




**ADVERTIMENT.** L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  <https://creativecommons.org/licenses/?lang=ca>

**ADVERTENCIA.** El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <https://creativecommons.org/licenses/?lang=es>

**WARNING.** The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>



**Universitat Autònoma  
de Barcelona**

# On Considering Semantics for Multi-image Processing

A dissertation submitted by **Danna Xue** to the Universitat Autònoma de Barcelona in fulfilment of the degree of **Doctor of Philosophy** in the Departament de Ciències de la Computació.

Bellaterra, May 21, 2024

Director	<p><b>Dr. Luis Herranz</b> Dpt. Tecnología Electrónica y de las Comunicaciones Universidad Autónoma de Madrid</p> <p><b>Dr. Javier Vazquez-Corral</b> Dept. Ciències de la computació and Centre de Visió per Computador Universitat Autònoma de Barcelona</p> <p><b>Prof. Yanning Zhang</b> School of Computer Science Northwestern Polytechnical University</p>
Thesis committee	<p><b>Prof. Graham Finlayson</b> School of Computing Sciences, Colour and Imaging Lab University of East Anglia</p> <p><b>Dr. Felipe Lumbreras Ruiz</b> Dept. Ciències de la computació and Centre de Visió per Computador Universitat Autònoma de Barcelona</p> <p><b>Dr. Marcos Escudero-Viñolo</b> Dpt. Tecnología Electrónica y de las Comunicaciones Universidad Autónoma de Madrid</p> <p><b>Dr. Pablo Arias Martínez</b> Dpt. Tecnología de la informació i les Comunicacions Universitat Pompeu Fabra</p> <p><b>Dr. Ana Serrano Pacheu</b> Dpt. Informática e Ingeniería de Sistemas Universidad de Zaragoza</p>




---

This document was typeset by the author using  $\text{\LaTeX} 2_{\epsilon}$ .

The research described in this book was carried out at the Centre de Visió per Computador, Universitat Autònoma de Barcelona. Copyright © 2024 by **Danna Xue**. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the author.

ISBN: 978-84-126409-4-6

Printed by Ediciones Gráficas Rey, S.L.



# Acknowledgements

Years of doctoral study suddenly draw to a close, and I find myself at a loss for words. On an ordinary morning, facing unfinished code and unread literature, I do not know how to recall this long and seemingly fleeting journey. I can only express my thanks in the most clichéd way.

First, I want to thank my advisors, Dr. Javier Vazquez-Corral, Dr. Luis Herranz, and Prof. Yanning Zhang, for giving me the utmost respect and acceptance, and allowing me to explore the directions that interest me. They have also provided me with opportunities to see a richer world, enabling me to break free from my own limitations and engage with outstanding scholars, helping me find my own goals. Their attitude of serious research is something that I need to constantly remind myself of on my future path.

I am grateful to the people of York University, Canada. Two months of learning experience have been extremely beneficial to me. The collaboration with them has shown me how outstanding individuals work with excellence. And the lovely group always reminded me of the best time working with my friends at NPU.

Thank you to the CVC staff, Monste, Gigi, Joan, Laura, Andrea, Jorge, Nuria, Kevin, Mireia, Eva, *etc*, for their assistance; they have always patiently and quickly helped me resolve various issues.

Thanks to my parents, who have always supported my choices and provided me with the strongest backing. I know that even if I don't achieve anything, there are always people in this world who love and support me.

Thanks to my dearest friends, Yi, Kai, Yixiong, Fei, Chuanming, Yuyang, Yachan, Qingshan, Yaxing, Chenshen, Shiqi, Sanket, Pei, Shaolin, Cheng, Rui, Axi, Yaoqi, Anqi, Xing, and *etc*. All the moments we have shared on happy, sad, busy, or mundane days are the most precious treasures in my life bank. In those moments, I have experienced joy and heartbreak. But whether it is joy or sorrow, I believe it is a journey I should go through. Although I have lost some things, like youth, hair, and

---

carefreeness, I have also become stronger.

Lastly, I want to share a paragraph from a piece of Chinese ancient pros “*Record of a Visit to Baochan Mountain*”, which encouraged me from the start of my Ph.D. “*The ancients’ observations of the sky and earth, mountains and rivers, plants and trees, insects and fish, birds and beasts often yielded insights. This was because their quest for understanding was deep, leaving no aspect unexplored. When a place is easily accessible, it attracts many travelers; when it is remote and difficult to reach, few ventures there. The marvels and wonders of the world, the extraordinary sights, often lie in the distant and challenging places where few people tread. Therefore, only those with determination can reach them. However, determination alone is not enough; one also needs the strength to overcome obstacles. Even with determination and strength, if one lacks perseverance, one may wander into darkness and confusion without finding anything. If one possesses determination and strength but does not slack in their efforts, yet still fails to achieve their goal, they may be ridiculed by others, but they will harbor regrets within themselves. However, if one exhausts their determination and cannot reach their goal, they can do so without regret.*”

It is regrettable that during my doctoral studies, I didn’t explore those more obscure and winding alleys, but the future is still long, and I am willing to continue striving.

# Abstract

In multi-image processing, leveraging semantic information is essential for content-aware operations and ensuring consistency across images. However, this presents challenges in obtaining high-precision semantic data quickly, tailoring semantic information to different tasks, and maintaining consistency across processing results. This thesis addresses these challenges through several proposed approaches:

**Slimmable semantic segmentation:** We introduce a flexible framework for training semantic segmentation models with knowledge distillation, enabling quick adaptation between accuracy and efficiency trade-offs. To further improve the accuracy of the compact models, boundary supervision is introduced to obtain better object boundary details.

**Semantic integration in recoloring:** We explore the integration of semantic features into palette-based image recoloring to enhance color consistency across multiple images. Moreover, we propose to introduce color naming features in color harmonization. We demonstrate that the integration of semantics improves image color consistency and harmony, producing better perceptual visual effects.

**Temporal impact analysis:** We investigate the impact of temporal information on multi-image restoration quality, highlighting the perception-distortion tradeoff and the importance of alignment. We demonstrate that the perception-distortion tradeoff still exists when introducing temporal information, and misalignment worsens both perception and distortion. Our analysis provides a reference for designing multi-frame restoration algorithms and potential shooting strategies.

Each approach contributes to overcoming the challenges of leveraging semantic information in multi-image processing, aiming to enhance both efficiency and effectiveness in various image processing applications.

**Key words:** *deep learning, semantic segmentation, image recoloring, image restoration, multi-image processing, perception-distortion tradeoff*





# Resumen

En el procesamiento de múltiples imágenes, aprovechar la semántica es esencial para operaciones basadas en el contenido y garantizar la consistencia entre imágenes. Esto presenta desafíos en la obtención rápida de datos semánticos de alta precisión, adaptar la información semántica a diferentes tareas y mantener la consistencia en los resultados. Esta tesis aborda estos desafíos a través de varios enfoques:

**Segmentación semántica adaptable:** Introducimos un marco flexible para entrenar modelos de segmentación semántica con destilación de conocimientos, lo que permite una rápida adaptación entre los compromisos de precisión y eficiencia. Para mejorar aún más la precisión de los modelos compactos, se introduce supervisión de contornos para obtener mejores detalles de los límites de los objetos.

**Integración semántica en la recoloración:** Exploramos la integración de características semánticas en la recolorización de imágenes basado en paletas para mejorar la consistencia del color en múltiples imágenes. Además, proponemos introducir características de nombres de colores en la armonización del color. Demostramos que la integración de semántica mejora la consistencia y armonía del color de la imagen, produciendo mejores efectos visuales perceptuales.

**Análisis del impacto temporal:** Investigamos el impacto de la información temporal en la calidad de restauración de múltiples imágenes, destacando el compromiso entre percepción y distorsión y la importancia de la alineación. Demostramos que el compromiso entre percepción y distorsión todavía existe al introducir información temporal, y que la falta de alineación empeora tanto la percepción como la distorsión. Nuestro análisis proporciona una referencia para diseñar algoritmos de restauración de múltiples fotogramas y estrategias de filmación potenciales.

Cada enfoque contribuye a superar los desafíos de aprovechar la información semántica en el procesamiento de múltiples imágenes, para mejorar tanto la eficiencia como la efectividad en diversas aplicaciones de procesamiento de imágenes.

**Palabras clave:** *aprendizaje profundo, segmentación semántica, recoloración de imágenes, restauración de imágenes, procesamiento de múltiples imágenes, compromiso percepción-distorsión*



# Resum

En el processament de múltiples imatges, l'aprofitament de la informació semàntica és essencial per a operacions conscients del contingut i per assegurar la consistència entre les imatges. Tanmateix, això presenta desafiaments en obtenir dades semàntiques d'alta precisió ràpidament, adaptar la informació semàntica a diferents tasques i mantenir la consistència en els resultats del processament. Aquesta tesi aborda aquests desafiaments mitjançant diversos enfocaments proposats:

**Segmentació semàntica adaptable:** Introduïm un marc flexible per a l'entrenament de models de segmentació semàntica amb destil·lació de coneixements, permetent una adaptació ràpida entre els compromisos de precisió i eficiència. Per millorar encara més la precisió dels models compactes, s'introdueix supervisió de contorns per obtenir millors detalls dels límits dels objectes.

**Integració semàntica en la recolorització:** Explorem la integració de característiques semàntiques en la recolorització múltiples imatges. A més, proposem introduir característiques de noms de colors en l'harmonització del color. Demostrem que la integració de la semàntica millora la consistència i harmonia del color de la imatge, produint millors efectes visuals perceptius.

**Anàlisi de l'impacte temporal:** Investiguem l'impacte de la informació temporal en la qualitat de restauració de múltiples imatges, destacant el compromís entre percepció i distorsió i la importància de l'alineació. Demostrem que el compromís entre percepció i distorsió encara existeix en introduir informació temporal, i que l'alineació empitjora tant la percepció com la distorsió. La nostra anàlisi proporciona una referència per dissenyar algoritmes de restauració de múltiples fotogrames i estratègies de filmació potencials.

Cada enfocament contribueix a superar els desafiaments de l'aprofitament de la informació semàntica en el processament de múltiples imatges, amb l'objectiu de millorar tant l'eficiència com l'eficàcia en diverses aplicacions.

**Paraules clau:** *aprenentatge profund, segmentació semàntica, recolorit d'imatges, restauració d'imatges, processament de múltiples imatges, compromís entre percepció i distorsió*



# Contents

<b>Abstract</b>	<b>iii</b>
<b>List of figures</b>	<b>xv</b>
<b>List of tables</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Semantic segmentation . . . . .	2
1.1.1 Limitations . . . . .	3
1.1.2 Objectives and approach . . . . .	4
1.2 Semantics based multi-image processing . . . . .	5
1.2.1 Limitations . . . . .	5
1.2.2 Objectives and approach . . . . .	6
1.3 Multi-image restoration . . . . .	7
1.3.1 Limitations . . . . .	7
1.3.2 Objectives and approach . . . . .	8
1.4 Goals and Outline . . . . .	9
<b>2 Slimmable semantic segmentation with boundary supervision</b>	<b>11</b>
2.1 Introduction . . . . .	11

## Contents

---

2.2	Related work . . . . .	13
2.2.1	Generic semantic segmentation . . . . .	13
2.2.2	Efficient semantic segmentation . . . . .	14
2.2.3	Dynamic neural networks . . . . .	15
2.3	Methodology . . . . .	15
2.3.1	Slimmable segmentation framework . . . . .	15
2.3.2	Stepwise downward distillation . . . . .	17
2.3.3	Semantic boundary guided loss . . . . .	18
2.4	Experiments . . . . .	20
2.4.1	Benchmarks and evaluation metrics . . . . .	20
2.4.2	Implementation details . . . . .	21
2.4.3	Ablation study . . . . .	22
2.4.4	Comparisons with real-time models . . . . .	30
2.5	Concluding remarks . . . . .	35
<b>3</b>	<b>Integrating high-level features for consistent palette-based multi-image recoloring</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Related work . . . . .	39
3.2.1	Photo-collection editing . . . . .	39
3.2.2	Palette-based image recoloring . . . . .	40
3.2.3	Color naming . . . . .	40
3.2.4	Saliency-aware image editing . . . . .	41
3.3	Methodology . . . . .	41

---

3.3.1	Multi-image recoloring framework . . . . .	42
3.3.2	White-balance correction module . . . . .	43
3.3.3	Saliency-guided palette grouping module . . . . .	44
3.3.4	Color-naming association module . . . . .	46
3.4	Experiments . . . . .	48
3.4.1	Experimental setting . . . . .	48
3.4.2	Qualitative results . . . . .	48
3.4.3	User study . . . . .	55
3.5	Applications . . . . .	58
3.5.1	Interactive multi-image recoloring . . . . .	58
3.5.2	Example of brochure design . . . . .	58
3.6	Concluding remarks . . . . .	60
<b>4</b>	<b>Palette-based color harmonization via color naming</b>	<b>61</b>
4.1	Introduction . . . . .	61
4.2	Related work . . . . .	63
4.2.1	Color harmonization . . . . .	63
4.2.2	Color naming . . . . .	63
4.2.3	Image enhancement . . . . .	64
4.3	Methodology . . . . .	64
4.3.1	Color prototypes generation with color naming . . . . .	65
4.3.2	Color palette extraction . . . . .	66
4.3.3	Color matching . . . . .	66
4.3.4	Palette-based image recoloring . . . . .	67

4.4	Experiments . . . . .	67
4.4.1	Experimental setup . . . . .	67
4.4.2	Quantitative results . . . . .	69
4.4.3	Ablation study . . . . .	69
4.4.4	Speed . . . . .	74
4.4.5	Qualitative results . . . . .	74
4.4.6	Application on multi-image recoloring . . . . .	76
4.4.7	Limitations . . . . .	76
4.5	Concluding remarks . . . . .	79
<b>5</b>	<b>Burst perception-distortion tradeoff: analysis and evaluation</b>	<b>81</b>
5.1	Introduction . . . . .	81
5.2	Related work . . . . .	83
5.2.1	Burst image restoration . . . . .	83
5.2.2	The Perception Distortion (P-D) tradeoff . . . . .	84
5.2.3	Learning-based frame selection for burst restoration . . . . .	84
5.3	Preliminaries . . . . .	85
5.3.1	The perception distortion tradeoff . . . . .	85
5.3.2	Burst image restoration . . . . .	85
5.4	Burst perception distortion tradeoff . . . . .	86
5.4.1	Area of the unattainable region . . . . .	87
5.4.2	Toy examples . . . . .	87
5.5	Experiments . . . . .	91
5.5.1	Experimental setting . . . . .	91



5.5.2	Perfectly aligned bursts . . . . .	93
5.5.3	Misaligned bursts . . . . .	94
5.6	Concluding remarks . . . . .	95
<b>6</b>	<b>Conclusion and future work</b>	<b>97</b>
6.1	Conclusions . . . . .	97
6.2	Future direction . . . . .	98
	<b>Publications</b>	<b>101</b>
	<b>Bibliography</b>	<b>126</b>



# List of Figures

2.1	Overview of our slimmable semantic segmentation framework. . . .	16
2.2	Difference maps between submodels. . . . .	18
2.3	FLOPs-mIoU spectrum of globally and partially slimmable networks on Cityscapes <i>val</i> . . . . .	24
2.4	Comparison of the distribution of error pixels. . . . .	27
2.5	Visual comparison of our boundary supervision on Cityscapes <i>val</i> . . .	28
2.6	Visual comparison of features on Cityscapes <i>val</i> . . . . .	33
2.7	Visual comparison of our boundary supervision on Cityscapes <i>val</i> . . .	34
3.1	Our multi-image color consistency framework. . . . .	38
3.2	Our image collection recoloring framework. . . . .	43
3.3	Saliency-based source palette grouping. . . . .	45
3.4	Color-naming association. . . . .	47
3.5	Results after applying the white-balance correction module. . . . .	49
3.6	Results of introducing saliency-guided palette grouping. . . . .	50
3.7	Example for our two approaches for saliency recoloring. . . . .	51
3.8	Results of adding the semantics-guided module on Cityscapes. . . . .	53
3.9	Results of adding the semantics-guided module on Viper. . . . .	54

## List of Figures

---

3.10 Results of adding the color-naming association module. . . . .	55
3.11 Results for the combination of all our modules. . . . .	56
3.12 Results of the psycho-physical experiment. . . . .	57
3.13 Our GUI interface. . . . .	59
3.14 Application on brochure design. . . . .	59
4.1 Comparison between color harmonization and our approach. . . . .	62
4.2 Our palette-based color modification framework. . . . .	65
4.3 Comparisons between applying different color distance measures. . . . .	72
4.4 The distribution of the amount of colors for each color name. . . . .	73
4.5 The probability distributions of each color name. . . . .	73
4.6 Comparison of images against other methods on FiveK dataset. . . . .	75
4.7 Comparison of images against other methods on PPR10K dataset. . . . .	77
4.8 An example of application on multi-image recoloring. . . . .	78
4.9 Examples of failure cases. . . . .	78
5.1 Single Image Restoration versus Burst Restoration. . . . .	82
5.2 The Area of the Unattainable Region (AUR). . . . .	87
5.3 Perception distortion curve of burst image. . . . .	90
5.4 P-D curves of perfectly aligned bursts. . . . .	93
5.5 P-D curves of misaligned bursts. . . . .	95
5.6 Results of burst with different types of displacement and alignment error. . . . .	95

# List of Tables

2.1	The FLOPs and number of parameters of different semantic segmentation networks. . . . .	12
2.2	Comparison of independent and slimmable models on Cityscapes <i>val</i> . . . . .	23
2.3	Comparison between globally and partially slimmable models on Cityscapes <i>val</i> . . . . .	25
2.4	Ablation of knowledge distillation with different loss function. . . . .	26
2.5	Ablation of different knowledge distillation strategies. . . . .	26
2.6	mIoUs of slimmable models trained with different boundary detection groundtruth. . . . .	28
2.7	mIoUs of slimmable models trained with different low-level features as the input of boundary detection head. . . . .	29
2.8	Architectures of ResNet50. . . . .	30
2.9	Comparison with state-of-the-art on Cityscapes <i>val</i> . . . . .	31
2.10	Comparison with state-of-the-art on CamVid <i>test</i> . . . . .	32
4.1	Comparison of image quality and harmony score on the camera raw version and DPE version of FiveK. . . . .	70
4.2	Comparison of image quality and harmony score on PPR10K and Kodak. . . . .	70
4.3	Ablation of the number of prototype colors. . . . .	71
4.4	Ablation of color similarity measures. . . . .	71

## List of Tables

---

4.5 Ablation of different color name on FiveK (Camera Raw). . . . .	72
4.6 Ablation of color searching space on FiveK (Camera Raw). . . . .	74
4.7 Average run time of our method. . . . .	74
5.1 Experimental settings for degraded burst images. . . . .	92

# 1 Introduction

When presented with an image, the human brain quickly recognizes familiar objects in a mere 100 milliseconds. This rapid comprehension stems from a blend of swift visual processing and cognitive analysis. As visual data reaches our eyes, it is transmitted to the brain's "Vision Center" located in the occipital cortex, where it is interpreted into electric signals, forming a visual map. We interpret images effortlessly and subconsciously, grasping their meanings without conscious effort. Visual symbols and icons serve as useful tools for creators to convey messages and meanings to audiences. However, beyond symbols, the interplay of complex imagery and visual elements within an image, *i.e.* context, also plays a crucial role. Familiar subjects or contexts facilitate rapid recognition and comprehension, while less familiar elements may require deliberate interpretation. Our experiences and acquired knowledge further inform our understanding, like building a mental visual dictionary, such as recognizing familiar faces. Through processes encompassing perception, processing, interpretation, understanding, and extrapolation, our brains transform visual stimuli into semantically meaningful information, shaping our thoughts and actions. In essence, decoding an image involves a multifaceted interplay of rapid visual processing and cognitive interpretation. One of the essential objectives of computer vision is to equip machines with similar processing capabilities, enabling them to quickly and accurately decipher semantic information from images.

***What are semantics?*** Semantics is the study of the meaning of words, constructions, and utterances, according to the definition from the book "*Foundations of Statistical Natural Language Processing*" [119]. In computer vision, semantics denotes the comprehension and interpretation of the content embedded within visual data, such as images or videos, at a higher conceptual level. This involves extracting significant information about objects, scenes, and relationships portrayed within the visual data.

***Why are semantics so important?*** The connection between the image and its semantics enables machines to comprehend the content of images in a way that is closer to human understanding. Moreover, by discerning the relationships

between different entities depicted in an image, deeper semantic insights beyond mere recognition can be derived. Semantic information about image content finds applications in various domains, including content-driven image editing tasks that require high-level decision planning, such as robotics, autonomous driving, and human-computer interaction.

***How to obtain semantic information from an image?*** Obtaining semantic information from an image typically involves using computer vision techniques and statistical methods. Some common approaches involve image classification, object detection, semantic segmentation, or instance segmentation. These methods can obtain semantic information at different levels of granularity, with segmentation-related tasks capable of achieving pixel-level semantic delineation, providing the foundation for precise image editing.

***What if there is more than one image?*** The situation with multiple images can be divided into two types: temporally related images, such as rapid burst shots or video sequences, wherein the frames capture a fixed scene with substantial overlapping content. Here, motion typically follows a discernible pattern, and inter-frame relationships can be established through techniques like optical flow for motion estimation. The other scenario is when no temporal relationship exists between images, but there might be some similarities, common objects or shared visual attributes like colors and shapes. For such images, inter-image relationships can only be established based on these shared characteristics.

This thesis endeavors to establish a comprehensive pipeline encompassing semantic extraction and utilization. Corresponding methodologies are devised to address challenges encountered at each stage of this pipeline, thereby fostering intuitive, rapid, and human-perceptive image processing. Furthermore, strategies for extracting and leveraging semantic associations across multiple images are discussed, enhancing result consistency amid the requirement to process diverse image datasets. Next, we will introduce the key aspects of the pipeline one by one, including the current studies and their limitations, and the solutions we propose in this thesis.

### 1.1 Semantic segmentation

Semantic segmentation is a computer vision task that involves assigning semantic labels to each pixel in an image, thus dividing the image into meaningful segments or regions. Unlike image classification, which classifies the entire image into a single category, semantic segmentation aims to provide a detailed understanding of the scene by labeling each pixel with the corresponding object or region it belongs to. Traditional methods distinguish regions by finding the commonality of pixels in dif-



ferent feature spaces [83, 155, 216]. In contrast, deep learning-based methods learn features in high-dimensional feature spaces. After the proposal of Fully Convolutional Network (FCN) [115], deep learning completely changed the development of segmentation. Since then, the performance of semantic segmentation on the Pascal VOC benchmark [40] has almost doubled (89% mIOU) [18, 22]. However, most existing deep learning-based segmentation models are still limited to a relatively small number of classes and scenes. These models rely heavily on dense annotations and cumbersome deep neural networks. In recent years, thanks to the development of large language models (LLMs) [144], there has been an explosion in the development of open-vocabulary [49, 97, 143, 183, 192, 193, 217], universal [26, 70, 87] segmentation models. This trend has also to some extent altered the paradigm of deep-learning-based semantic segmentation.

### 1.1.1 Limitations

**High computational complexity.** High-resolution images necessitate longer training times and increased memory consumption, posing challenges for edge devices. Real-time applications, such as autonomous driving or interactive image editing, require efficient algorithms capable of performing segmentation swiftly. Current research primarily focuses on enhancing semantic segmentation efficiency by developing compact backbone architectures [47, 88, 89, 137, 148, 165, 206, 221], implementing effective model compression techniques [23, 90, 94, 101, 141, 220], or leveraging reliable context and boundary information [23, 90, 94, 101, 141, 220]. However, these approaches typically accelerate inference using fixed network structures, which do not accommodate the varying resources across different devices. Even within a single device, the availability of hardware resources can fluctuate over time. To achieve an optimal accuracy-efficiency tradeoff, we could switch between models of different sizes. A straightforward method is to train multiple independent models with varied structures and parameters, then load the appropriate model during inference. Nevertheless, this approach demands extensive training time and substantial memory for storage.

**Boundary ambiguity.** Due to inherent pixel ambiguity at object boundaries, achieving fine distinctions poses challenges to semantic segmentation models. Additionally, segmenting small objects with ambiguous details and rare features is inherently challenging. Early methods improved edge segmentation through structured modeling [8, 9, 21, 78], which rely on Conditional Random Fields [83] as a post-processing module to improve the semantic boundaries, hindering end-to-end optimization. Many methods enhance edge details by fusing low-level features into high-level features, such as U-Net [150] and Feature Pyramid Network (FPN) [103], which merge high and low-level features through skip connections. This enables the

recovery of clearer object contours by incorporating higher-resolution low-level texture and color features on top of obtaining semantic information and the overall object position contours from the image. Some other approaches add constraints to edge regions during model training [1, 80, 92, 213], guiding the network to better learn edge region features. This includes constraints such as the edge detection loss functions [1, 92, 213] or resampling of edge pixels [80]. Cheng *et al.* [27] refine the segmentation results in a cascaded fashion, with downsampled low-resolution images first and with cropped high-resolution images to refine and correct local boundaries progressively from coarse to fine. The common issue of these methods is that they all introduce additional computational overhead to some extent.

### 1.1.2 Objectives and approach

Our goal is to design a simpler and more effective image semantic segmentation method, effectively improving the computational efficiency and processing accuracy of current deep learning models, and making it suitable for various hardware devices and requirements of different tasks, either to emphasize more accurate details or high-speed inference.

**Slimmable semantic segmentation with boundary supervision.** Accurate semantic segmentation models typically require significant computational resources, inhibiting their use in practical applications. Recent works rely on well-crafted lightweight models to achieve fast inference. However, these models cannot flexibly adapt to varying accuracy and efficiency requirements. To solve the problems in current semantic segmentation approaches, in Chapter 2, we propose a simple but effective slimmable semantic segmentation method, which can be executed at different capacities during inference depending on the desired accuracy-efficiency tradeoff. More specifically, we employ parametrized channel slimming by stepwise downward knowledge distillation during training. Motivated by the observation that the differences between segmentation results of each submodel are mainly near the semantic borders, we introduce an additional boundary guided semantic segmentation loss to further improve the performance of each submodel. We show that our proposed SlimSeg with various mainstream networks can produce flexible models that provide dynamic adjustment of computational cost and better performance than independent models. Extensive experiments on semantic segmentation benchmarks, Cityscapes and CamVid, demonstrate the generalization ability of our framework.

## 1.2 Semantics based multi-image processing

Semantics plays a critical role in improving the accuracy and contextual awareness of various computer vision applications, such as image reconstruction [181], image editing [116], and image generation [139]. In these applications, semantics serve a dual function in supporting different tasks. First, objects with distinct semantics often exhibit unique features, including characteristics such as color, texture, and shape. These distinctive features provide the algorithms with prior information for the purposeful recovery and modification of image details. Second, semantics aid in precise localization to their respective regions, allowing detailed processing tailored to the identified semantics. For instance, through semantic segmentation, semantics enable accurate selection and isolation of specific objects or regions within an image. This facilitates selective editing, where modifications can be applied solely to certain semantic classes, enhancing the precision and accuracy of the editing process. Integrating semantics into multiple image processing leads to a more comprehensive and context-aware understanding of visual data. It enables more sophisticated decision-making processes and supports the development of advanced applications.

### 1.2.1 Limitations

**Rely on accurate semantics.** The success of semantic-based image processing is heavily based on an accurate semantic segmentation mask. If the segmentation maps have misclassified categories or rough boundaries, subsequent editing tasks may produce undesirable results. The image processing approaches need to be robust to subtle variations in semantic content to avoid artifacts or distortions. We categorize existing semantic-based processing approaches into two types according to how they integrate semantics: implicit and explicit. A majority of implicit approaches treat semantics as the condition of a feature modulation module [91, 100, 116, 139, 174, 181, 190], such as the spatially-adaptive normalization [139]. This approach implicitly utilizes semantic segmentation maps and adopts soft logits of each pixel as network input to mitigate the adverse effects of misclassified semantic categories to some extent. Explicit approaches, on the other hand, directly process different regions of the image separately based on semantic segmentation, followed by fusion. Even with accurate segmentation results as guidance, separate processing of different parts can result in abrupt boundaries and inconsistencies between objects and their surroundings. Additional fusion modules are introduced to optimize edge regions [62, 116].

**Require task-specific semantics.** Due to the broad concepts and object ranges covered by semantics, different tasks often require different categories and forms of

semantic guidance. For instance, medical imaging [146], autonomous driving [91], and facial editing [149] all require different semantic categories. Semantic segmentation networks need to be designed and trained specifically for different tasks. Although recent open-vocabulary segmentation models [49, 97, 143, 183, 192, 193, 217] cover a wide variety of scenes and semantic categories, fine-tuning is generally required for specific applications [172, 189]. The forms of semantic guidance are also diverse, with text and semantic segmentation maps representing semantic forms from coarse to fine. In addition to object categories, different colors [167, 212], shapes [57, 76], and movements [20, 156] also carry specific semantics. The wide range of semantic forms and content necessitates the selection of appropriate semantic guidance based on task requirements in specific applications.

**Inconsistent and low-fidelity editing.** Through semantic similarity, connections can be established between different images, aiding algorithms in achieving consistent image processing. For a set of images without temporal relationships, semantic correspondence [51, 54, 59] between different instances of similar object categories can be established. However, due to the intra-class semantic differences, *e.g.* the appearance and shape variations of objects with the same semantics in different images, these correspondences often provide only rough matching, resulting in noticeable artifacts in practical applications. Additionally, current semantic-based image processing methods mainly focus on editing the content of images [104, 116], effectively utilizing semantic information. These methods are generally based on generative models such as Generative Adversarial Networks (GANs) [104, 116] or diffusion models [77], where the optimization objective is to generate images that follow a distribution as close as possible to the general distribution of natural images. However, in tasks such as image enhancement or restoration, we need to ensure fidelity to the original image while altering its appearance, *i.e.*, consistency with the content of the original image.

### 1.2.2 Objectives and approach

We explore the application of semantic information to the problem of image recoloring. We introduce high-level semantic features that include object category and color-naming information. We design task-specific modules to address the issues, including inaccurate semantic segmentation boundaries, perceptually-drastic and inharmonious changes, thereby obtaining more consistent and harmonious colors in the images.

**Integrating high-Level features for consistent multi-image recoloring.** Achieving visually consistent colors across multiple images is important when images are used in photo albums, websites, and brochures. Unfortunately, only a handful of methods address multi-image color consistency compared to one-to-one color

transfer techniques. Furthermore, existing methods do not incorporate high-level features that can assist graphic designers in their work. To address these limitations, in Chapter 3, we introduce a framework that builds upon a previous palette-based color consistency method and incorporates three semantic-related features: white balance, saliency, semantics, and color naming. We show how these features overcome the limitations of the prior multi-consistency workflow and showcase the user-friendly nature of our framework.

**Palette-based color harmonization via color naming.** Color harmony refers to combinations of colors that look pleasing together. We present a novel strategy to harmonize an image’s colors using color-palette manipulation and color naming. Palette-based color manipulation is a method that extracts a small number of colors to represent the image. Modifying the palette colors modifies the color appearance of the image. A color-naming model is a mechanism to categorize colors into a fixed number of basic color terms. Working from a color-naming model, in Chapter 4, we derive a set of *prototype colors* and demonstrate that mapping an image’s extracted color palette to the nearest prototype colors effectively harmonizes the image’s colors. This straightforward approach yields visually compelling, outperforming more color harmony complex methods.

### 1.3 Multi-image restoration

With the widespread use of mobile phone cameras and the growing popularity of related applications, the processing of videos and burst photos has gained significant attention. Multiple frames offer additional information, greatly enhancing image quality for tasks like video restoration and nighttime image enhancement. However, video and burst processing are more complex than single-image processing due to the temporal dimension of the data. Multiple frames provide several samples of the same scene, helping to reduce image noise and recover image details. Nonetheless, factors such as perspective changes during shooting, inevitable camera shake in handheld devices, and object motion within the scene often cause displacement between frames. If these movements are not corrected by aligning each frame, the final processed result may exhibit artifacts like blurring and ghosting. Therefore, a crucial challenge in time-series multi-frame image processing is how to effectively align multiple frames.

#### 1.3.1 Limitations

**Require precise alignments.** Precise flow estimation is always challenging. Early video restoration methods typically involve two steps [106, 107]: first, estimating

motion parameters through registration between frames, and then performing restoration based on registered frames. Such approaches have high requirements on the accuracy of flow estimation. For videos or burst captures taken over a short period, inter-frame motion estimation is generally based on optical flow estimation techniques [109, 157, 162, 175]. Xue *et al.* [198] reveal that standard optical flow might not be the optimal motion representation for video restoration, and propose a task-oriented flow (TOFlow) representation, utilizing an end-to-end trainable convolutional network that simultaneously performs motion analysis and video processing. DUF [73], TDAN [164], and EDVR [180] also circumvent this challenge through implicit motion compensation, surpassing flow-based methods. EDVR, for instance, performs implicit alignment using a pyramid and cascading architecture to handle large motions, while TDAN and EDVR introduce deformable convolutions [34] for alignment at the feature level. However, these implicit alignment modules leverage the strong expressive power of complex networks. Along with their benefits, they also pose challenges such as large model parameter sizes and low computational efficiency, especially when dealing with high-resolution images.

