Generative Models for Colorization of Visual Data

Thesis submitted in fulfilment of the requirements of the degree of

Doctor of Philosophy

in

Analytics

by

Subhankar Ghosh

Supervisor: Prof Michael Blumenstein
Co-Supervisor: Prof Umapada Pal

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

February 2025

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Subhankar Ghosh*, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

SIGNATURE: ______________________________

[Subhankar Ghosh]

DATE: 8th Feb, 2025

PLACE: Sydney, Australia

DEDICATION

To Pisimoni, Bapi, Maa, and Mejdibhai . . .

Acknowledgements

I extend my sincere gratitude to my principal supervisor, **Prof. Michael Blumenstein**, at the University of Technology Sydney (UTS), for his invaluable guidance and support throughout my PhD candidature. His continuous encouragement and insightful advice have been instrumental in making our research highly productive.

I am also deeply grateful to my co-supervisor, **Prof. Umapada Pal** of the Indian Statistical Institute (ISI), for his unwavering guidance, advice, and constant motivation. He has played the role of a critical reviewer, providing counterarguments and meticulously addressing various aspects of our research, greatly advancing its quality.

My heartfelt thanks go to my research guide, **Dr. Saumik Bhattacharya**, of the Indian Institute of Technology Kharagpur (IITKGP), for his support and invaluable advice.

My deepest thanks go out to Prof. Massimo Piccardi (UTS) and Prof. Wei Liu (UTS), members of the panel for evaluating my candidature, who provided me with valuable insights that greatly improved my research.

My sincere thanks go out to the faculty, administrative staff, and fellow researchers from the Schools of Software, CAI, and FEIT, as well as to the GRS staff at UTS for their support over the years. Similarly, I am grateful to the faculty members and fellow researchers of the CVPR Unit at the Indian Statistical Institute for their fruitful discussions during my visits to ISI.

I have greatly benefited from the constructive comments and suggestions of fellow researchers, domain experts I met at conferences and academic gatherings, and the anonymous reviewers of my papers. I sincerely thank each of them for their contributions. Special thanks to Prof. Kaushik Roy (WBSU), Dr. Abhijit Das (BITS), Dr. Arundhuti Tarafder (UPES), Prasun Roy (UTS), Tamaltaru Pal (ISI) for their help and support. In addition, I would like to thank our lab-mates Siladittya Manna, Kunal Biswas, Rakesh Dey, Arnab Halder, Alloy Das, Kunal Purkayastha, and Surajit Mukherjee. A special thanks to my flatmate, Rajkumar Bag, for his unwavering support and companionship. I acknowledge Dr. Chandranath Adak for providing this thesis template.

Finally, I am deeply thankful to my family members for their constant encouragement, enthusiasm, and unwavering support. To all those I may have inadvertently missed mentioning, please accept my heartfelt thanks.

Thank you everyone.

List of Publications

Related to this thesis:

1. **S. Ghosh**, P. Roy, S. Bhattacharya, U. Pal, and M. Blumenstein, *TIC: text-guided image colorization using conditional generative model*, Multimedia Tools and Applications 83, no. 14 (2024): 41121-41136.

2. **S. Ghosh**, S. Bhattacharya, P. Roy, U. Pal, and M. Blumenstein, *MMC: Multimodal colorization of images using textual description* , Signal, Image and Video Processing 19, no. 1 (2025): 1-10.

3. **S. Ghosh**, S. Bhattacharya, P. Roy, U. Pal, and M. Blumenstein, *λ -Color: Amplifying Long-Range Dependencies for Image Colorization*, In International Conference on Pattern Recognition, pp. 172-187. Springer, Cham, 2025

4. **S. Ghosh**, S. Bhattacharya, P. Roy, U. Pal, and M. Blumenstein, *Image Colorization using Diffusion by Solving Schrodinger Bridge Problem*, Under peer review as a journal article Computer Vision and Image Understanding.

5. **S. Ghosh**, S. Bhattacharya, P. Roy, U. Pal, and M. Blumenstein, *Exploring Color Information as Auxiliary Conditioning for Image Colourization in MS COCO Datase*, Under peer review as a journal article for Transactions on Artificial Intelligence.

