.8%
A2-ParkRunning3	-15.9%	-21.4%	-28.7%	-35.0%
Class A (2160p)	-14.9%	-24.2%	-17.5%	-31.2%
B-BasketballDrive	-14.4%	-17.8%	–	–
B-BQTerrace	-17.3%	-31.9%	–	–
B-Cactus	-13.5%	-20.1%	–	–
B-MarketPlace	-7.5%	-16.1%	–	–
B-RitualDance	-8.7%	-15.9%	–	–
Class B (1080p)	-12.3%	-20.4%	–	–
C-BasketballDrill	-12.4%	-16.3%	–	–
C-BQMall	-11.6%	-16.2%	–	–
C-PartyScene	-11.4%	-15.1%	–	–
C-RaceHorses	-10.1%	-16.3%	–	–
Class C (480p)	-11.4%	-16.0%	–	–
D-BasketballPass	-12.8%	-14.6%	–	–
D-BlowingBubbles	-10.1%	-14.1%	–	–
D-BQSquare	-20.6%	-24.1%	–	–
D-RaceHorses	-11.5%	-16.0%	–	–
Class D (240p)	-13.8%	-17.2%	–	–
Overall	-13.2%	-20.0%	-17.5%	-31.2%

and is likely to cause the CNN-based filter to become less effective. Similar results have been observed by other authors when the same CNN is employed for both PP and ILF [215][1].

[1]It is also noted that, in Table 4.7, the ILF results are better than PP for [216, 217]. This is because these CNN models employed for PP have been re-trained using data that is different [170] from that in their original literature.

Table 4.6 Compression results of the MFRNet-based EBDA and SRA for VTM 7.0.

Class-Sequence	CNN-based EBDA		CNN-based SRA	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
A1-Campfire	-7.6%	-18.6%	-12.3%	-33.8%
A1-FoodMarket4	-0.7%	-6.9%	-1.2%	-12.9%
A1-Tango2	-3.6%	-5.4%	-7.3%	-10.5%
A2-CatRobot1	-5.2%	-6.7%	+9.2%	-1.2%
A2-DaylightRoad2	-8.2%	-10.6%	+27.1%	-6.1%
A2-ParkRunning3	-0.4%	-6.0%	-11.4%	-17.9%
Class A (2160p)	-4.3%	-9.0%	+0.7%	-13.7%
B-BasketballDrive	-5.5%	-6.1%	–	–
B-BQTerrace	-2.0%	-3.1%	–	–
B-Cactus	-4.6%	-8.4%	–	–
B-MarketPlace	-1.1%	+2.6%	–	–
B-RitualDance	-1.5%	-5.9%	–	–
Class B (1080p)	-2.9%	-4.2%	–	–
C-BasketballDrill	-1.5%	-6.2%	–	–
C-BQMall	-4.7%	-5.4%	–	–
C-PartyScene	-3.5%	-3.2%	–	–
C-RaceHorses	-2.5%	-2.9%	–	–
Class C (480p)	-3.1%	-4.4%	–	–
D-BasketballPass	-7.1%	-7.6%	–	–
D-BlowingBubbles	-3.9%	-3.2%	–	–
D-BQSquare	-7.3%	-0.9%	–	–
D-RaceHorses	-4.5%	-3.5%	–	–
Class D (240p)	-5.7%	-3.8%	–	–
Overall	-4.0%	-5.7%	+0.7%	-13.7%

Compression Results for SRA and EBDA Coding Modules

Tables 4.5 and 4.6 summarise the compression performance of the MFRNet-based SRA and EBDA coding tools benchmarked on the HEVC HM 16.20 and VVC VTM 7.0. Similar to the previous PP and ILF compression results, MFRNet has also achieved evident coding gains on JVET-CTC test sequences when integrated into the HEVC HM and VVC VTM based on the assessment of VMAF, with average BD-rates of 31.2% (SRA) and 20.0% (EBDA) for HM 16.20, and 13.7% (SRA) and 5.7% (EBDA) for VTM 7.0. It can be observed that

the coding gains are higher when the VMAF is employed as the quality metric compared to those for PSNR for both SRA and EBDA coding modules based on the two standard codecs.

The MFRNet has been applied to enhance compression performance of four individual coding modules to evaluate its effectiveness in this chapter. It is also possible to employ MFRNet for multiple coding modules simultaneously to achieve even higher overall coding gains, as it demonstrated in [57] (SRA+EBDA). In this chapter, as our focus is on comparing network architectures, we have not provided more results for multiple coding modules.

Table 4.7 Comparison between MFRNet-based PP and ILF and existing CNN-based PP and ILF approaches for HEVC.

Compression Performance Comparisons for Post-Processing Tools (HEVC HM)						
Sequence (Class)	[217]	[216]	[218]	[167]	[170]	Ours
	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)
Class C (480p)	0.63%	-2.6%	-6.8%	-6.6%	-7.1%	**-13.0%**
Class D (240p)	1.73%	-2.6%	-8.0%	-4.8%	-7.3%	**-14.7%**
Overall	1.2%	-2.6%	-7.4%	-5.7%	-7.2%	**-13.9%**
Compression Performance Comparisons for In-loop Filtering Tools (HEVC HM)						
Sequence (Class)	[219]	[217]	[216]	[171]	[117]	Ours
	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)
Class C (480p)	-4.6%	-3.0%	-3.9%	-7.1%	-4.5%	**-8.8%**
Class D (240p)	-2.5%	-2.3%	-4.6%	-4.4%	-3.3%	**-10.0%**
Overall	-3.6%	-2.7%	-4.3%	-5.8%	-3.9%	**-9.4%**

4.3.3 Comparison between CNN-based PP and ILF Approaches

The coding performance of the proposed CNN model is compared here with other notable CNN-based PP and ILF methods developed for the HEVC and VVC Random Access configuration. These include [217, 218, 167, 216, 170, 216, 219, 171, 117, 224, 215, 220–222][2]. It should be noted that these approaches have not been re-implemented due primarily to a lack of their source code. Instead, their compression results are extracted directly from the corresponding literature.

[2]It is noted that [219] has been commonly used as a benchmark for ILF approaches, although it is not a CNN-based solution. This method has been included here due to its consistent performance and popularity.

Table 4.8 Comparison between MFRNet-based PP and ILF and existing CNN-based PP and ILF approaches for VVC.

Compression Performance Comparisons for Post-Processing Tools (VVC VTM)						
Sequence (Class)	[215]	[220]	[221]	[222]	[181]	Ours
	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)
Class A (2160p)	-2.0%	-1.3%	-1.2%	-0.2%	-3.3%	**-7.0%**
Class B (1080p)	-1.3%	-1.5%	0.4%	-0.2%	-2.6%	**-6.3%**
Class C (480p)	0.3%	-3.3%	2.2%	-0.6%	-3.9%	**-6.0%**
Class D (240p)	N/A	-5.0%	6.6%	-0.8%	-5.8%	**-7.6%**
Overall	-1.2%	-2.6%	1.6%	-0.4%	-3.8%	**-6.7%**
Compression Performance Comparisons for In-loop Filtering Tools (VVC VTM)						
Sequence (Class)	[172]	[215]	[220]	[223]	[224]	Ours
	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)
Class A (2160p)	-2.0%	-1.7%	-0.4%	-1.3%	N/A	**-5.0%**
Class B (1080p)	-1.4%	-0.6%	0.6%	-0.8%	-1.5%	**-4.3%**
Class C (480p)	0.2%	0.3%	-1.2%	-0.9%	-3.1%	**-3.3%**
Class D (240p)	N/A	N/A	-3.1%	-0.8%	-3.9%	**-5.5%**
Overall	-1.2%	-0.8%	-0.9%	-1.0%	-2.7%	**-4.6%**

Tables 4.7 and 4.8 summarise BD-rate (PSNR) results for five PP and five ILF methods (described above) for each host codec (HM and VTM) and compare them with our approach. Due to the limitations of results available in the literature, only results for Class C and D are compared for HEVC HM. It can be observed that, for both host codecs and for the two coding modules, when MFRNet is integrated into PP and ILF modules, it significantly outperforms competing methods, and the improvements are consistent across content classes. This is likely due to the advanced structures employed in the MFRNet architecture and the diversity of the training content used.

4.4 Summary

In this chapter, a new CNN architecture-MFRNet is presented for video coding enhancement modules, including post-processing (PP), in-loop filtering (ILF), spatial resolution adaptation

(SRA) and effective bit depth adaptation (EBDA). MFRNet comprises four multi-level feature review residual dense blocks, and employs a cascading structure to improve information flow. The experimental results demonstrate significant coding gains, with a 16.0% improvement for ILF, 21.0% for PP, 31.2% for SRA and 20.0% for EBDA over HM 16.20 in terms of VMAF, and a corresponding 5.1% for ILF, 7.1% for PP, 13.7% for SRA and 5.7% for EBDA against VTM 7.0. Further comparisons have shown the superiority of the MFRNet architecture over other existing popular deep networks when it was integrated with various coding modules.

It is noted that a pixel-wise loss function-$\ell 1$ loss was employed to train MFRNet models for tested approaches in this chapter. As discussed in Section 2.3.4, $\ell 1$ loss does not correlate well with subjective opinion scores and cannot capture the structural information of the images which is important for network training. This may lead to the sub-optimal perceptual quality of the final reconstructed frames. To address this issue, the novel and effective GAN-based algorithms and perceptual loss functions are further presented in the following chapter to generate realistic high frequency details and effectively optimise perceptual quality of compressed videos.

Chapter 5

Perceptually-inspired Deep Video Coding Enhancement

For CNN-based video coding enhancement modules, simple pixel-wise loss functions (i.e. $\ell 1$ loss) are commonly utilised to train the employed network architectures. These pixel-wise loss functions do not correlate well with subjective video quality. This inevitably leads to sub-optimal training performance when these coding methods are designed to improve perceptual video quality. Recently, new CNN-based image and video restoration methods have been proposed based on various Generative Adversarial Networks (GANs) and adversarial training methodologies [29, 89, 1, 2, 25, 7], which can achieve improved enhancement performance.

In this chapter, their applications are extended to video compression and new GAN architectures alongside novel training methodologies are further presented for different coding enhancement tools. These include MSRGAN, BDGAN, and CVEGAN. It is noted that the GAN architectures presented in this chapter are trained using the similar procedures that have been described in Section 2.2.3. This GAN training method contains two steps. Specifically, in the first training stage, the generator is trained independently (without discriminator) based on a typical CNN training methodology using perceptual loss functions (MS-SSIM is used for training the generators of MSRGAN and BDGAN, while a new perceptual loss function is employed to train CVEGAN's generator). Then, the pre-trained generator is utilised as initial model for the second step of training and is jointly trained with the discriminator using the adversarial training strategy (RaGAN algorithm is used for training both MSRGAN and BDGAN, while a novel GAN adversarial training methodology-ReSphereGAN is utilised to train CVEGAN). This two-stage training strategy can further fine-tune the generator's parameters and effectively improve its overall performance [28, 29].

The works presented in this chapter have been published in [1, 2, 7, 5].

5.1 Network Architectures

In this section, two preliminary works: MSRGAN and BDGAN are first presented for spatial resolution and effective bit depth adaptations (SRA and EBDA), respectively. Then, a novel GAN architecture (CVEGAN) and the new training methodologies are further presented for significantly improving coding gains based on the typical video coding enhancement modules.

5.1.1 Preliminary Work-I: MSRGAN for SRA

In an SRA coding framework, when the low resolution video frames are decoded, they are first up-sampled using a nearest neighbour filter to the original resolution[1]. A deep CNN (MSRGAN) is then employed to further enhance reconstruction quality. This is a modified version of the SRGAN model [28]. The architectures of the generator (MSRResNet) and the discriminator are shown in Figures. 5.1 and 5.3.

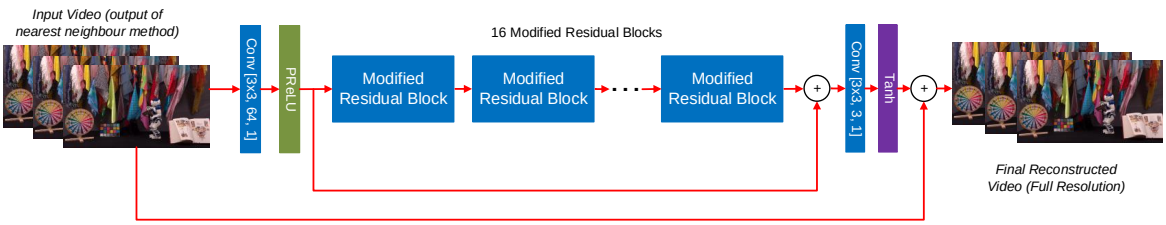


Figure. 5.1 Network architecture of the MSRGAN's Generator (MSRResNet).

The generator (MSRResNet) for the proposed MSRGAN has a similar architecture to the original SRGAN [28]. The input of the model is 96×96 YCbCr (4:4:4), compressed image blocks (nearest neighbour filter up-sampled), with the target to output colour image blocks close to the original at the same resolution.

MSRResNet contains 16 residual blocks after an initial convolutional layer, where each of these has two convolutional layers with a parametric ReLU (PReLU) layer in between. A skip connection is employed in each residual block connecting the input and the output of this block. Two additional skip connections are also used between the output of the initial convolutional layer and that of the 16 residual blocks, and between the input of the network and the output of the final convolutional layer (the one with a Tanh activation). All the convolutional layers in MSRResNet employ a kernel size of 3×3 with a stride of 1. The number of channels is 64, except the final convolutional layer which supports 3 feature maps.

[1] We have previously shown [173, 225] that nearest neighbour filtering can lead to slightly better reconstruction results.

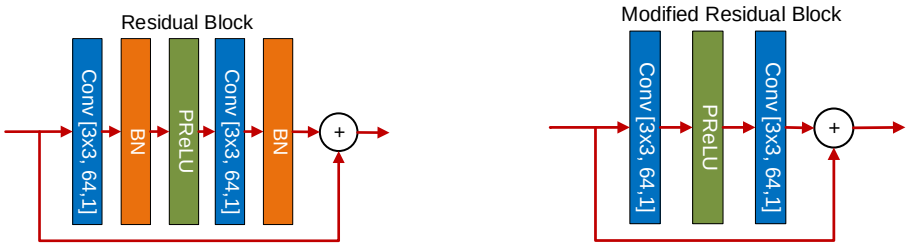


Figure. 5.2 (Left) The original residual block in SRGAN [28]. (Right) The modified residual block in MSRGAN.

Comparing to the original SRResNet, the presented generator does not contain batch normalisation (BN) layers in the residual blocks (as shown in Figure. 5.2). This is because BN layers were reported to affect the overall network performance [29] - generating unexpected artefacts - especially for deep GAN networks.

The discriminator in the presented network has an identical architecture to that of the original SRGAN [28]. Except for the first shallow feature extraction layer, the discriminator concatenates 7 convolutional layers, each of which contains a convolutional operation, a leaky ReLU (LReLU) and a BN. The kernel size of each convolutional layer is 3×3 with a stride of 1 or 2. Different numbers of feature maps are also employed in these layers, from 64 to 512 as shown in Figure. 5.3. After 7 convolutional layers, 2 dense layers with a LReLU followed by a Sigmoid layer are designed to produce the output of the discriminator.

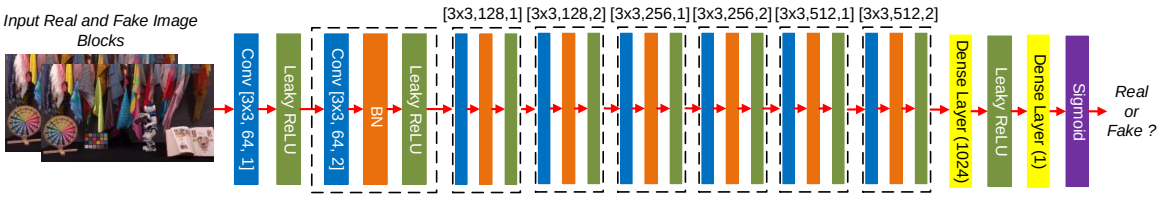


Figure. 5.3 Network architecture of the Discriminator for MSRGAN.

5.1.2 Preliminary Work-II: BDGAN for EBDA

The presented generative adversarial network architecture for EBD up-sampling (BDGAN) contains a generator (denoted as BDNet) and a discriminator. The BDNet structure is shown in Figure. 5.4, which takes a 96×96 YCbCr (4:4:4) decoded colour image block with effective bit depth (EBD) of 9 bit as input, and produces a 10 bit EBD image block in the same format, targeting the corresponding original block.

The network starts with two shallow feature extraction layers, each of which contains a convolutional layer and a LReLU [29] activation function. These are followed by 14 identical

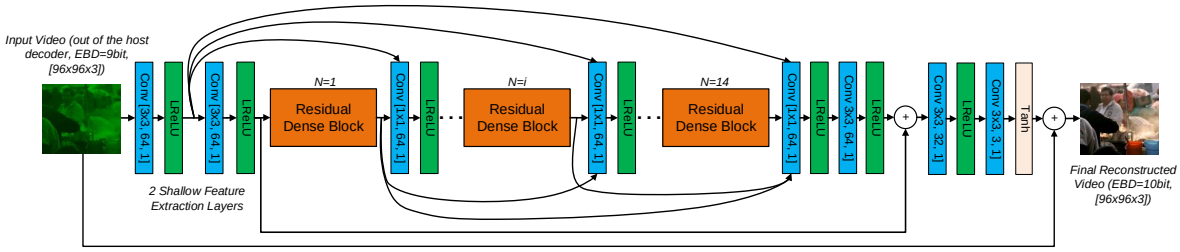


Figure. 5.4 Network architecture of the BDGAN's Generator (BDNet).

residual dense blocks (RDB) [66] as illustrated in Figure. 5.5. This RDB is different from the residual block used in SRGAN [28] - each convolutional layer in the RDB receives the feature maps from all preceding layers and feeds its output to subsequent layers. This structure preserves the feed-forward nature of the network and enables access to more previously extracted features, which improves the information flow [67]. An additional skip connection is employed in each RDB between the input of this RDB and the output of the last convolutional layer to further improve information flow and stabilise the training process [55].

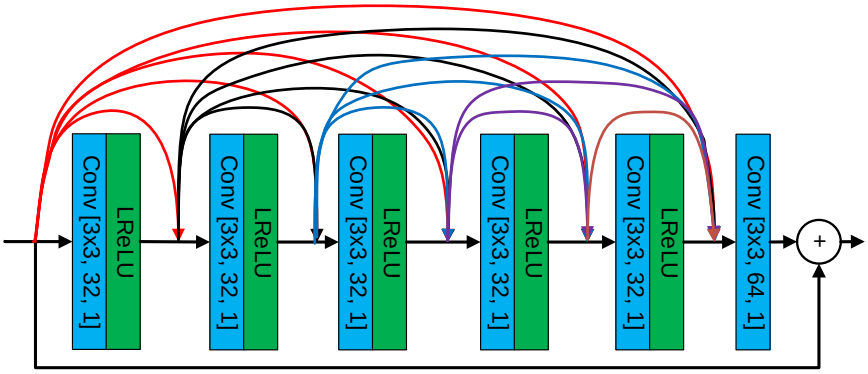


Figure. 5.5 Residual Dense Block (RDB) used in BDNet.

Moreover, the output of the first shallow feature extraction layer and the output of each RDB are cascaded [69, 85] into all the following RDBs through a single 1×1 convolutional layer. After 14 RDBs, another convolutional layer is employed. A skip connection is applied between the output of this layer and the output of the second shallow feature extraction layer. Finally, another two convolutional layers are used before the final output, and an additional skip connection is employed between the input of these two layers and the initial BDNet input. The stride value for all convolutional layers in BDNet is 1.

The BDGAN's discriminator has an identical structure to that of the MSRGAN shown in Figure. 5.3, which accepts the real and fake (output from the BDNet) image blocks with a full bit depth of 10 bit as input.

5.1.3 A New GAN Architecture: CVEGAN

As discussed in Section 2.3.4, most of the previous works employ simply combined loss functions (including, pixel-wise loss, simple quality metrics, feature map differences) with artificially configured combining weights in the GAN training process for image restoration and video compression enhancement. These loss functions have not been fully evaluated in terms of their correlation with subjective image or video quality. Additionally, most of GAN-based video coding enhancement methods often use relatively simple network architectures without including advanced structures (e.g. channel attention mechanism, etc.). These all lead to sub-optimal coding performance.

In this context, a novel GAN architecture for compressed video enhancement, denoted as CVEGAN, is presented, which contains novel Mul^{2}Res block, enhanced residual non-local block and enhanced convolutional block attention module.

The CVEGAN Architecture

CVEGAN follows the basic GAN framework [81], combining a generator and a discriminator. Its architecture is described in the following subsections.

Generator Architecture: The CVEGAN's generator, denoted as CVENet, is shown in Figure. 5.6. This takes a 96×96 YCbCr 4:4:4 compressed image block as the input and outputs a processed image block in the same format, targeting its original uncompressed version. The kernel sizes, a number of feature maps and stride values for each convolutional layer of CVENet are shown in Figure. 5.6.

CVENet has three unique structural features: (1) Mul^{2}Res blocks; (2) enhanced residual non-local blocks (ERNB); (3) enhanced convolutional block attention modules (ECBAM). These are described below. It can be seen that a Mish activation function [226] has been employed after all convolutional operations except for the final one. Mish has been previously reported to offer better performance than other commonly used functions such as ReLU, LReLU and PReLU [226].

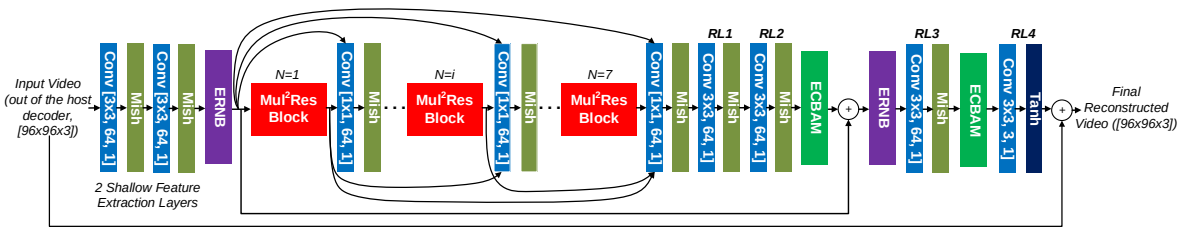


Figure. 5.6 Illustration of the CVEGAN's Generator (CVENet).

Specifically, CVENet first employs two shallow feature extraction (SFE) layers to extract local shallow features. Each of them contains a convolutional function along with a Mish activation function [226]. Followed these two SFE layers, an enhanced residual non-local block (ERNB) is employed to capture long-range dependencies between pixels within the input feature maps through the non-local operations [60, 73]. The outputs of this ERNB are then fed into seven Mul^{2}Res blocks to further extract deep features. In order to improve information flow between the convolutional layers and blocks in the network, the cascading connections [69, 4] (shown as black curves in Figure. 5.6) are utilised to connect these seven Mul^{2}Res blocks by feeding the first ERNB's outputs and the first six Mul^{2}Res blocks into following Mul^{2}Res blocks or into the first reconstruction layer (RL1 in Figure. 5.6) through a 1×1 convolutional layer alongside a Mish activation function. After RL1, the second reconstruction layer (RL2) and an enhanced convolutional block attention module (ECBAM) are employed to extract high-level features and apply channel and spatial attention mechanisms. A skip connection is further used to connect the outputs of this ECBAM and the first ERNB. Finally, a second ERNB is employed, which is followed by another two reconstruction layers (RL3 and RL4) and an ECBAM in between. The output of RL4 is then combined with the network input through a long skip connection to obtain the final image block. The skip connections used in CVENet are designed to bypass redundant information and stabilise training and evaluation processes [55, 212]. The architectures of the Mul^{2}Res block, ERNB and ECBAM used in CVENet are introduced below in detail.

