DCU

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

SCHOOL OF ELECTRONIC ENGINEERING

Colour Technologies for Content Production and Distribution of Broadcast Content

Marc Górriz Blanch, BSc, MSc

A Dissertation submitted in fulfillment of the requirements
for the award of Doctor of Philosophy (PhD)

Supervised by

Prof. Noel E. O'Connor (Dublin City University)
Dr. Marta Mrak (BBC)

January 2023

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Marc Gorriz Blanch Student number: 19211237
Date: January 2023

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisors, Dr. Marta Mrak and Prof. Noel O'Connor, for the opportunity to start this life-changing project and support me until the last day. Thank you for trusting me with this position and accepting me in the BBC team from the first day. Thank you, Marta, for your advice, guidance, and patience, and for making me the researcher I am today. Thank you, Noel, for your excellent contributions and continuous support during all these years.

I also want to thank the amazing people I met on this project. Thanks to my colleagues at the BBC, especially my teammates, Luka, Saverio, Andre, Gosala, Maria, Fiona and Rajitha; my fellow ESRs and JOLT colleagues, to Prof. Alan from DCU and all those who to a lesser or greater extent contributed to this work and to my development as a researcher. As a crucial collaborator in my research life, I would like to thank Xavi Giro-i-Nieto for sharing his passion for deep learning and for introducing me to this fascinating research. Thank you for all the opportunities and advice over the years, and for transforming that undergraduate student into the researcher I am today.

From the bottom of my heart, this project gave me the opportunity to meet my wife, Gabi. I cannot express in words how grateful I am for your love, your unconditional support and having you every day of my life. I love you bubu!

To my parents, Roberto and Fina, my sister Nuria, my iaios Josep and Isabel and the whole family: you have always encouraged me and pushed me to overcome all the challenges of my life. Thank you for being patient and supporting me from the distance all these years that I have been abroad. Thanks to my friends, Fran, Boyi, Gimena, Sergi and Isaac, for always being there; and special thanks to my family in Brazil who welcomed me like one of their own from the first day.

Finally, I would like to thank European Union's Horizon 2020 Marie Sklodowska Curie JOLT project for the funding to this research.

Seems unreal, but yes, we have finished now! To all of you, believing in me, made this thesis possible, thank you.

List of Publications

Main publications:

- **Marc Gorriz Blanch**, Marta Mrak, Alan F. Smeaton and Noel E. O'Connor. "End-to-End Conditional GAN-based Architectures for Image Colourisation". In IEEE International Workshop on Multimedia Signal Processing (MMSP). September 2019.

- **Marc Gorriz Blanch**, Saverio Blasi, Alan F. Smeaton and Noel E. O'Connor, Marta Mrak. "Chroma Intra Prediction with attention-based CNN architectures". In IEEE International Conference on Image Processing (ICIP). October 2020.

- Luka Murn, **Marc Gorriz Blanch**, Maria Santamaria, Fiona Rivera and Marta Mrak. "Towards Transparent Application of Machine Learning in Video Processing". In International Broadcasting Convention (IBC), 2020.

- **Marc Gorriz Blanch**, Saverio Blasi, Alan F. Smeaton and Noel E. O'Connor and Marta Mrak. "Attention-Based Neural Networks for Chroma Intra Prediction in Video Coding," in IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 2, pp. 366-377, Feb. 2021.

- **Marc Gorriz Blanch**, Issa Khalifeh, Alan F. Smeaton, Marta Mrak and Noel E. O'Connor. "Attention-based Stylisation for Exemplar Image Colourisation". In IEEE International Workshop on Multimedia Signal Processing (MMSP), October 2021.

- **Marc Gorriz Blanch**, Noel E. O'Connor and Marta Mrak. "Scene-adaptive Temporal Stabilisation for Video Colourisation using Deep Video Priors". In ACM International Conference on Multimedia (ACMMM). October 2022.

Other publications:

- Chengyi Zou, Shuai Wan, Tiannan Ji, Marta Mrak, **Marc Gorriz Blanch** and Luis Herranz. "Spatial Information Refinement for Chroma Intra Prediction in Video Coding". In IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). December 2021.

- Saiping Zhang, Marta Mrak, Luis Herranz, **Marc Górriz Blanch**, Shuai Wan and Fuzheng Yang. "DVC-P: Deep Video Compression with Perceptual Optimizations". In IEEE International Conference on Visual Communications and Image Processing (VCIP). December 2021.
- Saiping Zhang, Luis Herranz, Marta Mrak, **Marc Gorriz Blanch**, Shuai Wan and Fuzheng Yang. "DCNGAN: A Deformable Convolutional-Based GAN with QP Adaptation for Perceptual Quality Enhancement of Compressed Video". In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). May 2022.

Contents

1 Introduction **1**

1.1 Chapter overview 1

1.2 Perception of colour 1

1.3 Colour technologies in broadcasting 2

1.4 Motivation 4

1.5 Hypothesis and research questions 4

1.6 Thesis outline 5

2 Background concepts **7**

2.1 Chapter overview 7

2.2 Digital broadcasting workflow 7

2.2.1 AI trends in broadcasting 8

2.3 Basics of colour vision 9

2.4 Deep learning in computer vision 10

2.4.1 Training the model 11

2.4.2 Deep generative models 12

2.4.3 Attention models 13

I Content production: image and video colourisation 15

3 Fully automatic colourisation **16**

3.1 Chapter overview 16

3.2 Introduction 16

3.3 Related Work 18

3.4 Proposed Method 19

3.4.1 Conditional Adversarial Networks 20

3.4.2 Mini-batch normalisation 22

3.4.3 Weight regularisation 23

3.4.4 Multi-scale discriminators 23

3.5 Experiments 24

3.5.1 Data 24

3.5.2 Network architectures 24

3.5.3 Quantitative evaluation 25

3.6 Future work 27

3.7 Conclusions 28

4 Exemplar-based colourisation 29
4.1 Chapter overview . . . 29
4.2 Introduction . . . 29
4.3 Related work . . . 31
4.4 Proposed method . . . 32
4.4.1 Exemplar-based Colourisation Network . . . 33
4.4.2 Axial attention for unsupervised style transfer . . . 34
4.4.3 Training losses . . . 37
4.5 Experiments . . . 39
4.5.1 Training settings . . . 39
4.5.2 Comparison with colourisation methods . . . 41
4.5.3 Ablation study . . . 44
4.6 Future work . . . 45
4.7 Conclusions . . . 46

5 A pipeline for video colourisation 47
5.1 Chapter overview . . . 47
5.2 Introduction . . . 47
5.3 Related work . . . 49
5.3.1 Video colourisation . . . 49
5.3.2 Deep Video Prior . . . 49
5.3.3 Few-shot learning . . . 50
5.4 Method . . . 50
5.4.1 Extension of DVP to multiple scenes . . . 50
5.4.2 Few-shot training strategy . . . 52
5.4.3 Network architecture . . . 53
5.5 Experiments . . . 54
5.5.1 Training strategy . . . 54
5.5.2 Evaluation metrics . . . 56
5.5.3 Results . . . 56
5.5.4 Ablations . . . 57
5.6 A pipeline for video colourisation . . . 58
5.7 Conclusions . . . 60

II Content distribution: video compression 64

6 Neural networks for chroma intra prediction 65
6.1 Chapter overview . . . 65
6.2 Introduction . . . 65
6.3 Related Work . . . 67
6.4 Proposed method . . . 68
6.4.1 Cross-component boundary branch . . . 70
6.4.2 Luma convolutional branch . . . 70
6.4.3 Attention-based fusion module . . . 70
6.4.4 Prediction head branch . . . 71
6.5 Experiments . . . 72
6.5.1 Training settings . . . 72
6.5.2 Integration into VVC . . . 72

6.5.3 Results 73
6.6 Future work 74
6.7 Conclusions 75

7 Multi-model architectures and simplifications 76
7.1 Chapter overview 76
7.2 Introduction 76
7.3 Related work 78
7.4 Multi-model architectures 79
7.4.1 Multi-model size agnostic architecture 79
7.4.2 Simplified convolutions 82
7.4.3 Simplified cross-component boundary branch 84
7.4.4 Integer precision approximation 86
7.5 Experiments 89
7.5.1 Architecture configurations 89
7.5.2 Simulation Results 93
7.6 Future work 96
7.7 Conclusion 97

8 Conclusions 99
8.1 Hypothesis and research questions 100
8.2 Research contributions and proposed solutions 104
8.3 Recommendations and future work 105
8.4 Closing remarks 108

List of Figures

1.1 Colour information is essential to understand an object's properties. In this example, the gradient between green and red represents the maturity process of the coffee cherries, indicating the right moment to collect their seeds. 2

1.2 A colour television test at the Mount Kaukau transmitting station, New Zealand, in 1970. A test pattern with colour bars is sometimes used when no program material is available. 3

2.1 Broadcasting workflow for an integrated newsroom (modified from [1]). 8

2.2 a) human cone cell responses (S-hort, M-edium, and L-ong types) to monochromatic spectral stimuli; b) CIE $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ matching functions. 10

2.3 Example of CNN architecture for image classification. 12

3.1 Examples of desaturated results obtained with colourisation approaches based on regression. 17

3.2 Peter Jackson's World War I documentary, They Shall Not Grow Old. A successful colourisation example in a real broadcasting and restoration scenario. Computer-assisted technologies were used to colourise original footages during the conflict, provided by BBC Archives and the Imperial War Museum (IWM). 19

3.3 Proposed generator and discriminator architectures with IBN adaptation to the U-Net and PatchGAN architectures. Note the fake input to the discriminator is composed by the concatenation $(1 + 2)$ of the original greyscale input and the generated 2-dimensional colour channels. 20

3.4 Conditional GAN framework in [2] applied to image colourisation. . . 21

3.5 Comparison between mini-batch normalisation techniques. The volumes represent a batch of N samples with $H \times W \times C$ dimensions. Source from Group Normalisation paper [3]. 22

3.6 Learning curves of the generator during training. 26

3.7 Comparison of colour histograms over the test data. 27

3.8 Visual comparison of colourisation techniques. Note the improvement of the proposed configurations over the BL model: (1) the gain in colourfulness after applying SN, (2) the localisation improvement of MD and (3) the benefits of IBN architecture rather than applying BN and IN separately. 28

4.1 Visualisation of supervised colourisation methods, including scribble-based results from [4] and exemplar-based results from [5]. 31

4.2 XCNET: proposed architecture for exemplar-based image colourisation. T is the black and white frame with luma component only, and R is colour reference. Multi-scale outputs $P1$-$P4$ are used for training, where $P1$ are the colourised image components at targeted resolution. 33

4.3 Axial attention module described in Section 4.4.2. In the figure, BN means Batch Normalisation and A activation. 35

4.4 Visualisation of the axial-attention masks in the target image for a given row and column in the reference image. Note that column attention expands the input row-dependent activated space into a 2D lattice. 36

4.5 Recommendation pipeline in [5] used to extract colour references from Imagenet dataset. In case the target class is known, the classification step is avoided and the global ranking starts over the given class subset. Note that the method in [5] uses a global ranking of 200 images whilst in this work a top-5 ranking is considered. 40

4.6 Qualitative comparison of the existing and the proposed exemplar-based colourisation methods. 41

4.7 Visualisation of the user interface developed in this project for collecting the subjective test results. 43

4.8 Runtime comparison in seconds. 44

4.9 Visual comparison of each individual training loss contribution. . . . 45

4.10 Visual comparison of the attention module configurations, using standard attention or axial attention one time or two times. 45

5.1 Proposed framework for temporal stabilisation of frame-to-frame colourised videos. In addition to the DVP baseline, a scene segmentation and a few show training is used to learn scene-aware video priors. 51

5.2 Proposed architecture for stabilisation of frame-to-frame colourised videos. The model not only learns to stabilise an input sequence, but also to cluster the input frames into different scenes, by generating a class distribution vector y_t. 53

5.3 The proposed decoder block conditioned by the scene-wise embedding m_t^b. Notice that a similar architecture applies to the bottleneck, injecting the embedding vector into the pre-activations. 54

5.4 Example of mode collapse during initial pre-training. The AUC invariance during the initial iterations indicates convergence to a random state, which affects the performance of the posterior IRT training. 55

5.5 Evaluation in comparison with DVP method and processed frames at different timestamps. 57

5.6 Comparison of processing time for all test sequences. Notice the significant increase of DVP when the number of frames increase. 59

5.7 Proposed pipeline for exemplar-based video colourisation combining the image-based XCNET architecture (from Chapter 4) with the temporal stabilisation framework proposed in this chapter. 60

5.8 Evaluation in comparison with DVP method and processed frames at different timestamps. 61

5.9 Evaluation in comparison with DVP method and processed frames at different timestamps. 62

6.1 Visualisation of the attentive prediction process. The first image shows the original and predicted blocks and their respective reference samples, and the second image the corresponding attention masks for each boundary location. For each reference sample 0-16 the attention module generates its contribution to the prediction of individual pixels from a target 4×4 block. 66

6.2 General block diagram of VVC encoder. 67

6.3 Proposed architecture including the attention module used to fuse the output of the two first network branches. 69

6.4 Attention visualisation when predicting a block for different temperature T values. Notice the increasing attention sparsity when T decreases. Axes y represents the N^2 block locations and axes y the B positions. 72

7.1 Proposed multi-model attention-based architectures with the integration of the simplifications introduced in this paper. Notice that Scheme 1 adopts the multi-model architecture (Section 7.4.1) applying the methodology in Section 7.4.2 to simplify the convolutional layers within the luma convolutional branch and the prediction branch, and Scheme 2 further simplifies Scheme 1 by applying the methodology in Section 7.4.3 to simplify the cross-component boundary branch. More details about the model's hyperparameters and a description of the referred schemes can be found in Section 7.5. 80

7.2 Illustration of the proposed multi-model training and inference methodologies. Multiple block-dependent models $\theta_N(W^{(t)})$ are used during training time. A size-agnostic model with a single set of trained weighs W is then used during inference. 81

7.3 Training algorithm for the proposed multi-model architecture. 82

7.4 Visualisation of the receptive field of a 2-layer convolutional branch with 3×3 kernels. Observe that an output pixel in layer 2 is computed by applying a 3×3 kernel over a field F_1 of 3×3 samples from the first layer's output space. Similarly, each of the F_1 values are computed by means of another 3×3 kernel looking at a field F_0 of 5×5 samples over the input. 83

7.5 Visualisation of the learnt colour space resulting of encoding input YCbCr colours to the 3-dimensional hidden space of the autoencoder. 84

7.6 Comparison of logarithmic colour histograms for different sequences. . 94

List of Tables

3.1 Quantitative evaluation of colourisation techniques 25

4.1 Discriminator architecture 39
4.2 Quantitative comparison of the state-of-the art methods with the proposed method in different settings. Note that standard attention is only used in the ablation study, the rest of our combinations use axial attention. 42
4.3 Perceptual test results. The values show the percentage of images selected as genuine (natural) for each of the methods. 43

5.1 E_{hist}, F_{data} and FID comparison for different colourisation methods. . 58

6.1 Model hyperparameters per block size 73
6.2 BD-rate results anchoring to VTM-7.0 74
6.3 BD-rate results for constrained test 74
6.4 Prediction performance evaluation (PSNR) 75

7.1 Network hyperparameters during training 90
7.2 Network hyperparameters during inference 90
7.3 FLOPs per block size 91
7.4 Model complexity per block size 91
7.5 Prediction performance per block size 91
7.6 BD-Rates (%) sorted by Gini index 94
7.7 BD-Rate (%) for all proposed schemes and [6] under all-intra CTC . . 96

Abstract

"Colour Technologies for Content Production and Distribution of Broadcast Content"

Marc Górriz Blanch

The requirement of colour reproduction has long been a priority driving the development of new colour imaging systems that maximise human perceptual plausibility. This thesis explores machine learning algorithms for colour processing to assist both content production and distribution. First, this research studies colourisation technologies with practical use cases in restoration and processing of archived content. The research targets practical deployable solutions, developing a cost-effective pipeline which integrates the activity of the producer into the processing workflow. In particular, a fully automatic image colourisation paradigm using Conditional GANs is proposed to improve content generalisation and colourfulness of existing baselines. Moreover, a more conservative solution is considered by providing references to guide the system towards more accurate colour predictions. A fast-end-to-end architecture is proposed to improve existing exemplar-based image colourisation methods while decreasing the complexity and runtime. Finally, the proposed image-based methods are integrated into a video colourisation pipeline. A general framework is proposed to reduce the generation of temporal flickering or propagation of errors when such methods are applied frame-to-frame. The proposed model is jointly trained to stabilise the input video and to cluster their frames with the aim of learning scene-specific modes. Second, this research explored colour processing technologies for content distribution with the aim to effectively deliver the processed content to the broad audience. In particular, video compression is tackled by introducing a novel methodology for chroma intra prediction based on attention models. Although the proposed architecture helped to gain control over the reference samples and better understand the prediction process, the complexity of the underlying neural network significantly increased the encoding and decoding time. Therefore, aiming at efficient deployment within the latest video coding standards, this work also focused on the simplification of the proposed architecture to obtain a more compact and explainable model.

Chapter 1

Introduction

1.1 Chapter overview

This chapter provides a general introduction to this thesis, presents the motivation for this research, the hypotheses and research questions, and outlines the structure of the document. Section 1.2 showcases the importance of colour perception in computer vision, Section 1.3 analyses the applications of colour processing technologies in the broadcasting industry, Section 1.4 discusses the specific motivations underpinning this research. Finally, Section 1.5 presents the hypothesis and research questions and Section 1.6 outlines the organisation and structure of this thesis.

1.2 Perception of colour

The perception of colour is essential in modern image and video processing systems, playing an important role in many applications and domains, such as multimedia systems, computer vision, broadcasting and digital imaging. Over the years, image processing methodologies have primarily focused on grayscale images, treating colours as a dimensional extension of the intensity component or luminance. However, a better understanding of the colour components has been important to many image processing and pattern recognition systems, where colour-based features provide a richer level of abstraction towards solving complex computer vision tasks such as image retrieval, segmentation or classification.

While the luminance information provides valuable content-related information regarding shapes and structures, the perception of colour is essential for understanding the visual world, allowing the distinction between objects and physical variations, such as shadow gradations, light source reflections or reflectance variations. For example, as shown in Figure 1.1, the colour information is key to understand the level of maturity of the coffee beans. For this reason, the processing of chromatic

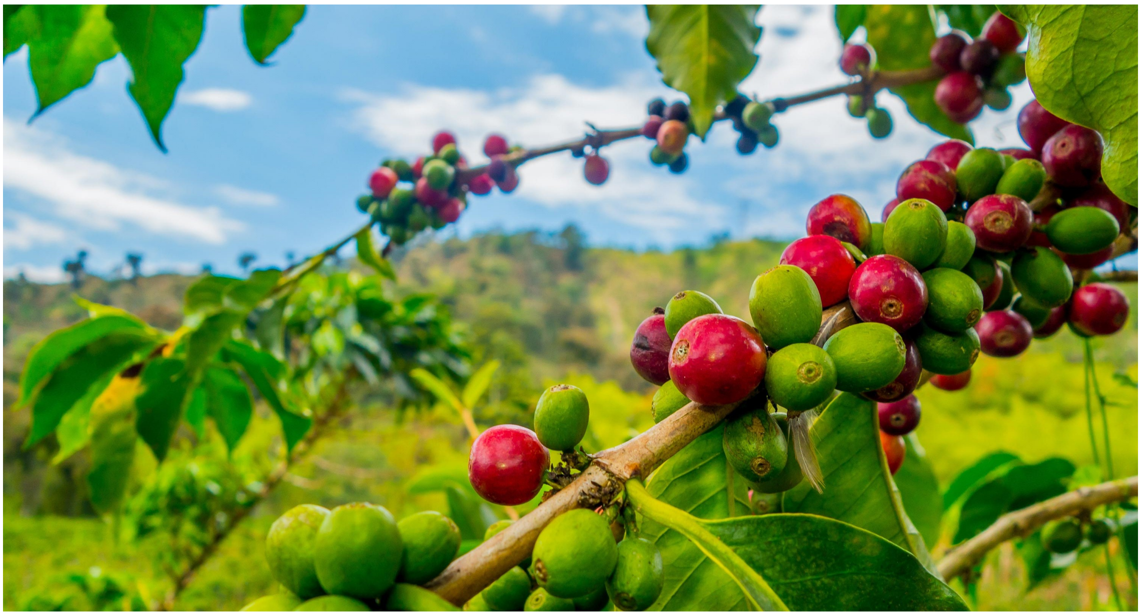


Figure 1.1: Colour information is essential to understand an object's properties. In this example, the gradient between green and red represents the maturity process of the coffee cherries, indicating the right moment to collect their seeds.

information and the improvement of colour quality have become a research area of significant interest for a wide variety of domains that traditionally have prioritised the use of luminance data alone. This includes imaging systems for medicine, meteorology or quality control for a wide range of industries.

1.3 Colour technologies in broadcasting

Colour technologies have long been the heart of the broadcasting industry. Back in the sixties, the invention of the colour television represented a total revolution, and one of the most complex and transformative technological innovations in the history of the world. As shown in Figure 1.2, everyone will remember the famous test pattern with colour bars that was sometimes used when no program material was available. Colour systems granted a unique and thoroughly modern form of seeing and representing the world, bringing to the viewers a more legible, realistic and emotive experience. Since then, broadcasting technologies have been in constant evolution playing a key roles in digital transformation. The convergence of digital entertainment required the replacement of deprecated analogical systems and the development of new processing and transmission methodologies which ensure more efficiency and flexibility. The requirement of colour reproduction has been a priority all along the process, and the design of new colour imaging systems has focused on maximising human perceptual plausibility. Digitisation has also driven the establishment of new computer vision disciplines, such as video enhancement

Figure 1.2: A colour television test at the Mount Kaukau transmitting station, New Zealand, in 1970. A test pattern with colour bars is sometimes used when no program material is available.

and restoration, which have had a significant impact on the broadcasting industry in recovering and colourising degraded content from legacy archives.

The proliferation of imaging devices followed by the rise of user-generated content, has prompted huge interest in the control and enhancement of video quality and colour fidelity. The variety in the image capturing and generation sources is being mirrored by an increasing diversity of the media for which colour images are reproduced. Consequently, colour imaging technologies have become essential in managing the reproduction of colour in a consistent and systematic way, such that the visual appearance remains perceptually constant. Another important element in the broadcasting chain are technologies for distribution. Video compression has become an essential asset for tackling the increasing demand for higher quality video content. Colour prediction plays an increasingly important role in video coding, proven to be effective in achieving better compression rates by means of exploiting cross-component information. Finally, Artificial Intelligence (AI) is increasingly creating disruption and innovation. Broadcasting applications driven by machine learning have rapidly shifted from research environments to deployed scenarios, enabling the automation of many production workflows. As recently reported by the International Telecommunication Union (ITU) [7], such technologies are providing improvements in production efficiency and correlated cost reduction, as well as optimised content delivery at lower bandwidth.

1.4 Motivation

This research investigates extending existing AI approaches for application to colour processing in video broadcast systems. Video technologies are essential in the digital broadcasting and production workflow, especially for content production and distribution. Video technologies include multimedia content processing, video compression, video enhancement and data analysis.

In the context of content production, this research focuses on automatic colourisation to be applied in video enhancement and restoration applications. Archived broadcast content such as previously broadcasted shows, and historical event coverage is of relevance to journalism to support incoming news stories, and provide context to current events. In particular, colourisation refers to the process of adding colours to greyscale content such that the results are perceptually and visually appealing. Therefore, the production of automatically colourised content does not seek to reproduce reality, rather to generate realistic and semantically meaningful content. This research targets practical deployable solutions, developing a cost-effective pipeline which integrates the activity of the user/producer into the processing workflow, together with automatic methodologies based on style transfer, deep image analogy computation and smart retrieval.

In parallel, technologies for content distribution are essential to effectively deliver the processed content to a broad audience. Colour prediction extends directly to research disciplines such as video compression. Specifically, within intra frame coding, chroma components are predicted with the aim to remove correlations within local regions of the image and the cross-component domain. This research investigates the use of neural networks for improving existent chroma prediction methods, aiming an efficient deployment within the latest video coding standards. Finally, explainability of neural networks is considered to verify the predictions of complex end-to-end models as well as to identify potential simplifications towards more efficient implementations.

1.5 Hypothesis and research questions

In this research, it is hypothesised that *AI-based colour processing techniques can be efficiently implemented into a current digital broadcasting workflow with a significant impact on video production and distribution.* The following research questions drive the research performed to investigate the proposed hypothesis:

1. Content production: image and video colourisation.
    - Can deep learning models for video colourisation be implemented into

a production pipeline for content enhancement and restoration? Some areas of interest include reference-based methodologies, user/producer interactivity and efficient implementations for image and video prediction.

- Which are the best practices to train a colourisation generative model to ensure content generalisation, colourfulness, temporal consistency and high-resolution predictions?
- Can style transfer approaches be integrated into an end-to-end video colourisation pipeline in order to perform exemplar-based colourisation? If so, which are the best practices to efficiently select appropriate references?

2. Content distribution: video compression and explainability.

   - Can chroma intra-prediction methods be improved by means of neural networks and be efficiently implemented into existing video coding standards?
   - Can deep learning models be designed and/or simplified by means of explainability in order to obtain efficient and cost-effective implementations?

1.6 Thesis outline

Given the nature and structure of the hypothesis, the chapters of this thesis are organised into two main areas of research: content production and content distribution.

Chapter 2 This chapter provides a comprehensive overview of the main background concepts and principles used in the following chapters. The first part provides a review of the use of AI in the broadcasting industry. A general workflow is presented, introducing the main production stages and areas this research focuses on. The common methods observed across the various task areas are then identified in order to find the overall AI trends within the digital broadcasting workflow. The second part provides the basics of colour vision, introducing the concept of colour spaces and its application in image and video processing. Finally, the third part describes deep learning as the current state-of-the-art approach to computer vision applications and gives a brief introduction to its basic principles.

The chapters regarding content production with image and video colourisation are organised as follows.

Chapter 3 introduces the challenges of fully automatic colourisation and proposes a solution applied to images. This chapter investigates the use of generative adversarial networks (GANs) to ensure content generalisation and colourfulness, facing the intrinsic ambiguity of such colourisation methodologies.

Chapter 4 proposes a more conservative approach for images that enables the use of colour references to guide the colour assignment process. This chapter introduces XCNET, a novel exemplar-based neural network for image colourisation that achieves fast and high-quality colour predictions. The proposed architecture integrates attention modules that learn how to extract and transfer style features from the reference images in an unsupervised way during the colourisation process.

Chapter 5 studies how to integrate the proposed image-based XCNET architecture into a video colourisation pipeline and identifies the main drawbacks and challenges. Finally, the research investigates more robust solutions for exemplar-based video colourisation and proposes a deployable pipeline for video broadcasting production.

The chapters regarding content distribution with video compression and explainability are organised as follows.

Chapter 6 introduces a novel neural network architecture for chroma intra prediction that integrates attention modules to control the contribution of each neighbouring reference sample when computing the prediction of each chroma pixel in the current block sample location. The proposed scheme better captures the relationship between the luma and chroma components, resulting in more accurate prediction samples.

Chapter 7 proposes a block-independent multi-model and training methodology that reduces the complexity of the proposed attention-based neural network whilst remaining competitive with the current state-of-the-art. Proposed simplifications include a framework to reduce the complexity of the convolutional operations, a simplified cross-component processing model using sparse autoencoders and a fast and cost-effective implementation using integer precision approximations.

Chapter 2

Background concepts

2.1 Chapter overview

This chapter provides a comprehensive overview of the main background concepts and principles used in the following chapters. Note that the selected content is high level in nature and more extensive detailed explanations of relevant material can be found in the later chapters. The first part provides a review of the use of AI in the broadcasting industry. A general workflow is presented in Section 2.2, introducing the main production stages and areas this research intends to focus on. The common methods observed across the various task areas are then identified in order to find the overall AI trends within the digital broadcasting workflow. Section 2.3 provides the basics of colour vision, introducing the concept of colour spaces and its application in image and video processing. Finally, Section 2.4 describes deep learning as the current state-of-the-art approach to computer vision and gives a brief introduction to its basic principles.

2.2 Digital broadcasting workflow

The broadcasting industry has been gradually shaped by a technological transformation that has brought innovation and a range of challenges to news practices and media production. Digitalisation of news has enabled the creation of *integrated newsrooms* [8], which have allowed the adoption of improved content creation pathways as well as novel consumption and distribution methodologies. Furthermore, the cross-platform distribution has stimulated the creation of new communication channels between the broadcasters and their audiences such as comment sections and social media [9].

As shown in Figure 2.1, digital newsrooms integrate different production and distribution pipelines into an end-to-end workflow that centralise all the audiovisual

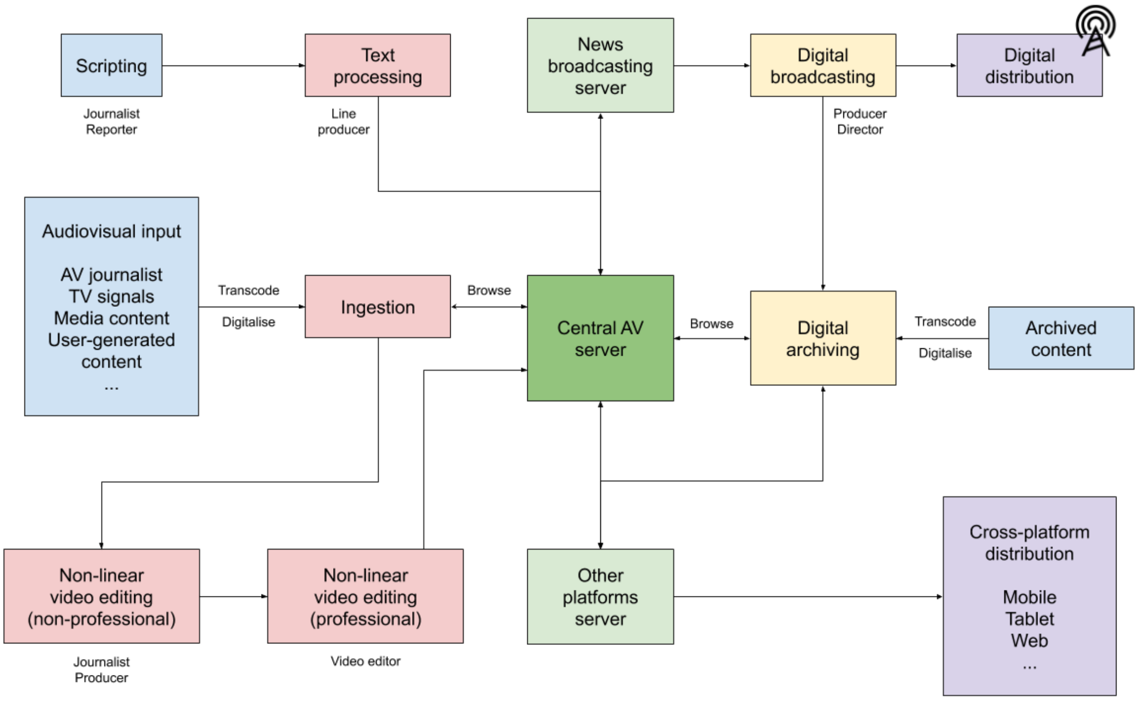


Figure 2.1: Broadcasting workflow for an integrated newsroom (modified from [1]).

content. Multiple inputs are supported including multimedia sources from TV reporters and operators, TV signals from other channels, media and user-generated content; and content from legacy archives. Audiovisual sources are ingested to the system by means of a transcoding and digitalisation process and further processed by non-linear editing systems which enable video enhancement and production [1]. Finally, all signals are stored into a centralised AV server which allows for efficient access and processing. Centralised content also allows the management of multiple distribution threads which complement the broadcasting emissions with digital archiving and cross-platform distribution [10].

Notice that the intended video production and distribution technologies in this research have significant application in many areas of the given workflow, specifically on video enhancement and colourisation and video compression for digital distribution and storage.

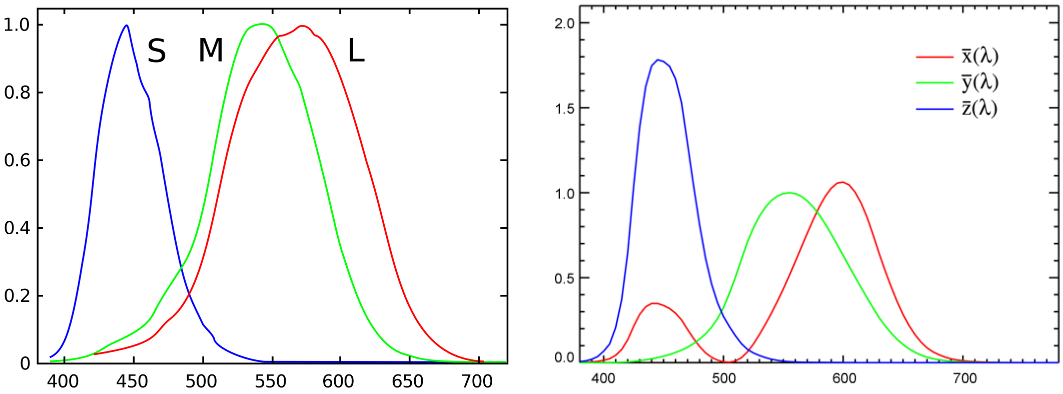
2.2.1 AI trends in broadcasting

Over the years, the adoption of AI technologies has enabled the automatisation of many stages of the broadcasting workflow. Broadcasters are incorporating AI into a wide range of areas such as automated content creation, bandwidth optimisation, content creation from legacy archives, targeting audience demographics, metadata creation, dynamic advertising and content personalisation [7]. TVU Networks showcased their AI-based *MediaMind*[1] solution at IBC 2019. AI engines are used to recognise video content and automate its creation and cross-platform distribution. During the process, TVU indexes video content using metadata automatically generated by AI, enabling the producers to create personalised content for specific audience groups. Similarly, the *Wibbitz*[2] platform uses AI to support video content creation and audience engagement by analysing and summarising text-based information into video storyboards. Its broadcasting partners include Reuters, Bloomberg, NBC and USA Today. The BBC has developed the Single Operator Mixing Application (*SOMA*)[3], an automatic technology for generating virtual camera views from high-resolution videos. The method enables interactivity between producers and the camera operators boosting the creation capacity as well as saving production costs. Wimbledon and IBM[4] used AI to generate automatic highlight sport clips. Facial and emotion recognition techniques were applied to select emotionally impactful highlights. Regarding content programming, BBC used AI algorithms to extract metadata from legacy archived content in advance of automatically programming two days of its BBC 4 channel based on demographic targeting[5]. Social media data is increasingly used within the integrated newsroom with a high potential to improve the efficiency of programme production. An example is the *Spectee*[6] service from Fuiji Television, using an AI engine to automatically analyse social media and then broadcast newsworthy topics via a streaming channel. The New York Times, The Guardian and The Economist are using the AI-based *Perspective*[7] tool to analyse the level of toxicity of their comment sections through keyword recognition. Finally, AI has significantly improved captioning systems allowing the broadcasters automatically produce quick and accurate captions and subtitles and even translations to foreign languages. An example is *Watson Captioning*[8] service from IBM, an AI-based speech recognition technology able to generate automatic captions to live video programming as well as on-demand content. Such methods allow the broadcasters increase their workflow efficiency and the viewing experience.

2.3 Basics of colour vision

Colour is a sensation created in response to excitation of the visual system by electromagnetic radiation known as light [11]. More specific, a visible colour is a source

[1] TVU MediaMind - `https://www.tvunetworks.com.ru/solutions/tvu-mediamind/`
[2] Wibbitz - `https://www.wibbitz.com/`
[3] SOMA - `https://www.bbc.co.uk/rd/blog/2017-07-compositing-mixing-video-browser`
[4] IBM Wimbledon - `https://www.ibm.com/uk-en/marketing/wimbledon/`
[5] AI TV on BBC 4.1 - `https://www.bbc.co.uk/programmes/p06jt9ng`
[6] Spectee - `https://corp.spectee.com/news/archives/04-2019`
[7] Perspective - `https://www.perspectiveapi.com/`
[8] Watson Captioning - `lhttps://www.ibm.com/products/watson-captioning-live`



a)

b)

Figure 2.2: a) human cone cell responses (S-hort, M-edium, and L-ong types) to monochromatic spectral stimuli; b) CIE $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ matching functions.

of light within the visible region of the electromagnetic spectrum (with a range of wavelengths from 390 nm to 700 nm approximately). The ability of the human eye to distinguish colours is based upon the varying sensitivity of the different cells in the retina to different wavelengths of light. Human colour vision is trichromatic, because the retina has three different photo-receptor cells, called cones, which respond to light radiation with different spectral response curves. As shown in Figure 2.2-a, the maximum absorption probabilities of the three receptor classes occur at 580, 540, and 440 nm, which represent the red, green and blue colours, respectively.