6. **S. Ghosh**, S. Bhattacharya, P. Roy, U. Pal, and M. Blumenstein, *Color-YOLO: Revolutionizing Image Colorization Potential using You Only Look Once and Cross-Attention Synergy*, Under peer review as a journal article for Multimedia Tools and Applications.

Others:

1. P. Roy, **S. Ghosh**, S. Bhattacharya, U. Pal, and M. Blumenstein*Scene Aware Person Image Generation through Global Contextual Conditioning*, In 2022 26th International Conference on Pattern Recognition (ICPR), pp. 2764-2770. IEEE, 2022.

2. P. Roy, **S. Ghosh**, S. Bhattacharya, U. Pal, and M. Blumenstein*TIPS: Text-Induced Pose Synthesis*, In European Conference on Computer Vision, pp. 161-178. Cham: Springer Nature Switzerland, 2022.

3. P. Roy, S. Bhattacharya,**S. Ghosh**, U. Pal, and M. Blumenstein*d-Sketch: Improving Visual Fidelity of Sketch-to-Image Translation with Pretrained Latent Diffusion Models without Retraining*, In International Conference on Pattern Recognition, pp. 293-309. Springer, Cham, 2025.

4. P. Roy, S. Bhattacharya,**S. Ghosh**, U. Pal, and M. Blumenstein*Semantically Consistent Person Image Generation* In International Conference on Pattern Recognition, pp. 277-292. Springer, Cham, 2025.

5. A. Das, S. Biswas, P. Roy, **S.Ghosh**, U. Pal, M. Blumenstein, J. Llados, S. Bhattacharya, *FASTER: A Font-Agnostic Scene Text Editing and Rendering framework*, Accepted WACV 2025.

Table of Contents

List of Figures

Figure **Page**

LIST OF TABLES

TABLE Page

Abstract

Colorization, a well-known problem in computer vision, is the process of adding color to grayscale or monochrome images, videos, or other visual data. However, due to the ill-posed nature of the task, image colorization is inherently challenging. Though researchers have made several attempts to make the colorization pipeline automatic, their methods often produce unrealistic results due to a lack of conditioning. As realistic visual data contain different colors, one of the significant challenges that visual colorization techniques encounter is handling varying objects' colors. To handle this, the colorization algorithms often require the best knowledge of semantic understanding of the scene. However, when this semantic information is introduced, the natural blending of colors becomes difficult. This thesis introduces a comprehensive framework for image colorization that addresses the challenges of consistency, realism, and scalability in real-world scenes. Here, at first, we provide a critical survey of research on image and video colorization in Chapter 2, where existing methods are comprehensively reviewed and discussed. We compared the existing strategies and extensively analyzed their advantages, disadvantages, and performances. Based on the limitations of the existing methods, we propose several novel techniques to have more robust colorization. Initially, we explored the long-range dependencies of λ Network for image colorization (Chapter 3). Next, object information was explored as an additional input to improve image colorization with a cross-attention mechanism (Chapter 4). We also attempt to integrate textual descriptions of the grayscale image as an auxiliary condition to improve the fidelity of the colorization process. Here, to train a larger network with sufficient gradient, RRDB(Residual in Residual Dense Block) is explored (Chapter 5). Further, to reduce the inherent ambiguity of object color, we introduce a novel multi-modal strategy that incorporates object information along with their color information in the colorization process (Chapter 6). To get a more realistic output, we also devised an adversarial training using GAN model, and later we introduced a diffusion-based method for superior performance (Chapter 7). As there is no available dataset with a rich textual description of the objects with corresponding color information, we also introduced a new dataset to

assist model training. Finally, we refine the diffusion-based technique without additional guidance to get a more realistic colorization model that can work in the wild without color information (Chapter 8).