Mul^{2}Res Block: As discussed in Chapter 2, the performance of deep CNNs is highly relevant to three primary factors: *depth* (i.e. the depth of networks), *width* (i.e. the number of feature maps), and *cardinality* (i.e. the size of transformation sets) [43, 74, 75]. It is noted that simply increasing the *depth* or *width* of the networks may have a potential risk of the vanishing gradient problem [43, 55]. This can lead to the difficulty of network training and the overall performance of the model cannot be effectively improved. It has been described in [75, 79] that the *cardinality* is a more effective way to improve representational ability and capacity of the network while maintaining appropriate computational complexity compared to other two factors.

In this context, we exploit the effectiveness of *cardinality* in network architecture and present a novel and highly modularised residual learning structure, Mul^{2}Res block. The new Mul^{2}Res Block structure contains multiple levels of multiple residual learning branches to exploit the *cardinality* characteristic of networks. Figure. 5.7 illustrates the Mul^{2}Res Block structure used in CVEGAN, which has four residual learning branches at two different levels. The number of branches and levels can be adapted for different applications based on the computational resources available.

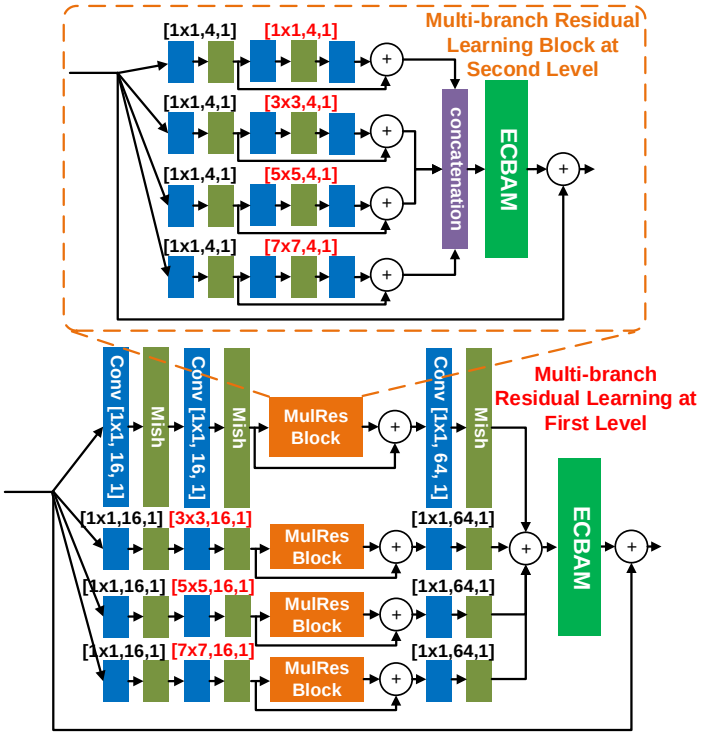


Figure. 5.7 Illustration of an Mul^{2}Res Block.

At the first level, the input of the Mul^{2}Res block is fed into four residual learning branches. Each branch has a convolutional layer with various kernel sizes (1×1, 3×3, 5×5 and 7×7). This diversifies the feature maps extracted by the convolutional layers using various receptive field sizes. The ECBAM is also employed at both levels before the final skip connection.

The Mul^{2}Res block at the second level also has four residual learning branches. The primary differences include (i) the MulRes Blocks at the first level are replaced by residual blocks, each of which contains two convolutional layers (with various kernel sizes for different branches) and a Mish activation function; (ii) the output from four branches are concatenated before feeding into the ECBAM.

Based on the discussion above, the proposed structure with multiple levels of multiple branches has been designed to increase the network's *cardinality*. This supports different kernel sizes which enable operations with various receptive fields and is expected to offer improved performance for content with complex textures [75, 210]. Moreover, ECBAM has been utilised in the Mul^{2}Res blocks to extract informative features from the source data in order to further improve the reconstruction ability of the network [68].

Enhanced Residual Non-local Block (ERNB): As discussed in Chapter 2, most of CNNs employed for image/video restoration and video compression enhancement often utilised conventional local convolution filters to extract features. In this case, the response of

each feature point is only based on the features within a local convolutional area which is determined by the kernel size of filters [43]. These convolutional operations with small kernel sizes cannot capture longer-range dependencies between pixels within the input image block or feature maps, hence the representational ability of networks cannot be further optimised [60, 73].

To address this issue, non-local operations have been previously proposed for image and video classification and recognition [60], and were also used for image restoration tasks [73]. The main idea of the non-local operation is that the response of a feature point with the index of i is computed based on all possible feature points (outside of the local convolutional kernel) within the whole feature map [60, 73].

The non-local operation can be generally described in equation (5.1) [60, 73]:

$$y_i = \sum_{\forall j} f(\psi(x_i), \xi(x_j))\omega(x_j) \tag{5.1}$$

where, y_i is the output of non-local operation for feature point i, j is the index which enumerates all possible feature point positions, x_i and x_j represent the input features at the indices of i and j respectively, $\psi(\cdot)$ and $\xi(\cdot)$ are two transformations that separately extract features based on the inputs x_i and x_j at different feature point indices, $f(\cdot)$ is a pairwise function used to fuse $\psi(x_i)$ and $\xi(x_j)$, and the $\omega(\cdot)$ is a weighting function producing the corresponding weights based on the input feature point x_j at the index of j. As described in [60, 73], the transformations $\psi(\cdot)$ and $\xi(\cdot)$, as well as the weighting function $\omega(\cdot)$ are implemented using three different 1×1 convolutional layers [60, 73].

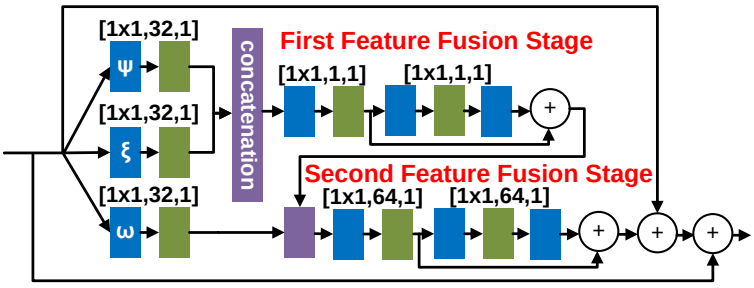


Figure. 5.8 Illustration of an ERNB.

It is noted that, the pairwise function $f(\cdot)$ in the original non-local algorithm was implemented using an embedded Gaussian function incorporated with a large matrix multiplication operation [60, 73]. This inevitably leads to much higher computational complexity, especially when the large feature maps are processed [227]. In this context, an enhanced residual non-local block (ERNB) was presented, as shown in Figure. 5.8. This employs concatenation operations and residual blocks to achieve feature fusion, avoiding large matrix

multiplication and facilitating training of the feature fusion process. A further modification is an introduction of a long skip connection employed to produce the ERNB output; this is designed to further stabilise the non-local learning.

Enhanced Convolutional Block Attention Module (ECBAM): In order to extract more informative and important features, attention mechanisms have been recently proposed for several computer vision tasks, including image classification, recognition, as well as image and video restoration [68, 59, 73, 107]. The experimental results have demonstrated that integrating attention mechanisms can effectively improve the representational ability and overall performance of networks [68, 59].

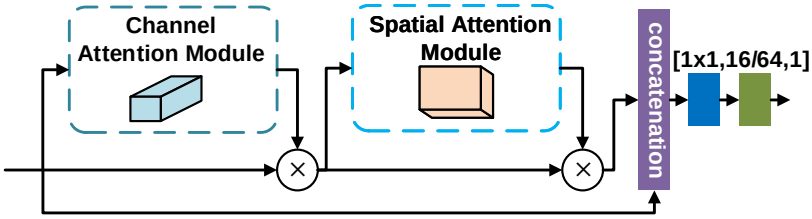


Figure. 5.9 Illustration of an ECBAM.

Inspired by the previous works, an enhanced convolutional block attention module (ECBAM) has been designed for CVEGAN (Figure. 5.9). This follows the basic structure of the convolutional block attention module (CBAM) [59], which comprises a channel and a spatial attention module. Rather than directly producing output from the second matrix multiplication as in [59], we have added a concatenation operation with a convolutional layer to achieve non-linear feature fusion. This has been previously reported to improve information flow and overall performance of the network [66].

Discriminator Architecture: Figure. 5.10 illustrates the architecture of the CVEGAN discriminator, which is based on SRGAN [28]. The primary differences include: (i) we employ two ERNBs - one after the first convolutional layer, and the other before the final convolutional layer - these extract non-local features and improve the representational capability; and (ii) we remove the final dense layer in SRGAN to output high (1024) dimensional feature points rather than a single 1D scalar.

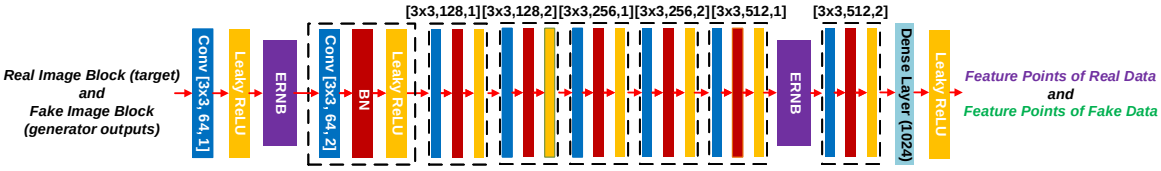


Figure. 5.10 Illustration of the CVEGAN's Discriminator.

5.1.4 Relationship between MSRGAN, BDGAN and CVEGAN

The MSRGAN and BDGAN are two preliminary works of GAN architectures for video coding enhancement in the thesis. Inspired by the MSRGAN, BDGAN's generator (BDNet) also removes BN layers in all convolutional blocks in order to avoid generating visually unpleasing artefacts. Different from MSRGAN's generator (MSRResNet), BDNet utilises cascading structures and residual dense connections to improve information flow and the overall performance of the network. Based on both MSRGAN and BDGAN, CVEGAN is further presented to significantly improve coding gains and visual quality of compressed video content. It is an enhanced version of these two networks, which not only uses cascading structures to improve information flow but also employs a couple of novel structures. These include: (1) Mul^{2}Res blocks which exploit multiple levels of multiple residual learning branches with different kernel sizes to increase *cardinality*, receptive field and capacity of the network; (2) ERNB that captures longer-range dependencies between pixels within feature maps or input image block to improve representational ability of the network; and (3) ECBAM module which conducts enhanced channel-spatial attention mechanisms to effectively improve the reconstruction ability of the network.

MSRGAN and BDGAN utilise the same discriminator architecture to output a 1D scalar for GAN adversarial training (entropy-based). CVEGAN's discriminator has the similar structures to that of used in MSRGAN and BDGAN. The main difference is that it further employs ERNB to improve representational ability of the network by extracting non-local feature information and outputs high dimensional (1024) feature points for GAN adversarial training (an *integral probability metric*-based algorithm and it will be discussed in detail in Section 5.2.2).

5.2 Network Training and Evaluation

In this section, the training and evaluation methodologies for preliminary works-MSRGAN and BDGAN, as well as the new CVEGAN architecture are described in detail.

5.2.1 Training Methodology-MSRGAN and BDGAN

Training Data Generation

We follow the procedures of training data generation for SRA and EBDA coding modules that are introduced in Section 3.2.3 to generate training material for MSRGAN and BDGAN.

It is noted that the host codec utilised to test these two preliminary works is based on the HEVC HM 16.20.

Loss Functions

Loss functions play a crucial role during the training of CNN/GAN models. In this work, in order to generate results with improved perceptual quality and realise efficient training, the loss function employed should ideally correlate well with subjective opinions and exhibit relatively low computational complexity. Based on these considerations, the multi-scale structural similarity index (MS-SSIM) [27] and SSIM [26] have been employed to train the presented networks[2]. MS-SSIM has been previously used to train CNN models (i.e. MSRResNet and BDNet) similar to [183], while SSIM has also been employed in the training of generative adversarial networks to achieve better visual quality [228].

The training of MSRGAN and BDGAN consists of two stages [28, 29]. At the first stage, the original MS-SSIM is used as the loss function to train the generator (MSRResNet and BDNet). The resulting models are then employed as initial models for the second stage of training.

At the second stage, in contrast to the original SRGAN [28], the loss functions from Relativistic GANs (RaGANs) [87] have been used to further stabilise the training process and improve the performance of the discriminator network [29], which is described below:

$$\mathcal{L}_G = 0.025 \times \ell 1 + \mathcal{L}_{\text{SSIM}} + 5 \times 10^{-3} \times \mathcal{L}_G^{Ra} \tag{5.2}$$

$$\mathcal{L}_D = \mathcal{L}_D^{Ra} = -E_{x_r}[\ln(\text{Sig}(C_d(x_r) - E_{x_f}[C_d(x_f)]))] - E_{x_f}[\ln(1 - (\text{Sig}(C_d(x_f) - E_{x_r}[C_d(x_r)])))] \tag{5.3}$$

where, $\mathcal{L}_G$ represents the loss function of the generator, while $\mathcal{L}_{\text{SSIM}}$ and $\mathcal{L}_G^{Ra}$ represent the SSIM loss and adversarial loss of the generator respectively. The adversarial loss of the generator $\mathcal{L}_G^{Ra}$ is defined by (5.4):

$$\mathcal{L}_G^{Ra} = -E_{x_r}[\ln(1 - (\text{Sig}(C_d(x_r) - E_{x_f}[C_d(x_f)])))] - E_{x_f}[\ln(\text{Sig}(C_d(x_f) - E_{x_r}[C_d(x_r)]))] \tag{5.4}$$

where, E stands for the mean operation, x_r and x_f are the real and fake image block respectively, and $C_d(\cdot)$ is the output of the discriminator of MSRGAN or BDGAN. 'Sig' here represents the Sigmoid function.

[2]During the training of the original SRGAN, $\ell 1$ (mean absolute difference) and VGG19 [84] were employed in the loss functions. It is however noted that mean absolute difference correlates poorly with subjective results, and a VGG19 based loss function also fails to perform well in the training of MSRGAN and BDGAN on compressed content.

In equation (5.3), $\mathcal{L}_D$ and $\mathcal{L}_D^{Ra}$ are the loss functions for the discriminator and the adversarial loss of the discriminator respectively.

5.2.2 Training Methodology-CVEGAN

Training Data Generation

Here, the same training data generation methodologies presented in Section 3.2.3 were also utilised to create training material for CVEGAN. It is noted that CVEGAN is tested based on HEVC HM 16.20 and VVC VTM 7.0.

Perceptually-inspired Loss Function

A similar training strategy to that used for MSRGAN and BDGAN architectures was employed to train CVEGAN, which consists of two stages: (i) CVENet is trained using a combined perceptual loss function to obtain an initial model for the second training step; (ii) CVENet is then trained jointly with the discriminator with a new training method ReSphereGAN. The new perceptual loss function is first presented in this subsection and a novel ReSphereGAN adversarial training methodology is further introduced in the following subsection.

In CVEGAN, a new loss function is employed which is a linear combination of the elementary transforms of six commonly used losses, $\ell 1$ (denoted as L_1), $\ell 2$ (L_2), gradient loss [77, 229] (L_3), VGG19-54 loss [29] (L_4), Structural Similarity Index (SSIM) loss [26] (L_5) and Multi-scale SSIM (MS-SSIM) loss [27] (L_6):

$$L_{\text{test}} = \sum_{i=1}^{6} a_i f(L_i) \tag{5.5}$$

Here, $f(\cdot)$ represents an elementary transformation [230], which can be either a constant, a power, a root, an exponential, a logarithmic, a trigonometric, an inverse trigonometric, a hyperbolic or an inverse hyperbolic function. a_i represents the linear combination weights, where a_6 is always set to 1. The range of all these six single losses is between 0 and 1. We have excluded non-linear combinations and combinations of different transformed losses due to their high computational and training complexity.

An eight-fold cross validation method [231] has been used to train the proposed combined loss function (equation (5.5)) based on eight publicly available subjective video quality databases. These include: the Netflix public database (70 test sequences) [232], BVI-HD (192) [175], CC-HD (108) [14], CC-HDDO (90) [233], MCL-V (96) [137], SHVC (32)

[234], IVP (100) [235], and VQEG-HD3 (72) [236]. All of these contain video sequences compressed using commonly used video codecs (H.264, HEVC, AV1, VVC or MPEG-2).

The eight databases were divided into two sub groups - seven training datasets and one for testing. An exhaustive search was performed among all tested transformation functions and their corresponding weighting parameters, the best of which was selected for this split to achieve the highest average correlation between the combined loss values and subjective scores for all seven training datasets. The Spearman-rank-order-correlation-coefficient (SROCC) is employed to quantify the correlation performance. The search range of each parameter is between 0 and 1, with an interval of 0.1.

Table 5.1 Cross-validation results over eight training-testing trails.

Loss Function	L_1	L_2	L_3	L_4	L_5	L_6
Average SROCC	0.5984	0.6478	0.3991	0.7085	0.5720	0.7168
Loss Function	L_7	L_8	L_9	L_{10}	L_{11}	Ours
Average SROCC	0.5430	0.5993	0.6008	0.6198	0.6591	**0.8067**

To avoid possible content bias due to a single training-testing split, this cross-validation method was performed for all eight splits. Table 5.1 presents the average SROCC performance on the test datasets among all eight splits for the trained loss functions, which are compared to the results from 11 commonly used loss functions in training image restoration and enhancement CNNs, L_1-L_6; linear combination of $\ell 1$ loss, gradient loss and VGG19-54 perceptual loss [83] (L_7); linear combination of $\ell 1$ loss and SSIM loss [1] (L_8); linear combination of $\ell 1$ loss and VGG19-54 perceptual loss [29, 85, 88, 89, 76] (L_9); linear combination of $\ell 1$ loss, gradient loss, SSIM and MS-SSIM losses and VGG19-54 perceptual loss [77] (L_{10}); linear combination of MSE and VGG19-54 perceptual loss [28] (L_{11}). It can be observed that our trained loss functions have an average SROCC value of 0.8067, which is significantly higher than those for other tested loss functions (L_1-L_{11}).

The transformation function used for all optimal loss functions across all eight training-testing splits, is the natural logarithm $ln(\cdot)$. We use the median values of the corresponding combination parameters, and normalise these to ensure $\sum_{i=1}^{6} a_i = 1$. The final combined loss function $\mathcal{L}_P$ used to train CVEGAN is given below:

$$\begin{aligned} \mathcal{L}_P = \; & 0.3 \cdot ln(\ell 1) + 0.2 \cdot ln(\text{SSIM\_loss}) + \\ & 0.1 \cdot ln(\ell 2) + 0.4 \cdot ln(\text{MSSSIM\_loss}) \end{aligned} \tag{5.6}$$

It should be noted that $\mathcal{L}_P$ remains within the range 0 to 1, and is differentiable, which is required to support back-propagation during training. Since the test sequences were

generated using a wide range of video codecs, this loss function should generalise well across image and video compression applications. $\mathcal{L}_P$ has been used here for training the CVENet (generator) during the first training stage.

GAN Adversarial Training Methodology

Here, the original SphereGAN [237] is first introduced which has been reported to provide better performance compared to other state-of-the-art GAN training methodologies. Based on this, a novel GAN training strategy, Relativistic SphereGAN (ReSphereGAN) is then presented which has effectively achieved evident improvements on network performance compared to the original SphereGAN.

The Original SphereGAN [237]: In [237], a new *integral probability metric* (IPM)-based GAN training methodology, SphereGAN was proposed and primarily used for unsupervised image generation tasks. This algorithm firstly projects feature points (outputs from the discriminator-shown as the purple and green feature points on the yellow plane in Figure. 5.11) of real (target data) and fake (generator's output) data from n-dimensional Euclidean feature space $\mathbb{R}^n$ (the yellow plane in Figure. 5.11) to the n-dimensional hypersphere space $\mathbb{S}^n$ (the yellow sphere in Figure. 5.11) through the inverse stereographic projection [237]. Then, the geometric moments (geodesic distances) between the north pole **N** and fake/real feature points were calculated using the *geometric-aware transformation function* derived in [237]. Based on these geodesic distances, the adversarial training process is applied to train generator and discriminator. The generator is to 'fool' discriminator by minimising the geodesic distance between the north pole and fake data feature points, while the discriminator tries to distinguish real and fake data by maximising the moment (distance) differences. The geodesic distances calculated in the hypersphere space are bounded in the hypersphere space, demonstrating the SphereGAN ensures stable GAN training [237].

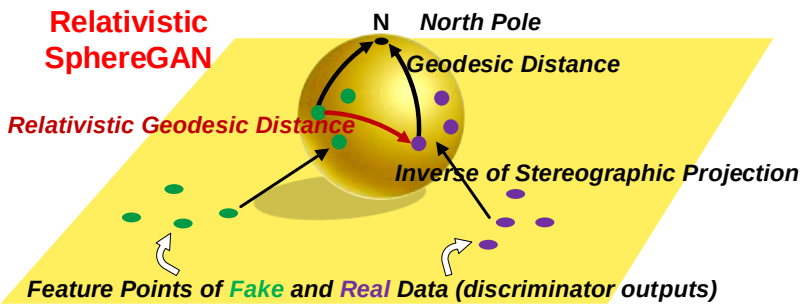


Figure. 5.11 Illustration of the proposed ReSphereGAN. The yellow plane and sphere represent the 1024-dimensional Euclidean feature space and hypersphere respectively. Green and purple points represent the feature points of fake and real data respectively.

The objective functions for generator and discriminator in original SphereGAN are shown below:

$$\begin{cases} \mathcal{L}_{gen} = -\sum_{m=1}^{M} E(d^m(\mathbf{N}, T(\mathbf{x_f}))), \\ \mathcal{L}_{disc} = \sum_{m=1}^{M} E(d^m(\mathbf{N}, T(\mathbf{x_f}))) - \sum_{m=1}^{M} E(d^m(\mathbf{N}, T(\mathbf{x_r}))) \end{cases} \tag{5.7}$$

where, $\mathcal{L}_{gen}$ and $\mathcal{L}_{disc}$ represent the loss functions of generator and discriminator respectively, $E(\cdot)$ represents the mean operation. $d^m(\mathbf{a}, \mathbf{b})$ is the *geometric-aware transformation function* [237], which calculates the *m-th* central geometric moment (geodesic distance) [237] in the hypersphere space between $\mathbf{a}$ and $\mathbf{b}$ (projected feature points in the hypersphere space). M is the moment value and set to 3 [237]. $T(\cdot)$ stands for the *inverse of stereographic projection* [237] from the Euclidean feature space to the hypersphere space. $\mathbf{N}$ is denoted as the north pole in the hypersphere space, and $\mathbf{x_r}$ and $\mathbf{x_f}$ are the real and fake feature points respectively in n-dimensional Euclidean feature space.

It can be observed from equation (5.7) that the loss function $\mathcal{L}_{gen}$ of the generator in original SphereGAN only related to the fake data $\mathbf{x_f}$ which cannot receive informative feedback from the real data $\mathbf{x_r}$. This will lead to the sub-optimal adversarial training performance [29, 87].