Since the human vision system only needs three types of photo-receptor cells, a three-component numerical basis is sufficient to represent a colour. In other words, a colour can be interpreted as a 3-dimensional vector that weights the contribution of each spectral function. The three components of the colour basis can be defined in different ways leading to various colour models (or colour spaces) [12]. The Commission Internationale de L'Eclairage (CIE) adopted the standard observer colour matching functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ as a result of a perceptual experiment conducted in 1930. Those curves represent the amount of primary red, green and blue light needed to represent each colour from the visible spectrum. As shown in Figure 2.2-b, the green curve corresponds to the luminosity function, which represents the sensitivity of the human eye to brightness. From these curves, several colour spaces can be defined based on the needs of sensing and displaying, some examples include CIE XYZ, RGB, CIE Lab or YUV.

2.4 Deep learning in computer vision

The last decade has seen approaches to computer vision shift towards models based on Deep Learning (DL), where the features used to build representations of the world are not hand crafted: they are learned from data through a training process. Deep learning has demonstrated superior performance compared to traditional methods when applied to a variety of challenging computer vision and image processing tasks. Methods based on Convolutional Neural Networks (CNNs) have been particularly successful in solving image classification [13] and object detection [14] problems, as well as regression problems including image segmentation [15], super-resolution [16] and restoration [17]. CNNs are a type of feed-forward artificial neural network in which the connectivity pattern between its neurons, is inspired by the organisation of the animal visual cortex. Individual cortical neurons respond to stimuli in a restricted region of space known as the receptive field. The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a convolution operation:

$$S(i,j) = \sum_m \sum_n X(m,n)W(i-m, j-n), \tag{2.1}$$

where X is the input with $m \times n$ dimensions and W is the convolutional kernel with learning parameters.

A CNN works similarly to a neural network (NN): each neuron receive an input, a dot product (Hadamard product or element-wise multiplication) between each input and its associated weight is performed, followed with a non-linearity. The most common hierarchical distribution of CNN layers contains: (1) input layer, containing the pixel values from input images, (2) convolutional layers, the core block of CNNs, computes a locally dot product (2D in the case of images) between the weights and a certain tiny region of the input volume, (3) non-linear layers, most of the times using a ReLU activation function [18] which applies an element-wise activation by thresholding at zero and (4) pooling layers that apply a spatial downsampling along the output volume, note that this operation can be replaced by a strided convolution or similar. Finally, weight sharing can be used to reduce the number of parameters that must be learned, which reduces model training time and cost, while making feature search insensitive to feature location in the image [19].

2.4.1 Training the model

The training algorithm is the learning process by which the model parameters (or weights) are updated by means of a series of iterations where the training samples are input to the model and the predicted outputs are evaluated on the sample labels or ground truth. The error from the model is measured by means of a cost function (or loss) which is minimised with the aim to guide the learning process.

The optimisation algorithm (or optimiser) used to minimise the loss function is

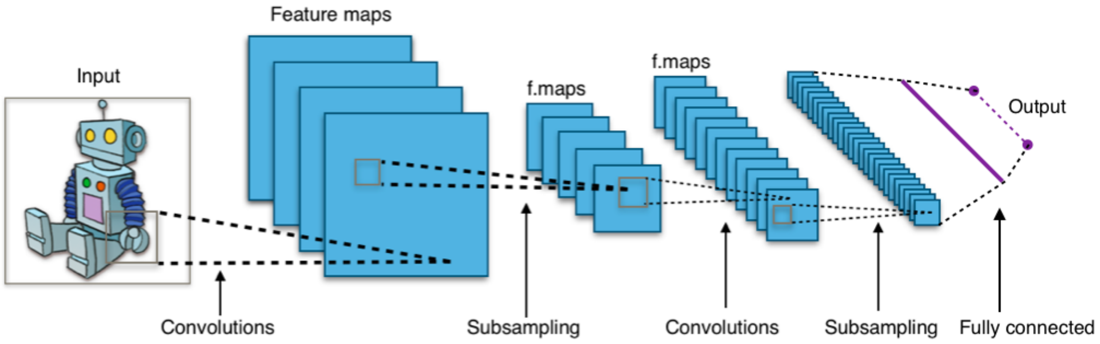


Figure 2.3: Example of CNN architecture for image classification.

a variation of stochastic gradient descent, which starts with a random initialisation of the network weights θ and for each iteration t, steps towards to a given direction that implies a faster reduction in the loss value. Mathematically, gradient descent computes the derivative of the loss function $\nabla_\theta \mathcal{L}(\theta_t)$ with respect to θ and updates the parameters in the opposite direction of the gradient, as follows:

$$\theta_{t+1} := \theta_t - \alpha \nabla_\theta \mathcal{L}(\theta_t), \tag{2.2}$$

where θ_{t+1} are the weights after the update, θ_t the current weights and α the learning rate, which is the size of the step that the optimiser will do at each iteration.

The most common strategy to apply gradient descent to any type of multi-layer feed-forward neural network is the use of backpropagation [20]. Backpropagation provides an efficient method for calculation the gradient of the loss by applying the chain rule and hence computing gradients from different layers with respect their weights. Mathematically, the gradient of the weights at layer l can be derived with respect to the output of the previous layer $o^{(l-1)}$ by means of the chain rule, as follows:

$$\frac{\partial \mathcal{L}(\theta^{(l)})}{\partial \theta^{(l)}} = \frac{\partial \mathcal{L}(\theta^{(l-1)})}{\partial o^{(l-1)}} \frac{\partial \mathcal{L}(o^{(l-1)})}{\partial \theta^{(l-1)}}. \tag{2.3}$$

This way, the error computed in the last layer is propagated backwards through the network to compute the gradients for each layer.

2.4.2 Deep generative models

The aim of generative modelling is to learn a representation of an intractable probability distribution $\mathcal{X}$ defined over $\mathbb{R}^n$, where n is typically large, and the distribution is complicated [21]. The goal is to obtain a generator $G : \mathbb{R}^q \to \mathbb{R}^n$ that maps samples from a tractable distribution $\mathcal{Z}$ supported in $\mathbb{R}^q$ to points in $\mathbb{R}^n$ that resemble the given data. Hence, the formulation assumes that for each sample $\mathbf{x} \sim \mathcal{X}$ there is at least one point $\mathbf{z} \sim \mathcal{Z}$, such that $G(\mathbf{z}) \approx \mathbf{x}$. Since the vector $\mathbf{z}$ that results in $\mathbf{x}$

is often unknown, it is commonly referred as latent variable within the latent space $\mathcal{Z}$. Therefore, $\mathcal{Z}$ can be treated as a probability distribution $p_{\mathcal{Z}}(\mathbf{z})$, such that the generator can be used to compute the likelihood of a particular samples $\mathbf{x}$:

$$p_{\mathcal{X}}(\mathbf{x}) = \int p_g(\mathbf{x}|\mathbf{z}) p_{\mathcal{Z}}(\mathbf{z}) d\mathbf{z}, \tag{2.4}$$

where the likelihood $p_g(\mathbf{x}|\mathbf{z})p_{\mathcal{Z}}$ measures how close $G(\mathbf{z})$ is to $\mathbf{x}$.

Deep generative models are neural networks with many hidden layers trained to approximate high-dimensional probability distributions with large number of samples. The main property is that such generative networks have a number of parameters significantly smaller than the amount of data used to train them, so they are forced to discover and efficiently internalise the essence of the data in order to generate it. Popular deep generative models include Normalising Flows (NF) [22], Variational Autoencoders (VAEs) [23], Generative Adversarial Networks (GANs) [24] and auto-regressive models [25].

In particular, this thesis applies GANs to automatic colourisation, so a more extensive explanation is made next. GANs are based on an adversarial learning in which a generative model G is trained simultaneously with a discriminative model D that outputs an scalar and estimates the probability that a sample came from the training data rather than G [24]. The training procedure for D is to maximise the probability of assigning the correct label to both training samples and samples from G, and simultaneously, the objective of G is to minimise $\log(1 - D(G(\mathbf{z})))$. Therefore, D and G play a two-player minimax game with value function $V(G, D)$, as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})} \left[\log D(\mathbf{x})\right] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})} \left[\log(1 - D(G(\mathbf{z})))\right]. \tag{2.5}$$

Although the above formulation proposes a log-likelihood optimisation, different strategies are proposed with the aim to gain stability and mitigate some training issues such as the mode collapse or vanishing gradients. Some examples include LSGAN [26] or WGAN [27].

2.4.3 Attention models

Attention mechanism has become an important concept in neural networks that has been researched within diverse application domain such as natural language processing [28, 29], speech [30] and computer vision [31]. The idea behind attention models is inspired by the human biological system and is based on reducing complex tasks by predicting smaller areas of attention that are processed sequentially in order to encourage more efficient learning.

Mathematically, attention models can be understood using a regression model as proposed in [29]. Given a training data of N samples comprising features and their corresponding target values $\{(x_1, y_1), (x_1, y_1), \ldots, (x_N, y_N)\}$, the objective is to predict the value $\hat{y}$ for a new query instance x. A simple estimator can be the average of all target values from all training instances: $\hat{y} = \frac{1}{N} \sum_n x_n$. However, a better estimation can be obtained by using a weighted average where the weights measure the relevance or contribution of each training sample to the prediction of the query: $\hat{y} = \frac{1}{N} \sum_n \mathbf{s}(x, x_n) y_n$, where $\mathbf{s}(x, x_n)$ is the relevance measure typically obtained with a normalised Gaussian kernel or other similarity metric. As noticed in [29], the attention mechanism can be viewed as a generalisation which enables a learning process of the weighting function.

Many attention mechanisms can be found in the literature, differing mainly on the definition of the weighting function. Some examples include content-based attention [32], additive attention [28], location-based attention [33] or self-attention [34]. One of the most impactful use cases of attention is the definition of the transformer network [29], which applies self-attention layers to outperform the state-of-the-art of many sequential tasks without using recurrent architectures.

Part I

Content production: image and video colourisation

Chapter 3

Fully automatic colourisation

3.1 Chapter overview

This chapter provides a detailed description of the contribution of this research to a recent approach to fully automatic image colourisation using end-to-end conditional Generative Adversarial Networks (GANs), as well as a thorough description of the configuration setup used in the experiments. Recent advances in conditional GANs are investigated to develop an end-to-end architecture based on Convolutional Neural Networks (CNNs) to directly map realistic colours to an input greyscale image. The work described in this chapter was published at the IEEE International Workshop on Multimedia Signal Processing (MMSP), 2019 [35]. Section 3.3 reviews related work in the literature, identifying the main draw-backs and possible improvements, Section 3.4 details the proposed methodology, Section 3.5 provides information about the implementation and data used in the experiments and a quantitative evaluation of the results while Section 3.6 identifies future work and Section 3.7 provides conclusions.

3.2 Introduction

Colourisation refers to the process of adding colours to greyscale or other monochrome images such that the colourised results are perceptually meaningful and visually appealing. In general, greyscale content is present in many multimedia applications: from *black and white* videos in old archives and videos with faded colours, to computer vision applications that discard the chroma component in order to simplify processing. However, while the luminance information provides valuable content-related information regarding shapes and structures, the perception of colour is important for modern video viewing. For this reason, adding chromatic information to images and improving the quality of colour has become a research area of signifi-

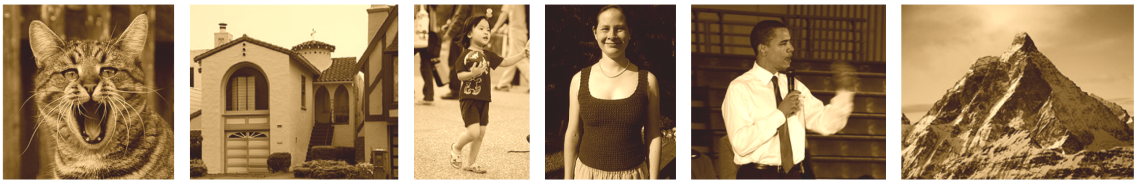


Figure 3.1: Examples of desaturated results obtained with colourisation approaches based on regression.

cant interest for a wide variety of domains that traditionally have resorted to using luminance data alone. This includes medical imaging [36], surveillance systems [37] or restoration of degraded historical images [17].

Recently, the emergence of deep learning has enabled the development of new colourisation algorithms which better generalise the natural data distribution of colours. Convolutional Neural Networks (CNNs) outperform many state-of-the-art methods based on hand-crafted features in tasks such as image enhancement, image classification or object detection [13, 14]. State-of-the-art colourisation methods based on Generative Adversarial Neural Networks (GANs) [2] aim to mimic the natural colour distribution of the training data by forcing the generated samples to be indistinguishable from natural images. Moreover, using an adversarial loss, the discriminator can learn a correct adaptation of the differences between generated and real images in the target domain. However, as shown in Figure 3.1, existing methods still suffer ambiguity when trying to predict realistic colours, often causing desaturated results. Nevertheless, GANs are a suitable basis for further tackling the desaturation problem and gaining colourfulness.

Motivated by the recent success of Conditional Adversarial Networks in image-to-image translation tasks, including colourisation [2, 38], this chapter proposes an automatic colourisation paradigm using end-to-end Convolutional Neural Network architectures. Improved colourisation is achieved by introducing techniques that improve the stability of the adversarial loss during training, leading to better colourisation of a variety of images from large multi-class datasets. Further enhancements are achieved by applying feature normalisation techniques which are widely used in style transfer models. The capabilities of adversarial models in image colourisation are improved by adapting an Instance-Batch Normalisation (IBN) convolutional architecture [39] to conditional GANs. The main contributions of this chapter are the following:

1. Analysis of drawbacks in state-of-the art methods for automatic image colourisation.

2. Identification of appropriate architectural features and training procedures which lead to a boosted GAN performance for image colourisation. The pro-

posed steps of improvement include:

(a) A novel generator-discriminator setting which adapts the IBN paradigm to an encoder-decoder architecture, enabling generalisation of the content's style changes while encouraging stabilisation during GAN training.

(b) The use of Spectral Normalisation (SN) [40] for improving the generalisation of the adversarial colourisation and preventing training instability.

(c) The use of multi-scale discriminators to achieve an improved colour generation in small areas and local details and a boosted colourfulness.

3.3 Related Work

Automatic colourisation was originally introduced in 1970 to describe a novel computer-assisted technology for adding colour to black and white movies and TV programs [41]. Although such semi-automatic method improved the efficiency of traditional hand-crafted techniques, it still required a considerable amount of manual effort and artistic experience to achieve acceptable results. Since then, it has been shown that the task is complex, ill-conditioned and inherently ambiguous due to the large degrees of freedom during the assignment of colour information [42].

In some cases, the semantics of the scene and the variations of the luminance intensity can help to infer priors of the colour distribution of the image. For example, an algorithm can successfully associate rapid changes to vegetation areas, assigning ranges of green to it, or smooth areas to sky, inferring blue tones. Nevertheless, in most cases the ambiguity in the decisions can lead a system to make random choices. For instance, the hypothetical prior of a car being red is the same as it being green or blue, although in reality the decision will converge towards the dominant samples in the training data. Another common issue is the well-known desaturated effect [42, 4], which is associated with treating automatic colourisation as a standard regression problem. Taking a greyscale input image, a parametric model can learn how to predict corresponding chrominance channels by minimising the Euclidean distance between the estimations and the ground truth. Nevertheless, basic solutions are commonly based on averaging the colours of the training examples. In this way the basic model produces desaturated results characterised by low absolute values in the colour channels when trained on large databases of natural images. Previously this problem has been addressed through a deep learning approach which introduced a rebalancing process during training with the aim of penalising the predicted colours based on their likelihood in the prior distribution of training data [42]. Such a method outperforms previous state-of-the-art approaches, including recent successes

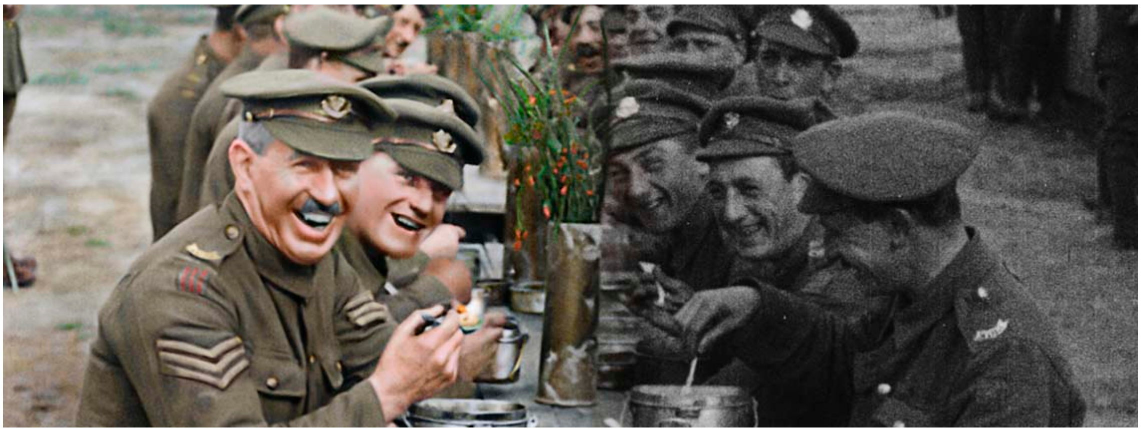


Figure 3.2: Peter Jackson's World War I documentary, They Shall Not Grow Old. A successful colourisation example in a real broadcasting and restoration scenario. Computer-assisted technologies were used to colourise original footages during the conflict, provided by BBC Archives and the Imperial War Museum (IWM).

with GANs, in which more complex architectures need to adopt the methodology to generalise the predicted colours.

As proposed in the *pix2pix* framework [2], a more traditional regression loss such as L_1 or L_2 distance is beneficial when included in the final objective function. This enables a conditional GAN to increase the error rate of the discriminator while producing realistic results close to the ground truth. Although such a framework achieves state-of-the-art performance across a range of image-to-image translation tasks, it still requires the aforementioned rebalancing method, targeting colour rarity far from the desaturated mean of natural data distributions. The high instability during training when a GAN deals with complex generator architectures and high-resolution training images, can lead the *pix2pix* framework to mode collapse, converging towards undesirable local equilibria between the generator and discriminator [43, 38]. This effect reduces the contribution of the adversarial loss in the multi-loss objective, giving the total weight of convergence to the regression loss and hence leading the system again to desaturated results.

3.4 Proposed Method

Aiming at colourisation of images, the goal of our method is to enable automatic CNN-based colourisation of an input greyscale image, denoted $X \in \mathbb{R}^{H \times W \times 1}$, where $H \times W$ is image dimension in pixels, and represented by the lightness channel L in the CIE Lab colour space [44]. To achieve this, it is essential to train an end-to-end CNN architecture capable of learning the direct mapping $\hat{Y} = \mathcal{F}(X)$ to the two associated *ab* colour channels $Y \in \mathbb{R}^{H \times W \times 2}$. As commonly used in the literature, CIE Lab colour space is chosen as it is designed to maintain perceptual uniformity

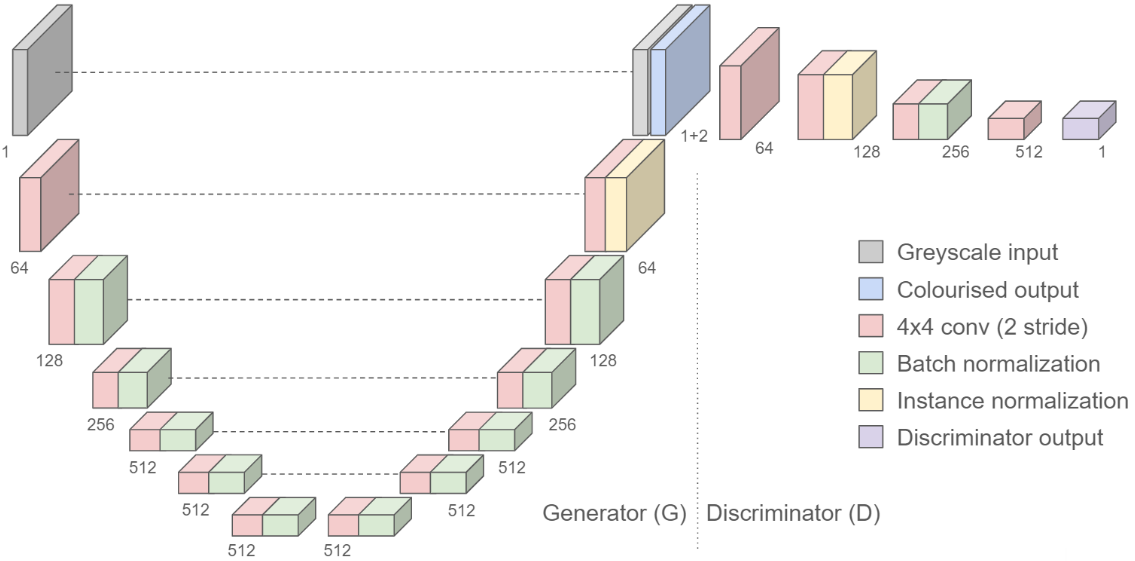


Figure 3.3: Proposed generator and discriminator architectures with IBN adaptation to the U-Net and PatchGAN architectures. Note the fake input to the discriminator is composed by the concatenation $(1 + 2)$ of the original greyscale input and the generated 2-dimensional colour channels.

and is more perceptually linear than other colour spaces [44]. The mapping function $\mathcal{F}(X, \theta)$ can be expressed in a neural network form as:

$$\mathcal{F}(X, \theta) = \mathcal{W}^{L+1} a_L(\dots a_1(\mathcal{W}^1 X) \dots), \tag{3.1}$$

where $\theta := \{\mathcal{W}^1, \dots, \mathcal{W}^{L+1}\}$ is the set of learning parameters for a L-layer CNN, omitting the bias terms for simplicity, and a_l the corresponding non-linear activation function, with $l = \{1, \dots, L\}$.

3.4.1 Conditional Adversarial Networks

A mapping function is learnt using a generative adversarial methodology with conditional GANs. This work uses the *pix2pix* framework [2] as baseline to solve image-to-image translation tasks such as generating realistic street scenes from semantic segmentation maps, aerial photography from cartographic maps or image colourisation from greyscale inputs. As per the traditional GANs setting [24], two CNNs (a generator G and discriminator D) are trained simultaneously in a minimax two-player game, with the objective of reaching the *Nash equilibrium* between them. Given an input greyscale image X and a vector of random noise Z, the aim of the generator G is to capture the original colour distribution of the training data and to learn a realistic mapping $G(X, Z; \theta_G)$ to the colourisation result. On the other hand, the discriminator D aims to distinguish real images from colourised ones through the mapping $D(X, Y; \theta_D)$, estimating the probability that a sample came from the

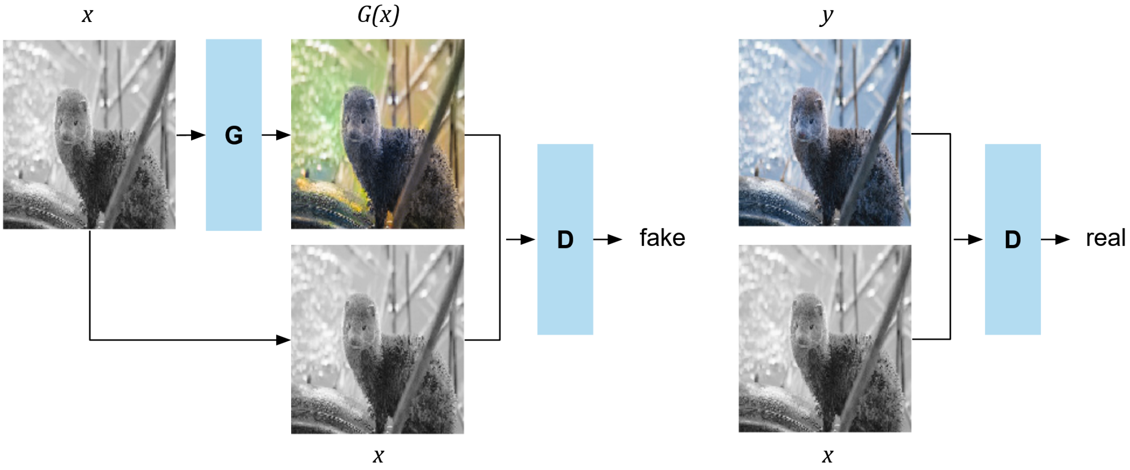


Figure 3.4: Conditional GAN framework in [2] applied to image colourisation.

training data rather than from G. Therefore, the conditional GAN framework will model the colour distribution of the training data following the minimax training strategy:

$$\min_G \max_D V(G, D) \tag{3.2}$$

where the objective function $V(G, D)$ is given by

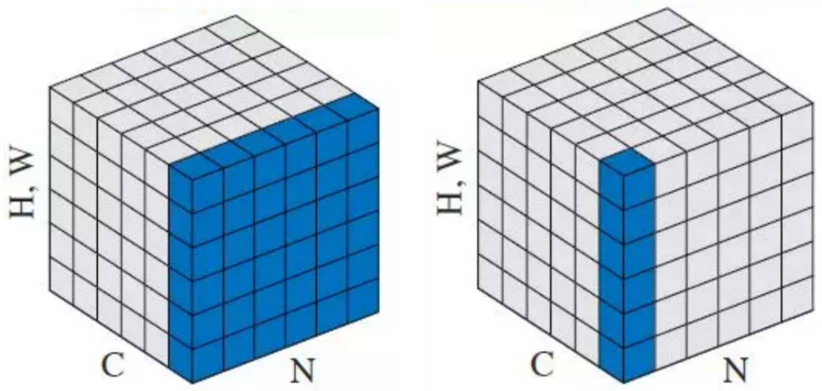
$$V(G, D) = J^{(G)}(\theta_G, \theta_D) + J^{(D)}(\theta_G, \theta_D) \tag{3.3}$$

$$J^{(D)}(\theta_G, \theta_D) = \mathbb{E}_{X,Y}[\log(D(X, Y))] + \mathbb{E}_{X,Z}[\log(1 - D(G(X, Z)))] \tag{3.4}$$

$$J^{(G)}(\theta_G, \theta_D) = -\mathbb{E}_{X,Z}[\log(D(X, G(X, Z))] + \lambda\mathbb{E}_{X,Y,Z}[\|Y - G(X, Z)\|_1] \tag{3.5}$$

using λ to control the contribution of the regression loss.

As suggested in recent works [24, 45], the standard loss function for the generator is redefined in order to guarantee non-saturation by maximising the probability of the discriminator being mistaken and converting the loss to a strictly decreasing function. Moreover, note the aforementioned L_1 distance introduced in the final generator objective to encourage a colourisation close to the ground truth outputs. Regarding the GAN architectures, the *pix2pix* framework uses a U-Net [46] as generator and a Markovian PatchGAN [2] as discriminator, yielding output probability maps based on the discrimination of $N \times N$ patches in the input domain. They exploit the intrinsic fully convolutional architecture of the discriminator to control the input patch size via its respective receptive field.



Batch normalisation Instance normalisation

Figure 3.5: Comparison between mini-batch normalisation techniques. The volumes represent a batch of N samples with $H \times W \times C$ dimensions. Source from Group Normalisation paper [3].

3.4.2 Mini-batch normalisation

The application of mini-batch normalisation techniques such as Batch Normalisation (BN) [47], have become a common practice in deep learning to accelerate the training of deep neural networks. In the case of GANs, the DCGAN architecture was proven that applying batch normalisation in both generator and discriminator architectures can be very beneficial to stabilise the GAN learning and to prevent a mode collapse due to poor initialisation [48]. Internally, batch normalisation preserves content-related information by reducing the covariance shift within a mini-batch during training. It uses the internal mean and variance of the batch to normalise each feature channel. On the other hand, Instance Normalisation (IN) [49] was proven to be beneficial in style transfer speeding-up fast stylisation. Image colourisation, as other style transfer techniques, aims to capture style information by learning features that are invariant to appearance changes, with the aim to generalise the colourisation process within a mini-batch of variable content. Therefore, unlike batch normalisation, IN uses the statistics of an individual sample instead of the whole mini-batch to normalise features.

Inspired by IBN-Net [39], in the presented approach BN and IN are combined in the same convolutional architecture with the aim to exploit the instance-based normalisation capabilities in style transfer while encouraging stabilisation during training, to both improve the learning and generalisation capacities of the GAN. This work adapts the residual IBN-Net architecture to a U-Net generator and a patch-based discriminator. The IBN-Net work discussed that appearance variance in a deep convolutional model mainly lies in shallow layers, while the feature discrimination for content is higher in deep layers. Therefore, IBN-Net avoids IN in deep layers to preserve content discrimination in deep features, while it keeps batch normalisation in the whole architecture to preserve content-related features at different levels. Figure 3.3 shows final proposed architectures for generator and discriminator. Note that normalisation is not applied to the input layers to avoid sample oscillation and model instability.

3.4.3 Weight regularisation

One common strategy to improve the generalisation of the network and to prevent instability during training is the use weight regularisation. This technique penalises proportionally the weights of the network based on their size, aiming to keep small values during training and hence preventing small changes in the input leading to large changes in the output. In the context of GANs, Spectral Normalisation (SN) [40] can be used as a regulariser to prevent exploiting gradients by controlling the Lipschitz constant of the discriminator. Theoretically, one function $f(a)$ is K-Lipschitz continuous if $\|f(u) - f(v)\|_2 \leq K\|u - v\|_2$, for any v, u. Spectral normalisation controls the Lipschitz constant of the discriminator by constraining the spectral norm of each layer. Being $\mathcal{W}$ a linear mapping between the pre-activations of two layers and x an input tensor, we can simplify the Lipschitz continuity requirement by setting the reference point $u = 0$, getting therefore $\|\mathcal{W}x\| \leq K\|x\|_2$ which is equivalent to $\langle(\mathcal{W}^T\mathcal{W} - K^2)x, x\rangle \leq 0, \forall x \in I$. Then, expanding x in the orthonormal basis of eigenvectors of $\mathcal{W}^T\mathcal{W}$ and following their demonstration in [40], we get $\sum_i (K^2 - \lambda_i)x_i^2 \leq 0$, proving that K must be the largest eigenvalue of $\mathcal{W}^T\mathcal{W}$ to satisfy the aforementioned constraints. Given that, they propose to perform spectral normalisation by simply replacing every weight W with $W/\sigma(W)$, where $\sigma(W)$ is the spectral norm or the largest single value of W.

3.4.4 Multi-scale discriminators

A challenge in colourisation is to achieve precision in small areas and local details. Using the Markovian PatchGAN discrimination in the *pix2pix* framework, colourfulness can be boosted by increasing the receptive field of the discriminator, albeit at the price of increasing the complexity with deeper architectures and loosing spatial information, commonly leading to blurry effects and tilling artefacts. A better solution is to use the multi-scale discrimination setting to tackle high-resolution image processing without varying the discriminator architecture [38]. This is achieved using N discriminators at different scales by downsampling the actual inputs. Therefore, keeping fixed the original discriminator architecture, variable receptive fields are obtained. These fields are larger at the coarsest levels, and the modified objective function $V'(G, D)$ in the GAN context is given as:

$$V'(G, D) = \sum_{n=0}^{N-1} V(G, D_n) \tag{3.6}$$

where $\{X_n,\ (G(X))_n\} \in \mathbb{R}^{M \times H_n \times W_n \times C}$ are the D_n inputs, with $H_n = H/2^n$, $W_n = W/2^n$ and N the number of discriminator scales.

3.5 Experiments

This section provides details about the implementation and data used in the experiments. Moreover, a quantitative evaluation of the performance and a detailed analysis of the results is also provided. Finally, some colourisation examples are shown comparing the baseline architecture with the proposed configurations.

3.5.1 Data

Training examples are generated from the *ImageNet* dataset [50]. Samples are selected from the reduced validation set, containing 50,000 *RGB* images uniformly distributed as 50 images per class. The test dataset is created by randomly selecting 10 images per class from the training set, generating up to 10,000 examples. All images are resized to 256 × 256 pixels and converted to the CIE Lab colour space.

3.5.2 Network architectures

The *pix2pix* framework is used as a GAN baseline. The generator consists of a U-Net encoder-decoder architecture, conforming to the following structure: $e64 : e128 : e256 : e512 : e512 : e512 : e512 - d512 : d512 : d512 : d256 : d128 : d64$. The encoder's e blocks consist of 4 × 4 convolutions with spectral normalisation and a stride of 2, followed by the normalisation layers as explained in Section 3.4.2 and a *Leaky ReLU* activation. The decoder d blocks apply the same block composition but using *ReLU* activations. The last layer is a 4 × 4 convolution with a *tanh* activation producing a 2-dimensional output space. For L layers of encoder-decoder architecture, skip connections between layers i at the encoder and layers $L-i$ at the decoder are applied in order to recover the information lost during the downsampling operations. After the generator, the discriminator is used in a form of 70 × 70 PatchGAN with the following fully convolutional architecture: $e64 : e128 : e256 : e512$. The output layer is a 4×4 convolution producing the output probability maps, and the input layer takes the concatenation between the original greyscale input and the original or generated colour channels. Regarding the multi-scale discrimination, a setup of 3 different discriminators is used, downsampling the original input volumes by a factor of 2 and 4.

3.5.3 Quantitative evaluation

Figure 3.6 illustrates the convergence behaviour of the adversarial and regression losses conforming the generator's objective function. A poor response from adversarial loss can be observed for the baseline *pix2pix* method, represented by the BN line, which rapidly collapses to a local minimum, giving all the weight of global convergence to the regression loss. A loss of colourfulness occurs after this point where the regression loss abruptly starts to overfit leading to the generation of desaturated colours. A considerable improvement results after adding spectral normalisation, the BN+SN line, where weight regularisation helps to stabilise the adversarial loss and slows down the convergence of the regression, hence preserving colourfulness and preventing overfitting. The aforementioned behaviour can be validated by observing the IBN+SN line. Although instance normalisation leads to instability due to increasing the variance of content-based features during training, a sudden improvement of the adversarial loss can be observed after epoch 16, where the combination of both normalisation techniques leads to colour generalisation while penalising the regression loss and helping the system to prevent desaturation.

Table 3.1: Quantitative evaluation of colourisation techniques

Method	L_1	PSNR [dB]	$\mathcal{L}_{perc}$
IN	9.92	26.70	63.85
BN + SN	10.76	25.69	58.58
IN + SN	9.89	26.73	60.36
BN + SN + MD	11.50	25.11	58.52
IBN + SN + MD	11.20	25.32	**57.77**
BN (baseline)	**9.83**	**26.77**	64.05

The effect of overfitting and lack of colourfulness can be evaluated by comparing deterministic measures, such as the averaged L_1 or L_2 distance, with perceptually-based ones designed to better capture the visual plausibility of the results. From the results summarised in Table 3.1 it can be observed that, unlike the perceptual evaluation, deterministic measures reflect poor performance for those models generating wider colour distributions, *e.g.* the L_2 chrominance distance of a red car colourised with a plausible blue will be always higher than being colourised with a desaturated colour. Additionally, the perceptual loss is computed using a *VGG19* model for image classification [51] pretrained on *Imagenet*. As proposed in previous works [52, 38], the L_1 distance between the convolutional features produced by

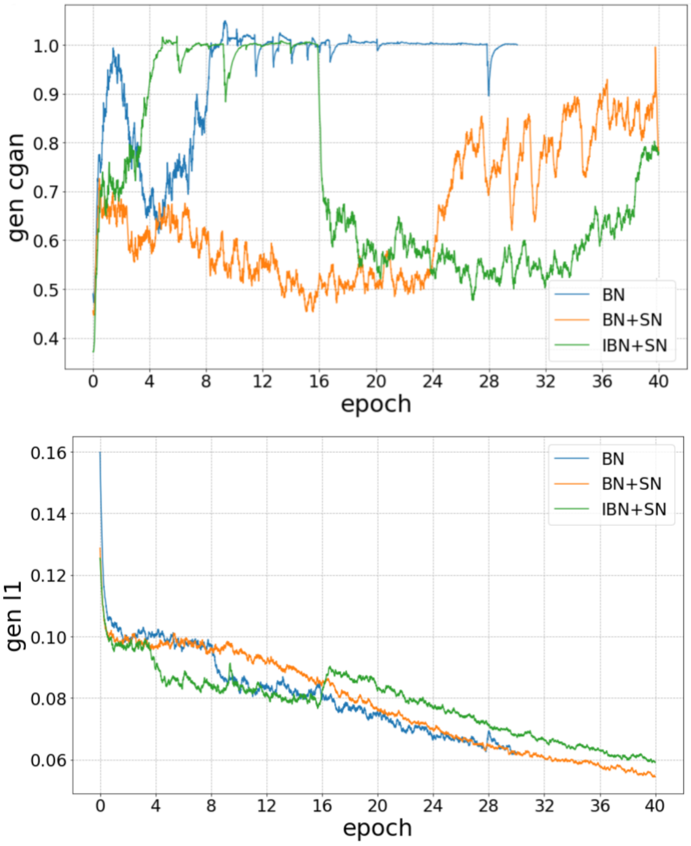


Figure 3.6: Learning curves of the generator during training.

classifying real and generated samples is averaged as:

$$\mathcal{L}_{perc} = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{N_i}\left(\|F^{(i)}(x) - F^{(i)}(G(x))\|_1^2\right) \tag{3.7}$$

where x is an input tensor, $F^{(i)}$ are the convolutional features from layers $relu\{i\}\_1$, $i \in \{1,\dots,5\}$ and N_i is the number of features of each volume.