**Different optimizing targets.** Image restoration aims to eliminate various degradations that adversely affect image quality, such as noise and blur, aiming at obtaining a restored image as close as possible to the ground truth image. Traditionally, being close to the ground truth meant having little distortion as measured for example in dBs (PSNR). In recent years, generative models [24, 182] have played a significant role in image restoration tasks. These models can generate images similar to real images without a specific target image, making them visually natural and realistic. When using generative models for image restoration, besides optimizing image fidelity, better perceptual quality can also be achieved. Blau *et al.* [13] demonstrate there exists a tradeoff between distortion and perceptual quality in image restoration tasks, implying that it is not possible to simultaneously improve both aspects. This perception-distortion tradeoff has been confirmed in single-frame image restoration tasks, but it is unclear how the introduction of the temporal dimension affects both aspects of image quality in restoration results.

### 1.3.2 Objectives and approach

Our research focuses on the reconstruction of sequential multi-frame images, aiming to explore both theoretical analysis and practical applications:

**Burst perception-distortion tradeoff: analysis and evaluation.** Burst image restoration attempts to effectively utilize the complementary cues appearing in sequential images to produce a high-quality image. Most current methods use all the available images to obtain the reconstructed image. However, using more images for burst restoration is not always the best option regarding reconstruction

quality and efficiency, as the images acquired by handheld imaging devices suffer from degradation and misalignment caused by the camera noise and shake. We extend the perception-distortion tradeoff theory by introducing multiple-image information. We propose the area of the unattainable region as a new metric for perception-distortion tradeoff evaluation and comparison. Based on this metric, we analyze the performance of burst restoration from the perspective of the perception-distortion tradeoff under both aligned bursts and misaligned bursts situations. Our analysis reveals the importance of inter-frame alignment for burst restoration and shows that the optimal burst length for the restoration model depends both on the degree of degradation and misalignment.

## 1.4 Goals and Outline

The primary objective of this thesis is to contribute to research by extracting and leveraging semantics in low-level vision tasks, including image recoloring, and image restoration. We not only apply semantics to enhance the performance of optimizing single-frame image processing but also investigate how to establish connections between images based on semantic relevance in multi-frame scenarios, achieving consistent appearance in editing effects. We consider situations involving burst, video, and unrelated temporal images with only semantic similarity. Through these investigations, we validate the significance of semantic information in various low-level computer vision tasks. In particular, our contributions are:

- In Chapter 2, We investigate a potential solution to “*How to extract accurate semantics more flexibly and efficiently?*”. We investigate a flexible semantic segmentation model architecture that can be adjusted according to task requirements, switching between various accuracy and processing efficiency. Each model comprises multiple sub-models of varying sizes, which are trained simultaneously and share parameters but are capable of independent inference. The unique structure of the model allows for knowledge distillation from larger to smaller sub-models during training, further enhancing the segmentation accuracy of the smaller models. Our results highlight that for CNN architectures, certain channels focus more on low-frequency information, such as image content and object outlines, while others prioritize high-frequency information. Moreover, in dense prediction tasks, the scale of the image decoder should match that of the encoder to avoid adverse effects on processing accuracy and efficiency.
- In Chapter 3 and Chapter 4, we explore solutions to “*How to utilize semantics for obtaining consistent and harmonious colors in images?*”. We believe that

semantically related high-level features have a positive impact on image processing, which we confirm in our experiments. In the two color-related tasks, we introduce features such as saliency, object categories, and color names. Instead of independently processing images based on different semantic regions, we integrate semantics into a palette-based recoloring framework and manipulate the colors in the palette according to the semantic features. This allows for differential treatment of colors based on semantics, ensuring the smooth appearance of the final recolored results. Moreover, we also find that perceptual color names not only contribute to achieving more consistent image colors but can also be applied to image harmonization. By mapping the image palette to a set of prototype colors selected based on color names, we obtain more harmonious image colors without artifacts. We show that introducing high-level semantic information helps achieve more perceptually natural image colors.

- In Chapter 5, we find a possible solution for “*How temporal information affect multi-frame restoration?*”. Multi-frame images introduce more information for recovering image details lost due to imaging system and noise limitations, while the motion between frames can also introduce artifacts such as blurring in the restored images. Through a series of experiments, we analyze and demonstrate the perception-distortion tradeoff still exists when introducing temporal information, and misalignment will worsen both perception and distortion. In addition, our analysis provides a reference to the design of multi-frame restoration algorithms and the potential shooting strategy. Our results show that longer bursts (*i.e.* more images) do not always lead to higher restoration quality, since misalignment will make the restoration result worse with more frames. Thus, the key to multi-frame restoration lies in the inter-frame alignment method. Furthermore, bursts provide a suitable starting point to study more complicated sequences such as videos, and thus, our theory, analysis, and evaluation method can also be extended to more general video restoration scenarios.

Finally, in Chapter 6, we draw the global conclusions arising from the entire Ph.D. work and prospect future work.



## 2 Slimmable semantic segmentation with boundary supervision\*

### 2.1 Introduction

Semantic segmentation predicts the semantic category corresponding to each pixel in an image. Various applications have benefited from advances towards more accurate results, such as autonomous driving [23, 47, 53, 88–90, 93, 101, 137, 205, 206, 219, 220], image synthesis and manipulation [139, 178], and medical imaging [86, 142]. Based on the pioneering fully convolutional network [115], previous studies have made important achievements by greatly increasing the performance on various challenging semantic segmentation benchmarks [15, 32, 40, 224]. Despite their superiority, these powerful models, built upon heavy deep neural networks, suffer from the low inference speed and strict requirements for computing devices.

Most of the existing works mainly address efficient semantic segmentation through (i) *designing compact backbone architectures* [47, 88, 89, 137, 148, 165, 206, 221], (ii) *effective model compression methods* [23, 90, 94, 101, 141, 220], (iii) *exploiting reliable context and boundary information* [53, 93, 113, 153, 205]. However, those methods mainly speed up the inference with fixed network structures, while in practice, the equipped resources are quite different across diverse devices. Even for the same device, the availability of hardware resources varies over time. Suppose we want to switch between models of different sizes according to the ideal accuracy-efficiency tradeoff. One straightforward way is to train multiple independent models with different structures and parameters and load a specific one during inference. However, it requires a longer training time and more memory for storage. Unlike previous works, we focus on improving the flexibility of the semantic segmentation model.

The recent work [210] proposes a slimmable neural network that can adjust the width of the network for different inference speeds. However, they mainly focus on image classification and only apply their slimmable models as backbones on instance segmentation tasks, while the other parts (*e.g.*, the decoder) are non-

---

\*This chapter is based on a publication in the ACM International Conference on Multimedia (ACMMM 2022) [197]

Table 2.1: The FLOPs and number of parameters of semantic segmentation networks (except for the backbone) and their proportions of the whole model, with image size  $1024 \times 2048$ .

Networks	SFNet [93]		DeepLabv3+ [22]	
Backbone	ResNet50	ResNet18	ResNet50	MobileNetv2
GFLOPs	436.3   72%	107.5   55%	663.5   45%	6.3   34%
Params	7.7M   25%	1.5M   12%	16.8M   40%	2.7M   59.3%

slimmable. Due to the resolution of the output image, even if a relatively simple structure is used in the decoder part, including up-sampling and multi-level feature aggregation *etc*, the decoder still requires a large amount of computation during inference. We show the computation cost (in FLOPs) and the number of parameters of several mainstream segmentation models, including SFNet [93] and DeepLabv3+ [22], in Table 2.1. In these models, the Pyramid Pooling Module (PPM) [222] and the decoder account for more than one-third of the overall calculation, while the parameters for most of them are the minority of the whole model. Based on [210], we focus on semantic segmentation and aim to lower computational cost from the perspective of reducing the overall size of the network, rather than just backbones. Motivated by this, we propose a slimmable semantic segmentation network (SlimSeg) that leverages the slimming mechanism to dynamically adjust the channel of features on every single layer. The network’s capacity can be switched with the size of width according to the computational requirements, thereby controlling the trade-off between accuracy and inference time. In addition, we apply stepwise downward inplace distillation for training smaller subnetworks, which means that smaller subnetworks are learned from the larger ones. This leads to consistent results between different submodels.

Moreover, we also found that the differences between the predicted results of slimmable subnetworks with different widths mainly exist along the semantic boundaries. Previous works [213, 226] also report that most existing segmentation models fail to make right predictions along the semantic boundaries. To further improve the segmentation quality on the boundary and narrow the accuracy gap between each subnetwork, we introduce a semantic boundary detection head on the low-level features and additional supervision named semantic boundary guided loss. This loss leverages the predicted boundaries as guidance to calculate a weighted bootstrapped cross-entropy. The boundary detection head can be removed during inference, so it does not introduce any additional computation.

Our SlimSeg is a general scheme that can adapt the existing segmentation models to width switchable models without any new structural design. The ex-

perimental results on Cityscapes [32] and CamVid [15] based on SFNet [93] and DeepLabv3+ [22] demonstrate the slimmable model has comparable accuracy to independent models. Furthermore, our method shows higher accuracy on smaller subnetworks with the stepwise downward distillation and proposed boundary guided loss. The contributions are summarized as follows:

- We propose a simple but effective slimmable semantic segmentation method (SlimSeg) which can adjust the capacity of the model depending on the desired trade-off between accuracy and efficiency.
- We present the boundary supervision, including a low-level boundary detection head and a boundary guided loss to improve the accuracy of semantic segmentation in boundary regions, especially for the smaller subnetworks.
- Extensive experiments and analysis indicate the efficacy and generalization ability of our proposed method, both quantitatively and qualitatively.

## 2.2 Related work

This section focuses on three main related topics: generic semantic segmentation, efficient semantic segmentation and dynamic neural networks.

### 2.2.1 Generic semantic segmentation

A typical semantic segmentation architecture generally includes two parts: encoder and decoder. The encoder module extracts image features through convolution and downsampling. Generally, the encoder is adapted from image classification models trained on ImageNet [35], such as VGG19 [154], ResNet [58], *etc.* Since semantic segmentation conduct pixel-level classification, the typical fully connected layers are replaced by convolutional layers [115]. To utilize the global context, the Pyramid Pooling Module (PPM) [22, 222] is employed to increase the receptive field without an increase in parameters. However, massive computations are introduced by PPM and other feature fusion modules performed on high-resolution features neighbor to the output. To pursue better global and local feature fusion, models [158, 223] based on more powerful backbones, such as HRNet [173] and ViT [37], have been proposed. These models have achieved higher accuracy, but are limited by the hardware requirements in practice. Our approach takes advantage of the sophisticated models and achieves variable capacity through width slimming, enabling fast inference while maintaining accuracy.

### 2.2.2 Efficient semantic segmentation

Efficient semantic segmentation needs to consider both accuracy and computational cost. Existing methods trade accuracy and speed along three different lines.

**Hand-crafted compact backbone architecture.** An effective backbone can greatly improve the upper bound of performance. The works [47, 88, 89, 137, 148, 165, 206, 221] design lightweight backbone architectures from scratch to pursue more efficient inference. Some works [89, 137, 221] devised multiscale image cascades and feature fusion mechanisms to achieve a good accuracy-speed trade-off. Others [88] improve existing network layers to create sufficient receptive field and densely utilize the contextual information. BiSeNet [206] introduced a shallow spatial branch to process full resolution images while learning context information by a deep branch.

**Machine-driven architecture optimization.** Neural Architecture Search (NAS) [227] is an effective technique to switch the labor-intensive architecture design to an automatic machine-driven optimization process, and this technique has been applied to semantic segmentation in recent years. From repeated cell structure [141, 220] to more flexible network structure [90], different types of network (e.g. graph convolution network [101]), or explicitly taking latency into consideration [23, 94]. FasterSeg [23] introduces the teacher-student co-searching and flexible multi-resolution branches aggregation structure. Although the latitude of the search space is continuously improved [219], it still requires longer training time and more effective search strategies.

**Feature mining and aggregation.** By exploiting the potential of existing lightweight models, rather than building new architectures, these methods learn more favorable context information. Knowledge distillation [61] has shown its effectiveness on segmentation tasks by improving the accuracy of a lightweight student model and speed-up its convergence by transferring learned knowledge from a sophisticated teacher network. Liu *et al.* [113, 153] provide a comprehensive analysis of feature distillation at different levels, from various cumbersome models to compact models. Others investigate multi-level feature aggregation to alleviate the side effects of up and down sampling [93] or enlarge the receptive field of lightweight networks [53, 205].

Although these efficient semantic segmentation approaches improve the accuracy-efficiency tradeoff from different perspectives, the resulting model is still limited to fixed size and operates at a single tradeoff. Unlike these methods, we enable adjustable computation with one single model and ensures good accuracy for each submodel of different size.

### 2.2.3 Dynamic neural networks

Dynamic neural networks [55] reduce average inference cost by adaptively changing characteristics of the computational graph, including the resolution, depth, and width. Reducing the **resolution** of the input image is the most straightforward way to lower computational costs. For images with relatively simple context, equivalent prediction accuracy can be achieved with lower resolutions. Some works [184, 204, 225] propose parallel training for multi-resolution inference with a single model. Networks with dynamic **depth** speed up inference by skipping residual blocks adaptively [96, 170, 179] or early exiting when shallower subnetworks have high enough confidence [67, 82, 204]. The number of feature channels, *i.e.* **width**, is also a key factor of efficiency. One way of enabling various channel inference is dynamic pruning. By identifying and skipping the insignificant channels during inference [48, 66, 85] or training a hypernetwork to select the filters [25], the channel complexity can be lessened. Moreover, [208–210] propose slimmable neural networks with embedded submodels sharing parameters that are executable at different widths, allowing immediate and adaptive accuracy-efficiency trade-offs at runtime. Based on the success of slimmable neural network, Liang *et al.* [85] improve the hardware efficiency by introducing a dynamic slimming gate that adaptively adjusts the network width with negligible extra computation cost. Although dynamic neural networks have shown their effectiveness on strategically allocating appropriate computational resources, most works still focus on image classification and some other low-level vision tasks, such as image compression [203], denoising [72] and image generation [63]. Different from previous works, we study dynamic semantic segmentation models through our analysis.

## 2.3 Methodology

In this section, we first introduce the pipeline of our proposed framework in Section 2.3.1. Then we describe the stepwise downward knowledge distillation and the semantic boundary guided loss in the Section 2.3.2 and Section 2.3.3, respectively.

### 2.3.1 Slimmable segmentation framework

Image semantic segmentation requires assigning a category label to each pixel in the image from several semantic categories. Given an image  $x$ , a segmentation network  $\mathcal{S}$  parameterized by  $\theta$  implements a mapping  $p = \mathcal{S}(x; \theta)$ , where each spatial element of  $p$  is a probability vector indicating the probability of each semantic category, from which the most probable is selected. Ideally, it should correspond to the category indicated in the corresponding ground truth segmentation map  $y$

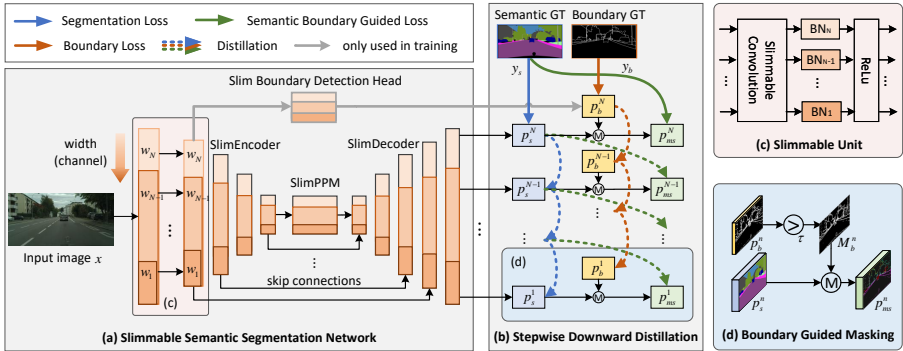


Figure 2.1: Overview of our slimmable semantic segmentation framework. (a) The whole network, including the encoder, PPM, decoder and boundary detection head, is slimmable. The boundary detection head can be removed during inference. (c) Each slimmable unit includes a slimmable convolution layer, independent BNs for each width and a ReLU layer. (b) The largest network with width  $w_N$  is supervised by the ground truth labels, and the smaller models with width  $w_n$  are learned from larger models with width  $w_{n+1}$  by stepwise distillation. (d) The predicted boundaries are used to generate boundary masked probability maps for calculating the boundary guided loss.

(coded as one-hot probability vectors per pixel). During training, the loss minimized is the cross-entropy  $\mathcal{L}_{CE}(p, y; \theta)$  between the predicted probability and the ideal one-hot label. In practice, this loss is averaged over the pixels in the image and the image-segmentation pairs  $(x, y)$  in the training dataset.

In this work, we propose a flexible semantic segmentation framework, named as SlimSeg, which can adapt its model capacity during inference via the slimming mechanism to accommodate various levels of computing power. More specifically, we define different sets of widths (*i.e.* number of channels in each convolutional layer) of the segmentation network. Thus, the segmentation network contains  $N$  subnetworks with parameters  $\{\theta_{w_1}, \theta_{w_2}, \dots, \theta_{w_N}\}$  with  $N$  increasing widths  $w_1 < w_2 < \dots < w_N$ , respectively. For every convolutional layer implementing slimming, the parameters are built as subsets of larger (sub)networks as  $\theta_{w_1} \subset \theta_{w_2} \subset \dots \subset \theta_{w_N} = \theta$ . Then, the objective of our task becomes optimizing all the subnetworks with  $\sum_{n=1}^N \mathcal{L}_{CE}(p^n, y; \theta_{w_n})$ , where  $p^n$  is the predicted category probability vector of the  $n^{\text{th}}$  subnetworks with parameters  $\theta_{w_n}$ . The loss is also averaged over pixels and training data, and then minimized over the parameters  $\theta$ . Note that we could also replace the (one-hot) ground truth label  $y$  with the soft label  $p_{n'}$  predicted by larger

subnetworks to distill its knowledge. We describe our loss functions in more detail in Section 3.2 and 3.3. Henceforth, we will also omit the explicit dependencies on the model parameters for the sake of simplicity.

The overall pipeline of our SlimSeg is illustrated in Fig. 2.1. We deploy width slimming on the entire network, including the encoder for feature extraction, the Pyramid Pooling Module (PPM) [222] and the decoder for feature aggregation and classification. The number of channels is adjusted through the slimmable convolutional layer [210], which produces different output feature channels by adjusting the number of filters. The slimmable convolution will result in a different output feature distribution. Following [210], we use independent batch normalization (BN) layers for each width, which only introduces very few parameters to the overall model.

### 2.3.2 Stepwise downward distillation

To utilize the knowledge learned by large submodels to guide the learning of the smaller submodels, we apply in-place knowledge distillation from larger (sub-) networks to smaller ones. Unlike previous knowledge distillation on segmentation [113, 153], we do not learn from an already trained (fixed) sophisticated model to improve another independent compact model. We introduce stepwise downward in-place distillation, where class probabilities estimated from the larger subnetwork are used as soft targets for training the next smaller subnetwork. The largest subnetwork is supervised by the ground truth labels. Note that the parameters of a smaller subnetwork are also a subset of larger ones, which means that the smallest subnetwork will learn the most important features implicitly to guarantee the accuracy of larger submodels. This leads to the following loss function:

$$\mathcal{L}_{seg} = \mathcal{L}_{CE}(p_s^N, y_s) + \sum_{n=1}^{N-1} \mathcal{L}_{KD}(p_s^n, p_s^{n+1}), \quad (2.1)$$

where  $\mathcal{L}_{CE}$  denotes the cross entropy loss, and  $p_s^n, y_s$  are the segmentation probability map predicted by the  $n^{th}$  submodel and the ground truth semantic label, respectively. Instead of computing the Kullback-Leibler divergence between two probabilities, we use soft target cross-entropy loss (we denote it as  $\mathcal{L}_{KD}$  to distinguish it from  $\mathcal{L}_{CE}$ , which applied with ground truth supervision). We found that the cross-entropy between two probabilities is more stable during training than the Kullback-Leibler divergence, which is also a common setting for the knowledge distillation in [113, 208–210].

In practice, stopping the gradients of the supervising tensor predicted by the larger width is necessary, so that the loss of a subnetwork will never back-propagate

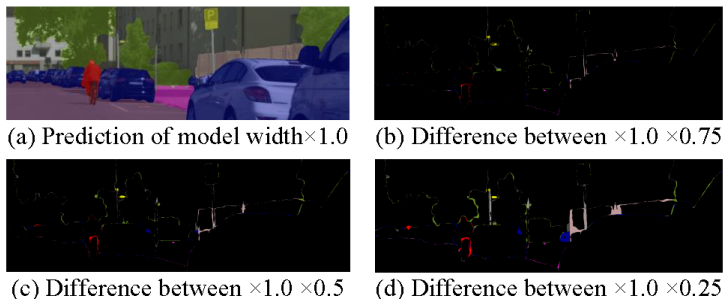


Figure 2.2: Difference maps between submodels. (a) Predicted semantic map of the  $\times 1.0$  model. (b)-(d) Difference map between the smaller submodels and the  $\times 1.0$  model, where the consistent (inconsistent) predicted pixels are shown as black background (ground truth color codes). Better view in color.

through the computation graph to larger subnetworks. We performed experiments on the effectiveness of distillation and the type of optimal teachers. The results show that using the probability map predicted by previous subnetworks as the soft target can lead to better performance. For more details, see Section 2.4.3.

### 2.3.3 Semantic boundary guided loss

Based on the training framework and distillation method presented above, we can already obtain varying amounts of computation of multiple subnetworks with partially shared parameters. To further improve the performance, especially for the smaller subnetworks, we compare the semantic labels predicted by different subnetworks trained only with the loss  $\mathcal{L}_{seg}$ . As illustrated in Fig. 2.2, the differences between the segmentation results of subnetworks with different widths are mainly near the borders between different semantic categories. Moreover, as the width decreases, the gap between the predictions gets larger.

Motivated by this observation, we introduce extra boundary supervision to improve the accuracy in those regions, especially for small subnetworks. Specifically, we introduce an additional boundary detection head with a simple structure, including a slimmable unit (Fig. 2.1 (c)) and a slimmable convolution layer with kernel size 1 followed by a sigmoid layer, on the low-level features. The output of this head  $p_b^N$  is supervised by the binary boundary masks generated by the semantic segmentation ground truth labels  $y_b$ . The pixels within 3 pixels from the semantic border are marked as boundary regions. We apply binary cross-entropy loss to constrain boundary detection with:



$$\mathcal{L}_b = \mathcal{L}_{BCE}(p_b^N, y_b) + \sum_{n=1}^{N-1} \mathcal{L}_{KD}(p_b^n, p_b^{n+1}), \quad (2.2)$$

where we also leverage knowledge distillation to subnetworks with the soft boundary labels predicted by the larger one, except for the largest width that uses the boundary ground truth  $y_b$ . Unlike [36], our boundary detection head is used only on training and can be removed during inference, so it does not introduce any extra computation. The head helps enhance the low-level features of boundary regions.

Besides, the estimated boundary also perform as a reference to resample the misclassified pixels on the border to calculate the boundary guided segmentation loss, which can be regarded as a hard sample mining strategy. As shown in Fig. 2.1 (d), taking the boundary probability map  $p_b$  predicted by the boundary detection head, we generate a confidence binary mask  $M_b$  to locate those pixels which might be situated near to semantic boundaries:

$$M_b(u, v) = \begin{cases} \text{valid}, & p_b(u, v) > \tau \\ \text{invalid}, & \text{otherwise} \end{cases}. \quad (2.3)$$

The values in  $M_b$  are element-wise calculated by comparing the boundary confidence score  $p_b$  at each location  $(u, v)$  with a predefined threshold  $\tau$ . We empirically set  $\tau$  to 0.7 in our experiments. Only valid pixels are included in the loss calculation. Similar to  $\mathcal{L}_{seg}$ , the cross-entropy loss and the knowledge distillation loss of the masked semantic probabilities  $p_{ms}^n = M_b^n(p_s^n)$ ,  $n \in \{1, 2, \dots, N\}$  are calculated with:

$$\mathcal{L}_g = \mathcal{L}_{CE}(p_{ms}^N, y_s) + \sum_{n=1}^{N-1} \mathcal{L}_{KD}(p_{ms}^n, p_s^{n+1}). \quad (2.4)$$

Then, the loss function for training our SlimSeg is calculated as a summation of the semantic segmentation loss  $\mathcal{L}_{seg}$ , boundary detection loss  $\mathcal{L}_b$  and the boundary guided segmentation loss  $\mathcal{L}_g$ :

$$\mathcal{L}_{full} = \mathcal{L}_{seg} + \lambda_1 \mathcal{L}_b + \lambda_2 \mathcal{L}_g \quad (2.5)$$

where  $\lambda_1, \lambda_2$  are hyperparameters, which are set to 10 and 1 in our experiments, respectively.

Finally, to clarify the training procedure of our proposed SlimSeg, we provide a Pytorch-style pseudo-code in Algorithm 1.

---

**Algorithm 1:** Slimmable semantic segmentation

---

**Ensure:** Dataset  $\mathcal{D}$ , width list  $\mathcal{W} = \{w_1, w_2, \dots, w_N\}$

**Require:** Slimmable semantic segmentation network  $\mathcal{S}$

```

1: for  $i = 1, 2, \dots, iteration$  do
2:   Get a mini-batch of image  $x$ , semantic label  $y_s$ , boundary label  $y_b$  from  $\mathcal{D}$ .
3:   Clear gradients of weights,  $optimizer.zeroGrad()$ .
4:   for  $w$  in  $sorted(\mathcal{W}, reverse = True)$  do
5:     Switch the BN layers to current width.
6:     Execute current subnetwork,  $p_s, p_b = \mathcal{S}(x; \theta_w)$ .
7:     Compute the masked probability,  $p_{ms} = M_b(p_s)$ .
8:     if  $w = w_N$  then
9:       Compute loss with ground truth,
           $loss = CE(p_s, y_s) + BCE(p_b, y_b) + CE(p_{ms}, y_s)$ .
10:    else
11:      Compute distillation loss,
           $loss = KD(p_s, y_s^t) + KD(p_b, y_b^t) + KD(p_{ms}, y_s^t)$ .
12:    end if
13:    if  $w > w_1$  then
14:      Save predicted probability  $p_s, p_b$  as teachers  $y_s^t, y_b^t$ .
15:    end if
16:    Compute gradients,  $loss.backward()$ .
17:  end for
18:  Update weights,  $optimizer.step()$ .
19: end for
20: return  $\mathcal{S}$ 

```

---

## 2.4 Experiments

### 2.4.1 Benchmarks and evaluation metrics

**Cityscapes.** Cityscapes [32] is a first-person perspective street-scene dataset with 19 semantic categories, 5000 fine annotated images with 2,975, 500 and 1,525 images for training, validation and testing, respectively. The high resolution of the images (1024×2048 pixels) poses a great challenge to real-time semantic segmentation. For a fair comparison, we only use the fine annotated images for training.

**CamVid.** CamVid [15] is a road scene dataset from the perspective of a driving automobile. It consists of 367, 101 and 233 images for training, validation and testing with resolution 720×960. Following the pioneering work [42, 206], we use the

subset of 11 semantic classes from the 32 provided categories for a fair comparison with existing methods. The pixels out of the selected classes are ignored.

**Evaluation metrics.** For quantitative evaluation, we report the mean of class-wise intersection-over-union (mIoU) for accuracy comparison. The floating-point operations per second (FLOPs) and frames per second (FPS) are adopted for efficiency comparison. Besides, we also give the number of parameters for model size.

### 2.4.2 Implementation details

**Training.** We use the stochastic gradient descent (SGD) algorithm to train our models with the batch size of 8, stochastic momentum of 0.9 and weight decay of  $5e-4$ . As a common practice, the “poly” learning rate strategy is applied, in which the initial rate is multiplied by  $\left(1 - \frac{iter}{iter_{max}}\right)^{power}$  at each iteration with the power of 0.9. All the models are trained for 100K iterations with an initial learning rate of 0.01 and Online Hard Example Mining (OHEM) [103] on two NVIDIA GeForce 3090Ti GPUs. Data augmentation includes random horizontal flip, random resizing with the scale range of [0.5, 2.0], and random cropping to  $768 \times 768$  for Cityscapes and  $720 \times 720$  for CamVid.

**Inference.** For inference, we use the whole image as an input to report performance, unless explicitly mentioned. Evaluation tricks such as sliding window inference and multiscale testing are not adopted. The measurement of inference time is executed on a single NVIDIA GeForce 2080Ti with CUDA 10.1, CUDNN 7.0, and we report the FPS without TensorRT acceleration.

**Architectures.** We conduct the experiments based on two mainstream semantic segmentation networks: SFNet [93] and DeepLabv3+ [22]. SFNet is based on the Feature Pyramid Network [103] architecture with a backbone network pre-trained on ImageNet classification [35] as encoder, a pyramid pooling module and a decoder aggregating multi-level features from the encoder. Similarly, DeepLabv3+ [22] includes a feature encoder, an atrous spatial pyramid pooling module and a simple decoder with only several convolutional layers and upsampling. For SFNet, we use the slimmable ResNet50 [210] pre-trained on ImageNet [35], and slimmable ResNet18, DFNetV1, DFNetV2 [94] without pre-training as encoder. For DeepLabv3+, we report the results using the slimmable ResNet50 and MobileNetv2 [210] (both are pre-trained on ImageNet) as encoder. The input of the boundary detection head is the low level features output by the second stage of the backbones. The resolution of the input features are down-sampled 4 times compared to the original image. We apply four width multipliers [0.25, 0.5, 0.75, 1.0] in our experiments, except for DeepLabv3+-MobileNetv2 with [0.35, 0.5, 0.75, 1.0].

### 2.4.3 Ablation study

We conduct ablation experiments to validate the effectiveness of our width slimming training scheme, knowledge distillation method and the proposed boundary guided loss.

#### Width slimming training scheme

We compare the slimmable model with their independently trained counterparts to demonstrate the effectiveness of the width slimming segmentation training scheme. The independent models have the same architecture as the slimmable subnetworks, but can only operate on a single width. Note that both the independent and slimmable models are trained with the loss  $\mathcal{L}_{full}$  in Eq.2.5 for fair comparison, and the independent models are supervised by ground truth. We report the mIoU, number of parameters (M) and FLOPs (GMac) in Table 2.2. The slimmable models outperform the independent models of all width on SFNet (ResNet50, ResNet18) and DeepLabv3+ (ResNet50, MobileNetv2), while for SFNet (DFNetv, DFNetv2), the larger independent models are better than the slimmable one. We think this is because DFNet [64] is a compact backbone designed for best speed accuracy trade-off by neural architecture search, which have very little space to be compressed. Therefore, the gap between slimmable SFNet-DFNets submodels with different widths is also larger than ResNets. In terms of the amount of computation, with about 56% of the whole FLOPs, the submodel with width $\times 0.75$  achieves comparable performance as the full model. Besides, a slimmable model saves about 50% memories for storing the parameters compared with several independent models, and number will increase if we have more switchable width.

#### Globally slimmable *v.s.* Partially slimmable

In this section, we present specific experimental results to illustrate why we choose to slim the entire segmentation framework instead of just a part of it. In addition to computational considerations, it is also because the use of more complex decoders cannot significantly improve the accuracy of submodels.

Yu *et al.* [210] has applied their slimmable ResNet50 backbone on instance segmentation task, but except for the slimmable ResNet50, the other parts of the Mask-RCNN (i.e. the lateral layers and the decoder) are non-slimmable. While in our work, we set the entire network to be adjustable in width, even the for the Pyramid Pooling Module, the lateral layers and the decoder. We report the mIoU and FLOPs of the **globally slimmable** models, the **partially slimmable** models (with only the slimmable backbone), and their independent counterparts in Table 2.3. Two kinds of structures, including SFNet [93] and Deeplabv3+ [22], both with

Table 2.2: Comparison of independent and slimmable models on Cityscapes *val*. Bold numbers indicate the better mIoUs.