Relativistic SphereGAN (ReSphereGAN): In order to address the issues of the original SphereGAN discussed above, we further presented the Relativistic SphereGAN training methodology as illustrated in Figure. 5.11, which is a modified version of the original SphereGAN [237]. Inspired by RaGAN [87], the relativistic geodesic distance between the projected real and fake feature points is also calculated in our loss functions at the second training stage. This modification further optimises the generator by obtaining gradient information from both real and fake data during the adversarial training process. Specifically, the loss functions for generator ($\mathcal{L}_{Re\_gen}$) and discriminator ($\mathcal{L}_{Re\_disc}$) (in the second training stage) are given below:

$$\begin{aligned} \mathcal{L}_{Re\_gen} = \quad & \mathcal{L}_P + 0.005 \cdot (-\sum_{m=1}^{M} E(d^m(\mathbf{N}, T(\mathbf{x_f}))) \\ & + \sum_{m=1}^{M} E(d^m(T(\mathbf{x_r}), T(\mathbf{x_f})))) \end{aligned} \tag{5.8}$$

$$\begin{aligned} \mathcal{L}_{Re\_disc} = & \sum_{m=1}^{M} E(d^m(\mathbf{N}, T(\mathbf{x_f}))) - \\ & \sum_{m=1}^{M} E(d^m(\mathbf{N}, T(\mathbf{x_r}))) - \sum_{m=1}^{M} E(d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))) \end{aligned} \tag{5.9}$$

Here, $\mathcal{L}_P$ represents the perceptual loss function shown in equation (5.6). The weight combining the perceptual loss ($\mathcal{L}_P$) and the generator adversarial loss was set to 0.005 based on several previous works [29, 1, 76] on GAN-based image restoration and enhancement.

Gradient Analysis for ReSphereGAN: The gradients generated from the losses during the second training stage are crucial for stabilising the GAN models. It is important to avoid vanishing gradients and explosion problems as pointed out in [81, 43]. The gradients of $d^m(\mathbf{N}, T(\mathbf{x_r}))$ and $d^m(\mathbf{N}, T(\mathbf{x_f}))$ have already been evaluated in [237, 238], so we just analyse the gradients of the new relativistic geodesic distance $d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))$.

Lemma 1. $E(\left\|\nabla_{(\mathbf{x_r},\mathbf{x_f})} d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))\right\|_2) < \infty$ for all m (∇ represents the derivation operation, and $\|\cdot\|_2$ is the Euclidean norm).

The proof of **Lemma 1** is provided below. This indicates that the proposed ReSphereGAN will ensure stable GAN learning with any moment m.

Proof. Based on the definition of geodesic distance in [237], the relativistic geodesic distance $d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))$ can be written as:

$$\begin{aligned} & d^m(T(\mathbf{x_r}), T(\mathbf{x_f})) \\ = \quad & \arccos^m\left(\frac{\|\mathbf{x_r}\|_2^2 \|\mathbf{x_f}\|_2^2 - \|\mathbf{x_r}\|_2^2 - \|\mathbf{x_f}\|_2^2 + 4\mathbf{x_r}\cdot\mathbf{x_f} + 1}{(\|\mathbf{x_r}\|_2^2 + 1)(\|\mathbf{x_f}\|_2^2 + 1)}\right) \\ \equiv \quad & \arccos^m(A) \end{aligned} \tag{5.10}$$

Here, $A \equiv \dfrac{\|\mathbf{x_r}\|_2^2 \|\mathbf{x_f}\|_2^2 - \|\mathbf{x_r}\|_2^2 - \|\mathbf{x_f}\|_2^2 + 4\mathbf{x_r}\cdot\mathbf{x_f} + 1}{(\|\mathbf{x_r}\|_2^2 + 1)(\|\mathbf{x_f}\|_2^2 + 1)}$, $A \in [-1, 1]$. $\mathbf{x_r}$ and $\mathbf{x_f}$ are the real and fake feature points respectively in n-dimensional Euclidean feature space.

According to the chain rule,

$$\frac{\partial d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))}{\partial \mathbf{x_r}} = \arccos^{m-1}(A) \cdot \frac{-m}{\sqrt{1-A^2}} \cdot \frac{\partial A}{\partial \mathbf{x_r}} \tag{5.11}$$

Based on the equation (5.11), the gradient of $d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))$ can be further obtained following the chain rule and the product rule of derivative:

$$\begin{aligned} & \nabla_{(\mathbf{x_r},\mathbf{x_f})} d^m(T(\mathbf{x_r}), T(\mathbf{x_f})) \\ = \quad & \frac{\partial^2 d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))}{\partial \mathbf{x_r} \partial \mathbf{x_f}} \\ = \quad & m(m-1) \cdot \arccos^{m-2}(A) \cdot \frac{1}{1-A^2} \cdot \frac{\partial A}{\partial \mathbf{x_f}} \cdot \frac{\partial A}{\partial \mathbf{x_r}} - \\ & m \cdot \arccos^{m-1}(A) \cdot \frac{A}{(1-A^2)^{\frac{3}{2}}} \cdot \frac{\partial A}{\partial \mathbf{x_f}} \cdot \frac{\partial A}{\partial \mathbf{x_r}} - \\ & m \cdot \arccos^{m-1}(A) \cdot \frac{1}{\sqrt{1-A^2}} \cdot \frac{\partial^2 A}{\partial \mathbf{x_r} \partial \mathbf{x_f}} \end{aligned} \tag{5.12}$$

Based on Lemmas 1 and 2 in [237, 238], we have

$$\begin{cases} \arccos(A) < \infty \\ \dfrac{\partial A}{\partial \mathbf{x_f}} \dfrac{\partial A}{\partial \mathbf{x_r}} < \infty \\ \dfrac{\partial^2 A}{\partial \mathbf{x_r} \partial \mathbf{x_f}} < \infty \end{cases} \tag{5.13}$$

According to Propositions 1 and 2 in [237] and Theorem 6.9 in [239], the geometric distance between the real and fake data feature points, $\mathbf{x_r}$ and $\mathbf{x_f}$, weakly converges to 0, for all moment values m:

$$d^m(T(\mathbf{x_r}), T(\mathbf{x_f})) = \arccos^m(A) \rightharpoonup 0 \tag{5.14}$$

Here, $\rightharpoonup$ represents *weak convergence*. This indicates that $\arccos^m(A) \neq 0$, and thus $A \neq \pm 1$ [230]. Therefore we can have:

$$1 - A^2 \neq 0 \tag{5.15}$$

According to equation (5.13) and (5.15), the gradient (and its mean) of the relativistic geodesic distance is bounded for all moment values m:

$$\nabla_{(\mathbf{x_r}, \mathbf{x_f})} d^m(T(\mathbf{x_r}), T(\mathbf{x_f})) < \infty, \tag{5.16}$$

$$E(\left\| \nabla_{(\mathbf{x_r}, \mathbf{x_f})} d^m(T(\mathbf{x_r}), T(\mathbf{x_f})) \right\|_2) < \infty \tag{5.17}$$

□

In practice, it is noted that although ReSphereGAN may generate relatively large gradients as the original SphereGAN, they can still be calculated during the training process when the Adam optimiser is used [47]. This has also been reported in [237, 238].

5.2.3 Training and Evaluation Configurations

In this chapter, the similar training and evaluation configurations (including training hyper-parameters setting, QP sub-grouping, etc.) as introduced in the Section 3.2.4 are employed in this chapter for all tested GAN architectures. It is noted that only the generators (MSRResNet, BDNet and CVENet) are employed in evaluation stage.

5.3 Experimental Configuration

5.3.1 Experiment Settings

In this chapter, we utilised similar experiment settings to those described in Section 3.2.5 for testing all presented GAN architectures (MSRGAN, BDGAN and CVEGAN). It is noted that the CVEGAN is tested on PP, SRA and EBDA coding modules based on both HEVC HM 16.20 and VVC VTM 7.0.

To further evaluate network generalisation, another two commonly used test databases alongside JVET-CTC SDR (see Section 2.3.1) have also been employed to test CVEGAN and the other three top performers, including UVG (6 test sequences) [146] and AOM main test dataset (21 test sequences) *objective-1-fast* (o-1-f)[3] [15]. None of the sequences in these three test datasets was included in the training database, BVI-DVC. It should be noted that only UHD (2160p) content from these databases was used to evaluate the SRA coding tool since, as previously reported [57], lower resolutions provide only limited and inconsistent coding gains. The JVET-CTC SDR standard test dataset is only used to evaluate the coding performance of MSRGAN and BDGAN.

In order to demonstrate the performance of the generative adversarial network and perceptually-inspired loss functions, the performance of MSRGAN, BDGAN and CVEGAN was also compared to that of MSRResNet (trained by $\ell 1$ loss), BDNet (trained by $\ell 1$ loss) and CVENet (trained by perceptual loss $\mathcal{L}_P$ shown in equation (5.6)).

5.3.2 Benchmarked CNN and GAN Architectures: CVEGAN Comparisons

Twenty-four popular and state-of-the-art CNN and GAN architectures (as introduced in Section 2.2.2 and Section 2.2.3), which have been widely used in image super-resolution, restoration and video compression, have been utilised for benchmarking CVEGAN in this chapter. These include: SRCNN [61], FSRCNN [62], VDSR [63], SRResNet [28], DRRN [64], EDSR [65], RDN [66], ESRResNet [29], RCAN [68], MSRResNet [1], CARN [69], UDSR [70], HR-EnhanceNet [71], RNAN [73], ADGAN [82], SRResCGAN [83], SRGAN [28], PCARNGAN [85], RCAGAN [77], MSRGAN [1], ESRGAN [29], RCAN-GAN [88], PatchESRGAN, [89], and RFB-ESRGAN [76]. All of these networks have been re-implemented and trained using the same framework (TensorFlow 1.8.0) with identical training material following the same training methodology and loss functions as described in

[3] A few source sequences have been excluded in UVG and o-1-f datasets, which have been employed in BVI-DVC as training data.

their original literature. During re-implementation, the input and output interfaces of these networks have been modified to satisfy the data format requirements.

All networks under test have been integrated into both PP and SRA coding tools and tested under the experiment settings as described in Section 5.3.1. The original HEVC HM 16.20 was used as the host codec and also as the benchmark anchor. The relative computational complexities of all test networks (the generator for the case of GANs) have also been calculated and benchmarked against the simplest SRCNN [61]. The training and evaluation processes were also executed on the same devices as described in Section 3.2.5.

5.3.3 Ablation Study: CVEGAN Comparisons

Five primary contributions in CVEGAN have been tested and compared to the state of the art for PP and SRA. All ablation studies are based on the HEVC HM 16.20 and tested on the JVET-CTC SDR dataset.

(1) **Mul^{2}Res Block** effectiveness has been evaluated by replacing it with other commonly used convolutional blocks for CNN-based image restoration, which include residual block (RB) [28], modified residual block (MRB) [1], residual dense block (RDB) [66], residual-in-residual dense block (RRDB) [29], residual channel attention block (RCAB) [68], Xception block [79] (*cardinality* is 4), ResNeXt block [75] (*cardinality* is 4) and ResNeSt block [80] (*cardinality* is 4).

(2) **ERNB** is substituted by the original non-local block [73] to evaluate its effectiveness.

(3) **ECBAM** is replaced by the original CBAM [59] for comparison.

(4) **ReSphereGAN** training has been compared with other commonly used GAN training approaches, including standard GAN [28], Relativistic average GAN (RaGAN) [29], Patch-GAN [89], conditional GAN (cGAN) [86], Wasserstein GAN-gradient penalty (WGAN-GP) [179] and the original SphereGAN [237].

(5) **Perceptual Loss Function** presented in this chapter was compared with other commonly used loss functions for GAN training (L_7-L_{11}), as described in the previous 'Perceptually-inspired Loss Function' section.

5.3.4 Subjective Test Configuration: CVEGAN Comparisons

A lab-based subjective test has also been conducted using a double stimulus methodology on a selection of network architectures (alongside the anchor HM and CVEGAN) for both PP and SRA. We collected subjective scores from twenty-eight subjects [4] using the reconstructed

[4]Due to the impact of the COVID-19 pandemic, a large lab-based subjective test employing non-expert participants could not be conducted.

videos of 12 UHD source sequences (QP 37 only). Further details of the testing configuration are described below.

Reference and Test Content

Twelve UHD (2160p) source sequences from the JVET-CTC SDR [39] and UVG [146] datasets are selected as source content in this subjective evaluation. They have been encoded by the original HEVC HM 16.20 and its two enhanced versions (QP 37 only) with CNN-based PP and SRA coding tools. Each tool further generated results using four different networks to perform CNN operations, including RNAN [73], RFB-ESRGAN [76], MSRGAN [1], and the proposed CVEGAN. The former three architectures were selected due to their relatively higher coding gains (see Tables 5.9 and 5.10) compared to other benchmark networks assessed by VMAF. This results in 9 different test versions for each source sequence.

Subjective Experiment Configurations

The subjective tests were conducted in a laboratory with a darkened, living room style environment. The background luminance level was set to 15% of the peak luminance of the monitor used [191]. All the video sequences were shown at their native framerates, on a SONY PVM-X550 4K OLED professional video monitor with a maximum viewing angle of 89° and an effective picture size (H×V) of 1209.6×680.4 mm. We connected this reference monitor (the spatial resolution was configured to 3840×2160) to a Windows PC running the MATLAB R2019b and Psychotoolbox 3.0. The viewing distance was set to be 1.6 times of the monitor height (718.4 mm) based on the ITU-R BT.500 [191].

In this experiments, the double stimulus continuous quality scale (DSCQS) methodology [191] was used. In each trial, participants were shown Sequence A and Sequence B twice. One of these is a source sequence and the other is one of its nine distorted versions. Their orders are randomly determined and unknown to the subjects. After viewing these two sequences, the participants were asked to rate the perceived quality of both videos, based on a continuous quality scale from 1 to 5 (1-Bad, 2-Poor, 3-Fair, 4- Good and 5-Excellent).

Twenty-eight subjects (16 male and 12 female)[5] participated in this experiment and their average age was 31.6 years. They all had normal or corrected-to-normal colour vision verified by using Snellen and Ishihara charts [191].

[5]Due to the impact of the COVID-19 pandemic, a limited number of participants were employed in this experiment.

Data Processing

Difference scores were calculated for each trial and each participant by subtracting the quality score of the distorted sequence from its corresponding reference source. Possible outliers were removed following the procedures described in [191]. Difference mean opinion scores (DMOS) were obtained for every trial by taking the mean of the difference scores. The average DMOS among all source sequences for each test version (HM 16.20, PP-RNAN, PP-RFB-ESRGAN, PP-MSRGAN, PP-CVEGAN, SRA-RNAN, SRA-RFB-ESRGAN, SRA-MSRGAN and SRA-CVEGAN) was then calculated, as shown in Table 5.15.

5.4 Results and Discussion

5.4.1 MSRGAN

Compression Performance-SRA

Table 5.2 summarises the BD-rate results on the JVET-CTC UHD tested sequences, where the proposed spatial resolution adaptation framework (using MSRResNet-$\ell 1$ or MSRGAN for up-sampling) is compared to the original HEVC HM 16.20. It can be observed that, based on VMAF, both MSRResNet-$\ell 1$ and MSRGAN offer significant coding gains against the original HM, with average BD-rate gains of -25.9% and -35.6% respectively. MSRGAN also achieves an additional 9.7% savings over MSRResNet-$\ell 1$ due to exploitation of the generative adversarial network and the perceptual loss function.

The improvement can further be demonstrated by comparing the subjective quality of reconstructed frames. Figure. 5.12 provides a perceptual comparison between the original HM 16.20 and the proposed methods using MSRResNet-$\ell 1$ and MSRGAN. It is noted that both MSRResNet-$\ell 1$ and MSRGAN reconstructions exhibit fewer visual artefacts than those of HM 16.20 at similar or even lower bit rates. In addition, MSRGAN results exhibit slightly more texture detail compared to MSRResNet-$\ell 1$.

Compression Performance-PP

The MSRGAN was further utilised to enhance VVC (VTM 4.0.1) and AV1 (AV1 libaom 1.0.0 – version 1.0.0-5ec3e8c) after being integrated into the post-processing (PP) coding module. The same coding configurations described in Section 3.2.3 and [5] have been employed for VVC VTM 4.0.1 and AV1 libaom 1.0.0 respectively. It is noted that, for VVC VTM 4.0.1, the compression performance is evaluated for both low (L-QPs: 22-37) and high QP (H-QPs: 27-42) ranges (as shown in Table 5.3). The network has been trained using the BVI-DVC

Table 5.2 Compression results of the MSRResNet and MSRGAN-based SRA for HM 16.20 based on the JVET-CTC UHD tested sequences.

Class-Sequence	MSRResNet-$\ell 1$		MSRGAN	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
A1-Campfire	-26.0%	-42.0%	-21.4%	**-46.2%**
A1-FoodMarket4	-13.6%	-22.0%	-11.2%	**-25.9%**
A1-Tango2	-17.0%	-23.0%	-13.8%	**-27.6%**
A2-CatRobot1	-5.3%	-22.5%	-0.2%	**-33.5%**
A2-DaylightRoad2	+9.5%	-20.2%	+15.5%	**-36.4%**
A2-ParkRunning3	-25.9%	-34.7%	-23.2%	**-44.0%**
Overall	-13.1%	-25.9%	-9.1%	**-35.6%**

database through the same training methodologies and experimental configurations which have been introduced in Sections 5.2.1 and 5.3.1.

Tables 5.3 and 5.4 summarise the compression performance of the proposed method when it is applied to VVC and AV1 compressed content. For $\ell 1$ trained CNNs, we note that the average bit-rate savings according to PSNR are 3.9% and 5.8% against the original VVC and AV1 respectively. If the perceptual quality metric, VMAF, is used to assess video quality, the coding gains are 4.2% over VVC and 2.7% over AV1. When we use perceptual loss function trained models for post-processing, the coding gains appear much more significant based on the assessment of VMAF – 13.9% and 10.5% over VVC and AV1 respectively. This is particularly evident for test sequences, such as *ParkRunning3*, *BQTerrace* and *BQSquare*, which present a large number of sharp edges.

5.4.2 BDGAN

Compression Performance-EBDA

Table 5.5 summarises the compression performance of the proposed method, with HM 16.20 used as the benchmark. In order to highlight the improvement obtained by using perceptual loss functions, the coding gains from an $\ell 1$ trained (using the same training material) BDNet for EBD up-sampling (BDNet-$\ell 1$) are also reported for comparison.

It is noted that EBD adaptation with BDGAN achieves consistent coding gains for all test sequences, with an average BD-rate (assessed by VMAF) of 24.8%, which is 7.4% higher than that for BDNet-$\ell 1$. When PSNR is employed for video quality assessment, the savings are much lower and not as significant as those for BDNet-$\ell 1$. Comparing to [213], where

Table 5.3 The compression performance of the MSRResNet and MSRGAN-based PP methods benchmarked on the original VVC VTM 4.0.1. Negative BD-rate values indicate coding gains.

Method	MSRResNet-$\ell 1$				MSRGAN			
Metric	PSNR		VMAF		PSNR		VMAF	
QP Range	H-QPs	L-QPs	H-QPs	L-QPs	H-QPs	L-QPs	H-QPs	L-QPs
Class-Sequence	BD-rate	BD-rate	BD-rate	BD-rate	BD-rate	BD-rate	BD-rate	BD-rate
A1-Campfire	-3.3%	-2.3%	-5.6%	-4.6%	+0.2%	-0.4%	-10.4%	-10.7%
A1-FoodMarket4	-2.6%	-2.0%	-3.8%	-3.0%	-0.0%	+0.1%	-8.4%	-7.4%
A1-Tango2	-3.3%	-2.9%	-3.4%	-3.0%	-1.1%	-0.6%	-7.8%	-9.3%
A2-CatRobot1	-5.2%	-5.2%	-4.6%	-4.4%	-0.6%	-1.1%	-14.4%	-17.8%
A2-DaylightRoad2	-6.0%	-7.1%	-6.8%	-7.2%	-1.1%	-2.1%	-19.5%	-23.4%
A2-ParkRunning3	-0.8%	-0.4%	-2.3%	-0.2%	+2.1%	+2.4%	-11.7%	-11.1%
Class A	-3.5%	-3.3%	-4.4%	-3.7%	-0.1%	+0.3%	-10.1%	-13.3%
B-BQTerrace	-2.2%	-1.0%	-6.1%	-1.1%	+2.0%	+0.3%	-23.3%	-28.8%
B-BasketballDrive	-3.4%	-3.1%	-1.8%	+2.7%	-0.9%	-0.9%	-7.9%	-8.7%
B-Cactus	-3.4%	-3.0%	-5.1%	-4.4%	+0.2%	-0.2%	-15.8%	-17.1%
B-MarketPlace	-2.6%	-2.3%	-4.8%	-4.0%	+1.2%	+0.3%	-17.7%	-18.2%
B-RitualDance	-3.8%	-3.5%	-4.6%	-2.6%	-1.1%	-1.2%	-11.7%	-11.2%
Class B	-3.1%	-2.6%	-4.5%	-1.9%	+0.3%	-0.3%	-15.3%	-16.8%
C-BQMall	-5.6%	-5.6%	-4.7%	-6.8%	-2.1%	-2.5%	-14.3%	-13.6%
C-BasketballDrill	-3.9%	-3.6%	-3.8%	-2.8%	-1.4%	-1.6%	-12.3%	-11.7%
C-PartyScene	-4.1%	-4.3%	-5.9%	-4.1%	-0.4%	-1.4%	-16.1%	-13.8%
C-RaceHorses	-3.1%	-2.1%	-3.4%	+1.2%	+0.2%	-0.4%	-11.9%	-10.4%
Class C	-4.2%	-3.9%	-4.5%	-3.1%	-0.9%	-1.5%	-13.7%	-12.4%
D-BQSquare	-8.7%	-9.6%	-10.1%	-11.6%	-4.0%	-4.5%	-16.6%	-19.5%
D-BasketballPass	-6.1%	-5.6%	-5.4%	-4.0%	-3.0%	-2.8%	-9.8%	-8.1%
D-BlowingBubbles	-3.7%	-3.8%	-4.8%	-3.8%	-0.5%	-1.3%	-16.1%	-14.5%
D-RaceHorses	-4.8%	-4.2%	-5.2%	-1.0%	-1.6%	-1.9%	-11.8%	-10.5%
Class D	-5.8%	-5.8%	-6.4%	-5.1%	-2.3%	-2.6%	-13.6%	-13.2%
Overall	-4.0%	-3.8%	-4.9%	-3.4%	-0.6%	-1.2%	-13.5%	-14.2%
	BD-rate=-3.9%		BD-rate=-4.2%		BD-rate=-0.9%		BD-rate=-13.9%	

Table 5.4 The compression performance of the the MSRResNet and MSRGAN-based PP methods benchmarked on the original AV1 1.0.0. Negative BD-rate values indicate coding gains.