Finally, colourfulness is evaluated by estimating the logarithmic colour distribution of the generated ab samples in the test dataset, comparing the proposed configurations with the prior distribution of the real data. As shown in Figure 3.7, SN provides improved colourfulness for both channels, reducing the area of intersection to the real data distribution with respect to the baseline methodology (BN) with uses Batch Normalisation. Finally, we improved the BN+SN setting by applying Multi-scale Discrimination (MD), which enables an increase in colourfulness by gaining detail in local and small areas. Examples of colourisation achieved by all analysed methods are presented in Figure 3.8.

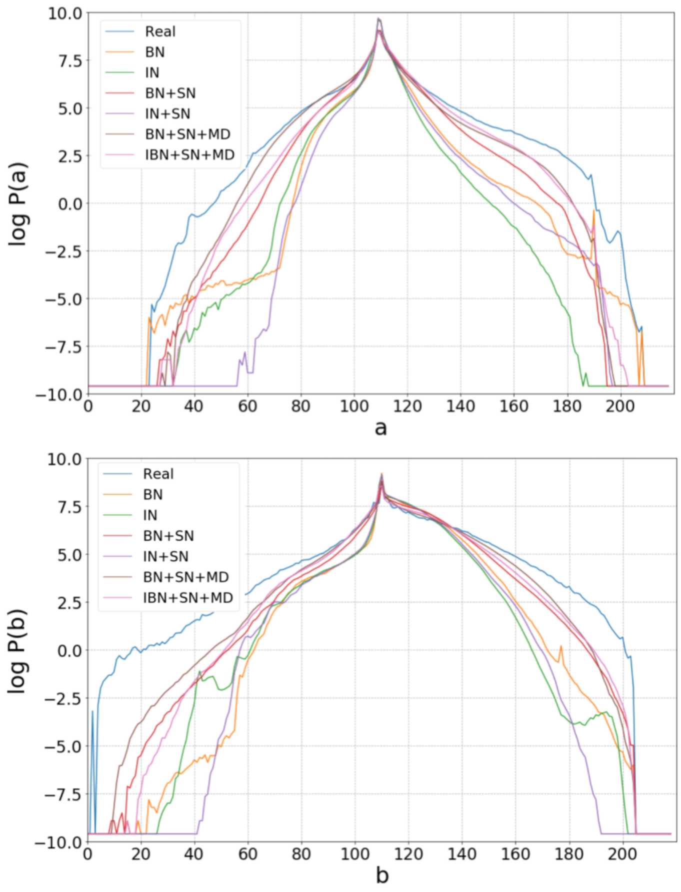


Figure 3.7: Comparison of colour histograms over the test data.

3.6 Future work

Although this work achieved promising results, mapping colours from a greyscale input is still a complex and ambiguous task. Therefore, more conservative solutions need to be researched reaching more realistic broadcasting use-cases by introducing colour references to guide the system towards more accurate colour predictions. The next chapter introduces a cost-effective architecture that outperforms state-of-the-art methods on exemplar-based image colourisation in competitive prediction times.

On the other hand, current work on fully automatic image colourisation needs to be improved in order to gain precision on local areas and consistently discriminate independent instances and objects in the scene. A possible research line might tackle this problem by adding the instance segmentation task in the prediction loop, so a two-head encoder-decoder architecture could be jointly trained to solve the colourisation task while outputting instance segmentation maps of the input grayscale image, in which the segmentation task can guide towards more accurate and discriminative predictions.

Finally, we collaborate with Polytechnic University of Catalonia (Barcelona, Spain) in [53], improving the methodology presented in this chapter by boosting the

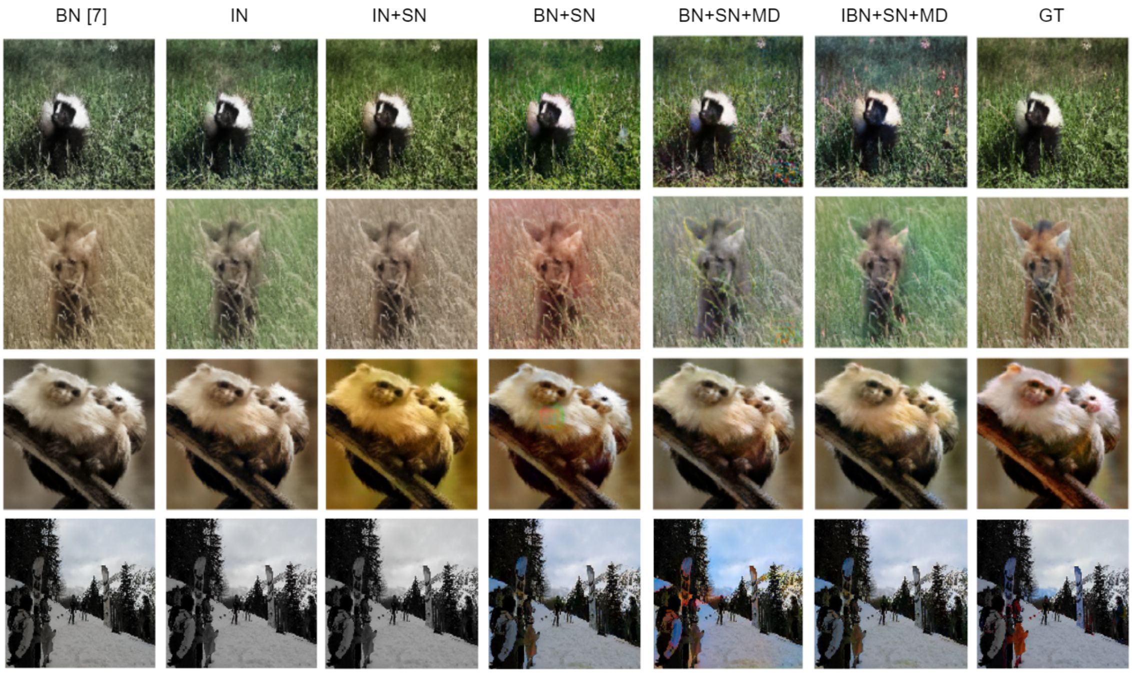


Figure 3.8: Visual comparison of colourisation techniques. Note the improvement of the proposed configurations over the BL model: (1) the gain in colourfulness after applying SN, (2) the localisation improvement of MD and (3) the benefits of IBN architecture rather than applying BN and IN separately.

realism and colourfulness of the generated predictions. Improvements are achieved by adding a feature reconstruction loss to guide the proposed training strategy. Such loss function uses a pre-trained VGG-16 [51] to match perceptual similarity between the real and generated images.

3.7 Conclusions

The work presented in this chapter improved the state-of-the art for automatic colourisation using conditional adversarial networks. The proposed GAN architecture integrates techniques from the literature to ensure good training stability and to increase the contribution of the adversarial loss during training, which prevents the GAN from collapsing into desaturated colours. It was also shown that batch normalisation and instance normalisation can be integrated together in a fully-convolutional encoder-decoder architecture within a GAN framework without lowering performance, and encouraging the assignment of more plausible colours. Finally, this work shows that by boosting the performance of the adversarial framework, reduction of the desaturation effect can be achieved due to improvement of the discrimination of unreliable colours.

Chapter 4

Exemplar-based colourisation

4.1 Chapter overview

This chapter proposes a more conservative approach for image colourisation to that described in the previous chapter that enables the use of colour references to guide the colour assignment process. An approach termed XCNET is proposed as a novel architecture that achieves fast and high-quality colour predictions. The proposed architecture integrates attention modules that learn how to extract and transfer style features from the reference images in an unsupervised way during the colourisation process. The work described in this chapter was published at the IEEE International Workshop on Multimedia Signal Processing (MMSP), 2021 [54]. Section 4.3 reviews related work in the literature, identifying the main draw-backs and possible improvements, Section 4.4 details the proposed methodology, Section 4.5 provides information about the implementation and data used in the experiments and a quantitative evaluation of the results while Section 4.6 identifies future work and Section 4.7 provides conclusions.

4.2 Introduction

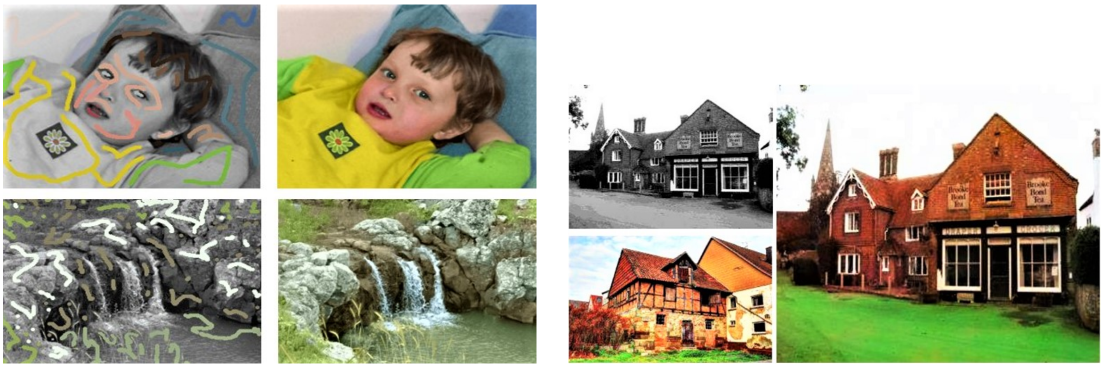
Although significant progress has been achieved, as Chapter 3 concluded, mapping colours from a grayscale input is a complex and ambiguous task due to the large degrees of freedom to arrive to a unique solution. In some cases, the semantics of the scene can help to infer priors of the colour distribution of the image, but in most cases the ambiguity in the decisions leads the system to make random choices, such as the colour of a car or a bird without further information. Thus, in order to overcome the ambiguity challenge, more conservative solutions propose the involvement of human interaction during the colour assignment process, introducing methodologies such as scribbled-based colourisation [4, 55, 56, 57, 58, 59, 60, 61, 62] or exemplar-based colourisation [63, 64, 5, 65, 66, 67, 68, 69, 55, 70, 71, 72, 73, 74]. Specifically, colourisation by example can be automated by means of a retrieval system to select content related references, which can also be used as a recommender for semi-automatic frameworks [5]. However, existing methods are either highly sensitive to the selection of references (need of similar content, position and size of related objects) or extremely complex and time consuming. For instance, most exemplar-based approaches require a style transfer or similar method to compute the semantic correspondences between the target and the reference before starting the colourisation process. This fact usually increments the system complexity by requiring twofold pipelines with separate and even independent style transfer and colourisation systems.

This chapter proposes a straightforward end-to-end solution which integrates attention modules that learn how to extract and transfer style features from the reference to the target in an unsupervised way during the colourisation process. Moreover, axial attention [75] is adopted to reduce the overall complexity and achieve a simple and fast architecture easily scalable to high resolution inputs. As shown in Figure 4.2, the proposed architecture uses a pre-trained backbone to extract semantic and style features at different scales from the grayscale target and colour reference. Then, attention modules at different resolutions extract analogies between both feature sources and automatically yield output feature maps that fuse the style of the reference to the content of the target. Finally, a multi-scale pyramid decoder generates colour predictions at multiple resolutions, enabling the representation of higher-level semantics and robustness on the variance of scale and size of the local areas of content. The main advantage of such an end-to-end solution is that the attention modules learn how to perform style transfer based on the needs of the colourisation decoder in order to encourage high quality and realistic predictions, even if the reference significantly mismatches the target content. Moreover, it generalises the similarity computation of previous image analogy approaches in a way that does not constrain the similarity to a specific local patch search (attention modules can be interpreted as a set of long-term deformable kernels) and to specific similarity metrics. Finally, the proposed architecture introduces a novel design of the conventional transformer, enabling a modular combination of multi-head attention layers at different resolutions.

Overall, the contributions of this work are threefold:

- A fast-end-to-end architecture for exemplar-based colourisation that improves existing methods while decreasing significantly the complexity and runtime.

- A multi-scale interpretation of the axial transformer for unsupervised style transfer and features analogy.



Scribble-based colourisation

Exemplar-based colourisation

Figure 4.1: Visualisation of supervised colourisation methods, including scribble-based results from [4] and exemplar-based results from [5].

- A multi-loss training strategy that combines a multi-scale adversarial loss with conventional style transfer and exemplar-based colourisation losses.

4.3 Related work

State-of-the-art in modern digital colourisation algorithms can be roughly grouped into three main paradigms: automatic learning-based, scribble-based and exemplar-based colourisation.

As described in Chapter 3, automatic learning-based methods can perform colourisation with end-to-end architectures which learn the direct mapping of every grayscale pixel to the colour space. Such approaches require large image datasets to train the network parameters without user intervention. However, in most cases they produce results which aren't colourful due to treating the colourisation process as a regression problem. As identified in the literature, well-designed loss functions such as adversarial loss [2, 35], classification loss [76, 77] or perceptual loss [38] or their combination with regularisation [42] is needed to better capture the colour distribution of the input content and enable more colourful results. A different approach is proposed in PixColor [78], solving the automatic colourisation task as an autoregressive problem. Such methods predict the colour distribution of every pixel by conditioning to the grayscale input and the joint colour distribution of previous pixels. Similarly, ColTran [79] addresses the same methodology by using an axial transformer [75]. Autoregressive methods become impractical for colourisation due to the high dimensionality of the colour distribution and the related complexity of decoding high resolution images. For instance, even for modelling 8-bit RGB inputs only, the model needs to predict 256^3 values.

Scribble-based colourisation interactively propagates initial strokes or colour points annotated by the user to the whole grayscale image. An optimisation approach [62] is proposed to propagate the user hints by using an adaptative clustering in the high dimensional affinity space. Alternatively, a Markov Random Field for propagating the scribbles [4] is proposed under the rationale that adjacent pixels with similar intensity should have similar colours. Finally, a deep learning approach [60] fuses low-level cues along with high-level semantic information to propagate the user hints.

Exemplar-based colourisation uses a colour reference to condition the prediction process. An early approach proposed the matching of global colour statistics [68], but yielded unsatisfactory results since it ignored local spatial information. More accurate approaches considered the extraction of correspondences at different levels, such as pixels [80], super-pixels [72, 67], segmented regions [55, 69, 63] or deep features [5, 74]. Based on the extraction of deep image analogies from a pre-trained VGG-19 network [51], a deep learning framework uses previously computed similarity maps to perform exemplar-based colourisation [5]. Such a method is posteriorly extended to video colourisation using a temporal consistency loss to enforce temporal coherency [5]. An alternative approach proposed the use of style transfer techniques based on AdaIN [81] to generate an initial stylised version which is further refined with a colourisation network [64]. Finally, a novel framework was proposed to fuse the semantic colours and global color distribution of the reference image towards the prediction of the final colour images [65].

Finally, the architecture presented in this chapter adopts axial attention to reduce the complexity of the overall system. As introduced in the axial transformer [75], attention is performed along a single axis, reducing the effective dimensionality of the attention maps and hence the complexity of the overall transformer. Such an approach managed to approximate conventional attention by focusing sequentially to each of the dimensions of the input tensor. An application was proposed to perform panoptic segmentation [82], integrating axial attention modules into a modified version of DeepLab [15], and improving the original baseline.

4.4 Proposed method

Aiming at exemplar-based colourisation, the goal of this method is to enable the colourisation of a grayscale target $T_L \in \mathbb{R}^{1\times H\times W}$ based on the colour of a reference $R_{Lab} \in \mathbb{R}^{3\times H\times W}$, where $H \times W$ is the image dimension in pixels, represented in the *CIE Lab* colour space [44]. Note that the target's L index refers specifically to the luminance channel. To achieve this, an exemplar-based colourisation network is trained to model the mapping $\hat{T}_{ab} = F(T_L \mid R_{Lab})$ to the target *ab* colour channels, conditioned to the reference R_{Lab} channels. *CIE Lab* colour space is chosen as it

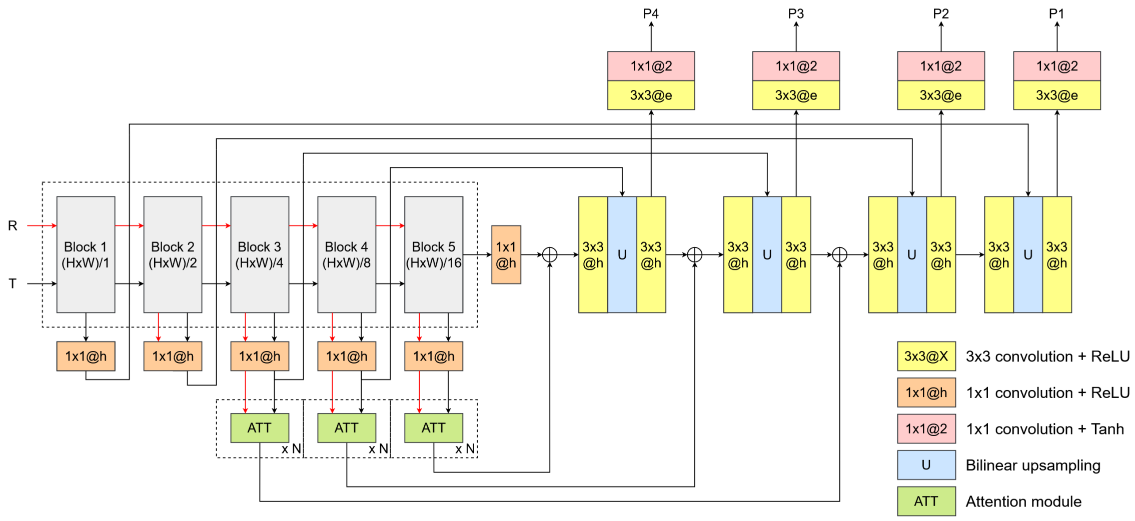


Figure 4.2: XCNET: proposed architecture for exemplar-based image colourisation. T is the black and white frame with luma component only, and R is colour reference. Multi-scale outputs $P1$-$P4$ are used for training, where $P1$ are the colourised image components at targeted resolution.

is designed to maintain perceptual uniformity and is more perceptually linear than other colour spaces [44]. This work assumes a normalised range of values between $[-1, 1]$ for each of the Lab channels.

4.4.1 Exemplar-based Colourisation Network

As shown in Figure 4.2, the proposed architecture is composed of four parts: the feature extractor backbone, the axial attention modules, the multi-scale pyramid decoder and the prediction heads.

First, both the target T_L and the reference R_{Lab} images are fed into a pre-trained feature extractor backbone to obtain L multi-scale activated feature maps F_T^l, F_R^l in an intermediate position of the l convolutional block, where $l = \{1 \dots L\}$, and the last activated feature map only for the target input F_T^B, which is the output of the backbone. Note that the features have progressively coarser volumes with increasing levels. Without loss of generality, the experiments in this paper consider a VGG-19 network pre-trained on ImageNet [50], extracting features F_T^l and F_R^l from the first Rectified Linear Unit ($ReLU$) activation of every convolutional block (*relu*$\{l\}$*_1* from VGG-19), and F_T^B which is the output of the encode (from *relu*$\{5\}$*_3* in VGG-19). Note that in order to feed T_L into the pre-trained network, the luminace channel is triplicated to obtain a 3-dimensional input space. Then, all F_T^l, F_R^l pairs and F_T^B are projected onto a h-dimensional space by means of a 1×1 convolution plus $ReLU$ activation [83], to obtain $\hat{F}_T^l$, $\hat{F}_R^l$ and $\hat{F}_T^B \in \mathbb{R}^{h \times H_l \times W_l}$, respectively.

Next, N pairs $(\hat{F}_T^l, \hat{F}_R^l)$, where $l = \{L - N + 1 \dots L\}$, are fed into N axial attention modules to compute a multi-head attention mask describing the deep correspondences between both sources. Then, the style of the reference source is transferred into the content of the target source by matrix multiplication of the attention mask with the reference source. Section 4.4.2 describes the axial attention module in depth and provides more information about the logic behind style transfer via attention. This process yields N h-dimensional fused feature maps $\hat{F}^l_{TR}$.

After generating the multi-scale fused features, a multi-scale pyramid decoder composed on $L-1$ stacked decoders and prediction heads is employed to map $\hat{F}^B_T$ into $L-1$ colour predictions at different scales using the corresponding fused features $\hat{F}^l_{TR}$. Thus, starting with $O^5 = \hat{F}^B_T$, each decoder $l = \{4, 3, 2, 1\}$ performs a five-fold operation: (1) adds F^l_{TR} with the output of the previous decoder O^{l-1}, (2) applies a 3×3 convolution plus *ReLU* activation, (3) upsamples the resultant feature map by a factor of 2, (4) similar to the *U-Net* architecture [46] concatenates the resultant upsampled map with the projected target feature map F^l_T as skip connection and (5) refines the resultant map with another 3×3 convolution plus *ReLU* activation which projects back the concatenated volume of $2h$ dimensions into the initial h dimensions, yielding an output volume O^l.

Finally, the prediction heads map the decoded feature volumes O^l into the output channels $\hat{T}^l_{ab}$. Each prediction head is composed of an e-dimensional 3×3 convolution plus *ReLU* activation and 1×1 convolution plus hyperbolic tangent (*Tanh*) activation to generate the *ab* colour channels.

4.4.2 Axial attention for unsupervised style transfer

Given two projected sources of features $\hat{F}^l_T$, $\hat{F}^l_R$ relative to the target and reference respectively, the goal of the axial attention module is to combine them in a way that the style codified in the reference features is transferred into similar content areas within the target features.

Style transfer between two sources of features has been solved in many different ways, although in most cases only artistic style is targeted without contemplating the semantic analogies between both sources. Some strategies include the use of perceptual losses for training feed-forward networks for image transformation [52], in order to encourage the transformed images to produce similar features to the style reference when both are fed into a pre-trained loss network (e.g. VGG-16). A faster strategy is to use Adaptative Instance Normalisation (AdaIN) [81] to align the mean and variance of the content features with those of the style features. Finally, another paradigm tackles deep image analogies for multi-scale visual attribute transfer [84], but the analogy computation and transfer process are performed via a *PatchMatch* algorithm [85], which is computationally expensive.

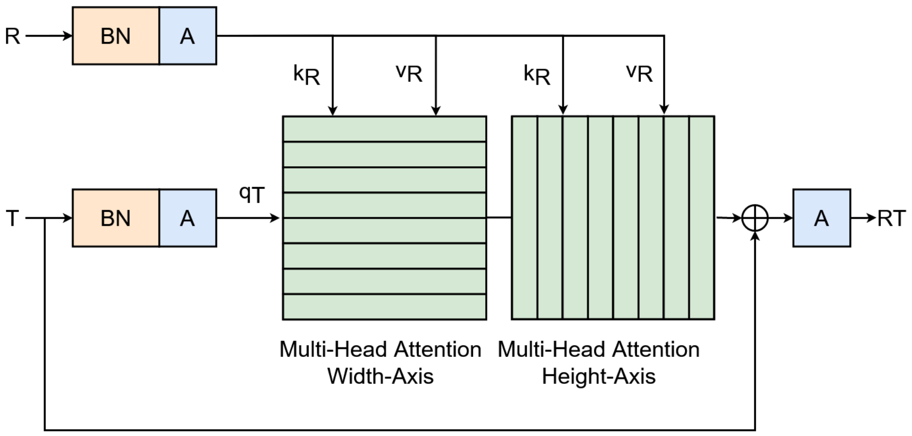

R
BN
A
kR
vR
kR
vR
T
BN
A
qT
A
RT
Multi-Head Attention Width-Axis
Multi-Head Attention Height-Axis

Figure 4.3: Axial attention module described in Section 4.4.2. In the figure, BN means Batch Normalisation and A activation.

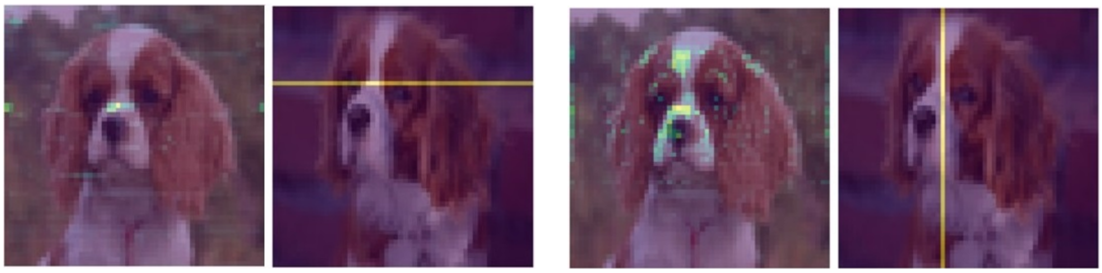
This work proposes the use of attention to perform such processes faster and in an unsupervised way. In contrast with image analogy methods based on *PatchMatch* algorithms, attention does not need to constrain to a specific local search technique (even if it can act as a set of long-term deformable kernels) nor the similarity metric (e.g. correlation loss, cosine similarity) since the module learns it automatically. Attention was introduced to tackle the problem of long-range interactions in sequence modelling [29, 86, 87, 34]. However, attention modules have been employed recently to improve computer vision tasks such as object detention [88] or image classification [31], by providing contextual information from other sources of information. Following the same rationale, an attention mechanism can solve the semantic analogy problem in style transfer by focusing on the most relevant areas of the style source when decoding each voxel in the content source.

Following the original definition of stand-alone attention [29, 89, 90], given a projected target and reference feature maps $\hat{F}^l_T$, $\hat{F}^l_R$, the fused feature map at position $o = (i, j)$, $\hat{F}^l_{TR(o)}$ is computed as follows:

$$\hat{F}_{TR(o)} = \sum_{p \in \mathcal{N}^l} softmax_p(\hat{q}^{l\intercal}_{T(o)} \hat{k}^l_{R(p)}) \hat{v}^l_{R(p)}, \tag{4.1}$$

where $\mathcal{N}^l \in \mathbb{R}^{H_l \times W_l}$ is the whole 2D location lattice. Furthermore, the queries to the target source $\hat{q}^l_{T(o)} = W_q \hat{F}^l_{T(o)}$ and the keys and values from the reference source $\hat{k}^l_{R(o)} = W_k \hat{F}^l_{R(o)}$, $\hat{v}^l_{R(o)} = W_v \hat{F}^l_{R(o)}$ are all linear projections of the target and reference projected sources $\hat{F}^l_{T(o)}$ and $\hat{F}^l_{R(o)}$, respectively, $\forall o \in \mathcal{N}$, where $W_q, W_k, W_v \in \mathbb{R}^{h \times h}$ are all the learnable parameters. The $softmax_p$ denotes a softmax operation applied to all possible p positions within the 2D lattice $\mathcal{N}^l$.

Next, a position-sensitive learned positional encoding [91, 92, 82] is adopted to encourage the attention modules to model dynamic prior of where to look at in the



Row attention

Column attention

Figure 4.4: Visualisation of the axial-attention masks in the target image for a given row and column in the reference image. Note that column attention expands the input row-dependent activated space into a 2D lattice.

receptive field of the reference source ($m \times m$ region within $\mathcal{N}^{l}$). Positional encoding has proven to be beneficial in computer vision tasks to exploit spatial information and capture shapes and structures within the sources of input features. Therefore, as in [82], a key, query and value dependent positional encoding are applied to Equation 4.1 as follows:

$$\begin{aligned} \hat{F}_{TR(o)} = \sum_{p \in \mathcal{N}^{l}_{m\times m(o)}} & softmax_{p}(\hat{q}^{l\intercal}_{T(o)}\hat{k}^{l}_{R(p)} + \hat{q}^{l\intercal}_{T(o)} r^{q}_{(p-o)} + \\ & + \hat{k}^{l\intercal}_{T(p)} r^{k}_{(p-o)})(\hat{v}^{l}_{R(p)} + r^{v}_{(p-o)}), \end{aligned} \tag{4.2}$$

where $\mathcal{N}^{l}_{m\times m(o)}$ is the local $m \times m$ local region centred around location $o = (i, j)$, and $r^{q}_{(p-o)}$, $r^{k}_{(p-o)} and\ r^{v}_{(p-o)}$ the learned relative positional encoding for queries, keys and values, respectively. The inner products $\hat{q}^{l\intercal}_{T(o)} r^{q}_{(p-o)}$ and $\hat{k}^{l\intercal}_{T(p)} r^{k}_{(p-o)}$ measure the compatibilities from location p to o within the queries and keys space, and $r^{v}_{(p-o)}$ guides the output $\hat{F}_{TR(o)}$ to retrieve content within the values space.

Finally, axial attention [75] is adopted to reduce the complexity of the original formulation $\mathcal{O}(H_l W_l m^2)$ to $\mathcal{O}(H_l W_l m)$ by computing the attention operations along a 1-dimensional axial lattice $1 \times m$, instead of across the whole $\{m \times m\}$ space. Following the formulation as in stand-alone axial-DeepLab [82], the global attention operation is simplified by defining an axial-attention layer that propagates the information along the width-axis followed by another one along the height-axis. In this work, we set a span $m = \{H_l, W_l\}$ equal to the input image resolution ($\mathcal{O}(H_l W_l m)$), but such values can be reduced for high resolution inputs. Finally, multi-head attention can be performed by applying N single axial attention heads with head-dependent projections W^{n}_{q}, W^{n}_{k}, W^{n}_{v}, posteriorly concatenating the results of each head and projecting the final output maps by means an output 1×1 convolution.

As shown in Figure 4.3, a succession of multi-head *weight-height* axial attention layers are integrated to design the axial attention module for unsupervised style transfer. Given F_T^{lh}, F_R^{lh} inputs, such module performs a three-fold operation: (1) normalise the target and reference projected sources by means of batch normalisation plus $ReLU$ activation, (2) fuse the normalised sources by means of the multi-head *weight-height* axial attention layers, and (3) add resulting features to the target source identity F_T^{lh} plus activate the output with a $ReLU$ activation.

4.4.3 Training losses

Usually, the objective of colourisation is to encourage that the predicted $\hat{T}_{ab}$ colour channels are as close as possible to the ground truth T_{ab} in the original training dataset. However, this objective does not apply in exemplar-based colourisation, where $\hat{T}_{ab}$ should be customized by the colour reference R_{Lab} while preserving the content of the grayscale target T_L. Therefore, the definition of the training strategy is not straightforward, as penalising $\hat{T}_{ab}$ and T_{ab} is not accurate. Then, the objective is to enable the reliable transfer of reference colours to the target content towards obtaining a colour prediction faithful to the reference. This work takes advantage of the pyramidal decoder to combine state-of-the-art exemplar-based losses with an adversarial training at multiple resolutions. Hence, a multi-loss training strategy is proposed to combine a *smooth*-L_1 loss, a colour histogram loss and a total variance regularisation, as in [71], with a multi-scale adversarial loss by means of multiple patch-based discriminators [2]. In order to handle multi-scale losses, average pooling with a factor of 2 is applied to both target and reference to successively generate the multi-scale ground truth T_{ab}^l and R_{ab}^l.

***Smooth*-L_1 loss.** In order to induce dataset priors in cases when the content of the reference is significantly different than the the target, a pixel loss L_{pixel}^l based on Huber loss [93] (also known as *smooth*-L_1) is proposed to encourage realistic predictions. L_{pixel}^l loss can be summarised as follows:

$$\begin{gathered} L_{pixel}^l(T^l, \hat{T}^l) = \frac{1}{HW} \sum_i \sum_j z(i,j) \\ z = \begin{cases} \frac{1}{2}(T^l - \hat{T}^l)^2, & \left|T^l - \hat{T}^l\right| < 1 \\ \left|T^l - \hat{T}^l\right| - \frac{1}{2}, & \text{otherwise} \end{cases}. \end{gathered} \tag{4.3}$$

Colour histogram loss. In order to fully capture the global colour distribution of the reference image $\mathcal{H}_R^l$ and penalise the differences in respect to the predicted colour distribution $\mathcal{H}_{\hat{T}}^l$, a colour histogram loss is considered. Without loss of generality, the following describes how to approximate the target histogram $\mathcal{H}_{\hat{T}}^l$, but the same formulation can be applied to $\mathcal{H}^l_R$. Following the method in [65], the *ab* colour space in the range $[-1, 1]$ is quantised using a step d. Subsequently, quantised colour histograms $A^l_{\hat{T}}$, $B^l_{\hat{T}}$ are obtained for the channels a and b respectively. Then, the target histogram $\mathcal{H}^l_{\hat{T}}$ is computed as follows:

$$\mathcal{H}^l_{\hat{T}} = \frac{1}{H_l W_l} \left(A^l_{\hat{T}}\right)^\intercal B^l_{\hat{T}}. \tag{4.4}$$

Finally, the histogram loss L^l_{hist} is defined as a symmetric χ^2 distance [94] as follows:

$$L^l_{hist} = 2 \sum_{q=1}^{Q} \frac{\left(\mathcal{H}^l_{\hat{T}}(q) - \mathcal{H}^l_R(q)\right)^2}{\mathcal{H}^l_{\hat{T}}(q) + \mathcal{H}^l_R(q) + \epsilon}, \tag{4.5}$$

where ϵ prevents infinity overflows and $Q = \left(\frac{2+d}{d}\right)^2$ is the number of histogram bins. In this work, $\epsilon = 10^{-5}$, $d = 0.1$ and hence $Q = 441$.

Total variance regularisation. In order to encourage low variance along neighbouring pixels of the predicted colour channels $\hat{T}^l$, the total variance loss L^l_{TV} is computed as follows:

$$\begin{aligned} L^l_{TV} = &\sum_i \sum_j \left(\left|\hat{T}^l(i+1, j) - \hat{T}^l(i, j)\right| + \right. \\ &\left. + \left|\hat{T}^l(i, j+1) - \hat{T}^l(i, j)\right| \right). \end{aligned} \tag{4.6}$$

Adversarial loss. With the aim to guide the previous losses towards realistic decisions, an adversarial strategy based on *LS-GAN* [26] is proposed, using the ground truth colour targets T^l_{Lab} as original sources and a patch-based discriminator D as described in Table 4.1. Note that within the GAN framework, the proposed exemplar-based colourisation network would be the generator. Both generator loss L^l_G and discriminator loss L^l_D are computed as follows:

$$\begin{aligned} L^l_D = &\frac{1}{2}\mathbb{E}_{T^l \sim P_T} \left[\left(D(T^l) - 1\right)^2\right] + \\ &+ \frac{1}{2}\mathbb{E}_{\hat{T}^l \sim P_{\hat{T}}} \left[D(\hat{T}^l)^2\right] \\ L^l_G = &\frac{1}{2}\mathbb{E}_{\hat{T}^l \sim P_{\hat{T}}} \left[\left(D(\hat{T}^l) - 1\right)^2\right]. \end{aligned} \tag{4.7}$$

The total discriminator loss L_D is computed by adding the L individual multi-

Layer	Kernel	Stride	Dim in	Dim out
Conv2D Leaky ReLU	4×4	2	3	64
Conv2D Batch Norm Leaky ReLU	4×4	2	64	128
Conv2D Batch Norm Leaky ReLU	4×4	2	128	256
Conv2D Batch Norm Leaky ReLU	4×4	1	256	512
Conv2D	1×1	1	512	1

Table 4.1: Discriminator architecture

scale losses as follows:

$$L_D = \sum_{l=1}^{L} L_D^l. \tag{4.8}$$

Finally, the total multi-scale loss L_{total} is computed as:

$$\begin{aligned} L_{total} = &\sum_{l=1}^{L}(\lambda_{pixel} L_{pixel}^l + \lambda_{hist} L_{hist}^l + \\ &+ \lambda_{TV} L_{TV}^l + \lambda_G L_G^l), \end{aligned} \tag{4.9}$$

where λ_{pixel}, λ_{hist}, λ_{TV} and λ_G are the multi-loss weights which specify the contribution of each individual loss.