Chapter 1

Introduction

Old legacy movies and historical videos are in black and white format as there was no suitable technology to capture color information during recordings. Black and white or grayscale images can be restored by new real-life colorization, which gives life to old pictures and videos. The main aim of this colorization process is to add color to a grayscale image such that the newly generated image is visually appealing and meaningful. In recent years, after the introduction of generative adversarial networks (GANs) [20], colorization techniques have seen a leap, and the state-of-the art performances have been reported on several recent databases [7, 14, 87]. These colorization techniques [4, 33, 48, 55, 72, 99] are quite different in many aspects, such as different types of loss functions, network architecture, learning strategies, etc. However, the existing colorization processes mostly follow unconditional generation where the colors are predicted only from the grayscale input image. This might lead to ambiguous results as the prediction of color from grayscale information is inherently ill-posed. To increase the fidelity in the colorization pipeline, in this thesis, we propose text-guided and object information-based colorization methods where object information along with color descriptions about the objects present in the grayscale image are used as auxiliary conditions to achieve more robust colorized results.

1.1 Background

The process of image colorization has also changed a lot from the old manual or artistic way to an enhanced computational methods-driven technique. Artists have been literally painting on photographs and even film frames to make colorized versions of black-and-white photos since the early 20th century. But it worked in a long and convoluted way, that was highly inefficient, subjective, and required a fairly high degree of artistic/color theory interpretive expertise.

With the rise of computer vision and machine learning fields, researchers started to dig out automatic techniques for image colorization. Scribble-based colorization had already led to some automation in the computational methods, although with user inputs through scribbles providing an indication of colors. Due to its dependency on manual inputs in the form of scribbles, it is difficult to use scribble-based techniques for large datasets. To mitigate the problem, auto-colorization was born out of the recent machine learning advancements, especially convolutional neural networks (CNN), where fully automatic methods evolved to create plausible colors for grayscale images without external input from a human.

Recent advances in the field of deep learning, such as CNNs (Convolutional Neural Networks), GANs (Generative Adversarial Networks), Diffusion models, and Attention mechanisms have revolutionized how colorization is done. These methods are trained on large sets of colored images, teaching them to predict reliable colors for grayscale images. For a more versatile interaction, we introduced text-based colorization as one of the new approaches that extends the system to take textual descriptions and produces more reliable colored output.

1.2 Motivation

1. **Colorizing for Preservation:** Adding colors to old photographs not only increases the perceptual quality, but also creates more relatable and immersive experience for the audiences. Thus, colorization breathes new life into old films, documentaries, and images with historical values.

2. **Improving interpretability:** In instrumentation, we use different colors so that it is easy for our eyes to distinguish between different segments. For instance, in a big instrument, different parts are usually marked with different colors for better understanding. Similarly, in natural scenes, objects like trees or the sky or

ocean have typically identifiable colors that convey relevant information about a scene. In domains such as medical imaging, where anatomies need to be distinguished and abnormalities better highlighted, colorization can greatly improve the interpretability of black-and-white images through automatic colorization.

3. **Creative and Aesthetic Applications:** Colorization is not just a technical challenge but also an artistic one. Researchers and artists alike are interested in how machine can be used to create aesthetically pleasing and artistically relevant results. For instance, in the entertainment industry, adding colors to black-and-white films allows them to be appreciated by a broader audience. Thus, colorized images have a huge demand in media production, advertising, and artistic creation.

4. **Enhancing Accessibility:** In certain fields, such as medical imaging, enhancing black-and-white images with color can provide clearer distinctions between different tissues or structures, helping professionals to diagnose and analyze conditions more effectively. In other cases, such as satellite imaging or security footage, adding color to grayscale images may help to improve clarity and usability.

5. **Machine Learning in Colorization:** The art of converting black-and-white images to color has been one area where computer vision and machine learning have truly pushed the envelope. This is designed to be ambiguous due to the inherent uncertainty that in a scene the relationship between an object and its color is often not unique. The above problem motivated researchers to come up with ingenious solutions using neural networks, GANs, and self-supervised learning techniques. However, these models suffer from various limitations, like unrealistic generation of color, color bleeding, low saturation of color, less variability in color, etc. Thus, it is important to address these problems related to colorization in order to improve the performance of a model.

The motivation of this thesis is to develop deep learning-based techniques for the colorization of grayscale visual data because of its importance in numerous applications as outlined above. To address the same, this thesis proposes several novel colorization techniques in the following chapters.