Method	MSRResNet-$\ell 1$		MSRGAN	
Metric	PSNR	VMAF	PSNR	VMAF
Class-Sequence	BD-rate	BD-rate	BD-rate	BD-rate
A1-Campfire	-5.0%	-7.3%	-1.1%	-11.8%
A1-FoodMarket4	-4.0%	-8.9%	-1.1%	-9.6%
A1-Tango2	-4.6%	-8.2%	-1.9%	-10.7%
A2-CatRobot1	-5.9%	-5.9%	-1.9%	-15.0%
A2-DaylightRoad2	-7.7%	-5.0%	-3.1%	-17.2%
A2-ParkRunning3	-1.9%	-2.2%	+0.4%	-11.3%
Class A	-4.9%	-6.3%	-1.5%	-12.6%
B-BQTerrace	-4.5%	+1.4%	-1.3%	-10.2%
B-BasketballDrive	-6.0%	-3.0%	-2.9%	-7.1%
B-Cactus	-3.8%	-3.3%	-0.7%	-11.9%
B-MarketPlace	-3.0%	-4.5%	-0.5%	-17.4%
B-RitualDance	-4.7%	-5.0%	-2.3%	-11.1%
Class B	-4.4%	-2.9%	-1.5%	-11.5%
C-BQMall	-6.3%	-2.4%	-2.9%	-9.4%
C-BasketballDrill	-6.8%	-2.4%	-3.1%	-9.9%
C-PartyScene	-7.8%	+2.3%	-3.3%	-9.1%
C-RaceHorses	-4.1%	-3.8%	-1.8%	-8.4%
Class C	-6.3%	-1.6%	-2.8%	-9.2%
D-BQSquare	-16.1%	+11.2%	-8.4%	-4.5%
D-BasketballPass	-7.0%	-3.8%	-3.9%	-8.2%
D-BlowingBubbles	-6.2%	+2.0%	-2.8%	-9.7%
D-RaceHorses	-5.6%	-3.0%	-3.0%	-7.7%
Class D	-8.7%	+1.6%	-4.5%	-7.5%
Overall	-5.8%	-2.7%	-2.4%	-10.5%

(a) Campfire (HM 16.20, QP=37) (b) Campfire (MSRResNet-ℓ1, QP=37) (c) Campfire (MSRGAN, QP=37)

(d) Tango2 (HM 16.20, QP=37) (e) Tango2 (MSRResNet-ℓ1, QP=37) (f) Tango2 (MSRGAN, QP=37)

(g) DaylightRoad2 (HM 16.20, QP=37) (h) DaylightRoad2 (MSRResNet-ℓ1, QP=37) (i) DaylightRoad2 (MSRGAN, QP=37)

Figure. 5.12 Perceptual comparisons between the HM 16.20 and the proposed approach using MSRResNet-ℓ1 and MSRGAN (patches extracted from the 26th, 17th and the 270th frames of *Campfire*, *Tango2* and *DaylightRoad2* reconstructed sequences respectively and amplified by 4 times).

only PSNR-based results were presented, additional 3.9% coding gains have been achieved by using BDNet-ℓ1. This is due to the more advanced CNN model employed and the large video database used for training.

The example blocks from the reconstructed frames generated by the anchor HM 16.20, and EBD up-sampling with BDNet-ℓ1 and BDGAN are shown in Figure. 5.13. It can be observed that for both BDGAN and BDNet-ℓ1, the reconstructed content exhibits improved

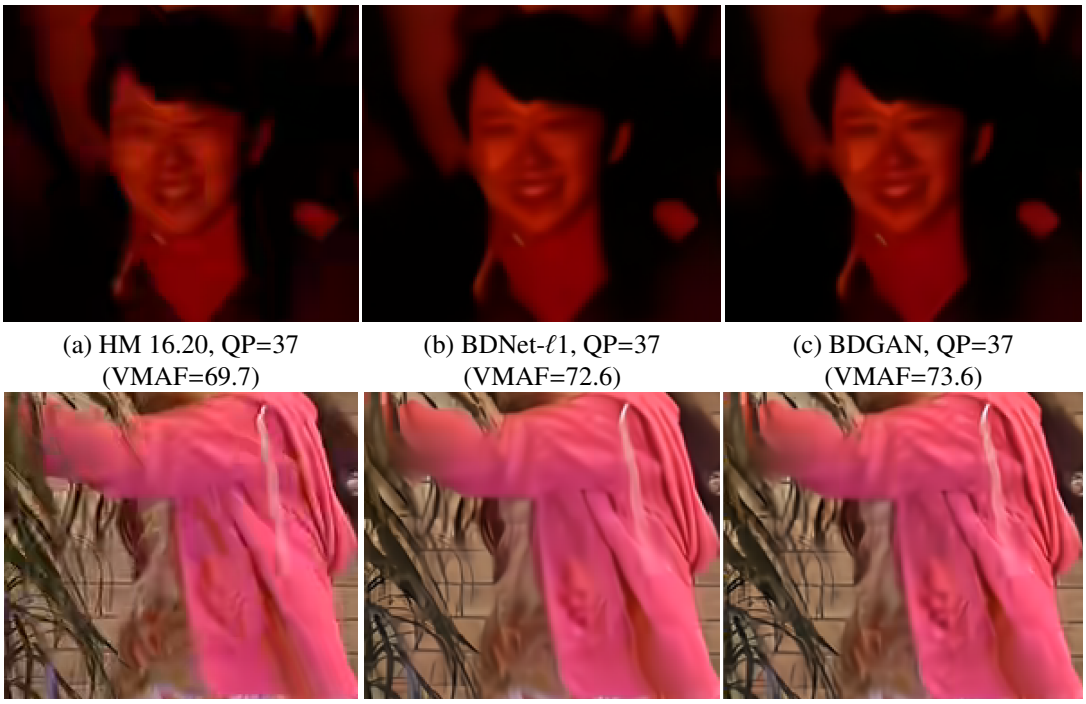
Table 5.5 Compression results of the BDNet and BDGAN-based EBDA for HM 16.20 based on the JVET-CTC tested sequences.

Sequence	BDNet-$\ell 1$		BDGAN	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
A1-Campfire	-16.7%	-25.3%	-14.8%	-29.7%
A1-FoodMarket4	-7.0%	-13.5%	-3.2%	-14.7%
A1-Tango2	-9.2%	-16.9%	-5.3%	-19.2%
A2-CatRobot1	-12.0%	-22.2%	-7.4%	-29.6%
A2-DaylightRoad2	-14.4%	-25.0%	-7.8%	-35.7%
A2-ParkRunning3	-14.2%	-19.9%	-11.9%	-28.0%
Class A (2160p)	**-12.3%**	-20.5%	-8.4%	**-26.2%**
B-BasketballDrive	-10.7%	-14.4%	-6.3%	-20.6%
B-BQTerrace	-10.2%	-28.8%	-5.3%	-41.1%
B-Cactus	-10.2%	-18.7%	-5.9%	-30.0%
B-MarketPlace	-4.6%	-13.7%	-1.4%	-25.3%
B-RitualDance	-7.1%	-13.3%	-3.5%	-19.4%
Class B (1080p)	**-8.6%**	-11.0%	-4.5%	**-27.3%**
C-BasketballDrill	-9.2%	-14.6%	-5.3%	-21.7%
C-BQMall	-9.0%	-13.7%	-4.9%	-22.3%
C-PartyScene	-7.7%	-12.5%	-2.9%	-22.9%
C-RaceHorses	-8.7%	-14.1%	-5.0%	-22.0%
Class C (480p)	**-8.7%**	-13.7%	-4.5%	**-22.2%**
D-BasketballPass	-10.3%	-12.7%	-7.0%	-18.7%
D-BlowingBubbles	-7.5%	-12.6%	-3.7%	-22.3%
D-BQSquare	-17.0%	-23.9%	-9.8%	-26.1%
D-RaceHorses	-9.7%	-14.4%	-6.3%	-22.0%
Class D (240p)	**-11.1%**	-15.9%	-6.7%	**-22.3%**
Overall	**-10.3%**	-17.4%	-6.2%	**-24.8%**

perceptual quality, with fewer blocking artefacts compared to the anchor. BDGAN results also exhibit more texture detail and higher contrast than those for BDNet-$\ell 1$.

5.4.3 CVEGAN

This section presents an analysis of the rate quality performance of CVEGAN (based on PP, SRA and EBDA) and a comprehensive comparison with twenty-four state-of-the-art



(a) HM 16.20, QP=37 (VMAF=69.7)
(b) BDNet-ℓ1, QP=37 (VMAF=72.6)
(c) BDGAN, QP=37 (VMAF=73.6)
(d) HM 16.20, QP=37 (VMAF=71.4)
(e) BDNet-ℓ1, QP=37 (VMAF=74.4)
(f) BDGAN, QP=37 (VMAF=76.9)

Figure. 5.13 Example blocks of the reconstructed frames for the anchor HM 16.20, EBD up-sampling with BDNet-ℓ1 and BDGAN (their bitstreams have similar bit rates). These are from the 175th and 162nd frames of *Campfire* and *PartyScene* sequences respectively and amplified by 4 times.

network architectures (based on PP and SRA) in the context of video compression enhancement. Perceptual comparisons are also given to aid further evaluation of its effectiveness. Results demonstrate that CVEGAN provides superior coding performance compared to its counterparts based on both objective quality assessments and subjective comparisons.

Compression Results for PP Coding Module

Table 5.6 summarises the compression performance of the PP coding module (with the CVEGAN) when it is integrated into HEVC HM 16.20 and VVC VTM 7.0. It can be observed that our approach achieves significant and consistent coding gains on all test sequences when integrated into HEVC, with the average BD-rates of -10.2% and -23.4% based on PSNR and VMAF respectively. The coding gains are lower for VTM, but are still evident with the average BD-rates of -2.5% and -8.0% respectively based on the assessment of PSNR and VMAF respectively. It can also be noted that, for both host codecs, the bitrate savings according to VMAF are generally higher than those for PSNR.

Table 5.6 Compression results of the CVEGAN-based PP for HM 16.20 and VTM 7.0.

Class-Sequence	CNN-based PP (HM 16.20)		CNN-based PP (VTM 7.0)	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
A1-Campfire	-11.3%	-31.1%	-2.1%	-10.8%
A1-FoodMarket4	-8.4%	-18.9%	-1.5%	-9.9%
A1-Tango2	-12.1%	-22.1%	-2.0%	-12.2%
A2-CatRobot1	-13.5%	-31.0%	-2.9%	-15.2%
A2-DaylightRoad2	-16.6%	-34.0%	-3.6%	-16.2%
A2-ParkRunning3	-7.4%	-27.9%	-0.8%	-8.0%
Class A (2160p)	-11.6%	-27.5%	-2.2%	-12.1%
B-BasketballDrive	-10.4%	-20.5%	-2.2%	-7.3%
B-BQTerrace	-14.5%	-41.1%	+0.1%	-9.1%
B-Cactus	-11.2%	-26.8%	-1.7%	-11.2%
B-MarketPlace	-8.1%	-21.7%	-2.3%	-11.6%
B-RitualDance	-7.9%	-19.4%	-2.9%	-9.3%
Class B (1080p)	-10.4%	-25.9%	-1.8%	-9.7%
C-BasketballDrill	-8.9%	-18.6%	-2.8%	-5.3%
C-BQMall	-9.0%	-19.0%	-3.2%	-4.4%
C-PartyScene	-8.7%	-20.7%	-2.6%	-6.4%
C-RaceHorses	-6.5%	-18.8%	-2.1%	-4.2%
Class C (480p)	-8.3%	-19.3%	-2.7%	-5.1%
D-BasketballPass	-7.1%	-13.8%	-3.2%	-2.9%
D-BlowingBubbles	-7.2%	-15.6%	-2.2%	-3.7%
D-BQSquare	-17.1%	-27.6%	-4.8%	-0.1%
D-RaceHorses	-7.3%	-16.1%	-3.4%	-3.4%
Class D (240p)	-9.7%	-18.3%	-3.4%	-2.5%
Overall	-10.2%	-23.4%	-2.5%	-8.0%

Compression Results for SRA and EBDA Coding Modules

Tables 5.7 and 5.8 summarise the compression performance of the CVEGAN-based SRA and EBDA coding tools benchmarked on the HEVC HM 16.20 and VVC VTM 7.0. CVEGAN has achieved evident coding gains on JVET-CTC test sequences when integrated into the HEVC HM and VVC VTM based on the assessment of VMAF, with average BD-rates of 38.4% (SRA) and 26.3% (EBDA) for HM 16.20, and 20.3% (SRA) and 7.8% (EBDA) for VTM 7.0. It can be also observed that the coding gains achieved based on the VMAF are

Table 5.7 Compression results of the CVEGAN-based EBDA and SRA for HM 16.20.

Class-Sequence	CNN-based EBDA		CNN-based SRA	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
A1-Campfire	-16.1%	-34.1%	-27.9%	-49.9%
A1-FoodMarket4	-5.3%	-21.8%	-15.3%	-30.6%
A1-Tango2	-6.3%	-25.0%	-19.3%	-31.4%
A2-CatRobot1	-10.0%	-33.9%	-7.0%	-36.9%
A2-DaylightRoad2	-10.6%	-36.9%	+7.3%	-37.8%
A2-ParkRunning3	-13.5%	-30.8%	-26.8%	-43.9%
Class A (2160p)	-10.3%	-30.4%	-14.8%	-38.4%
B-BasketballDrive	-8.6%	-23.4%	–	–
B-BQTerrace	-8.2%	-44.0%	–	–
B-Cactus	-8.6%	-29.7%	–	–
B-MarketPlace	-4.0%	-24.7%	–	–
B-RitualDance	-5.6%	-22.3%	–	–
Class B (1080p)	-7.0%	-28.8%	–	–
C-BasketballDrill	-5.6%	-21.5%	–	–
C-BQMall	-5.6%	-21.9%	–	–
C-PartyScene	-4.8%	-23.6%	–	–
C-RaceHorses	-7.5%	-21.7%	–	–
Class C (480p)	-5.9%	-22.2%	–	–
D-BasketballPass	-6.2%	-16.7%	–	–
D-BlowingBubbles	-2.7%	-18.5%	–	–
D-BQSquare	-8.4%	-30.5%	–	–
D-RaceHorses	-7.3%	-19.0%	–	–
Class D (240p)	-6.2%	-21.2%	–	–
Overall	-7.6%	-26.3%	-14.8%	-38.4%

much higher than those for PSNR for both SRA and EBDA coding modules based on the two standard codecs. This is mainly due to the reason that the CVEGAN is trained using perceptual loss function instead of the pixel-wise loss function ($\ell 1$ or $\ell 2$ loss).

Comparisons with CNN and GAN Architectures

Tables 5.9-5.10 summarise the compression performance generated by CVEGAN and the 24 CNN/GAN networks when they are integrated into post-processing (PP) and spatial

Table 5.8 Compression results of the CVEGAN-based EBDA and SRA for VTM 7.0.

Class-Sequence	CNN-based EBDA		CNN-based SRA	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
A1-Campfire	-3.1%	-22.6%	-9.4%	-37.9%
A1-FoodMarket4	+1.3%	-9.3%	-1.6%	-16.0%
A1-Tango2	+1.5%	-9.3%	-5.2%	-14.9%
A2-CatRobot1	+0.1%	-13.4%	+14.0%	-12.0%
A2-DaylightRoad2	-0.1%	-17.7%	+37.2%	-16.7%
A2-ParkRunning3	+1.4%	-13.7%	-8.9%	-24.3%
Class A (2160p)	+0.2%	-14.3%	+4.4%	-20.3%
B-BasketballDrive	-0.9%	-8.8%	–	–
B-BQTerrace	+5.3%	-3.9%	–	–
B-Cactus	-0.1%	-13.0%	–	–
B-MarketPlace	+8.1%	-3.7%	–	–
B-RitualDance	+0.3%	-9.2%	–	–
Class B (1080p)	+2.5%	-7.7%	–	–
C-BasketballDrill	+2.6%	-5.1%	–	–
C-BQMall	-0.0%	-3.3%	–	–
C-PartyScene	+0.1%	-4.4%	–	–
C-RaceHorses	-0.7%	-2.7%	–	–
Class C (480p)	+0.5%	-3.9%	–	–
D-BasketballPass	-1.4%	-5.7%	–	–
D-BlowingBubbles	+0.4%	-1.8%	–	–
D-BQSquare	-0.8%	+0.6%	–	–
D-RaceHorses	-1.9%	-1.8%	–	–
Class D (240p)	-0.9%	-2.2%	–	–
Overall	+0.6%	-7.8%	+4.4%	-20.3%

resolution adaptation (SRA) coding tools in the context of HEVC. It can be observed that for both PP and SRA coding tools, although based on PSNR (the pixel-wise distortion-based quality metric), the proposed CVEGAN is not the best performer among all the tested networks, it outperforms all 24 architectures based on the perceptual quality metric VMAF. Considering that VMAF offers a much higher correlation with subjective scores compared to PSNR [189, 175, 182], the effectiveness of the proposed algorithm in term of video quality enhancement is evident. The additional coding gains in terms of BD-rate (based on VMAF) compared to other networks are greater than 1.8% and 2.6% for PP and SRA respectively.

Table 5.9 Comprehensive comparison results (in terms of BD-rate (%) based on both PSNR and VMAF) between the CVEGAN and 24 benchmark networks when they are integrated into the PP coding tool for HEVC compression. The result sets {i/j/k} in this table stands for the BD-rate values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test network is also provided for comparison.

Networks	CNN-based PP		
	BD-rate (PSNR)	BD-rate (VMAF)	Relative Complexity
SRCNN [61]	-1.9/–/–	-7.4/–/–	1.0×
FSRCNN [62]	-1.6/–/–	-7.3/–/–	1.37×
VDSR [63]	-1.9/–/–	-7.6/–/–	2.05×
DRRN [64]	-10.8/–/–	-14.9/–/–	2.70×
EDSR [65]	-10.0/–/–	-14.6/–/–	4.50×
SRResNet [28]	-9.8/–/–	-12.7/–/–	2.45×
MSRResNet [1]	-10.4/–/–	-14.2/–/–	2.46×
CARN [69]	-11.2/–/–	-15.4/–/–	2.23×
UDSR[70]	-11.4/–/–	-16.0/–/–	3.04×
HR-EnhanceNet [71]	-11.3/–/–	-16.4/–/–	2.80×
ESRResNet [29]	-11.8/–/–	-17.7/–/–	3.82×
RCAN [68]	-12.1/–/–	-18.5/–/–	4.82×
RDN [66]	-12.2/–/–	-17.0/–/–	3.46×
RNAN [73]	-12.5/-14.1/-11.7	-19.2/-23.9/-21.5	5.78×
ADGAN [82]	-1.3/–/–	-7.7/–/–	2.46×
SRResCGAN [83]	-7.1/–/–	-10.4/–/–	1.71×
SRGAN [28]	-7.4/–/–	-12.9/–/–	2.46×
PCARNGAN [85]	-8.3/–/–	-16.0/–/–	2.25×
RCAGAN [77]	-9.1/–/–	-16.8/–/–	3.31×
MSRGAN [1]	-6.5/-8.7/-5.9	-21.1/-25.7/-23.5	2.46×
ESRGAN [29]	-8.7/–/–	-17.9/–/–	3.82×
RCAN-GAN [88]	-9.3/–/–	-18.0/–/–	4.84×
PatchESRGAN [89]	-9.0/–/–	-18.1/–/–	3.82×
RFB-ESRGAN [76]	-9.1/-10.8/-7.9	-18.3/-23.2/-20.4	4.58×
CVENet	-9.5/-11.3/-8.5	-21.3/-26.0/-23.6	2.80×
CVEGAN	-10.2/-11.9/-9.0	**-23.4/-27.8/-25.3**	2.80×

We also evaluated the models obtained after the first training step (trained with our perceptual loss function $\mathcal{L}_P$), denoted as CVENet in Tables 5.9 and 5.10. Its overall performance is slightly lower than that of the final CVEGAN, with up to 2.1% and 2.2% BD-rate differences (based on VMAF) for PP and SRA respectively. This demonstrates the improvement due to the second training stage using the proposed ReSphereGAN.

Table 5.10 Comprehensive comparison results (in terms of BD-rate (%) based on both PSNR and VMAF) between the CVEGAN and 24 benchmark networks when they are integrated into the SRA coding tool for HEVC compression. The result sets {i/j/k} in this table stands for the BD-rate values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test network is also provided for comparison.

Networks	CNN-based SRA		
	BD-rate (PSNR)	BD-rate (VMAF)	Relative Complexity
SRCNN [61]	-3.1/–/–	-21.1/–/–	1.0×
FSRCNN [62]	-4.5/–/–	-20.9/–/–	1.28×
VDSR [63]	-6.6/–/–	-18.3/–/–	3.79×
DRRN [64]	-15.0/–/–	-33.2/–/–	5.01×
EDSR [65]	-13.4/–/–	-30.1/–/–	8.33×
SRResNet [28]	-13.2/–/–	-30.0/–/–	4.46×
MSRResNet [1]	-14.6/–/–	-32.7/–/–	4.52×
CARN [69]	-15.5/–/–	-33.5/–/–	4.15×
UDSR[70]	-15.7/–/–	-33.3/–/–	5.62×
HR-EnhanceNet [71]	-15.8/–/–	-33.1/–/–	5.56×
ESRResNet [29]	-16.1/–/–	-33.6/–/–	7.10×
RCAN [68]	-17.1/–/–	-35.1/–/–	8.98×
RDN [66]	-16.6/–/–	-34.5/–/–	6.41×
RNAN [73]	-17.4/-9.3/–	-34.8/-31.6/–	10.79×
ADGAN [82]	-5.1/–/–	-18.5/–/–	4.56×
SRResCGAN [83]	-10.3/–/–	-27.2/–/–	3.16×
SRGAN [28]	-10.9/–/–	-30.2/–/–	4.52×
PCARNGAN [85]	-12.3/–/–	-33.7/–/–	4.18×
RCAGAN [77]	-13.7/–/–	-33.9/–/–	6.13×
MSRGAN [1]	-9.1/-3.3/–	-35.6/-32.9/–	4.54×
ESRGAN [29]	-12.5/–/–	-33.8/–/–	7.15×
RCAN-GAN [88]	-13.9/–/–	-34.1/–/–	9.03×
PatchESRGAN [89]	-12.8/–/–	-34.2/–/–	7.20×
RFB-ESRGAN [76]	-12.9/-4.5/–	-34.3/-31.0/–	8.52×
CVENet	-14.2/-5.9/–	-36.4/-33.3/–	5.23×
CVEGAN	-14.8/-6.4/–	**-38.4/-35.5/–**	5.23×

Tables 5.9-5.10 also show the relative complexities of all test networks, which are benchmarked on that of SRCNN. It is noted that the relative complexity of CVEGAN is only 2.8 times of that for SRCNN, which is relatively low compared to many network architectures including EDSR, UDSR, ESRResNet, RCAN, RDN, RNAN, RCAGAN, ESRGAN, RCAN-GAN, PatchESRGAN and RFB-ESRGAN.

Table 5.11 Ablation study results for PP coding tool based on the HEVC HM 16.20. The result sets {i/j/k} in this table stands for the BD-rate (%) values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test variant is also provided for comparison.