Finally, the total multi-scale loss L_{total} is computed as:

$$\begin{aligned} L_{total} = &\sum_{l=1}^{L}(\lambda_{pixel} L_{pixel}^l + \lambda_{hist} L_{hist}^l + \\ &+ \lambda_{TV} L_{TV}^l + \lambda_G L_G^l), \end{aligned} \tag{4.10}$$

where λ_{pixel}, λ_{hist}, λ_{TV} and λ_G are the multi-loss weights which specify the contribution of each individual loss.

4.5 Experiments

4.5.1 Training settings

A training dataset based on *ImageNet* [50] is generated by sampling $225,000$ images from the 750 most popular categories (300 images per class), which include:

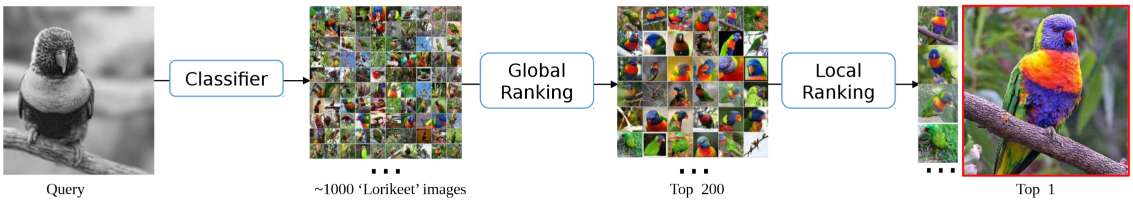


Figure 4.5: Recommendation pipeline in [5] used to extract colour references from Imagenet dataset. In case the target class is known, the classification step is avoided and the global ranking starts over the given class subset. Note that the method in [5] uses a global ranking of 200 images whilst in this work a top-5 ranking is considered.

animals, plants, people, scenery, food, transportation and artifacts. Pairs of target-reference images are randomly generated based on the correspondence recommendation pipeline proposed in [5]. As shown in Figure 4.5, a top-5 global ranking is first created by minimising the L_2 distance between the features of the target and the rest of the of the same class, extracted at the first fully connected layer of a pre-trained VGG-19 with ImageNet and projected into 128 dimensions via PCA transformation [95]. Next, following the process in [5], the global ranking is refined by a local search selecting the most similar image by means of a patch-based similarity. The top-1 reference is selected by minimising the cosine distance between 16×16 patches corresponding to the most similar position-wise feature vector at the *relu*{*4*}*_3* space of the same pre-trained VGG-19, from both target and reference candidate. Finally, pairs of target-reference images are randomly sampled on-the-fly during training by using a weighted uniform distribution of 3 categories, with a weight α_c: top-1 reference ($\alpha_1 = 0.6$), random choice among the top-5 candidates ($\alpha_2 = 0.3$) and random choice among the rest of images of the same class ($\alpha_3 = 0.1$). Testing data is generated in a similar way, sampling $45,000$ pairs of target-reference images from the training subset (different targets than in training) at the same categories (60 images per class). All images are resized to 224×224 pixels, converted to the CIE Lab colour space and normalised into the range $[-1, 1]$ for each channel.

All the experiments use multi-head attention layers of 8 heads, a hidden dimension $h = 256$ and a prediction head dimension $e = 64$. As shown in the Figure 4.2, a backbone with 5 convolutional blocks is used, starting the decoding process from a resolution of $(H \times W)/16$ pixels and decoding 4 different multi-scale predictions. Although several ablations are performed, the best trade-off between complexity and performance is achieved by applying the attention modules from the block 3. All models are trained around 30 epochs using an Adam optimiser [96] with a learning rate of 10^{-5}. The multi-loss weights $\lambda_{pixel} = 100$, $\lambda_{hist} = 2$, $\lambda_{TV} = 50$ and $\lambda_G = 1$ are used for all the experiments. Finally, all models are implemented in Pytorch 1.7.0 [97] and trained with a single GPU using a batch size of around $4-12$

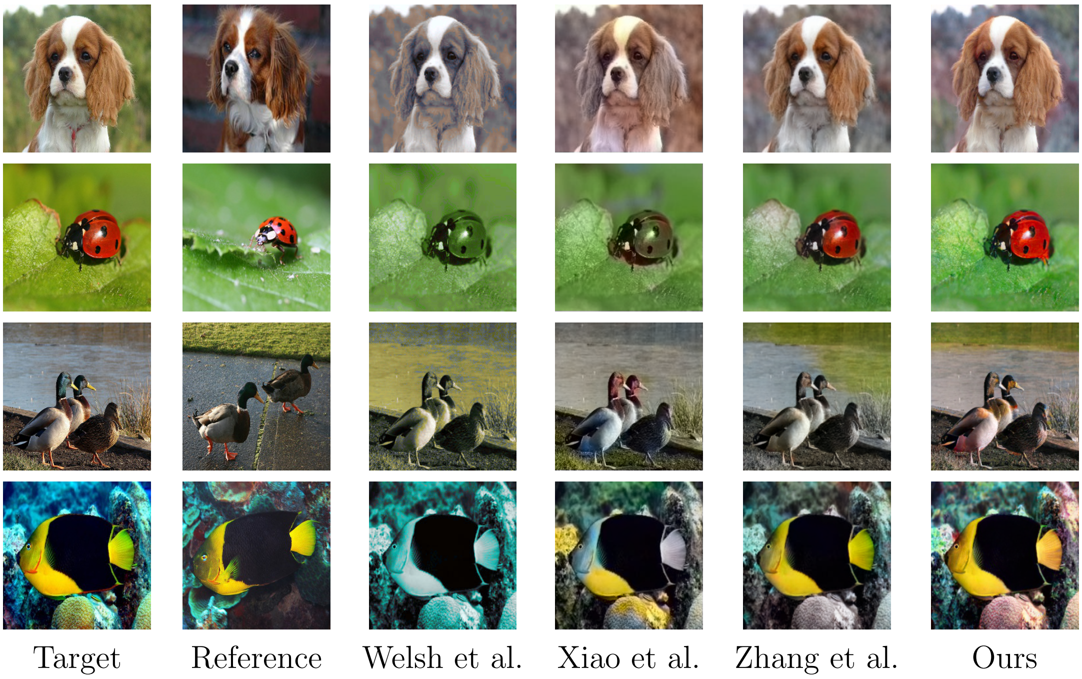


Figure 4.6: Qualitative comparison of the existing and the proposed exemplar-based colourisation methods.

samples.

4.5.2 Comparison with colourisation methods

In order to compare our approach with existing exemplar-based colourisation methods [74, 73, 68], a test dataset is collected by randomly sampling $5,000$ target-reference pairs from the validation set defined in Section 4.5.1. To provide a fair comparison, all results are obtained by running the original publicly available codes and models provided by the authors.

A qualitative comparison for a selection of representative cases is shown in Figure 4.6. From this comparison, our method along with Zhang et al. [74] produce the most visual appealing results, being able to transfer effectively the colours from the reference. Both methods show that image analogy methodology better captures local information from semantically related objects and leads to more precise colour predictions. On the contrary, the methods from Welsh et al. [68] and Xiao et al. [73], based on global histogram estimation, fail to detect precise patterns and only map overall tones from the reference. The proposed multi-loss strategy, incorporating histogram and adversarial loss at different resolutions, enables more colorful and saturated results. However, unlike the conservative colourisation of [74], the instability of the adversarial training can lead to some colour noise, as can be seen in the 4th row of Figure 4.6. A better control of the adversarial loss could boost our

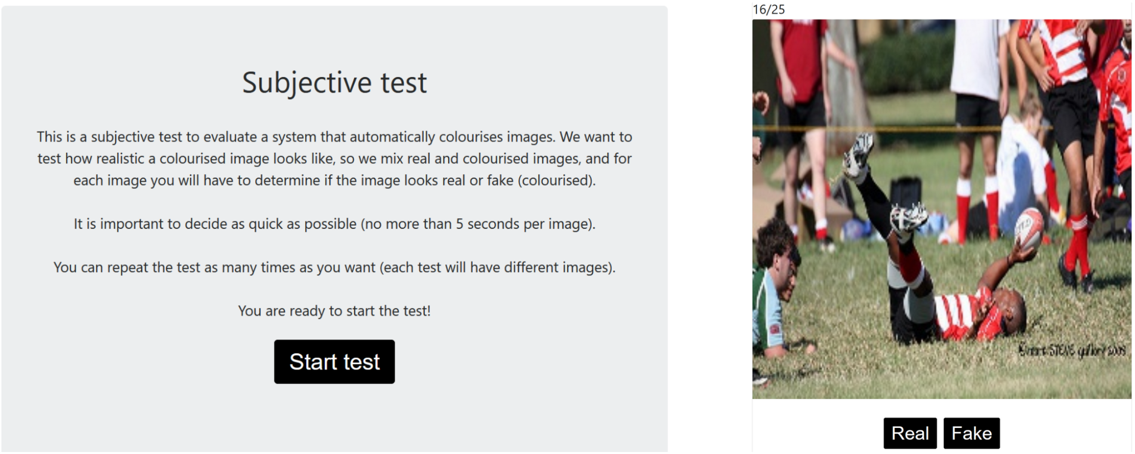
Method	HIS	SSIM	Top-1 Acc	Top-5 Acc
Welsh et al. [68]	0.55	0.78	50.3%	74.1%
Xiao et al. [73]	0.59	0.84	54.8%	79.2%
Zhang et al. [74]	0.66	**0.88**	65.6%	84.8%
Ours axial att.	0.72	0.87	68.1%	89.1%
Ours standard att.	**0.74**	**0.88**	**69.7%**	**90.1%**
Ours single module	0.68	**0.88**	67.6%	88.9%
Ours w/o adv. loss	0.70	**0.88**	67.5%	89.2%
Ours w/o pix. loss	0.68	0.86	67.5%	86.7%
Ours w/o hist. loss	0.54	**0.88**	65.4%	89.2%

Table 4.2: Quantitative comparison of the state-of-the art methods with the proposed method in different settings. Note that standard attention is only used in the ablation study, the rest of our combinations use axial attention.

method's performance, reaching the stability of [74] while producing more colourful and visually appealing predictions.

Moreover, a quantitative comparison is shown in Table 4.2, using three different metrics: Histogram Intersection Similarity (HIS) [2] relative to the reference image, Structural Similarity Index Measure (SSIM) relative to the target ground truth image and classification accuracy. First, HIS score measures the averaged colour histogram intersection between the reference and predicted images. As shown in the results, our method along with [74], which are both based on semantic-related analogies, achieve higher HIS scores suggesting a better transfer of the reference colours. On the contrary, the methods in [73] and [68], based on global histogram estimation, slightly lower the HIS score due to the averaged colourisation in ambiguous cases where the target and reference objects are not recognised. SSIM score is used to estimate structural similarity of each method. As can be observed, the methods achieving a more precise colourisation obtain higher SSIM score. The method in [74] achieves the same score as ours, suggesting that more stable predictions help to better retain the structural information of the target image. Finally, our method outperforms all other methods on image recognition accuracy when the colour predictions are fed into a VGG-16 pre-trained on ImageNet. The obtained results indicate that the proposed method overall outperforms previous methods, which is also reflected by the classification performance.

In addition to qualitative and quantitative comparisons, a perceptual test is performed to validate overall results and to detect possible failure cases. A total of 100 target-reference pairs are randomly sampled from the test dataset and colourised using our method and the three state-of-the-art methods [74, 73, 68]. Therefore, 500 images are generated, including 100 original images and 400 images which the



Subjective test

This is a subjective test to evaluate a system that automatically colourises images. We want to test how realistic a colourised image looks like, so we mix real and colourised images, and for each image you will have to determine if the image looks real or fake (colourised).

It is important to decide as quick as possible (no more than 5 seconds per image).

You can repeat the test as many times as you want (each test will have different images).

You are ready to start the test!

Figure 4.7: Visualisation of the user interface developed in this project for collecting the subjective test results.

Method	Naturalness (%)
Real images	85.51%
Ours	**61.30%**
Zhang et al. [74]	60.04%
Xiao et al. [73]	46.10%
Welsh et al. [68]	36.10%

Table 4.3: Perceptual test results. The values show the percentage of images selected as genuine (natural) for each of the methods.

colour is predicted. Each individual test session randomly selects 25 images and shows them one by one to viewers, which included participants with technical and non-technical backgrounds. Then, each participant has to annotate if the colours in each image appear to be genuine (natural) or not. The study was performed 190 times, generating a total of 4, 750 annotations. Figure 4.7 shows the user interface developed to collect the results. Table 4.3 shows the percentage of the annotations that evaluated colours as genuine with reference to the total number of all the annotations for the specific method, our approach (61.30%) slightly outperforms the method in [74] (60.04%). As discussed in the visual comparison, the potential production of colour noise might have lowered the performance of our method. In contrast, the stability of [74] enabled a considerably high rate despite its conservative colourisation. Finally, the methods in [73] (46.10%) and [68] (36.10%) achieve the lowest results.

Finally, the runtime is also compared to highlight the efficiency of the proposed end-to-end architecture. All the results are obtained using the implementation provided by the authors. Runtime values are obtained on a machine with 3.60GHz Intel Xeon Gold 5122 CPU and a single NVIDIA GeForce RTX 2080 Ti GPU. As shown

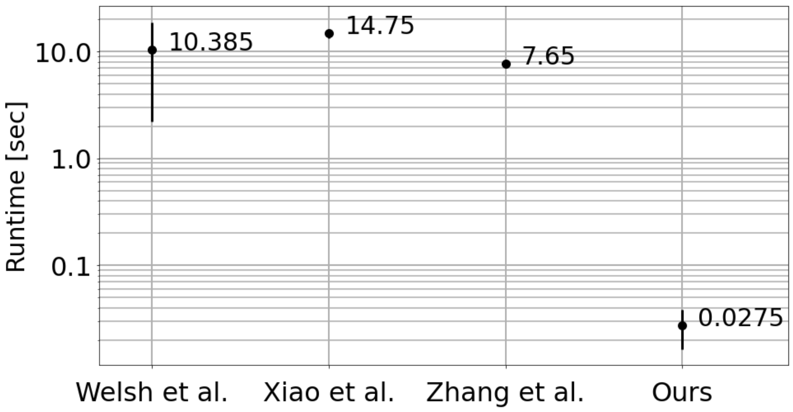


Figure 4.8: Runtime comparison in seconds.

in Figure 4.8, among neural network based methods, the pyramid structure in Xiao et al. [73] costs most of the time. The method from Zhang et al. [74] slightly reduces runtime but the Patch Match search used in Deep Image Analogy [84] consumes a lot of time. On contrast, our end-to-end approach significantly reduces complexity achieving runtimes of 20 ms per image.

4.5.3 Ablation study

Several experiments are performed with the aim to evaluate the effects of the different architectural choices and training hyperparameters. The ablation study includes the analysis of the attention module, comparing the performance obtained with the standard attention operation and the proposed axial attention simplifications in Section 4.4.2. Moreover, the number of attention operations at each scale is also evaluated. Finally, the contribution of each of the training losses is validated by removing them separately from the total multi-loss function and study their effects on the final predictions. As discussed in Section 4.4.2, axial attention is adopted to reduce the complexity of the original attention formulation $\mathcal{O}(H_l W_l m^2)$ to $\mathcal{O}(H_l W_l m)$ by computing the attention operations along a single 1-dimensional axis, instead of across the whole $\{m \times m\}$ space. Although axial attention is applied to both the horizontal and vertical axis to approximate the standard performance, a significant loss is identified in Table 4.2. A visual comparison is shown in Figure 4.10, where standard attention yields to more precise results, being able to capture longer-term relationships. In order to refine the axial approximation and being able to derive more complex relationships, the attention module is applied 2 consecutive times. As shown, such approach outperforms the single configuration in both quantitative and qualitative evaluations. Finally, the individual contribution of each training loss is evaluated by removing them one by one from the multi-loss configuration. As shown in Table 4.2 and Figure 4.9, a major drop in HIS score is identified in the absence of

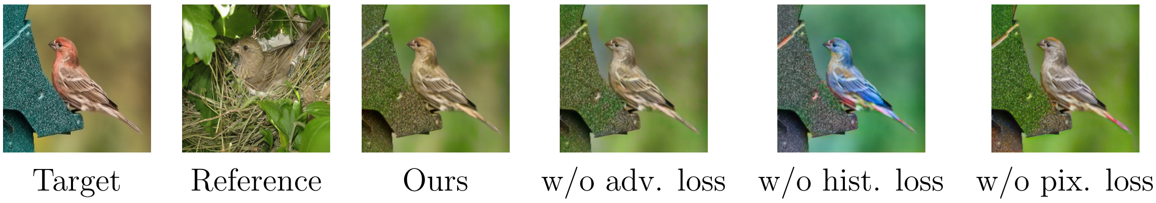


Figure 4.9: Visual comparison of each individual training loss contribution.

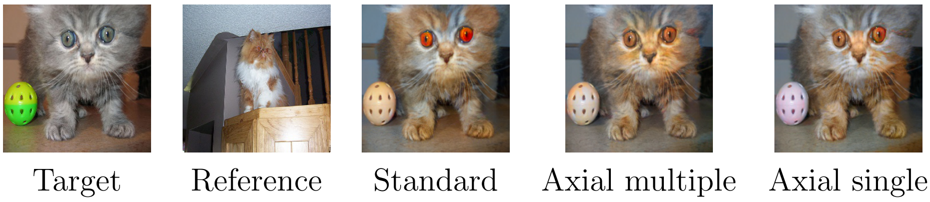


Figure 4.10: Visual comparison of the attention module configurations, using standard attention or axial attention one time or two times.

the histogram loss, indicating its importance to guide the learning process towards an effective transfer of reference colours. The absence of the adversarial loss also lowers the performance, dropping by 0.2 the HIS score and 0.6% the Top-1 accuracy. However, a higher effect is shown in the visual comparison, where a clear loss of both colourfulness and naturalness can be observed.

4.6 Future work

Several directions can be considered to expand the current work towards different research lines and applications:

- The XCNET architecture can be improved by further tweaking the attention module. A cost-effective solution can be investigated to improve attention performance (keeping the axial approach or proposing a better methodology) without increasing the overall complexity. Moreover, further improvements can be obtained by improving the feature extraction backbone, e.g. by investigating the effect of the input colour space and possibly use a different backbone for the grayscale target and colour references. Finally, a better performance could possibly be obtained by enabling multiple reference inputs.

- The XCNET architecture could be generalised to other applications and domains aiming at unsupervised analogy computation between pairs of inputs. Some examples include video frame interpolation, video compression or any other reference-based approach.

- In order to design more realistic colourisation and restoration frameworks, the current approach needs to allow high resolution inputs and to be adapted to video content. The next chapter presents preliminary insights in this direction and proposes future research lines.

4.7 Conclusions

This chapter introduces XCNET, a novel architecture for exemplar-based colourisation. The proposed model integrates attention modules at different resolutions that learn how to perform style transfer in an unsupervised way towards decoding realistic colour predictions. This methodology significantly simplifies previous exemplar-based approaches, unifying the feature matching with the colourisation process and therefore achieving a fast end-to-end colourisation. Moreover, in order to further reduce the model complexity, axial attention is proposed to simplify the standard attention operations and hence reduce the computation intensity. The proposed method outperforms state-of-the-art methods in both visual quality and complexity, and significantly reduces the runtime.

Chapter 5

A pipeline for video colourisation

5.1 Chapter overview

This chapter studies how to integrate the proposed image-based colourisation architectures (such as ColorGAN from Chapter 3 and XCNET from Chapter 4) into a practical video colourisation pipeline. In particular, we focus on temporal stabilisation for such frame-to-frame image colourisation methods proposing an optimised few-shot training strategy to learn scene-aware video priors. The work described in this chapter was published at the ECCV 2022 Workshop on AI for Creative Video Editing and Understanding (CVEU 2022). Section 5.3 reviews related work in the literature, identifying the main draw-backs and possible improvements, Section 5.4 details the proposed methodology, Section 5.5 provides information about the implementation and data used in the experiments and a quantitative evaluation of the results. Section 5.6 proposes an ultimate video colourisation pipeline combining the proposed image-based colourisation methods with the introduced temporal stabilisation framework. Finally, Section 5.7 provides conclusions and identifies future work.

5.2 Introduction

Video restoration is in increasing demand in the production industry in order to both deliver historical content in high quality and to support innovation in the creative sector [98]. Video colourisation in particular is still a challenging task due to its ambiguity in the solution space and the requirement of global spatio-temporal consistency. Prior to automatic colourisation methods, producers relied on specialists to perform manual colourisation, resulting in a time consuming and sometimes a prohibitively expensive manual process. Researchers have thus endeavoured to develop computer-assisted methodologies in order to automate the colourisation process and reduce production costs. Early methods relied on frame-to-frame image colourisation techniques propagating colour scribbles [4, 57, 56] or reference colours [68, 63, 69]. The problem that typically occurs when processing is applied on a single frame without consideration of the neighbouring frames is temporal flickering. Similarly, propagation of errors can occur if the temporal dimension is not taken into account when characteristics (e.g colour) of previous frame are transferred to the current frame. Improved results can be obtained by considering a more robust propagation and imposing refinements with temporal constrains [74, 99].

Instead of improving temporal consistency using task-specific solutions, methods that generalise to various tasks can be applied. An example is the work in [100], which proposes a general approach agnostic to a specific image processing algorithm. The method takes the original video (black and white in the case of colourisation) and the per-frame processed counterpart (initially colourised version) and solves a gradient domain optimisation problem to minimise the temporal warping error between consecutive frames. An extension of such an approach takes into account object occlusions by leveraging information from a set of key-frames [101]. Another example was proposed in [102], adopting a perceptual loss to maintain perceptual similarity between output and processed frames. However, most methods rely on a dense correspondence backend (e.g. optical flow or PatchMatch [85]), which quickly becomes impractical in real-world scenarios due to the increased processing time needed. A novel solution proposed the use of Deep Video Prior by training a convolutional network on video content to enforce consistency between pairs of corresponding output patches [103]. The method solves multimodal consistency by means of Iteratively Reweighted Training, which learns to select a main mode among multiple inconsistent ones and discard those outliers leading to flickering artifacts. The main limitation is the requirement to train in test time, which makes the method extremely time-consuming in practice. For instance, training depends on the content, motion and length of the input video, requiring a large number of iterations to generalise to complex sequences with multiple shots and scene changes.

This paper proposes a framework for temporal stabilisation of frame-to-frame colourised videos with an optimised few-shot training strategy to learn scene-aware video priors. The proposed architecture is jointly trained to stabilise the input video and to cluster the input frames with the aim of learning scene-specific modes. Learnt embeddings are posteriorly injected into the decoder process to guide the stabilisation of specific scenes. A clustering algorithm for scene segmentation is used to select meaningful frames and to generate pseudo-labels to supervise the scene-aware training. Experimental results demonstrate the generalisation of the Deep Video Prior baseline [103], obtaining improved performance in complex sequences with small amounts of training data and fewer iterations.

5.3 Related work

5.3.1 Video colourisation

Although several works attempted to solve the video colourisation problem as an end-to-end fully automatic task [104], most rely on single frame colourisation. This is because image colourisation, compared to video colourisation, achieves higher visual quality and naturalness. Propagation methods are commonly used to stabilise the temporal coherence between frames. For instance, the work in [105] propose Video Propagation Networks (VPN) to process video frames in an adaptive manner. VPN approach applies a neural network for adaptative spatio-temporal filtering. First it connects all the pixels from current and previous frames and propagates associated information across the sequence. Then it uses a spatial network to refine the generated features. Another example is the Switchable Temporal Propagation Network [106], based on a Temporal Propagation Network (TPN), which models the transition-related affinity between a pair of frames in a purely data-driven manner. In this way, a learnable unified framework for propagating a variety of visual properties from video frames, including colour, can be achieved. Aiming at improving the efficiency of deep video processing, colourisation and propagation can be performed at once. An example is the method in [74] that is based on a recurrent video colourisation framework, which combines colourisation and propagation sub-networks to jointly predict and refine results from a previous frame. A direct improvement is the method in [99] that uses masks as temporal correspondences and hence improves the colour leakage between objects by wrapping colours within restricted masked regions over time.

5.3.2 Deep Video Prior

Methods for temporal stabilisation usually promote blind temporal consistency by means of dense matching (optical flow or PatchMatch [85]) to define a regularisation loss that minimises the distance between correspondences in the stabilised output frames [100]. Such methods are trained with large datasets with pairs of grayscale inputs and colourised frames. Notice that such frameworks are blind to the image processing operator and can be used for multiple tasks such as super-resolution, denoising, dehazing, etc. In contrast, Deep Video Prior (DVP) can implicitly achieve such regularisation by training a convolutional neural network [103]. This method only requires training on the single test video, and no training dataset is needed. To address the challenging multimodal inconsistency problem, an Iteratively Reweighted Training (IRT) strategy is used in DVP approach. The method selects one mode from multiple possible modes for the processed video to ensure temporal consistency and preserve perceptual quality.

5.3.3 Few-shot learning

Few-shot learning was introduced to learn from a limited number of examples with supervised information [107, 108]. For example, although current methods on image classification outperform humans on ImageNet [13], each class needs sufficient amount of labelled images, which can be difficult to obtain. Therefore, few-shot learning can reduce the data gathering effort for data-intensive applications [109]. Many related topics use this methodology, such as meta-learning [110, 111], embedding learning [112, 113] and generative modelling [114, 107]. The method proposed in this chapter uses few-shot learning as training strategy to reduce processing time and to generalise to long and complex video sequences.

5.4 Method

This section describes the proposed extension of DVP baseline for multiple scenes, followed by the optimised few-shot training strategy which enables reduced processing time by removing the time response of DVP conditioned to the number of input frames. Finally, DVP architecture is modified by adding a classification sub-network which clusters the input frames with the objective of learning scene-specific priors.

5.4.1 Extension of DVP to multiple scenes

Given a grayscale input sequence $\{I_t\}_{t=1}^{T}$ of T frames and its colourised counterpart $\{P_t\}_{t=1}^{T}$ created using an image colourisation operator F, the goal is to learn the mapping $\hat{G}(\theta) : \{P_t\}_{t=1}^{T} \longrightarrow \{O_t\}_{t=1}^{T}$, such that $\{O_t\}_{t=1}^{T}$ is a temporally stable output without flickering artifacts and θ are the network parameters. Due to the superior performance of image colourisation compared to video methods [100, 104], an image operator is applied frame-to-frame and the proposed framework is used to resolve temporal issues. Therefore, from a random initialisation, $\hat{G}(\theta)$ is optimised in each iteration by means of the reconstruction loss L_{data} (e.g. L_1 distance) between $\hat{G}(I_t;\theta)$ and P_t:

$$\arg\min_{\theta} L_{data}(\hat{G}(I_t;\theta), P_t). \tag{5.1}$$

As shown in Figure 5.1, the proposed method extends the DVP framework [103] for video sequence with multiple scenes. In particular, the proposed method defines a scene as a change of content, e.g. a camera shot, appearance of new objects, etc. In particular, the input sequence $\{I_t\}_{t=1}^{T}$ of T frames is divided into S scenes, where typically $S \ll T$, and $\{s_t\}_{t=1}^{T}$ is the scene index for each frame. In order

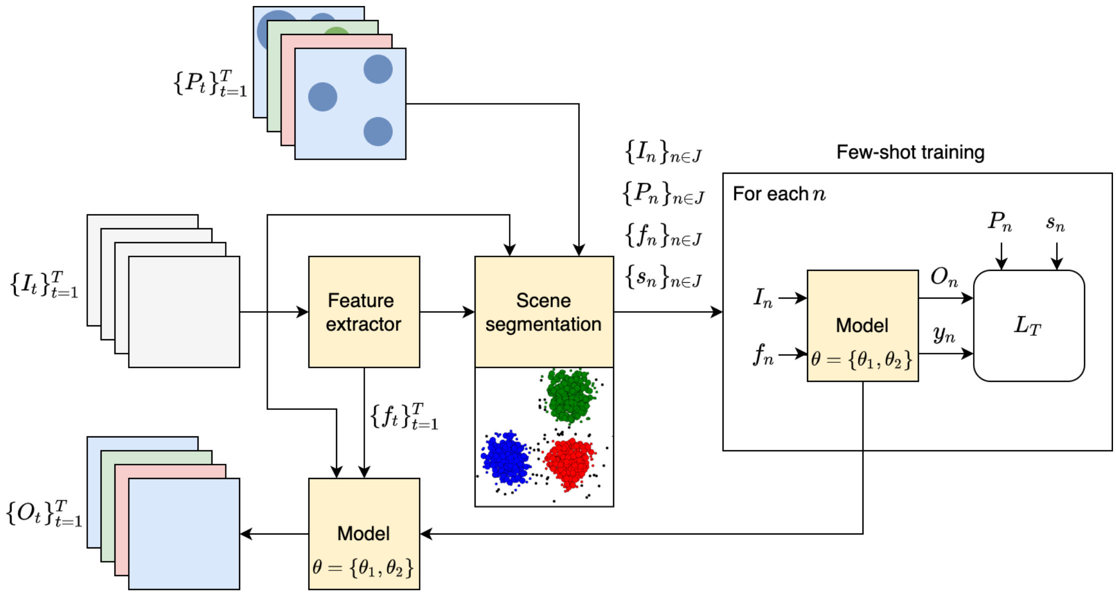


Figure 5.1: Proposed framework for temporal stabilisation of frame-to-frame colourised videos. In addition to the DVP baseline, a scene segmentation and a few show training is used to learn scene-aware video priors.

to learn scene-specific modes, the proposed network not only learns to stabilise the input sequence, but also to cluster its frames into different scenes by generating a class distribution vector $y_t \in \mathbb{R}^S$. As shown in Figure 5.2, an external feature vector f_t (from frame I_t) is provided in order to guide the clustering process. f_t can be obtained from a suitable neural network, e.g. from VGG-16 classification head [51]. Finally, y_t is used to generate scene-specific priors which are posteriorly injected into the different stages of the network decoder. Therefore, the proposed model combines two different sub-models, denoted by $\hat{G}(\theta) = \{\hat{G}_1(\theta_1), \hat{G}_2(\theta_2)\}$, where $\theta = \{\theta_1, \theta_2\}$ are all the network parameters, $\hat{G}_1(\theta_1) : \{P_t\}_{t=1}^T \longrightarrow \{O_t\}_{t=1}^T$ and $\hat{G}_2(\theta_2) : \{f_t\}_{t=1}^T \longrightarrow \{y_t\}_{t=1}^T$.

The neural network is then trained to jointly improve the temporal consistency of the input video frames $\{I_t\}_{t=1}^T$ (enforcing $\{O_t\}_{t=1}^T$ to be close to $\{P_t\}_{t=1}^T$) and classify them into the corresponding scenes $\{s_t\}_{t=1}^T$. Following DVP baseline, an IRT strategy is used to address the problem of averaging when the difference of multiple modes is large (e.g. pixel with more than one possible colourisation solution). In particular, a confidence map C_t is used to enforce the selection of a main mode per pixel from multiple modes, while it ignores the outliers (minor modes leading to flickering artifacts). In practice, DVP doubles the number of output channels (e.g. 6 channels for RGB images) to obtain two output versions: a main frame O_t^{main} and an outlier frame O_t^{minor}. The confidence map $C_{t,i}$ at iteration i is calculated by:

$$C_{t,i} = \begin{cases} 1 & d(O_{t,i}^{main}, P_t) < \max\{L_1(O_{t,i}^{minor}, P_t), \delta\} \\ 0 & \text{otherwise} \end{cases}, \tag{5.2}$$

where d is the function to measure the distance between pixels and δ is a threshold. Therefore, the model parameters at iteration $(i+1)$ can be optimised using $C_{t,i}$ which guides the training loss:

$$\begin{aligned} \theta^{i+1} = & \arg\min_{\theta}\{L_{data}(C_{t,i} \odot O_{t,i}^{main}, C_{t,i} \odot P_t) + \\ & + L_{data}((1 - C_{t,i}) \odot O_{t,i}^{minor}, (1 - C_{t,i}) \odot P_t)\}. \end{aligned} \tag{5.3}$$

Then, a multi-loss function is proposed combining the IRT loss L_{IRT} between $\hat{G}_1(I_t;\theta_1)$ and P_t, and the cross-entropy loss L_{class} between $\hat{G}_2(f_t;\theta_2)$ and s_t:

$$L_T = L_{IRT}\left(\hat{G}_1(I_t;\theta_1), P_t\right) + L_{class}\left(\hat{G}_2(f_t;\theta_2), s_t\right). \tag{5.4}$$

5.4.2 Few-shot training strategy

The main limitation of DVP is the long processing time due to the need for training at inference time. This fact makes the method impractical for long sequences. This paper proposes to speed up the training process reducing the number of iterations by means of a few-shot training strategy. Such strategy selects a reduced set of N frames $\{I_n\}_{n\in J} \subset \{I_t\}_{t=1}^{T}$, where $J \subset \{1, 2, \ldots T\}$ and $N < T$. Notice that for completeness $I_n :\neq I_t$. Selected few-shot samples are then used to train the model for generalisation to the remaining frames during inference time. The proposed model makes this solution feasible thanks to its scene-aware capacity to generalise to variable content. This approach makes the model more robust for processing of sequences with changes (e.g. with high motion) as it temporally downsamples the input.

The selection of N frames for few-shot training is based on a twofold process: scene segmentation and selection of representative frames per scene. Scene segmentation is performed in an unsupervised way via clustering of deep features $\{f_t\}_{t=1}^{T}$ with K-Means algorithm [115]. Dimensionality reduction is performed by Principal Components Analysis (PCA) in order to reduce complexity and shorten the clustering time. The number of scenes (e.g. number of clusters) is unknown and variable for each input video. Hence a suitable number of clusters is computed by running KMeans K times and selecting the elbow of the averaged distortion curve, where the distortion of each sample is computed relative the centroid of its cluster. This method allows a fast and effective scene segmentation approach.

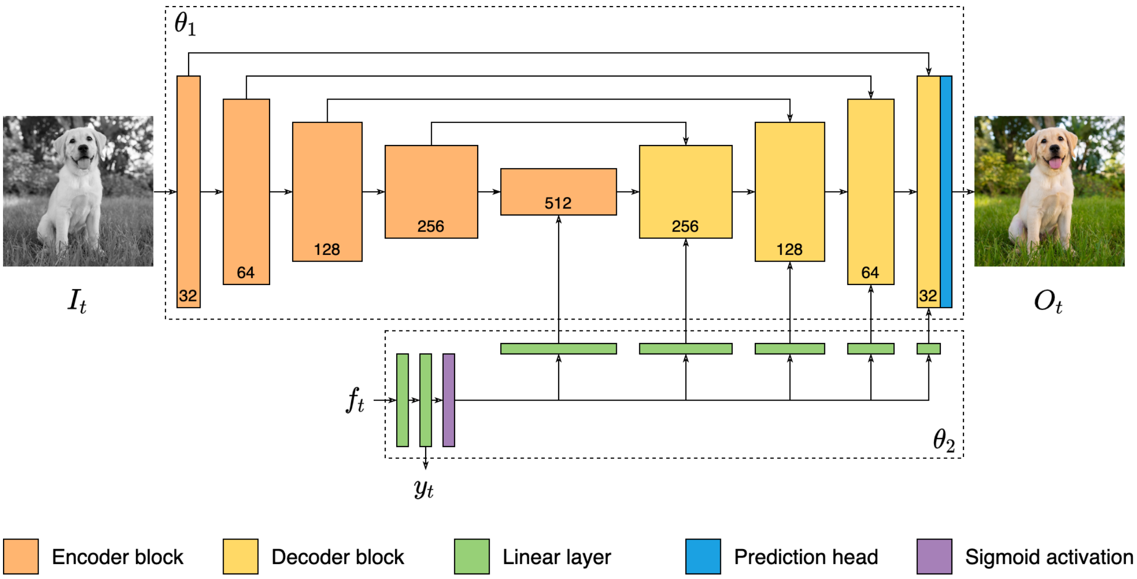


Figure 5.2: Proposed architecture for stabilisation of frame-to-frame colourised videos. The model not only learns to stabilise an input sequence, but also to cluster the input frames into different scenes, by generating a class distribution vector y_t.

Unsupervised clustering of input frames allows the generation of pseudo-labels for training the proposed classification sub-model. Notice that clustering errors will be mitigated thanks to the few-shot training, since the trained classifier will generalise to unseen frames (and potential uncertainties between scenes) during inference time. After segmentation of the input video into the scenes, suitable frames are selected from each scene by sub-clustering frames in that scene to cover a balanced span of different content. KMeans is applied again with a fixed number of clusters and a number of frames is randomly sampled from each sub-cluster. The number of selected frames per cluster and sub-clusters is proportional to the total number of frames in the given sub-cluster.