Network	Width	Independent		Slimmable		FLOPs
		mIoU	Param	mIoU	Param	
SFNet ResNet50	×1.0	78.3	31.20	<b>78.4</b> (0.1↑)	31.29	607.9
	×0.75	77.3	17.57	<b>77.9</b> (0.6↑)		343.4
	×0.5	76.3	7.82	<b>77.4</b> (1.1↑)		153.9
	×0.25	73.2	1.97	<b>74.4</b> (1.2↑)		39.4
SFNet ResNet18	×1.0	75.0	12.87	<b>75.6</b> (0.6↑)	12.89	243.4
	×0.75	74.0	7.24	<b>74.8</b> (0.8↑)		137.4
	×0.5	71.4	3.22	<b>72.5</b> (1.1↑)		61.5
	×0.25	65.5	0.79	<b>67.3</b> (1.8↑)		15.7
SFNet DFNetv2	×1.0	<b>73.6</b>	17.88	73.1 (0.5↓)	17.91	80.2
	×0.75	<b>71.4</b>	10.06	71.1 (0.3↓)		45.2
	×0.5	<b>70.0</b>	4.48	69.8 (0.2↓)		20.2
	×0.25	62.5	1.12	<b>64.2</b> (1.7↑)		5.2
SFNet DFNetv1	×1.0	<b>70.0</b>	8.42	69.4 (0.6↓)	8.44	32.8
	×0.75	<b>67.8</b>	4.74	67.0 (0.8↓)		18.6
	×0.5	65.0	2.11	<b>65.3</b> (0.3↑)		8.4
	×0.25	57.8	0.52	<b>59.8</b> (2.0↑)		2.2
DeepLabv3+ ResNet50	×1.0	78.0	40.35	<b>78.4</b> (0.4↑)	40.44	1463
	×0.75	77.6	22.71	<b>78.2</b> (0.6↑)		824.3
	×0.5	76.7	10.11	<b>77.6</b> (0.9↑)		347.6
	×0.25	74.0	2.54	<b>75.6</b> (1.6↑)		92.9
DeepLabv3+ MobileNetv2	×1.0	66.9	4.53	<b>67.9</b> (1.0↑)	4.58	18.5
	×0.75	63.3	2.57	<b>67.0</b> (3.7↑)		12.2
	×0.5	58.6	1.16	<b>64.3</b> (5.7↑)		5.7
	×0.35	56.1	0.57	<b>61.1</b> (5.0↑)		3.3

slimmable ResNet50 [210] pretrained on ImageNet as backbone, are tested. For the partially slimmable models, the number of channels in non-slimmable parts is fixed and the same as that of subnetwork with width ×1.0 in globally slimmable model. As illustrated in Fig. 2.3, the computation reduction brought by seldom slimming the backbone is relatively small. For partially slimmable models, the mIoU gap between submodels of different width is smaller than the gap between globally slimmable submodels, and the range of FLOPs is narrower, due to the fixed non-slimmable parts in partially slimmable submodels. At the same time, compared with partially slimmable models, globally slimmable models have more obvious

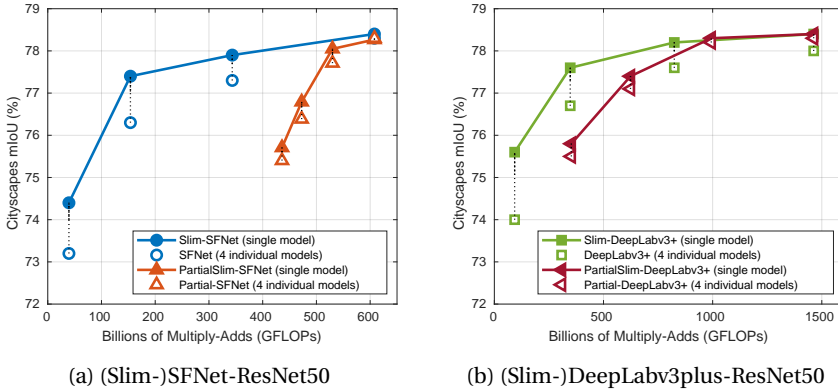


Figure 2.3: FLOPs-mIoU spectrum of globally and partially slimmable networks on Cityscapes *val*.

advantages on mIoU than corresponding independent models. Since the number of parameters of the non-backbone part in SFNet accounts for higher percentage than that in DeepLabv3, the difference on mIoU and FLOPs between globally and partially slimmable models is also larger.

Overall, comparing the mIoU-FLOPs curves of the globally and partially slimmable model, the globally slimmable model can achieve higher mIoU than the partially slimmable model with the same amount of computation, especially for smaller submodels. Therefore, we believe that globally slimmable semantic segmentation network leads to a better accuracy-efficiency tradeoff.

### Stepwise downward distillation

To make the most of the knowledge learned by larger submodels, we test different distillation settings and demonstrate the effectiveness of our distillation method.

**Does inplace knowledge distillation work?** We compare the mIoUs of training the slimmable model with and without stepwise downward distillation in Table 2.4. For the smallest subnetwork with width $\times 0.25$ , the mIoUs consistently improve with distillation under all combinations of loss functions. With the distillation strategy proposed by our work, mIoUs improve on all subnetworks, and among them, the smallest subnetwork with width $\times 0.25$  has the largest increase (0.8%) from 73.6% to 74.4%.

**Which is the best teacher for small submodels?** We train our slimmable model with soft targets predicted by different models as teachers in knowledge distilla-

Table 2.3: Comparison between globally slimmable models and partially slimmable models on Cityscapes *val*. \*Note that both the independent and slimmable models are trained with the sum of the three losses in Equation 2.5 for fair comparison.

Network	Slim-Part	Width	Independent*		Slimmable*		GFLOPs
			mIoU	Param	mIoU	Param	
SFNet ResNet50	Backbone (Partially)	×1.0	78.3	31.2	<b>78.3</b> (0.0↓)	31.3	608.0
		×0.75	77.8	20.2	<b>78.0</b> (0.2↓)		529.9
		×0.5	76.4	12.1	<b>76.8</b> (0.4↓)		472.6
		×0.25	75.4	6.9	<b>75.7</b> (0.3↓)		436.0
	Backbone +PPM +Decoder (Globally)	×1.0	78.3	31.3	<b>78.4</b> (0.1↓)	31.3	607.9
		×0.75	77.3	17.6	<b>77.9</b> (0.6↓)		343.4
		×0.5	76.3	7.8	<b>77.4</b> (1.1↓)		153.9
		×0.25	73.2	2.0	<b>74.4</b> (1.2↓)		39.4
DeepLabv3+ ResNet50	Backbone (Partially)	×1.0	78.3	40.4	<b>78.4</b> (0.1↓)	40.4	1462.8
		×0.75	78.2	26.3	<b>78.3</b> (0.1↓)		993.7
		×0.5	77.1	15.1	<b>77.4</b> (0.3↓)		623.7
		×0.25	75.5	6.9	<b>75.8</b> (0.3↓)		352.7
	Backbone +PPM +Decoder (Globally)	×1.0	78.0	40.4	<b>78.4</b> (0.4↓)	40.4	1462.8
		×0.75	77.6	22.7	<b>78.2</b> (0.6↓)		824.3
		×0.5	76.7	10.1	<b>77.6</b> (0.9↓)		347.6
		×0.25	74.0	2.5	<b>75.6</b> (1.6↓)		92.9

tion. For the student subnetwork  $\mathcal{S}(\theta_{w_n})$ , 'prev', 'largest', 'mean' indicates that the soft target is the predicted probability  $p^{n+1}$  of the last larger subnetwork  $\mathcal{S}(\theta_{w_{n+1}})$ ,  $p^N$  of the largest subnetwork  $\mathcal{S}(\theta_{w_N})$  [209] and the average of all the predictions  $\frac{1}{N-n} \sum_{j=n+1}^N p^j$  by the subnetwork larger than the current model  $\mathcal{S}(\theta_{w_{n+1}}), \dots, \mathcal{S}(\theta_{w_N})$ , respectively. Different from the setting of 'mean', 'larger' represents using the average loss of all the larger submodels' distillation. The mIoU of our slimmable model under different teacher settings are reported in Table 2.5. Note that all the models are trained with the sum of the three losses proposed. Our 'prev' setting, the step-wise downward distillation, outperform others by higher mIoU 74.4% and 77.37% on width ×0.25 and ×0.5. Using the average loss of all larger submodels results in better mIoUs on the larger submodels with width ×0.75 and ×1.0, but even lower mIoU than models trained without distillation on width ×0.25 and ×0.5. The results are consistent with the phenomenon that student network's performance degrades when the gap between student and teacher is too large [125].

Table 2.4: Ablation of knowledge distillation (KD) with different loss function by Slim-SFNet-ResNet50 on Cityscapes *val*.

KD	GT			Soft Target			mIoU (%)			
	$\mathcal{L}_{seg}$	$\mathcal{L}_b$	$\mathcal{L}_g$	$\mathcal{L}_{seg}$	$\mathcal{L}_b$	$\mathcal{L}_g$	$\times 0.25$	$\times 0.5$	$\times 0.75$	$\times 1.0$
w/o	✓						71.82	75.97	76.92	77.90
	✓	✓					73.08	76.34	77.12	78.14
	✓		✓				72.49	76.47	77.82	78.35
	✓	✓	✓				73.63	76.92	77.77	78.26
w	✓			✓			71.94	75.86	76.64	77.55
	✓	✓		✓	✓		73.12	76.04	77.21	78.21
	✓		✓	✓		✓	72.94	76.16	77.41	78.37
	✓	✓	✓	✓	✓	✓	<b>74.40</b>	<b>77.37</b>	<b>77.87</b>	<b>78.43</b>

 Table 2.5: Ablation of different knowledge distillation (KD) strategies with Slim-SFNet-ResNet50 on Cityscapes *val*. **Bold** numbers and *italic* numbers indicate the best and second best results.

KD	Teacher	Loss	mIoU (%)			
			$\times 0.25$	$\times 0.5$	$\times 0.75$	$\times 1.0$
w/o	-	$\mathcal{L}_{CE/BCE}(p^n, y)$	73.63	76.92	77.77	78.26
w	prev	$\mathcal{L}_{KD}(p^n, p^{n+1})$	<b>74.40</b>	<b>77.37</b>	77.87	78.43
	largest	$\mathcal{L}_{KD}(p^n, p^N)$	73.64	76.72	77.04	78.38
	mean	$\mathcal{L}_{KD}(p^n, \frac{1}{N-n} \sum_{j=n+1}^N p^j)$	73.24	76.25	77.53	77.85
	larger	$\frac{1}{N-n} \sum_{j=n+1}^N \mathcal{L}_{KD}(p^n, p^j)$	73.25	75.87	<b>78.02</b>	<b>78.61</b>

### Boundary supervision

**Accuracy on boundary.** As shown in Table 2.4, with boundary detection loss  $\mathcal{L}_b$ , the mIoUs on all widths are improved, especially for the smallest submodels, with 1.2% increase from 71.94% to 73.12%. For slimmable models trained without  $\mathcal{L}_b$  but with the boundary guided segmentation loss  $\mathcal{L}_g$ , we use the binary boundary ground truth label as a mask to generate a masked probability map  $p_{ms}$ . The boundary guided segmentation loss with ground truth labels also helps on improving the mIoUs on all width. With all the losses together, we get the best performance on all the submodels.

To demonstrate the improvements on semantic borders, we illustrate the histogram of the error pixels in Fig. 2.4. It shows the statistics of error pixels numbers



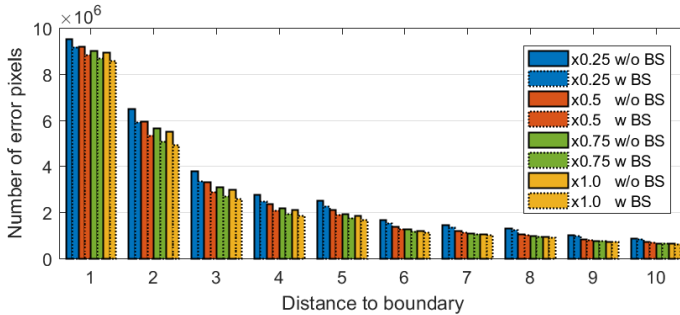


Figure 2.4: Comparison of the distribution of error pixels between slimmable models trained with and without boundary supervision (BS) on Cityscapes *val*. The model with boundary supervision has less error predictions on the boundary.

and their Euclidean distances to the nearest boundaries on 500 Cityscapes *val* images. Overall, the improved pixels are mainly distributed on the semantic borders. The improvement number of pixels within the range of 5 pixels along the borders accounts for about 50% of the total. Some qualitative results on Cityscapes *val* are shown in Fig. 2.5. With the boundary supervision, the predicted segmentation maps of each width model are more consistent, especially on the boundary regions. Segmentation results for some interior regions are also improved.

**Boundary groundtruth.** Boundary segmentation ground truth labels are generated based on the semantic segmentation ground truth labels, since we only focus on the boundaries between different semantic categories rather than the obvious image edges inside the regions with the same semantic category. In the experiments presented in other tables, we set the radius of the boundary region to 3 pixels. Here we compare the mIoUs of the Slim-SFNet-ResNet50 models using different boundary ground truth labels in Table 2.6. When radius equals to 3 pixels, the mIoU of each subnetwork is the optimal. Smaller boundary regions benefit to exploit hard samples, but when the number of boundary samples is too small, it is not conducive to the network to fully learn the characteristics of boundary samples.

**Input of the Boundary Detection Head.** The input of our boundary detection head is the low-level features extracted by the first few layers of the backbone networks. We report the mIoUs of models trained with different low-level features as boundary detection input. As shown in Table 2.8, ‘conv1’, ‘conv2\_x’, ‘conv3\_x’ represent the first three stages of ResNet50 [58], where ‘x’ indicates the numbers of the residual blocks. According to Table 2.7, slimmable models trained with boundary supervision, including the boundary detection head and the loss functions  $\mathcal{L}_b$  and  $\mathcal{L}_g$ ,

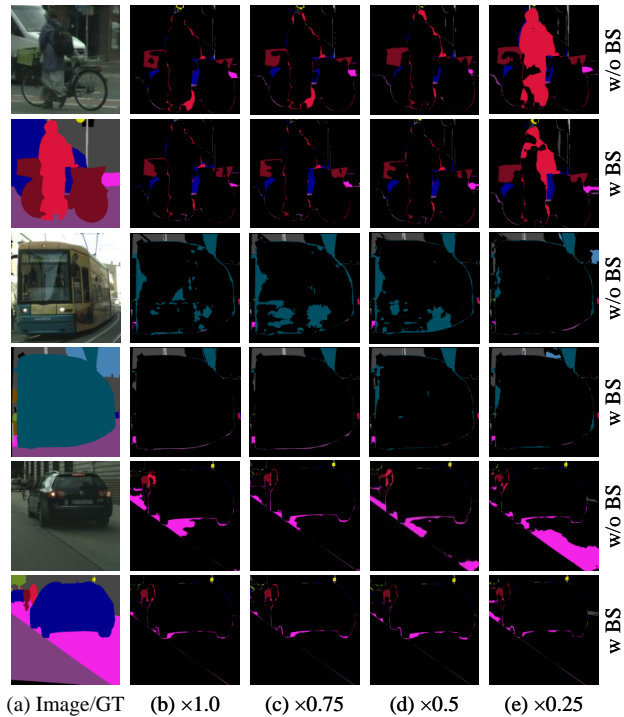


Figure 2.5: Visual comparison of our boundary supervision on Cityscapes *val*, in terms of errors in predictions, where correctly predicted pixels are shown as black background while wrongly predicted pixels are colored with their ground truth label color codes. Submodels with boundary supervision perform better on small objects and semantic borders.

Table 2.6: mIoUs of slimmable models trained with different boundary detection groundtruth.

Radius (pixels)	mIoU (%)			
	$\times 0.25$	$\times 0.5$	$\times 0.75$	$\times 1.0$
1	74.10	76.63	77.59	77.91
3	<b>74.40</b>	<b>77.37</b>	<b>77.86</b>	<b>78.43</b>
5	73.63	76.02	76.93	77.38

Table 2.7: mIoUs of slimmable models trained with different low-level features as the input of boundary detection head.

Boundary Head	Input Features	mIoU (%)				
		×0.25	×0.5	×0.75	×1.0	Ave
w/o	-	71.94	75.86	76.64	77.55	75.50
w	conv1	73.56	76.99	77.54	78.18	76.57
	conv2_3	<b>74.40</b>	<b>77.37</b>	<b>77.86</b>	<b>78.43</b>	<b>77.02</b>
	conv3_4	74.12	76.72	77.57	78.32	76.68

outperform the slimmable model without boundary supervision. Using the features output by layer ‘conv2\_3’ of ResNet50 lead to higher overall mIoU.

The layer ‘conv1’ contains only one convolutional layer. Although the extracted features can identify image edges, what we need is boundaries between different semantic categories, which contains semantic information to a certain extent. Moreover, as our main task is to perform semantic segmentation, in addition to exploiting the boundary pixels, the context information of the object itself is more important. If edge constraint is added to the feature output by the layer ‘conv1’, it will have a greater impact on subsequent features, so we add the boundary supervision on the deeper features. The features output by the layer ‘conv3\_4’ have been processed by three stages of convolutions, and contain some semantic information, so the overall mIoU outperforms the model using ‘conv1’. However, since the resolution of the features output by the layer ‘conv3\_4’ are down-sampled by 8 times compared to the original image, boundaries and details have been lost, so using the output features of the layer ‘conv3\_4’ as input for boundary detection is worse than layer ‘conv2\_3’.

**More visualization results.** To show the effectiveness of the boundary supervision, we present more visualization results of both features and segmentation results on Cityscapes *val* [32]. As shown in Fig. 2.6, the features in the regions near the boundary and the textured details of the objects, such as the head of the truck, are enhanced with boundary supervision, so the corresponding segmentation results in these areas have fewer errors. In Fig. 2.7, we compare the error maps of segmentation predictions between slimmable models with and without boundary supervision. As we can see, not only the semantic boundaries are improved with boundary supervision, but also the thin small objects, such as the pole, fence, traffic sign, have better results especially for small submodels. The gaps between the predictions of submodels with different width are narrowed with boundary supervision.

Table 2.8: Architectures of ResNet50 [58]. Down-sampling is performed by conv3\_1, conv4\_1, and conv5\_1 with a stride of 2. The input image size is  $3 \times 1024 \times 2048$ .

Layer Name	Output Size (C×H×W)	ResNet50
conv1	$64 \times 512 \times 1024$	$7 \times 7$ , stride 2
conv2_x	$256 \times 256 \times 512$	$3 \times 3$ max pool, stride 2 $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	$512 \times 128 \times 256$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	$1024 \times 64 \times 128$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5_x	$2048 \times 32 \times 64$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$

#### 2.4.4 Comparisons with real-time models

We compare our method with other existing state-of-the-art real-time methods on Cityscapes and CamVid. All the methods are evaluated by single-scale inference with the input image sizes listed for fair comparison. Our speed is tested on one GTX 2080Ti GPU with full image resolution as input, and we report the speed of without TensorRT acceleration.

**Results on Cityscapes.** We present the mIoU and inference speed of our slimmable SFNet-ResNet50 and SFNet-ResNet18 (both backbones are pretrained on ImageNet) and other real-time segmentation methods in Table 2.9. Our Slim-SFNet-ResNet50 achieves result (77.3%) with FPS 23.8. With ResNet18 as backbone, our method achieves 74.3% mIoU with 51.4 FPS.

**Results on CamVid.** Since the inference speed of different models is measured under different conditions, we list the corresponding GPU type. Table 2.10 shows the comparison results on CamVid between our method and SoTA methods. Our network achieves competitive trade-off between performance and speed by 80.1% (72.5% without ImageNet pretraining) mIoU with 55.7 FPS, which outperforms the original independently trained SFNet.

Table 2.9: Comparison with state-of-the-art on Cityscapes *val*. ‡ indicates the model is not pretrained on ImageNet.

Method	Resolution	Backbone	mIoU	FLOPs	FPS	Param
BiSeNetV1 [207]	768×1536	Xception39	69.0	14.8	105.8	5.8
BiSeNetV1 [207]	768×1536	ResNet18	74.8	55.3	65.5	49
CAS‡ [220]	768×1536	Searched	71.6	-	108	-
GAS‡ [101]	767×1537	Searched	72.4	-	163.9	-
DF1-Seg [94]	1024×2048	DFNetv1	74.1	-	106.4	-
DF2-Seg1 [94]	1024×2048	DFNetv2	75.9	-	67.2	-
DF2-Seg2 [94]	1024×2048	DFNetv2	76.9	-	56.3	-
SFNet [93]	1024×2048	ResNet18	78.7	247	18	12.9
BiSeNetV2‡ [206]	1024×2048	None	73.4	21.3	-	-
BiSeNetV2-L‡ [206]	512×1024	None	75.8	118.5	47.3	4.6
FasterSeg‡ [23]	1024×2048	Searched	73.1	28.2	108.4	4.4
STDC2-Seg75 [42]	768×1536	STDC2	77.0	54.9	97†	16.1
MSFNet [53]	1024×2048	ResNet18	77.2	96.8	41	-
CABiNet [205]	1024×2048	MBNetv3-s	76.6	12	76.5	2.64
CABiNet [205]	1024×2048	ResNet18	76.7	66.4	54.5	9.2
DDRNet-Seg [137]	1024×2048	DDRNet-23	79.5	143.1	37.1	20.1
Slim-SFNet			74.4	39.4	46.2	2.0
×[0.25, 0.5, 0.75, 1.0]	1024×2048	ResNet50	77.3	153.9	23.8	7.8
(Ours)			77.8	343.4	13.2	17.6
			78.4	607.9	9.0	31.2
Slim-SFNet			70.4	15.7	74.9	0.8
×[0.25, 0.5, 0.75, 1.0]	1024×2048	ResNet18	74.3	61.5	51.4	3.2
(Ours)			76.7	137.4	30.8	7.2
			77.9	243.6	21.8	12.9

Table 2.10: Comparison with state-of-the-art on CamVid *test* with image size 720×960. IM and CS represent using extra data, ImageNet and Cityscapes, for pretraining, respectively. †indicates the FPS is measured with TensorRT acceleration.

Method	Extra	Backbone	mIoU	FPS	GPU
BiSeNetV1 [207]	IM	Xception39	65.6	175	GTX1080Ti
BiSeNetV1 [207]	IM	ResNet18	68.7	116.3	GTX1080Ti
CAS [220]	None	Searched	71.2	169	TitanXp
GAS [101]	None	Searched	72.8	153.1	TitanXp
SFNet [93]	IM	ResNet18	73.8	36	GTX1080Ti
MSFNet [53]	None	None	75.4	91	GTX2080Ti
STDC1-Seg [42]	IM	STDC1	73.0	198†	GTX1080Ti
STDC2-Seg [42]	IM	STDC2	73.9	152†	GTX1080Ti
BiSeNetV2 [206]	CS	None	76.7	124.5	GTX1080Ti
BiSeNetV2-L [206]	CS	None	78.5	32.7	GTX1080Ti
DDRNet-Seg [137]	CS	DDRNet-23	80.6	94	GTX2080Ti
Slim-SFNet ×[0.25,0.5,0.75,1.0] (Ours)	CS	ResNet50	78.0	57.1	GTX2080Ti
			80.6	47.9	
			81.6	31.7	
			81.7	21.8	
Slim-SFNet ×[0.25,0.5,0.75,1.0] (Ours)	IM	ResNet18	71.0	102.8	GTX2080Ti
			73.6	98	
			74.8	72.6	
			75.2	55.7	
Slim-SFNet ×[0.25,0.5,0.75,1.0] (Ours)	CS	ResNet18	75.0	102.8	GTX2080Ti
			77.9	98	
			79.5	72.6	
			80.1	55.7	

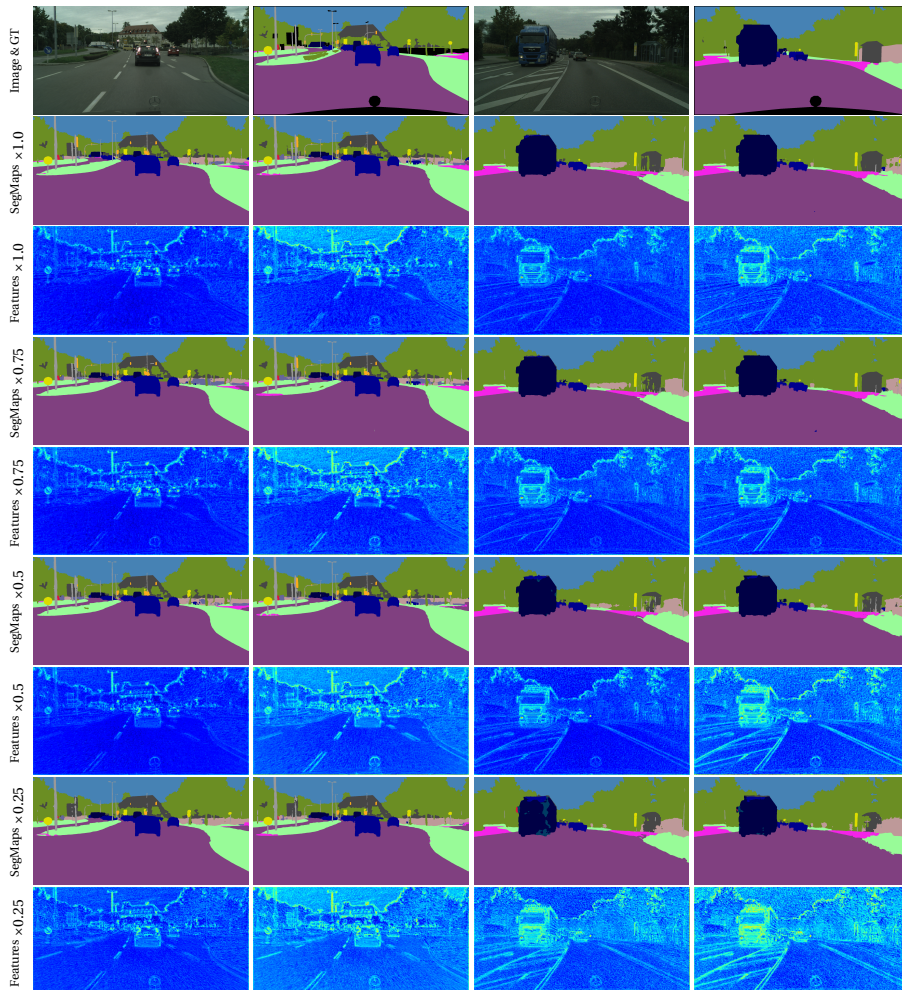


Figure 2.6: Visual comparison of our boundary supervision on Cityscapes *val*, in terms of the average feature maps of the output of layer 'conv2\_3' in ResNet50. Column 1 and 3 are the colored semantic segmentation maps and average features predicted by slimmable submodels without boundary supervision. The brighter color indicates the larger number of features. Column 2 and 4 are the results with boundary supervision. With boundary supervision, the features in boundary and textured regions are enhanced, which results in better segmentation results of these area.

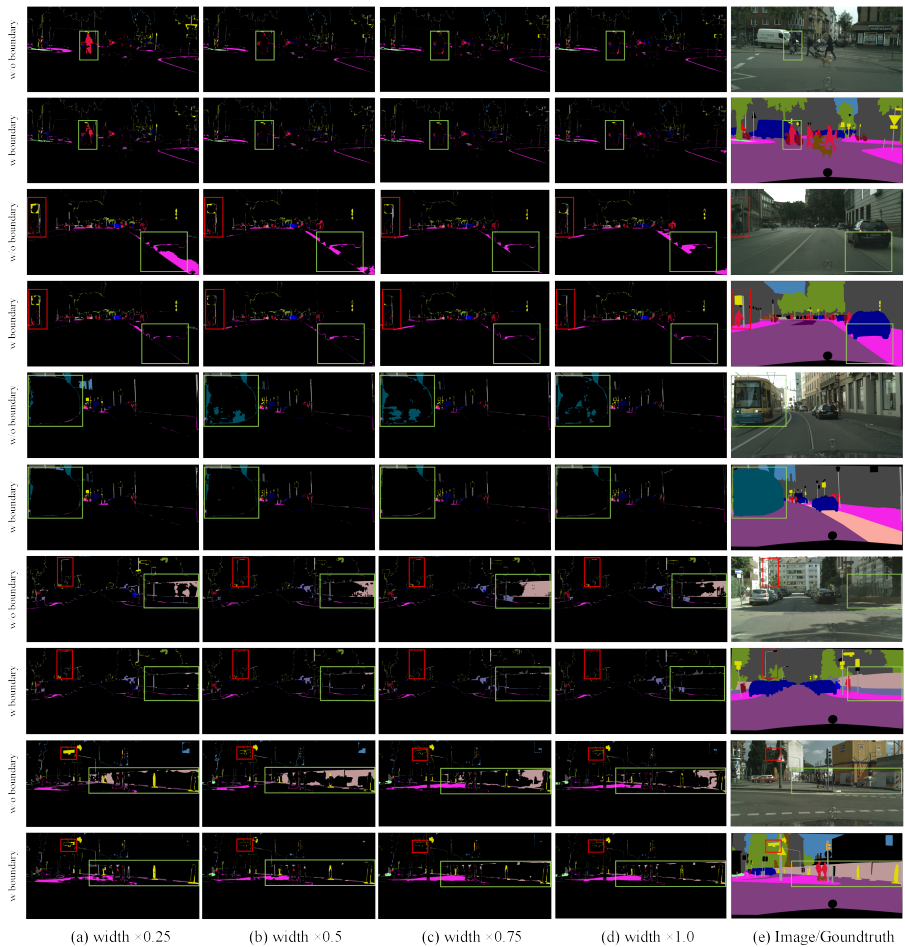


Figure 2.7: Visual comparison of our boundary supervision on Cityscapes *val*, in terms of errors in predictions, where correctly predicted pixels are shown as black background while wrongly predicted pixels are colored with their ground truth labels color codes. Models with boundary supervision performs better on small objects, such as poles and traffic signs, and semantic borders. Please zoom in for better viewing.



## 2.5 Concluding remarks

In this chapter, we propose a general slimmable semantic segmentation method, which enables adjustable accuracy-efficiency tradeoff through a width-swicthable segmentation network. We demonstrate the effectiveness of stepwise downward distillation on improving the performance of smaller subnetworks, and with less amount of features saved during training compared with other distillation strategies. Based on the observation of the difference between the predictions of each subnetwork, we introduce boundary supervision on low-level features of the network and propose a boundary guided loss to further improve the segmentation results of pixels along semantic borders. We demonstrate the effectiveness of the proposed method through extensive experiments with different mainstream semantic segmentation networks on the Cityscapes and CamVid. Our proposed method improves the accuracy of the smaller submodels without significant accuracy drops in large submodels.

Our work tackles the design of efficient and adjustable segmentation methods. In contrast to the SoTA real-time semantic segmentation methods, the performance of our methods do not rely on well-crafted compact network architectures. The experimental results demonstrated that our method can be directly applied to the mainstream segmentation frameworks and turn the fixed-computation models into adjustable ones. In this work, we use globally consistent width multipliers, but the optimal width of can be different for each layer, so we believe that the accuracy-efficiency tradeoff still has room for improvement. Furthermore, combining with image content, input resolution, and depth of the network, the dynamic inference can be further explored.



## 3 Integrating high-level features for consistent palette-based multi-image recoloring\*

### 3.1 Introduction

The need for color uniformity among a collection of images is relevant for applications such as photo collection editing and manipulation of images to have a coherent look and feel for use in websites and brochures. Achieving color consistency among a collection of images is a challenging problem.

Most existing works targeting color consistency focus on transferring colors between a single source image and a target image [160]. However, these color transfer methods often prove inadequate when dealing with multiple images that require a cohesive color theme. Alternatively, there are methods aimed at editing collections of photos [52, 138], but they come with specific prerequisites, such as the presence of identical objects (people, buildings, *etc.*) across the different images for feature matching. Furthermore, these methods are not designed to replace the original set of colors with a completely new color scheme.

In recent years, researchers have leveraged palette-based image recoloring [38, 69, 134, 218] to address challenges in multi-image color consistency. These approaches extract a color palette for each input image (source palettes) and generate a combined palette that represents all the images together (group palette). Recoloring is performed by matching individual images' source palette colors to colors in the group palette. Such techniques naturally allow the incorporation of palettes containing colors that were not originally present in the images (*e.g.* a color palette describing a brochure or website).