1.3 Challenges

The ultimate goal of image colorization research is to develop models that can automatically produce high-quality, realistic colorized outputs that are indistinguishable from ground-truth color images. However, there are several key research challenges in this domain:

1. **Ambiguity:** A single grayscale image can have multiple possible colorizations. For instance, the color of a car in a grayscale image is indeterminate- red, blue, or green are all valid options here. Research should focus on developing models that can deal with for this ambiguity and produce multiple plausible colorizations based on context or even user preferences.

2. **Incorporating Context and Semantic Understanding:**: Successful colorization requires understanding the context and the content of the image. For example, the model should know that the sky is typically blue, the grass is green, and water bodies are often blue or gray. This needs deeper scene understanding and object recognition.

3. **User Interactivity:** As the perceptual preferences of viewers vary greatly, the colorization process should include more user interactivity, allowing humans to guide the colorization process easily through simple input (like providing color hints or using natural language descriptions). Text-based colorization, for example, may open up new possibilities where users can describe the scene to achieve a desired outcome.

4. **Realism and Consistency:** While it is easy to apply colors, ensuring that they look natural and blend well is a challenging issue for automatic systems. Colors should blend naturally with surrounding regions, shadows, light, etc, and should not appear artificial or out of place. Generating colors that are not only realistic but also consistent across the entire image is another challenge.

1.4 Objectives

Image colorization aims to restore and enhance black-and-white images by automatically predicting and applying realistic colors and different techniques. It is motivated by the desire to preserve historical visuals, improve image interpretability across fields like medical imaging, and push the boundaries of computer vision and creative applications. The main objectives of the thesis are as follows:

1. To develop an automatic method that can colorize grayscale images without the need for human intervention, reducing the time and effort traditionally required for manual colorization.
2. To restore old black-and-white photos or films with colors that are historically accurate, helping to maintain cultural and historical integrity.
3. To address the inherent ambiguity in colorizing grayscale images by producing plausible color options or providing contextually correct color predictions.
4. To enable the model to understand objects and scenes within the image, so that the model can apply colors that are consistent with the nature and other objects of the scene.

1.5 Research questions

1. How does the model handle historical or artistic images where accurate color references may not be available, and what strategies can be used to estimate plausible colors?
2. How well does a text-based colorization model perform in applying correct colors to specific objects using textual descriptions in different images?
3. Can a colorization model consistently apply appropriate colors to objects (e.g., sky, grass, clothing) across multiple images with varying contexts?
4. How can we effectively develop a new dataset for image colorization using text embeddings, and what are the key challenges in training a deep learning model in the absence of an existing large-scale annotated dataset for text-based colorization?
5. How can we effectively develop diffusion models that have revolutionized image generation and image-to-image translation?

1.6 Contributions

The main contributions of the thesis are as follows:

1. A long-range lambda mechanism ($\lambda-$Networks) has been proposed to achieve the faithful generation of color. Our key contribution is to include the long-range interactions without a transformer-based attention model for the colorization task. To the best of our knowledge, this is the first attempt to use lambda abstraction to invoke attention in the colorization process. Extensive experiments show that our proposed method significantly outperforms the SOTA algorithms.

2. We also propose a novel colorization network that uses multi-modal feature attention to color images more accurately. Our method of end-to-end model architecture produces color variations with diverse structures, shapes, and hues. We proposed incorporating additional sources of information, such as semantic segmentation maps and object recognition algorithm features to reduce ambiguity in the colorization process. This additional information guides the colorization process, fostering more consistent and accurate results.

3. To the best of our knowledge, a descriptive text-guided image colorization approach has been proposed for the first time in this study. A novel GAN pipeline is proposed that exploits textual descriptions as an auxiliary condition. We extensively evaluate our framework using qualitative and quantitative measures. In comparison with the state-of-the-art (SOTA) algorithms, it was found that the proposed method generates results with better perceptual quality. The textual color description acts as an additional conditioning to increase the fidelity in the final colorized output.

4. Colorization methods by conditioning on textual color information and introducing both a novel dataset and a baseline method for evaluation are also explored. A multi-modal pipeline is proposed that colorizes the image using language information, which is considered auxiliary conditioning in the colorization process. We also proposed a novel dataset based on the color information of every object of the COCO dataset. We generate color-coded textual information by associating class labels with their respective objects in the images.