CVEGAN Variants		CNN-based PP		
		BD-rate (PSNR)	BD-rate (VMAF)	Relative Complexity
w/o Mul^{2}Res block	w/ ResNeSt block [80]	-9.4/–/–	-22.3/–/–	2.81×
	w/ RCAB block [68]	-9.0/–/–	-22.0/–/–	3.07×
	w/ RRDB block [29]	-8.8/–/–	-21.6/–/–	3.16×
	w/ RDB block [66]	-7.1/–/–	-20.9/–/–	2.95×
	w/ ResNeXt block [75]	-6.7/–/–	-20.6/–/–	2.65×
	w/ Xception block [79]	-6.6/–/–	-20.1/–/–	2.65×
	w/ MRB block [1]	-6.3/–/–	-18.3/–/–	2.24×
	w/ RB block [28]	-6.1/–/–	-16.6/–/–	2.24×
w/o ERNB	w/ Non-local block [73]	-9.9/–/–	-22.1/–/–	3.22×
w/o ECBAM	w/ CBAM block [59]	-10.0/–/–	-22.3/–/–	2.78×
w/o ReSphereGAN	w/ SphereGAN [237]	-9.7/–/–	-22.1/–/–	2.80×
	w/ RaGAN [29]	-9.6/–/–	-21.8/–/–	2.80×
	w/ cGAN [86]	-9.6/–/–	-21.6/–/–	2.80×
	w/ PatchGAN [89]	-9.4/–/–	-21.6/–/–	2.79×
	w/ WGAN-GP [179]	-9.1/–/–	-21.7/–/–	2.80×
	w/ Standard GAN [28]	-8.2/–/–	-21.5/–/–	2.80×
w/o $\mathcal{L}_P$	w/ L_7 loss [83]	-4.5/–/–	-19.2/–/–	2.80×
	w/ L_8 loss [1]	-7.5/–/–	-21.9/–/–	2.80×
	w/ L_9 loss [29]	-5.7/–/–	-20.2/–/–	2.80×
	w/ L_{10} loss [77]	-6.4/–/–	-21.4/–/–	2.80×
	w/ L_{11} loss [28]	-5.8/–/–	-20.4/–/–	2.79×
CVENet		-9.5/–/–	-21.3/–/–	2.80×
CVEGAN		-10.2/–/–	**-23.4**/–/–	2.80×

Comparisons with MFRNet, MSRGAN, BDGAN and CVEGAN

The compression performance of the networks presented in Chapters 4 (MFRNet) and 5 (MSRGAN, BDGAN and CVEGAN) is further compared in this section. Tables 5.13 and 5.14 summarise the compression performance comparisons between the MFRNet, MSRGAN, BDGAN and CVEGAN for three typical coding enhancement modules (PP, SRA and EBDA) based on the HEVC HM 16.20 and VVC VTM 7.0. It can be observed that when PSNR is used as the quality assessment method, the MFRNet has provided highest BD-rate savings for all tested coding modules over three GAN architectures based on the HM 16.20 and

Table 5.12 Ablation study results for SRA coding tool based on the HEVC HM 16.20. The result sets {i/j/k} in this table stands for the BD-rate (%) values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test variant is also provided for comparison.

CVEGAN Variants		CNN-based SRA		
		BD-rate (PSNR)	BD-rate (VMAF)	Relative Complexity
w/o Mul^{2}Res block	w/ ResNeSt block [80]	-14.3/–/–	-37.1/–/–	5.25×
	w/ RCAB block [68]	-14.0/–/–	-36.6/–/–	5.72×
	w/ RRDB block [29]	-13.8/–/–	-36.3/–/–	5.92×
	w/ RDB block [66]	-12.3/–/–	-35.9/–/–	5.67×
	w/ ResNeXt block [75]	-12.0/–/–	-35.7/–/–	4.85×
	w/ Xception block [79]	-11.8/–/–	-35.2/–/–	4.92×
	w/ MRB block [1]	-9.4/–/–	-34.6/–/–	4.03×
	w/ RB block [28]	-8.6/–/–	-33.1/–/–	4.07×
w/o ERNB	w/ Non-local block [73]	-14.5/–/–	-37.2/–/–	6.20×
w/o ECBAM	w/ CBAM block [59]	-14.2/–/–	-37.0/–/–	5.22×
w/o ReSphereGAN	w/ SphereGAN [237]	-14.3/–/–	-37.1/–/–	5.23×
	w/ RaGAN [29]	-14.2/–/–	-36.8/–/–	5.26×
	w/ cGAN [86]	-14.1/–/–	-36.7/–/–	5.26×
	w/ PatchGAN [89]	-13.8/–/–	-36.8/–/–	5.31×
	w/ WGAN-GP [179]	-13.2/–/–	-36.6/–/–	5.28×
	w/ Standard GAN [28]	-12.6/–/–	-36.5/–/–	5.26×
w/o $\mathcal{L}_P$	w/ L_7 loss [83]	-9.6/–/–	-34.1/–/–	5.21×
	w/ L_8 loss [1]	-12.4/–/–	-37.0/–/–	5.23×
	w/ L_9 loss [29]	-10.7/–/–	-35.4/–/–	5.21×
	w/ L_{10} loss [77]	-11.5/–/–	-36.3/–/–	5.26×
	w/ L_{11} loss [28]	-10.7/–/–	-35.5/–/–	5.20×
CVENet		-14.2/–/–	-36.4/–/–	5.23×
CVEGAN		-14.8/–/–	**-38.4**/–/–	5.23×

VTM 7.0. This is mainly due to the fact that the MFRNet was trained using the pixel-wise loss function ($\ell 1$ loss). When perceptual and GAN adversarial training methodologies are employed (in MSRGAN, BDGAN and CVEGAN), higher coding gains can be achieved based on perceptual quality metrics, such as VMAF.

Ablation Study Results

As mentioned in Section 5.3.3, five primary contributions of CVEGAN have been compared with multiple alternative structures. The full results are presented in Tables 5.11 and 5.12

Table 5.13 Comparison between CNN and GAN architectures presented in Chapters 4 and 5 in the context of PP, SRA and EBDA for HM 16.20.

CNN/GAN Model	CNN-based PP		CNN-based SRA		CNN-based EBDA	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
MFRNet	**-14.1%**	-21.0%	**-17.5%**	-31.2%	**-13.2%**	-20.0%
MSRGAN	–	–	-9.1%	-35.6%	–	–
BDGAN	–	–	–	–	-6.2%	-24.8%
CVEGAN	-10.2%	**-23.4%**	-14.8%	**-38.4%**	-7.6%	**-26.3%**

Table 5.14 Comparison between CNN and GAN architectures presented in Chapters 4 and 5 in the context of PP, SRA and EBDA for VTM 7.0.

CNN/GAN Model	CNN-based PP		CNN-based SRA		CNN-based EBDA	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
MFRNet	**-6.7%**	-7.1%	+0.7%	-13.7%	**-4.0%**	-5.7%
CVEGAN	-2.5%	**-8.0%**	+4.4%	**-20.3%**	+0.6%	**-7.8%**

for PP and SRA coding modules, respectively. It can be observed that compared to each of the five primary features of CVEGAN, all its counterparts offer lower coding gains (based on VMAF) for both PP and SRA applications. This shows the effectiveness of these new structures.

Perceptual Comparisons

Figures. 5.14-5.21 present the subjective comparison results among the proposed CVEGAN, the anchor HEVC and other top performing architectures for both PP and SRA. It can be clearly observed that the perceptual quality of the CVEGAN output for both two coding modules has been effectively optimised with fewer blocking artefacts, more high frequency textural details and higher contrast compared to the anchor HEVC HM 16.20 as well as other popular network architectures. The perceptual quality improvements associated with CVEGAN provide further validation of our approach.

A lab-based subjective test has also been conducted using a double stimulus methodology to further compare the visual quality of the output of different network architectures (including the anchor HM 16.20 and CVEGAN). Based on Tables 5.9 and 5.10, the top three performers have been selected to be compared with the CVEGAN, these include RNAN [73], MSRGAN

[1] and RFB-ESRGAN [76]. The details of the subjective testing configurations have been described in Section 5.3.4.

Table 5.15 Subjective results based on 12 UHD source sequences from the JVET-CTC and UVG test datasets.

PP	Anchor (HM 16.20)	[73]	[1]	[76]	CVEGAN
Average DMOS	1.97	1.61	1.57	1.58	**1.53**
SRA	Anchor (HM 16.20)	[73]	[1]	[76]	CVEGAN
Average DMOS	1.97	1.38	1.40	1.42	**1.33**

Table 5.15 presents the average DMOS (difference of the mean opinion score) values of all evaluated sequences for CVEGAN, HEVC and the three top-performing networks. The average DMOS for CVEGAN is lower than that for HEVC anchor and the other networks, providing further evidence of its effectiveness.

5.5 Summary

In this chapter, the perceptually-inspired video coding enhancement algorithms are presented which are based on GAN architectures and perceptual loss functions. Firstly, two preliminary works, MSRGAN and BDGAN are presented for spatial resolution and effective bit depth adaptations, respectively. Using the simple perceptual loss functions (i.e. SSIM and MS-SSIM) and RaGAN in training process, both the MSRGAN and BDGAN have achieved evident coding gains (assessed by VMAF) over the HEVC HM 16.20.

In order to achieve more significant coding gains and higher reconstruction quality, the network structures were re-designed and a novel GAN architecture-CVEGAN has been further presented for video compression enhancement. This network, when integrated into conventional video coding systems (HEVC HM 16.20 and VVC VTM 7.0), has enabled significantly improved coding performance compared to many state-of-the-art architectures across multiple test datasets. The experimental results demonstrate evident coding gains, with a 23.4% improvement for PP, 38.4% for SRA and 26.3% for EBDA over HM 16.20 in terms of VMAF, and a corresponding 8.0% for PP, 20.3% for SRA and 7.8% for EBDA against VTM 7.0. This enhanced performance can be attributed to the use of several new features including novel Mul^{2}Res blocks, ERNB, ECBAM, the new ReSphereGAN training methodology and perceptually-inspired loss functions.

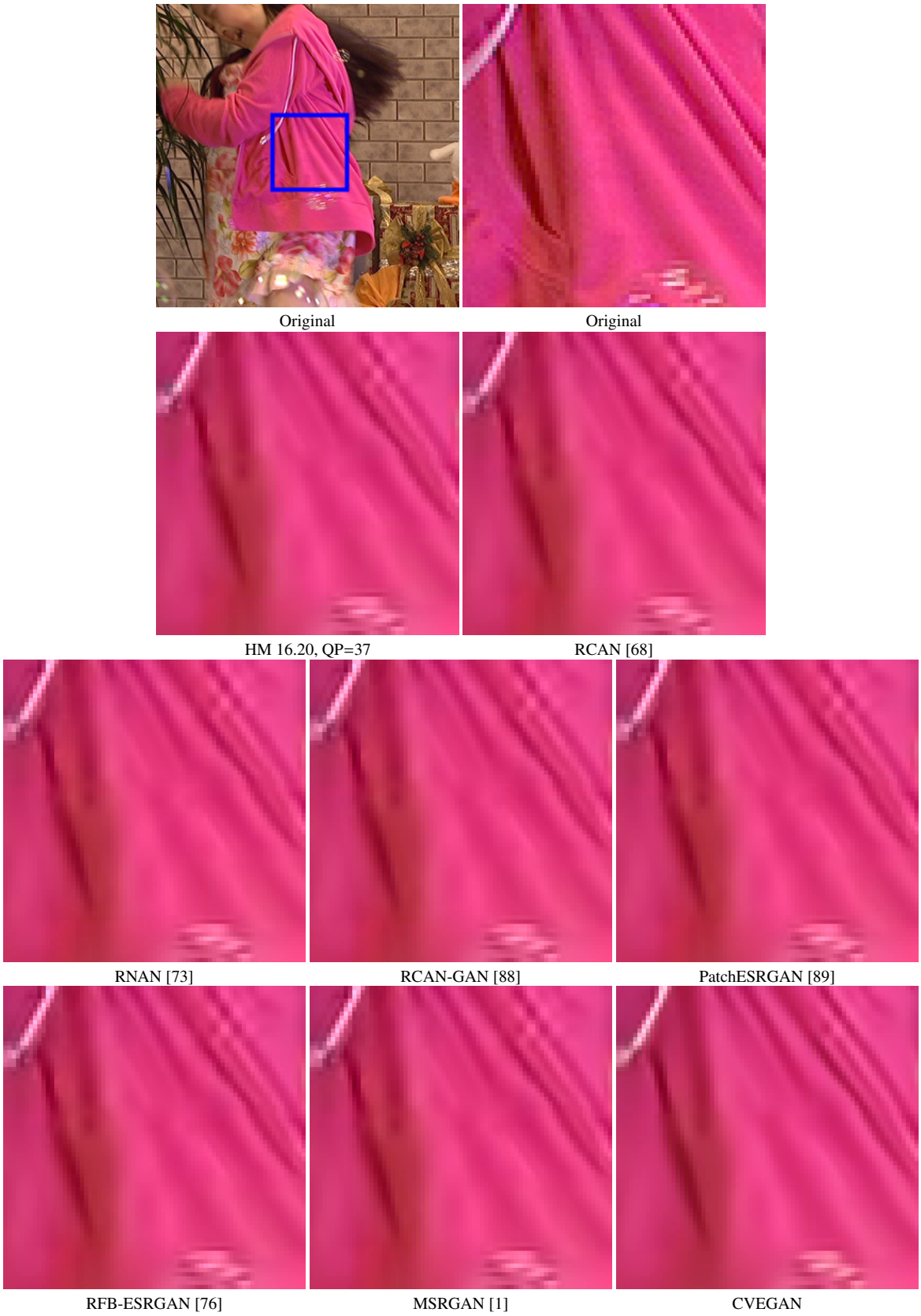


Figure. 5.14 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2, 3 and 4 correspond to the 170th frame of the *PartyScene* sequence.

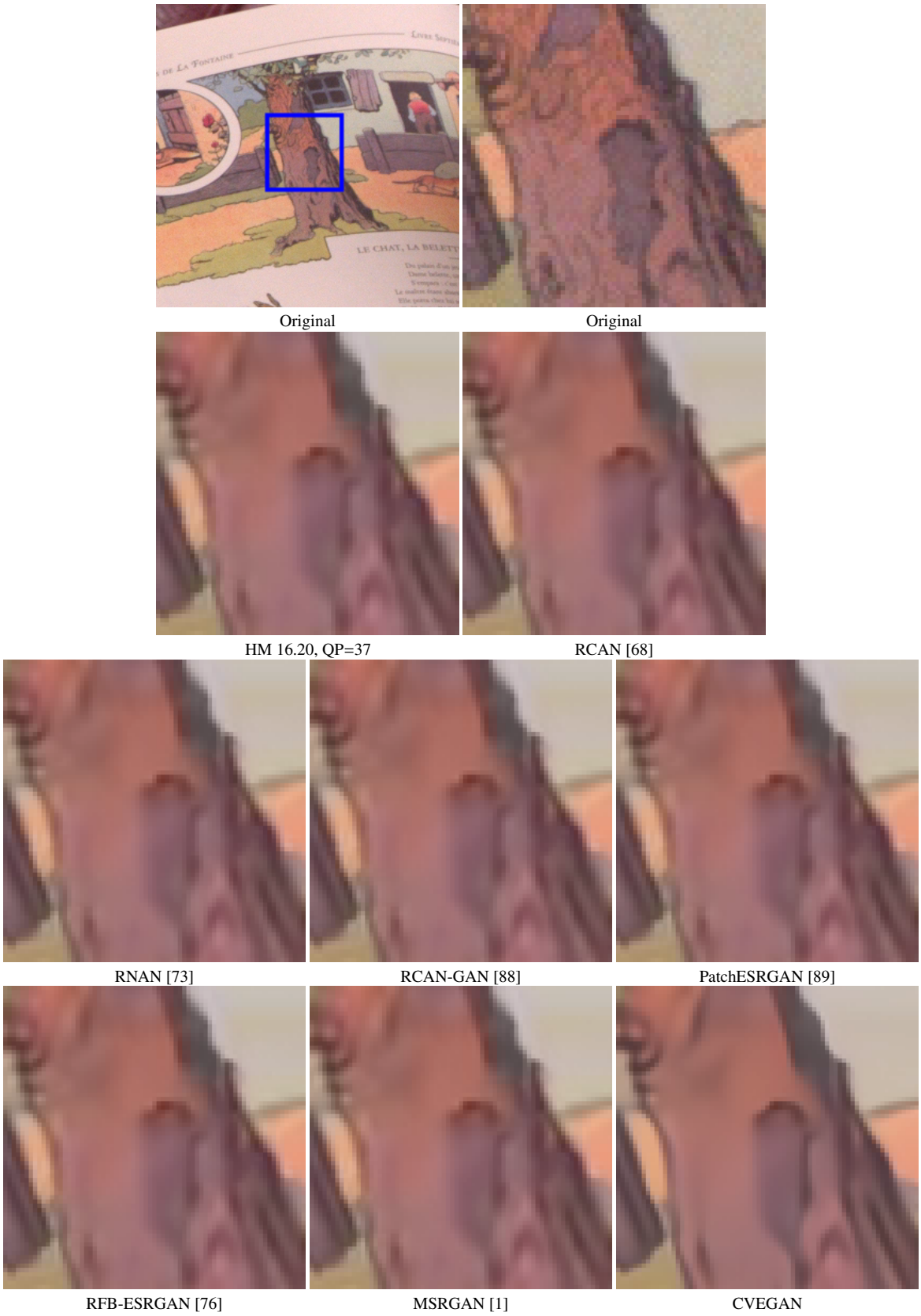


Figure. 5.15 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2, 3 and 4 correspond to the 104th frame of the *CatRobot1* sequence.

Figure. 5.16 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2, 3 and 4 correspond to the 250th frame of the *DaylightRoad2* sequence.

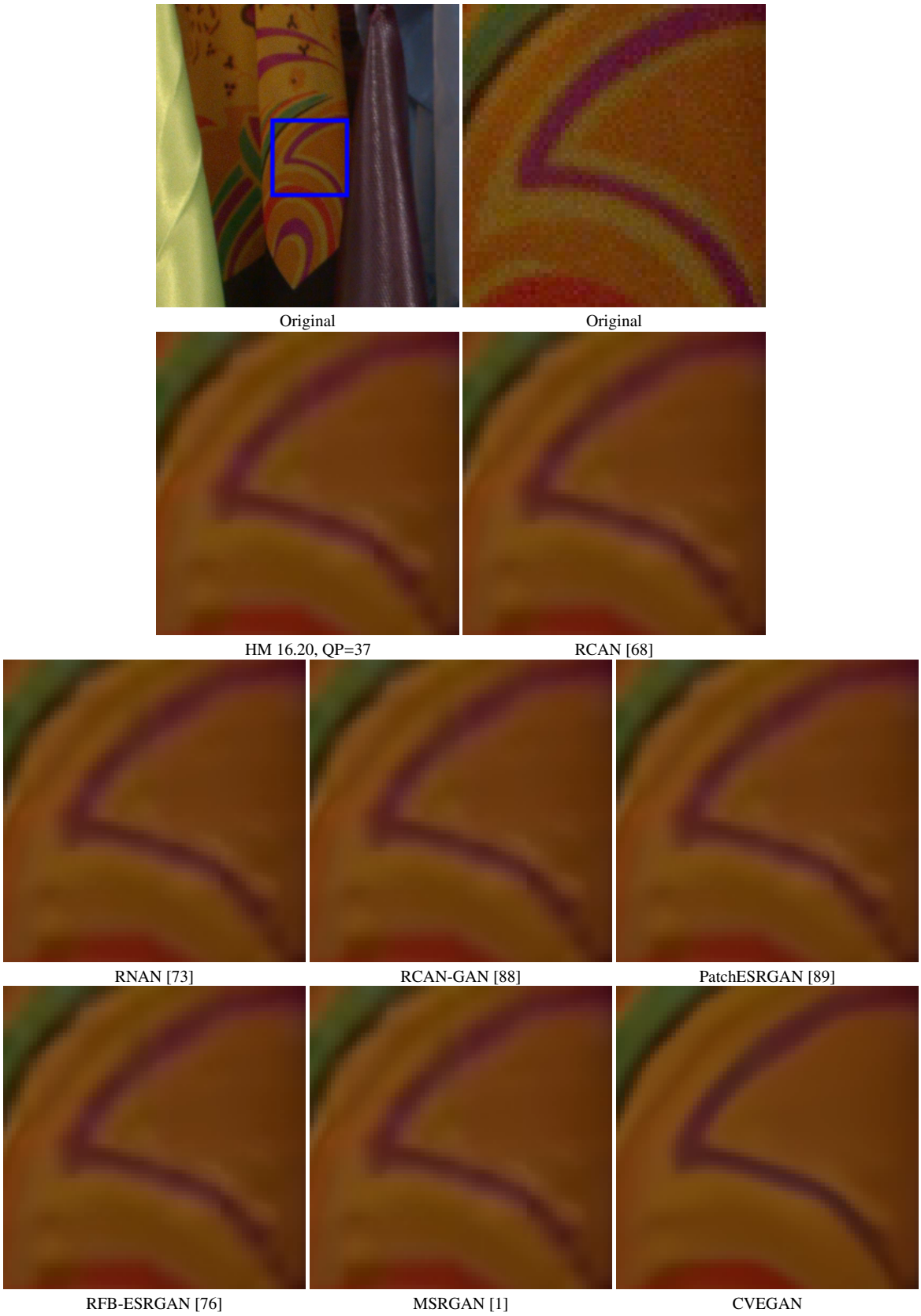


Figure. 5.17 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based PP. The bit consumption in each example set is identical for all tested versions. Rows 1, 2, 3 and 4 correspond to the 216th frame of the *CatRobot1* sequence.

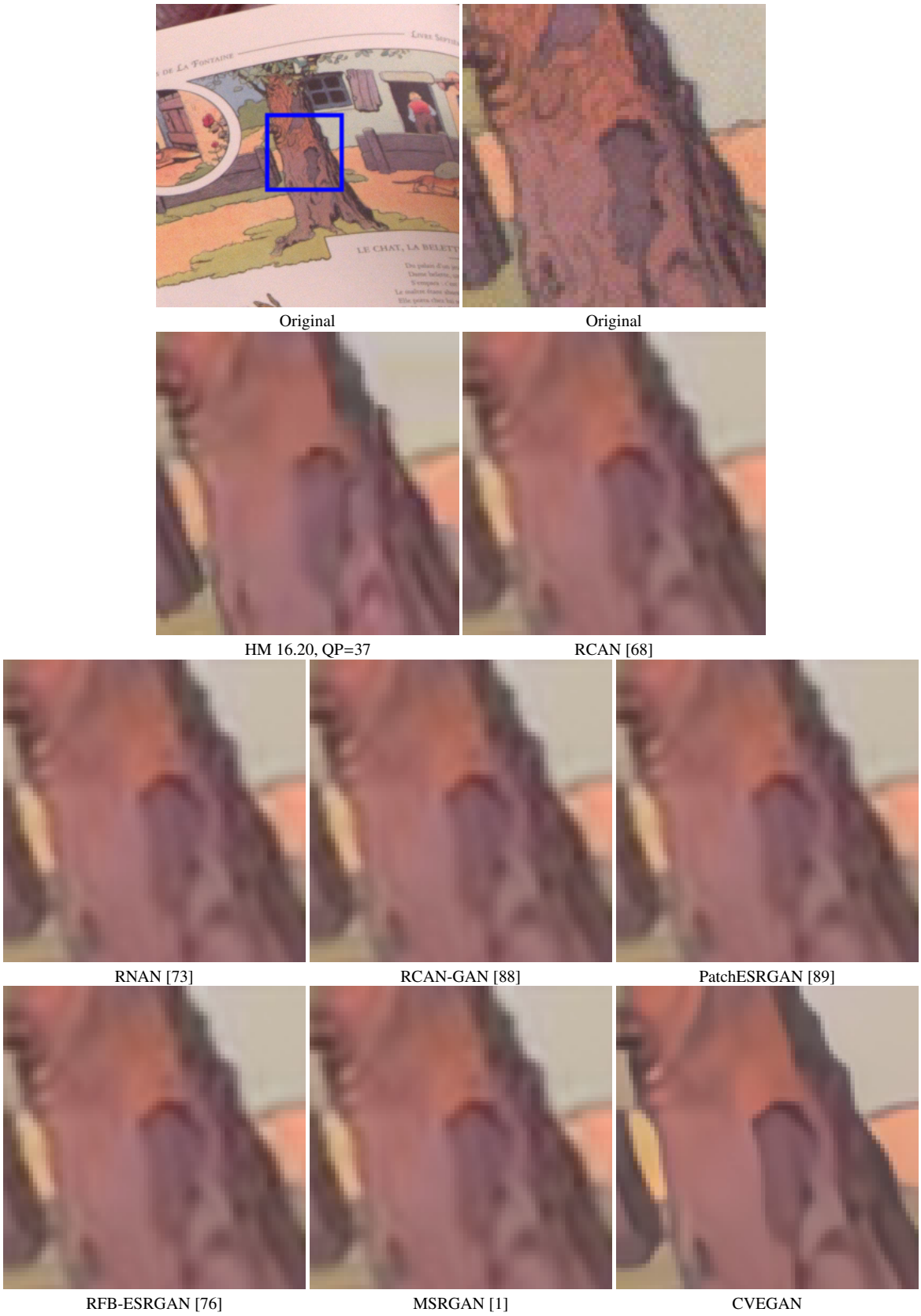


Figure. 5.18 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based SRA. The bit consumption in each example set is similar for all tested versions. Rows 1, 2, 3 and 4 correspond to the 104th frame of the *CatRobot1* sequence.