Existing palette-based image recoloring methods rely on low-level color statistics and often overlook high-level features related to visual perception. To address this limitation, we propose a comprehensive framework for multi-image recoloring that incorporates state-of-the-art palette-based techniques [134] and high-level visual features. By integrating these high-level elements, our framework aims to achieve image collections with enhanced color consistency and perceptually natural results. Specifically, we propose to include three modules—white balance correc-

---

\*This chapter is based on a publication in Computer Graphics Forum, 2023 [194]

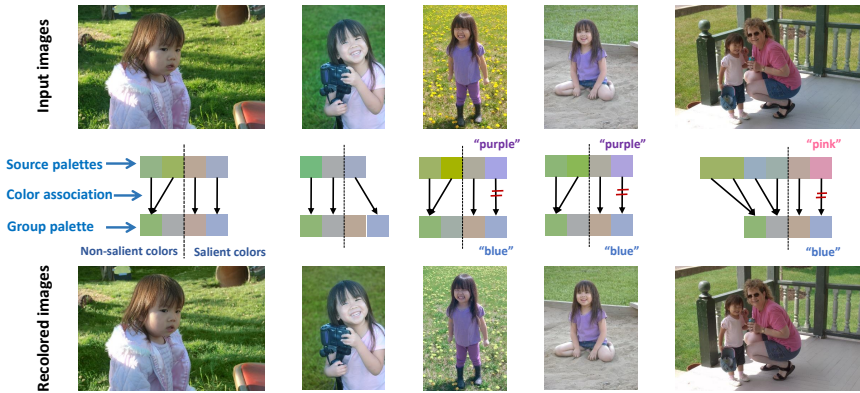


Figure 3.1: Our multi-image color consistency framework uses a palette-based recoloring strategy where each input image’s palette colors are mapped to a group palette. The estimation of the source palettes, group palette, and their associations are factors in three high-level features: *white balance*, *saliency*, and *color naming*. The recolored collection is visually pleasing and shares visually consistent colors. Top: Input image collection with inconsistent colors. Bottom: Our recolored results. Images are from [17].

tion, saliency-guided palette grouping, and color naming association—to complete the current palette-based multi-image recoloring framework. These individual modules contribute as follows:

**White balance correction.** Noticeable color differences of photos captured of similar scenes often arise due to different (or incorrect) white balance settings. Strong color casts can adversely affect both individual and group color palette extraction. To ameliorate such color casts, we introduce a white balance correction model to identify and correct the wrong white balance in the image.

**Saliency-guided palette grouping.** Existing multi-image consistency algorithms often ignore the importance of salient regions when establishing color associations between the source and group palettes. Consequently, while the overall color consistency of the image collection improves, it may lead to inconsistencies in the colors of salient areas due to the influence of non-salient regions. To address this issue, we propose using a saliency module that detects saliency regions to ensure consistent colors across both prominent (salient) and less prominent (non-salient) areas.

**Color-naming association.** Although we often think of colors sharing similar hues as being visually similar, they can be perceived as distinctly different colors. To

avoid unnatural color changes, we introduce a color-naming association procedure, which compares the similarity between the image color palette and the group color palette through the probability of each of the colors belonging to each of the 11 basic color names [6, 167].

As shown in Fig. 3.1, by incorporating these three high-level features into a multi-image consistency framework, we showcase significant improvements over relying solely on low-level color statistics. To validate the effectiveness of our framework, we conduct a user study that demonstrates a strong preference for images recolored using our approach.

## 3.2 Related work

Related works are presented for the following: photo-collection editing, palette-based recoloring, color naming, and saliency-aware image editing.

### 3.2.1 Photo-collection editing

Popular image editing software like Adobe Photoshop and Lightroom offer batch processing functionality to apply editing operations to an entire image collection. However, this approach often falls short in adapting to the collection’s varying scene content and lighting conditions.

Many prior color consistency approaches leverage shared content among the image coloration, such as recurring architectural structures or people. These methods employ techniques to adjust color transformation curves [52, 138, 191] or optimize white balance and gamma correction parameters [138] to enhance color consistency across the collection. Such methods, however, can be ineffective when the input collection lacks common content. To address this challenge, Nguyen *et al.* [134] introduced a palette-based framework for generic multi-image recoloring to adapt to different image contents within a collection. Their method focuses on palette manipulation, allowing users to intuitively adjust image colors. By operating in the *Lab* color space, it effectively avoids the issue of over-saturation. In a similar vein, addressing the challenge of scene changes in videos, Du *et al.* [38] proposed a 4D skew polytope with a limited number of vertices. This polytope serves as an approximate enclosure for video pixels across color and time dimensions, implicitly defining time-varying palettes.

In this paper, we build upon the framework proposed by Nguyen *et al.* [134]. Our key contribution is to incorporate high-level features into Nguyen *et al.*’s basic framework. As described in Sec. 3.1, these high-level features focus on white-balance correction, salient region considerations in individual palette extraction,

and color naming as a way to make associations between individual and group palettes.

### 3.2.2 Palette-based image recoloring

The palette-based image recoloring approach was introduced by Shapira *et al.* [152]. This approach simplifies image manipulation by summarizing an image with a small set of colors (a color palette). This technique allows for easy adjustments to the image by modifying the individual palette colors (*i.e.*, changing a palette color to a new color). This straightforward, user-friendly approach does not rely on extensive professional knowledge or reference images. The technique's success depends on extracting a good representative palette and adjusting the palette colors correctly. Palette extraction methods are typically categorized into two types: clustering-based and geometry-based. Clustering-based methods [19, 218] determine palette by the frequency of color occurrence. Geometry-based approaches [159–161, 185] construct convex hulls in various color spaces, with the vertices serving as the palette colors.

Clustering algorithms for palette extraction can be adversely impacted by strong color cast due to scene illumination (*i.e.* white balance) and locally distinctive colors with low occurrence. To address the illumination problem, Iwasa *et al.* [69] and Liu *et al.* [112] perform intrinsic color decomposition and limit recoloring to the reflectance image. However, inaccurate decomposition methods can adversely affect the final recoloring result, and recoloring only the reflectance image may not be intuitive as the color of the composed image changes when the illumination image is combined. Some k-means-based methods utilize color histograms for clustering, which may overlook small but significant colors, resulting in a non-representative palette. Kang *et al.* [75] enhance palette extraction by computing patch uniformity for local patches.

In contrast to the methods described earlier, our framework has a distinct objective of achieving color consistency across a complete set of images by merging the palettes of each individual image. To address the challenge of varying color casts due to illumination among images, our framework incorporates a white balance correction module to remove strong color casts from images when needed.

### 3.2.3 Color naming

Color names are the words used to describe and differentiate colors. Color naming systems can vary across cultures and languages. Seminal work by Berlin and Kay [7] found that most societies and cultures share a set of 11 linguistic distinct color names: *red, orange, brown, yellow, blue, pink, purple, green, black, gray, white*. Color

naming is crucial in design, industries, and vision research. Recent color naming models involve probabilistic graphical models [60, 114], deep neural networks [211], and statistical approaches [6] to map physical color stimuli to corresponding color names. The goal is to improve the accuracy and consistency of color naming predictions. The representations of the color name are also constantly being expanded, from the primary 11 colors to more detailed color classifications [212]. Other studies [129] are exploring the cultural and linguistic factors influencing color naming systems across different languages and cultures.

Color names can be considered features that are connected to human perception. By ensuring the consistency of color names between the source color and target color, we can mitigate unnatural color transitions to a certain degree.

#### 3.2.4 Saliency-aware image editing

Salient objects of an image are those on which our attention focuses first. In particular, the salient part of the image stands out from its surround because of a difference in one or more physical factors, discontinuities, or lack of correlation [74]. Saliency is widely used in image editing based on the natural perception difference of human vision to different regions of the image. The key details and saliency structures can be preserved by distinguishing the optimization target of salient and non-salient areas, improving the perceptual visual quality. The different methods differ in how they focus on highlighting the salient region desired by users: color transferring [121, 123, 168], cropping [166], or object removal [71].

In this chapter, we propose using saliency as a reference to differentiate prominent colors in salient areas from those in other regions. This strategy categorizes palette colors as belonging to salient or non-salient regions. When applying the group palette to recolor individual images, the two categories can be treated separately, minimizing the influence of improper color combinations on the salient objects in the image.

### 3.3 Methodology

In this section, we first introduce the pipeline of our proposed framework in Section 3.3.1. Then we describe the three proposed modules including the white-balance correction module, the saliency-guided palette grouping module and the semantics-guided palette grouping module in the Section 3.3.2, 3.3.3 and 3.3.4, respectively.

### 3.3.1 Multi-image recoloring framework

Consider a collection of images  $\{I_s^i\}_{i=1}^n$ , where  $n$  represents the number of images. The objective of multi-image recoloring is to obtain a set of recolored images  $\{I_t^i\}_{i=1}^n$  that retain the same content but exhibit improved perceptual color consistency. Palette-based image collection recoloring typically encompasses three primary steps: (1) extraction of the source palette for each image, (2) generation and matching of a group palette to each image, and (3) recoloring the images based on the palette adjustments.

In the initial step, we aim to derive a source palette for each image, denoted as  $P_s^i = \{c_s^1, c_s^2, \dots, c_s^{k_i}\}$ . These palettes comprise the primary colors specific to each image. The source palette is extracted by k-means clustering. The number of cluster centers  $k_i$  is determined by the percentage of explained variance, calculating the ratio between the total distortion and within-group distortion for different  $k_i$  values. During the process, the k-means clustering is performed varying  $k_i$  from 2 to 7. The optimal  $k_i$  value is chosen when the ratio is lower than 0.1, and the cluster centers are the source palette colors. Next, we proceed to the second step, where a unified group palette for the entire image collection, denoted as  $P_g = \{c_g^1, c_g^2, \dots, c_g^{k_g}\}$ , is generated. The group palette is determined using a weighted k-means clustering method that incorporating two additional terms [134]. The first term aims at avoiding palette reduction, where multiple colors in the source palette are assigned to the same color in the group palette. The second term aims at accommodating unassociated colors, that is, colors in the source palette that are not assigned to the group palette. An optimization is performed until a group palette is obtained and source color associations to the group palette no longer change (see [134] for more details). Finally, this matching solution is then utilized to produce the recolored images. Color mapping is performed in the *Lab* color space's *ab* channels, determined by the matching between each source palette color  $c_s$  and its corresponding group color  $c_g$ . The weights are determined by an inverse distance weighting function, assigning larger weights to closer palette colors.

The process described above relies solely on the statistical color information present in the images, enabling its applicability to image collections encompassing diverse content. However, it is important to acknowledge that color perception extends beyond pixel-level attributes [41] and that higher-level cues significantly shape our perception of color [135].

We present a comprehensive framework to address the limitations of existing multiple image recoloring approaches (illustrated in Fig. 3.2). As previously discussed, our framework incorporates three modules based on high-level features into the recoloring processing: white-balance correction, saliency-guided palette grouping, and color-naming association.



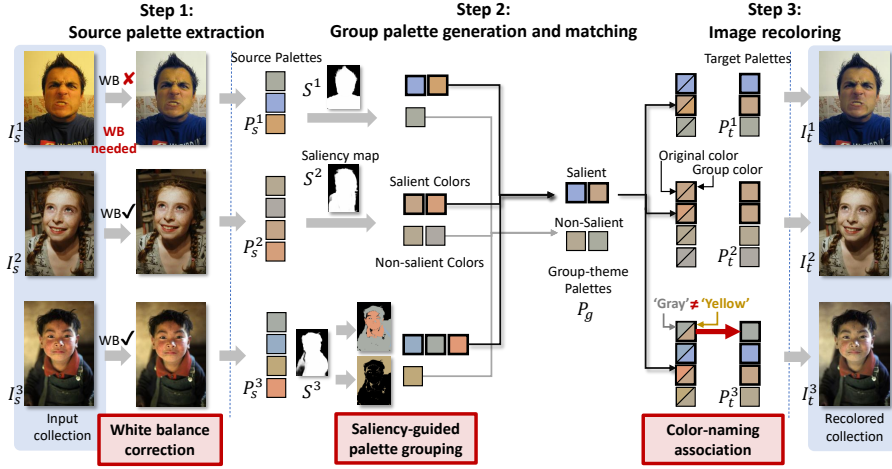


Figure 3.2: Our image collection recoloring framework. Given the input collection of images,  $I_s^i$ . Step 1: the source palette  $P_s^i$  of each image  $I_{wb}^i$  is extracted by k-means clustering. White-balance correction (either automatic or manual) is applied before the clustering procedure. Step 2: Colors in each source palette  $P_s^i$  are categorized based on the saliency map  $S^i$  into two groups: a sub-palette for the salient regions, and a sub-palette for the non-salient regions. The group palette  $P_g$  is computed based on the salient and non-salient source palettes of all the input images, respectively. The colors in each source palette match the color in the group palette. Color associations between the source and the group palette with inconsistent color names are removed. Step 3: The images are recolored based on the mapping between source palette  $P_s^i$  and group palette  $P_g$ .

### 3.3.2 White-balance correction module

White balance is a critical process applied by digital cameras. White balance aims at mimicking the color constancy ability of the human visual system. This ability allows us to perceive the color of an object the same, even when viewed under different illuminations. For example, we can perceive a sheet of paper as “white” under yellowish tungsten or bluish outdoor light. Significant research efforts have been dedicated to developing white-balancing methods within camera pipelines [4, 16, 50, 65, 169]. However, relatively little attention has been directed toward addressing the issue of enhancing images with incorrect white balance.

The presence of incorrect white balance can significantly impact the overall color distribution of an image. This, in turn, poses challenges when using palette-

based image recoloring techniques, as the extracted source palette and group palette may be biased towards the illuminant color (refer to Fig. 3.2). Our framework incorporates a pre-processing step to address images with strong color casts to ensure a collection with natural-looking colors. Specifically, we employ the method proposed by Afifi *et al.* [3]. In our framework, we allow users to manually select which images undergo white balance correction and which do not. For the automatic processing of images, we evaluate the color difference for all the images in the input collection to determine whether white balance correction should be applied. In particular, we compute the following metric:

$$\Delta E = \frac{1}{n} \sum_{i=1}^n \sqrt{(L_s^i - L_{wb}^i)^2 + (a_s^i - a_{wb}^i)^2 + (b_s^i - b_{wb}^i)^2}, \quad (3.1)$$

between the original input image  $I_s(L_s, a_s, b_s)$  and the image after white-balance correction  $I_{wb}(L_{wb}, a_{wb}, b_{wb})$ , where  $n$  is the number of pixels of the image. If the average color difference  $\Delta E$  of all the pixels in the image is larger than the set threshold  $d_{wb}$ , we consider the white balance of the original image to be inaccurate. In this case, we use the white balance corrected image as input for the subsequent processing. Note that the white balance correction is only applied to images deemed improperly white-balanced since well-white-balanced images do not impact the overall result, and therefore, leaving them uncorrected has little influence on the final output.

### 3.3.3 Saliency-guided palette grouping module

Salient regions play an important role in the initial stages of our visual system, as they are prioritized for further processing in the visual cortex, shaping our overall understanding of an image. During the process of group recoloring, it is important to try to preserve salient regions, even though their colors contribute to only a small portion of the images. Failure to do so can lead to a deviation in the perceived collection of images from the intended representation.

We introduce a module that generates a content-aware group palette, utilizing the saliency map as a reference. Specifically, we categorize the source palette into two distinct sub-palettes: the salient palette  $P_{s,salient}$  and the non-salient palette  $P_{s,non-salient}$ . One approach to obtain these sub-palettes is by extracting palettes separately from the image's salient and non-salient regions, respectively. This can be achieved using the masked image regions, as illustrated in Fig. 3.3 (a). However, this method may result in overlapping foreground and background palettes due to deficiencies in the saliency map. These deficiencies can include inaccurate edge segmentation and the presence of spurious regions.

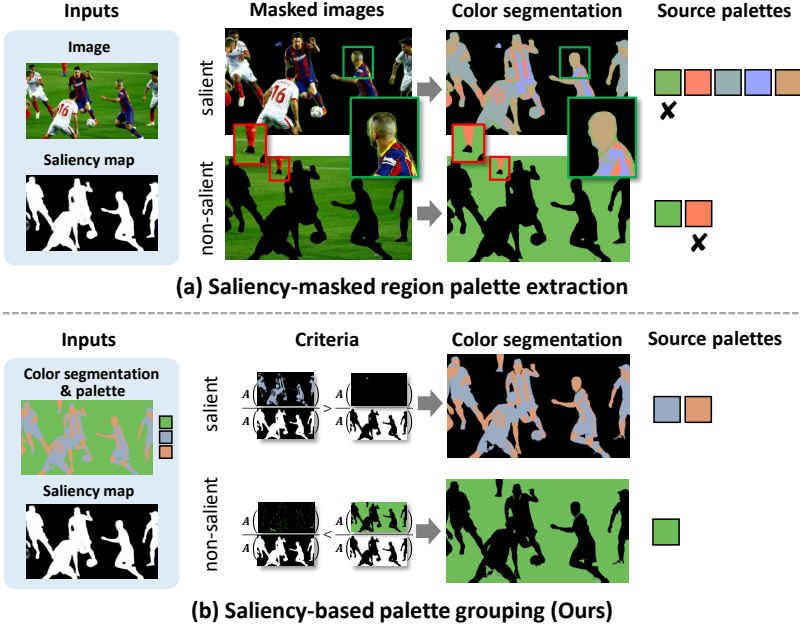


Figure 3.3: Saliency-based source palette grouping. The method in (a) uses the saliency map as a reference to extract the palette from the salient and non-salient regions, which may produce similar colors. (b) Our approach combines the saliency map and color segmentation to obtain palettes of salient colors, reducing the impact of inaccurate edges and missing areas in the saliency map.

To avoid such artifacts, we first extract the source palette by k-mean clustering following [134] and assign each pixel of the image with a color from the source palette to get a color segmentation map  $L$ . The value of each pixel in  $L$  represents the color of its corresponding cluster center. As shown in Fig.3.3 (b), the sub-palettes are then decided by comparing the proportions of the number of pixels with the same color label in the salient and non-salient regions using the following expression:

$$\begin{cases} c_s^k \in P_{s,salient}, & \frac{\mathcal{A}(L=c_s^k \cap S > \gamma)}{\mathcal{A}(S > \gamma)} > \frac{\mathcal{A}(L=c_s^k \cap S \leq \gamma)}{\mathcal{A}(S \leq \gamma)}, \\ c_s^k \in P_{s,non-salient}, & \text{elsewhere} \end{cases}, \quad (3.2)$$

where  $\mathcal{A}$  indicates the area of the region.  $L$  is the color segmentation map, where

$L = c_s^i$  indicates the region with color  $c_s^i$ . The saliency map  $S$  encodes a per-pixel probability of that pixel belonging to the salient region. To distinguish between salient and non-salient pixels, we use a threshold  $\gamma$ . Specifically, when  $S > \gamma$ , it indicates the presence of salient regions. This approach effectively assesses the importance of colors in different areas and facilitates the separation of salient and non-salient colors. In our framework, we use a CNN-based saliency object detection method [187] to obtain the saliency map—alternative approaches could also be used.

After organizing the original palette into salient and non-salient colors, the group palette is generated for each category. This avoids inconsistencies in the colors of salient and non-salient areas due to the influence of another region. The matching process between each image's source palette and the group palette is performed considering both salient and non-salient colors. This matching can be done in two different ways.

- **Both salient and non-salient.** To color match the salient and non-salient parts of the palette separately. In this case, our saliency module performs consistency in the salient and non-salient colors separately.
- **Non-salient only.** To only color match the non-salient while allowing the salient part to be kept as it was originally. In this way, salient regions are left unmodified, while the images' background is consistent.

Note that we use the first approach unless mentioned otherwise. The second method is suitable for some special scenarios where users want to maintain the diversity of the salient area.

### 3.3.4 Color-naming association module

In specific scenarios, there may be noticeable differences between the colors selected in the source palette and the group palette, resulting in unnatural recoloring. For example, in Fig. 3.10, the colors of the feather (column 1) and the clouds (column 3) underwent significant changes. These unnatural color transformations are easily noticeable to observers due to their large shifts in hue [105].

To address these issues, we propose a module in the group recoloring process that considers color naming. Specifically, our approach constrains the recoloring to only those colors that share the same color name based on the 11 basic color terms (red, orange, brown, pink, purple, yellow, green, blue, black, grey, white) [7]. This means that an orange color from the source palette, for instance, will not be transformed into a yellow color in the group palette. By incorporating color names as relevant perceptual features to assess the color differences between the

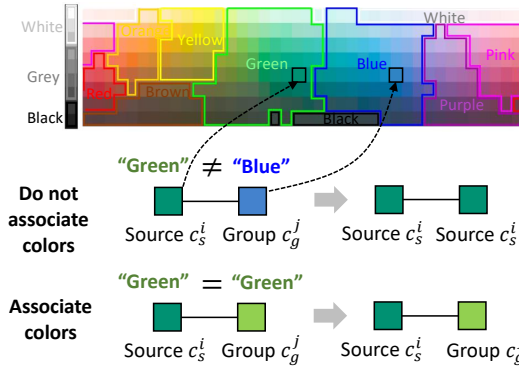


Figure 3.4: Color-naming association. The matched source color and group color are not associated if they have different color names.

matched colors in the source and group palettes, our aim is to maintain perceptual consistency between the input image and the recolored image. Compared to directly applying a threshold in perceptual color spaces, such as *Lab*, our approach greatly minimizes unnatural hue shifts with the incorporation of color-naming associations.

We employ a color naming identification model to determine the association between the source palette color  $c_s^i$  and the corresponding matched group palette color  $c_g^j$ . This model generates a probability vector indicating the likelihood of each RGB color belonging to specific color names (specifically, we utilize the model described in [167]). Subsequently, we calculate the Euclidean distance between the probability distributions of the source color  $p_{c_s^i}$  and the group palette color  $p_{c_g^j}$ . If this distance exceeds a certain threshold, denoted as  $d_{name}$ , we consider the color match inappropriate; therefore, the source color remains unchanged. By adjusting the value of  $d_{name}$ , we can control the extent of color modification in the recolored image as

$$\begin{cases} \text{associate match } (c_s^i, c_g^j), & \|p_{c_s^i} - p_{c_g^j}\|_2 \leq d_{name} \\ \text{do not associate match } (c_s^i, c_g^i), & \text{otherwise} \end{cases}. \quad (3.3)$$

An example of this procedure is shown in Fig. 3.4.

## 3.4 Experiments

### 3.4.1 Experimental setting

As previously described, our framework adds three modules to the basic framework proposed by Nguyen *et al.* [134]. The modules can be used independently or together to solve the hard cases in multi-image recoloring. The framework can be run in automatic mode, or interactively to obtain different recolored collections for different needs.

All results presented in this section were obtained using our automatic processing. To ensure a fair comparison, we limited the number of group colors and kept the parameters for our model unchanged. Specifically, we set the values for  $d_{wb}$ ,  $\gamma$ , and  $d_{name}$  to 20, 0.9, and 0.8 respectively. For the automatic process, the number of group palette colors is the average of the number of source palette colors. Thresholds used in the automatic process were chosen experimentally through parameter search (visually guided grid search). While adaptive variables could be a better way, implementing them is challenging since some inputs will inevitably require a user in the loop. Note that our interactive framework also allows the user to manually select images for white-balance correction, adjust the thresholds for each module, and apply different group palette numbers. The images used for these experiments were sourced from the MIT-Adobe FiveK Dataset [17] and Flickr [44]. Our multi-image recoloring framework is implemented with Python, and the user study computations are implemented using MATLAB's Psychtoolbox.

We compare against the two versions presented in the Nguyen *et al.*'s algorithm: the basic framework and the "unassigned" version. The latter one breaks connections among colors by penalizing large distances between the source and group colors. In both cases, we use the same parameters as proposed in [134].

### 3.4.2 Qualitative results

#### White-balance Correction Module

Color cast due to incorrect white balance is particularly noticeable when images contain people, as our perception is highly sensitive to the appearance of faces. An example showcasing the effectiveness of our white-balance module is illustrated in Fig. 3.5. In the first row, the original images are displayed in (a)-(e). In the second and third rows (versions of Nguyen's method), the dominant yellow color in (e) influences the palette extraction, resulting in the exclusion of the blue colors present in (a) and (e) from the palette selection. As a consequence, non-realistic outcomes are produced for these two images. However, this problem is resolved

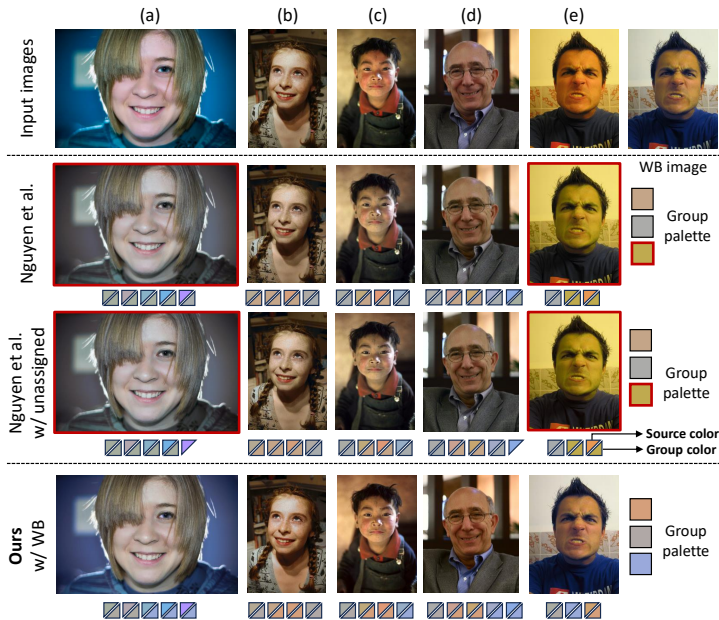


Figure 3.5: Results after applying the white-balance correction module are presented in the following order: input images, results from two versions by Nguyen *et al.*, and our results. In the images, red boxes are used to highlight unsatisfactory recolored images and palette colors. We obtain better color consistency among the recolored images by incorporating the white balance correction. The images used for this demonstration were sourced from Flickr [44].

by incorporating our white-balance correction module. In particular, a significant improvement can be observed in the last column of the first row. This correction enables the group palette to represent the blue colors present in the images better, leading to a more consistent set of recolored images that exhibit natural and vibrant tones across different skin tones—as shown in the last row.

### Saliency-guided Module

Our method can distinguish salient and non-salient areas even when the saliency map is inaccurate. This is shown in Fig. 3.6, where the saliency map—shown in the second row—does not completely distinguish the salient objects (in this case, the buildings), but our salient colors—shown in the third row—do correctly distinguish

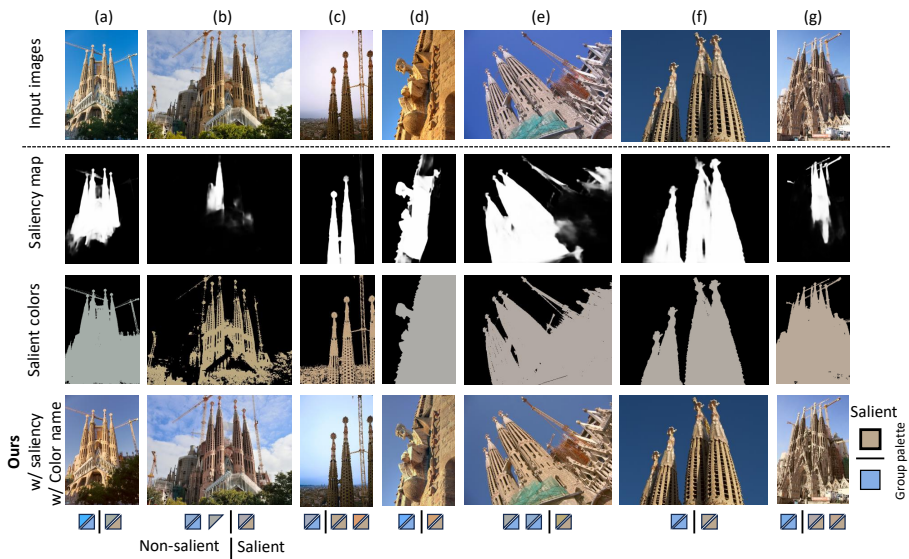


Figure 3.6: Results of introducing saliency-guided palette grouping. Even when inaccurate saliency maps are used (second row), our saliency colors procedure (third row) can cover all the salient elements, producing recolored images (last row) with consistency in both the salient and the non-salient regions. Images are from Flickr [44].

them. The results obtained by our method—last row—improve due to the correct categorization of the salient and non-salient regions at the palette level. In this way, the salient color obtained with our approach covers the common architecture, while the non-salient one covers the sky region.

Fig. 3.7 provides an illustrative example of the two types of matching options that can be performed using saliency information. Looking at the row that presents the result for the method that matches both salient and non-salient regions, we can observe that all the flowers (salient regions) are modified to a more consistent red color, while the leaves (non-salient regions) are recolored into a more consistent green color. The results show that the separation of salient and non-salient palettes avoids inconsistent color caused by unwanted associations between salient and non-salient colors. For the standard version of Nguyen’s approach without separation, the salient region in Fig. 3.7 (a), (c), and (e) will produce inconsistent recolored results due to the association with the non-salient color palette green. On the other hand, in Fig. 3.7-last row where only the non-salient region undergoes





Figure 3.7: Example for our two approaches for saliency recoloring. From top to bottom: Original image, the two versions of Nguyen *et al.*, the starting saliency map, the salient colors obtained by our procedure, and the results from our two versions. Red boxes mark the unsatisfactory recolored images. In our first result, we can see how the flowers are all converted to red by making both salient and non-salient regions consistent, and the leaves get a more middle green tone. In our second result, as we only match the non-salient regions, the flowers keep the same colors as in the original images, while the leaves are modified as in the previous case. We can see how we can obtain results that look natural in both of our results, in contrast to the results in Nguyen’s approach. Images are from Flickr [44].

consistency adjustments, our method results in a more uniform green color for the leaves, while the flowers retain their original colors. For this non-salient-only setting, the unassigned version of Nguyen’s approach achieves a result similar to

our last case. Nevertheless, it is worth mentioning that Nguyen’s approach relies on the color differences in the  $ab$  channels of  $Lab$  space, making it difficult to unassign unnatural color changes, as will be further explained in the subsequent module and user experiments.

### Semantics-guided module

The results of saliency detection can be considered as a binary classification, effectively distinguishing prominent foreground objects from the background, which is more suitable for scenes with relatively simple image content. However, for specific scenarios and applications, it may be necessary to divide the image into more complex regions or alter the colors of semantically significant objects. Therefore, building upon the saliency-guided module, we also do experiments on multi-image recoloring guided by semantic segmentation results.

To better demonstrate the results of semantic segmentation-based processing, we randomly selected several groups of images from various autonomous driving datasets, Viper [147] and Cityscapes [32], for experiments. For the corresponding semantic guidance, we utilize the groundtruth labels and the predictions from the SlimSeg model proposed in Chapter 2. Following the same approach as the saliency-guided module, we first extract the color palette of all the images in the input collection, and then classify the colors into several groups where the color is dominant in the region of this group. This method is the most straightforward way to extend binary classification results (saliency map) to multi-class classification results (semantic segmentation map). In Fig. 3.8 and Fig. 3.9, we illustrate the recoloring images under similar autonomous driving scenes but in a quite different style, such as different cities, illumination, *etc.* We take the common classes including “car”, “sky”, “road”, and “building” into consideration for recoloring. With the category-wise group palette solving, our approach gets more consistent color, especially in the sky and road region (see Fig. 3.9). Meanwhile, even if the semantic map predicted by the model is not accurate, it will not have a catastrophic impact on the results (see Fig. 3.8). It is worth mention that our approach could be applied to domain adaptation tasks as an augmentation method to narrow the color distribution gap between different domains.

### Color-naming association module

In Fig. 3.10, we show from top to bottom the input images, the results for both versions of Nguyen, and the results applying our color naming association module. As we can see in the figure, the standard module of Nguyen presents unrealistic colors in (a)-(c), while the unassigned version only solved the problem for (b), but it does not prevent the method from obtaining a greenish bird in (a) and blue clouds



Figure 3.8: Results of adding the semantics-guided module. From top to bottom: Input images from Cityscapes [32], the result of Nguyen *et al.*, the input semantic maps predicted by the Slim-DeepLabv3 model in Chapter 2, and our recoloring results. There is noticeable noise in the “road” region of the semantic segmentation prediction. However, our method is still able to achieve color-consistent recoloring results under the guidance of such noisy semantic maps. Particularly, in the regions of the “car” in column (a) and (c), our method obtains more consistent colors compared to the results without semantic guidance.

in (c). The reason is that, for this last version of Nguyen, the assignment is solved by minimizing the cost function giving a small penalization to the unassigned colors. However, this optimization does not consider any high-level perceptual features.

When applying our color naming association procedure, the results show how it addresses the previously mentioned problems. This is obtained by the ability of our module to break any matching in which the source palette and the group palette represent a different color name. Our method breaks a gray-green link in (a), a purple-blue link in (b), a gray-blue link in (c), and a brown-green link in (f).

Color naming helps mitigate drastic color changes (often due to hue shifts) in the recolored images. This restriction is well-suited for images containing objects with strong memory color associates, like skin tones, skies, and foliage/plants. Alternatively, relaxing the color name constraints or using no restrictions prioritizes larger consistency within the recolored collection, making it more suitable for creative applications like graphic design (Fig. 3.14). In such cases, applying color-naming association might limit the extent of achievable color transformations.