5.4.3 Network architecture

As shown in Figure 5.2, the architecture of the model proposed at Section 5.4.1 is composed of two sub-networks (denoted by $\hat{G}_1(\theta_1)$, $\hat{G}_2(\theta_2)$). Its inputs are a frame $I_t \in \mathbb{R}^{1\times H\times W}$, where $H \times W$ are the input dimensions, and its feature vector $f_t \in \mathbb{R}^{1\times d}$ (from VGG-16 classification head), where d are the number of its dimensions. The proposed architecture outputs two colour stabilised versions (main and minor frames) $O_t \in \mathbb{R}^{6\times H\times W}$ of the input frame, and a class distribution vector $y_t \in \mathbb{R}^{1\times S}$, which is the product of clustering the input to a particular scene.

I_t is processed by 4 encoder blocks which downsample the input by a factor of 2, generating $I_t^b \in \mathbb{R}^{e_b\times H_b\times W_b}$, where $b = 1, \dots 4$ is the block index, e_b is the number of dimensions and $\{H_b, W_b\} = max\left(2^5, \frac{\{H,W\}}{2^b}\right)$. The bottleneck block converts I_t^4 into

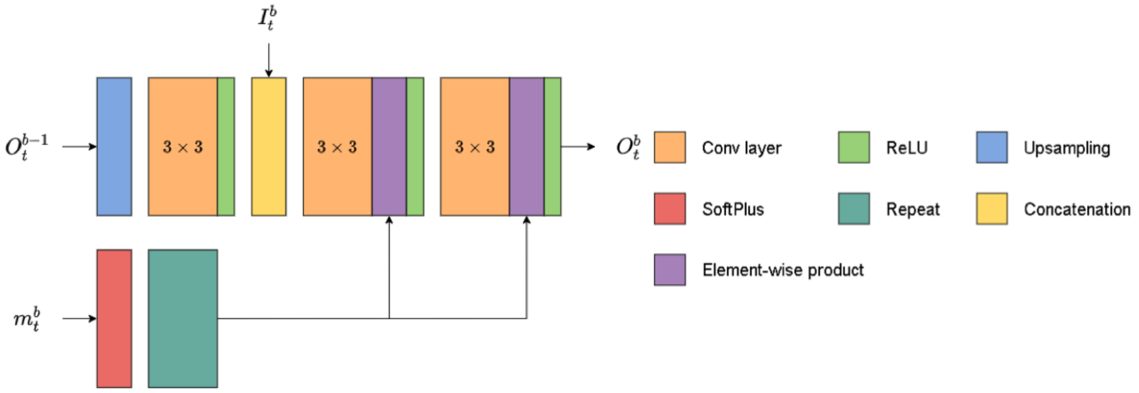


Figure 5.3: The proposed decoder block conditioned by the scene-wise embedding m_t^b. Notice that a similar architecture applies to the bottleneck, injecting the embedding vector into the pre-activations.

$O_t^5 \in \mathbb{R}^{o_5 \times H_5 \times W_5}$, where o_5 are the number of output dimensions. In parallel, f_t is processed by 2 linear layers to generate deep embeddings $f_t^1 \in \mathbb{R}^{1 \times d}$, $f_t^2 \in \mathbb{R}^{1 \times S}$. f_t^2 is both activated with a softmax operation to generate the class distribution vector y_t and with a sigmoid operation to generate the scene-aware mask a_t that will be injected into the bottleneck and decoder blocks. a_t is processed by a sequence of linear layers which generate 5 scene-aware embeddings $m_t^b \in \mathbb{R}^{1 \times o_b}$, where $b = 1, \dots 5$ and o_b are the dimensions of the bottleneck and decoder outputs. Finally, as shown in Figure 5.3, m_t^b are injected into the corresponding blocks as follows: (1) m_t^b is activated with a SoftPlus operation (smooth approximation of ReLU) and spatially repeated to generate a volume $M_t^b \in \mathbb{R}^{e_b \times H_b \times W_b}$, (2) M_t^b is element-wise multiplied to each pre-activation within the corresponding block. 4 decoder blocks with skip connections are then applied to upsample the inputs by factor of 2, generating $O_t^b \in \mathbb{R}^{o_b \times H_b \times W_b}$, where $b = 1, \dots 4$. Finally, a decoder head is applied to map O_t^1 into the output frames O_t.

5.5 Experiments

This section provides an overview of the experimental setup, as well as the details related to the training data and strategy and evaluation metrics and results. The comparison is carried out relative to the DVP baseline.

5.5.1 Training strategy

As shown in DVP, the network needs to be initialised with the main mode in order to guide the main outputs towards a specific mode. DVP selects the first image as reference for the main mode and pre-trains the network for a given number of iterations. However, when the reference image contains outliers, and those are

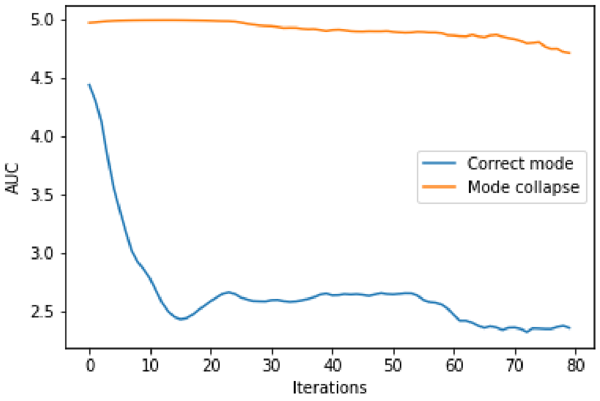


Figure 5.4: Example of mode collapse during initial pre-training. The AUC invariance during the initial iterations indicates convergence to a random state, which affects the performance of the posterior IRT training.

treated as main mode, the performance of this approach is not satisfactory. To address that, this work proposes the use of colour histograms to detect outliers when specific bins present high variance across the sequence. In particular, colourised frames P_t are converted into CIE La*b* colour space [44], and 2D colour histograms $\mathcal{H}_t \in \mathbb{R}^{Q \times Q}$ are obtained by matrix multiplication of individual histograms for a* and b* channels, where Q is the number of bins. Next, a mask $\bar{M} \in \mathbb{R}^{Q \times Q}$ is computed to locate those bins present in all the frames. Hence, bins out of the mask will represent an outlier. $\bar{M} = \prod_{t=1}^{T} M_t$, where M_t masks the bins different than zero. Finally, main mode reference frame P_{t^*} is obtained, where $t^* = \arg\min_t \sum \mathcal{H}_t \odot (M_t - \bar{M})$.

On the other hand, as shown in Figure 5.4, few-shot training might lead the network into mode collapse, rapidly converging into a random state. Mode collapse is detected when the Area Under the Curve (AUC) of the generated colour histograms vary below a threshold during a given number of iterations. In this case, the initial pre-training is repeated with random initialisation of the network weights. Due to the significant difference of complexity, classifier and stabiliser (U-Net) sub-networks are optimised using different learning rates. Overall, Adam optimiser is adopted, using a learning rate of 10^{-4} for θ_1 and 10^{-6} for θ_2. All the experiments are performed with a single GPU and using a batch size of 8 samples. Initial pre-training iterations are set to 350, and 150 frames are used for few-shot training.

Following DVP [103], this work uses the test set collected by [100], composed by 20 videos of around 200 frames from Videvo dataset[1], and extended with 8 longer videos from Videvo and Hollywood2 dataset [116], to evaluate the performance for more complex content.

[1] https://www.videvo.net/

5.5.2 Evaluation metrics

Temporal inconsistency. DVP uses wrapping error to measure temporal inconsistency by means of optical flow. However, the quality of optical flow computation and the corresponding occlusion mask might decrease when dealing with flickering content. To mitigate this issue and to better capture colour artifacts, histogram inconsistency is adopted to measure the temporal similarity in the colour domain. Being $\mathcal{H}_t$ and $\mathcal{H}_{t-1}$ the colour histograms of frames t and $t-1$, respectively, temporal histogram inconsistency E_{hist} is defined as a symmetric χ^2 distance as follows:

$$E_{hist} = 2\sum_{q=1}^{Q^2} \frac{(\mathcal{H}_{t,q} - \mathcal{H}_{t-1,q})^2}{(\mathcal{H}_{t,q} + \mathcal{H}_{t-1,q}) + \epsilon}, \tag{5.5}$$

where ϵ prevents infinity overflows and Q is the number of bins.

Performance degradation. Temporal stabilisation has to be achieved without degrading the original colourisation. Since stabilised ground truth is not available, this work uses data fidelity F_{data} between $\{O_t\}_{t=1}^{T}$ and $\{P_t\}_{t=1}^{T}$ as follows:

$$F_{data} = \frac{1}{T}\sum_{t=1}^{T} PSNR\left(P_t, O_t\right). \tag{5.6}$$

Notice that data fidelity can decrease when frames contain large amount of outliers. Therefore, perceptual quality is also evaluated using Fréchet Inception Distance (FID) [117] with the ground truth.

5.5.3 Results

Table 5.1 shows quantitative comparison results between DVP method [103], our method and the proposed ablations in Section 5.5.4. Two image-based fully-automatic colourisation methods are considered: colourful image colourisation (CIC) [42] and ChromaGAN (CGAN) [77]. The reference-based image colourisation method XC-NET [54] is also considered. Such methods, which colourise frames based on a reference image, introduce even larger flickering issues than fully auto-colourisation based networks. References are sampled from the Imagenet dataset [13] using the correspondence recommendation pipeline proposed in [5, 54]. Finally, the quality of the original predictions $\{P_t\}_{t=1}^{T}$ obtained using CIC method is studied to evaluate the effect of the proposed stabilisation. Moreover, Figure 5.6 shows the processing time of both DVP and our method in relation to the number of frames.

As can be seen from E_{hist} results, both DVP and our method significantly increase the temporal consistency compared to the original predictions, and although

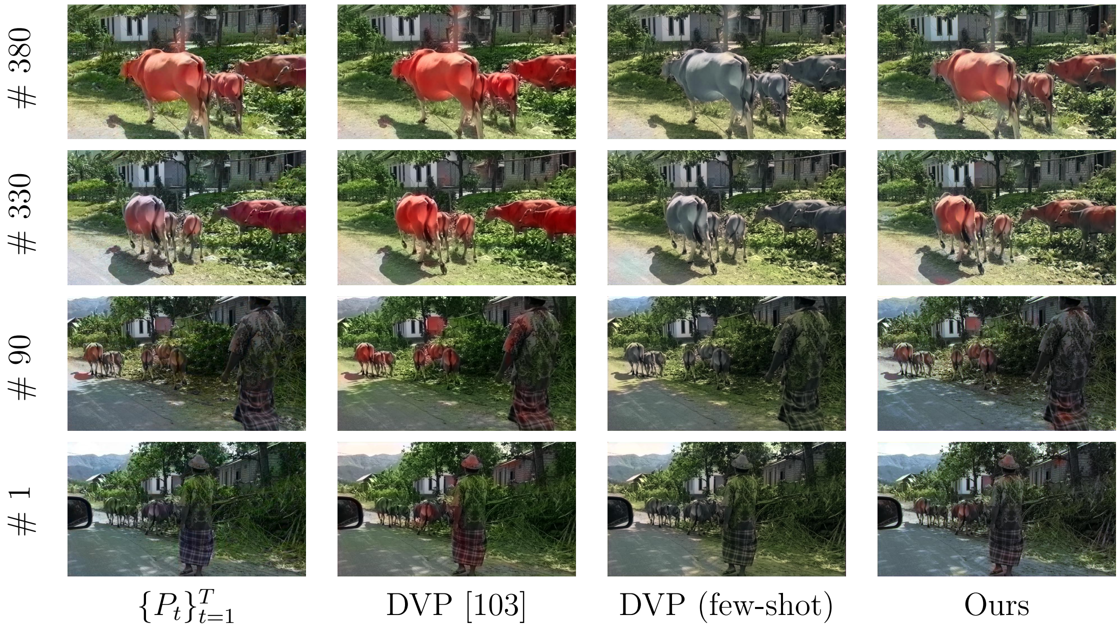


Figure 5.5: Evaluation in comparison with DVP method and processed frames at different timestamps.

DVP obtains slightly better results, our method significantly reduces the processing time for long scenes. The drop in performance when using XCNET is due to the colourfulness of the corresponding predictions and the higher concentration of flickering artefacts, compared to CIC or CGAN.

As shown in Figure 5.5, the frames at different times in the same shot suffer from inconsistent colourisation (notice the same object across various frames with different colour). DVP and DVP with few-shot training temporal both provide more consistent results, but still the main mode is either not correctly chosen or the colours are plain, resulting in less natural appearance. This is reflected in data fidelity results, where our method achieves the best performance. FID also confirm this fact, as DVP lowers the perceptual quality of the original predictions due to its strong stabilisation and degradation of input colours. Finally, as shown in Figure 5.6, the few-shot strategy allowed a fix amount of training iterations, resulting into a flat time response independent to the length of the input sequence. Note that the total time may increase proportionally to the number of scenes, due to the individual initial pre-training per scene.

5.5.4 Ablations

An ablation study is performed to analyse the importance of the proposed scene-aware architecture. First, DVP is tested with the proposed few-shot training strategy. As shown in Table 5.1 and Figure 5.5 (DVP few-shot), without using a clas-

Table 5.1: E_{hist}, F_{data} and FID comparison for different colourisation methods.

Method	$E_{hist} \downarrow$			
	$\{P_t\}_{t=1}^{T}$	CIC [42]	CGAN [77]	XCNET [54]
DVP [103]	20.96	2.30	**1.58**	3.30
Ours		3.08	2.54	3.59
DVP (few-shot)		3.75	3.79	3.10
Ours (first frame)		**1.39**	2.14	**2.69**

Method	$F_{data} \uparrow$ [dB]		
	CIC [42]	CGAN [77]	XCNET [54]
DVP [103]	19.12	19.32	18.94
Ours	**28.63**	**30.31**	**26.56**
DVP (few-shot)	18.14	18.47	18.67
Ours (first frame)	28.46	30.21	26.40

Method	FID ↓			
	$\{P_t\}_{t=1}^{T}$	CIC [42]	CGAN [77]	XCNET [54]
DVP [103]	122.74	126.38	111.16	100.21
Ours		121.16	105.65	**97.96**
DVP (few-shot)		129.68	114.98	102.22
Ours (first frame)		**119.76**	**104.03**	99.92

sification sub-network, DVP is unable to generalise to complex sequences and the input colours are significantly degraded. This drop in performance proves the importance of the classification sub-network to perform effective few-shot training. Finally, a second ablation is performed to evaluate the proposed initialisation mechanism in Section 5.5.1, which proposes the best reference for main mode per scene by means of histogram characteristics. As shown in Table 5.1 (ours first frame), a drop in performance is observed when using the first frame as main mode reference (as DVP proposes), proving the effectiveness of the proposed methodology. Notice that original DVP performance could be improved by using the same initialisation mechanism.

5.6 A pipeline for video colourisation

The ultimate goal of the content production part of this thesis is to propose a realistic and deployable solution for content enhancement and restoration, focused on image and video colourisation. Previous chapters proposed methods for image colourisation which improved state-of-the-art methods in performance, model complexity, runtime, etc. In particular, Chapter 3 introduced ColorGAN, a fully-automatic image colourisation method based on Conditional GANs, which improved the colourfulness and perceptual realism of previous GAN-based fully-automatic methods. Moreover,

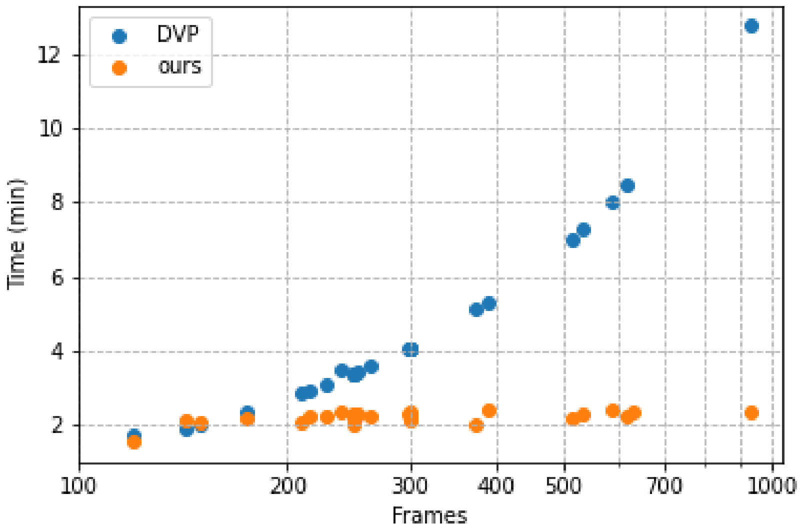


Figure 5.6: Comparison of processing time for all test sequences. Notice the significant increase of DVP when the number of frames increase.

in order to reduce ambiguity and generate more accurate predictions, Chapter 4 explored exemplar-based methods which involve the interaction of the producer. XCNET is then introduced, a fast end-to-end architecture that improves existing reference-based methods while decreasing the complexity and runtime. Both image-based methods can be integrated into a video colourisation pipeline by using the proposed temporal stabilisation framework from this chapter.

As shown in Figure 5.7, the proposed pipeline processes a grayscale input sequence $\{I_t\}_{t=1}^{T}$ of T frames and generates a colourised output $\{O_t\}_{t=1}^{T}$. Although the proposed scheme integrates XCNET to perform reference-based colourisation, a few modifications are needed to integrate ColorGAN and thus obtain a fully-automatic version. Reference-based colourisation is performed on a scene-by-scene basis using specific references. Therefore, the scene segmentation approach from Section 5.4.2 is used to cluster feature vectors relative to each frame $\{f_t\}_{t=1}^{T}$, into S scenes. Feature vectors are generated by means of the *Feature Extractor* module (e.g. VGG-16 classification head). Then, every frame is classified by means of the *Scene Segmentation* module, which generates scene indexes $\{s_t\}_{t=1}^{T}$ that are used by the *Scene Sampler* to yield the Q frames belonging to each scene $\{I_q\}_{q=1}^{Q}$. Reference-based colourisation is performed frame-to-frame for each scene by means of the XCNET network and the reference is automatically retrieved with the *Reference Retrieval* module. This work uses the recommendation pipeline [5] from Chapter 4 to retrieve similar colour references from Imagenet dataset [50]. Alternatively, references can be provided manually by the producer. Finally, outputs for each scene $\{P_q\}_{q=1}^{Q}$ are merged by means of the *Scene Composer* module to generate a colourised output

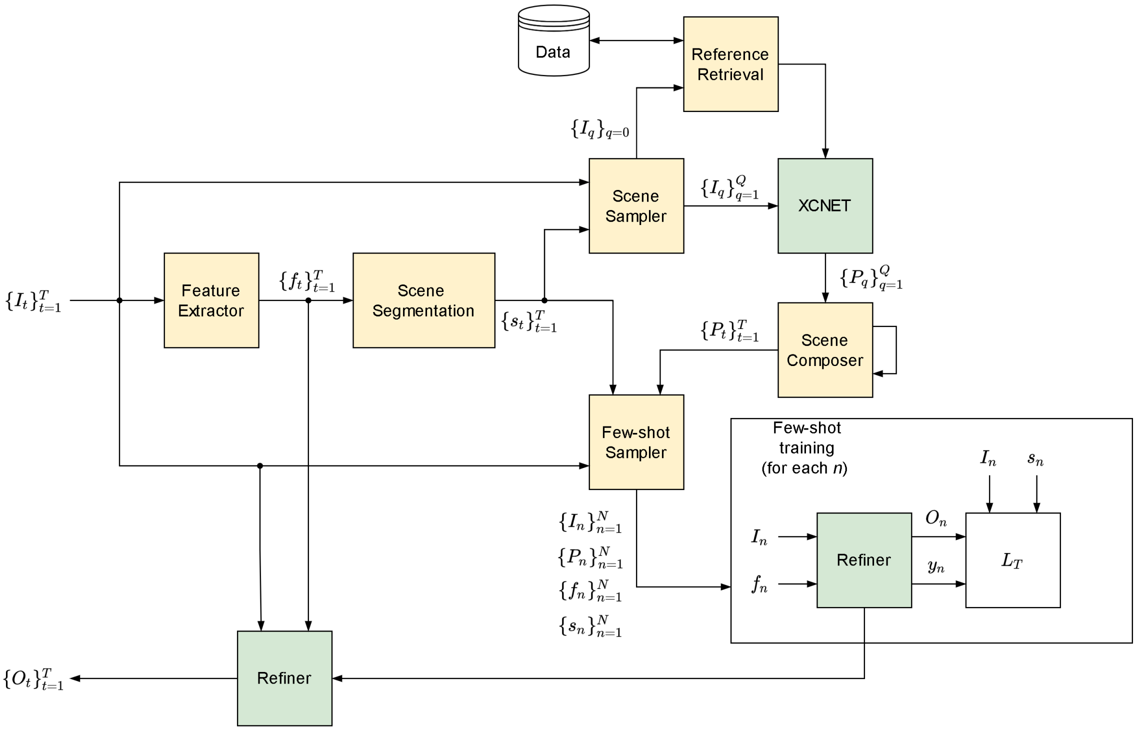


Figure 5.7: Proposed pipeline for exemplar-based video colourisation combining the image-based XCNET architecture (from Chapter 4) with the temporal stabilisation framework proposed in this chapter.

$\{P_t\}_{t=1}^{T}$.

As discussed in this chapter, frame-to-frame colourisation usually leads to temporal inconsistencies such as flickering artifacts, hence the last step is to apply the proposed temporal stabilisation framework to gain temporal consistency. A refiner network is trained using the few-shot strategy in Section 5.4.2, and a stable colourisation output $\{O_t\}_{t=1}^{T}$ is posteriorly generated.

5.7 Conclusions

This chapter proposed a general framework for temporal stabilisation of frame-to-frame colourised videos using scene-aware deep video priors. The framework includes an optimised few-shot training strategy to reduce the processing time of DVP baseline by removing its time response conditioned on the number of input frames. In order to handle complex sequences with multiple scenes, the DVP architecture is modified by adding a classification sub-network which clusters the input frames with the objective of learning scene-specific priors. Experimental results show that our method improves data fidelity and perceptual quality and achieves similar temporal consistency to DVP while reducing the processing time in long sequences. Finally, a pipeline for reference-based video colourisation is proposed, integrating the proposed stabilisation framework with image-based approaches from previous chapters.

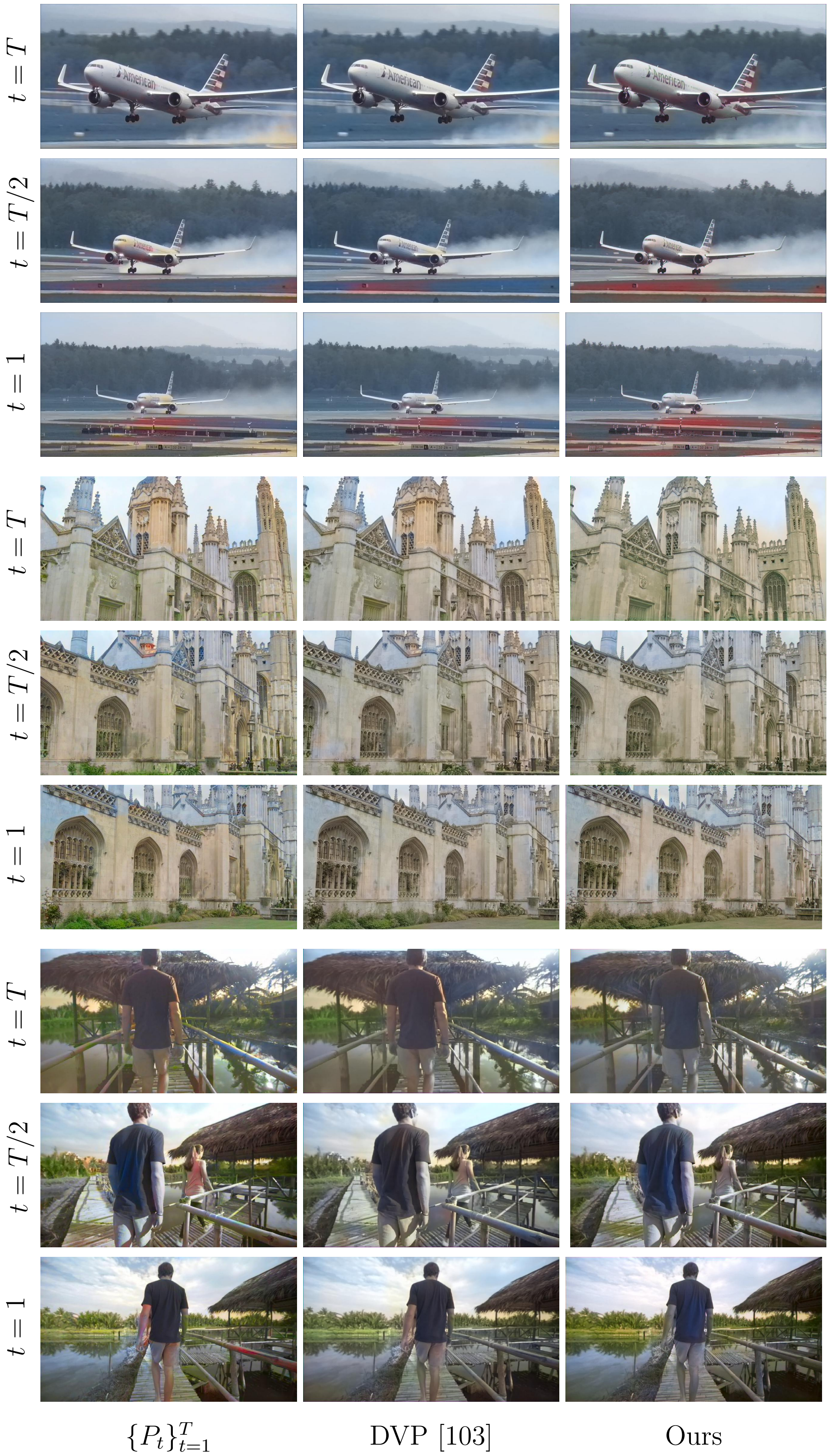


Figure 5.8: Evaluation in comparison with DVP method and processed frames at different timestamps.

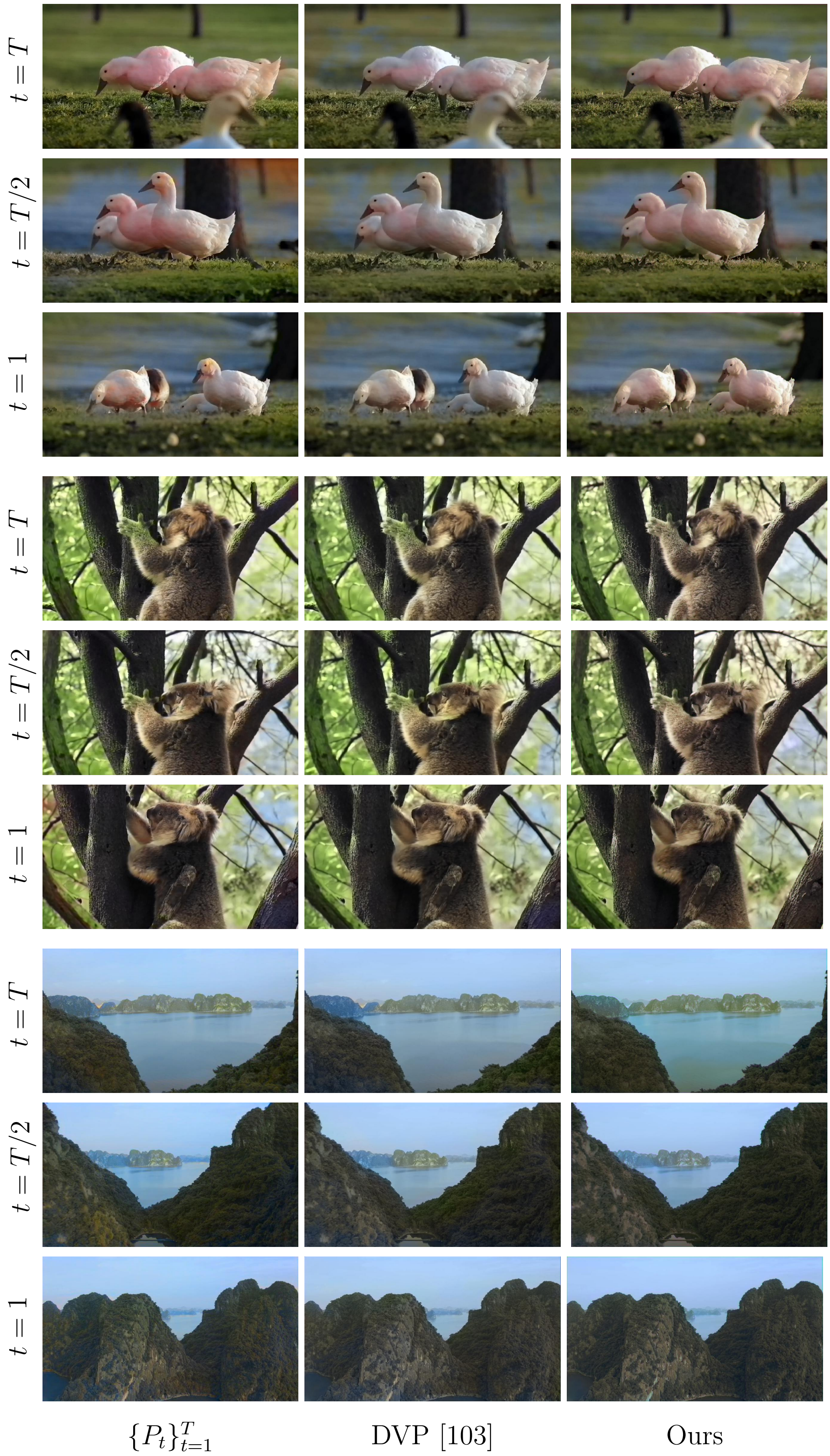


Figure 5.9: Evaluation in comparison with DVP method and processed frames at different timestamps.

As future work, model efficiency can be further improved by simplifying the network architecture or by using techniques such as pruning or weights quantisation. Moreover, finer tuning of colourisation could be achieved by improving the scene segmentation process in order to obtain more precise scene priors. Finally, the proposed video colourisation pipeline still requires a two-step process (frame-to-frame colourisation followed by temporal stabilisation), this can be simplified by an unified framework, integrating the DVP methodology into an end-to-end video colourisation pipeline.

Part II

Content distribution: video compression

Chapter 6

Neural networks for chroma intra prediction

6.1 Chapter overview

This chapter provides a detailed description of the contribution of this research to a recent approach to chroma intra-prediction using attention-based CNNs, as well as a thorough description of the configuration setup used in the experiments. The experiments and results described in this chapter have published at the IEEE International Conference on Image Processing (ICIP), 2020 [6]. Section 6.3 reviews related work in the literature, identifying the main drawbacks and possible improvements, Section 6.4 details the proposed methodology, Section 6.5 provides information about the implementation and data used in the experiments and a quantitative evaluation of the results while Section 6.6 identifies future work and Section 6.7 provides conclusions.

6.2 Introduction

Efficient video compression has become an essential component of multimedia streaming. The convergence of digital entertainment followed by the growth of web services such as video conferencing, cloud gaming and real-time high-quality video streaming, prompted the development of advanced video coding technologies capable of tackling the increasing demand for higher quality video content and its consumption on multiple devices. New compression techniques enable a compact representation of video data by identifying and removing spatial-temporal and statistical redundancies within the signal. This results in smaller bitstreams, enabling more efficient storage and transmission as well as distribution of content at higher quality, requiring reduced resources.

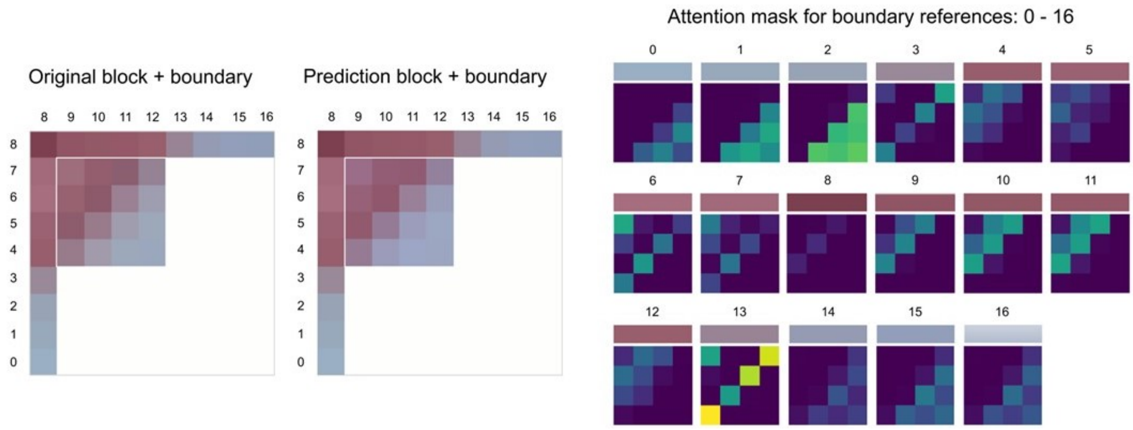


Figure 6.1: Visualisation of the attentive prediction process. The first image shows the original and predicted blocks and their respective reference samples, and the second image the corresponding attention masks for each boundary location. For each reference sample 0-16 the attention module generates its contribution to the prediction of individual pixels from a target 4 × 4 block.

Among the fundamental blocks of typical video coding schemes, intra prediction exploits spatial redundancies within a frame by predicting samples of the current block from already reconstructed samples in its close surroundings. The latest draft of the Versatile Video Coding (VVC) standard [118] (referred to as VVC in the rest of this chapter) allows a large number of possible intra modes to be used on the luma component, including up to 67 directional modes and other advanced methods, at the cost of a considerable amount of signalling data. Conversely, to limit the impact of mode signalling on compression performance, a reduced number of modes is employed to intra-predict chroma samples, including the Planar, DC, pure horizontal and pure vertical modes, and the Derived Mode (DM, corresponding to using the same mode used to predict the collocated luma block). In addition to traditional modes, more recent research introduced schemes which further exploit cross-component correlations between the luma and chroma components. Such correlations motivated the development of the Cross-Component Linear Model (CCLM, or simply LM in this chapter) intra modes. When using CCLM, the chroma components are predicted from already reconstructed luma samples using a linear model. Nonetheless, the limitation of simple linear predictions comes from its high dependency on the selection of predefined reference samples. Improved performance can be achieved using more sophisticated Machine Learning (ML) mechanisms [119, 120], which are able to derive more complex representations of the reference data and hence boost the prediction capabilities. Unlike previous methods where neighbouring references are used regardless of their location, this work proposes a new ML-based cross-component intra-prediction method which is capable of learning the spatial relations between reference and predicted samples.

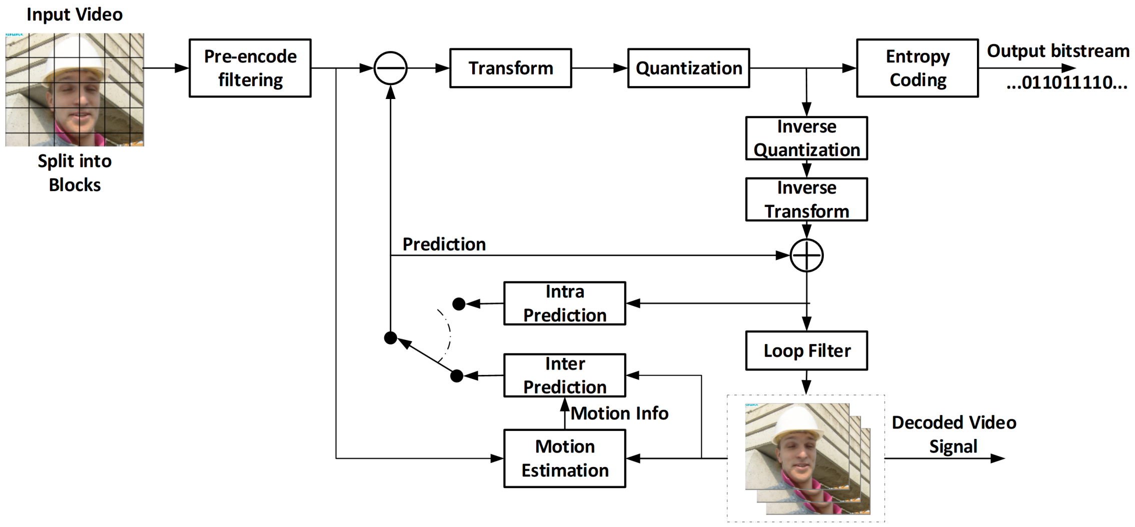


Figure 6.2: General block diagram of VVC encoder.