Figure. 5.19 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based SRA. The bit consumption in each example set is similar for all tested versions. Rows 1, 2, 3 and 4 correspond to the 250th frame of the *DaylightRoad2* sequence.

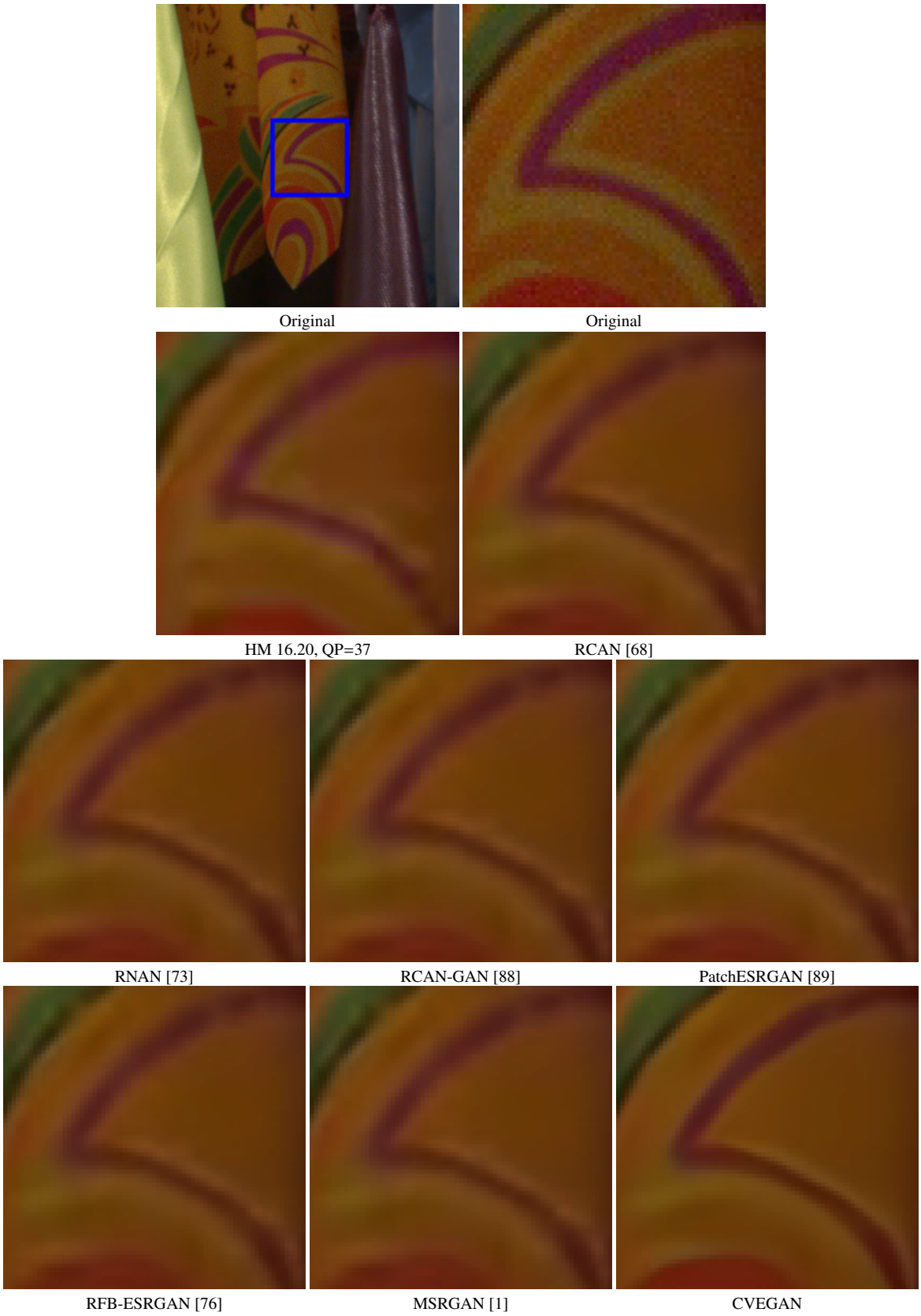


Figure. 5.20 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based SRA. The bit consumption in each example set is similar for all tested versions. Rows 1, 2, 3 and 4 correspond to the 216th frame of the *CatRobot1* sequence.

Figure. 5.21 One set of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), six state-of-the-art networks and the proposed CVEGAN for CNN-based SRA. The bit consumption in each example set is similar for all tested versions. Rows 1, 2, 3 and 4 correspond to the 161st frame of the *RaceNight* sequence.

Chapter 6

Complexity Analysis

In previous Chapters 4 and 5, several novel network architectures and new training methodologies have been presented to provide significant coding gains over the standard codecs (HEVC HM 16.20, VVC VTM 7.0 and AV1) based on the typical video coding enhancement modules. However, some of these methods suffer from relatively high computational complexity, which may result in compatibility issues when they are employed in practise. In this context, it is important to comprehensively analyse the complexity issue (through the analysis of the execution time of different networks in this chapter) of these types of methods, and further develop low complexity network architectures and coding tools. In this chapter, the latencies of the network architectures which have been presented in the previous chapters are first analysed. Based on these, a new low complexity CNN-based video compression framework is then presented for the SRA coding module. It employs a CNN model for video down-sampling at the encoder and uses a Lanczos3 filter to reconstruct full resolution at the decoder. This approach offers a trade-off solution between computational complexity and coding performance, and enables flexible complexity allocation between the encoder and decoder.

The work presented in this chapter has been published in [5, 6].

6.1 Complexity Analysis for the Proposed Networks

In this section, the latencies of different network architectures are first analysed. Then, the relationship between the number of residual blocks used in MSRResNet (one of our primary works presented in Chapter 5) and coding performance is further presented. It is noted that all the experiments in this chapter are based on a shared cluster, BlueCrystal Phase 4 (BC4) based in the University of Bristol [211], in which each node contains two 14 core 2.4 GHz Intel E5-2680 V4 (Broadwell) CPUs, 128 GB of RAM, and NVIDIA P100 GPU devices.

6.1.1 Latency Analysis

Latency is an important measurement in deep learning to assess the complexity of neural networks for a specific application. It is generally defined as the time taken to process one image block [43]. In this section, the image block size is fixed to 96×96 which is the same as the block size used to train presented networks in previous chapters.

Table 6.1 summarises the latency results for different network architectures, including four networks presented in previous chapters (MFRNet, MSRResNet, BDNet and CVENet) and other popular CNN/GAN architectures as described in Sections 2.2.2 and 2.2.3.

Table 6.1 Latency results (millisecond-ms) of different network architectures.

	SRCNN	FSRCNN	VDSR	DRRN	EDSR
Latency (ms)	27.24	37.32	57.21	74.91	119.86
	SRResNet	CARN	UDSR	HR-EnhanceNet	ESRResNet
Latency (ms)	33.51	61.30	84.44	78.18	106.24
	RCAN	RDN	RNAN	ADGAN	SRResCGAN
Latency (ms)	130.21	95.88	160.72	69.73	49.13
	PCARNGAN	RCAGAN	RCAN-GAN	PatchESRGAN	RFB-ESRGAN
Latency (ms)	61.42	92.62	131.91	106.33	126.12
	MFRNet	**MSRResNet**	**BDNet**	**CVENet**	
Latency (ms)	43.88	33.85	45.54	56.89	

It can be observed from Table 6.1 that SRCNN has the shortest latency compared to other networks due to the simple network structures. The presented architectures in this thesis (MFRNet, MSRResNet, BDNet and CVENet) have appropriate latencies which are smaller than other complex networks, such as ESRResNet, RCAN, RDN, RNAN RCAGAN, RCAN-GAN, PatchESRGAN, and RFB-ESRGAN. This demonstrates that the carefully designed network architectures can relatively reduce computational cost which is beneficial for reducing the complexity of CNN-based video coding enhancement modules.

6.1.2 Analysis of Network Complexity and Coding Performance

In this section, the relationship between the number of residual blocks and compression performance has been further investigated. Here, the MSRResNet (presented in Chapter

5) was trained ($\ell 1$ loss function) using the BVI-DVC database based on the same training methodologies and experimental configurations which have been described in Sections 3.2.3-3.2.5. The MSRResNet was integrated into the post-processing coding tool and fully evaluated on the JVET-CTC test sequences using the Random Access configuration (Main10 profile) based on the VVC VTM 4.0.1 [5].

Figure. 6.1 shows the coding gains (in terms of PSNR) and algorithm relative complexity (ratio of the average execution time of CNN-based coding tool and anchor codec over all tested sequences and QP values) using CNN models with different numbers of residual blocks (N=4, 8, 12, 16, 20, 24, 28 and 32) utilised in MSRResNet to process VVC VTM QP 42 compressed content. Again the relative complexity here is benchmarked on the original VVC decoder. It can be observed that, when the number of residual blocks (N) increases from 4 to 16, the PSNR gain relative to the original VVC content increases in a linear fashion. However, when the number of residual blocks exceeds 20, the overall coding gain starts to decrease. This provides evidence that the residual block number in the proposed work is an optimal selection.

It can be also noted that simply increasing the depth of the network (e.g. increasing the number of residual blocks) without changing main structures may lead to worse network performance and significantly increasing computational complexity. This is likely due to the vanishing gradient issue occurred in deep CNN training process [55].

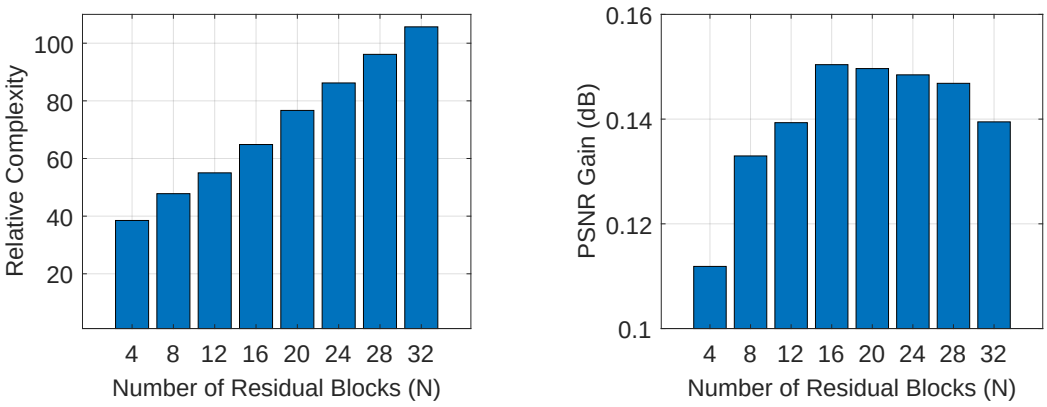


Figure. 6.1 (Left) Relative complexity (decoding) for different number of residual blocks. (Right) PSNR gains for different number of residual blocks.

6.2 New Coding Framework with Reduced Computational Complexity

In this section, a new low complexity CNN-based coding framework is presented which is based on one of the primary spatial resolution adaptation (SRA) workflow scenarios.

6.2.1 Different SRA Scenarios

A generic spatial resolution adaptation (SRA) framework for video compression is illustrated in Figure. 6.2. According to the various possible approaches employed in spatial down-sampling and up-sampling, there exist four different scenarios:

- **Scenario 1**: Simple filters (e.g. Lanczos3) are used for both down-sampling and up-sampling.
- **Scenario 2**: A CNN model is utilised for down-sampling, and up-sampling is achieved through simple filtering.
- **Scenario 3**: A simple filter is employed for down-sampling, while a CNN-based super-resolution approach is used for up-sampling.
- **Scenario 4**: Both down-sampling and up-sampling processes are CNN-based.

It is noted that Scenarios 1, 3 and 4 have been previously investigated [209, 173, 240]. However, we are not aware of successful examples based on Scenario 2. Different from the conventional SRA methods (e.g. Scenario 1 or 3), in Scenario 2, the CNN model was used as a pre-processing filter to spatially down-sample original video content before encoding. In this case, the employed CNN needs to ensure that the host codec can achieve optimal rate-distortion performance when encoding the down-sampled video frames. This requires the CNN to produce content with adequate information for decoder-based up-sampling (the final distortion should be low). At the same time, the CNN output should be relatively 'easy' for encoding (the bitrate should not be too high). In order to optimise this process, the ideal solution is to include a standard codec in the network training process to encode CNN output and produce corresponding bitrates for rate-distortion optimisation. However, the existing standard codecs (e.g. HEVC HM, VVC VTM) are currently not compatible with the deep learning libraries and are therefore difficult to be integrated for end-to-end training.

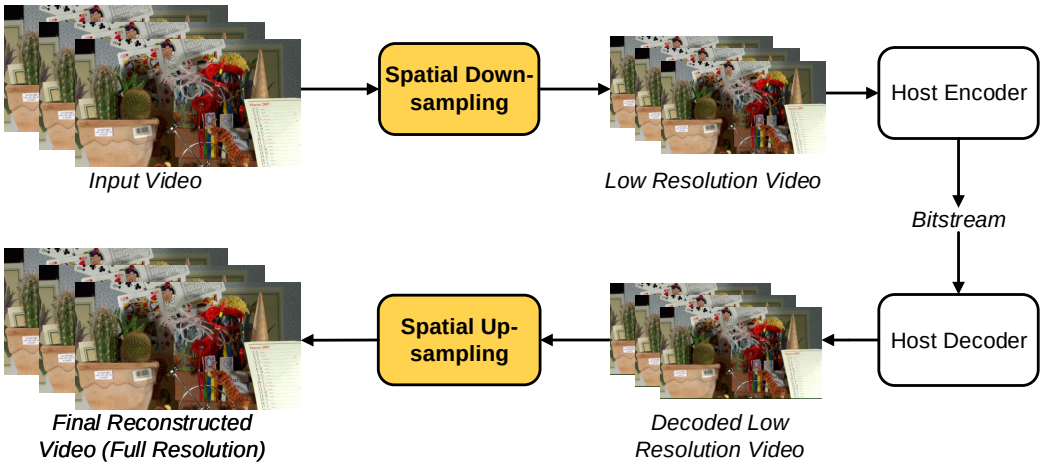


Figure. 6.2 Diagram of the generic spatial resolution adaptation workflow.

6.2.2 SRA with CNN-based Down-sampling

Among the four different scenarios described above, Scenarios 1 and 2 are commonly associated with low decoding complexity due to the use of simple filters for spatial resolution super-resolving. In this work, we solely focus on Scenario 2, targeting improved overall coding performance by integrating a down-sampling CNN.

It is noted that when simple filters (e.g. Bicubic or Lanczos3) are used to achieve spatial resolution down-sampling (as in Scenarios 3), constant filter parameters often result in the loss of spatial information during the down-sampling process. When CNNs are employed for down-sampling, more important high frequency information may be preserved which is beneficial for reconstructing high quality full-resolution video frames at the decoder. Moreover, although the employed CNN operation will increase the encoder complexity, the relative overhead is much lower than cases where the CNNs are used at the decoder for up-sampling. This approach offers a trade-off solution between computational complexity and coding performance, and enables flexible complexity allocation between the encoder and decoder.

The architecture of the spatial down-sampling network (DSNet) is shown in Figure. 6.3. This network is based on a modified version of our own CNN architecture developed for bit depth up-sampling (BDNet presented in Chapter 5) [2]. It takes a 96×96 YCbCr (4:4:4) image block as input, and outputs a low resolution block (48×48) in the same format. The input signal is first processed by a shallow spatial down-sampling layer and a feature extraction layer, each of which consists of a convolutional layer and a Leaky ReLU (LReLU) activation function. After the shallow feature extraction layer, 14 residual dense blocks (RDBs) [66] were employed to further extract dense features. Multiple cascading connections (shown as

black curves in Figure. 6.3) are designed to connect these 14 RDBs and feed the outputs of the spatial down-sampling layer and each RDB (G_i, $i = 1, 2, ..., 13$) into the subsequent RDBs or the first reconstruction layer (RL1) through a 1×1 convolutional layer with a LReLU activation function. A skip connection is further utilised to connect the outputs of the shallow feature extraction layer and RL1. Another reconstruction layer (RL2) is employed which is followed by the final convolution layer to produce the residual signal. Finally, the input image block is spatially down-sampled by a factor of 2 using a Bilinear filter and combined with the residual signal using a long skip connection to output the final down-sampled image block. The number of feature maps, kernel sizes and stride values for all convolutional layers are presented in Figure. 6.3.

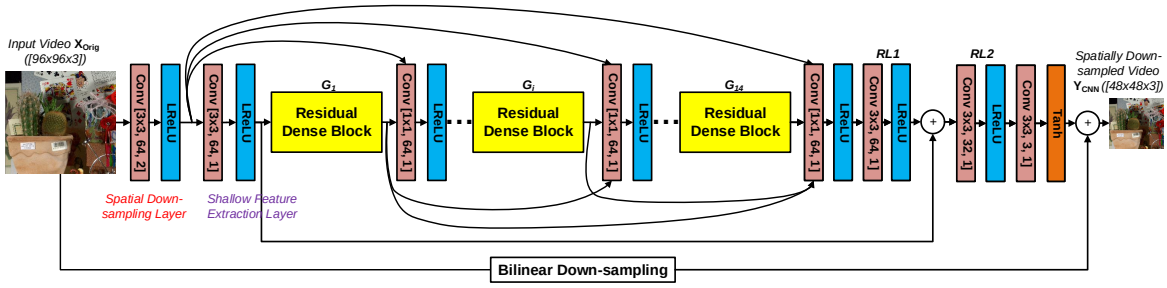


Figure. 6.3 Network architecture of the proposed DSNet.

Figure. 6.4 illustrates the structure of each RDB employed in the proposed network. Due to the dense connection, the convolutional layer in each RDB fully reuses features from its preceding layers [67], which effectively improves information flow between these layers. An additional skip connection is also designed to connect the input and output of each RDB in order to stabilise training and evaluation processes [55].

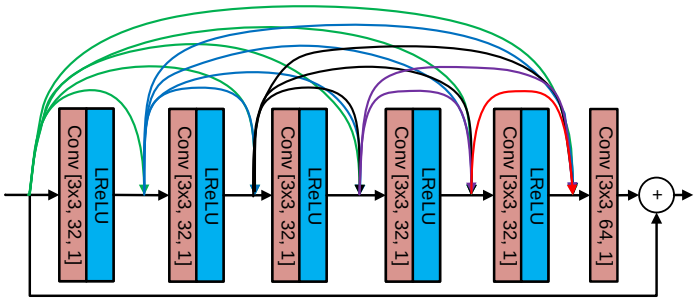


Figure. 6.4 Residual Dense Block (RDB) used in DSNet.

6.2.3 Training Methodology

Training Database

To effectively train the presented DSNet, the video training database BVI-DVC which has been described in Chapter 3 is employed to generate training material. The video frames of these sequences were randomly selected, segmented into 96×96 image blocks, and converted into YCbCr 4:4:4 format. During this process, block rotation is applied to achieve data augmentation. This results in approximately 200,000 image blocks in total. They are employed as both input and the training target of the CNN.

Loss Function

Loss functions are a key component in CNN training. Since the proposed network is used for resolution down-sampling before encoding, the CNN output (at low resolution) should preserve sufficient spatial information to enable high fidelity full resolution reconstruction (up-sampling) at the decoder. On the other hand, too much high frequency information may lead to higher bitrates during compression. This results in an optimisation problem which is similar to traditional rate-distortion optimisation [9, 241] in image and video coding. In this context, the conventional loss functions discussed in Section 2.3.4 which are only used to optimise the quality of reconstructed video frames cannot be directly employed to train DSNet.

To solve this problem, the ideal solution would be to conduct an end-to-end optimisation which includes an image or video codec in the training loop – to encode the low resolution CNN output and generate a real bitstream whose corresponding bitrate (R) could be measured. The decoded low-resolution image block could then be up-sampled using a simple filter (e.g. Lanczos3) to get the final reconstructed full resolution content for comparison with its original (uncompressed full resolution) counterpart to calculate overall distortion (D). The loss function ($\mathcal{L}$) used during CNN training can be designed as equation (6.1) employing the Lagrange multiplier method.

$$\mathcal{L} = D(\mathbf{p}) + \lambda_{\text{CNN}} \cdot R(\mathbf{p}) \tag{6.1}$$

Here, $\mathbf{p}$ represents the CNN parameters which need to be optimised during training. λ_{CNN} is the Lagrange multiplier which is used to trade off the relationship between D and R.

In practice, it is noted that conventional image or video codecs, such as HEVC HM, cannot be integrated into the training loop due to incompatibility with existing machine learning libraries (e.g. TensorFlow and PyTorch) [240]. In order to optimise the proposed network and achieve superior overall rate quality performance (based on the framework in

Scenario 2), a loss function is proposed to emulate the rate-distortion optimisation process, as shown in equation (6.2).

$$\mathcal{L}_{\mathrm{DSNet}} = \mathrm{MSE}(\mathrm{X}_{\mathrm{Orig}}, \mathrm{Y}_{\mathrm{CNN\_BicUp}}) + \lambda_{\mathrm{DSNet}} \cdot (\mathrm{MSE}(\mathrm{Y}_{\mathrm{L3}}, \mathrm{Y}_{\mathrm{CNN}}) + \omega \cdot \mathcal{L}_{\mathrm{MS-SSIM}}(\mathrm{Y}_{\mathrm{L3}}, \mathrm{Y}_{\mathrm{CNN}})) \tag{6.2}$$

The first term $\mathrm{MSE}(\mathrm{X}_{\mathrm{Orig}}, \mathrm{Y}_{\mathrm{CNN\_BicUp}})$ calculates the mean squared error (MSE) between the original full resolution input block ($\mathrm{X}_{\mathrm{Orig}}$) and the Bicubic up-sampled CNN output ($\mathrm{Y}_{\mathrm{CNN\_BicUp}}$). This accounts for the distortion generated during the resolution adaptation process. $\mathrm{MSE}(\mathrm{Y}_{\mathrm{L3}}, \mathrm{Y}_{\mathrm{CNN}})$ represents the MSE between the CNN output low resolution image block ($\mathrm{Y}_{\mathrm{CNN}}$) and the Lanczos3 filter down-sampled (from the original) low resolution image block (Y_{L3}), and their MS-SSIM [27] loss is also obtained in the term $\mathcal{L}_{\mathrm{MS-SSIM}}(\mathrm{Y}_{\mathrm{L3}}, \mathrm{Y}_{\mathrm{CNN}})$. The weighted linear combination between $\mathrm{MSE}(\mathrm{Y}_{\mathrm{L3}}, \mathrm{Y}_{\mathrm{CNN}})$ and $\mathcal{L}_{\mathrm{MS-SSIM}}(\mathrm{Y}_{\mathrm{L3}}, \mathrm{Y}_{\mathrm{CNN}})$ is employed to estimate the bitrate level when the CNN low resolution output is compressed. ω and λ_{DSNet} are two constant parameters representing the weights used in the combination model and the Lagrange multiplier respectively.