## Chapter 3. Integrating high-level features for consistent palette-based multi-image recoloring

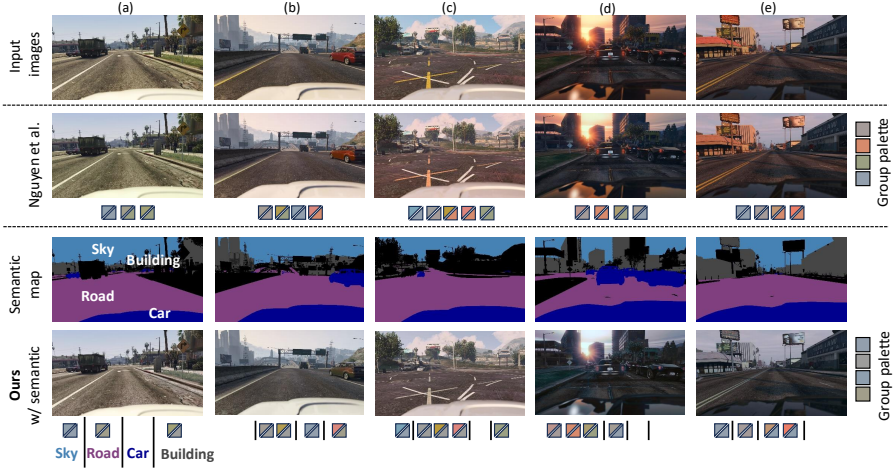


Figure 3.9: Results of adding the semantics-guided module. From top to bottom: Input images from Viper [32], the result of Nguyen *et al.*, the input groundtruth semantic labels, and our recoloring results. After incorporating semantic guidance, the hues of the sky and buildings in column (d) and (e) have shifted from warm tones to cool tones, resulting in a more consistent overall color tone of the images.

Our color-naming association operates as a binary choice between source and target colors. Applying this association can compromise color consistency among the recolored collection in some specific scenarios. This said, our approach is still better than methods that directly interpolate between the source and group colors. Using direct interpolation might work for some cases (for example the flower image in Fig. 3.10 (b)) as “blue” and “purple” are similar enough colors; but direct interpolation drastically fails when it is required to mix complementary colors, resulting in gray hues. An example of this is the red flower in Fig. 3.7 (a): the interpolation between “red” and “green” cannot produce a pleasant output.

### Modules Combination

We show the results of combining the saliency module with the other modules in Fig. 3.11. In the first two rows of the figure, the input images and the result for the unassigned version of Nguyen are shown. We can see how this last method presents washed-out colors in the t-shirts in (a) and how the yellow t-shirt and the yellow numbers in (c) are turned green.

The third and fourth rows show the saliency map and the salient colors used by

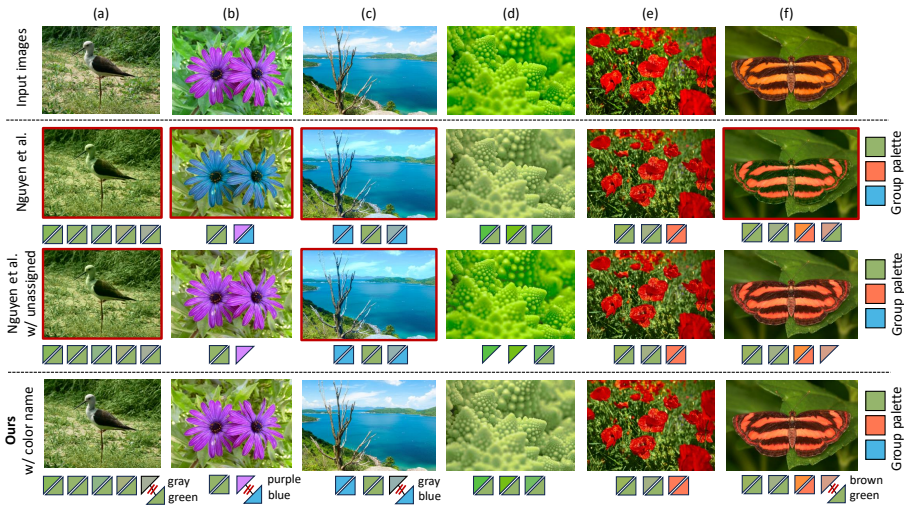


Figure 3.10: Results of adding the color-naming association module. From top to bottom: Input images, the two versions by Nguyen *et al.*, and our result. Red boxes highlight the unsatisfactory recolored images. In our results, we can see how the color-naming association module breaks matches in the (a), (b), (c), and (f), due to them having different color names in the source and the group palettes. This allows our method to obtain more natural images, avoiding the problems of the Nguyen’s approach, namely in the green color of the bird in (a) and the blue color of the clouds in (c). Images are from Flickr [44].

our method. The last three rows show the results of using only our saliency module, both the saliency and the color naming modules, and all three modules. In Fig. 3.11 we use the saliency approach that matches both salient and non-salient regions.

We can see that using only the saliency module might not be enough, and some colors can still be changed in undesirable ways (see the white t-shirt in (d)). But combining the saliency module with the color-naming and the white-balance modules can reduce such problems, providing more color consistency images across the image collection.

### 3.4.3 User study

Given the subjective nature of our framework, we perform a user study to determine preferences among the different images. We created ten groups of images and computed the color consistency results by the two versions of Nguyen *et al.* [134]

### Chapter 3. Integrating high-level features for consistent palette-based multi-image recoloring



Figure 3.11: Results for the combination of all our modules. From top to bottom we show the input images, the unassigned version of Nguyen, the starting saliency map, our salient colors, and our results with saliency, with saliency and color naming, and with all the three modules. Red boxes highlight the unsatisfactory recolored images. We can see how our approach is able to improve the results of Nguyen, especially in (a) and (c) -see the colors of the t-shirts in (a), and the yellow of the t-shirt and the numbers in (c)-. Images are from Flickr [44].

(standard and unassign) and three versions of our approach (only saliency, saliency + color naming, and saliency + color naming + white balance). The groups of images were selected to represent challenging scenarios for the baseline approach proposed in [134]. We also included in the experiment the input not-consistent images. Therefore, the number of comparisons was  $10 \times 15 = 150$ , where 15 is the number of combinations of 6 methods chosen in sets of 2.

The experiment consisted of a forced-choice pairwise paradigm, in which the

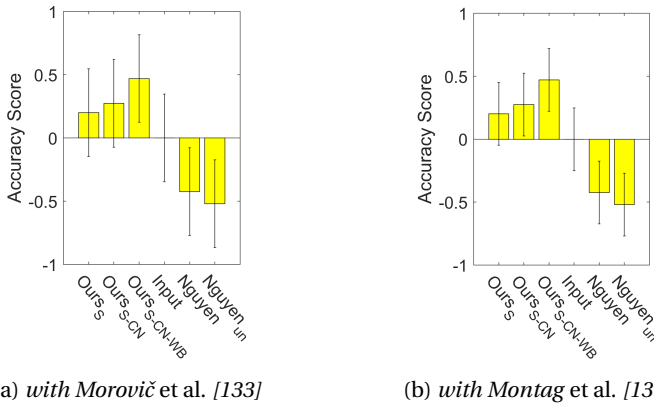


Figure 3.12: Results of the psycho-physical experiment using the Thurstone Case V test. S stands for saliency, CN for color naming, WB for white balance and “un” for unassigned. (a) and (b) shows the results of using two different methods to compute **confidence intervals**. Our method is statistically significantly better than the two versions of [134]. With (b) [130] the statistical significance of our results stands with a larger difference.

groups of images obtained by any two of the methods were randomly shown on the left and right sides of the screen. The experiment was conducted on a DELL P2317H monitor with the following  $x, y$  primaries—red: 0.6513, 0.3383; green: 0.3246, 0.6182; blue: 0.1556, 0.0441; white: 0.3114, 0.3328—with a peak white of 177.65 nits. The display was viewed at a distance of approximately 70 cm so that 40 pixels subtended 1 degree of visual angle. The experiment was conducted in a dark room.

The study consisted of 15 observers. All observers had normal color vision (tested using the Ishihara color blindness test). The observers were asked to select the most color-consistent group of images while penalizing for both artifacts and unnatural colors.

We have analyzed the result of our experiment in terms of the Thurstone Case V Law of Comparative Judgment. Fig. 3.12 presents the results for the whole set of 150 comparisons. For readers unfamiliar with Thurstonian analysis [163], a raw scoring matrix (that records the number of times each of the methods is preferred/not preferred against the others) is recorded. Various assumptions are made that allow the raw scores to be translated into a standardized ( $z$ -score) unit together with confidence intervals. The higher the  $z$ -score, the more a given algorithm is preferred.

In Fig. 3.12, we combine all the observers' results and convert the raw score matrix to the standardized z-score representation and the confidence intervals following the approaches of Morovič *et al.* [133] and Montag *et al.* [130]. The average score is indicated by the yellow bars' top (or bottom). The vertical lines show the 95% confidence intervals. Clearly, Fig. 3.12 shows that our method delivers preferred outputs and, importantly, that both our method with saliency and color naming and our method with the three high-level features are statistically significantly better (at the 95% level) than the two versions from Nguyen *et al.* because the confidence intervals do not overlap.

### 3.5 Applications

#### 3.5.1 Interactive multi-image recoloring

The framework presented in this chapter is a fully working interactive system based on Python and Tkinter (see Fig. 3.13). Using our software, users can view and interact with the image recoloring process. Once the user loads a collection of images, the system automatically initiates the processing and provides visual representations of the source palette for each image, the group palette, and the resulting recolored image collection.

Our system allows users to choose between utilizing the modules described in this chapter and manually selecting colors to manipulate the palettes. For a detailed demonstration of the interface, please refer to the supplemental materials, which include a screen recording.

#### 3.5.2 Example of brochure design

Finally, Fig. 3.14 shows the ability of our method when used to prepare images for use in a brochure that uses an external color palette. We show in (a) the original brochure and in (b) and (c) two brochures in which the images have been modified by our framework using two different color palettes. In this example, the original images lack color consistency among themselves and with the brochure. Using our framework, the final brochure has a more consistent color appearance. In addition, Our method can adapt to different brochure color palettes. This example was computed by directly considering the given external color palette as the group palette of our framework.



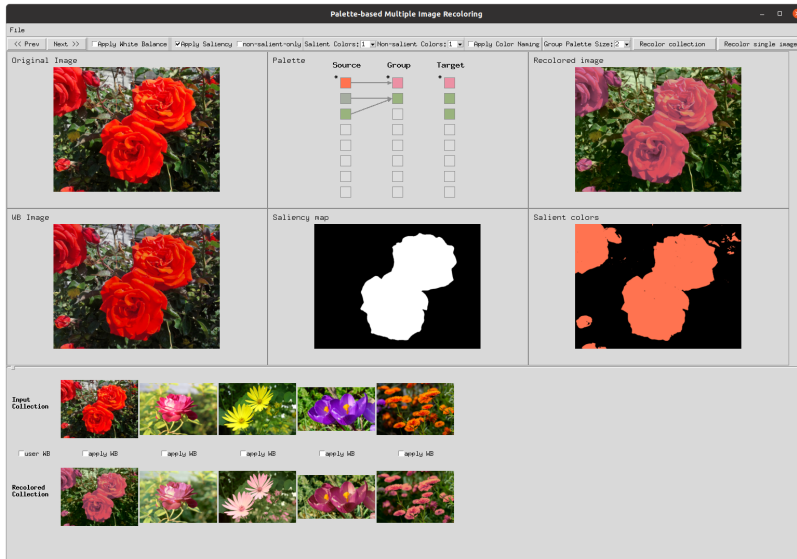


Figure 3.13: Our GUI for interactive palette-based multi-image recoloring.



Figure 3.14: Application of multi-image recoloring with different group palettes on brochure design. (a) shows the original brochure. (b) and (c) show the brochures produced by images recolored with two different group palettes.

### **3.6 Concluding remarks**

We have introduced an interactive framework for achieving multi-image color consistency. Our framework incorporates white balance, saliency, and color naming within a general palette-based recoloring system. The combination of these additional high-level constraints significantly improves overall results and produces recolored collections free of unwanted artifacts. Through qualitative examples and a user study, we have established that our approach surpasses the current state-of-the-art methods in terms of both visual quality and user preference.

We currently mainly focus on utilize saliency for its generalization ability and efficiency. As a binary classification, saliency detection forms the basis for incorporating complex semantic categories while maintaining computational efficiency. Future directions for this research will involve incorporating new modules that go one step further in prioritizing semantic information within the images. One potential avenue is to leverage open-vocabulary semantic segmentation methods [79, 110] to identify and match similar semantic concepts. By integrating these techniques, we can enhance the recoloring process by considering the underlying meaning and context of the image content. This incorporation of semantic-based modules holds promise for further improving the overall quality and coherence of the recolored images.

The nature of our task is to compromise some color diversity of an individual image for better consistency. Additionally, due to the clustering-based color palette extraction, if there are too many different colors in the image, the clustering-based method tends to extract grayish colors in the palette. A potential alternative is to use geometry-based palette extraction methods or add an extra restriction on group palette solving.

# 4 Palette-based color harmonization via color naming\*

## 4.1 Introduction

Color modification is a crucial cornerstone in graphic design, where color coherence [134] and harmony [30] play a vital role in various applications like advertisements and brochures. However, most methods for color manipulation focus on applying color themes to meet specific design requirements, while often overlooking the preservation of the image's natural and harmonious color composition. In this work, we take a different approach by revisiting two widely used color modification methods: palette-based image recoloring and color naming. Specifically, we demonstrate how enhancing an extracted color palette considering a color-naming model leads to more harmonious image colors. This improvement is evident compared to a method explicitly designed for color harmonization.

Palette-based color manipulation allows for intuitive adjustments of different colors, empowering designers to establish specific color schemes for achieving color harmony and enhancing visual appeal [134]. The process begins with palette extraction, where a set of representative colors, known as a color palette [152], is identified from an image. This palette can be manually selected [2, 102] or automatically generated using algorithms like k-means clustering [19, 218] or convex hulls [159, 161]. Once the color palette is obtained, it serves as a reference for modifying the colors in the image. Modifications may include color correction, recoloring [19, 75, 134, 218], or creating color harmonies [160].

Different from image enhancement approaches that aim to enhance colors [31, 145, 177], color harmonization is a technique intended to create balance and coherence among different colors within an image. Color harmonization typically follows specific color schemes to produce visually pleasing compositions. Cohen-Or *et al.* [30] have defined harmonious colors as those that adhere to a pre-defined hue distribution represented by harmonious templates. By mapping the colors in an image to this distribution using defined rules, the resulting image aligns with aesthetic design principles. Following this idea, methods have targeted improving the harmonic template search via predominant hue colors [5, 68, 95, 171] or color histograms [151]. Color harmonization is a common practice used to enhance

---

\*This chapter is based on a publication in IEEE Signal Processing Letters, 2024 [196]

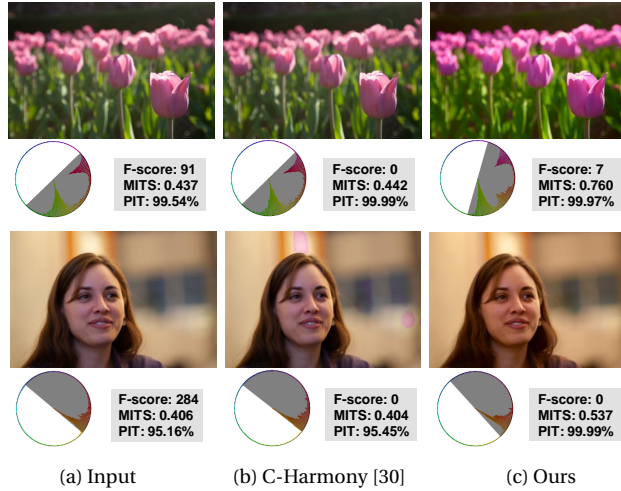


Figure 4.1: Comparison between the results of color harmonization (Cohen-Or *et al.* [30]) and our approach. For each image, the optimal harmonious template (*i.e.* gray areas) on the hue wheel is shown. The collection of colors inside the gray areas is considered to be harmonious. Lower F-score and higher MITS and PIT indicate more harmonious colors. Our proposed method produces harmonized and realistic colors without few artifacts.

visual aesthetics, particularly in fields such as graphic design.

Color naming is the practice of associating names or labels with specific colors. The seminal work by [7] established that basic color terms that are universally recognized: *red, orange, brown, yellow, blue, pink, purple, green, black, gray, white*. Different models [6, 140, 167, 212] explored how to parameterize RGB values into probabilities, which indicate the likelihood that the RGB values belong to each of these color names. Color naming categories are related to the human naming of specific objects, helping to ensure the relationship between content and colors by constraining color names.

It is important to note that both saturation and color distribution play crucial roles in achieving color harmony [30]. For this reason, we introduce the concept of color-name stability as a reference in image color adjustment. The goal is to enhance the image while maintaining its original color names. Our palette-based method achieves this by modifying color distribution to obtain a representative palette with the same color names while enhancing image saturation. This approach, unlike directly increasing overall image saturation, does not compromise

color harmony. We demonstrate how the color-name stability hypothesis within an extracted palette results in an output image with harmonized image colors (see Fig. 4.1). Our experimental results demonstrate that our method can improve image color harmonization aesthetically and statistically.

## 4.2 Related work

### 4.2.1 Color harmonization

C-Harmony [30] is the groundbreaking work for color harmonization. Many approaches are derived from it, performing modifications focusing on specific aspects. For example, to speed up the template searching process of C-Harmony, [68, 95, 151, 171] use the predominant hue color or the color histogram for optimal harmonious template searching. Wan *et al.* presents a component-based pre-harmonization strategy to preserve the hue distribution of the harmonized images [171]. Baveye *et al.* considers saliency when estimating the harmonious template and color mapping to preserve the color of the most attractive visual areas [5]. Yang *et al.* [201] propose a palette harmony score estimation approach based on neural networks. Marino *et al.* [120] apply color harmonization on recoloring augmented reality content according to the real background. These approaches are not deep-learning-based and follow the scheme of searching for the best color palette and mapping the color outside the template to the color inside, while our method harmonizes the image differently. We recolor the image using a pair of source and target palettes. The target palette is sourced from a group of prototype colors selected by the color naming model. In short, our method does not minimize any harmony metric.

### 4.2.2 Color naming

Color naming approaches can be divided into two categories based on the data used: color-naming chips [6, 140] and real-world images [167, 212]. Benavente and Parraga *et al.* [6, 140] developed a parametric model for automatic color naming using labeled color chips, where each color category is represented as a fuzzy set with a parametric membership function. Unlike color-naming chips, which consist of isolated colors, real-world images exhibit a richer and continuous spectrum of colors from pixel to pixel. Color-naming chips, typically under ideal lighting on a color-neutral background, present a stark contrast to the challenges posed by real-world images, which lack a neutral reference color and involve physical variations like shading and different light sources [167]. Van De Weijer *et al.* [167]

explored learning color names from noisy internet images using probabilistic latent semantic analysis. Yu *et al.* [212] expanded the set of color terms by adding 28 new color names and computed highly discriminative color name representations of arbitrary length. While color naming has extensive applications in design, no previous studies have examined its role in color harmonization.

### 4.2.3 Image enhancement

Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further image analysis. Most studies focus on making images more visually appealing by adjusting various parameters, like brightness, sharpness, saturation, and *etc.* Due to the powerful capabilities of deep learning, current image enhancement methods are mainly based on supervised learning to minimize the difference between expert-retouched images. Meanwhile, they also incorporate some classical traditional image processing methods as references, such as histogram equalization [145], color curve adjustments [132], filter-based enhancement [131], 3D Look-Up Tables [177, 202, 214], *etc.* Rahman *et al.* [145] introduces an improved histogram equalization method to enhance low-contrast images. The method only focuses on gray-tone images and does not consider the relationship between image content and color. CURL [132] learns to adjust global image properties such as color, saturation, and luminance using human-interpretable image enhancement curves. DeepLPF [131] learns to automatically enhance images with learned spatially local filters of three different types. 3D Look-Up Tables (LUTs) provide a way to apply complex color transformations consistently across images, by mapping input color values to desired output color values in the three-dimensional color space. The core idea of [177, 202, 214] is to learn a set of LUTs as a basis, and they achieve fast transformations by predicting a set of weights through a network using downsampled low-resolution images and then enhancing the images with the weighted basic LUTs. These approaches focus more on how to produce visually appealing images rather than on making the overall color distribution of the image conform to specific color design principles.

## 4.3 Methodology

Our approach involves four key steps: color prototype generation with color naming (Section 4.3.1), color palette extraction (Section 4.3.2), color matching (Section 4.3.3), and palette-based image recoloring (Section 4.3.4). Color prototype generation with color naming and color matching are the most crucial, as they involve deriving the new set of colors that will form the “basis” of the image. Our method

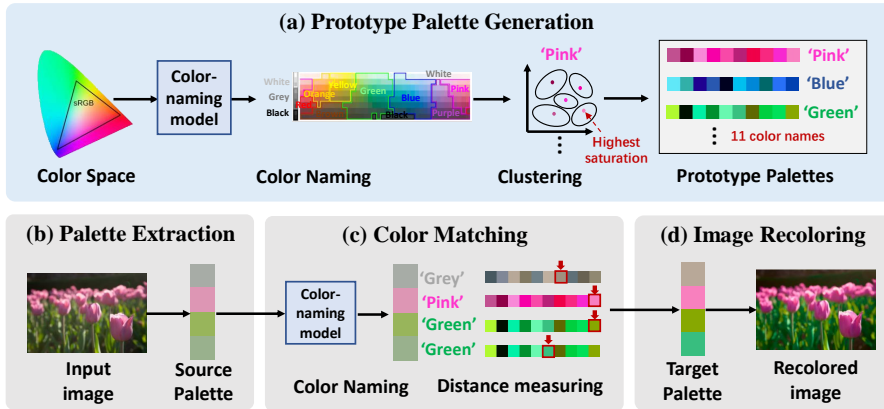


Figure 4.2: The proposed palette-based color modification framework. All colors in the color space are categorized into 11 classes by a color-naming model, and colors with the same color name are clustered. The color with the highest saturation in each cluster is selected as a color in the prototype palettes. Given an input image, the source palette of the image is extracted. Then, the color name of each color in the source palette is identified by the same color-naming model. By searching for the color with the smallest difference from the source color in the prototype palette with the same color name, the target color is obtained. Finally, the image is recolored based on the target palette to achieve the color modification.

mainly focuses on these steps by identifying a suitable target palette, ensuring that the recolored image based on this palette exhibits increased saturation while minimizing hue shifts. Fig. 4.2 shows an overview of our approach.

### 4.3.1 Color prototypes generation with color naming

The objective of this step is to generate the set of candidate colors to be the prototype palettes. Our method is based on keeping the names of the colors in the palette, and therefore, we enforce that the names of the colors in the source palette are unchanged in the final target palette. To identify the color name, we apply a color-naming model [167], which assigns 11 color probabilities to any input value in the *sRGB* color space. These probabilities represent the extent to which the RGB values can be named with a specific term. We first run the color-naming model for the possible values in the whole *sRGB* color space, where  $R, G, B \in [0, 255]$ , with a  $8 \times 8 \times 8$  grid. In this way, the color space is divided into 11 different parts, one per color term. Then, k-means clustering is applied to these color values at each part,

resulting in  $n$  distinct partitions per color name. Due to the varying sizes of the regions covered by different color terms, we select the number of candidate colors for each color term based on the area they cover. Specifically, we select a basic 10 colors for each term. Besides, among all the 11 color names, “green”, “blue” and “purple” have the highest counts, so we add an extra 5 colors for each of these 3 color names. Additionally, to ensure consistency in human skin tones in portraits, we also add 5 more colors for “red” and “pink”.

Up to this point, we have ensured that the colors in each prototype palette are of homogeneous color term. Next, to guarantee a large saturation value from each constructed partition, we select the color with the highest saturation in the cluster as the candidate color for the prototype. Therefore, we end up with a prototype of candidate colors per color term  $P^p = \{c_1^p, c_2^p, \dots, c_n^p\}$ , which have both stable color names and high saturation.

### 4.3.2 Color palette extraction

For an input image, we extract a color palette  $P^s = \{c_1^s, c_2^s, \dots, c_l^s\}$  that represents the primary colors of the image. To reduce computational complexity, we first compute the color histogram of the image and extract the color palette  $P^s$  by k-means clustering based on this histogram, where the cluster centers are selected as the colors in the palette. The color histogram is computed on  $ab$  channels in the  $Lab$  color space to avoid the influence of regions with excessively low or high lightness on the clustering results. In most cases, the number of the cluster center  $l$  is typically set to a fixed value, with  $l = 5$  being the most common choice in graphic design. However, since the richness of colors varies among different images, the optimal number of colors may vary for each image. To avoid an excessive number of similar colors in the palette, we determine the optimal number of clustering centers based on the percentage of explained variance [134]. For each value of  $l$ , ranging from 2 to 7, we calculate the within-group distortion. This involves the summation of the distance  $d_l$  of each point in the cluster to its center. The total distortion  $d_1$  is the summation of distances between each color point and the overall mean color. We select the optimal value of  $l$  when  $\frac{d_1 - d_l}{d_1 - d_7} > \gamma$ ,  $\gamma = 0.93$ .

### 4.3.3 Color matching

This stage defines the target palette by selecting the best representative among the candidate colors for each color in the source color palette. To this end, given the RGB values of the source palette color, we run the color naming model and select possible colors for the target palette those in the prototype that share the same color name. Then, among all these colors, we choose the one that has the



smallest difference to the source color as the target one. Here, we compute the RGB Euclidean distance for color distance measurement. Note that as there might be some source colors having similar probabilities of belonging to multiple color names, a relaxed search space is applied for the target color search. More in detail, instead of only getting the prototype colors for a single color term, we select the prototype colors for any color term that 1) has a probability greater than 15%, and 2) is among the top three color names in terms of probability.

#### 4.3.4 Palette-based image recoloring

Finally, we recolor the input image with the target palette. With the matched source and the target colors, color mapping is executed by inverse distance weighting, where larger weights are assigned to closer colors [134]. More specifically, given the pixel value of the input image  $I^s$ , source palette  $P^s = \{c_1^s, c_2^s, \dots, c_l^s\}$  and the corresponding target palette  $P^t = \{c_1^t, c_2^t, \dots, c_l^t\}$ , the pixel in the recolored image  $I^t$  with coordinate  $(x, y)$  is determined by

$$I^t(x, y) = I^s(x, y) + \sum_{i=1}^l w_i (c_i^s - c_i^t), \quad (4.1)$$

where the weight factor  $w_i(x, y)$  is

$$w_i = \frac{1}{\sum_{i=1}^l |I^s(x, y) - c_i^s|^2 + \varepsilon}, \quad \varepsilon = 10^{-4}. \quad (4.2)$$

The recolored image has higher saturation and a more distinct color for color naming.

## 4.4 Experiments

### 4.4.1 Experimental setup

**Dataset.** *MIT-Adobe FiveK (FiveK)* [17] contains 5,000 raw images taken with DSLR cameras, covering a broad range of scenes, subjects, and lighting conditions. This dataset provides raw image data in .DNG format, while most image enhancement or color retouching tasks are typically performed on 8-bit sRGB images. Therefore, preprocessing of the original raw format data is necessary. In the main paper, we present the results on Camera Raw version of FiveK. We used the Camera Raw tool in Photoshop, which preprocesses image colors based on the metadata of the raw images, automatically selecting configuration files such as “ACR 4.4”, “ACR 4.6”, or

“Adobe Standard” depending on the camera model. This preprocessing method produces images with more vibrant colors, consistent with the sRGB images saved directly from camera shots. While most current color enhancement methods are based on pair learning, and they use DPE version [24] of FiveK in their experiments. The raw images are resized and converted to the sRGB space through Adobe Lightroom without additional adjustments, resulting in images that appear grayish color. For fair comparison with deep learning based approaches, we present the results of the test set—last 500 images. *Kodak* [81] contains 24 8-bit sRGB images. The test images of FiveK and Kodak are resized with the shorter side set to 512 pixels. **Portrait Photo Retouching (PPR10K)** [98] contains 11,161 portrait photos. We use the validation split (the last 2,286 images) of the source 360p 16bit sRGB images, and convert them to 8 bits to fit the color naming model.

**Competing methods.** We compare our results versus recent deep-learning approaches, CURL [132], DeepUPE [176], 3DLUT [214], and AdaInt [202] that aim to minimize the intent of a photographer, for FiveK, we use the pretrained models target to the Expert C in sRGB color space, for PPR10K, we use the pretrained models target to the Expert A in sRGB color space; an unsupervised deep-learning-based image enhancement approach, CLIP-LIT [99]; a traditional image enhancement method, SRIE [46]; and the color harmonization approach, C-Harmony [30].

**Image Quality Metrics.** Our image modification paradigm does not have any real ground-truth, since our goal is not to approximate the user intent, *e.g.* Expert C in FiveK. We quantitatively evaluate the method by two non-reference image quality assessment metrics: NIQE [128] and BRISQUE [126]. These two metrics assess the perceptual naturalness of images.

**Color Harmony Metrics.** We evaluate color harmonious degree with three metrics: F-score [30], Mean Inside-Template Saturation (MITS), and Percentage of Inside-Template pixels (PIT). These metrics measures color harmony of an image  $I$  by comparing hue distribution  $H$  and saturation  $S$  with respect to a certain harmonious scheme  $(m, \alpha)$ , where  $T_m$  is the template and  $\alpha$  is the associated orientation.  $x \in I_{in}$  and  $x \in I_{out}$  indicates the pixels inside and outside the scheme, respectively. F-score is calculated as:

$$F(I, (m, \alpha)) = \sum_{x \in I_{out}} \|H(x) - E_{T_m(\alpha)}(x)\| \cdot S(x), \quad (4.3)$$

where  $E_{T_m(\alpha)}(x)$  indicates the template border hue of  $T_m(\alpha)$  that is closest to the hue of the pixel of the image.  $\|H(x) - E_{T_m(\alpha)}(x)\|$  denotes the hue distance from a pixel to the nearest boundary of the harmonic scheme  $T_m(\alpha)$ , measured in radians, on the hue wheel. A smaller F-score indicates that there are fewer pixels with hues outside the template, and that those out-of-template pixels have lower saturation.

With the same optimal harmonious scheme found by F-score, we compute PIT that quantifies the proportion of pixels within the template:

$$PIT(I) = \frac{N(x \in I_{in})}{N(x \in I)}, \quad (4.4)$$

where  $N()$  denotes the number of pixels that meet the specified criteria.

MITs calculates the average saturation of pixels within the template:

$$MITs(I) = \frac{\sum_{x \in I_{in}} S(x)}{N(x \in I_{in})}. \quad (4.5)$$

Larger PIT and MITs suggests more pixels with higher saturation values inside the optimal template, respectively, therefore complementing the F-score metric.

#### 4.4.2 Quantitative results

Table 4.1 and Table 4.2 look at the results for three different datasets. It shows that for traditional blind quality assesment metrics (NIQE, BRISQUE), both our method and C-Harmony are competitive against traditional state-of-the-art enhancement models. It is important to remember that our goal is to not only obtain an image that is enhanced but also better in terms of its color scheme. For the ability of color harmonizing, our method is the best for both MITs and PIT and second for F-score in these datasets, which indicates that our results exhibit a distribution that aligns more closely with the harmonic template and possess higher saturation. Also, we should remark that C-Harmony [30] is optimized to minimize the F-score, the only one of the three harmonization metrics where it outperforms us. Also, since the image enhancement methods (SRIE, CURL, DeepUPE, CLIP-LIT, 3DLUT, and AdaInt) do not target to optimize color harmony, their performance on PIT is somehow lacking (around 90%), while our method and C-Harmony are over 98%.

#### 4.4.3 Ablation study

In this subsection, we conduct the ablation study on both the Camera Raw version and DPE version of the FiveK, including the ablation of prototype color numbers (Table 4.3) and the ablation of color similarity measures (Table 4.4). The results follow the same trend for the two versions.

##### Number of prototype colors

The prototype palette determines the richness of colors in the recolored image. We therefore compare the differences in image quality and color harmonious score

## Chapter 4. Palette-based color harmonization via color naming

Table 4.1: Comparison of image quality and harmony score on the camera raw version [17] and DPE version [111] of FiveK. ■, ■ indicates smaller and larger values are better, respectively.) **Bold**, *italic* indicate the best and second best results.