5. An instance object colorization method has been developed in this study. The proposed IOC (Instance object colorization) module utilizes instance label image colorization, exploiting object-colour associations. To achieve superior performance, we design the IOC module as a multi-task network. To the best of our knowledge, this is the first attempt to design a multi-task network for the colorization task, considering the object-level instances. A multi-modal pipeline is proposed that

colorizes the image using language information, which is considered auxiliary conditioning in the colorization process. To ensure high fidelity over colors, a novel loss function is also proposed that captures the overall color consistency of a scene.

6. We explore how the problem of image colorization can be addressed using recent generative techniques, like diffusion algorithm by solving Schrödinger bridge problem. This work aims to perform image colorization by incorporating two key components: adversarial learning and regularization. By leveraging the principles of stochastic differential equations, we attempt to map a grayscale image into the respective color image. This enables us to assign appropriate colors to grayscale images, revitalizing them and enhancing their visual appeal. The proposed method also uses adversarial loss to mitigate the exponentially increasing computational complexity due to the high dimensionality of the color.

1.7 Organization of the Thesis

This thesis spans multiple dimensions, including algorithmic innovation, multimodal strategies, and the integration of advanced machine learning techniques like diffusion models and cross-attention mechanisms in image colorization. These advancements enable us to achieve superior results in colorization, pushing the boundaries of what is possible in computer vision. The organization of the thesis is as follows.

Chapter 2 presents a comprehensive literature review, examining existing research in the field. It highlights key advancements and methodologies related to image colorization. The chapter identifies gaps in the literature and positions the current study within this context.

Chapter 3 introduces a hybrid model that combines convolutional layers with lambda layers, enabling efficient feature extraction and contextual understanding of the image. This architectural design ensures that the network can capture both local and global dependencies, resulting in colorized outputs that are not only visually appealing but also contextually accurate. This chapter addresses the **Research question 1**.

Chapter 4 aims to resolve the inherent ambiguity of colorization by taking a multimodal approach to ensure consistent and realistic outputs. To further enhance the realism of the colorized images, we also introduce the Cross Attention-Based Attention (CABA) module. These result in a cohesive and visually consistent colorization that captures the nuances of the scene. This chapter also addresses the **Research question 1**.

Chapter 5 represents one of the most innovative aspects of our approach where we incorporate textual descriptions as auxiliary conditions in the colorization process. Unlike traditional methods that rely solely on grayscale inputs, our model takes two inputs: the grayscale image and its corresponding encoded text description. This chapter addresses the **Research questions 2 and 3**.

Chapter 6 aims to further refine the colorization process, where we introduce the Instance Object Colorization (IOC) module. This module is conditioned on both the grayscale image and its associated language description, enabling object-level colorization. A fusion model then integrates these object segments to generate a cohesive and fully colorized image. This chapter addresses the **Research questions 2 and 3**.

Chapter 7 Shows that our method not only enhances the aesthetic appeal of the colorized images but also ensures that the results are consistent and reliable across a wide range of inputs. For this work, we curated a dataset using the COCO dataset, enriched with textual descriptions of object colors. This chapter addresses the **Research questions 2, 3 and 4**.

Chapter 8 discusses recent advancements in diffusion models that have revolutionized image generation and image-to-image translation. Based on these developments, we propose a colorization method that leverages the Schrödinger Bridge image-to-image translation framework. This chapter addresses the **Research question 5**.

Chapter 9 concludes the thesis by summarizing the research contributions and outlining the potential directions for future work.

CHAPTER 2

Literature Review

Over the past two decades, image colorization has emerged as a prominent focus in computer vision research. Initially, conventional machine learning methods drove advancements in this field [5, 27, 45]. However, recent years have witnessed a shift towards deep learning (DL) methodologies, propelled by their success across various domains [3, 58, 75, 80]. DL-based automatic image colorization systems have demonstrated remarkable performance [4, 8, 10, 40, 42, 45, 72, 80, 99]. The task presents unique challenges, including ambiguous color mappings, semantic understanding, and maintaining structural consistency. Overcoming these challenges has led to the development of diverse approaches, ranging from traditional machine learning to advanced deep learning techniques.