6.2.4 Training and Evaluation Configurations

The same procedures of network training configurations that have been introduced in Section 3.2.4 were utilised to train the presented DSNet. The used parameter values for the Lagrange multiplier λ_{DSNet} and the weight ω were 30 and 1/6 respectively which have been determined based on the ten-fold cross-validation method using the BVI-DVC database.

During network evaluation, each full resolution frame of the test sequence is segmented into 96×96 overlapping blocks with the overlap sizes of 24 and/or 16 pixels (determined by the block's location within an input frame) using the similar methods discussed in Section 3.2.4, and converted to YCbCr 4:4:4 format as network input. The network output image blocks (with the size of 48×48) are then converted to the original format (YCbCr 4:2:0) and aggregated in the same way which has been introduced in Section 3.2.4 to generate the spatially down-sampled video frame.

6.2.5 Experimental Configurations

The presented spatial down-sampling CNN architecture has been integrated into the spatial resolution adaptation (SRA) framework (Scenario 2) and fully evaluated with HEVC HM 16.20 as the host codec. The evaluation followed the All Intra (Main 10 Profile) configuration used in the JVET-CTC [242] using six JVET UHD sequences as test material. None of these

test sequences was used for training the presented CNN model. Four initial base quantisation parameter (QP) values are employed: 27, 32, 37 and 42.

In order to achieve similar bitrate ranges and hence a meaningful comparison between the presented approach and the original HEVC, a fixed QP offset of -6 is applied on the base QP value during encoding when SRA is enabled [173]. It is also noted that the coding improvement achieved by spatial resolution adaptation is highly content dependent. For some sequences at certain QP values, SRA may not offer coding gains over the original host codec. Therefore, a quantisation resolution optimisation (QRO) module [57] has been employed, which uses a machine learning-based approach to make decisions on resolution adaptation based on a spatial resolution dependent quality metric, SRQM [243], temporal information (TI) [244] and initial base QP values. For cases when spatial resolution adaptation is not activated, the test sequences will be compressed using the original HEVC HM with the initial base QP.

To benchmark the coding performance of SRA Scenario 2, we have generated results for SRA Scenarios 1, 3 and 4 alongside original HEVC compression (HM 16.20). For simple filter-based down- and/or up-sampling in Scenarios 1 and 3, we have used Lanczos3 filters. In Scenarios 3 and 4, a previously developed super-resolution CNN, MSRResNet [1, 245] is employed for resolution up-sampling at the decoder.

6.2.6 Results and Discussion

Compression Performance

Table 6.2 Compression performance comparison between various SRA scenarios and the original HEVC HM 16.20 (AI configuration) ("↓" and "↑" represent spatial down-sampling and up-sampling respectively).

Sequence	Scenario 1 L3 ↓ & L3 ↑	Scenario 2 CNN ↓ & L3 ↑	Scenario 3 L3 ↓ & CNN ↑	Scenario 4 CNN ↓ & CNN ↑
	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate (PSNR)	BD-rate(PSNR)
Campfire	-7.4%	-8.8%	-17.8%	-18.6%
FoodMarket4	-7.8%	-8.4%	-12.3%	-13.6%
Tango2	-9.7%	-10.6%	-12.8%	-14.8%
CatRobot1	-4.4%	-5.2%	-13.0%	-14.7%
DaylightRoad2	-2.6%	-3.8%	-7.9%	-10.2%
ParkRunning3	-23.6%	-24.5%	-26.9%	-28.2%
Average	-9.2%	-10.2%	-15.1%	-16.7%

Table 6.2 summaries the compression performance for the four different SRA approaches (Scenarios 1-4) when they are compared to that of the original HEVC HM 16.20 using Bjøntegaard Delta [40] measurement (BD-rate) based on the assessment of PSNR (luminance channel only). It can be observed that when a CNN is employed for resolution down-sampling (Scenarios 2 and 4), additional coding gains have been achieved over those scenarios where Lanczos3 down-sampling is applied (Scenario 1 and Scenario 3 respectively). This improvement is consistent among all six test sequences. It can also be noticed that when CNN-based super-resolution is utilised (Scenarios 3 and 4), the bitrate savings are more significant (up to 16.7%) against Scenario 1 and 2 with Lanczos3 up-sampling.

Complexity Analysis

The encoder and decoder complexities for all five evaluated methods (HM 16.20 and SRA Scenario 1-4) were also calculated. The encoding was executed on a shared cluster, BlueCrystal Phase 3 [246], based at the University of Bristol, which has 223 base blades. Each blade contains 16 2.6GHz SandyBridge cores and 64GB RAM. The decoding was conducted on a PC with an Intel(R) Core(TM) i7-4770K CPU @3.5GHz, 24GB RAM and NVIDIA P6000 GPU device. The encoding and decoding execution times for SRA Scenario 1-4 are all benchmarked against those for the original HEVC HM.

Table 6.3 reports the average (for the four evaluated QPs) relative encoding (Enc.) and decoding (Dec.) complexities of the four SRA scenarios for six test sequences. When a Lanczos3 filter is used for down-sampling (Scenario 1 and 3), the encoding complexity is only 44% of that for the original HM. This is due to the simplicity of the down-sampling filter and the encoding of low resolution content. The proposed CNN-based down-sampling in Scenario 2 and 4 can also reduce the overall encoding time by approximately 30% and offers better overall rate quality performance compared to Lanczos3 down-sampling (as shown in Table 6.2). It is also noted that the decoding complexity has also been slightly reduced in Scenario 2 and 4 (with CNN based down-sampling) compared to Scenario 1 and 3 (based on Lanczos3 down-sampling) respectively. This may be because CNN-based down-sampling generated content is relatively easy to compress.

As mentioned in Section 6.2.3, the employed training strategy is a sub-optimal solution. The results presented in this section demonstrate its potential, while acknowledging that and overall coding performance can be further enhanced if a more realistic end-to-end optimisation is applied. Our approach is particularly relevant to application scenarios where there is a limited resource available at the decoder or where CNN-based up-sampling cannot be supported. In such cases, we have shown that consistent coding gains can still be achieved

Table 6.3 Relative complexity for four SRA Scenarios.

Sequence	Scenario 1 L3 ↓ & L3 ↑		Scenario 2 CNN ↓ & L3 ↑		Scenario 3 L3 ↓ & CNN ↑		Scenario 4 CNN ↓ & CNN ↑	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
Campfire	0.47×	0.96×	0.83×	0.95×	0.47×	29.5×	0.83×	29.3×
FoodMarket4	0.23×	0.71×	0.54×	0.68×	0.23×	25.4×	0.54×	25.2×
Tango2	0.32×	0.98×	0.52×	0.95×	0.32×	30.5×	0.52×	30.2×
CatRobot1	0.53×	1.20×	0.85×	1.10×	0.53×	34.3×	0.85×	34.1×
DaylightRoad2	0.34×	0.98×	0.57×	0.94×	0.34×	29.2×	0.57×	29.0×
ParkRunning3	0.73×	0.89×	0.94×	0.86×	0.73×	26.9×	0.94×	26.5×
Average	0.44×	0.95×	0.71×	0.90×	0.44×	29.3×	0.71×	29.1×

by re-distributing the computational complexity from the decoder to the encoder (as in Scenario 2) by applying CNN-based down-sampling instead of CNN-based super-resolution.

6.3 Summary

In this chapter, the latencies of the previously presented network architectures are first analysed. Based on these, a low complexity CNN-based spatial resolution adaptation framework for video compression is then presented. This method employs a CNN-based down-sampling approach before encoding and applies up-sampling at the decoder using a simple filter. The proposed approach has been integrated with HEVC HM 16.20 and evaluated on JVET-CTC UHD test sequences (under the All Intra configuration). Improved coding performance has been achieved compared to the original HEVC HM (with an average coding gain of 10.2% based on PSNR) and against Lanczos3 filter based re-sampling (with an overall additional coding improvement of 1%), coupled with reduced computational complexity at both encoder (29%) and decoder (10%).

Chapter 7

Conclusion

From the introduction of the first international standard in 1984, video compression has played an essential role in the application and uptake of video technologies across film, television, terrestrial and satellite transmission, surveillance and particularly Internet video [57, 9]. Inspired by recent breakthroughs in AI technology, in this thesis, important works have been presented which employ deep learning methods such as CNNs or GANs to enhance video coding algorithms [23]. In this chapter, the main contributions of these works and potential future work are summarised.

The work presented in this chapter has been published in [1–7].

7.1 Contributions

This thesis has explored advanced deep learning methodologies to significantly improve coding performance over the current standard codecs, HEVC HM 16.20, VVC VTM 7.0 and AV1 based on the typical CNN-based video coding enhancement modules, post-processing (PP), in-loop filtering (ILF), spatial resolution adaptation (SRA) and effective bit depth adaptation (EBDA). The main contributions are summarised as follows.

- **A new extensive and representative video database (BVI-DVC) for training deep video compression algorithms (Chapter 3).** This database contains 800 video sequences carefully selected from publicly available databases and websites, covering a large variety of content types (including different video textures-static and dynamic) and scenes. The BVI-DVC database has effectively optimised the generalisation ability of the networks and significantly improved coding performance compared to the commonly used image and video training datasets. The overall additional coding improvements by using the BVI-DVC database for all tested coding modules and CNN

architectures are up to 10.3% (assessed by PSNR) and 8.1% (assessed by VMAF). This database has recently been used by JVET Ad-hoc Group 11 (Neural-network-based video coding).

- **A new CNN architecture, MFRNet, for video compression enhancement (Chapter 4).** MFRNet comprises novel multi-level feature review residual dense blocks, and employs a cascading structure to effectively improve the overall performance of the network. The experimental results demonstrate significant coding gains have been achieved against the standard codecs (HEVC HM 16.20 and VVC VTM 7.0) and other state-of-the-art deep networks (average additional coding gains up to 11.6% and 11.7% have been achieved based on PSNR and VMAF, respectively).

- **Novel GAN architectures including CVEGAN for enhancing perceptual video quality of compressed content (Chapter 5).** The CVEGAN employs novel Mul^{2}Res blocks, enhanced residual non-local blocks and enhanced convolutional block attention modules. The training strategy has also been re-designed specifically for video compression applications, to employ a relativistic sphere GAN (ReSphereGAN) training methodology together with new perceptual loss functions. The experimental results demonstrate that the CVEGAN has enabled significantly improved coding performance compared to many state-of-the-art architectures (an overall additional coding improvement up to 18.1% has been provided based on VMAF).

- **New low complexity CNN-based coding framework (Chapter 6).** The high computational complexity issue of deep learning-based coding methods has also been addressed in this thesis by designing a new framework with a CNN-based down-sample approach. This framework has achieved improved coding performance compared to the original HEVC HM (with an overall coding gain of 10.2% based on PSNR) and against Lanczos3 filter based re-sampling (with an average additional coding gain of 1%), coupled with reduced computational complexity at both encoder (29%) and decoder (10%).

In this thesis, several novel network architectures have been presented for video coding enhancement using the basic backbone structure as discussed in Section 2.2.2. Based on the discussions and analyses outlined in the previous chapters, there are some important aspects related to image and video restoration network development can be further concluded. These include: (1) the residual learning structures are an essential component of networks which can stabilise training and evaluation processes (especially when training deeper networks); (2) improving information flow is very important which can effectively enhance networks'

representational ability and capacity. Commonly used ways to achieve this goal include employing residual dense connections, cascading and feature review structures, etc.; (3) the *cardinality* is one of the important factors related to networks performance. Increasing *cardinality* with multiple convolutional branches can effectively improve the capacity and overall performance of the networks. In addition, employing convolutional layers with various kernel sizes can relatively enlarge receptive field of the networks which also leads to the improvement of overall performance; and (4) the representational and reconstruction abilities of the networks can be also enhanced by exploiting the non-local and channel-spatial attention operations. They are more suitable for the networks which are designed to optimise visual quality of image or video content. It is noted that the network structures discussed above were developed for conventional SDR 2D image or video data. When enhancing the quality of other content types (e.g. high dynamic range or 360° content), the architectures of these structures may need to be modified further.

7.2 Future Work

According to the developed approaches and results generated, the future work should mainly focus on the following aspects:

- New training databases containing immersive video content. The BVI-DVC training database mainly designed for conventional SDR video content compression, while CNN-based video coding methods can also be applied to other video content types, such as high dynamic range, high frame rate, 360°, etc. Future work should focus on developing large training databases with more immersive video formats (including higher dynamic range, higher frame rates, etc.) which are more suitable for different content types.

- New high-performance network architectures with reduced computational complexity. The presented CNN and GAN architectures in this thesis have achieved significant coding gains over standard codecs and the state-of-the-art network architectures. However, they are also associated with increased coding complexity. Future work should carefully address network complexity issue. The possible solutions include re-designing the lightweight network architectures and model compression algorithms.

- Novel perceptual inspired loss-functions. The new perceptually-inspired loss function presented in this thesis has achieved better correlation performance with the subjective opinions compared to other existing popular loss functions. However, the correlation

performance is still far from perfect. In this case, it is worth developing a more effective CNN-based video quality assessment metric. It should have the following features: (1) higher video quality prediction accuracy and low computational complexity; (2) differentiable and easily integrated into the commonly used deep learning libraries (e.g. TensorFlow and PyTorch). This CNN-based quality assessment algorithm can be utilised as a perceptual loss function in network training process to further optimise the perceptual quality of compressed video content.

- End-to-end training and optimisation. The experimental results presented in Chapter 6 demonstrate the potential of the low complexity CNN-based SRA coding framework. Future work should continue to enhance the CNN training methodology, for example integrating an End-to-End video compression framework for rate-distortion optimisation. In addition, future work should focus on extending the approach to inter-coding configurations and optimise our CNN models for different quantisation level ranges.

Publications

arXiv Papers

1. D. Ma, F. Zhang, and D. R. Bull, "CVEGAN: a perceptually-inspired GAN for compressed video enhancement," arXiv preprint arXiv:2011.09190, 2020 (under review by Elsevier Journal of Signal Processing: Image Communication).

Journal Papers

1. D. Ma, F. Zhang, and D. R. Bull, "BVI-DVC: a training database for deep video compression," IEEE Transactions on Multimedia, 2021.

2. F. Zhang, D. Ma, C. Feng, and D. R. Bull, "Video compression with CNN-based post processing," IEEE MultiMedia, vol. 28, no. 4, pp. 74 - 83, 2021.

3. D. Ma, F. Zhang, and D. R. Bull, "MFRNet: a new CNN architecture for post-processing and in-loop filtering," IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 2, pp. 378 - 387, 2020.

4. A. V. Katsenou, G. Dimitrov, D. Ma, and D. R. Bull, "BVI-SynTex: a synthetic video texture dataset for video compression and quality assessment," IEEE Transactions on Multimedia, vol. 23, pp. 26 - 38, 2020.

Conference Papers

1. A. Mackin, D. Ma, F. Zhang, and D. R. Bull, "A subjective study on videos at various bit depths," in 2021 IEEE Picture Coding Symposium (PCS). IEEE, 2021, pp. 1 - 5.

2. D. Ma, F. Zhang, and D. R. Bull, "GAN-based effective bit depth adaptation for perceptual video compression," in 2020 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2020, pp. 1 - 6.

3. D. Ma, F. Zhang, and D. R. Bull, "Video compression with low complexity CNN-based spatial resolution adaptation," in Applications of Digital Image Processing XLIII, vol. 11510. International Society for Optics and Photonics, 2020, pp. 115100D.

4. D. Ma, M. F. Afonso, F. Zhang, and D. R. Bull, "Perceptually-inspired super-resolution of compressed videos," in Applications of Digital Image Processing XLII, vol. 11137. International Society for Optics and Photonics, 2019, pp. 310 - 318.

5. D. Ma, A. V. Katsenou, and D. R. Bull, "A synthetic video dataset for video compression evaluation," in 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019, pp. 1094 - 1098.

6. A. V. Katsenou, D. Ma, and D. R. Bull, "Perceptually-aligned frame rate selection using spatio-temporal features," in 2018 IEEE Picture Coding Symposium (PCS). IEEE, 2018, pp. 288 - 292.

Other Publications

1. A. V. Katsenou, A. Mackin, D. Ma, F. Zhang and D. R. Bull, "Exploring the challenges of higher frame rates: from quality assessment to frame rate selection," IEEE COMSOC MMTC Communications - Frontiers (E-Letter), vol. 13, no. 3, pp. 5 - 10, 2018 (invited).

Databases

1. BVI-DVC: https://fan-aaron-zhang.github.io/BVI-DVC/ (BVI-DVC has been recently used by MPEG JVET to develop CNN-based coding for Versatile Video Coding.)

2. BVI-SynTex: https://data.bris.ac.uk/data/dataset/320ua72sjkefj2axcjwz7u7yy9

References

[1] D. Ma, M. F. Afonso, F. Zhang, and D. R. Bull, "Perceptually-inspired super-resolution of compressed videos," in *Applications of Digital Image Processing XLII*, vol. 11137. International Society for Optics and Photonics, 2019, pp. 310–318.

[2] D. Ma, F. Zhang, and D. R. Bull, "GAN-based effective bit depth adaptation for perceptual video compression," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[3] D. Ma, F. Zhang, and D. R. Bull, "BVI-DVC: a training database for deep video compression," *IEEE Transactions on Multimedia*, 2021.

[4] D. Ma, F. Zhang, and D. R. Bull, "MFRNet: a new CNN architecture for post-processing and in-loop filtering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 378–387, 2020.

[5] F. Zhang, D. Ma, C. Feng, and D. R. Bull, "Video compression with CNN-based post processing," *IEEE MultiMedia*, vol. 28, no. 4, pp. 74–83, 2021.

[6] D. Ma, F. Zhang, and D. R. Bull, "Video compression with low complexity CNN-based spatial resolution adaptation," in *Applications of Digital Image Processing XLIII*, vol. 11510. International Society for Optics and Photonics, 2020, p. 115100D.

[7] D. Ma, F. Zhang, and D. R. Bull, "CVEGAN: a perceptually-inspired GAN for compressed video enhancement," *arXiv preprint arXiv:2011.09190*, 2020.

[8] C. V. N. Index, "Cisco visual networking index: forecast and methodology, 2016–2021," *Complete Visual Networking Index (VNI) Forecast*, vol. 12, no. 1, pp. 749–759, 2017.

[9] D. R. Bull and F. Zhang, *Intelligent Image and Video Compression: Communicating Pictures*. Elsevier, 2021.

[10] ITU-T Rec. H.264, "Advanced Video Coding for generic audiovisual services," ITU-T Std., (2005).

[11] ITU-T Rec. H.265, "High Efficiency Video Coding," ITU-T Std., (2015).

[12] B. Bross, J. Chen, S. Liu, and Y.-K. Wang, "Versatile Video Coding (Draft 10)," in *JVET-S2001. ITU-T and ISO/IEC*, 2020.

[13] B. Bross, J. Chen, J. Ohm, G. Sullivan, and Y. Wang, "Developments in international video coding standardization after AVC with an overview of Versatile Video Coding (VVC)," in *Proceedings of IEEE*, 2021, pp. 1–31.

[14] F. Zhang, A. V. Katsenou, M. Afonso, G. Dimitrov, and D. R. Bull, "Comparing VVC, HEVC and AV1 using objective and subjective assessments," *arXiv preprint arXiv:2003.10282*, 2020.

[15] Y. Chen, D. Mukherjee, J. Han, A. Grange, Y. Xu, S. Parker, C. Chen, H. Su, U. Joshi, C.-H. Chiang *et al.*, "An overview of coding tools in AV1: the first video codec from the alliance for open media," *APSIPA Transactions on Signal and Information Processing*, vol. 9, 2020.

[16] AV2 Is In R&D As The Eventual Successor To The AV1 Video Codec, https://www.phoronix.com/scan.php?page=news_item&px=AV2-Video-Codec-In-Research.

[17] 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.

[18] Audio and Video Coding Standard Work-group of China, http://www.avs.org.cn/english/index.asp.

[19] C. Tian, Y. Xu, L. Fei, and K. Yan, "Deep learning for image denoising: a survey," in *International Conference on Genetic and Evolutionary Computing*. Springer, 2018, pp. 563–572.

[20] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: a survey," *arXiv preprint arXiv:1902.06068*, 2019.

[21] G. Yao, T. Lei, and J. Zhong, "A review of convolutional-neural-network-based action recognition," *Pattern Recognition Letters*, vol. 118, pp. 14–22, 2019.

[22] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: a comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[23] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wanga, "Image and video compression with neural networks: a review," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[24] 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.

[25] 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.

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[27] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[28] 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 (CVPR)*, 2017, pp. 4681–4690.

[29] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "ESRGAN: enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 1–16.

[30] CCITT/SG XV Rec. H.120, "Codecs for videoconferencing using primary group transmission," ITU-T Std., (1989).

[31] ITU-T Rec. H.261, "Video codec for audiovisual services at $p \times 64$ kbit/s," ITU-T Std., (version 1, 1990; version 2, 1993).

[32] ITU-T Rec. H.262 and ISO/IEC 13818–2 (MPEG–2 Video), "Generic coding of moving pictures and associated audio information–part2: Video," ITU-T Std., (version 1, 1994).

[33] ITU-T Rec. H.263, "Video coding for low bitrate communication," ITU-T Std., (version 1, 1995; version 2, 1998; version 3, 2000).

[34] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[35] 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.

[36] S. Liu, E. Alshina, J. Pfaff, M. Wien, P. Wu, and Y. Ye, "Report of AHG11 meeting on neural-network-based video coding on 2020-07-30," in *the JVET meeting*, no. JVET-T0042. Teleconference: ITU-T, ISO/IEC, 2020.

[37] S. Liu, E. Alshina, J. Pfaff, M. Wien, P. Wu, and Y. Ye, "JVET AHG report: neural-network-based video coding," in *the JVET meeting*, no. JVET-T0011-v2. Teleconference: ITU-T, ISO/IEC, 2020.

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

[39] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Suhring, "JVET common test conditions and software reference configurations for SDR video," in *the JVET meeting*, no. JVET-M1001. ITU-T, ISO/IEC, 2019.

[40] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in *13th VCEG Meeting, no. VCEG-M33,Austin, Texas*, 2001, pp. USA: ITU–T.

[41] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep learning for single image super-resolution: a brief review," *IEEE Transactions on Multimedia*, 2019.

[42] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[44] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[45] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[46] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

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

[48] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[49] T. Dozat, "Incorporating nesterov momentum into Adam," in *Proceedings of the International Conference on Learning Representations Workshop (ICLRW)*, 2016.