Data	FiveK (Camera Raw)					FiveK (DPE)				
	NIQE	BRISQUE	F-score	MITS	PIT(%)	NIQE	BRISQUE	F-score	MITS	PIT(%)
Expert C	-	-	-	-	-	<b>3.35</b>	37.21	3465	0.393	87.18
SRIE [46]	<b>3.32</b>	41.09	1477	0.368	90.97	3.48	38.13	1466	0.280	90.37
CURL [132]	3.46	41.85	2841	0.440	87.93	3.43	<b>35.61</b>	2500	0.404	89.66
DeepUPE [176]	3.40	40.88	1502	0.399	91.38	3.45	36.96	1374	0.334	91.42
CLIP-LIT [99]	3.36	44.18	1489	0.366	91.03	3.37	40.47	1466	0.280	90.39
3DLUT [214]	3.48	41.98	2406	0.453	88.90	3.44	36.94	2637	<i>0.428</i>	90.22
AdaInt [202]	3.44	42.50	2379	<i>0.455</i>	89.21	3.44	36.93	2574	0.426	90.14
C-Harmony [30]	3.52	<b>38.17</b>	<b>58</b>	0.368	<i>98.39</i>	3.80	<i>36.51</i>	<b>42</b>	0.280	<i>98.21</i>
Ours	3.56	<i>39.51</i>	<i>312</i>	<b>0.558</b>	<b>98.59</b>	3.79	37.17	<i>119</i>	<b>0.506</b>	<b>99.34</b>

Table 4.2: Comparison of image quality and harmony score on PPR10K [98] and Kodak [81]. ■, ■ indicates smaller and larger values are better, respectively.) **Bold**, *italic* indicate the best and second best results.

Data	PPR10K					Kodak				
	NIQE	BRISQUE	F-score	MITS	PIT(%)	NIQE	BRISQUE	F-score	MITS	PIT(%)
SRIE [46]	<i>4.00</i>	<i>43.59</i>	412	0.372	93.13	<b>2.90</b>	49.36	1104	0.320	91.86
CURL [132]	4.24	46.32	928	0.419	90.30	3.06	51.58	1909	0.442	89.00
DeepUPE [176]	4.09	45.23	318	0.408	94.55	3.08	51.93	895	0.371	92.52
CLIP-LIT [99]	<b>3.72</b>	46.14	414	0.370	93.03	3.24	55.23	1100	0.319	91.77
3DLUT [214]	4.29	46.53	688	0.432	91.08	3.06	50.00	1674	<i>0.465</i>	88.70
AdaInt [202]	4.20	45.60	671	<i>0.446</i>	90.48	3.07	49.87	1709	0.463	88.47
C-Harmony [30]	4.07	<b>42.23</b>	<b>14</b>	0.375	<i>98.20</i>	3.06	<i>48.95</i>	<b>8</b>	0.316	<i>98.44</i>
Ours	4.26	44.46	<i>104</i>	<b>0.509</b>	<b>98.50</b>	3.05	<b>48.72</b>	<i>80</i>	<b>0.473</b>	<b>99.28</b>

when varying the number of prototype colors, from 5 to 50 for each color name. As the number of prototype colors decreases, the color harmony-related metrics of the images improve. This is because, with a limited color palette, the color range of the recolored images is constrained, making it easier to map different source colors to the same target color. With 10 prototype colors for each color name and 5 additional colors to the largest area ones, most of the harmony metrics showed improvement. Moreover, these extra colors help reduce hue shifts to some extent, particularly in the case of skin tones.

Table 4.3: Ablation of the number of prototype colors.

Number	Names	FiveK (Camera Raw)					FiveK (DPE)				
		NIQE	BRISQUE	F-score	MITS	PIT(%)	NIQE	BRISQUE	F-score	MITS	PIT(%)
5	top1	3.573	39.374	987	<b>0.643</b>	97.01	<b>3.760</b>	37.118	677	<b>0.605</b>	97.79
10	top1	3.564	39.668	890	0.608	96.95	<b>3.760</b>	37.124	476	0.560	98.00
15	top1	3.559	38.842	1278	0.549	93.11	3.775	37.049	946	0.472	94.23
20	top1	3.545	38.905	1699	0.530	91.87	3.771	37.015	1290	0.451	92.93
50	top1	<b>3.531</b>	<b>38.654</b>	1679	0.458	92.15	3.788	36.998	1409	0.370	92.62
10+5	top1	3.568	39.608	427	0.602	98.11	3.775	<b>36.950</b>	265	0.559	98.75
10+5	top2	3.559	39.508	318	0.563	98.56	3.797	37.184	121	0.511	<b>99.34</b>
10+5	top3	3.556	39.510	<b>312</b>	0.558	<b>98.59</b>	3.793	37.166	<b>119</b>	0.506	<b>99.34</b>

Table 4.4: Ablation of color similarity measures.

Similarity	FiveK (Camera Raw)					FiveK (DPE)				
	NIQE	BRISQUE	F-score	MITS	PIT(%)	NIQE	BRISQUE	F-score	MITS	PIT(%)
Angular	3.541	<b>38.83</b>	980	<b>0.502</b>	95.14	<b>3.788</b>	37.229	719	<b>0.419</b>	95.52
Probability	<b>3.538</b>	38.90	893	0.510	95.25	<b>3.788</b>	37.216	627	0.435	96.01
Euclidean	3.556	39.51	<b>312</b>	<b>0.558</b>	<b>98.59</b>	3.793	<b>37.166</b>	<b>119</b>	<b>0.506</b>	<b>99.34</b>

### Color similarity measures

We also compared different color similarity measurement methods, including Euclidean distance, angular distance [43], and color-naming probability similarity [167] for color matching. The first two directly measure color distance in the color space, while the latter assesses similarity in color-naming space by calculating cross-entropy between color probability distributions for different color names. Table 4.4 shows that the Euclidean distance-based method outperforms others in color harmony-related metrics. However, this approach may occasionally alters the image’s warm or cool tones (see Fig. 4.3), something that is avoided by using the angular distance.

### Color-naming models

We compare three different color naming methods, Párraga *et al.* [140], Van De Weijer *et al.* [167] and Yu *et al.* [212]. As shown in Fig. 4.4, the overall color name distributions of the three methods are similar. The main difference between the three methods lies in the probability distribution of colors belonging to different color names (Fig. 4.5), where [140] yields sharp probabilities, while [167] and [212] produces a soft distribution. Therefore, when applying [167] and [212] on determining color names, we adopt the approach mentioned in Section II-C to expand the search range of the target palette appropriately. Table 4.5 shows the results

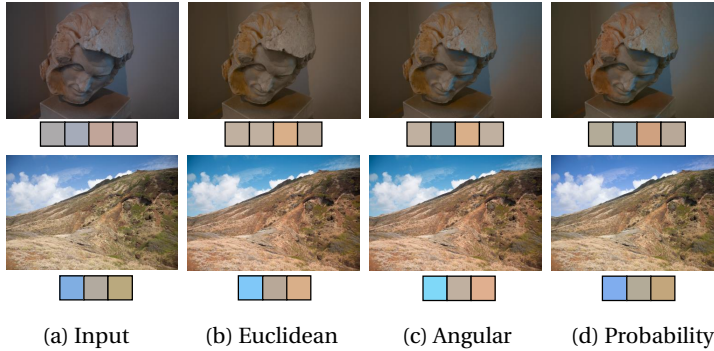


Figure 4.3: Comparisons between the results of applying different color distance measures to search for the target palette.

Table 4.5: Ablation of different color name on FiveK (Camera Raw).

Methods	NIQE	BRISQUE	F-score	MITS	PIT(%)
Ours (w [140])	3.56	<b>38.77</b>	1410	0.51	91.44
Ours (w [167])	3.56	39.51	<b>276</b>	<b>0.53</b>	<b>98.11</b>
Ours (w [212])	<b>3.55</b>	39.23	817	0.48	95.48

of applying different color naming methods. [167] and [212] have better harmony scores than [140]. The difference between these kinds of approaches is that [167] and [212] learn from real-world images, while [140] learn color names from labeled color chips. Color naming chips under ideal lighting on a color-neutral background greatly differ from the challenge of color naming in images coming from real-world applications without a neutral reference color and with physical variations such as shading effects and different light sources [167]. Color naming chips are isolated colors, while the real-world image covers richer colors that continuously change from pixel to pixel.

### Color space for color matching

The main motivation for searching in the *RGB* space is that the color naming models operate in the *RGB* space. So, we directly compute prototype colors and search for matching colors in the *RGB* space, thus avoiding implementing further color space conversions. Also, searching in the *RGB* space also shows quantitative superiority.

Regarding effectiveness, we present the results of performing prototype color computing and color search in the *Lab* color space and the *RGB* color space, re-

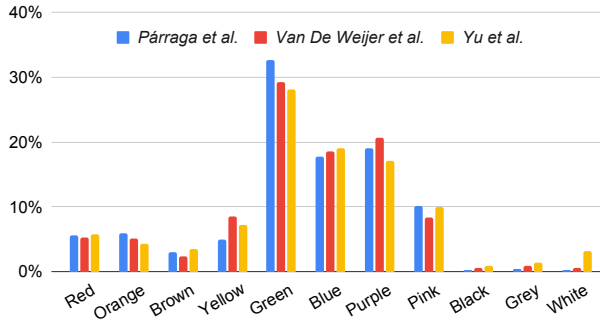


Figure 4.4: The distribution of the amount of colors for each color name with different color naming methods, Párraga *et al.* [140] Van De Weijer *et al.* [167] and Yu *et al.* [212].

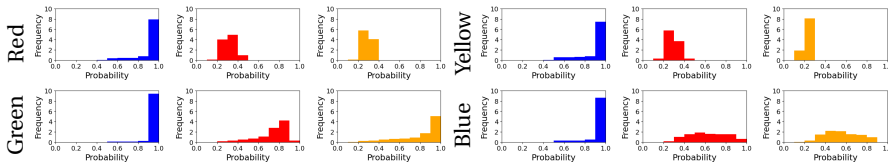


Figure 4.5: The probability distributions of each color name. Blue, red and orange for Párraga *et al.* [140] Van De Weijer *et al.* [167] and Yu *et al.* [212], respectively.

spectively, in Table 4.6. Performing both clustering and search in the *RGB* color space shows the best results in harmonic scores, while searching in the *ab* channels of the *Lab* space results in better image quality.

Regarding efficiency, after solving for the prototype palette, we save both the *RGB* and the *Lab* prototype colors as a dictionary for subsequent color search and matching of each image. The color search is based on the source palette, which contains only 2 to 7 different colors. In our workflow, we only need to perform the color transformation once for each color in the palette: we transform the source color from the *Lab* space to the *RGB* space to compute its color name and find its match in the corresponding prototype palette. The time required for these conversions is almost negligible compared to pixel-wise color mapping.

Table 4.6: Ablation of color searching space on FiveK (Camera Raw).

clustering	search	NIQE	BRISQUE	F-score	MITS	PIT(%)
<i>RGB</i>	<i>RGB</i>	3.56	39.51	<b>276</b>	<b>0.53</b>	<b>98.11</b>
<i>RGB</i>	<i>Lab (ab)</i>	<b>3.52</b>	<b>38.48</b>	1395	0.40	91.84
<i>Lab (ab)</i>	<i>Lab (ab)</i>	3.54	38.63	1063	0.47	93.17

#### 4.4.4 Speed

We implement C-harmony [30] and our method with Python 3.8. We compared the computational speed of C-harmony [30] and our method on the FiveK testset with Ubuntu 20.22 with Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz. The input images are resized with the shorter side set to 512 pixels. For C-harmony, it takes 37 seconds per frame on average to find the optimal template and angle, and another 0.5 seconds per frame to modify the image color based on the optimal template. Our method takes around 0.6 seconds per frame to complete the whole process. Table 4.7 shows the processing time required for each stage of our method. The first stage, “Color Prototypes Generation with Color Naming” needs to be computed only once, and then the prototype palette can be saved and used for all the images. While stages “Color Palette Extraction”, “Color Matching”, and “Palette-based Image Recoloring” need to be processed separately for each image. Currently, we have not introduced parallel computation for acceleration.

Table 4.7: Average run time of our method.

Stage	Step	Time (s)
pre-computation	1) Prototypes Generation	7.532
per-image processing	2) Palette Extraction	0.397
	3) Color Matching	0.002
	4) Image Recoloring	0.147
Average	step 2)+3)+4)	0.547

#### 4.4.5 Qualitative results

Compared to other methods, our approach does not produce artifacts in regions with high brightness, which happens in CURL (see Fig. 4.6 (c)). Additionally, in most cases, our approach aims to increase image saturation while minimizing significant hue shifts, thus avoiding the unnatural color changes that happen in color harmonization methods (in Fig. 4.6 (e)). CLIP-LIT [99] tends to increase the



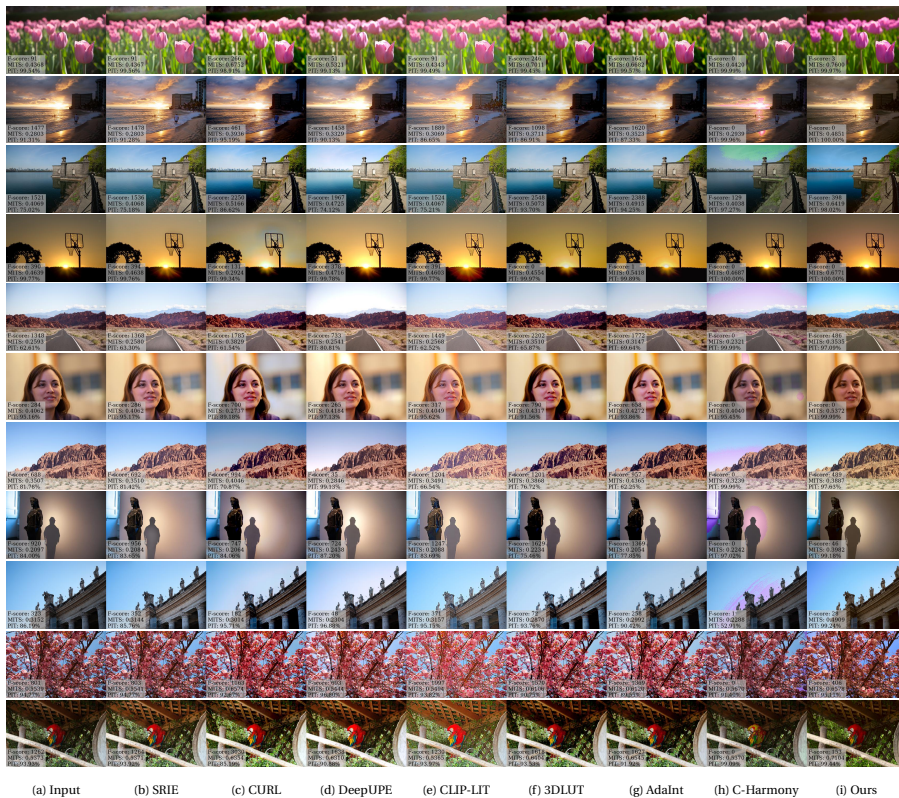


Figure 4.6: Comparison of images against other methods on FiveK dataset (Camera Raw). Our method produce more saturated colors while preventing unnatural color shifts. Most samples processed by C-Harmony [30] have noticeable artifacts.

brightness of the image, so it produces over-bright images. The images from PPR10K dataset are shown in Fig. 4.7. DeepUPE [176] tends to increase the brightness of the image, so sometimes it may produce an image that is over-bright, for instance, the portrait of the boy in Fig. 4.7. Among them, ours can produce images with smooth gradient change. CURL [132], DeepUPE [176], 3DLUT [214], and AdaInt [202] and our approach can produce naturally enhanced images, while ours have higher harmonic scores.

### 4.4.6 Application on multi-image recoloring

In Chapter 3, we propose a multi-image recoloring method that improves the color consistency through recoloring a group of images with a group palette. The color palette of individual image and the common group palette of the input image group are computed through color clustering. However, such palette extraction methods often suffer from a common issue: when there are many colors in the image, the color in the extracted palette tends to an average solution, *i.e.* the color is grayer, resulting in recolored images appearing less saturated than the original images and diminishing their visual appeal. Here, we introduce a color harmonization method proposed in this chapter to mitigate the inherent drawbacks of cluster-based palette extraction methods. Specifically, we map the obtained group palette to the extracted prototype palettes, thereby enhancing color saturation. Certainly, this approach can also be applied to each individual image or its respective palette. However, to ensure color consistency across a set of images and reduce computational overhead, we opt to directly adjust the group palette and then apply the optimized group palette for recoloring each image. As illustrated in the Fig. 4.8, compared to not performing color mapping, applying the color harmonization method proposed in this chapter effectively enhances image saturation and alleviates the issue of image desaturation.

### 4.4.7 Limitations

Although we have taken some measures to avoid color shifts, *e.g.*, increasing the number of prototype colors for some color names, expanding the color searching space, the recolored images will still have some color shifts, especially in portraits. This is because of the smaller range of natural skin colors, and the human perception is more sensitive to such kind of objects, making colors outside the natural range easily noticeable, and our method exacerbates this problem by increasing the saturation of the image (see Fig. 4.9(a)). This is a common problem of the approaches that edit image colors without considering image semantic information.

Another problem is the decrease in color diversity. It usually occurs when there are several similar colors in the source palette, and the prototype color matched with these colors is the same one (see Fig. 4.9(b)).

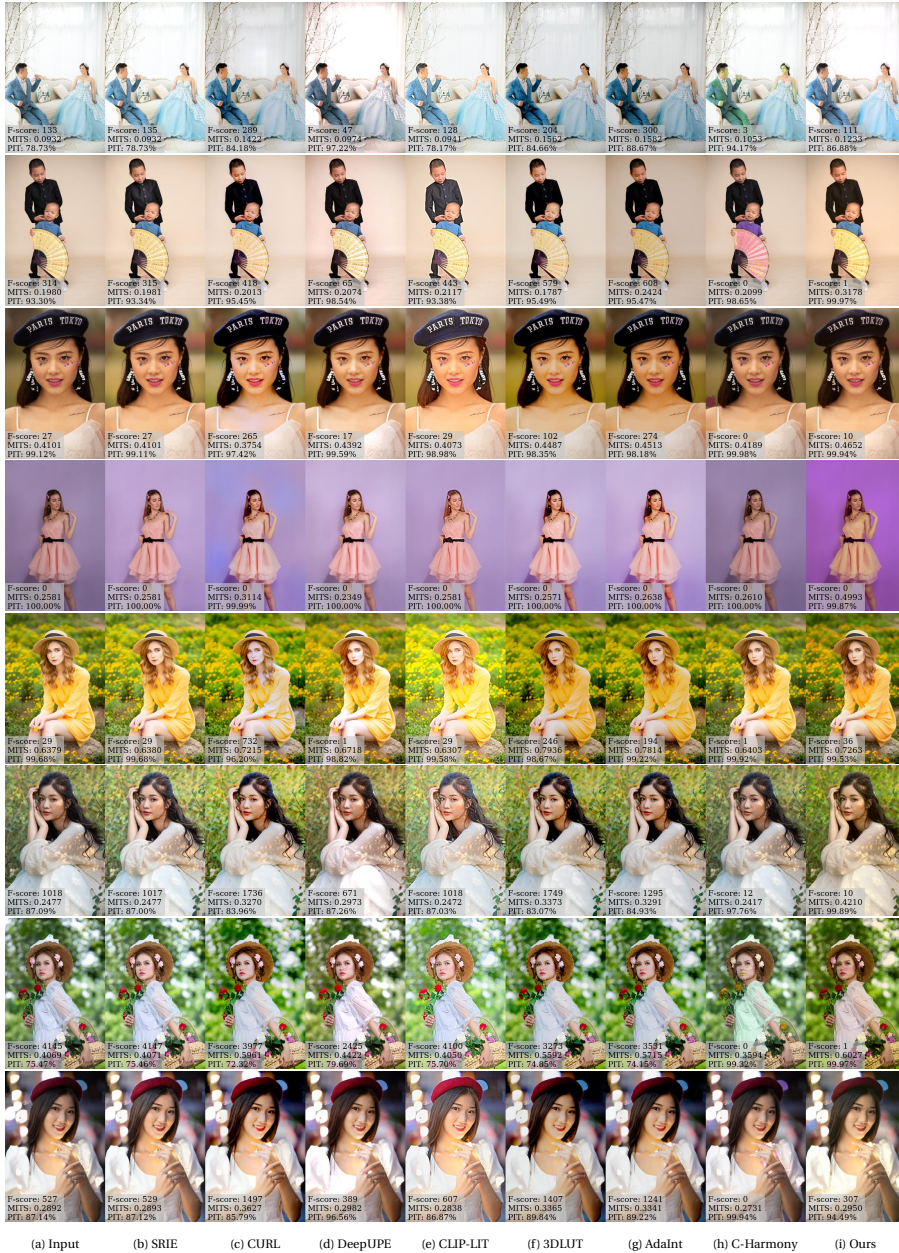


Figure 4.7: Comparison of images against other methods on PPR10K dataset.



Figure 4.8: An example of application on multi-image recoloring. The images recolored with the mapped group palette can effectively increase the saturation and keep the color consistency among multiple images.

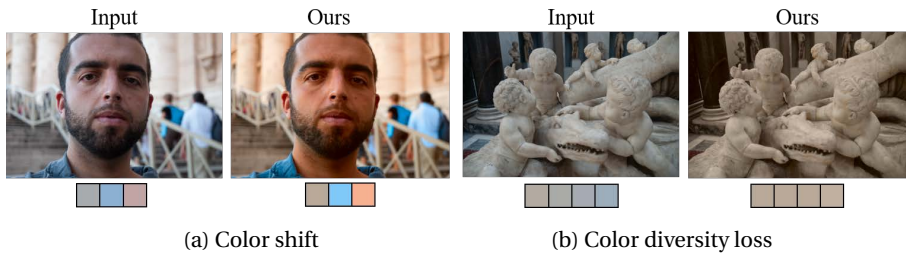


Figure 4.9: Examples of failure cases. (a) shows the output image with unnatural skin tone, (b) shows the output image losing color diversity.

## 4.5 Concluding remarks

In this chapter, we propose an image color modification method based on palette. Our method uses color name as a reference to increase the saturation without compromising the perceptual color name of the original colors in the image. Our method is particularly well-suited for image modification in graphic design, where adhering to a color scheme becomes paramount. The experiments demonstrate the generalization ability of our approach.

The perception of color names is a complex issue, since a color's name is not only related to its hue, but is also greatly influenced by other factors, such as saturation, since all the color goes to either gray, white, or black when adjusting saturation or value to an extreme value. So in this chapter, we only map the color in  $ab$  channel in  $Lab$  color space, and keep the lightness channel always unchanged.

For image quality evaluation, most of the metrics focus on the image quality under different levels of distortion, while there are almost no dedicated models or metrics available to evaluate the perceptual quality and realism of image colors, and we believe that this is crucial for image color optimization. Therefore, the quality assessment metrics especially for fine-grained perceptual color naturalness is a possible direction.



# 5 Burst perception-distortion tradeoff: analysis and evaluation\*

## 5.1 Introduction

Image restoration aims, given an image that has experienced some degradation process, to restore it to obtain the original image. This problem has been widely studied in the literature for many years and relates to several sub-problems, such as denoising, super-resolution, deblurring, *etc*. Recently, the trend has moved towards burst image restoration. Burst is a sequence of images captured in rapid succession. The main reason for this shift is the current ubiquity of smartphones, since these devices can easily acquire this sequential data and process it to produce better-quality images. Burst image restoration has the advantage that multiple frames provide complementary information to the reference one, leading to higher resolution [11, 12, 39, 117, 118, 188], lower noise level [12, 124], and higher dynamic range [56], while also introducing uncertainty caused by motion or camera shake [10]. This misalignment problem introduced by multiple images may lead to restored images with ghost artifacts, and blurry [199]. Recent works [11, 12] explicitly align burst images by estimating optical flows [157], or implicitly by deformable convolutions [39, 117, 118]. In practice, even after these alignment methods, two images are rarely perfectly aligned due to the degradation and the appearance of artifacts.

The evaluation of image restoration is generally carried out from two aspects: the perceptual quality and the distortion. Blau *et al.* [13] first characterized the Perception-Distortion (P-D) Tradeoff in single image restoration. More specifically, they proved that distortion and perceptual quality are at odds with each other so that no image restoration algorithm can optimize the two indicators to the best at the same time in practice. The P-D curve comprehensively shows the upper bound and range of continuous changes of two types of evaluation criteria. The generative-adversarial-nets (GANs) provide a principled way to approach the P-D bound by varying the hyperparameter between distortion loss and perception loss, thus producing estimators along the P-D curve, and therefore obtaining the P-D tradeoff curve. In [45, 200] authors prove that the P-D curve can be acquired

---

\*This chapter is based on a publication in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022) [195]

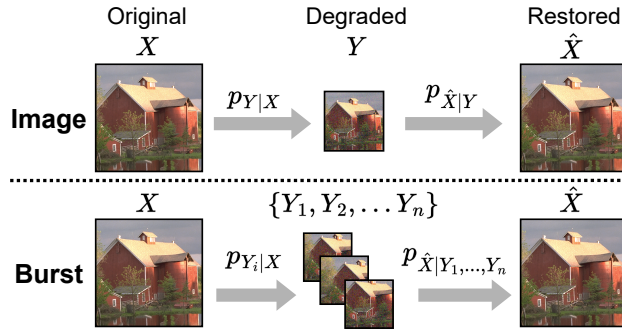


Figure 5.1: **Single Image Restoration** versus **Burst Restoration**. Burst introduces the problem of misalignment between images in the burst. See details in Section 2.

by linear interpolation between two models, which greatly simplifies the steps to obtain the P-D curve. Similar tradeoffs are also proved existing in classification [108] and image compression [14]. However, these works only focus on single-image tasks, and the perception-distortion tradeoff in the case of multiple images has not been studied yet.

Our work studies the perception-distortion tradeoff from multiple images, thus generalizing the case of a single image. In particular, we focus on the case of burst image restoration with relatively stable noise and camera shaking. Through the analysis, we found that using more images does not always lead to better reconstruction quality due to the misalignment between each image. The optimal burst length (*i.e.* the number of images in a burst) for restoration depends on the shake and noise levels.

Burst image restoration (BIR) enables higher quality images without resorting to better imaging equipment. In addition to low distortion, current algorithms also pay attention to the perceptual quality of the restored images. Unlike single-frame image restoration, bursts have an additional time dimension. The motivation of this chapter is to understand how temporal information and, in particular, temporal misalignment due to camera shake, will affect the restoration quality, and how this impact affects perception and distortion, and their tradeoff.

To the best of our knowledge, despite the progression of BIR algorithms in recent years, there is no work that analyses multi-frame restoration from the perspective of P-D tradeoff. Our work verifies that the P-D tradeoff still exists when introducing temporal information, and misalignment will worsen both perception and distortion. In addition, our analysis provides a reference to the design of multi-frame restoration algorithms and the potential shooting strategy. Our results



show that longer bursts (*i.e.* more images) do not always lead to higher restoration quality, since misalignment will make the restoration result worse with more frames. Thus, the key of multi-frame restoration lies in the inter-frame alignment method. Furthermore, bursts provide a suitable starting point to study more complicated sequences such as videos, and thus, *our theory, analysis, and evaluation method can also be extended to more general video restoration scenarios.*

In summary, the contribution is threefold:

- We propose the Burst Perception-Distortion Tradeoff by introducing multiple-frame information.
- We propose AUR as a new method for multi-frame restoration evaluation, which comprehensively reflects the perception and distortion quality of the restored image.
- We analyse the Burst P-D tradeoff under the influence of image noise and shake, and found the effect of inter-frame misalignment on burst restoration.

And two main conclusions are drawn:

- When all the frame in the sequence are perfectly aligned, and the noise level in each frame is lower than the signal itself, using more frames for burst restoration lead to better reconstruction quality.
- When the frames are not well-aligned, there exist an optimal frame number for restoration depending on the content of the image, noise level and displacement level.

## 5.2 Related work

### 5.2.1 Burst image restoration

Various restoration tasks can benefit from burst data, such as denoising [12, 124], based on the assumption that the noise in each frame is independent. If the images in a burst are taken with different exposure time, the images can be merged for HDR imaging [33, 56, 186]. Burst super-resolution [11, 12, 28, 39, 84, 117, 118, 122] has received a lot of attention recently, since [188] has demonstrated that the subtle misalignment between each frame can provide multiple aliased samplings of the underlying scene. Burst restoration also provides a more practical way to alleviate the problems of insufficient dynamic range and high-resolution textures due to limited camera aperture and sensor size [188]. However, the main problem needs to

be solved is the misalignment between the images caused by natural hand motion [10]. Although a certain degree of spatial shift is beneficial for super-resolution, when the shift is large, the restoration results may appear blurry or ghost artifacts [199]. Recent works [11, 12] conduct alignment explicitly by estimating optical flows [157], or implicitly by deformable convolutions [39, 84, 117, 118]. Although the flow estimation methods are adopted, some artifacts still appear in practical applications, since the displacement between some repeating textures is difficult to be estimated accurately.

### 5.2.2 The Perception Distortion (P-D) tradeoff

In image restoration tasks, the tradeoff is proved existing between the perceptual quality of the restored image and the degree of distortion between the restored image and the original one [13]. The P-D curve comprehensively shows the upper bound and range of continuous changes of two types of evaluation criteria. The generative-adversarial-nets (GANs) provide a principled way to approach the P-D bound by varying the hyperparameter between distortion loss and perception loss, thus producing estimators along the P-D curve. [45, 200] proves that the P-D curve can be acquired by linear interpolation between two models, which greatly simplifies the steps to obtain the P-D curve. This provides an efficient approach for getting the P-D curve without independently training several models with different hyperparameter between distortion and perception loss. The P-D tradeoff has also been extended to other tasks, including image compression [14] and image classification [108], proving that similar tradeoffs also exist between compression ratio, and classification accuracy. However, the existing works only consider the tradeoff in single-frame image processing, and do not consider the impact of sequential information on perceptual quality and distortion.

### 5.2.3 Learning-based frame selection for burst restoration

Although the introduction of multi-frame information helps to improve the quality of the restored image (*i.e.* richer details, less noise, and higher dynamic range). However, affected by various factors, such as the restoration method, image content and degradation level, it not always true that the more images are used for processing, the better the image quality will be. Moreover, since the current methods are mostly based on large deep learning models, using multi-frame processing will not only reduce the processing efficiency but also prolong the training time. Recent work [215] proposes a new concept of image restoration potential (IRP), which reflects how much the quality of an image can be improved by restoration. To some extent, IRP proves that different images can be restored to different degrees of

quality improvement. Therefore, it is necessary to design corresponding methods to guide the setting of shooting parameters and the selection of images for restoration. Many approaches focus on designing optimal shooting mode selection methods, especially exposure length [33, 136, 186]. We conduct experiments on the burst super resolution task, and our work focus more on the theoretical analysis. Our experiments also provide references for selecting the number of frames in burst restoration tasks.

## 5.3 Preliminaries

### 5.3.1 The perception distortion tradeoff

The original P-D tradeoff formulation considers the case of a single degraded image  $y$ , which is observed according to some conditional distribution  $p_{Y|X}$ , where  $x \sim p_X$  would be the underlying true original image. This formulation assumes that the degradation is not reversible, *i.e.* cannot be estimated from  $y$  without error, which is typically the case in image restoration. Thus, given the degraded image  $y$ , a restored image  $\hat{x}$  is estimated according to the conditional distribution  $p_{\hat{X}|Y}$ . The problem setting is described in Fig. 5.1 (top).

Two performance metrics are defined: *distortion*  $E[\Delta(\hat{x}, x)]$  that measures how similar the restored image is to the actual original image, and *perception* (*i.e.* perceptual quality)  $d(p_{\hat{X}}, p_X)$  that measures the divergence between the distribution of reconstructed images  $p_{\hat{X}}$  and the distribution of natural images  $p_X$ . The perception-distortion function of the restoration task is given by

$$P(D) = \min_{p_{\hat{X}|Y}} d(p_X, p_{\hat{X}}) \quad \text{s.t. } E[\Delta(X, \hat{X})] \leq D. \quad (5.1)$$

The main finding in this formulation is that the region under the P-D function is not attainable, and the P-D function represents points where an improvement of one metric implies a worsening of the other.

### 5.3.2 Burst image restoration

We focus on the burst restoration with three degradation factors: noise, camera shake and downsampling. In this case, the  $i^{th}$  observed image in a burst with  $n$  images is related to the (unknown) original image via the following relation

$$y_i[\mathbf{u}] = x[\alpha\mathbf{u} + \mathbf{v}_i] + \epsilon_i[\mathbf{u}], \quad (5.2)$$

where  $\mathbf{u}$  represents the coordinates in the low resolution grid, in contrast to the high resolution grid  $\mathbf{u}'$  in which  $x[\mathbf{u}']$  is represented.  $\alpha$  is the subsampling

factor,  $v_i$  represents the displacement due to camera shake, and  $\epsilon_i$  represents the camera noise. We assume they are independent and identically distributed. The single image case corresponds to  $n = 1$ , which implies no misalignment, *i.e.*  $y[\mathbf{u}] = x[\alpha\mathbf{u}] + \epsilon[\mathbf{u}]$ .