This chapter comprehensively overviews image and video colorization techniques, exploring various methodologies, architectures, advancements, and datasets. Different evaluation metrics used for colorization are also discussed here.

2.1 Image Colorization

According to the challenges outlined in chapter 1, it is important to present a more aligned and focused discussion in this chapter for its better clarity.. Image colorization is challenging due to ambiguity, consistency, and the inherently ill-posed nature of the task, as multiple plausible colorizations can exist for a single grayscale image. Addressing those, existing techniques can be broadly categorized into guidance-based and guidance-free methods. Guidance-based methods rely on additional information such as reference images, semantic maps, or textual descriptions. These methods transfer color from a source to a target image using features like object boundaries or textures. While they often produce high-quality results, their performance heavily depends on the quality and relevance of the guidance provided. Guidance-free methods, on the other hand, learn to colorize images using only the grayscale input. These are typically based on deep learning models, such as CNNs or GANs, which learn color distributions directly from large-scale datasets. Although they offer more automation and scalability, they may suffer from desaturated or unrealistic outputs due to a lack of semantic context. Below, we have discussed it in detail.

2.1.1 Guidance-based Colorization

2.1.1.1 Object Based Colorization

Instance-aware Image Colorization [72] presented a colorization approach capable of coloring a wide range of objects with diverse contexts. The network backbone of [72] was adopted from Zhang et al. [101]. In [68], the authors used unsupervised hierarchical disentanglement enables fine-grained object generation and discovery in their work. It learns structured representations without labels, capturing detailed object variations.

Wu et al. [92] employed a pre-trained GAN for feature matching and introduced variety by altering the latent space for the subsequent GAN network. However, challenges arose when the pre-trained GAN produced misleading features, leading to the generation of unnatural colors.

In contrary, Conditional GAN-based methods [65, 81] generate high-resolution, photorealistic images from semantic label maps, surpassing existing techniques in quality and resolution for deep image synthesis and editing.

InstaGAN [55] leverages instance information to enhance multi-instance transfiguration and introduces a context-preserving loss to maintain an identity outside target instances.

2.1.1.2 Text-Based Colorization

Text-based colorization networks utilize textual descriptions as input to guide the colorization process. The notable approaches are discussed below.
The language-conditioned colorization network was developed by Manjunatha *et al.* [53]. To colorize the gray image, the authors used the text captions as a condition of the color.

The author proposed a CNN architecture of eight blocks, where each block has a sequence of convolution layers and batch normalization. The authors used the MS COCO dataset [27] for the training, and the network's output is 50×50.

The text2color [4] model consists of two conditional adversarial networks: Text to Palette Generation network and Palette based colorization network. Text to Palette Generation network is trained using the palette-and-text dataset. Text to Palette Generation network generator learns the color palette from the text and identifies the fake and real color palettes. Huber loss [88] is used as a loss function in this network. The palette-based colorization network is based on an U-NET [62] architecture that utilizes color palette as a conditioning. In the discriminator, the author designed a series of convo-leakyRelu architecture.

In [90], the authors proposed a language-based colorization network using a color-corresponding matrix and a soft-gated injection module.

Zabari et al. [97] proposes a framework that employs an image diffusion technique guided by granular text prompts. The integration enhances semantic appropriateness and user control in the colorization process.

2.1.1.3 User-Guided Colorization

User-guided networks incorporate user inputs, such as points, strokes, or scribbles, to guide the colorization process.

Sangkloy et al. [64] proposed an end-to-end GAN-based approach. The architecture adopted an encoder-decoder structure with residual blocks.

Zhang et al. [101] developed a real-time system with local and global hint networks. Local hints preprocessed inputs to yield color distributions, while global hints utilized saturation and histogram statistics. The network comprised ten blocks with convolutional layers, ReLU activation, and batch normalization. The author proposes an adaptive colorization method [43] leveraging Vision Transformers, enabling efficient propagation of user hints to distant regions in images.