A novel ML-based cross-component intra-prediction method is proposed, introducing a new attention module capable of tracking the contribution of each neighbouring reference sample when computing the prediction of each chroma pixel, as shown in Figure 6.1. As a result, the proposed scheme better captures the relationship between the luma and chroma components, resulting in more accurate prediction samples. However, such NN-based methods significantly increase the codec complexity, increasing the encoder and decoder times by up to 120% and 947%, respectively.

6.3 Related Work

Colour images are typically represented by three colour components (e.g. RGB, YCbCr). The YCbCr colour scheme is often adopted by digital image and video coding standards (such as JPEG, MPEG-1/2/4 and H.261/3/4) due to its ability to compact the signal energy and to reduce the total required bandwidth. Moreover, chrominance components are often subsampled by a factor of two to conform to the YCbCr 4:2:0 chroma format, in which the luminance signal contains most of the spatial information. Nevertheless, cross-component redundancies can be further exploited by reusing information from already coded components to compress another component. In the case of YCbCr, the Cross-Component Linear model (CCLM) [121] uses a linear model to predict the chroma signal from a subsampled version of the already reconstructed luma block signal. The model parameters are derived at both the encoder and decoder sides without needing explicit signalling in the bitstream. Another example is the Cross-Component Prediction (CCP) [122] which resides at the transform unit (TU) level regardless of the input colour space. In case of YCbCr, a subsampled and dequantised luma transform block (TB) is used to modify the chroma TB at the same spatial location based on a context parameter signalled in the bitstream. An extension of this concept modifies one chroma component using the residual signal of the other one [123]. Such methodologies significantly improved the coding efficiency by further exploiting the cross-component correlations within the chroma components.

In parallel, recent success of deep learning application in computer vision and image processing influenced design of novel video compression algorithms. In particular in the context of intra-prediction, a new algorithm [120] was introduced based on fully-connected layers and CNNs to map the prediction of block positions from the already reconstructed neighbouring samples, achieving BD-rate (Bjontegaard Delta rate) [124] savings of up to 3.0% on average over HEVC, for approx. 200% increase in decoding time. The successful integration of CNN-based methods for luma intra-prediction into existing codec architectures has motivated research into alternative methods for chroma prediction, exploiting cross-component redundancies similar to the aforementioned LM methods. A novel hybrid neural network for chroma intra prediction was recently introduced in [119]. A first CNN was designed to extract features from reconstructed luma samples. This was combined with another fully-connected network used to extract cross-component correlations between neighbouring luma and chroma samples. The resulting architecture uses complex non-linear mapping for end-to-end prediction of chroma channels. However, this is achieved at the cost of disregarding the spatial location of the boundary reference samples and significant increase of the complexity of the prediction process. To this end, an improved cross-component intra-prediction model based on neural networks is proposed, as illustrated in this chapter.

6.4 Proposed method

Similarly to the model in [119], the proposed method adopts a scheme based on three network branches that are combined to produce prediction samples. The first two branches work concurrently to extract features from the available reconstructed samples, including the already reconstructed luma block as well as the neighbouring luma and chroma reference samples. The first branch (referred to as cross-component boundary branch) aims at extracting cross-component information from neighbouring reconstructed samples, using an extended reference array on the left of, and above the current block, as illustrated in Fig. 6.1. The second branch (referred to as luma convolutional branch) extracts spatial patterns over the collocated reconstructed luma block applying convolutional operations. The features from the two branches are fused together by means of an attention model, as detailed in the

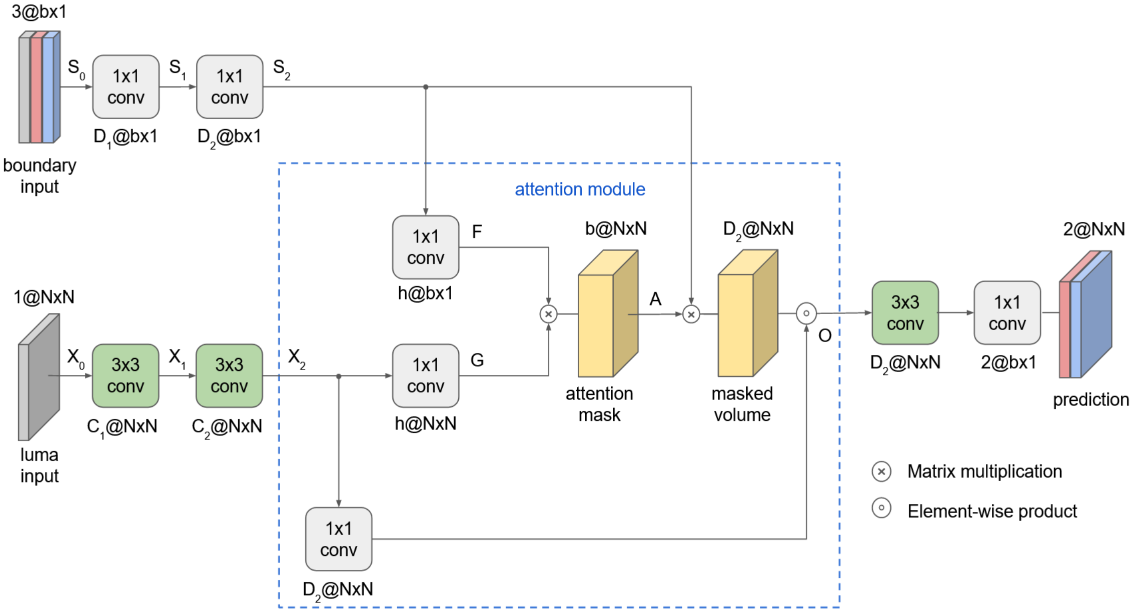


Figure 6.3: Proposed architecture including the attention module used to fuse the output of the two first network branches.

rest of this section. The output of the attention model is finally fed into the third network branch, to produce the resulting Cb and Cr predictions.

An illustration of the proposed network architecture is presented in Fig. 6.3. Without loss of generality, only square blocks of pixels are considered in this work. After intra-prediction and reconstruction of a luma block in the video compression chain, luma samples can be used for prediction of co-located chroma components. In this discussion, the size of a luma block is assumed to be (downsampled to) $N \times N$ samples, which is the size of the co-located chroma block. This may require the usage of conventional downsampling operations, such as in the case of using chroma sub-sampled picture formats such as 4:2:0. Note that a video coding standard treats all image samples as unsigned integer values within a certain precision range based on the internal bit depth. However, in order to utilise common deep learning frameworks, all samples are converted to floating point and normalised to values within the range $[0, 1]$. For the chroma prediction process, the reference samples used include the co-located luma block $X_0 \in \mathbb{R}^{N \times N}$, and the array of reference samples $B_c \in \mathbb{R}^b$, $b = 4N+1$ from the left and from above the current block (Figure 6.1), where $c = Y$, C_b or C_r refers to the three colour components. B is constructed from samples on the left boundary (starting from the bottom-most sample), then the corner is added, and finally the samples on top are added (starting from the left-most sample). In case some reference samples are not available, these are padded using a predefined value, following the standard approach defined in VVC. Finally, $S_0 \in \mathbb{R}^{3 \times b}$ is the cross-component volume obtained by concatenating the three reference arrays B_Y, B_{Cb} and B_{Cr}.

6.4.1 Cross-component boundary branch

The first branch (referred to as the cross-component boundary branch) extracts cross component features from $S_0 \in \mathrm{I\!R}^{3\times b}$ by applying I consecutive D_i - dimensional 1×1 convolutional layers to obtain the $S_i \in \mathrm{I\!R}^{D_i \times b}$ output feature maps, where $i = 1, 2 \dots I$. By applying 1×1 convolutions, the boundary input dimensions are preserved, resulting in an D_i-dimensional vector of cross-component information for each boundary location. The resulting volumes are activated using a Rectified Linear Unit (ReLU) non-linear function. S_i can be expressed in a neural network form as:

$$S_i\left(S_{i-1}, W_i\right) = \mathcal{F}\left(W_i S_{i-1}^T + b_i\right), \tag{6.1}$$

where $W_i \in \mathrm{I\!R}^{D_i \times D_{i-1}}$ and b_i are the i-layer weights and bias respectively, $D_0 = 3$, and $\mathcal{F}$ is a ReLU activation function.

6.4.2 Luma convolutional branch

In parallel with the extraction of the cross component features, the second branch (referred to as the luma convolutional branch) extracts spatial patterns over the co-located reconstructed luma block X_0 by applying convolutional operations. The luma convolutional branch is defined by J consecutive C_j-dimensional 3×3 convolutional layers with a stride of 1, to obtain $X_j \in \mathrm{I\!R}^{C_j \times N^2}$ feature maps from the N^2 input samples, where $j = 1, 2 \dots J$. Similar to the cross-component boundary branch, in this branch a bias and a ReLU activation are applied within convolutional layer. X_j is expressed as:

$$X_j\left(X_{j-1}, W_j\right) = \mathcal{F}\left(W_j * X_{j-1} + b_j\right), \tag{6.2}$$

where $W_j \in \mathrm{I\!R}^{D_j \times D_{j-1}}$ and b_j are the j-layer weights and bias, respectively, and X_0 is the input luma block.

6.4.3 Attention-based fusion module

The concept of ”attention-based” learning is a well-known idea used in deep learning frameworks, to improve the performance of trained networks in complex prediction tasks. The idea behind attention models is to reduce complex tasks by predicting smaller ”areas of attention” that are processed sequentially in order to encourage more efficient learning. In particular, self-attention (or intra-attention) is used to assess the impact of particular input variables on the outputs, whereby the prediction is computed focusing on the most relevant elements of the same sequence [34, 86]. Extending this concept to chroma intra-prediction, this work combines the features from the two aforementioned network branches in order to assess the impact of each input variable with respect to their spatial locations. This addresses previous limitations of similar cross-component prediction techniques, which generally discard the spatial relation of the neighbouring reference and the predicted samples. The feature maps (S_I and X_J) from both branches are each convolved using a 1×1 kernel, to project them into two corresponding reduced feature spaces. Specifically, S_I is convolved with a filter $W_F \in \mathbb{R}^{h \times D}$ to obtain the h-dimensional feature matrix F. Similarly, X_J is convolved with a filter $W_G \in \mathbb{R}^{h \times C}$ to obtain the h-dimensional feature matrix G. The two matrices are multiplied together to obtain the pre-attention map $M = G^T F$. Finally, the attention matrix $A \in \mathbb{R}^{N^2 \times b}$ is obtained applying a softmax operation to each element of M, to generate the probability of each boundary location being able to predict a sample location in the block. Each value $\alpha_{j,i}$ in A is obtained as:

$$\alpha_{j,i} = \frac{\exp\left(m_{i,j}/T\right)}{\sum_{n=0}^{b-1} \exp\left(m_{n,j}/T\right)}, \tag{6.3}$$

where $j = 0, ..., N^2 - 1$ represents the sample location in the predicted block, $i = 0, ..., b - 1$ represents a reference sample location, and T is the softmax temperature parameter controlling the smoothness of the generated probabilities, with $0 < T \leq 1$. Notice that the smaller the value of T, the more localised are the obtained attention areas resulting in correspondingly fewer boundary samples contributing to a given prediction location.

The weighted sum of the contribution of each reference sample in predicting a given sample at a specific location is obtained by computing the matrix multiplication between the cross-component boundary features S_I and the attention matrix A, or formally $S_I^T A$. In order to further refine $S_I^T A$, this weighted sum can be multiplied by the output of the luma branch. To do so, the output of the luma branch must be transformed to change its dimensions by means of a 1×1 convolution using a matrix $W_{\bar{x}} \in \mathbb{R}^{D \times C}$ to obtain a transformed representation $\bar{X}$, then $O = \bar{X} \odot (S_I^T A)$, where $\odot$ is the element-wise product.

6.4.4 Prediction head branch

The output of the attention model is fed into the third network branch, to compute the predicted chroma samples. In this branch, a final CNN is used to map the fused features from the first two branches are combined by means of the attention model into the final chroma prediction. The prediction head branch is defined by two convolutional layers, applying E-dimensional 3×3 convolutional filters and then 2-dimensional 1×1 filters for deriving the two chroma components at once.

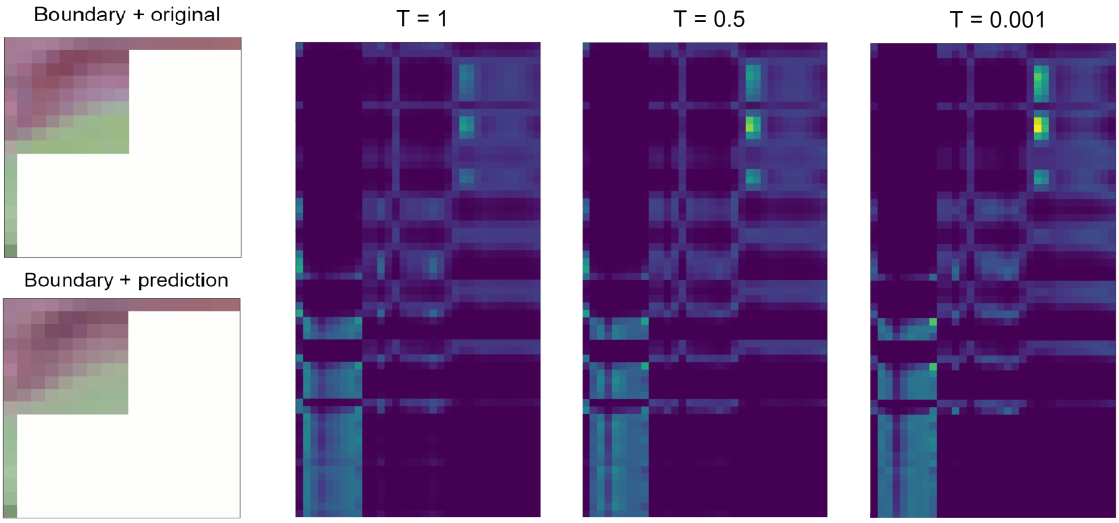


Figure 6.4: Attention visualisation when predicting a block for different temperature T values. Notice the increasing attention sparsity when T decreases. Axes y represents the N^2 block locations and axes y the B positions.

6.5 Experiments

6.5.1 Training settings

Training examples were extracted from the DIV2K dataset [125], which contains high-definition high-resolution content of large diversity. This database contains 800 training samples and 100 samples for validation, providing 6 lower resolution versions with downsampling by factors of 2, 3 and 4 with a bilinear and unknown filters. For each data instance, one resolution was randomly selected and then M blocks of each $N \times N$ sizes ($N = 4, 8, 16$) were chosen, making balanced sets between block sizes and uniformed spatial selections within each image. Moreover, 4:2:0 chroma subsampling is assumed, where the same downsampling filters implemented in VVC are used to downsample co-located luma blocks to the size of the corresponding chroma block. All the schemes were trained from scratch using the Adam optimiser [96] with a learning rate of 10^{-4}.

6.5.2 Integration into VVC

The methods introduced in the paper where integrated within a VVC encoder, using the VVC Test Model (VTM) 7.0 [126]. The integration of the proposed NN-based cross-component prediction into the VVC coding scheme requires normative changes not only in the prediction process, but also in the way the chroma intra-prediction mode is signalled in the bitstream and parsed by the decoder.

A new block-level syntax flag is introduced to indicate whether a given block makes use of one of the proposed schemes. If the proposed NN-based method is used, a prediction is computed for the two chroma components. No additional information is signalled related to the chroma intra-prediction mode for the block. Conversely, if the method is not used, the encoder proceeds in signalling the chroma intra-prediction mode as in conventional VVC specifications. For instance, a subsequent flag is signalled to identify if conventional LM modes are used in the current block or not. The prediction path also needs to accommodate the new NN-based predictions. This largely reuses prediction blocks that are needed to perform conventional CCLM modes. In terms of mode selection at the encoder side, the new NN-based mode is added to the conventional list of modes to be tested in full rate-distortion sense.

Table 6.1: Model hyperparameters per block size

Branch	4 × 4	8 × 8	16 × 16
CC Boundary	16, 32	32, 64	64, 96
Luma Conv	32, 32	64, 64	96, 96
Attention	16, 16, 32	16, 16, 64	16, 16, 96
Output	32, 2	64, 2	96, 2

6.5.3 Results

The proposed methodology is tested under the Common Test Conditions (CTC) [127], using the suggested all-intra configuration for VVC with a QP of 22, 27, 32 and 37. BD-rate is adopted to evaluate the relative compression efficiency with respect to the latest VVC anchor. Besides, a joint cross-component metric (YCbCr) [124] is considered to evaluate the influence of the chroma gains when signalling the luma component. Test sequences include 26 video sequences of different resolutions known as Classes A, B, C, D and E. Due to the nature of the training set, only natural content sequences were considered, and screen content sequences (Class F in the CTC) were excluded from the tests. It is worth mentioning that in these tests, all block sizes were allowed to be used by the VVC encoder, including all rectangular shapes as well as larger blocks that are not supported by the proposed method. As such, the algorithm potential is highly limited, given that it is only applied to a limited range of blocks. Nonetheless, the algorithm is capable of providing consistent compression gains. The overall results are summarised in Table 6.2, showing average BD-rate reductions of 0.14% 0.69%, and 0.52% for Y, Cb and Cr components respectively, and an average joint YCbCr BD-rate (calculated as in [128]) reduction of 0.20%.

Moreover, in order to further evaluate performance of the scheme, a constrained test is also performed whereby the VVC partitioning process is limited to using only the supported square blocks of 4×4, 8×8 and 16×16 sizes. A corresponding anchor

Table 6.2: BD-rate results anchoring to VTM-7.0

	Y	Cb	Cr	YCbCr
Class A1	-0.18%	-0.84%	-0.58%	-0.23%
Class A2	-0.13%	-0.57%	-0.38%	-0.19%
Class B	-0.15%	-0.65%	-0.67%	-0.21%
Class C	-0.17%	-0.63%	-0.41%	-0.22%
Class D	-0.17%	-0.63%	-0.61%	-0.21%
Class E	-0.08%	-0.80%	-0.47%	-0.16%
Overall	-0.15%	-0.68%	-0.53%	-0.20%

Table 6.3: BD-rate results for constrained test

	Y	Cb	Cr	YCbCr
Class A1	-0.26%	-2.17%	-1.96%	-0.53%
Class A2	-0.22%	-2.37%	-1.64%	-0.50%
Class B	-0.23%	-2.00%	-2.17%	-0.45%
Class C	-0.26%	-1.64%	-1.41%	-0.44%
Class D	-0.25%	-1.55%	-1.67%	-0.42%
Class E	-0.03%	-1.35%	-1.77%	-0.24%
Overall	-0.22%	-1.84%	-1.78%	-0.43%

was generated for this test. Table 6.3 summarises the results for the constrained test, showing a considerable improvement over the constrained VVC anchor. Average BD-rate reductions of 0.22%, 1.84% and 1.78% are reported for the Y, Cb and Cr components respectively, as well as an average joint YCbCr reduction of 0.43%. In terms of complexity, even though several simplifications were considered during the integration process, the proposed solution significantly impacts the encoder and decoder time up to 120% and 947% on average, respectively. Future simplifications have to be adopted in order to increase computational efficiency of the scheme. Finally, the trained models were compared with the state-of-the-art hybrid architecture [119] with the aim to evaluate the influence of the proposed attention module. Table 6.4 summarises the results for prediction accuracy along DIV2K test set by means of averaged PSNR. Significant improvements in PSNR are reported on all block sizes.

6.6 Future work

As future work, a complete set of network models for all VVC block sizes should be implemented in order to ensure a full usage of the proposed approach leading to the promising results shown in the constrained experiment. Moreover, as described in Chapter 7, interpretability is required to deeply understand how the models obtains their predictions with the aim to promote simplifications towards a cost-effective

Table 6.4: Prediction performance evaluation (PSNR)

Model	4x4	8x8	16x16
Hybrid CNN [119]	28.61	31.47	33.36
Ours	**30.23**	**33.13**	**36.13**

implementation.

6.7 Conclusions

In this chapter existing approaches for chroma intra-prediction based on neural networks were improved, introducing a new attention module which is capable of learning spatial relations when extracting the correlational features from the neighbouring reference samples to the block prediction samples. The proposed architecture was integrated into the latest VVC anchor, signalled as a new chroma intra-prediction mode working in parallel with traditional modes towards predicting the chroma component samples. Experimental results show the effectiveness of the proposed method, achieving remarkable compression efficiency.

Chapter 7

Multi-model architectures and simplifications

7.1 Chapter overview

This chapter proposes a block-independent multi-model and training methodology that reduces the complexity of the proposed attention-based neural network in Chapter 6 whilst remaining competitive with state-of-the-art approaches. Proposed simplifications include a framework to reduce the complexity of the convolutional operations, a simplified cross-component processing model using sparse autoencoders and a fast and cost-effective implementation using integer precision approximations. The experiments and results described in this chapter were published in the IEEE Journal of Selected Topics in Signal Processing (Volume: 15, Issue: 2, Feb. 2021) [129]. Moreover, the interpretability rationale behind the simplified convolutions has been published in the International Broadcasting Convention, 2020 [130]. This chapter is organised as follows: Section 7.3 provides a brief overview on the related work, Section 7.4 presents the proposed simplifications and Section 7.5 shows experimental results while Section 7.6 identifies future work and Section 7.7 provides conclusions.

7.2 Introduction

Advanced video compression algorithms are often complex and computationally intense, significantly increasing the encoding and decoding time. Therefore, despite bringing high coding gains, their potential for application in practice is limited. Methods based on Convolutional Neural Networks (CNNs) [119, 6] provided significant improvements at the cost of two main drawbacks: the associated increase in system complexity and the tendency to disregard the location of individual reference samples. Related works deployed complex neural networks (NNs) by means of model-based interpretability [131]. For instance, VVC recently adopted simplified NN-based methods such as Matrix Intra Prediction (MIP) modes [132] and Low-Frequency Non Separable Transform (LFNST) [133]. For the particular task of block-based intra-prediction, the usage of complex NN models can be counterproductive if there is no control over the relative position of the reference samples. When using fully-connected layers, all input samples contribute to all output positions, and after the consecutive application of several hidden layers, the location of each input sample is lost. This behaviour clearly runs counter to the design of traditional approaches, in which predefined directional modes carefully specify which boundary locations contribute to each prediction position.

A novel ML-based cross-component intra-prediction method is proposed in Chapter 6 (also referred as [6] in the rest of this chapter), introducing a new attention module capable of tracking the contribution of each neighbouring reference sample when computing the prediction of each chroma pixel. As a result, the proposed scheme better captures the relationship between the luma and chroma components, resulting in more accurate prediction samples. However, such NN-based methods significantly increase the codec complexity, increasing the encoder and decoder times by up to 120% and 947%, respectively.

This chapter focuses on complexity reduction in video coding with the aim to derive a set of simplified and cost-effective attention-based architectures for chroma intra-prediction. Understanding and distilling knowledge from the networks enables the implementation of less complex algorithms which achieve similar performance to the original models. Moreover, a novel training methodology is proposed in order to design a block-independent multi-model which outperforms the state-of-the-art attention-based architectures and reduces inference complexity. The use of variable block sizes during training helps the model to better generalise on content variety while ensuring higher precision on predicting large chroma blocks. The main contributions of this chapter are the following:

- A competitive block-independent attention-based multi-model and training methodology;
- A framework for complexity reduction of the convolutional operations;
- A simplified cross-component processing model using sparse auto-encoders;
- A fast and cost-effective attention-based multi-model with integer precision approximations.

7.3 Related work

A novel hybrid neural network for chroma intra prediction was recently introduced in [119]. A first CNN was designed to extract features from reconstructed luma samples. This was combined with another fully-connected network used to extract cross-component correlations between neighbouring luma and chroma samples. The resulting architecture uses complex non-linear mapping for end-to-end prediction of chroma channels. However, this is achieved at the cost of disregarding the spatial location of the boundary reference samples and significant increase of the complexity of the prediction process. As shown in [6], after a consecutive application of fully-connected layers in [119], the location of each input boundary reference sample is lost. Therefore, the fully-convolutional architecture in [6] better matches the design of the directional VVC modes and is able to provide significantly better performance. The use of attention models enables effective utilisation of the individual spatial location of the reference samples [6]. The concept of "attention-based" learning is a well-known idea used in deep learning frameworks, to improve the performance of trained networks in complex prediction tasks [29, 86, 87]. In particular, self-attention is used to assess the impact of particular input variables on the outputs, whereby the prediction is computed focusing on the most relevant elements of the same sequence [34]. The novel attention-based architecture introduced in [6] reports average BD-rate reductions of -0.22%, -1.84% and -1.78% for the Y, Cb and Cr components, respectively, although it significantly impacts the encoder and decoder time.

One common aspect across all related work is that whilst the result is an improvement in compression this comes at the expense of increased complexity of the encoder and decoder. In order to address the complexity challenge, this chapter aims to design a set of simplified attention-based architectures for performing chroma intra-prediction faster and more efficiently. Recent works addressed complexity reduction in neural networks using methods such as channel pruning [134, 135, 136] and quantisation [137, 138, 139]. In particular for video compression, many works used integer arithmetic in order to efficiently implement trained neural networks on different hardware platforms. For example, the work in [140] proposes a training methodology to handle low precision multiplications, proving that very low precision is sufficient not just for running trained networks but also for training them. Similarly, the work in [141] considers the problem of using variational latent-variable models for data compression and proposes integer networks as a universal solution of range coding as an entropy coding technique. They demonstrate that such models enable reliable cross-platform encoding and decoding of images using variational models. Moreover, in order to ensure deterministic implementations on hardware platforms, they approximate non-linearities using lookup tables. Finally, an efficient implementation of matrix-based intra prediction is proposed in [142], where a performance analysis evaluates the challenges of deploying models with integer arithmetic in video coding standards. Inspired by this knowledge, this paper develops a fast and cost-effective implementation of the proposed attention-based architecture using integer precision approximations. As shown Section 7.5.2, while such approximations can significantly reduce the complexity, the associated drop of performance is still not negligible.

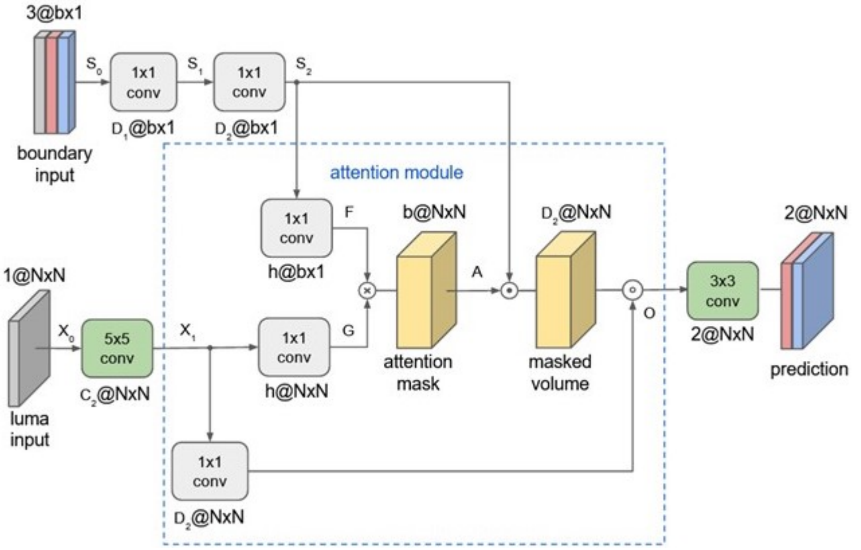
7.4 Multi-model architectures

This section introduces a new multi-model architecture which improves the baseline attention-based approach [6]. An introduction of the notation and formalities can be found in the previous chapter, please refer at Section 6.4. The main improvement comes from its block-size agnostic property as the proposed approach only requires one model for all block sizes. Furthermore, a range of simplifications is proposed with the aim to reduce the complexity of related attention-based architectures while preserving prediction performance as much as possible. The proposed simplifications include a framework for complexity reduction of the convolutional operations, a simplified cross-component boundary branch using sparse autoencoders and insights for fast and cost-effective implementations with integer precision approximations. Figure 7.1 illustrates the proposed multi-model attention-based schemes with the integration of the simplifications described in this section.

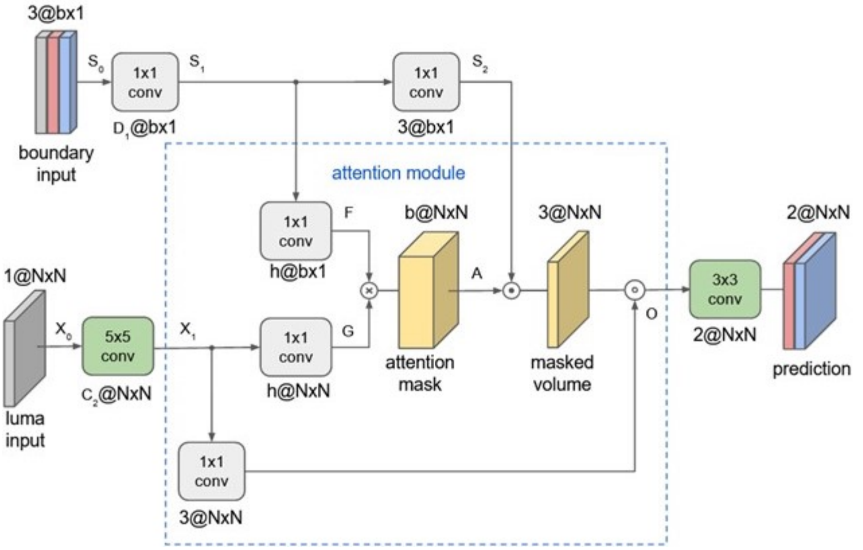
7.4.1 Multi-model size agnostic architecture

In order to handle variable block sizes, previous NN-based chroma intra-prediction methods employ different architectures for blocks of different sizes. These architectures differ in the dimensionality of the networks, which depend on give block size, as a trade-off between model complexity and prediction performance [119]. Given a network structure, the depth of the convolutional layers is the most predominant factor when dealing with variable input sizes. This means that increasingly complex architectures are needed for larger block sizes, in order to ensure proper generalisation for these blocks which have higher content variety. Such a factor significantly increases requirements for inference because of the number of multiple architectures.

In order to streamline the inference process, this work proposes a novel multi-model architecture that is independent of the input block size. Theoretically, a convolutional filter can be applied over any input space. Therefore, the fully-convolutional nature of the proposed architecture (1×1 kernels for the cross-



Scheme 1: Multi-model architecture with simplified convolutions



Scheme 2: Scheme 1 with simplified cross-component branch

Figure 7.1: Proposed multi-model attention-based architectures with the integration of the simplifications introduced in this paper. Notice that Scheme 1 adopts the multi-model architecture (Section 7.4.1) applying the methodology in Section 7.4.2 to simplify the convolutional layers within the luma convolutional branch and the prediction branch, and Scheme 2 further simplifies Scheme 1 by applying the methodology in Section 7.4.3 to simplify the cross-component boundary branch. More details about the model's hyperparameters and a description of the referred schemes can be found in Section 7.5.

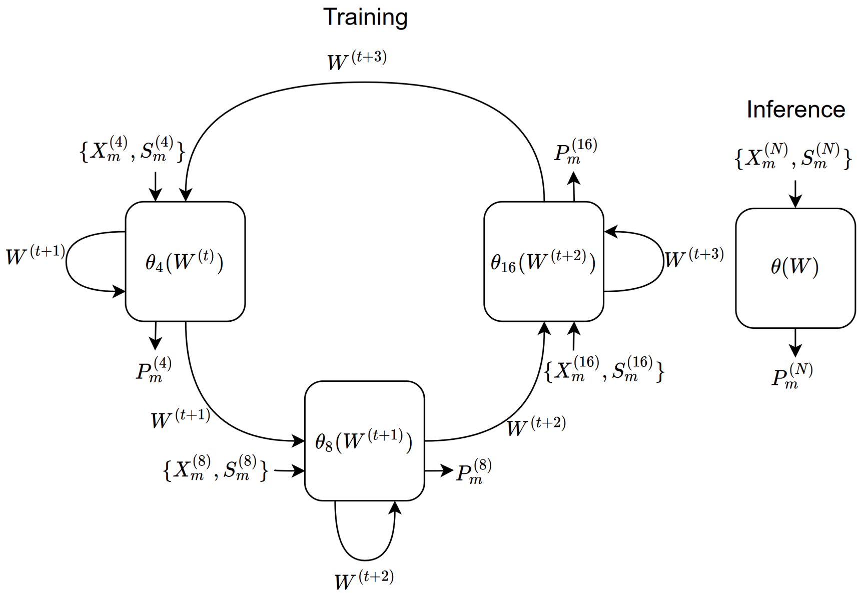


Figure 7.2: Illustration of the proposed multi-model training and inference methodologies. Multiple block-dependent models $\theta_N(W^{(t)})$ are used during training time. A size-agnostic model with a single set of trained weighs W is then used during inference.

component boundary branch and 3×3 kernels for the luma convolutional one) allows the design of a size agnostic architecture. As shown in Figure 7.2, the same task can be achieved using multiple models with different input sizes sharing the weights, such that a unified set of filters can be used a posterior, during inference. The given architecture must employ a number of parameters that is sufficiently large to ensure proper performance for larger blocks, but not too large to incur overfitting for smaller blocks.

Figure 7.3 describes the algorithmic methodology employed to train the multi-model approach. As defined in Section 6.4, the co-located luma block $X_0 \in \mathbb{R}^{N\times N}$ and the cross-component volume $S_0 \in \mathbb{R}^{3\times b}$ are considered as inputs to the chroma prediction network. Furthermore, for training of a multi-model the ground-truth is defined as $Z_m^{(N)}$, for a given input $\{X_m^{(N)}, S_m^{(N)}\}$, and the set of instances from a database of M samples or batches is defined as $\{X_m^{(N)}, S_m^{(N)}, Z_m^{(N)}\}$, where $m = 0, 1 \ldots M-1$ and $N \in \{4,\ 8,\ 16\}$ is the set of supported square block sizes $N \times N$ (the method can be extended to a different set of sizes). As shown in Figure 7.2, multiple block-dependent models $\theta_N(W)$ with shared weights W are updated in a concurrent way following the order of supported block sizes. At training step t, the individual model $\theta_N(W^{(t)})$ is updated obtaining a new set of weights $W^{(t+1)}$. Finally, a single set of trained weights W is used during inference, obtaining a size-agnostic model $\theta(W)$. Model parameters are updated by minimising the Mean Square Error

```
Require: $\{X_m^{(N)}, S_m^{(N)}, Z_m^{(N)}\}$, $m \in [0, M)$, $N \in \{4, 8, 16\}$
Require: $\theta_N(W^{(t)})$: $N$ model with shared weights $W^{(t)}$
Require: $\mathcal{L}_{reg}^{(t)}$: Objective function at training step $t$
$t \leftarrow 0$ (initialise timestep)
while $\theta_t$ not converged do
    for $m \in [0, M)$ do
        for $N \in \{4, 8, 16\}$ do
            $t \leftarrow t + 1$
            $\mathcal{L}_{reg}^{(t)} \leftarrow MSE(Z_m^{(N)}, \theta_N(X_m^{(N)}, S_m^{(N)}; W^{(t-1)}))$
            $g^{(t)} \leftarrow \nabla_W \mathcal{L}_{reg}^{(t)}$ (get gradients at step $t$)
            $W^{(t)} \leftarrow optimiser(g^{(t)})$
        end for
    end for
end while
```

Figure 7.3: Training algorithm for the proposed multi-model architecture.

(MSE) regression loss $\mathcal{L}_{reg}$, as in:

$$\mathcal{L}_{reg}^{(t)} = \frac{1}{C \cdot N^2} \| Z_m^{(N)} - \theta_N(X_m^{(N)}, S_m^{(N)}; W^{(t-1)} \|_2^2, \tag{7.1}$$

where $C = 2$ refers to the number of predicted chroma components, and $\theta_N(W^{(t-1)})$ is the block-dependent model at training step $t - 1$.

7.4.2 Simplified convolutions

Convolutional layers are responsible for most of the network's complexity. For instance, based on the network hyperparameters from experiments in Section 7.5, the luma convolutional branch and the prediction head branch (with 3×3 convolutional kernels) alone contain $46,882$ out of $51,714$ parameters, which constitute more than 90% of the parameters in the entire model. Therefore, the model complexity can be significantly reduced if convolutional layers can be simplified. This subsection explains how a new simplified structure beneficial for practical implementation can be devised by removing activation functions, i.e. by removing non-linearities. It is important to stress that such process is devised only for application on carefully selected layers, i.e. for branches where such simplification does not significantly reduce expected performance.