Given a sequence of  $n$  degraded images  $\{y_1, y_2, \dots, y_n\}$  taken continuously, the burst image restoration estimator  $G: Y \rightarrow Z$  utilizes the information of each observed frame together to reconstruct an image  $z$  close to the original high-quality image  $x$ . The discriminator  $D: Z \rightarrow [0, 1]$  learns to distinguish the reconstructed examples generated by the estimator from real data. In contrast, the goal of the generator is to fool the discriminator by mimicking real data. Formally, the objective function of this GAN based restoration framework is formulated as follows:

$$\begin{aligned} \min_G \max_D \mathbb{E}_{z \sim p_Z} [\log(D(z))] \\ + \underbrace{\mathbb{E}_{y_i \sim p_Y} [\log(1 - D(G(y_1, y_2, \dots, y_n)))]}_{\text{perception}} \\ + \underbrace{\mathbb{E}_{y_i \sim p_Y} \|G(y_1, y_2, \dots, y_n) - x\|_1}_{\text{distortion}}, \end{aligned} \quad (5.3)$$

The difficulty of burst restoration is mainly about dealing with misalignment between each frame. In existing methods, inter-frame alignment can be performed in image space or in feature space. Here we think that the alignment module is included in the estimator and is not listed separately.

## 5.4 Burst perception distortion tradeoff

In our case, we generalize the previous formulation to the case in which a burst of  $n$  degraded images  $\{y_1, y_2, \dots, y_n\}$  is observed from the same underlying image  $x$ , each  $y_i$  being a sample from the distribution  $p_{Y_i|X} = p_{Y|X}$ , since we assume them independent and identically distributed. Critically for our analysis, there exists camera shake that may result in small misalignment between the images. Then, given the sequence of degraded images  $\{y_1, y_2, \dots, y_n\}$ , a restoration algorithm estimates a restored image  $\hat{x}$  according to the conditional distribution  $p_{\hat{X}|Y_1, Y_2, \dots, Y_n}$  (see Fig. 5.1 (bottom)). The burst perception-distortion function is thus defined as

$$P(D) = \min_{p_{\hat{X}|Y_1, Y_2, \dots, Y_n}} d(p_X, p_{\hat{X}}) \quad \text{s.t.} \quad E[\Delta(X, \hat{X})] \leq D. \quad (5.4)$$

Note that this formulation generalizes the single image P-D function and introduces the new misalignment problem between images in the burst.

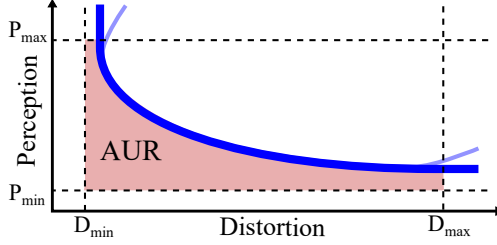


Figure 5.2: Illustration of the Area of the Unattainable Region (AUR) within the ranges  $[P_{min}, P_{max}]$  and  $[D_{min}, D_{max}]$ . The P-D curve is extended when derivative is 0 or inf (from the light blue curve to the dark blue curve) to avoid the ill effects caused by model training.

### 5.4.1 Area of the unattainable region

While the P-D plane and P-D functions are the main tools to compare the performance of restoration algorithms, we propose the *area of the unattainable region* (AUR), that is, the area under the P-D function as metric for more convenient comparison (see Fig. 5.2). This metric summarizes the performance in one single value. While AUR can be applied to the single image case, it is particularly convenient to study the influence of factors, such as burst length, in the burst case.

Since the AUR could be infinite, we define it within a range of perception and distortion values of interest  $[P_{min}, P_{max}]$  and  $[D_{min}, D_{max}]$ , respectively, as

$$\text{AUR} = \int_{D_{min}}^{D_{max}} \hat{P}(D) dx \quad (5.5)$$

where  $\hat{P}(D)$  is  $P(D)$  clamped to the range  $[P_{min}, P_{max}]$ .

### 5.4.2 Toy examples

Following [13], we present two toy examples of burst restoration only considering noise, and burst with both noise and displacement, respectively.

#### Burst without displacement

Suppose that in a burst with  $n$  images, the  $i^{th}$  observed image  $Y_i = X + N_i$ , where the original perfect image  $X \sim N(0, \sigma^2)$  and the noises  $N_i \sim N(0, \sigma_{N_i}^2)$  are independent. Take  $MSE(\cdot, \cdot)$  to be the Mean Square Error (MSE) distortion and  $d_{KL}(\cdot, \cdot)$  to be the Kullback-Leibler (KL) divergence representing the perceptual quality. For simplicity,

we assume that the final restored image with the input burst is

$$\hat{X} = \frac{1}{n} \sum_{i=1}^n a_i Y_i = \frac{1}{n} \sum_{i=1}^n a_i (X + N_i). \quad (5.6)$$

Since  $\hat{X}$  is a zero-mean Gaussian random variable, the KL divergence between two zero-mean normal distributions is given by

$$d_{KL}(p_X || p_{\hat{X}}) = \ln \left( \frac{\sigma_{\hat{X}}}{\sigma_X} \right) + \frac{\sigma_X^2}{2\sigma_{\hat{X}}^2} - \frac{1}{2}, \quad (5.7)$$

and the MSE between  $X$  and  $\hat{X}$  is given by

$$MSE(X, \hat{X}) = E[(X - \hat{X})^2] = \sigma_X^2 - 2\sigma_{X\hat{X}} + \sigma_{\hat{X}}^2. \quad (5.8)$$

Substituting Equation 5.6, we obtain that

$$\begin{aligned} \sigma_{\hat{X}}^2 &= Var \left[ \frac{1}{n} \sum_{i=1}^n a_i (X + N_i) \right] \\ &= \frac{1}{n^2} \sum_{i=1}^n a_i^2 \sigma_X^2 + \frac{1}{n^2} \sum_{i=1}^n a_i^2 \sigma_{N_i}^2 + 2 \sum_{i \neq j}^n Cov \left( \frac{1}{n} a_i X, \frac{1}{n} a_j X \right) \\ &\quad + 2 \sum_{i \neq j}^n Cov \left( \frac{1}{n} a_i X, \frac{1}{n} a_j X \right) + 2 \sum_{i, j=1}^n Cov \left( \frac{1}{n} a_i X, \frac{1}{n} a_j N_j \right) \\ &= \frac{1}{n^2} \sum_{i=1}^n a_i^2 \sigma_X^2 + \frac{1}{n^2} \sum_{i=1}^n a_i^2 \sigma_{N_i}^2 + \frac{2}{n^2} \sum_{i \neq j}^n a_i a_j \sigma_X^2, \end{aligned} \quad (5.9)$$

$$\begin{aligned} \sigma_{X\hat{X}} &= Cov(X\hat{X}) = E((X - \mu_X)(\hat{X} - \mu_{\hat{X}})) = E[X\hat{X}] \\ &= E \left[ X \cdot \frac{1}{n} \sum_{i=1}^n a_i (X + N_i) \right] = E \left[ \frac{1}{n} \sum_{i=1}^n a_i X^2 \right] + E \left[ \frac{1}{n} \sum_{i=1}^n a_i X \cdot N_i \right] \\ &= \frac{1}{n} \sum_{i=1}^n a_i (Var(X) + E(X)^2) \\ &= \frac{1}{n} \sum_{i=1}^n a_i \sigma_X^2. \end{aligned} \quad (5.10)$$

Substituting  $\sigma_X^2 = 1$ , we obtain

$$d_{KL}(p_X || p_{\hat{X}}) = \ln \left( \frac{\sigma_{\hat{X}}}{\sigma_X} \right) + \frac{\sigma_X^2}{2\sigma_{\hat{X}}^2} - \frac{1}{2} = \ln \left( \frac{1}{n} \sqrt{A_i} \right) + \frac{n^2}{2A_i} - \frac{1}{2}, \quad (5.11)$$

$$MSE(X, \hat{X}) = E[(X - \hat{X})^2] = \sigma_X^2 - 2\sigma_{X\hat{X}} + \sigma_{\hat{X}}^2 = 1 - \frac{2}{n} \sum_{i=1}^n a_i + \frac{1}{n^2} A_i. \quad (5.12)$$

$$A_i = \left( \sum_{i=1}^n a_i^2 + 2 \sum_{i \neq j}^n a_i a_j + \sum_{i=1}^n a_i^2 \sigma_{N_i}^2 \right). \quad (5.13)$$

Then we can derive a closed-form solution to

$$P(D) = \min_{d_{KL}(a_i)} \quad \text{s.t. } MSE(a_i) \leq D. \quad (5.14)$$

We start from a simple case, where  $\hat{X}$  is reconstructed from 2 degraded samples  $Y_1, Y_2$ , so  $\hat{X} = a_1 Y_1 + a_2 Y_2$  and  $n = 2$ , then

$$a_1^{mmse} = \frac{2\sigma_{N_2}^2}{\sigma_{N_1}^2 \sigma_{N_2}^2 + \sigma_{N_1}^2 + \sigma_{N_2}^2}, \quad a_2^{mmse} = \frac{2\sigma_{N_1}^2}{\sigma_{N_1}^2 \sigma_{N_2}^2 + \sigma_{N_1}^2 + \sigma_{N_2}^2} \quad (5.15)$$

$$D_{min} = 1 + \frac{-2\sigma_{N_1}^2 - 2\sigma_{N_2}^2}{\sigma_{N_1}^2 \sigma_{N_2}^2 + \sigma_{N_1}^2 + \sigma_{N_2}^2} + \frac{\sigma_{N_1}^4 + \sigma_{N_2}^4 - 2\sigma_{N_1}^2 \sigma_{N_2}^2 + \sigma_{N_1}^4 \sigma_{N_2}^2 + \sigma_{N_1}^2 \sigma_{N_2}^4}{(\sigma_{N_1}^2 \sigma_{N_2}^2 + \sigma_{N_1}^2 + \sigma_{N_2}^2)^2} \quad (5.16)$$

For 3 degraded observations  $Y_1, Y_2, Y_3$ , then we have  $\hat{X} = a_1 Y_1 + a_2 Y_2 + a_3 Y_3$  and  $n = 3$ , so

$$\begin{aligned} a_1^{mmse} &= \frac{3\sigma_{N_2}^2 \sigma_{N_3}^2}{\sigma_{N_1}^2 \sigma_{N_2}^2 \sigma_{N_3}^2 + \sigma_{N_1}^2 \sigma_{N_2}^2 + \sigma_{N_2}^2 \sigma_{N_3}^2 + \sigma_{N_1}^2 \sigma_{N_3}^2}, \\ a_2^{mmse} &= \frac{3\sigma_{N_1}^2 \sigma_{N_3}^2}{\sigma_{N_1}^2 \sigma_{N_2}^2 \sigma_{N_3}^2 + \sigma_{N_1}^2 \sigma_{N_2}^2 + \sigma_{N_2}^2 \sigma_{N_3}^2 + \sigma_{N_1}^2 \sigma_{N_3}^2}, \\ a_3^{mmse} &= \frac{3\sigma_{N_1}^2 \sigma_{N_2}^2}{\sigma_{N_1}^2 \sigma_{N_2}^2 \sigma_{N_3}^2 + \sigma_{N_1}^2 \sigma_{N_2}^2 + \sigma_{N_2}^2 \sigma_{N_3}^2 + \sigma_{N_1}^2 \sigma_{N_3}^2}. \end{aligned} \quad (5.17)$$

Considering the most extreme case when  $n \rightarrow +\infty$ , then we get  $a \rightarrow 1$ ,  $MSE(X, \hat{X}) \rightarrow 0$ ,  $d_{KL}(X, \hat{X}) \rightarrow 0$ , which means the perception distortion curve will get closer to the origin when introducing more images without displacement. We illustrate the curves with one, two and three observations in Fig. 5.3.

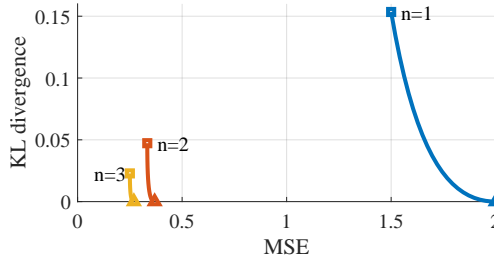


Figure 5.3: Perception distortion curve of burst image. As we get more observations  $Y_i$ , the reconstruction  $\hat{X}$  will be infinitely approaching the original sharp image  $X$ , the perception distortion curve is approaching the origin.

### Burst with displacement

For the more complex situation with misalignment between images, suppose that the degraded image  $Y_i = B_i * X(t + d_i) + N_i$ , where the original sharp image  $X \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$ , the blur kernel  $B_i \sim N(\mathbf{0}, \sigma_{B_i}^2 \mathbf{I})$  and the noise  $N_i \sim N(\mathbf{0}, \sigma_{N_i}^2 \mathbf{I})$  are independent, and  $d_i$  indicates displacement. In order to analysis the perception and distortion of burst restoration with interframe motion, we try to analyze this problem in the frequency domain with  $Y_i(\omega) = B_i(\omega)X(\omega)e^{j\omega d_i} + N_i(\omega)$ .

Substituting  $\hat{X} = \frac{1}{n} \sum_{i=1}^n a_i Y_i$ , we obtain that

$$\hat{X}(\omega) = \frac{1}{n} \sum_{i=1}^n a_i Y_i(\omega) = \frac{1}{n} X(\omega) \rho e^{j\omega \theta} + \frac{1}{n} \sum_{i=1}^n a_i N_i(\omega) \quad (5.18)$$

$$\rho = \sqrt{\sum_{i=1}^n a_i^2 |B_i(\omega)|^2 + \sum_{i=1}^n \sum_{1 \leq i < k \leq n} 2a_i a_k |B_i(\omega)| |B_k(\omega)| \cos(d_i - d_k)} \quad (5.19)$$

$$\theta = \arctan \left( \frac{\sum_{i=1}^n a_i |B_i(\omega)| \sin(d_i)}{\sum_{i=1}^n a_i |B_i(\omega)| \cos(d_i)} \right) \quad (5.20)$$

Substituting  $\sigma_X^2 = 1$ , we can obtain

$$d_{KL}(p_X \| p_{\hat{X}}) = \ln \left( \frac{\sigma_{\hat{X}}}{\sigma_X} \right) + \frac{\sigma_X^2}{2\sigma_{\hat{X}}^2} - \frac{1}{2} = \ln \left( \frac{1}{n} \sqrt{A_i} \right) + \frac{n^2}{2A_i} - \frac{1}{2} \quad (5.21)$$

$$MSE(X, \hat{X}) = E[(X - \hat{X})^2] = \sigma_X^2 - 2\sigma_{X\hat{X}} + \sigma_{\hat{X}}^2 = 1 - \frac{2}{n} \sqrt{A_i} + \frac{A_i}{n^2} \quad (5.22)$$



$$A_i = \sum_{i=1}^n a_i^2 + 2 \sum_{i=1}^n \sum_{i \neq j}^n a_i a_j \cos(d_i - d_j) + \sum_{i=1}^n a_i^2 \sigma_{N_i}^2 \quad (5.23)$$

As the displacement  $d_i - d_j$  between  $Y_i$  and  $Y_j$  increases, both  $d_{KL}(p_X \| p_{\hat{X}})$  and  $MSE(X, \hat{X})$  increase. This indicates that increasing displacement between frames in a burst negatively impacts both the perceptual quality and distortion of the restoration results.

## 5.5 Experiments

For the experiments, we focus on the particular restoration problem of burst super-resolution under camera noise, which is a common and representative problem with three degradation factors: noise, camera shake and downsampling.

### 5.5.1 Experimental setting

**Dataset.** Describable Textures Dataset (DTD) [29] is a natural texture database consisting of 5640 images with 47 categories (120 images for each). Image sizes range between  $300 \times 300$  and  $640 \times 640$ . The data is split into three equal parts for training, validation, and testing, with 40 images per class, for each split.

We generate a synthetic burst super-resolution dataset based on the DTD dataset. Each image is center-cropped to  $128 \times 128$  to get the high-resolution (HR) ground truth, and the low-resolution (LR) image is obtained by bilinear interpolation with a scaling factor of  $\times 4$ . Following the burst synthesizing process provided by [10], in each burst, we randomly add Poisson noise (*i.e.* shot noise)  $n_p \sim P(\lambda_p)$  and Gaussian noise (*i.e.* readout noise)  $n_g \sim N(0, \sigma_g)$  to each LR image. The first image in each burst is the reference frame aligned with HR. For the rest images in the burst, we add random translation on both vertical and horizontal axis  $\Delta x_s \sim N(0, \sigma_s), \Delta y_s \sim N(0, \sigma_s)$ .

**Training details.** We look at burst super-resolution methods to analyse the quality of the reconstructed image. In order to navigate the Perception-Distortion Tradeoff, we consider the ESRGAN [182] network trained with two stages, where the first stage is distortion-oriented and the second is perception-oriented. We linearly interpolate the parameters of these two models by  $\theta^{interp} = (1 - \alpha)\theta^D + \alpha\theta^P$  to obtain a continuous P-D curve. We repeat this training for each different noise and shake levels given in Table 5.1.

More specifically, we first train the distortion-oriented model only with L1 loss. The learning rate is initialized as  $2 \times 10^{-4}$  and decayed by a factor of 2 every 50 epochs. Then this model is employed as initialization for the generator. We fine-tune the

Item	Value
Gaussian Noise ( $\sigma_g$ )	[0, 10, 20, 30, 40]
Poisson Noise ( $\lambda_p$ )	[0, 1, 2, 3, 4]
Shake ( $\sigma_s$ )	[0, 1, 2, 3, 4] (pixel)
Burst Length ( $n$ )	[1, 5, 10, 20, 30, 40, 50]

Table 5.1: Experimental settings for degraded burst images. For  $\sigma_s = 0$ , only the single-frame models are trained.

generator with adversarial loss and perceptual loss to optimize the perceptual quality. The learning rate is set to  $1 \times 10^{-4}$  and halved at every 25 epochs. Our model contains 23 residual blocks, and all the images in a burst are concatenated as input. We optimize using Adam with  $\beta_1$  of 0.9 and  $\beta_2$  of 0.99, batch size of 16. We train and test the models using PyTorch on an NVIDIA GeForce 3090Ti GPU.

**Modification of P-D curve.** In addition, before calculating ADO and AUC, there are two problems that need to be solved, including the problem of inflection points in the curve and the problem of scale gaps between different metrics.

Firstly, in the training process, we first train a model that only optimizes the distortion loss to get the initial anchor point  $A_1$  of the curve, and then add the perception loss and adversarial loss on this basis, optimizing the perceptual quality and reducing the distortion, to get the end anchor point  $A_k$  of the curve. The interpolation anchor points in the curve are obtained through image interpolation. Since the distortion is also optimized during the second stage, the MSE model obtained on the first stage is not always the optimal on distortion. The distortion of the interpolation anchor points may be better than the initial point, so the curve will produce the first kind of inflection point. When training with the adversarial loss, GAN will generate some unpleasant artifacts, these artifacts will make the perceptual quality of the image worse, so it will cause the second inflection point. [13] proves that there is a tradeoff between the optimal perception and distortion, and the perceptual quality will get worse as the distortion getting better, and vice versa. We therefore consider the result of the shaded area (shown in Figure 5.2) to be the reach of the model. In order to reduce the impact of model training problems, we modified the curves with inflection points.

Secondly, since there are different metrics to measure the perception and distortion, the value ranges of different metrics are quite different. For example, NIQE ranges from 2 to 100, while RMSE is generally less than 0.2 on our dataset. Therefore, it is necessary to normalize the value first to get rid of the influence of scale gaps and fairly considering both two dimensions.

Bursts are generally collected in a very short period. Therefore, the content

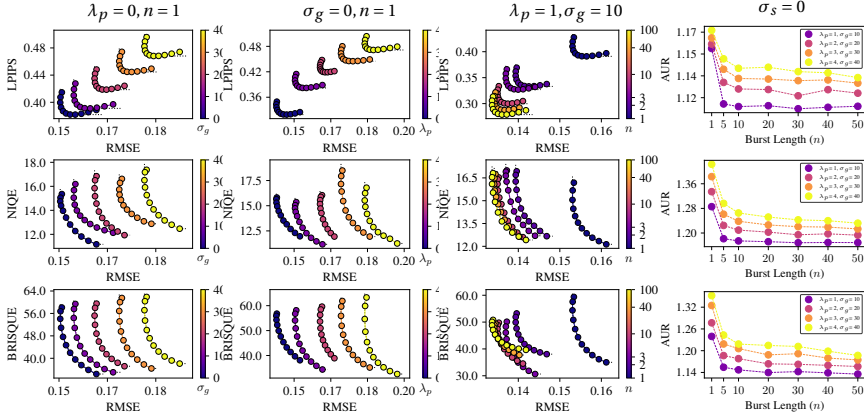


Figure 5.4: P-D curves of perfectly aligned bursts. Columns 1-3 compare the P-D curves with different levels of Gaussian noise, Poisson noise, and burst length, respectively. Column 4 shows the AUR. When images are perfectly aligned, and the noise level in each image is lower than the signal itself, using more images for burst restoration leads to better restoration quality. Note that the black dash line in the P-D planes indicates the modified P-D curves.

and imaging conditions of a burst are basically the same. The main differences between each frame are degrees of noise and misalignment. Since image noise is unavoidable in the imaging process, we analyse two common cases: when all images in a burst are perfectly aligned and when the images are not aligned.

For evaluation, we measure perceptual quality using the no-reference metrics NIQE [127] and BRISQUE [126], and for distortion, we measure RMSE. We calculate AUR of LPIPS-RMSE, NIQE-RMSE and BRISQUE-RMSE with  $[D_{min}, D_{max}] = [0, 0.3]$ ,  $[P_{min}, P_{max}] = [0, 150]$ .

### 5.5.2 Perfectly aligned bursts

This setting covers two cases: (1) *The burst is captured in a stable condition, i.e. there is no shake or motion during imaging.* (2) *The accurate motion parameters or flow between frames can be measured by equipment or estimated by algorithms.* In this case, we only consider the impact of noise on the quality of restoration results.

Foremost, the P-D curves in Fig. 5.4 prove that the P-D tradeoff still exists in burst restoration. As the noise level of the input image increases, for both Gaussian noise and Poisson noise, the P-D curve lies further from the origin, which indicates

that both the perceptual quality of the restoration image and distortion are getting worse (See Fig. 5.4 column 1-2). At a certain noise level, perception and distortion improve as the input burst length increases. When the burst length reaches a certain number, the benefit of using more images for processing decreases, since most information has already been restored. As illustrated in Fig. 5.4 column 3, the two-frame P-D curve shows a wide margin over the single-frame curve, but the gap between 10 and 100 images is quite narrow. The AUR value (see Fig. 5.4 column 4) also indicates the same tendency, proving the AUR curve captures well the P-D plane. The analysis of a perfectly aligned burst proves the importance of alignment for multi-frame processing. When a burst is well aligned, the more images the input has, the higher the image quality obtained for both perception and fidelity.

### 5.5.3 Misaligned bursts

For bursts taken by a handheld camera, shake is almost an unavoidable problem. Misaligned bursts result from two possible conditions: (1) *Direct restoration without any alignment*. (2) *Misalignment resulting from inaccurate motion or flow estimation*. In our case, we consider the impact of both shake and noise. Here we assume that the entire burst is acquired in a very short period, so only the random shake is considered. Let us also note that this case can also be understood as a proxy for the error of alignment methods.

**Level of alignment errors.** As shown in Fig. 5.5, as the burst length increases, the restoration results gradually get better at first, and after reaching the optimum quality at a certain burst length, the image quality gradually gets worse. When the displacement between images is relatively small, complementary information between different images helps recover more image details. However, when the displacement between images is too large, using more images to restore will worsen the quality. As illustrated in Fig. 5.5 column 2, when  $\lambda_p = 1, \sigma_g = 10$ , *i.e.* the AUR curves for the first image column, 20 is the optimal length for burst with shake  $\sigma_s = 1, 2, 3$ . As the noise level increases (column 3,  $\lambda_p = 3, \sigma_g = 30$ ), the larger the length of the input burst is, the better the quality of the restored image is. Therefore, the optimal burst length is determined by both the shake and the noise level.

**Types of alignment errors.** We study a more realistic case where *the handshake can be temporal correlated* by adding linear-increased translation in one direction and random translation in another. Fig.5.6 (left) shows the comparison between the burst with random displacement and temporal-correlated displacement. In addition, *the alignment error is related to the alignment method*. Fig.5.6 (right) shows the AUR of using different alignment methods, including image-level alignment with estimated optical flow [157], and feature-level alignment with deformable convolution [180], to cover various alignment errors. Inaccurate optical flows introduce

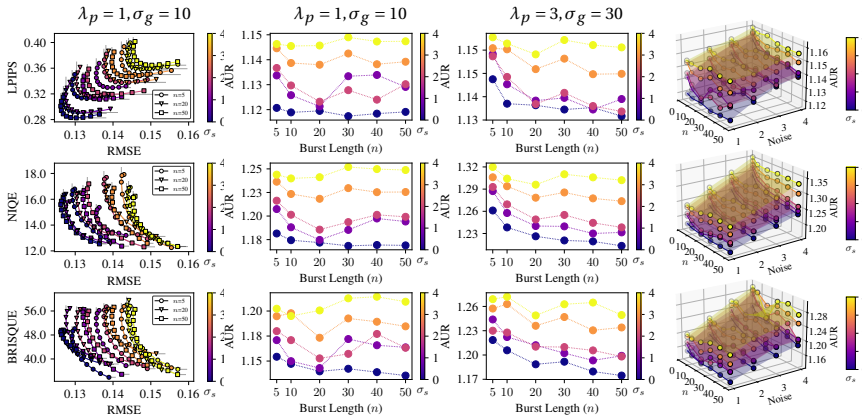


Figure 5.5: P-D curves of misaligned bursts. Column 1 compares the P-D curves with different levels of shake. Column 2,3,4 shows the AUR under different levels of noise and shake. When the burst is imperfectly aligned, an optimal burst length for restoration exists, depending on the noise level and displacement level.

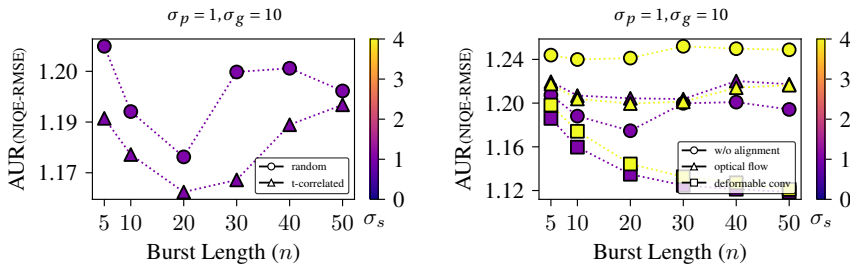


Figure 5.6: Results of burst with different types of displacement (left) and alignment error (right).

a certain level of alignment errors, which may even worsen the image quality, while still with the existence of optimal burst lengths. Deformable convolution can better align the features, so it shows the similar trend as the perfectly aligned burst, the more images the better. In these more realistic and diverse cases, the results show the same tendency as the implicit motion represent.

## 5.6 Concluding remarks

In this chapter, we extend the theory of single-frame perception-distortion tradeoff to multiple images, in particular to bursts. We analyze the impact of noise and shake

on multi-frame restoration from the perspective of the P-D tradeoff and determine the importance of alignment for burst restoration. On this basis, we propose a new metric for evaluating and comparing P-D curves. Our work provides a reference for the design of multi-frame restoration methods. In the case of estimable image noise and camera shake, the analysis results can also be used as a reference for selecting the optimal burst length.

We focus on fast-captured burst image restoration tasks. Therefore, we only consider misalignment caused by short-term camera movements, limiting our analysis to global rigid body motion. However, real-world applications can be more complex, involving factors such as object motion-induced blur and inconsistent global motion of different objects. Additionally, many models currently employ implicit motion estimation methods, simplifying our experiments to primarily focus on idealized camera angle changes and simple analyses of optical flow estimation errors. For future research directions, designing better inter-frame alignment methods is undoubtedly crucial for multi-frame image processing. Additionally, if we consider more subtle differences, misalignment can be beneficial for tasks like super-resolution, as multiple differently captured photos provide more scene sampling, aiding in better-recovering image detail information. However, no studies have yet focused on analyzing this subtle misalignment.

## 6 Conclusion and future work

### 6.1 Conclusions

In this thesis, we introduce various semantic priors in low-level vision tasks and investigate different ways to acquire and integrate the semantics in single-image and multi-image processing tasks. We always consider both the accuracy and efficiency of the algorithm to ensure that our proposed approaches can adapt to different devices and scenarios, providing a better interactive experience and interpretability.

In Chapter 2, we propose a general slimmable semantic segmentation method, which enables an adjustable accuracy-efficiency trade-off through a width-switchable segmentation network. We demonstrate the effectiveness of stepwise downward distillation on improving the performance of smaller subnetworks, and with less amount of features saved during training compared with other distillation strategies. Based on the observation of the difference between the predictions of each subnetwork, we introduce boundary supervision on low-level features of the network and propose a boundary-guided loss to further improve the segmentation results of pixels along semantic borders. We demonstrate the effectiveness of the proposed method through extensive experiments with different mainstream semantic segmentation networks on the Cityscapes and CamVid. Our proposed method improves the accuracy of the smaller submodels without significant accuracy drops in large submodels.

In Chapter 3, we introduce an interactive framework for achieving multi-image color consistency. Our framework incorporates white balance, saliency, and color naming within a general palette-based recoloring system. The combination of these additional high-level constraints significantly improves overall results and produces recolored collections free of unwanted artifacts. Through qualitative examples and a user study, we have established that our approach surpasses the current state-of-the-art methods in terms of both visual quality and user preference.

In Chapter 4, we propose an image color modification method based on the palette. Our method uses color names as a reference to increase the saturation without compromising the perceptual color name of the original colors in the image. Our method is particularly well-suited for image modification in graphic design, where adhering to a color scheme becomes paramount. The experiments

demonstrate the generalization ability of our approach.

In Chapter 5, we extend the theory of (single-frame) perception-distortion tradeoff to multiple images, in particular to bursts. We analyze the impact of noise and shake on multi-frame restoration from the perspective of the P-D tradeoff and determine the importance of alignment for burst restoration. On this basis, we propose a new metric for evaluating and comparing P-D curves. We believe that our work provides a reference for the design of multi-frame restoration methods. In the case of estimable image noise and camera shake, the analysis results can also be used as a reference for selecting the optimal burst length.

## 6.2 Future direction

Although we strive to provide better solutions to address existing research issues, even the most comprehensive approaches always have limitations. Moreover, due to the rapid development of computer vision and deep learning technologies, results obtained from studies three years ago are now far from the state-of-the-art. We have summarized the limitations of our methods and potential avenues for improvement in future work.

In terms of semantic extraction, we explored more flexible network architectures and proposed corresponding training methods. However, our compression was limited to adjusting the network width, *i.e.* number of channels of the conventional layers, and the compression parameters were globally consistent. Future exploration is needed to achieve flexible architecture adjustments in new model architectures, such as transformer-based segmentation models, which can better model global context relationships and achieve more accurate segmentation results. Moreover, the relationship between image semantics and natural language is closely intertwined. Exploring how to leverage information from large language models for semi-supervised or unsupervised semantic segmentation model training, or directly incorporating language models to assist low-level vision tasks, could be an interesting direction.

In terms of image color editing, although our method achieved better color effects by integrating semantics, it is limited by issues with palette extraction methods based on clustering. Additionally, to avoid degradation in image quality due to brightness changes, our method only considers hue during recoloring, which may not adapt well to scenes with large brightness differences between images, such as those containing both daytime and nighttime conditions. To address these issues, more complex semantic understanding and recoloring frameworks may be necessary.

In video or burst restoration and reconstruction, we demonstrated the im-



portance of accurate inter-frame motion estimation for image quality restoration. However, most current methods still rely on optical flow estimation, which may result in the loss of globally consistent motion information. This problem has garnered more attention, and introducing canonical space or 3D information may be future directions. Moreover, further exploration of the relationship between segmentation accuracy, perception, and distortion. Figuring out the relationship between semantics and image quality can deepen our understanding of the role of semantics in image restoration tasks.

Furthermore, there are many directions worth exploring in semantic understanding and image editing, including specific attribute understanding and generation based on diffusion models, such as color and motion. We hope that the methods and experimental results provided in this thesis could offer some insights for future research.



## Publications

1. **Danna Xue**, Fei Yang, Pei Wang, Luis Herranz, Jinqiu Sun, Yu Zhu, Yanning Zhang. *SlimSeg: Slimmable semantic segmentation with boundary supervision*. ACM International Conference on Multimedia (ACM MM), 2021.
2. **Danna Xue**, Luis Herranz, Javier Vazquez-Corral, Yanning Zhang. *Burst Perception-Distortion Tradeoff: Analysis and Evaluation*. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.
3. **Danna Xue**, Javier Vazquez-Corral, Luis Herranz, Yanning Zhang, Michael S. Brown. *Integrating High-Level Features for Consistent Palette-based Multi-image Recoloring*. Computer Graphics Forum, 2023.
4. **Danna Xue**, Javier Vazquez-Corral, Luis Herranz, Yanning Zhang, Michael S. Brown. *Palette-based Color Harmonization via Color Naming*. IEEE Signal Processing Letters, 2024.