[50] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[51] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, vol. 12, no. 7, 2011.

[52] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[53] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[54] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*. PMLR, 2015, pp. 448–456.

[55] 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 (CVPR)*, 2016, pp. 770–778.

[56] C. Li, L. Song, R. Xie, and W. Zhang, "CNN based post-processing to improve HEVC," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 4577–4580.

[57] F. Zhang, M. Afonso, and D. R. Bull, "ViSTRA2: video coding using spatial resolution and effective bit depth adaptation," *Signal Processing: Image Communication*, p. 116355, 2021.

[58] 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.

[59] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[60] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7794–7803.

[61] 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.

[62] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 391–407.

[63] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1646–1654.

[64] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3147–3155.

[65] 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 (CVPRW)*, 2017, pp. 136–144.

[66] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2472–2481.

[67] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[68] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.

[69] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 252–268.

[70] J. Cai, S. Gu, R. Timofte, and L. Zhang, "NTIRE 2019 challenge on real image super-resolution: methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1–13.

[71] A. Ignatov and R. Timofte, "NTIRE 2019 challenge on image enhancement: methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1–9.

[72] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5693–5703.

[73] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019, pp. 1–18.

[74] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[75] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1492–1500.

[76] T. Shang, Q. Dai, S. Zhu, T. Yang, and Y. Guo, "Perceptual extreme super-resolution network with receptive field block," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1–10.

[77] J. Cai, Z. Meng, and C. Man Ho, "Residual channel attention generative adversarial network for image super-resolution and noise reduction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 454–455.

[78] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[79] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258.

[80] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, "ResNeSt: Split-attention networks," *arXiv preprint arXiv:2004.08955*, 2020.

[81] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[82] K. Lin, T. H. Li, S. Liu, and G. Li, "Real photographs denoising with noise domain adaptation and attentive generative adversarial network," in *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1–5.

[83] R. Muhammad Umer, G. Luca Foresti, and C. Micheloni, "Deep generative adversarial residual convolutional networks for real-world super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1–9.

[84] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[85] N. Ahn, B. Kang, and K.-A. Sohn, "Photo-realistic image super-resolution with fast and lightweight cascading residual network," *arXiv preprint arXiv:1903.02240*, 2019.

[86] T. Miyato and M. Koyama, "cGANs with projection discriminator," *arXiv preprint arXiv:1802.05637*, 2018.

[87] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," *arXiv preprint arXiv:1807.00734*, 2018.

[88] H. Ren, A. Kheradmand, M. El-Khamy, S. Wang, D. Bai, and J. Lee, "Real-world super-resolution using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1–9.

[89] X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li, and F. Huang, "Real-world super-resolution via kernel estimation and noise injection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1–10.

[90] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125–1134.

[91] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: an overview," *Neural Networks*, 2020.

[92] H. Tokunaga, Y. Teramoto, A. Yoshizawa, and R. Bise, "Adaptive weighting multi-field-of-view CNN for semantic segmentation in pathology," in *Proceedings of*

the IEEE International Conference on Computer Vision (CVPR), 2019, pp. 12 597–12 606.

[93] K. Xu, L. Wen, G. Li, L. Bo, and Q. Huang, "Spatiotemporal CNN for video object segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2019, pp. 1379–1388.

[94] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3156–3164.

[95] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 305–321.

[96] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "Multi-fiber networks for video recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 352–367.

[97] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[98] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6154–6162.

[99] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2017, pp. 4836–4845.

[100] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: multi-object tracking and segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2019, pp. 7942–7951.

[101] H. Xue, C. Liu, F. Wan, J. Jiao, X. Ji, and Q. Ye, "DANet: divergent activation for weakly supervised object localization," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2019, pp. 6589–6598.

[102] J. Choe and H. Shim, "Attention-based dropout layer for weakly supervised object localization," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2019, pp. 2219–2228.

[103] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2019, pp. 2642–2651.

[104] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2019, pp. 3343–3352.

[105] 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 (CVPR)*, 2017, pp. 2462–2470.

[106] P. Liu, M. Lyu, I. King, and J. Xu, "SelFlow: self-supervised learning of optical flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4571–4580.

[107] R. Feng, J. Gu, Y. Qiao, and C. Dong, "Suppressing model overfitting for image super-resolution networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 0–0.

[108] T. Isobe, S. Li, X. Jia, S. Yuan, G. Slabaugh, C. Xu, Y.-L. Li, S. Wang, and Q. Tian, "Video super-resolution with temporal group attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8008–8017.

[109] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2019, pp. 3155–3164.

[110] T. Ehret, A. Davy, J.-M. Morel, G. Facciolo, and P. Arias, "Model-blind video denoising via frame-to-frame training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 369–11 378.

[111] S. Zhou, J. Zhang, J. Pan, H. Xie, W. Zuo, and J. Ren, "Spatio-temporal filter adaptive network for video deblurring," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2019, pp. 2482–2491.

[112] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4463–4471.

[113] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3703–3712.

[114] X. Zhu, C. Vondrick, C. C. Fowlkes, and D. Ramanan, "Do we need more training data?" *International Journal of Computer Vision*, vol. 119, no. 1, pp. 76–92, Aug 2016. [Online]. Available: https://doi.org/10.1007/s11263-015-0812-2

[115] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "TDAN: Temporally deformable alignment network for video super-resolution," *arXiv preprint arXiv:1812.02898*, 2018.

[116] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2, July 2001, pp. 416–423 vol.2.

[117] 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.

[118] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet large scale visual recognition

challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[119] 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 (CVPRW)*, July 2017.

[120] S. Wan, M.-Z. Wang, H. Gong, C.-Y. Zou, Y.-Z. Ma, J.-Y. Huo, and et al., "CE10: integrated in-loop filter based on CNN (Tests 2.1, 2.2 and 2.3)," in *the JVET meeting*, no. JVET-O0079. Gothenburg, Sweden: ITU-T, ISO/IEC, 2019.

[121] T. Hashimoto, E. Sasaki, and T. Ikai, "AHG9: separable convolutional neural network filter with squeeze-and-excitation block," in *the JVET meeting*, no. JVET-K0158. Ljubljana, Slovenia: ITU-T, ISO/IEC, 2018.

[122] Y. Wang, H. Zhu, Y. Li, Z. Chen, and S. Liu, "Dense residual convolutional neural network based in-loop filter for HEVC," in *2018 IEEE Visual Communications and Image Processing (VCIP)*, Dec 2018, pp. 1–4.

[123] 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.

[124] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: a dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[125] R. Szeto, X. Sun, K. Lu, and J. J. Corso, "A temporally-aware interpolation network for video frame inpainting," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[126] Z. Zhang, L. Chen, R. Xie, and L. Song, "Frame interpolation via refined deep voxel flow," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1473–1477.

[127] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The Kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[128] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A short note about Kinetics-600," *arXiv preprint arXiv:1808.01340*, 2018.

[129] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the Kinetics-700 human action dataset," *arXiv preprint arXiv:1907.06987*, 2019.

[130] L. Smaira, J. Carreira, E. Noland, E. Clancy, A. Wu, and A. Zisserman, "A short note on the Kinetics-700-2020 human action dataset," *arXiv preprint arXiv:2010.10864*, 2020.

[131] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[132] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick *et al.*, "Moments in time dataset: one million videos for event understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 502–508, 2019.

[133] Y. Wang, S. Inguva, and B. Adsumilli, "YouTube UGC dataset for video compression research," in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2019, pp. 1–5.

[134] Tencent Video Dataset, https://multimedia.tencent.com/open/tvd.

[135] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, and T. Huang, "Robust video super-resolution with learned temporal dynamics," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2507–2515.

[136] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE transactions on Image Processing*, vol. 19, no. 6, pp. 1427–1441, 2010.

[137] J. Y. Lin, R. Song, C.-H. Wu, T. Liu, H. Wang, and C.-C. J. Kuo, "MCL-V: a streaming video quality assessment database," *Journal of Visual Communication and Image Representation*, vol. 30, pp. 1–9, 2015.

[138] C. Keimel, J. Habigt, T. Habigt, M. Rothbucher, and K. Diepold, "Visual quality of current coding technologies at high definition iptv bitrates," in *2010 IEEE International Workshop on Multimedia Signal Processing*. IEEE, 2010, pp. 390–393.

[139] H. Wang, I. Katsavounidis, J. Zhou, J. Park, S. Lei, X. Zhou, M.-O. Pun, X. Jin, R. Wang, X. Wang *et al.*, "VideoSet: a large-scale compressed video quality dataset based on jnd measurement," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 292–302, 2017.

[140] S. Nah, S. Baik, S. Hong, G. Moon, S. Son, R. Timofte, and K. Mu Lee, "NTIRE 2019 challenge on video deblurring and super-resolution: dataset and study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 0–0.

[141] S. Nah, R. Timofte, S. Baik, S. Hong, G. Moon, S. Son, and K. Mu Lee, "NTIRE 2019 challenge on video deblurring: methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 0–0.

[142] 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.

[143] F. Zhang and D. R. Bull, "A parametric framework for video compression using region-based texture models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1378–1392, 2011.

[144] D. Ma, A. V. Katsenou, and D. R. Bull, "A synthetic video dataset for video compression evaluation," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1094–1098.

[145] A. V. Katsenou, G. Dimitrov, D. Ma, and D. R. Bull, "BVI-SynTex: a synthetic video texture dataset for video compression and quality assessment," *IEEE Transactions on Multimedia*, vol. 23, pp. 26–38, 2020.

[146] 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.

[147] J. Lin, D. Liu, H. Li, and F. Wu, "M-LVC: multiple frames prediction for learned video compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3546–3554.

[148] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017, pp. 1–27.

[149] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 771–10 780.

[150] O. Rippel, S. Nair, C. Lew, S. Branson, A. G. Anderson, and L. Bourdev, "Learned video compression," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 3454–3463.

[151] A. Djelouah, J. Campos, S. Schaub-Meyer, and C. Schroers, "Neural inter-frame compression for video coding," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6421–6429.

[152] E. Agustsson, D. Minnen, N. Johnston, J. Balle, S. J. Hwang, and G. Toderici, "Scale-space flow for end-to-end optimized video compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8503–8512.

[153] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: an end-to-end deep video compression framework," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 006–11 015.

[154] F. Jiang, W. Tao, S. Liu, J. Ren, X. Guo, and D. Zhao, "An end-to-end compression framework based on convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 3007–3018, 2017.

[155] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, "Generative adversarial networks for extreme learned image compression," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 221–231.

[156] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool, "Practical full resolution learned lossless image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 629–10 638.

[157] A. Habibian, T. v. Rozendaal, J. M. Tomczak, and T. S. Cohen, "Video compression with rate-distortion autoencoders," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 7033–7042.

[158] C.-H. Yeh, Z.-T. Zhang, M.-J. Chen, and C.-Y. Lin, "HEVC intra frame coding based on convolutional neural network," *IEEE Access*, vol. 6, pp. 50 087–50 095, 2018.

[159] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, "Fully connected network-based intra prediction for image coding," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3236–3247, 2018.

[160] Z. Zhao, S. Wang, S. Wang, X. Zhang, S. Ma, and J. Yang, "Enhanced bi-prediction with convolutional neural network for High Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[161] L. Zhao, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Enhanced motion-compensated video coding with deep virtual reference frame generation," *IEEE Transactions on Image Processing*, 2019.

[162] S. Puri, S. Lasserre, and P. Le Callet, "CNN-based transform index prediction in multiple transforms framework to assist entropy coding," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 798–802.

[163] S. Jimbo, J. Wang, and Y. Yashima, "Deep learning-based transformation matrix estimation for bidirectional interframe prediction," in *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2018, pp. 726–730.

[164] M. M. Alam, T. D. Nguyen, M. T. Hagan, and D. M. Chandler, "A perceptual quantization strategy for HEVC based on a convolutional neural network trained on natural images," in *Applications of Digital Image Processing XXXVIII*, vol. 9599. International Society for Optics and Photonics, 2015, p. 959918.

[165] R. Song, D. Liu, H. Li, and F. Wu, "Neural network-based arithmetic coding of intra prediction modes in HEVC," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.

[166] C. Ma, D. Liu, X. Peng, and F. Wu, "Convolutional neural network-based arithmetic coding of DC coefficients for HEVC intra coding," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1772–1776.

[167] 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*, 2019.

[168] J. Lin, D. Liu, H. Yang, H. Li, and F. Wu, "Convolutional neural network-based block up-sampling for HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[169] S. Liu, A. Segall, E. Alshina, J. Boyce, M. Wien, and D. Grois, "Methodology and reporting template for neural-network-based coding tool testing," in *the JVET meeting*, no. JVET-T0041-v4. Teleconference: ITU-T, ISO/IEC, 2020.

[170] W. Lin, X. He, X. Han, D. Liu, J. See, J. Zou, H. Xiong, and F. Wu, "Partition-aware adaptive switching neural networks for post-processing in HEVC," *IEEE Transactions on Multimedia*, 2019.

[171] 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.

[172] Y.-L. Hsiao, O. Chubach, C.-Y. Chen, T.-D. Chuang, C.-W. Hsu, Y.-W. Huang, and et al., "CE10-1.2: convolutional neural network loop filter," in *the JVET meeting*, no. JVET-O0056. Gothenburg, Sweden: ITU-T, ISO/IEC, July 2019.

[173] M. Afonso, F. Zhang, and D. R. Bull, "Video compression based on spatio-temporal resolution adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 275–280, 2019.

[174] W. Wang, R. Guo, Y. Tian, and W. Yang, "CFSNet: toward a controllable feature space for image restoration," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 4140–4149.

[175] F. Zhang, F. M. Moss, R. Baddeley, and D. R. Bull, "BVI-HD: a video quality database for HEVC compressed and texture synthesized content," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2620–2630, 2018.

[176] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.

[177] L.-H. Chen, C. G. Bampis, Z. Li, A. Norkin, and A. C. Bovik, "Perceptually optimizing deep image compression," *arXiv preprint arXiv:2007.02711*, 2020.

[178] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[179] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[180] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 214–223.

[181] 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.

[182] C. G. Bampis, Z. Li, and A. C. Bovik, "Spatiotemporal feature integration and model fusion for full reference video quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[183] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[184] D. M. Chandler and S. S. Hemami, "VSNR: a wavelet-based visual signal-to-noise ratio for natural images," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2284–2298, 2007.

[185] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Transactions on image processing*, vol. 15, no. 2, pp. 430–444, 2006.

[186] F. Zhang and D. R. Bull, "A perception-based hybrid model for video quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 6, pp. 1017–1028, 2015.

[187] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, 2004.

[188] K. Seshadrinathan and A. C. Bovik, "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 335–350, 2010.

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

[190] S. Li, F. Zhang, L. Ma, and K. N. Ngan, "Image quality assessment by separately evaluating detail losses and additive impairments," *IEEE Transactions on Multimedia*, vol. 13, no. 5, pp. 935–949, 2011.

[191] ITU-R, "Methodology for the subjective assessment of the quality of television pictures," in *ITU-R Standard BT.500-11*. ITU-R, 2002.

[192] F. M. Moss, K. Wang, F. Zhang, R. Baddeley, and D. R. Bull, "On the optimal presentation duration for subjective video quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 11, pp. 1977–1987, 2015.

[193] Videvo Free Stock Video Footage, https://www.videvo.net/.

[194] IRIS32FREE 4K UHD FREE FOOTAGE, https://www.youtube.com/channel/UCjJee-JAzoRRH5T0wqe7_tw/.

[195] Harmonic Inc 4K demo footage, http://www.harmonicinc.com/4k-demo-footage-download/, (Accessed: 1st May 2017).

[196] M. A. Papadopoulos, F. Zhang, D. Agrafiotis, and D. R. Bull, "A video texture database for perceptual compression and quality assessment," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 2781–2785.

[197] 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.

[198] A. Mackin, F. Zhang, and D. R. Bull, "A study of high frame rate video formats," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1499–1512, 2018.

[199] 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.

[200] C. G. Bampis, Z. Li, A. K. Moorthy, I. Katsavounidis, A. Aaron, and A. C. Bovik, "Study of temporal effects on subjective video quality of experience," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5217–5231, 2017.

[201] C. G. Bampis, Z. Li, I. Katsavounidis, T.-Y. Huang, C. Ekanadham, and A. C. Bovik, "Towards perceptually optimized end-to-end adaptive video streaming," *arXiv preprint arXiv:1808.03898*, 2018.

[202] Mitch Martinez FREE 4K STOCK FOOTAGE, http://mitchmartinez.com/free-4k-red-epic-stock-footage/.

[203] Dareful-Completely Free 4K Stock Video, https://www.dareful.com/.

[204] H. Wang, W. Gan, S. Hu, J. Y. Lin, L. Jin, L. Song, P. Wang, I. Katsavounidis, A. Aaron, and C.-C. J. Kuo, "MCL-JCV: a JND-based H.264/AVC video quality assessment dataset," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1509–1513.

[205] I. Katsavounidis, "Chimera video sequence details and scenes," tech. rep., Netflix, (November 2015).

[206] C. Keimel, A. Redl, and K. Diepold, "The TUM high definition video datasets," in *2012 Fourth International Workshop on Quality of Multimedia Experience*. IEEE, 2012, pp. 97–102.

[207] Ultra Video Group, http://ultravideo.cs.tut.fi/#testsequences/.

[208] D. R. Bull, F. Zhang, and M. Afonso, "Description of SDR video coding technology proposal by University of Bristol," in *the JVET meeting*, no. JVET-J0031. San Diego, US: ITU-T, ISO/IEC, April 2018.

[209] M. Afonso, F. Zhang, A. Katsenou, D. Agrafiotis, and D. R. Bull, "Low complexity video coding based on spatial resolution adaptation," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3011–3015.

[210] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "EDVR: video restoration with enhanced deformable convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1–10.

[211] BlueCrystal Phase 4, https://www.acrc.bris.ac.uk/protected/bc4-docs/.

[212] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 065–11 074.

[213] F. Zhang, M. Afonso, and D. R. Bull, "Enhanced video compression based on effective bit depth adaptation," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1720–1724.

[214] Y. Li and S. Liu, "BoG report on neural networks for video coding," in *the JVET meeting*, no. JVET-M0904. Marrakech, Morocco: ITU-T, ISO/IEC, Jan 2019.

[215] H. Yin, R. Yang, X. Fang, and S. Ma, "CE10-1.7: Adaptive convolutional neural network loop filter," in *the JVET meeting*, no. JVET-O0063. Gothenburg, Sweden: ITU-T, ISO/IEC, July 2019.

[216] Y. Wang, H. Zhu, Y. Li, Z. Chen, and S. Liu, "Dense residual convolutional neural network based in-loop filter for HEVC," in *2018 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2018, pp. 1–4.

[217] 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.

[218] L. Ma, Y. Tian, P. Xing, and T. Huang, "Residual-based post-processing for HEVC," *IEEE MultiMedia*, vol. 26, no. 4, pp. 67–79, 2019.

[219] X. Zhang, R. Xiong, W. Lin, J. Zhang, S. Wang, S. Ma, and W. Gao, "Low-rank-based nonlocal adaptive loop filter for high-efficiency video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 10, pp. 2177–2188, 2016.

[220] S. Wan, M. Wang, Y. Ma, J. Huo, H. Gong, C. Zou, and et al., "CE10: integrated in-loop filter based on cnn (Tests 2.1, 2.2 and 2.3)," in *the JVET meeting*, no. JVET-O0079. Gothenburg, Sweden: ITU-T, ISO/IEC, July 2019.

[221] Y. Kidani, K. Kawamura, K. Unno, and S. Naito, "CE10-1.10/CE10-1.11: evaluation results of CNN-based filtering with on-line learning model," in *the JVET meeting*, no. JVET-O0131. Gothenburg, Sweden: ITU-T, ISO/IEC, July 2019.

[222] Y. Kidani, K. Kawamura, K. Unno, and S. Naito, "CE10-2.10/CE10-2.11: evaluation results of CNN-based filtering with off-line learning model," in *the JVET meeting*, no. JVET-O0132. Gothenburg, Sweden: ITU-T, ISO/IEC, July 2019.

[223] Y. Wang, T. Ouyang, C. Zou, Y. Li, and Z. Chen, "CE10: dense residual convolutional neural network based in-loop filter (Tests 2.5 and 2.7)," in *the JVET meeting*, no. JVET-O0101-v2. Gothenburg, Sweden: ITU-T, ISO/IEC, July 2019.

[224] 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.

[225] M. Afonso, F. Zhang, and D. R. Bull, "Spatial resolution adaptation framework for video compression," in *Applications of Digital Image Processing XLI*, vol. 10752. International Society for Optics and Photonics, 2018, p. 107520L.

[226] D. Misra, "Mish: a self regularized non-monotonic neural activation function," *arXiv preprint arXiv:1908.08681*, 2019.

[227] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: more features from cheap operations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1580–1589.

[228] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Deep universal generative adversarial compression artifact removal," *IEEE Transactions on Multimedia*, 2019.

[229] R. Muhammad Umer, G. Luca Foresti, and C. Micheloni, "Deep generative adversarial residual convolutional networks for real-world super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 438–439.

[230] W. Rudin *et al.*, *Principles of mathematical analysis*. McGraw-hill New York, 1964, vol. 3.

[231] D. C. Howell, *Statistical methods for psychology*. Cengage Learning, 2012.

[232] Netflix Public Dataset, https://github.com/Netflix/vmaf/blob/master/resource/doc/datasets.md.

[233] A. V. Katsenou, F. Zhang, M. Afonso, and D. R. Bull, "A subjective comparison of AV1 and HEVC for adaptive video streaming," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 4145–4149.

[234] JCT-VC, "SHVC verification test results," in *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG*. ITU-T SG16 WP3, ISO/IEC JTC1/SC29/WG, 2016.

[235] IVP Subjective Quality Video Database, http://ivp.ee.cuhk.edu.hk/research/database/subjective/.

[236] VQEG-HD3, https://www.its.bldrdoc.gov/vqeg/projects/hdtv/hdtv.aspx.

[237] S. W. Park and J. Kwon, "Sphere generative adversarial network based on geometric moment matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4292–4301.

[238] S. W. Park and J. Kwon, "SphereGAN: Sphere generative adversarial network based on geometric moment matching and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[239] C. Villani, *Optimal transport: old and new*. Springer Berlin Heidelberg, 2008, vol. 338.

[240] Y. Li, D. Liu, H. Li, L. Li, Z. Li, and F. Wu, "Learning a convolutional neural network for image compact-resolution," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1092–1107, 2018.

[241] F. Zhang and D. R. Bull, "Rate-distortion optimization using adaptive lagrange multipliers," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3121–3131, Oct 2019.

[242] F. Bossen, J. Boyce, X. Li *et al.*, "JVET common test conditions and software reference configurations for SDR video," *Joint Video Experts Team (JVET) of ITU-T SG*, vol. 16, 2018.

[243] A. Mackin, M. Afonso, F. Zhang, and D. R. Bull, "SRQM: a video quality metric for spatial resolution adaptation," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 283–287.

[244] 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.

[245] F. Zhang, M. Afonso, and D. R. Bull, "Enhanced video compression based on effective bit depth adaptation," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1720–1724.

[246] BlueCrystal Phase 3, https://www.acrc.bris.ac.uk/acrc/phase3.htm/.