Consider specific two-layer convolutional branch (e.g. luma convolutional branch from Figure 2) formulated as:

$$Y = \mathcal{R}(W_2 * \mathcal{R}(W_1 * X + b_1) + b_2) \tag{7.2}$$

where C_i are the number of features in layer i, $b_i \in \mathrm{I\!R}^{C_i}$ are biases, $K_i \times K_i$ are

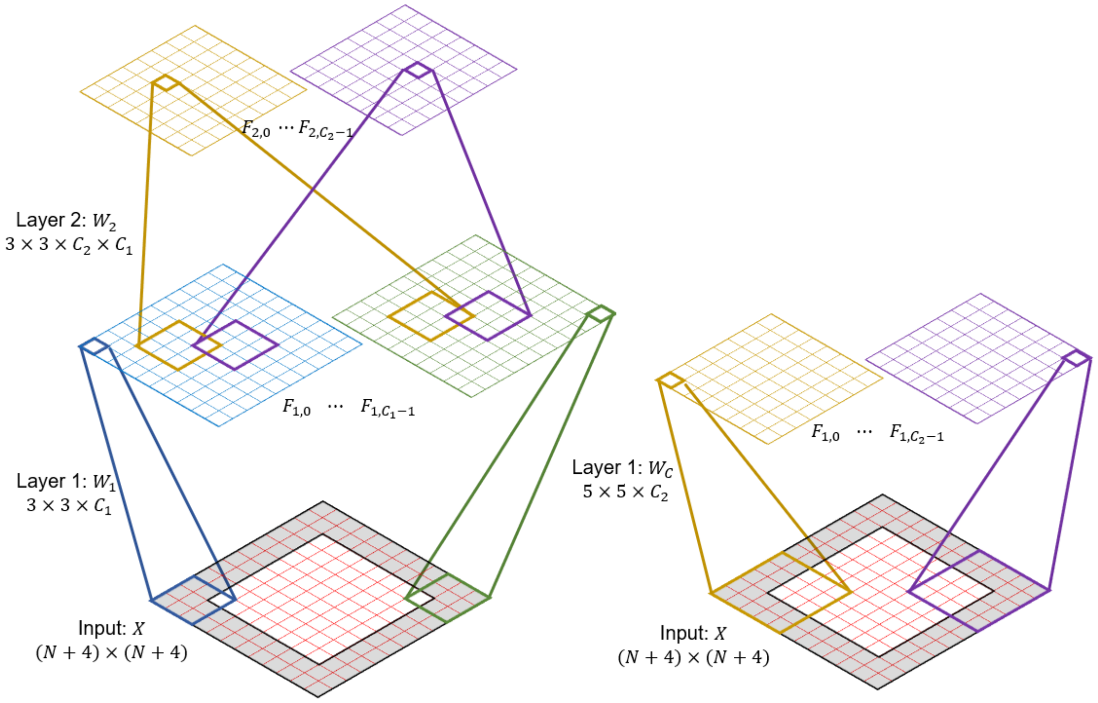


Figure 7.4: Visualisation of the receptive field of a 2-layer convolutional branch with 3×3 kernels. Observe that an output pixel in layer 2 is computed by applying a 3×3 kernel over a field F_1 of 3×3 samples from the first layer's output space. Similarly, each of the F_1 values are computed by means of another 3×3 kernel looking at a field F_0 of 5×5 samples over the input.

square convolutional kernel sizes, $W_1 \in \mathbb{R}^{K_1^2 \times C_0 \times C_1}$ and $W_2 \in \mathbb{R}^{K_2^2 \times C_1 \times C_2}$ are the weights and bias of the first $(i=1)$ and the second $(i=2)$ layers, respectively, C_0 the dimensions of the input feature map, $\mathcal{R}$ is a Rectified Linear Unit (ReLU) non-linear activation function and $*$ denotes convolution operation. Input to the branch is $X \in \mathbb{R}^{N^2 \times C_0}$ and the result is a volume of features $Y \in \mathbb{R}^{N^2 \times C_2}$, which correspond to X_0 and X_2, respectively. Removing non-linearities, the given branch can be simplified as:

$$\hat{Y} = W_2 * (W_1 * X + b_1) + b_2, \tag{7.3}$$

where it can be observed that a new convolution and bias terms can be defined using trained parameters from the two initial layers, to form a new single layer:

$$\hat{Y} = W_c * X + b_c, \tag{7.4}$$

where $W_c \in \mathbb{R}^{[\hat{K}^2 \times C_0] \times C_2}$ is the function of W_1 and W_2 with $\hat{K} = K_1 + K_2 - 1$, and b_c is a constant vector derived from W_2, b_1 and b_2. Figure 7.4 (a) illustrates the operations performed in Eq. 7.2 for $K_1 = K_2 = 3$ and $C = 1$. Analysing the receptive field of the whole branch, a pixel within the output volume Y is computed by applying a $K_2 \times K_2$ kernel over a field F_1 from the first layer's output space.

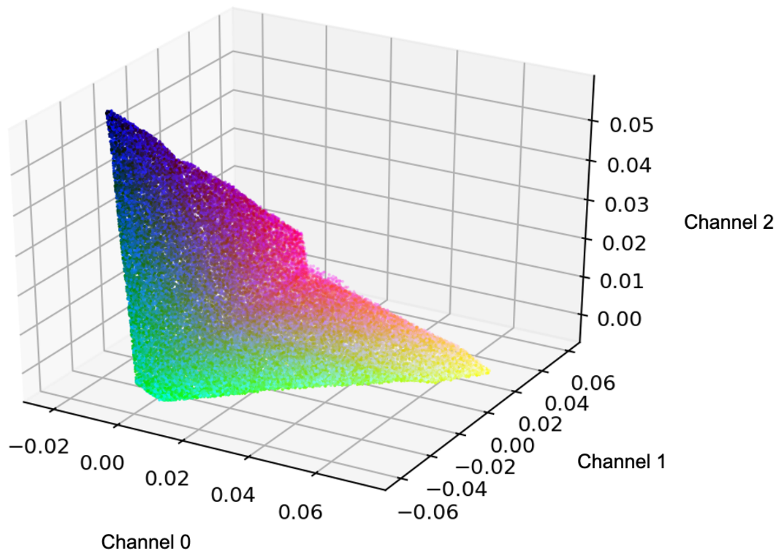


Figure 7.5: Visualisation of the learnt colour space resulting of encoding input YCbCr colours to the 3-dimensional hidden space of the autoencoder.

Similarly, each of the F_1 values are computed by means of another $K_1 \times K_1$ kernel looking at a field F_0. Without the non-linearities, and equivalent of this process is simplified, Figure 7.4 (b) and Eq. 7.4. Notice that $\hat{K} = K_1 + K_2 - 1$ equals 5 in the example in Figure 7.4. For a variety of parameters, including the values of C_0, C_i and K_i used in [6] and in this paper, this simplification of concatenated convolutional layers allows reduction of model's parameters at inference time, which will be shown in Section 7.5.1.

Finally, it should be noted that we limit the removal of activation functions only to branches which include more than one layer, from which at least one layer has $K_i > 1$, and only the activation functions between layers in the same branch are removed (to be able to merge them as in Equation 7.4). In such branches with at least one $K_i > 1$ the number of parameters is typically very high, while the removal of non-linearities typically does not impact prediction performance. Activation functions are not removed from the remaining layers. It should be noted that in the attention module and at the intersections of various branches the activation functions are critical and therefore are left unchanged. Section 7.5.1 performs an ablation test to evaluate the effect of removing the non-linearities, and a test to evaluate how would a convolutional branch directly trained with large-support kernels $\hat{K}$ perform.

7.4.3 Simplified cross-component boundary branch

In the baseline model, the cross-component boundary branch transforms the boundary inputs $S \in \mathbb{R}^{3 \times b}$ into D_J-dimensional feature vectors. More specifically, after applying $J = 2$ consecutive 1×1 convolutional layers, the branch encodes each boundary colour into a high dimensional feature space. It should be noted that a colour is typically represented by 3 components, indexed within a system of coordinates (referred to as the colour space). As such, a three-dimensional feature space can be considered as the space with minimum dimensionality that is still capable of representing colour information. Therefore, this work proposes the use of autoencoders (AE) to reduce the complexity of the cross-component boundary branch, by compacting the D-dimensional feature space into a reduced, 3-dimensional space. An AE tries to learn an approximation to the identity function $h(x) \approx x$ such that the reconstructed output $\hat{x}$ is as close as possible to the input x. The hidden layer will have a reduced dimensionality with respect to the input, which also means that the transformation process may introduce some distortion, i.e. the reconstructed output will not be identical to the input.

An AE consists of two networks, the encoder f which maps the input to the hidden features, and the decoder g which reconstructs the input from the hidden features. Applying this concept, a compressed representation of the input can be obtained by using the encoder part alone, with the goal of reducing the dimensionality of the input vectors. The encoder network automatically learns how to reduce the dimensions of the input vectors, in a similar fashion to what could be obtained applying a manual Principal Component Analysis (PCA) transformation. The transformation learned by the AE can be trained using the same loss function that is used in the PCA process [143]. Figure 7.5 shows the mapping function of the resulting colour space when applying the encoder network over the YCbCr colour space.

Overall, the proposed simplified cross-component boundary branch consists of two 1×1 convolutional layers using Leaky ReLU activation functions with a slope $\alpha = 0.2$. First, a D-dimensional layer is applied over the boundary inputs S to obtain $S_1 \in \mathbb{R}^{D \times b}$ feature maps. Then, S_1 is fed to the AE's encoder layer f with output 3 dimensions, to obtain the hidden feature maps $S_2 \in \mathbb{R}^{3 \times b}$. Finally, a third 1×1 convolutional layer (corresponding to the AE decoder layer g) is applied to generate the reconstructed maps $\tilde{S}_1$ with D-dimensions. Notice that the decoder layer is only necessary during the training stage to obtain the reconstructed inputs necessary to derive the values of the loss function. Only the encoder layer is needed when using the network, in order to transform the input feature vectors into the 3 dimensional, reduced vectors. Figure 7.1 illustrates the branch architecture and its integration within the simplified multi-model.

Finally, in order to interpret the behaviour of the branch and to identify prediction patterns, a sparsity constraint can be imposed on the loss function. Formally, the following can be used:

$$\mathcal{L}_{AE} = \frac{\lambda_r}{D \cdot b} \|S_1 - \tilde{S}_1\|_2^2 + \frac{\lambda_s}{3 \cdot b} \|S_2\|_1, \tag{7.5}$$

where the right-most term is used to keep the activation functions in the hidden space remain inactive most of the time, and only return non-zero values for the most descriptive samples. In order to evaluate the effect of the sparsity term, Section 7.5.1 performs an ablation test that shows its positive regularisation properties during training.

The objective function in Equation 7.5 can be updated such that the global multi-model loss $\mathcal{L}$ considers both $\mathcal{L}_{reg}$ and $\mathcal{L}_{AE}$ as:

$$\mathcal{L} = \lambda_{reg} \mathcal{L}_{reg} + \lambda_{AE} \mathcal{L}_{AE} \tag{7.6}$$

where λ_{reg} and λ_{AE} control the contribution of both losses.

7.4.4 Integer precision approximation

While the training algorithm results in IEEE-754 64-bit floating point weights and prediction buffers, an additional simplification is proposed in this paper whereby the network weights and prediction buffers are represented using fixed-point integer arithmetic. This is beneficial for deployment of resulting multi-models in efficient hardware implementations, which complex operations such as Leaky ReLU and softmax activation functions can become serious bottlenecks. All the network weights obtained after the training stage are therefore appropriately quantised to fit 32-bit signed integer values. it should be noted that integer approximation introduces quantisation errors, which may have an impact on the performance of the overall predictions.

In order to prevent arithmetic overflows after performing multiplications or additions, appropriate scaling factors are defined for each layer during each of the network prediction steps. To further reduce the complexity of the integer approximation, the scaling factor K_l for a given layer l is obtained as a power of 2, namely $K_l = 2^{O_l}$, where O_l is the respective precision offset. This ensures that multiplications can be performed by means of simple binary shifts. Formally, the integer weights $\tilde{W}_l$ and biases $\tilde{b}_l$ for each layer l in the network with weights W_l and bias b_l can be obtained as:

$$\tilde{W}_l = \lfloor W_l \cdot 2^{O_l} \rfloor ; \tilde{b}_l = \lfloor b_l \cdot 2^{O_l} \rfloor . \tag{7.7}$$

The offset O_l depends on the offset used on the previous layer O_{l-1}, as well as on an internal offset O_x necessary to preserve as much decimal information as possible, compensating for the quantisation that occurred in the previous layer, namely $O_l = O_x - O_{l-1}$.

Furthermore, in this approach the values predicted by the network are also integers. In order to avoid defining large internal offsets at each layer, namely having large values of O_x, an additional stage of compensation is applied to the predicted values, to keep their values in the range of 32-bit signed integer. For this purpose, another offset O_y is defined, computed as $O_y = O_x - O_l$. The values generated by layer l are then computed as:

$$Y_l = ((\tilde{W}_l^T X_l + \tilde{b}_l) + (1 << (O_y - 1))) >> O_y, \tag{7.8}$$

where $<<$ and $>>$ represent the left and right binary shifts, respectively, and the offset $(1 << (O_y - 1))$ is considered to reduce the rounding error.

Complex operations requiring floating point divisions need to be approximated to integer precision. The Leaky ReLU activation functions applied on the cross-component boundary branch use a slope $\alpha = 0.2$ which multiplies the negative values. Such an operation can be simply approximated by defining a new activation function $\tilde{A}(x)$ for any input x as follows:

$$\tilde{A}(x) = \left\{ \begin{array}{ll} 0 & : x \geq 0 \\ 26 \cdot x >> 7 & : x < 0 \end{array} \right\} \tag{7.9}$$

Conversely, the softmax operations used in the attention module are approximated following a more complex methodology, similar to the one used in [144]. Consider the matrix M as defined in Equation 7.10 and a given row j in M, and a vector m_j as input to the softmax operation. First, all elements m_j in a row are subtracted by the maximum element in the row, namely:

$$\hat{m}_{i,j} = (m_{i,j}/T - \max_i(m_{i,j}/T)) \tag{7.10}$$

where T is the temperature of the softmax operation, set to 0.5 as previously mentioned. The transformed elements $\hat{m}_{i,j}$ range between the minimum signed integer value and zero, because the arguments $\hat{m}_{i,j}$ are obtained by subtracting the elements in M by the maximum element in each row. To further reduce the possibility of overflows, this range is further clipped to a minimum negative value, set to predetermined number V_e, so that any $\hat{m}_{i,j} < V_e$ is set equal to V_e.

The elements $\hat{m}_{i,j}$ are negative integer numbers within the range $[V_e, 0]$, meaning there is a fixed number of $N_e = |V_e| + 1$ possible values they can assume. To further simplify the process, such an exponential function is replaced by a pre-computed look-up table containing N_e integer elements. To minimise the approximation error, the exponentials are scaled by a given scaling factor before being approximated to the nearest integer and stored in the corresponding look-up table *LUT-EXP*. Formally, for a given index k, where $0 \leq k \leq N_e - 1$, the k-th integer input is obtained as $s_k = V_e + k$. The k-th element in the look-up table can then be computed as the approximated, scaled exponential value for s_k, or:

$$LUT\text{-}EXP(k) = \lfloor K_e e^{s_k} \rfloor \tag{7.11}$$

where $K_e = 2^{O_e}$ is the scaling factor, chosen in a way to maximise the preservation of the original decimal information. When using the look-up table during the prediction process, given an element $\hat{m}_{i,j}$ the corresponding index k can be retrieved as: $k = |V_e - \hat{m}_{i,j}|$, to produce the numerator in the softmax function.

The integer approximation of the softmax function can then be written as:

$$\hat{\alpha}_{j,i} = \frac{LUT\text{-}EXP(|V_e - \hat{m}_{i,j}|)}{D(j)}, \tag{7.12}$$

where:

$$D(j) = \sum_{n=0}^{b-1} LUT\text{-}EXP(|V_e - \hat{m}_{n,j}|), \tag{7.13}$$

Equation 7.12 implies performing an integer division between the numerator and denominator. This is not ideal, and integer divisions are typically avoided in low complexity encoder implementations. A simple solution to remove the integer division can be obtained by replacing it with a binary shift. However, a different approach is proposed in this paper to provide a more robust approximation that introduces smaller errors in the division. The denominator $D(j)$ as in Equation 7.13 is obtained as the sum of b values extracted from *LUT-EXP*, where b is the number of reference samples extracted from the boundary of the block. As such, the largest blocks under consideration (16×16) will result in the largest possible value of reference samples b_{MAX}. This means that the maximum value that this denominator can assume is obtained when $b = b_{MAX}$ and when all input $\hat{m}_{i,j} = 0$ (which correspond to $LUT\text{-}EXP(|V_e|) = K_e$), corresponding to $V_s = b_{MAX} K_e$. Similarly, the minimum value (obtained when $\hat{m}_{i,j} = V_e$) is 0. Correspondingly, $D(j)$, can assume any positive integer value in the range $[0, V_s]$.

Considering a given scaling factor $K_s = 2^{O_s}$, integer division by $D(j)$ can be approximated using a multiplication by the factor $M(j) = \lfloor K_s / D(j) \rfloor$. A given value of $M(j)$ could be computed for all $V_s + 1$ possible values of $D(j)$. Such values can then be stored in another look-up table *LUT-SUM*. Clearly though, V_s is too large which means *LUT-SUM* would be impractical to use due to storage and complexity constraints. For that reason, a smaller table is used, obtained by quantising the possible values of $D(j)$. A pre-defined step Q is used, resulting in $N_s = (V_s + 1)/Q$ quantised values of $D(j)$. The table *LUT-SUM* of size N_s is then filled accordingly, where each element in the table is obtained as:

$$LUT\text{-}SUM(l) = \lfloor K_s/(lQ) \rfloor \tag{7.14}$$

Finally, when using the table during the prediction process, given an integer sum $D(j)$, the corresponding index l can be retrieved as: $l = \lfloor D(j)/Q \rfloor$. Following from these simplifications, given an input $\hat{m}_{i,j}$ obtained as in Equation 7.10, the integer sum $D(j)$ obtained from Equation 7.13, and a quantisation step Q, the simplified integer approximation of the softmax function can eventually be obtained as:

$$\tilde{\alpha}_{j,i} = LUT\text{-}EXP(|V_e - \hat{m}_{i,j}|) \cdot LUT\text{-}SUM(\lfloor D(j)/Q \rfloor), \tag{7.15}$$

Notice that $\tilde{\alpha}_{j,i}$ values are finally scaled by $K_o = K_e \cdot K_s$.

7.5 Experiments

7.5.1 Architecture configurations

The proposed multi-model architectures and simplifications (Section 7.4) are implemented in 3 different schemes:

- Scheme 1: Multi-model architecture (Section 7.4.1) applying the methodology in Section 7.4.2 to simplify the convolutional layers within the luma convolutional branch and the prediction branch, as illustrated in Figure 7.1.
- Scheme 2: The multi-model architecture in Scheme 1 applying the methodology in Section 7.4.3 to simplify the cross-component boundary branch. As shown in Figure 7.1, the integration of the simplified branch requires modification of the initial architecture with changes in the attention module and the prediction branch.
- Scheme 3: Architecture in Scheme 1 with the integer precision approximations described in Section 7.4.4.

In contrast to previous state-of-the-art methods, the proposed multi-model does not need to adapt its architecture to the input block size. Notice that the fully-convolutional architecture introduced in [6] enables this design and is able to significantly reduce the complexity of the cross-component boundary branch in [119], which uses size-dependent fully-connected layers. Table 7.1 shows the network hyperparameters of the proposed schemes during training, whereas Table 7.2 shows

Table 7.1: Network hyperparameters during training

Branch $(C_{in}, K \times K, C_{out})$	**Scheme 1 & 3**	**Scheme 2**
CC Boundary	$3, 1 \times 1, 32$ $32, 1 \times 1, 32$	$3, 1 \times 1, 32$ $32, 1 \times 1, 3$
Luma Convolutional	$1, 3 \times 3, 64$ $64, 3 \times 3, 64$	$1, 3 \times 3, 64$ $64, 3 \times 3, 64$
Attention Module	$32, 1 \times 1, 16$ $64, 1 \times 1, 16$ $64, 1 \times 1, 32$	$32, 1 \times 1, 16$ $64, 1 \times 1, 16$ $64, 1 \times 1, 3$
Prediction Head	$32, 3 \times 3, 32$ $32, 1 \times 1, 2$	$3, 3 \times 3, 3$ $3, 1 \times 1, 2$

Table 7.2: Network hyperparameters during inference

Branch $(C_{in}, K \times K, C_{out})$	**Scheme 1 & 3**	**Scheme 2**
CC Boundary	$3, 1 \times 1, 32$ $32, 1 \times 1, 32$	$3, 1 \times 1, 32$ $32, 1 \times 1, 3$
Luma Convolutional	$1, 5 \times 5, 64$	$1, 5 \times 5, 64$
Attention Module	$32, 1 \times 1, 16$ $64, 1 \times 1, 16$ $64, 1 \times 1, 32$	$32, 1 \times 1, 16$ $64, 1 \times 1, 16$ $64, 1 \times 1, 3$
Prediction Head	$32, 3 \times 3, 2$	$3, 3 \times 3, 2$

the resulting hyperparameters for inference after applying the proposed simplifications. As shown in Tables 7.4 and 7.5, the employed number of parameters in the proposed schemes represents the trade-off between complexity and prediction performance, within the order of magnitude of related attention-based CNNs in [6]. The proposed simplifications significantly reduce (around 90%) the original training parameters, achieving lighter architectures for inference time. Table 7.4 show that the inference version of Scheme 2 reduces to around 85%, 96% and 99% the complexity of the hybrid CNN models in [119] and to around 82%, 96% and 98% the complexity of the attention-based models in [6], for $4 \times 4, 8 \times 8$ and 16×16 input block sizes, respectively. Finally, in order to provide more insights about the computational cost and compare the proposed schemes with the state-of-the-art methods, Table 7.3 shows the number of floating point operations (FLOPs) for each architecture per block size. The reduction of operations (e.g. additions and matrix multiplications) to arrive to the predictions is one the predominant factors towards the given speedups. Notice the significant reduction of FLOPs for the proposed inference models.

In order to obtain a preliminary evaluation of the proposed schemes and to compare their prediction performance with the state-of-the-art methods, the trained models were tested on the DIV2K validation set (with 100 multi-resolution images)

Table 7.3: FLOPs per block size

Model (parameters)	4 × 4	8 × 8	16 × 16
Hybrid CNN [119]	51465	187273	711945
Attention-based CNN [6]	42795	165451	186146
Scheme 1 & 3 (train/inference)	102859/13770		
Scheme 2 (train/inference)	79103/7225		

Table 7.4: Model complexity per block size

Model (parameters)	4 × 4	8 × 8	16 × 16
Hybrid CNN [119]	24435	96116	369222
Attention-based CNN [6]	21602	83106	186146
Scheme 1 & 3 (train/inference)	51714/7074		
Scheme 2 (train/inference)	39371/3710		

Table 7.5: Prediction performance per block size

Model (PSNR)	4 × 4	8 × 8	16 × 16
Hybrid CNN [119]	28.61	31.47	33.36
Attention-based CNN [6]	30.23	33.13	36.13
[6] with multi-model	30.55	33.21	36.05
Scheme 1 single layer training	30.36	33.05	35.88
Scheme 2 without sparsity	29.89	32.66	35.64
(proposed) Scheme 1	30.54	33.20	35.99
(proposed) Scheme 2	29.91	32.68	35.67

by means of averaged PSNR. Test samples were obtained with the same methodology as used in Section 6.5.1 for generating the training dataset. Notice that this test uses the training version of the proposed schemes. As shown in Table 7.5, the multi-model approach introduced in Scheme 1 improves the attention-based CNNs in [6] for 4×4 and 8×8 blocks, while only a small performance drop can be observed for 16×16 blocks. However, because of using a fixed architecture for all block sizes, the proposed multi-model architecture averages the complexity of the individual models in [6] (Table 7.4), slightly increasing the complexity of the 4×4 model and simplifying the 16×16 architecture. The complexity reduction in the 16×16 model leads to a small drop in performance. As can be observed from Table 7.5 , the generalisation process induced by the multi-model methodology ([6] with multi-model, compared to [6]) can minimise such drop by distilling knowledge from the rest of block sizes, which is especially evident for 8×8 blocks where a reduced architecture can improve the state-of-the-art performance.

Finally, the simplifications introduced in Scheme 2 (e.g. the architecture changes required to integrate the modified cross-component boundary branch within the original model) lower the prediction performance of Scheme 1. However, the highly simplified architecture is capable of outperforming the hybrid CNN models in [119], observing training PSNR improvements of an additional 1.30, 2.21 and 2.31 dB for $4 \times 4, 8 \times 8$ and 16×16 input block sizes, respectively. The combination of attention-based architectures with the proposed multi-model methodology (Scheme 1) considerably improves the NN-based chroma intra-prediction methods in [119], showing training PSNR improvements by additional 1.93, 1.73 and 2.68 dB for the supported block sizes. In Section 7.5.2 it will be shown how this relatively small PSNR differences lead to significant differences in codec performance.

Several ablations were performed in order to evaluate the effects of the proposed simplifications. First, the effect of the multi-model methodology is evaluated by directly converting the models in [4] to the size-agnostic architecture in Scheme 1 but without the simplifications in Section 7.4.2 ([6] with multi-model). As can be shown in Table 7.5, this methodology improves the 4×4 and 8×8 models, with special emphasis in the 8×8 case where the number of parameters is smaller than in [6]. Moreover, the removal of non-linearities in Scheme 1 does not significantly affect the performance, with a negligible PSNR loss of around 0.3 dB ([6] with multi-model compared with Scheme 1). Secondly, in order to evaluate the simplified convolutions methodology in Section 7.4.2, a version of Scheme 1 was trained with single-layer convolutional branches with large support kernels (e.g. instead of training 2 linear layers with 3×3 kernels and then combining them into 5×5 kernels for inference, training directly a single-layer branch with 5×5 kernels). Experimental results show the positive effects of the proposed methodology, observing a significant drop of performance when a single-layer trained branch is applied (Scheme 1 with single layer training compared with Scheme 1). Finally, the effect of the sparse autoencoder of Scheme 2 is evaluated by removing the sparsity term in Equation 7.5. As can be observed, the regularisation properties of the sparsity term, i.e. preventing large activations, boosts the generalisation capabilities of the multi-model and slightly increases the prediction performance by around 0.2 dB. (Scheme 2 without sparsity compared with Scheme 2).

7.5.2 Simulation Results

The VVC reference software VTM-7.0 is used as our benchmark and our proposed methodology is tested under the Common Test Conditions (CTC) [127], using the suggested all-intra configuration for VVC with a QP of 22, 27, 32 and 37. In order to fully evaluate the performance of the proposed multi-models, the encoder configuration is constrained to support only square blocks of $4 \times 4, 8 \times 8$ and 16×16 pixels. A corresponding VVC anchor was generated under these conditions. BD-rate is adopted to evaluate the relative compression efficiency with respect to the latest VVC anchor. Test sequences include 26 video sequences of different resolutions: 3840×2160 (Class A1 and A2), 1920×1080 (Class B), 832×480 (Class C), 416×240 (Class D), 1280×720 (Class E) and screen content (Class F). The "EncT" and "DecT" are "Encoding Time" and "Decoding Time", respectively.

A colour analysis is performed in order to evaluate the impact of the chroma channels on the final prediction performance. As suggested in previous colour prediction works [35], standard regression methods for chroma prediction may not be effective for content with wide distributions of colours. A parametric model which is trained to minimise the Euclidean distance between the estimations and the ground truth commonly tends to average the colours of the training examples and hence produce desaturated results. As shown in Figure 7.6, several CTC sequences are analysed by computing the logarithmic histogram of both chroma components. The width of the logarithmic histograms is compared to the compression performance in Table 7.6. Gini index [145] is used to quantify the width of the histograms, obtained as

$$Gini(H) = 1 - \sum_{b=0}^{B-1} \left(\frac{H(b)}{\sum_{k=0}^{B-1} H(k)} \right)^2 \tag{7.16}$$

being H a histogram of B bins for a given chroma component. Notice that the average value between both chroma components is used in Table 7.6. A direct correlation between Gini index and coding performance can be observed in Table 7.6, suggesting that Scheme 1 performs better for narrower colour distributions. For instance, the Tango 2 sequence with a Gini index of 0.63 achieves an average Y/Cb/Cr

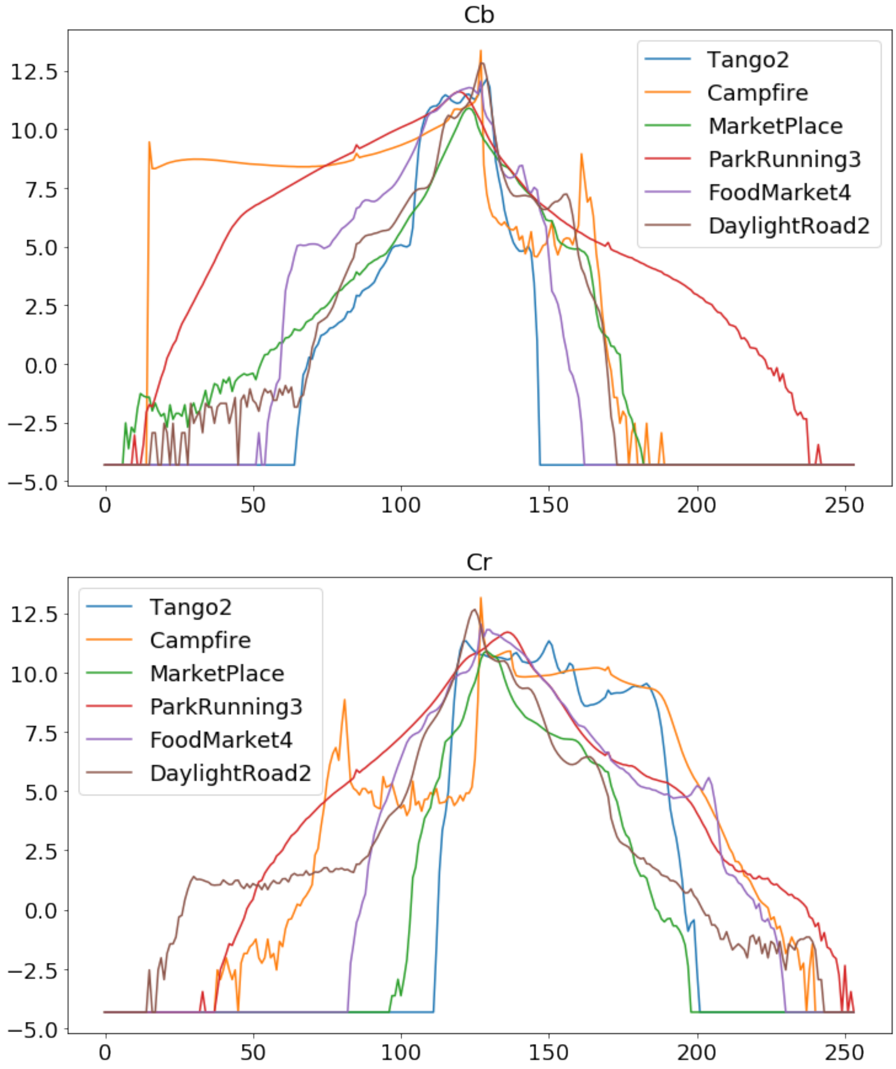


Figure 7.6: Comparison of logarithmic colour histograms for different sequences.

Table 7.6: BD-Rates (%) sorted by Gini index

Sequence	**Scheme 1**			Gini
	Y	Cb	Cr	
Tango2	-0.46	-8.13	-3.13	0.63
MarketPlace	-0.59	-2.46	-3.06	0.77
FoodMarket4	-0.16	-1.60	-1.55	0.85
DaylightRoad2	-0.09	-5.74	-1.85	0.89
Campfire	-0.21	0.14	-0.88	0.98
ParkRunning3	-0.31	-0.73	-0.77	0.99

BD-rates of -0.46%/-8.13%/-3.13%, whereas Campfire with wide colour histograms (Gini index of 0.98), obtains average Y/Cb/Cr BD-rates of -0.21%/0.14%/-0.88%. Although the distributions of chroma channels can be a reliable indicator of prediction performance, wide colour distributions may not be the only factor in restricting chroma prediction capabilities of proposed methods, which can be investigated in future work.

A summary of the component-wise BD-rate results for all the proposed schemes and the related attention-based approach in [6] is shown in Table 7.7 for all-intra conditions. Scheme 1 achieves an average Y/Cb/Cr BD-rates of -0.25%/-2.38%/-1.80% compared with the anchor, suggesting that the proposed multi-model size agnostic methodology can improve the coding performance of the related attention-based block-dependent models. Besides improving the coding performance, Scheme 1 significantly reduces the encoding (from 212% to 164%) and decoding (from 2163% to 1302%) times demonstrating the positive effect of the inference simplification.

Finally, the proposed simplifications introduced in Scheme 2 and Scheme 3 further reduce the encoding and decoding time at the cost of a drop in the coding performance. In particular, the simplified cross-component boundary branch introduced in Scheme 2, achieves an average Y/Cb/Cr BD-rates of -0.13%/-1.56%/-1.63% and, compared to Scheme 1, reduces the encoding (from 164% to 146%) and decoding (from 1302% to 665%) times. Scheme 3 has lower reduction of encoding time (154%) than Scheme 2, but it achieves higher reduction in decoding time (665%), although the integer approximations lowers the performance achieving average Y/Cb/Cr BD-rates of -0.16%/-1.72%/-1.38%.

As described in Section 7.4, the simplified schemes introduced here tackle the complexity reduction of Scheme 1 with two different methodologies. Scheme 2 proposes direct modifications on the original architecture which need to be retrained before being integrated in the prediction pipeline. Conversely, Scheme 3 directly simplifies the final prediction process by approximating the already trained weights from Scheme 1 with integer-precision arithmetic. Therefore, the simulation results suggest that the methodology in Scheme 3 is better at retaining the original performance since a retraining process is not required. However, the highly reduced architecture in Scheme 2 is capable of approximating the performance of Scheme 3 and further reduce the decoder time.

Overall, the comparison results in Table 7.7 demonstrate that proposed models offer various trade-offs between compression performance and complexity. While it has been shown that the complexity can be significantly reduced, it is still not negligible. Challenges for future work include integerisation of the simplified scheme (Scheme 2) while preventing the compression drop observed for Scheme 3. Recent approaches, including a published one which focuses on intra prediction [142],

Table 7.7: BD-Rate (%) for all proposed schemes and [6] under all-intra CTC

	Class A1			Class A2			Class B			Class C			Class D			Class E			Class F			Overall			EncT [%]	DecT [%]
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr		
Scheme 1	-0.28	-3.20	-1.85	-0.25	-3.11	-1.54	-0.26	-2.28	-2.33	-0.30	-1.92	-1.57	-0.29	-1.70	-1.77	-0.13	-1.59	-1.45	-0.25	-2.38	-1.80	-0.25	-2.38	-1.80	164 %	1302 %
Scheme 2	-0.08	-1.24	-1.26	-0.12	-1.59	-1.31	-0.15	-1.80	-2.21	-0.20	-1.41	-1.62	-0.18	-1.42	-1.73	-0.08	-1.67	-1.40	-0.13	-1.56	-1.63	-0.13	-1.56	-1.63	146 %	665 %
Scheme 3	-0.19	-2.25	-1.56	-0.13	-2.44	-1.12	-0.16	-1.78	-2.05	-0.20	-1.44	-1.29	-0.20	-1.64	-1.41	-0.07	-0.75	-0.46	-0.16	-1.72	-1.38	-0.16	-1.72	-1.38	154 %	512 %
Anchor + [6]	-0.26	-2.17	-1.96	-0.22	-2.37	-1.64	-0.23	-2.00	-2.17	-0.26	-1.64	-1.41	-0.25	-1.55	-1.67	-0.03	-1.35	-1.77	-0.21	-1.90	-1.81	-0.21	-1.90	-1.81	212 %	2163 %

demonstrate that sophisticated integerisation approaches can help retain compression performance of originally trained models while enabling them to become significantly less complex and thus be integrated into future video coding standards.