# Bibliography

- [1] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11075–11083, 2019.
- [2] Adobe. Adobe color. <https://color.adobe.com/>, Accessed: 2023-08-04.
- [3] Mahmoud Afifi, Brian L. Price, Scott Cohen, and Michael S. Brown. When color constancy goes wrong: correcting improperly white-balanced images. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1535–1544, 2019.
- [4] Jonathan T. Barron and Yun-Ta Tsai. Fast fourier color constancy. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6950–6958, 2017.
- [5] Yoann Baveye, Fabrice Urban, Christel Chamaret, Vincent Demoulin, and Pierre Hellier. Saliency-guided consistent color harmonization. In *International Workshop on Computational Color Imaging*, pages 105–118. Springer, 2013.
- [6] Robert Benavente, Maria Vanrell, and Ramon Baldrich. Parametric fuzzy sets for automatic color naming. *JOSA A*, 25(10):2582–2593, 2008.
- [7] Brent Berlin and Paul Kay. *Basic color terms: Their universality and evolution*. Univ of California Press, 1991.
- [8] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Semantic segmentation with boundary neural fields. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3602–3610, 2016.
- [9] Gedas Bertasius, Lorenzo Torresani, Stella X Yu, and Jianbo Shi. Convolutional random walk networks for semantic image segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 858–866, 2017.

- [10] Goutam Bhat, Martin Danelljan, Radu Timofte, Yizhen Cao, Yuntian Cao, Meiya Chen, Xihao Chen, Shen Cheng, Akshay Dudhane, Haoqiang Fan, et al. Ntire 2022 burst super-resolution challenge. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1041–1061, 2022.
- [11] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deep burst super-resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9209–9218, 2021.
- [12] Goutam Bhat, Martin Danelljan, Fisher Yu, Luc Van Gool, and Radu Timofte. Deep reparametrization of multi-frame super-resolution and denoising. In *International Conference on Computer Vision (ICCV)*, pages 2460–2470, 2021.
- [13] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6228–6237, 2018.
- [14] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning (ICML)*, pages 675–685. PMLR, 2019.
- [15] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision (ECCV)*, pages 44–57. Springer, 2008.
- [16] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1–26, 1980.
- [17] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 97–104. IEEE Computer Society, 2011.
- [18] João Carreira and Cristian Sminchisescu. CPMC: automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Recognition and Machine Analyses*, 34(7):1312–1328, 2012.
- [19] Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. Palette-based photo recoloring. *ACM Transactions on Graphics (ToG)*, 34(4):139:1–139:11, 2015.

- 
- [20] Jiawei Chen and Chiu Man Ho. Mm-vit: Multi-modal video transformer for compressed video action recognition. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 1910–1921, 2022.
- [21] 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 Recognition and Machine Analyses*, 40(4):834–848, 2017.
- [22] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.
- [23] Wuyang Chen, Xinyu Gong, Xianming Liu, Qian Zhang, Yuan Li, and Zhangyang Wang. Fasterseg: Searching for faster real-time semantic segmentation. In *International Conference on Learning Representations (ICLR)*, 2020.
- [24] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. Deep photo enhancer: unpaired learning for image enhancement from photographs with gans. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6306–6314. Computer Vision Foundation / IEEE Computer Society, 2018.
- [25] Zhourong Chen, Yang Li, Samy Bengio, and Si Si. You look twice: Gaternet for dynamic filter selection in cnns. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9172–9180, 2019.
- [26] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1299, 2022.
- [27] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8890–8899, 2020.
- [28] Wooyeong Cho, Sanghyeok Son, and Dae-Shik Kim. Weighted multi-kernel prediction network for burst image super-resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 404–413, 2021.

- [29] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [30] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. Color harmonization. *ACM Transactions on Graphics (ToG)*, 25(3):624–630, 2006.
- [31] Marcos V Conde, Florin Vasluianu, Javier Vazquez-Corral, and Radu Timofte. Perceptual image enhancement for smartphone real-time applications. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 1848–1858, 2023.
- [32] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.
- [33] Omer Dahary, Matan Jacoby, and Alex M Bronstein. Digital gimbal: End-to-end deep image stabilization with learnable exposure times. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11936–11945, 2021.
- [34] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.
- [35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009.
- [36] Henghui Ding, Xudong Jiang, Ai Qun Liu, Nadia Magnenat Thalmann, and Gang Wang. Boundary-aware feature propagation for scene segmentation. In *International Conference on Computer Vision (ICCV)*, pages 6819–6829, 2019.
- [37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.



- 
- [38] Zhengjun Du, Kai-Xiang Lei, Kun Xu, Jianchao Tan, and Yotam I. Gingold. Video recoloring via spatial-temporal geometric palettes. *ACM Transactions on Graphics (ToG)*, 40(4):150:1–150:16, 2021.
- [39] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Burst image restoration and enhancement. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5759–5768, 2022.
- [40] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [41] Mark D Fairchild. *Color appearance models*. John Wiley & Sons, 2013.
- [42] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9716–9725, 2021.
- [43] Graham D. Finlayson and Roshanak Zakizadeh. Reproduction angular error: an improved performance metric for illuminant estimation. In *British Machine Vision Conference (BMVC)*. BMVA Press, 2014.
- [44] Flickr. Flickr. <https://www.flickr.com/>, 2004.
- [45] Dror Freirich, Tomer Michaeli, and Ron Meir. A theory of the distortion-perception tradeoff in wasserstein space. *Conference on Neural Information Processing Systems (NeurIPS)*, 34:25661–25672, 2021.
- [46] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping (Steven) Zhang, and Xinghao Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2782–2790. IEEE Computer Society, 2016.
- [47] Roland Gao. Rethinking dilated convolution for real-time semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR) Efficient CV workshop*, 2021.
- [48] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng-zhong Xu. Dynamic channel pruning: Feature boosting and suppression. In *International Conference on Learning Representations (ICLR)*, 2018.

- [49] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision (ECCV)*, pages 540–557. Springer, 2022.
- [50] Arjan Gijsenij, Theo Gevers, and Joost van de Weijer. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, 20(9):2475–2489, 2011.
- [51] Kamal Gupta, Varun Jampani, Carlos Esteves, Abhinav Shrivastava, Ameesh Makadia, Noah Snavely, and Abhishek Kar. Asic: Aligning sparse in-the-wild image collections. In *International Conference on Computer Vision (ICCV)*, pages 4134–4145, 2023.
- [52] Yoav HaCohen, Eli Shechtman, Dan B. Goldman, and Dani Lischinski. Optimizing color consistency in photo collections. *ACM Transactions on Graphics (ToG)*, 32(4):38:1–38:10, 2013.
- [53] Feng Lu Haiyang Si, Zhiqiang Zhang. Real-time semantic segmentation via multiply spatial fusion network. In *British Machine Vision Conference (BMVC)*. BMVA Press, 2020.
- [54] Kai Han, Rafael S. Rezende, Bumsuh Ham, Kwan-Yee K. Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Snet: Learning semantic correspondence. In *International Conference on Computer Vision (ICCV)*, Oct 2017.
- [55] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Recognition and Machine Analyses*, 2021.
- [56] Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.
- [57] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 574–584, 2022.
- [58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

- 
- [59] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. *Conference on Neural Information Processing Systems (NeurIPS)*, 36, 2024.
- [60] Jeffrey Heer and Maureen C. Stone. Color naming models for color selection, image editing and palette design. In *CHI Conference on Human Factors in Computing Systems*, pages 1007–1016, 2012.
- [61] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *Advances in Neural Information Processing Systems (NIPS) Deep Learning Workshop*, 2(7), 2015.
- [62] Seunghoon Hong, Xinchun Yan, Thomas S Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. *Conference on Neural Information Processing Systems (NeurIPS)*, 31, 2018.
- [63] Liang Hou, Zehuan Yuan, Lei Huang, Huawei Shen, Xueqi Cheng, and Changhu Wang. Slimmable generative adversarial networks. In *Conference on Artificial Intelligence (AAAI)*, volume 35, pages 7746–7753, 2021.
- [64] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019.
- [65] Yuanming Hu, Baoyuan Wang, and Stephen Lin.  $FC^4$ : Fully convolutional color constancy with confidence-weighted pooling. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 330–339, 2017.
- [66] Weizhe Hua, Yuan Zhou, Christopher M De Sa, Zhiru Zhang, and G Edward Suh. Channel gating neural networks. *Conference on Neural Information Processing Systems (NeurIPS)*, 32, 2019.
- [67] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense networks for resource efficient image classification. In *International Conference on Learning Representations (ICLR)*, 2018.
- [68] Xing Huo and Jieqing Tan. An improved method for color harmonization. In *International Congress on Image and Signal Processing*, pages 1–4. IEEE, 2009.

- [69] Shun Iwasa and Yasushi Yamaguchi. Color selection and editing for palette-based photo recoloring. In *International Conference on Image Processing (ICIP)*, pages 2257–2261, 2018.
- [70] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2989–2998, 2023.
- [71] Lai Jiang, Mai Xu, Xiaofei Wang, and Leonid Sigal. Saliency-guided image translation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16509–16518, 2021.
- [72] Zutao Jiang, Changlin Li, Xiaojun Chang, Ling Chen, Jihua Zhu, and Yi Yang. Dynamic slimmable denoising network. *IEEE Transactions on Image Processing*, 32:1583–1598, 2023.
- [73] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3224–3232, 2018.
- [74] Shiva Kamkar, Hamid Moghaddam, and Reza Lashgari. Early visual processing of feature saliency tasks: A review of psychophysical experiments. *Frontiers in Systems Neuroscience*, 12:54, 10 2018.
- [75] Ju-Mi Kang and Youngbae Hwang. Hierarchical palette extraction based on local distinctiveness and cluster validation for image recoloring. In *International Conference on Image Processing (ICIP)*, pages 2252–2256, 2018.
- [76] Alexander Kapitanov, Karina Kvanchiani, Alexander Nagaev, Roman Kraynov, and Andrei Makhliarchuk. Hagrid–hand gesture recognition image dataset. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 4572–4581, 2024.
- [77] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6007–6017, 2023.
- [78] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X Yu. Adaptive affinity fields for semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 587–602, 2018.

- 
- [79] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4015–4026, 2023.
- [80] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9799–9808, 2020.
- [81] Eastman Kodak. Kodak lossless true color image suite. <http://r0k.us/graphics/kodak/>, 1999.
- [82] Alexandros Kouris, Stylianos I Venieris, Stefanos Laskaridis, and Nicholas Lane. Multi-exit semantic segmentation networks. In *European Conference on Computer Vision (ECCV)*, pages 330–349. Springer, 2022.
- [83] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NIPS)*, pages 109–117, 2011.
- [84] Bruno Lecouat, Jean Ponce, and Julien Mairal. Lucas-kanade reloaded: End-to-end super-resolution from raw image bursts. In *International Conference on Computer Vision (ICCV)*, pages 2370–2379, 2021.
- [85] Changlin Li, Guangrun Wang, Bing Wang, Xiaodan Liang, Zhihui Li, and Xiaojun Chang. Dynamic slimmable network. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8607–8617, 2021.
- [86] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8300–8311, 2021.
- [87] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3050, 2023.
- [88] G Li and J Kim. Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. In *British Machine Vision Conference (BMVC)*. BMVA Press, 2020.

- [89] Hanchao Li, Pengfei Xiong, Haoqiang Fan, and Jian Sun. Dfanet: Deep feature aggregation for real-time semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9522–9531, 2019.
- [90] Peike Li, Xuanyi Dong, Xin Yu, and Yi Yang. When humans meet machines: Towards efficient segmentation networks. In *British Machine Vision Conference (BMVC)*, 2020.
- [91] Rui Li, Danna Xue, Shaolin Su, Xiantuo He, Qing Mao, Yu Zhu, Jinqiu Sun, and Yanning Zhang. Learning depth via leveraging semantics: Self-supervised monocular depth estimation with both implicit and explicit semantic guidance. *Pattern Recognition*, 137:109297, 2023.
- [92] Xiangtai Li, Xia Li, Li Zhang, Guangliang Cheng, Jianping Shi, Zhouchen Lin, Shaohua Tan, and Yunhai Tong. Improving semantic segmentation via decoupled body and edge supervision. In *European Conference on Computer Vision (ECCV)*, pages 435–452. Springer, 2020.
- [93] Xiangtai Li, Ansheng You, Zhen Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, Shaohua Tan, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *European Conference on Computer Vision (ECCV)*, pages 775–793. Springer, 2020.
- [94] Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9145–9153, 2019.
- [95] Xujie Li, Hanli Zhao, Guizhi Nie, and Hui Huang. Image recoloring using geodesic distance based color harmonization. *Computational Visual Media*, 1:143–155, 2015.
- [96] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8553–8562, 2020.
- [97] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Open-vocabulary object segmentation with diffusion models. In *International Conference on Computer Vision (ICCV)*, pages 7667–7676, 2023.
- [98] Jie Liang, Hui Zeng, Miaomiao Cui, Xuansong Xie, and Lei Zhang. Ppr10k: A large-scale portrait photo retouching dataset with human-region mask and

- group-level consistency. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 653–661, 2021.
- [99] Zhexin Liang, Chongyi Li, Shangchen Zhou, Ruicheng Feng, and Chen Change Loy. Iterative prompt learning for unsupervised backlit image enhancement. In *International Conference on Computer Vision (ICCV)*, pages 8094–8103, 2023.
- [100] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin’ichi Satoh. Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In *European Conference on Computer Vision (ECCV)*, pages 683–700. Springer, 2020.
- [101] Peiwen Lin, Peng Sun, Guangliang Cheng, Sirui Xie, Xi Li, and Jianping Shi. Graph-guided architecture search for real-time semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4203–4212, 2020.
- [102] Sharon Lin and Pat Hanrahan. Modeling how people extract color themes from images. In *SIGCHI Conference on Human Factors in Computing Systems*, page 3101–3110, 2013.
- [103] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.
- [104] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *Conference on Neural Information Processing Systems (NeurIPS)*, 34:16331–16345, 2021.
- [105] Ingmar Lissner, Jens Preiss, Philipp Urban, Matthias Scheller Lichtenauer, and Peter Zolliker. Image-difference prediction: From grayscale to color. *IEEE Transactions on Image Processing*, 22(2):435–446, 2013.
- [106] Ce Liu and William T Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In *European Conference on Computer Vision (ECCV)*, pages 706–719. Springer, 2010.
- [107] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE Transactions on Pattern Recognition and Machine Analyses*, 36(2):346–360, 2013.

- [108] Dong Liu, Haochen Zhang, and Zhiwei Xiong. On the classification-distortion-perception tradeoff. *Conference on Neural Information Processing Systems (NeurIPS)*, 32, 2019.
- [109] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4571–4580, 2019.
- [110] Quande Liu, Youpeng Wen, Jianhua Han, Chunjing Xu, Hang Xu, and Xiaodan Liang. Open-world semantic segmentation via contrasting and clustering vision-language embedding. In *European Conference on Computer Vision (ECCV)*, pages 275–292, 2022.
- [111] Risheng Liu, Long Ma, Yiyang Wang, and Lei Zhang. Learning converged propagations with deep prior ensemble for image enhancement. *IEEE Transactions on Image Processing*, 28(3):1528–1543, 2018.
- [112] Xinhua Liu, Lu Zhu, Shuchang Xu, and Shunpeng Du. Palette-based recoloring of natural images under different illumination. In *IEEE International Conference on Computer and Communication Systems*, pages 347–351, 2021.
- [113] Yifan Liu, Changyong Shu, Jingdong Wang, and Chunhua Shen. Structured knowledge distillation for dense prediction. *IEEE Transactions on Pattern Recognition and Machine Analyses*, 2020.
- [114] Yuanliu Liu, Zejian Yuan, Badong Chen, Jianru Xue, and Nanning Zheng. Illumination robust color naming via label propagation. In *International Conference on Computer Vision (ICCV)*, pages 621–629, 2015.
- [115] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.
- [116] Wuyang Luo, Su Yang, Xinjian Zhang, and Weishan Zhang. Siedob: Semantic image editing by disentangling object and background. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1868–1878, 2023.
- [117] Ziwei Luo, Youwei Li, Shen Cheng, Lei Yu, Qi Wu, Zhihong Wen, Haoqiang Fan, Jian Sun, and Shuaicheng Liu. Bsrt: Improving burst super-resolution with swin transformer and flow-guided deformable alignment. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 998–1008, 2022.



- [118] Ziwei Luo, Lei Yu, Xuan Mo, Youwei Li, Lanpeng Jia, Haoqiang Fan, Jian Sun, and Shuaicheng Liu. Ebsr: Feature enhanced burst super-resolution with deformable alignment. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 471–478, 2021.
- [119] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.
- [120] Emanuele Marino, Fabio Bruno, and Fotis Liarokapis. Color harmonization, deharmonization and balancing in augmented reality. *Applied Sciences*, 11(9):3915, 2021.
- [121] Roey Mechrez, Eli Shechtman, and Lihi Zelnik-Manor. Saliency driven image manipulation. *Machine Vision and Applications*, 30(2):189–202, 2019.
- [122] Nancy Mehta, Akshay Dudhane, Subrahmanyam Murala, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. Adaptive feature consolidation network for burst super-resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1279–1286, 2022.
- [123] S. Mahdi H. Miangoleh, Zoya Bylinskii, Eric Kee, Eli Shechtman, and Yağiz Aksoy. Realistic saliency guided image enhancement. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 186–194, June 2023.
- [124] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2502–2510, 2018.
- [125] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Conference on Artificial Intelligence (AAAI)*, volume 34, pages 5191–5198, 2020.
- [126] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.
- [127] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2012.

- [128] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.
- [129] Will Monroe, Jennifer Hu, Andrew Jong, and Christopher Potts. Generating bilingual pragmatic color references. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2155–2165, 2018.
- [130] Ethan D. Montag. Empirical formula for creating error bars for the method of paired comparison. *Journal of Electronic Imaging*, 15(1):010502, 2006.
- [131] Sean Moran, Pierre Marza, Steven McDonagh, Sarah Parisot, and Gregory Slabaugh. Deeplpf: Deep local parametric filters for image enhancement. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12826–12835, 2020.
- [132] Sean Moran, Steven McDonagh, and Gregory G. Slabaugh. CURL: neural curve layers for global image enhancement. In *International Conference on Pattern Recognition (ICPR)*, pages 9796–9803. IEEE, 2020.
- [133] Ján Morovič. *Color gamut mapping*. John Wiley & Sons, 2008.
- [134] Ho Man Rang Nguyen, Brian L. Price, Scott Cohen, and Michael S. Brown. Group-theme recoloring for multi-image color consistency. *Computer Graphics Forum*, 36(7):83–92, 2017.
- [135] Maria Olkkonen, Thorsten Hansen, and Karl R Gegenfurtner. High-level perceptual influences on color appearance. *Visual Experience: Sensation, Cognition, and Constancy*, pages 179–98, 2012.
- [136] Emmanuel Onzon, Fahim Mannan, and Felix Heide. Neural auto-exposure for high-dynamic range object detection. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7710–7720, 2021.
- [137] Huihui Pan, Yuanduo Hong, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes. *IEEE Transactions on Intelligent Transportation Systems*, 24(3):3448–3460, 2022.
- [138] Jaesik Park, Yu-Wing Tai, Sudipta N. Sinha, and In-So Kweon. Efficient and robust color consistency for community photo collections. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 430–438, 2016.

- 
- [139] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2346, 2019.
- [140] C Alejandro Párraga, Robert Benavente, Maria Vanrell, and Ramon Baldrich. Psychophysical measurements to model intercolor regions of color-naming space. *Journal of Imaging Science and Technology*, 53(3):031106, 2009.
- [141] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.
- [142] Dian Qin, Jia-Jun Bu, Zhe Liu, Xin Shen, Sheng Zhou, Jing-Jun Gu, Zhi-Hua Wang, Lei Wu, and Hui-Fen Dai. Efficient medical image segmentation based on knowledge distillation. *IEEE Transactions on Medical Imaging*, 40(12):3820–3831, 2021.
- [143] Jie Qin, Jie Wu, Pengxiang Yan, Ming Li, Ren Yuxi, Xuefeng Xiao, Yitong Wang, Rui Wang, Shilei Wen, Xin Pan, et al. Freeseg: Unified, universal and open-vocabulary image segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19446–19455, 2023.
- [144] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.
- [145] Hafijur Rahman and Gour Chandra Paul. Tripartite sub-image histogram equalization for slightly low contrast gray-tone image enhancement. *Pattern Recognition*, 134:109043, 2023.
- [146] Sheng Ren, Kehua Guo, Xiaokang Zhou, Bin Hu, Feihong Zhu, and Entao Luo. Medical image super-resolution based on semantic perception transfer learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022.
- [147] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *International Conference on Computer Vision (ICCV)*, pages 2232–2241, 2017.

- [148] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2017.
- [149] Andrés Romero, Luc Van Gool, and Radu Timofte. Smile: Semantically-guided multi-attribute image and layout editing. In *International Conference on Computer Vision (ICCV)*, pages 1924–1933, 2021.
- [150] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241, 2015.
- [151] Nikhil Sawant and Niloy J Mitra. Color harmonization for videos. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 576–582. Citeseer, 2008.
- [152] L. Shapira, A. Shamir, and D. Cohen-Or. Image appearance exploration by model-based navigation. *Computer Graphics Forum*, 28(2):629–638, 2009.
- [153] Changyong Shu, Yifan Liu, Jianfei Gao, Yan Zheng, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *International Conference on Computer Vision (ICCV)*, 2021.
- [154] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [155] João V. B. Soares, Jorge J. G. Leandro, Roberto M. Cesar Jr., Herbert F. Jelinek, and Michael J. Cree. Retinal vessel segmentation using the 2-d morlet wavelet and supervised classification. *IEEE Transactions on Medical Imaging*, 25(9):1214–22, 2005.
- [156] Jonathan Stroud, David Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 625–634, 2020.
- [157] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018.
- [158] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.

- 
- [159] Jianchao Tan, Jose I. Echevarria, and Yotam I. Gingold. Efficient palette-based decomposition and recoloring of images via rgbxy-space geometry. *ACM Transactions on Graphics (ToG)*, 37(6):262, 2018.
- [160] Jianchao Tan, Jose I. Echevarria, and Yotam I. Gingold. Palette-based image decomposition, harmonization, and color transfer. *CoRR*, abs/1804.01225, 2018.
- [161] Jianchao Tan, Jyh-Ming Lien, and Yotam I. Gingold. Decomposing images into layers via rgb-space geometry. *ACM Trans. Graph.*, 36(1):7:1–7:14, 2017.
- [162] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419. Springer, 2020.
- [163] Louis L Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273, 1927.
- [164] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3360–3369, 2020.
- [165] Michael Treml, José Arjona-Medina, Thomas Unterthiner, Rupesh Durgesh, Felix Friedmann, Peter Schuberth, Andreas Mayr, Martin Heusel, Markus Hofmarcher, Michael Widrich, et al. Speeding up semantic segmentation for autonomous driving. In *Advances in Neural Information Processing Systems Workshop*, 2016.
- [166] Yi Tu, Li Niu, Weijie Zhao, Dawei Cheng, and Liqing Zhang. Image cropping with composition and saliency aware aesthetic score map. In *Conference on Artificial Intelligence (AAAI)*, pages 12104–12111, 2020.
- [167] Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009.
- [168] Javier Vazquez-Corral and Marcelo Bertalmío. Gamut mapping for visual attention retargeting. In *Color and Imaging Conference (CIC)*, 2017.
- [169] Javier Vazquez-Corral, Maria Vanrell, Ramon Baldrich, and Francesc Tous. Color constancy by category correlation. *IEEE Transactions on Image Processing*, 21(4):1997–2007, 2012.

- [170] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *European Conference on Computer Vision (ECCV)*, pages 3–18, 2018.
- [171] Yanli Wan, Zhen Tang, Zhenjiang Miao, and Bo Li. Image composition with color harmonization. *International Journal of Pattern Recognition and Artificial Intelligence*, 26(03):1254001, 2012.
- [172] Di Wang, Jing Zhang, Bo Du, Minqiang Xu, Lin Liu, Dacheng Tao, and Liangpei Zhang. Samrs: Scaling-up remote sensing segmentation dataset with segment anything model. *Conference on Neural Information Processing Systems (NeurIPS)*, 36, 2024.
- [173] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Recognition and Machine Analyses*, 43(10):3349–3364, 2020.
- [174] Pei Wang, Yu Zhu, Danna Xue, Qingsen Yan, Jinqiu Sun, Sung-eui Yoon, and Yanning Zhang. Take a prior from other tasks for severe blur removal. *Computer Vision and Image Understanding*, page 104027, 2024.
- [175] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19795–19806, 2023.
- [176] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6849–6857. Computer Vision Foundation / IEEE, 2019.
- [177] Tao Wang, Yong Li, Jingyang Peng, Yipeng Ma, Xian Wang, Fenglong Song, and Youliang Yan. Real-time image enhancer via learnable spatial-aware 3d lookup tables. In *International Conference on Computer Vision (ICCV)*, pages 2471–2480, 2021.
- [178] 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 *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8798–8807, 2018.

- 
- [179] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skip-net: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 409–424, 2018.
- [180] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *International Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 0–0, 2019.
- [181] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 606–615, 2018.
- [182] 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 *European Conference on Computer Vision Workshop (ECCVW)*, pages 0–0, 2018.
- [183] Xudong Wang, Shufan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Hierarchical open-vocabulary universal image segmentation. *Conference on Neural Information Processing Systems (NeurIPS)*, 36, 2024.
- [184] Yikai Wang, Fuchun Sun, Duo Li, and Anbang Yao. Resolution switchable networks for runtime efficient image recognition. In *European Conference on Computer Vision (ECCV)*, pages 533–549. Springer, 2020.
- [185] Yili Wang, Yifan Liu, and Kun Xu. An improved geometric approach for palette-based image decomposition and recoloring. *Computer Graphics Forum*, 38(7):11–22, 2019.
- [186] Zhouxia Wang, Jiawei Zhang, Mude Lin, Jiong Wang, Ping Luo, and Jimmy Ren. Learning a reinforced agent for flexible exposure bracketing selection. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1820–1828, 2020.
- [187] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13022–13031, 2020.

- [188] Bartłomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (ToG)*, 38(4):1–18, 2019.
- [189] Junde Wu, Rao Fu, Huihui Fang, Yuanpei Liu, Zhaowei Wang, Yanwu Xu, Yueming Jin, and Tal Arbel. Medical sam adapter: Adapting segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.12620*, 2023.
- [190] Yuhui Wu, Chen Pan, Guoqing Wang, Yang Yang, Jiwei Wei, Chongyi Li, and Heng Tao Shen. Learning semantic-aware knowledge guidance for low-light image enhancement. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1662–1671, 2023.
- [191] Menghan Xia, Jian Yao, Renping Xie, Mi Zhang, and Jinsheng Xiao. Color consistency correction based on remapping optimization for image stitching. In *IEEE International Conference on Computer Vision Workshops*, pages 2977–2984, 2017.
- [192] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *European Conference on Computer Vision (ECCV)*, pages 736–753. Springer, 2022.
- [193] Xin Xu, Tianyi Xiong, Zheng Ding, and Zhuowen Tu. Masqclip for open-vocabulary universal image segmentation. In *International Conference on Computer Vision (ICCV)*, pages 887–898, 2023.
- [194] Danna Xue, J Vazquez Corral, Luis Herranz, Yanning Zhang, and Michael S Brown. Integrating high-level features for consistent palette-based multi-image recoloring. In *Computer Graphics Forum*, page e14964. Wiley Online Library, 2023.
- [195] Danna Xue, Luis Herranz, Javier Vazquez Corral, and Yanning Zhang. Burst perception-distortion tradeoff: analysis and evaluation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.
- [196] Danna Xue, Javier Vazquez-Corral, Luis Herranz, Yanning Zhang, and Michael S Brown. Palette-based color harmonization via color naming. *IEEE Signal Processing Letters*, 2024.



- 
- [197] Danna Xue, Fei Yang, Pei Wang, Luis Herranz, Jinqiu Sun, Yu Zhu, and Yanning Zhang. Slimseg: Slimmable semantic segmentation with boundary supervision. In *ACM International Conference on Multimedia (ACMMM)*, pages 6539–6548, 2022.
- [198] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125, 2019.
- [199] Qingsen Yan, Dong Gong, Qinfeng Shi, Anton van den Hengel, Chunhua Shen, Ian Reid, and Yanning Zhang. Attention-guided network for ghost-free high dynamic range imaging. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1751–1760, 2019.
- [200] Zeyu Yan, Fei Wen, and Peilin Liu. Optimally controllable perceptual lossy compression. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 24911–24928. PMLR, 2022.
- [201] Bailin Yang, Tianxiang Wei, Xianyong Fang, Zhigang Deng, Frederick WB Li, Yun Ling, and Xun Wang. A color-pair based approach for accurate color harmony estimation. *Computer Graphics Forum*, 38(7):481–490, 2019.
- [202] Canqian Yang, Meiguang Jin, Xu Jia, Yi Xu, and Ying Chen. Adaint: Learning adaptive intervals for 3d lookup tables on real-time image enhancement. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17522–17531, 2022.
- [203] Fei Yang, Luis Herranz, Yongmei Cheng, and Mikhail G Mozerov. Slimmable compressive autoencoders for practical neural image compression. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4998–5007, 2021.
- [204] Le Yang, Yizeng Han, Xi Chen, Shiji Song, Jifeng Dai, and Gao Huang. Resolution adaptive networks for efficient inference. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2369–2378, 2020.
- [205] Michael Ying Yang, Saumya Kumaar, Ye Lyu, and Francesco Nex. Real-time semantic segmentation with context aggregation network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 178:124–134, 2021.

- [206] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129(11):3051–3068, 2021.
- [207] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 325–341, 2018.
- [208] Jiahui Yu and Thomas Huang. Autoslim: Towards one-shot architecture search for channel numbers. *arXiv preprint arXiv:1903.11728*, 2019.
- [209] Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *International Conference on Computer Vision (ICCV)*, pages 1803–1811, 2019.
- [210] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018.
- [211] Lu Yu, Yongmei Cheng, and Joost van de Weijer. Weakly supervised domain-specific color naming based on attention. In *International Conference on Pattern Recognition (ICPR)*, pages 3019–3024, 2018.
- [212] Lu Yu, Lichao Zhang, Joost van de Weijer, Fahad Shahbaz Khan, Yongmei Cheng, and C. Alejandro Párraga. Beyond eleven color names for image understanding. *Machine Vision and Applications*, 29(2):361–373, 2018.
- [213] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In *European Conference on Computer Vision (ECCV)*, pages 489–506. Springer, 2020.
- [214] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE Transactions on Pattern Recognition and Machine Analysis*, 44(4):2058–2073, 2020.
- [215] Cheng Zhang, Shaolin Su, Yu Zhu, Qingsen Yan, Jinqiu Sun, and Yanning Zhang. Exploring and evaluating image restoration potential in dynamic scenes. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2067–2076, 2022.

- 
- [216] Chunming Zhang, Yongchun Xie, Da Liu, and Li Wang. Fast threshold image segmentation based on 2d fuzzy fisher and random local optimized. *IEEE Transactions on Image Processing*, 26(3):1355–1362, 2017.
- [217] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *International Conference on Computer Vision (ICCV)*, pages 1020–1031, 2023.
- [218] Qing Zhang, Chunxia Xiao, Hanqiu Sun, and Feng Tang. Palette-based image recoloring using color decomposition optimization. *IEEE Transactions on Image Processing*, 26(4):1952–1964, 2017.
- [219] Xiong Zhang, Hongmin Xu, Hong Mo, Jianchao Tan, Cheng Yang, Lei Wang, and Wenqi Ren. Dcnas: Densely connected neural architecture search for semantic image segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13956–13967, 2021.
- [220] Yiheng Zhang, Zhaofan Qiu, Jingen Liu, Ting Yao, Dong Liu, and Tao Mei. Customizable architecture search for semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11641–11650, 2019.
- [221] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.
- [222] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.
- [223] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6881–6890, 2021.
- [224] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 633–641, 2017.
- [225] Mingjian Zhu, Kai Han, Enhua Wu, Qiulin Zhang, Ying Nie, Zhenzhong Lan, and Yunhe Wang. Dynamic resolution network. *Conference on Neural Information Processing Systems (NeurIPS)*, 34, 2021.

## Bibliography

---

- [226] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8856–8865, 2019.
- [227] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.