7.6 Future work

The proposed attention-based methodology guides the fusion between the boundary and current luma block information. However, the predictive ability of CNNs is limited, and simply making the network deeper does not lead to more accurate predictions. On the other hand, side information is useful for performance improvement. In this spirit, we collaborated with Northwestern Polytechnical University (Xi'an, China) to improve the proposed network by spatial information refinement [146]. In particular, considering the different importance of boundary information for the predicted chroma pixel at each location, we propose a method for fusing reconstructed value data with the location information and finally, we use a convolutional layer instead of original chroma down-sampling filter to better preserve features in the luma information.

As future work, we aim to implement a complete multi-model for all VVC block sizes in order to ensure a full usage of the proposed approach building on the promising results shown in the constrained test conditions. Moreover, an improved approach for integer approximations may enable the fusion of all proposed simplifications, leading to a fast and powerful multi-model.

7.7 Conclusion

This chapter showcased the effectiveness of attention-based architectures in performing chroma intra-prediction for video coding. A novel size-agnostic multi-model and its corresponding training methodology were proposed to reduce the inference complexity of previous attention-based approaches. Moreover, the proposed multi-model was proven to better generalise to variable input sizes, outperforming state-of-the-art attention-based models with a fixed and much simpler architecture. Several simplifications were proposed to further reduce the complexity of the original multi-model. First, a framework for reducing the complexity of convolutional operations was introduced and was able to derive an inference model with around 90% fewer parameters than its relative training version. Furthermore, sparse autoencoders were applied to design a simplified cross-component processing model capable of further reducing the coding complexity of its preceding schemes. Finally, algorithmic insights were proposed to approximate the multi-model schemes in integer-precision arithmetic, which could lead to fast and hardware-aware implementations of complex operations such as softmax and Leaky ReLU activations.

The proposed schemes were integrated into the VVC anchor VTM-7.0, signalling the prediction methodology as a new chroma intra-prediction mode working in parallel with traditional modes towards predicting the chroma component samples. Experimental results show the effectiveness of the proposed methods, retaining compression efficiency of previously introduced neural network models, while offering two different directions for significantly reducing coding complexity, which translates to reduced encoding and decoding times.

Chapter 8

Conclusions

The requirement of colour has played an important role in the digitalisation process of the broadcasting industry. Colour television followed by the convergence of digital entertainment led to the replacement of deprecated analogue systems and the development of new processing and transmission methodologies which ensure more efficiency and flexibility, whilst maximising human perceptual plausibility. Moreover, new computer vision disciplines such as video enhancement and restoration have gained importance with the aim to bring back to life deprecated legacy content from digital archives. Inspired by the recent success of artificial intelligence and convolutional neural networks in many computer vision applications, this thesis explored the integration of such technologies to improve colour processing in different parts of the broadcasting workflow.

In the context of content production, this thesis tackles the prediction of colour from black and white content and explores the challenges to deploy a video colourisation pipeline into a realistic production and restoration workflow. In particular, Chapter 3 studies the use of Generative Adversarial Networks to perform fully-automatic image colourisation and identifies improved architectural features that lead to better training stability and reduction of the well-known averaging or desaturated effect. In order to reduce ambiguity of automatic colourisation, Chapter 4 explores exemplar-based methods which allow the interaction of the producer into the prediction pipeline with the aim to control the predicted outcomes by means of colour references. An attention mechanism is proposed to compute feature analogies from the reference image and to enable style transfer in an unsupervised way during the prediction process. With the aim to promote a cost-effective implementation, axial attention is adopted to reduce the overall complexity and to achieve a simple and fast architecture that is easily scalable to high resolution inputs. Finally, Chapter 5, explores how to perform video colourisation by applying the proposed image colourisation models frame-to-frame while coping with the associated temporal inconsistencies and flickering artefacts. A temporal stabilisation framework is proposed using scene-aware deep video priors, which generalises the DVP baseline [103] to complex sequences with multiple scenes. The framework includes an optimised few-shot training to reduce the processing time of DVP by removing its time response conditioned on the number of input frames. As a result, findings from Chapters 3, 4 and 5 are combined to propose an easily-deployable video colourisation prototype.

In the context of content distribution, colour prediction is a key element in most video coding standards. In particular, within intra frame coding, chroma information is predicted with the aim to remove correlations within local regions of the image and the cross-component domain. The size of the bitstream can be therefore reduced, and better compression rates can be achieved to enable more efficient content delivery to the broad audience. The research presented in this thesis investigates the use of neural networks for improving existing chroma prediction methods, aiming an efficient deployment within the latest video coding standards. In particular, Chapter 6 introduces a novel neural network architecture for chroma intra prediction that integrates attention modules to control the contribution of each neighbouring reference sample when computing the prediction of each chroma pixel in a given block sample. The proposed scheme better captures the relationship between the luma and chroma components, resulting in more accurate prediction samples. However, coding efficiency and the associated encoding and decoding times is as important as quality for modern video coding standards. Therefore, despite the high coding gains, the potential application of neural networks in real scenarios is limited. Chapter 7 focuses on complexity reduction with the aim to simplify the proposed attention-based architecture while retaining most of the original performance. Proposed simplifications include a framework to reduce the complexity of the convolutional operations, a simplified cross-component processing model using sparse autoencoders and a fast and cost-effective implementation using integer precision approximations.

In this chapter, Section 8.1 revisits the hypothesis described in Chapter 1, and how the research presented in this thesis addresses the hypothesis through the research questions also introduced in Chapter 1. Section 8.2 summarizes the research contributions of this thesis. Section 8.3 elaborates on the suggestions for future research introduced in the individual chapters of the thesis. Finally, Section 8.4 provides the closing remarks for this thesis.

8.1 Hypothesis and research questions

The hypothesis introduced in Chapter 1 is discussed in this section in light of the research presented in the previous chapters. Each of the associated research questions are addressed to provide more concise descriptions of the various contributions of this thesis.

Hypothesis: *AI-based colour processing techniques can be efficiently implemented into a current digital broadcasting workflow with a significant impact on video production and distribution.*

Research question 1: Can deep learning models for video colourisation be implemented into a production pipeline for content enhancement and restoration? Some areas of interest include reference-based methodologies, user/producer interactivity and efficient implementations for image and video prediction.

Although previous works attempted to solve video colourisation as an end-to-end fully automatic task [104], most relied significantly on single frame colourisation followed by propagation [4, 57, 56]. This is because image colourisation, compared to video colourisation, achieves higher visual quality and naturalness. However, propagation methods are usually complex and lead to amplification of errors. Chapter 5 demonstrated the effectiveness of Deep Video Prior (DVP) [103] to promote blind temporal consistency and how well-performing image-based colourisation methods from Chapters 3 and 4 can be efficiently applied to achieve stable video colourisation. Moreover, an easily-deployable pipeline is proposed, which combines the findings from the content production part of this thesis. The XCNET architecture, introduced in Chapter 4, achieves efficient reference-based colourisation and enables the involvement of the producer in the selection of colour references. Finally, efficiency is considered as an essential aspect in the design of most of the pipeline components. For instance, XCNET architecture achieves end-to-end inference by integrating attention modules which allow the computation of analogies with the reference image in an unsupervised way and reduce the computation overhead of previous methods relying on Patch Match [85] or similar nearest neighbour search. Moreover, axial attention allows a cost-effective implementation of the attention operations which facilitates scalability to high-resolution inputs. Another example is the design of the video stabilisation framework in Chapter 5, in which a few-shot training strategy is proposed to reduce the impact of prior learning by removing the time response of DVP baseline conditioned on the number of input frames. The proposed architecture is jointly trained to stabilise the input video and to cluster the input frames with the aim of learning scene-specific modes. Experimental results demonstrate the generalisation of DVP, obtaining improved performance in complex sequences with small amounts of training data and fewer iterations.

Research question 2: Which are the best practices to train a colourisation generative model to ensure content generalisation, colourfulness, temporal consistency and high resolution predictions?

Chapter 3 explores the use of conditional GANs for fully automatic image colourisation and introduces techniques that improve the stability of the adversarial loss during training, leading to better colourisation of a wide variety of images from large multi-class datasets. Further enhancements are achieved by applying feature normalisation techniques which are widely used in style transfer models. The capabilities of adversarial models in image colourisation are improved by adapting an Instance-Batch Normalisation (IBN) convolutional architecture [39] to conditional GANs. Finally, experimental results show that by boosting the performance of the adversarial framework, reduction of the desaturation effect can be achieved due to improvement of the discrimination of unreliable colours. Proposed techniques improve colourisation performance of baseline method using conditional GANs 3, in both perceptual realism and colourfulness. On the other hand, as shown in Chapter 4, the training of XCNET needs to promote the reliable transfer of reference colours to the target content towards obtaining a colour prediction faithful to the reference. The XCNET training strategy combines state-of-the-art exemplar-based metrics with fully automatic objectives (including adversarial loss), which promote dataset priors in the case of ambiguity. Finally, as mentioned in research question 1, both temporal consistency and high resolution scalability are derived from the cost-effective design of the video colourisation pipeline, achieved by combining efficient image-based colourisation techniques with generalised DVP framework to promote blind temporal consistency.

Research question 3: Can style transfer approaches be integrated into an end-to-end video colourisation pipeline in order to perform exemplar-based colourisation? If so, which are the best practices to efficiently select appropriate references?

Chapter 4 presents XCNET, a straightforward end-to-end solution for exemplar-based image colourisation which integrates attention modules that learn how to extract and transfer style features from the reference to the target in an unsupervised way during the colourisation process. In contrast to image analogy methods based on *PatchMatch*, attention does not need to constrain to a specific local search technique (even if it can act as a set of long-term deformable kernels) nor the similarity metric (e.g. correlation loss, cosine similarity) since the module learns it automatically. Experimental results demonstrate the efficiency of the proposed end-to-end solution, significantly reducing the runtime of previous exemplar-based works based on deep analogy computation [5]. Finally Chapter 5 proposes a pipeline for video colourisation, where the XCNET architecture can be easily integrated to perform exemplar-based predictions. Based on the method used in Chapter 4 to create training pairs of target-reference images, a retrieval system is integrated into the colourisation pipeline to assist the producer to search for meaningful references based on a similarity search on an external dataset.

Research question 4: Can chroma intra-prediction methods be improved by means of neural networks and be efficiently implemented into existing video coding standards?

Chapter 6 proposes a novel neural network for cross-component intra-prediction, introducing a new attention module capable of tracking the contribution of each neighbouring reference sample when computing the prediction of each chroma pixel. The proposed architecture is integrated into the VVC anchor (VTM-7.0), signalled as a new chroma intra-prediction mode working in parallel with traditional modes towards predicting the chroma component samples. Experimental results shows the effectiveness of the proposed architecture, reporting average BD-rate reductions of -0.21%, -1.90% and -1.81% for the Y, Cb and Cr components, respectively, with respect to the VVC anchor. Although improving coding performance, this NN-based method significantly increases the codec complexity, reporting encoder and decoder times of 212% and 2163%, respectively. The size-agnostic multi-model proposed in Chapter 7 (Scheme 1), further improves the coding performance with BD-rate reductions of -0.21%, -1.90% and -1.81% and manages to reduce the encoder and decoder times to 164% and 1302%, respectively. Moreover, the simplifications proposed in the same chapter manage to derive an inference model with around 90% fewer parameters and hence reduce the encoder and decoder times to 154% and 512%, respectively, for Scheme 3.

Research question 5: Can deep learning models be designed and/or simplified by means of explainability in order to obtain efficient and cost-effective implementations?

Chapter 7 focuses on complexity reduction and specifically tackles the model proposed in Chapter 6, with the aim to derive a set of simplified and cost-effective attention-based architectures for chroma intra-prediction. Understanding and distilling knowledge from the networks enables the implementation of less complex algorithms which achieve similar performance to the original models. The chapter introduces a novel framework for complexity reduction of the convolutional operations, responsible for most of the network's complexity. For instance, simplification of convolutions enables the development of an inference model with around 90% fewer parameters. Moreover, a simplified cross-component boundary branch is proposed under the rationale that a three-dimensional feature space can be considered as the space with minimum dimensionality that is still capable of representing colour information. Therefore, the chapter proposes the use of autoencoders to reduce the complexity of the cross-component boundary branch, by compacting the original space dimensional into a reduced 3-dimensional space. Finally, algorithmic insights are proposed to approximate the multi-model schemes in integer-precision arithmetic, which could lead to fast and hardware-aware implementations of complex operations such as softmax and Leaky ReLU activations.

8.2 Research contributions and proposed solutions

The contributions of this research are collected and summarised in the following list:

- Chapter 3: Fully automatic colourisation

  1. A GAN-based setting for fully automatic image colourisation, which adapts the IBN paradigm [39] to an encoder-decoder architecture. The proposed method enables generalisation of the content's style changes while encourages stabilisation during adversarial training.
  2. The use of Spectral Normalisation (SN) [40] for improving the generalisation of the adversarial colourisation and preventing training instability.
  3. The use of multi-scale discriminators to achieve an improved colour generation in small areas and local details and a boosted colourfulness.

- Chapter 4: Exemplar-based colourisation

  1. Introduction of XCNET, a fast-end-to-end architecture for exemplar-based image colourisation that improves existing methods while decreasing significantly the complexity and runtime.
  2. A multi-scale interpretation of the axial transformer for unsupervised style transfer and features analogy, which reduce the complexity of the attention operations.
  3. A multi-loss training strategy that combines a multi-scale adversarial loss with conventional style transfer and exemplar-based colourisation losses.

- Chapter 5: A pipeline for video colourisation

1. A framework for temporal stabilisation of frame-to-frame colourised videos with an optimised few-shot training strategy to learn scene-aware video priors.
2. A CNN architecture that is jointly trained to stabilise the input video and to cluster the input frames with the aim of learning scene-specific modes.
3. A clustering algorithm for scene segmentation to select meaningful frames and to generate pseudo-labels to supervise the scene-aware training.
4. An easily-deployable pipeline which integrates image-based colourisation methods from previous chapters to enable stable exemplar-based video colourisation.

- Chapter 6: Neural networks for chroma intra prediction

  1. A novel NN-based cross-component intra-prediction method, introducing a new attention module capable of tracking the contribution of each neighbouring reference sample when computing the prediction of each chroma pixel.
  2. Integration of the neural network into the latest VVC anchor, signalled as a new chroma intra-prediction mode working in parallel with traditional modes towards predicting the chroma component samples.

- Chapter 7: Multi-model architectures and simplifications

  1. A competitive block-independent attention-based multi-model and training methodology.
  2. A framework for complexity reduction of the convolutional operations.
  3. A simplified cross-component processing model using sparse auto-encoders.
  4. A fast and cost-effective attention-based multi-model with integer precision approximations.

Potential directions for future work are also identified in each of the chapters to further push the field towards more realistic outputs and to improve the applicability of the proposed methods in real-world challenges. Section 8.3 elaborates on those observations.

8.3 Recommendations and future work

This subsection compiles the main research directions introduced across the thesis, and potential opportunities for future research in these areas.

With regards to image and video colourisation:

- Improve quality of image colourisation to gain precision on local areas and different objects:
    - Most colourisation methods tend to struggle with complex scenes with a variety of objects and details, specially when they are trained with datasets with constrained amount of content such as Imagenet. Most methods tend to create colour leakage in the boundary of uncertain objects, or directly colourise such objects with averaged colours. A possible research line might tackle this problem by adding the instance segmentation task in the prediction loop, so a two-head encoder-decoder architecture could be jointly trained to solve the colourisation task while outputting instance segmentation maps of the input grayscale image, in which the segmentation task can guide towards more accurate and discriminative predictions.
    - Perceptual improvements could be achieved by adding a feature reconstruction loss to guide the training strategy. Such loss function could use a pre-trained VGG-16 [51] to match perceptual similarity between the real and generated images. Other alternatives include the use of FID [117] as a perception metric or a hybrid perception and PSRN-oriented training similar than the work in [16].
- Improve and generalise the XCNET architecture for exemplar-based colourisation:
    - The XCNET architecture can be improved by further tweaking the attention module. A cost-effective solution can be investigated to improve attention performance (keeping the axial approach or proposing a better methodology) without increasing the overall complexity. Moreover, further improvements could be obtained by improving the feature extraction backbone, e.g. investigate the effect of the input colour space and possibly use a different backbone for the grayscale target and colour references. Finally, better performance can be obtained by enabling multiple reference inputs.
    - Masked attention can be implemented to enhance discrimination of local objects by injecting instance segmentation masks into the attention modules.
    - The XCNET architecture could be generalised to other applications and domains aiming at unsupervised analogy computation between pairs of

inputs. Some examples include video frame interpolation, video compression or any other reference-based approach.

- Further optimisation of the blind temporal stabilisation framework:
    - Although the few-shot training strategy significantly reduces training runtime of the DVP baseline, the model still needs a few minute processing to yield the stabilised version. Model efficiency could be further improved by simplifying the network architecture or by using techniques such as pruning or weights quantisation. Moreover, a possible research line might consider fine-tuning priors from a generic model or checkpoint from previous scene to reduce the training overhead.
    - Finer tuning of colourisation could be achieved by improving the scene segmentation process in order to obtain more precise scene priors.
    - Consider processing video data instead of individual frames in order to generalise deep video prior paradigm into a unified and more efficient framework.

Finally, with regards to video compression and explainability:

- Improve attention-based architecture by enhancing performance while reducing model complexity:
    - The proposed attention-based methodology guides the fusion between the boundary and current luma block information. However, the predictive ability of CNNs is limited, and simply making the network deeper does not lead to more accurate predictions. Extending the side information and the methodology to measure the level of importance of each reference sample could boost the model performance towards more complex and uncertain situations and allow a reduction of overall complexity, specially in the convolutional branches.
    - Extension of the multi-model strategy for all VVC supported block sizes, including non-square blocks, could be investigated in order to ensure full usage of the proposed approach building on the promising results shown in the constrained test conditions.
- Further simplifications to approach acceptable codec complexity in future coding standards:
    - Investigate new linearisation methodologies to further reduce the processing steps of the inference model, such as concatenation, multiplication or

fusion of multiple branches. Exploring distillation techniques could also be a promising research line.

- An improved approach for integer approximations may enable the fusion of all proposed simplifications, leading to a fast and powerful multi-model.

8.4 Closing remarks

The broadcasting industry has been in constant evolution since the beginning of the digital era. New technologies have emerged to adapt to the increasing diversity of multimedia devices in which new forms of content are captured and reproduced. Artificial Intelligence is becoming a key player in the process providing improvements while automating many stages of the production and distribution workflows. However, the rapid advances in the field makes the shift from research to real world scenarios a complicated task. As seen in this thesis, research usually prioritises performance over efficiency, assuming powerful computational resources are available at the application side. However, this does not always happen, for instance, as shown in Chapter 7, fields such as video coding demand low complex models to be deployed in hardware-aware implementations.

This thesis targets practical applications for improving colour processing in content production and distribution scenarios. Efficiency is considered as a key aspect in the design of the proposed algorithms, and easily-deployable models are intended. Nevertheless, regarding content production research, further work needs to be done to upgrade current methods to real production requirements. For example, scalability to high-definition content, compatibility to editing video formats, variable colour spaces or support to high dynamic range. Finally, although this thesis targets colour processing technologies, some research findings can be generalised to other computer vision disciplines. An example is the attention mechanism used in XCNET to compute unsupervised analogies between pairs of inputs, which can be applied for instance to video frame interpolation, video compression or any other reference-based applications. Other examples are the proposed simplifications in Chapter 7, such as the linearisation of convolutional operations or integer precision approximations, applicable to other research scenarios which demand cost-effective implementations. Therefore, similar methodologies can extend colour processing to other AI-based video enhancement and distribution problems enabling new application scenarios in the digital broadcasting workflow and beyond.

Bibliography

[1] Trisha Lin, "Changes of journalistic practices and workflow in integrated television newsrooms: A multiple case study of early adopters in taiwan," *Journal of Communications Management*, 2014.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[3] Yuxin Wu and Kaiming He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[4] Anat Levin, Dani Lischinski, and Yair Weiss, "Colorization using optimization," in *ACM Trans. on Graphics*, 2004, vol. 23.

[5] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan, "Deep exemplar-based colorization," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–16, 2018.

[6] Marc Gorriz Blanch, Saverio Blasi, Alan Smeaton, Noel E. O'Connor, and Marta Mrak, "Chroma intra prediction with attention-based cnn architectures," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 783–787.

[7] International Telecommunication Union, "Artificial intelligence systems for programme production and exchange. report, itu-r bt.2447-0," 2019, April.

[8] Peter Powell, "New direction for the integrated newsroom," *TVB Europe*, vol. 7, no. 9, pp. 1–3, 1998.

[9] Leif Kramp and Wiebke Loosen, "The transformation of journalism: From changing newsroom cultures to a new communicative orientation?," in *Communicative Figurations*, pp. 205–239. Palgrave Macmillan, Cham, 2018.

[10] José Alberto García Avilés and Miguel Carvajal, "Integrated and cross-media newsroom convergence: Two models of multimedia news production—the

cases of novotecnica and la verdad multimedia in spain," *Convergence*, vol. 14, no. 2, pp. 221–239, 2008.

[11] Gary Waldman, *Introduction to light: The physics of light, vision, and color*, Courier Corporation, 2002.

[12] Thomas Young, "The bakerian lecture. on the theory of light and colours," in *Abstracts of the Papers Printed in the Philosophical Transactions of the Royal Society of London.* The Royal Society London, 1832, number 1, pp. 63–67.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. on Neural Networks*, vol. 8, no. 1, 1997.

[15] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[16] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[17] Srinivasa G. Narasimhan and Shree K. Nayar, "Contrast restoration of weather degraded images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 6, pp. 713–724, 2003.

[18] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] Robert Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, pp. 65–93. Elsevier, 1992.

[21] Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski, "Deep generative stochastic networks trainable by backprop," in *International Conference on Machine Learning.* PMLR, 2014, pp. 226–234.

[22] Danilo Rezende and Shakir Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538.

[23] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[25] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent, "A neural probabilistic language model," *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[26] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[27] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[28] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015.

[31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[32] Alex Graves, Greg Wayne, and Ivo Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[33] Minh-Thang Luong, Hieu Pham, and Christopher D Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[34] Jianpeng Cheng, Li Dong, and Mirella Lapata, "Long short-term memory-networks for machine reading," *arXiv preprint arXiv:1601.06733*, 2016.

[35] Marc Górriz Blanch, Marta Mrak, Alan F. Smeaton, and Noel E. O'Connor, "End-to-end conditional gan-based architectures for image colourisation," in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, 2019, pp. 1–6.

[36] Przemyslaw Lagodzinski and Bogdan Smolka, "Colorization of medical images," *Pattern Recognition Letters*, vol. 17, 1995.

[37] Anwaar-ul Haq, Iqbal Gondal, and M Murshed, "Automated multi-sensor color video fusion for nighttime video surveillance," in *The IEEE symposium on Computers and Communications*, 2010.

[38] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[39] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang, "Two at once: Enhancing learning and generalization capacities via IBN-Net," in *Proc. of the European Conf. on Computer Vision*, 2018.

[40] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida, "Spectral normalization for generative adversarial networks," *arXiv:1802.05957*, 2018.

[41] Wilson Markle and Brian Hunt, "Coloring a black and white signal using motion detection," July 5 1988, US Patent 4,755,870.

[42] Richard Zhang, Phillip Isola, and Alexei A Efros, "Colorful image colorization," in *European Conf. on Computer Vision*, 2016.

[43] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, "Improved techniques for training GANs," in *Advances in Neural Information Proc. Systems*, 2016.

[44] Christine Connolly and T Fleiss, "A study of efficiency and accuracy in the transformation from RGB to CIELAB color space," *IEEE Trans. on Image Proc.*, vol. 6, no. 7, 1997.

[45] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi, "Image colorization using generative adversarial networks," in *Int'l Conf. on Articulated Motion and Deformable Objects*, 2018.

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int'l Conf. on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[47] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.

[48] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv:1511.06434*, 2015.

[49] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv:1607.08022*, 2016.

[50] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[51] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[52] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conf. on Computer Vision*, 2016.

[53] Marc Gorriz, Laia Tarrés, Xavier Giro-i Nieto, and Marta Mrak, "Gan-based image colourisation with feature reconstruction loss," .

[54] Marc Gorriz Blanch, Issa Khalifeh, Noel E. Ox2019;Connor, and Marta Mrak, "Attention-based stylisation for exemplar image colourisation," in *2021 IEEE 23rd International Worksåhop on Multimedia Signal Processing (MMSP)*, 2021, pp. 1–6.

[55] Revital Ironi, Daniel Cohen-Or, and Dani Lischinski, "Colorization by example.," in *Rendering techniques*. Citeseer, 2005, pp. 201–210.

[56] Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005, pp. 351–354.

[57] Liron Yatziv and Guillermo Sapiro, "Fast image and video colorization using chrominance blending," *IEEE transactions on image processing*, vol. 15, no. 5, pp. 1120–1129, 2006.

[58] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng, "Manga colorization," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 1214–1220, 2006.

[59] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum, "Natural image colorization," in *Proceedings of the 18th Eurographics conference on Rendering Techniques*, 2007, pp. 309–320.

[60] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros, "Real-time user-guided image colorization with learned deep priors," *arXiv preprint arXiv:1705.02999*, 2017.

[61] Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo, "User-guided deep anime line art colorization with conditional adversarial networks," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1536–1544.

[62] Kun Xu, Yong Li, Tao Ju, Shi-Min Hu, and Tian-Qiang Liu, "Efficient affinity-based edit propagation using kd tree," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, pp. 1–6, 2009.

[63] Aurélie Bugeau, Vinh-Thong Ta, and Nicolas Papadakis, "Variational exemplar-based image colorization," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 298–307, 2013.

[64] Zhongyou Xu, Tingting Wang, Faming Fang, Yun Sheng, and Guixu Zhang, "Stylization-based architecture for fast deep exemplar colorization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9363–9372.

[65] Peng Lu, Jinbei Yu, Xujun Peng, Zhaoran Zhao, and Xiaojie Wang, "Gray2colornet: Transfer more colors from reference image," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3210–3218.

[66] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley, "Color transfer between images," *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001.

[67] Alex Yong-Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, Yu-Wing Tai, Siu-Yeung Cho, Ping Tan, and Stephen Lin, "Semantic colorization with internet images," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, pp. 1–8, 2011.

[68] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller, "Transferring color to greyscale images," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002, pp. 277–280.

[69] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang, "Local color transfer via probabilistic segmentation by expectation-maximization," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, 2005, vol. 1, pp. 747–754.

[70] François Pitié, Anil C Kokaram, and Rozenn Dahyot, "Automated colour grading using colour distribution transfer," *Computer Vision and Image Understanding*, vol. 107, no. 1-2, pp. 123–137, 2007.

[71] Yuji Morimoto, Yuichi Taguchi, and Takeshi Naemura, "Automatic colorization of grayscale images using multiple images on the web," in *SIGGRAPH 2009: Talks*, pp. 1–1. 2009.

[72] Raj Kumar Gupta, Alex Yong-Sang Chia, Deepu Rajan, Ee Sin Ng, and Huang Zhiyong, "Image colorization using similar images," in *Proceedings of the 20th ACM international conference on Multimedia*, 2012, pp. 369–378.

[73] Chufeng Xiao, Chu Han, Zhuming Zhang, Jing Qin, Tien-Tsin Wong, Guoqiang Han, and Shengfeng He, "Example-based colourization via dense encoding pyramids," in *Computer Graphics Forum*. Wiley Online Library, 2020, vol. 39, pp. 20–33.

[74] Bo Zhang, Mingming He, Jing Liao, Pedro V Sander, Lu Yuan, Amine Bermak, and Dong Chen, "Deep exemplar-based video colorization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8052–8061.

[75] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans, "Axial attention in multidimensional transformers," *arXiv preprint arXiv:1912.12180*, 2019.

[76] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich, "Learning representations for automatic colorization," in *European conference on computer vision*. Springer, 2016, pp. 577–593.

[77] Patricia Vitoria, Lara Raad, and Coloma Ballester, "Chromagan: adversarial picture colorization with semantic class distribution," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2445–2454.

[78] Sergio Guadarrama, Ryan Dahl, David Bieber, Mohammad Norouzi, Jonathon Shlens, and Kevin Murphy, "Pixcolor: Pixel recursive colorization," *arXiv preprint arXiv:1705.07208*, 2017.

[79] Manoj Kumar, Dirk Weissenborn, and Nal Kalchbrenner, "Colorization transformer," *arXiv preprint arXiv:2102.04432*, 2021.

[80] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng, "Intrinsic colorization," in *ACM SIGGRAPH Asia 2008 papers*, pp. 1–9. 2008.

[81] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[82] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 108–126.

[83] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[84] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang, "Visual attribute transfer through deep image analogy," *arXiv preprint arXiv:1705.01088*, 2017.

[85] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 24, 2009.

[86] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[87] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit, "A decomposable attention model for natural language inference," *arXiv preprint arXiv:1606.01933*, 2016.

[88] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[89] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[90] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.

[91] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi, "On the relationship between self-attention and convolutional layers," *arXiv preprint arXiv:1911.03584*, 2019.

[92] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens, "Stand-alone self-attention in vision models," *arXiv preprint arXiv:1906.05909*, 2019.

[93] Peter J Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*, pp. 492–518. Springer, 1992.

[94] Jan Puzicha, Thomas Hofmann, and Joachim M Buhmann, "Non-parametric similarity measures for unsupervised texture segmentation and image retrieval," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1997, pp. 267–272.

[95] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky, "Neural codes for image retrieval," in *European conference on computer vision*. Springer, 2014, pp. 584–599.

[96] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[97] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[98] Marta Mrak, “Ai gets creative,” in *Proceedings of the 1st International Workshop on AI for Smart TV Content Production, Access and Delivery*, New York, NY, USA, 2019, AI4TV ’19, p. 1–2, Association for Computing Machinery.

[99] Naofumi Akimoto, Akio Hayakawa, Andrew Shin, and Takuya Narihira, “Reference-based video colorization with spatiotemporal correspondence,” *arXiv preprint arXiv:2011.12528*, 2020.

[100] Nicolas Bonneel, James Tompkin, Kalyan Sunkavalli, Deqing Sun, Sylvain Paris, and Hanspeter Pfister, “Blind video temporal consistency,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 1–9, 2015.

[101] Chun-Han Yao, Chia-Yang Chang, and Shao-Yi Chien, “Occlusion-aware video temporal consistency,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 777–785.

[102] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang, “Learning blind video temporal consistency,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 170–185.

[103] Chenyang Lei, Yazhou Xing, and Qifeng Chen, “Blind video temporal consistency via deep video prior,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1083–1093, 2020.

[104] Chenyang Lei and Qifeng Chen, “Fully automatic video colorization with self-regularization and diversity,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3753–3761.

[105] Varun Jampani, Raghudeep Gadde, and Peter V Gehler, “Video propagation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 451–461.

[106] Sifei Liu, Guangyu Zhong, Shalini De Mello, Jinwei Gu, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz, “Switchable temporal propagation network,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 87–102.

[107] Li Fei-Fei, Rob Fergus, and Pietro Perona, “One-shot learning of object categories,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[108] Michael Fink, “Object classification from a single example utilizing class relevance metrics,” *Advances in neural information processing systems*, vol. 17, 2004.

[109] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni, “Generalizing from a few examples: A survey on few-shot learning,” *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[110] Chelsea Finn, Pieter Abbeel, and Sergey Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *International conference on machine learning.* PMLR, 2017, pp. 1126–1135.

[111] Sachin Ravi and Hugo Larochelle, “Optimization as a model for few-shot learning,” 2016.

[112] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi, “Learning feed-forward one-shot learners,” *Advances in neural information processing systems*, vol. 29, 2016.

[113] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales, “Learning to compare: Relation network for few-shot learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[114] Harrison Edwards and Amos Storkey, “Towards a neural statistician,” *arXiv preprint arXiv:1606.02185*, 2016.

[115] James MacQueen et al., “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.* Oakland, CA, USA, 1967, vol. 1, pp. 281–297.

[116] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid, “Actions in context,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2009, pp. 2929–2936.

[117] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *Advances in neural information processing systems*, vol. 30, 2017.

[118] B. Bross, J. Chen, and S. Liu, “Versatile Video Coding (VVC) draft 7,” Geneva, Switzerland, October 2019.

[119] Yue Li, Li Li, Zhu Li, Jianchao Yang, Ning Xu, Dong Liu, and Houqiang Li, "A hybrid neural network for chroma intra prediction," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1797–1801.

[120] J Pfaff, P Helle, D Maniry, S Kaltenstadler, B Stallenberger, P Merkle, M Siekmann, H Schwarz, D Marpe, and T Wiegand, "Intra prediction modes based on neural networks," *Doc. JVET-J0037-v2, Joint Video Exploration Team of ITU-T VCEG and ISO/IEC MPEG*, 2018.

[121] Kai Zhang, Jianle Chen, Li Zhang, Xiang Li, and Marta Karczewicz, "Enhanced cross-component linear model for chroma intra-prediction in video coding," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3983–3997, 2018.

[122] L. T. Nguyen, A. Khairat and D. Marpe, "Adaptive inter-plane prediction for RGB content," *Document JCTVC-M0230*, Incheon, April 2013.

[123] Mischa Siekmann, Ali Khairat, Tung Nguyen, Detlev Marpe, and Thomas Wiegand, "Extended cross-component prediction in hevc," *APSIPA transactions on signal and information processing*, vol. 6, 2017.

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

[125] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 114–125.

[126] S. Kim J. Chen, Y. Ye, "Algorithm description for versatile video coding and test model 7 (vtm 7)," *Document JVET-P2002*, Geneva, October 2019.

[127] J. Boyce, K. Suehring, X. Li, and V. Seregin, "JVET common test conditions and software reference configurations," *Document JVET-J1010*, Ljubljana, Slovenia, July 2018.

[128] F. Bossen, "On reporting combined YUV BD rates," *Document JVET-N0341*, Geneva, Switzerland, March 2019.

[129] Marc Górriz Blanch, Saverio Blasi, Alan F. Smeaton, Noel E. O'Connor, and Marta Mrak, "Attention-based neural networks for chroma intra prediction in video coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 366–377, 2021.

[130] Luka Murn, Marc Gorriz Blanch, Maria Santamaria, Fiona Rivera, and Marta Mrak, "Towards transparent application of machine learning in video processing," *arXiv preprint arXiv:2105.12700*, 2021.

[131] Luka Murn, Saverio Blasi, Alan F. Smeaton, Noel E. O'Connor, and Marta Mrak, "Interpreting cnn for low complexity learned sub-pixel motion compensation in video coding," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 798–802.

[132] Philipp Helle, Jonathan Pfaff, Michael Schäfer, Roman Rischke, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, "Intra picture prediction for video coding with neural networks," in *2019 Data Compression Conference (DCC)*. IEEE, 2019, pp. 448–457.

[133] Xin Zhao, Jianle Chen, Amir Said, Vadim Seregin, Hilmi E Egilmez, and Marta Karczewicz, "Nsst: Non-separable secondary transforms for next generation video coding," in *2016 Picture Coding Symposium (PCS)*. IEEE, 2016, pp. 1–5.

[134] Yihui He, Xiangyu Zhang, and Jian Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1389–1397.

[135] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu, "Discrimination-aware channel pruning for deep neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 875–886.

[136] Ting-Wu Chin, Ruizhou Ding, Cha Zhang, and Diana Marculescu, "Towards efficient model compression via learned global ranking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1518–1528.

[137] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713.

[138] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer, "Zeroq: A novel zero shot quantization framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13169–13178.

[139] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan, "Generative low-bitwidth data free quantization," *arXiv preprint arXiv:2003.03603*, 2020.

[140] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David, "Training deep neural networks with low precision multiplications," *arXiv preprint arXiv:1412.7024*, 2014.

[141] Johannes Ballé, Nick Johnston, and David Minnen, "Integer networks for data compression with latent-variable models," in *International Conference on Learning Representations*, 2018.

[142] Michael Schäfer, Björn Stallenberger, Jonathan Pfaff, Philipp Helle, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, "Efficient fixed-point implementation of matrix-based intra prediction," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3364–3368.

[143] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[144] Xue Geng, Jie Lin, Bin Zhao, Anmin Kong, Mohamed M Sabry Aly, and Vijay Chandrasekhar, "Hardware-aware softmax approximation for deep neural networks," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 107–122.

[145] Russell Davidson, "Reliable inference for the gini index," *Journal of econometrics*, vol. 150, no. 1, pp. 30–40, 2009.

[146] Chengyi Zou, Shuai Wan, Tiannan Ji, Marta Mrak, Marc Gorriz Blanch, and Luis Herranz, "Spatial information refinement for chroma intra prediction in video coding," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2021, pp. 1422–1427.