The paper [11] introduces a framework that guides users on where to provide color hints, enhancing the efficiency of interactive sketch colorization.

The study [15] focuses on automatic image colorization by learning representations from large-scale data, which can be adapted for user-guided scenarios. Aksoy et al. [1] presents a deep learning approach that simulates user inputs to guide the colorization process interactively. Xiao *et al.* [95] developed an Interactive colorization model based on the U-Net [62] architecture to colorize grayscale images. The architecture utilizes global and local inputs. The network consists of 4 modules: a feature extraction module, a dilated module, a global input module, and a reconstruction module. Feature extraction module takes three inputs(grayscale image, local input, and gradient map) that are merged with an element-wise summation. Then the output of the features extraction layer module is connected with a dilated module with one convolution layer. After dilating the module's output, the feature is further processed by a reconstruction module. The reconstruction module consists of many convolutions and de-convolution modules. After the reconstruction, the output feature is combined with the input grey image and generates the colorized image.

2.1.2 Guidance-Free Colorization

2.1.2.1 RGB Image-based Colorization

Deep colorization [10] was the first method to employ CNNs for image colorization. The architecture consisted of five fully connected layers with ReLU activation. During training, the least-squares error was used as the loss function. Features were extracted at three levels: low-level grayscale values, mid-level DAISY features [75], and high-level semantic labels. The network input size was 256×256, and the SUN dataset [96] was used for training. The output features are passed through a series of residual blocks followed by batch normalization [30] and leaky ReLU [52]. After the last residual block, the features are passed through the convolution block to get the three-channel image, i.e., RGB image as output. The authors used the ImageNet [14] model for the feature extraction. CaffeNet (a variation of Alexnet [39]), Googlenet [66], Vgg16 [74], and Resnet50 [23] are used in the same setting with three datasets.

Zhang et al. [99] introduced a CNN-based approach that formulated colorization as a classification task. The network predicted the probability distribution over a quantized color gamut. It consisted of eight blocks, each with convolutional layers, ReLU activation, and batch normalization [30]. This method struggled with accurately colorizing objects with appropriate colors.

Deep depth colorization [8] leveraged pre-trained ImageNet networks as feature extractors. The architecture included residual blocks with batch normalization and Leaky ReLU [52]. The network mapped depth information to RGB channels, achieving impressive results in object recognition and colorization.

Kumar et al. [40] used a conditional autoregressive transformer for generating low-resolution images and a network with two parallel networks for coarse and fine colorization.

Colorformer [32] presents the network that comprises a transformer-based encoder and a color memory decoder, featuring a global-local attention operation enhancing global receptive field dependencies. With a color memory module storing semantic-color mappings, the decoder leverages image-adaptive queries to produce vivid and diverse colorization outcomes. Bigcolor [35] focuses on vivid color synthesis while reducing the burden of synthesizing image structures. By learning a generative color prior that complements spatial structures, their method is facilitated by a BigGAN-inspired encoder-generator network.

CT2 [89] presents an end-to-end transformer-based model that leverages transformers' long-range context extraction using holistic architecture to enhance color diversity.

2.1.2.2 Diffusion Based Colorization

In DDcolor [33], a dual decoder model is introduced for spatial resolution restoration. This model incorporates a query-based color decoder, which enhances features across multi-scale representations of color. By leveraging dual decoders, DDcolor effectively addresses spatial details while preserving color fidelity, contributing to the overall quality of the colorized images.

In Palette [63], the authors used the diffusion model for colorization. The paper also presents denoising, restoration, and JPEG compression for image data.

The study by Liu et al. [12] introduces an enhanced colorization model that utilizes the powerful T2I diffusion model. By leveraging this approach, the model achieves impressive and varied colorization results while maintaining a high level of perceptual quality. Their research demonstrates the effectiveness of the proposed model in the field of image colorization.

The authors presents CtrlColor [46], a multimodal colorization method utilizing the pre-trained Stable Diffusion model. It addresses user interaction and flexibility limitations, supporting unconditional and conditional image colorization with inputs like text prompts, strokes, and exemplars.

In [59], the authors propose a model based on the diffusion model, replacing the Unet with ED-UNet, and train it on various datasets to address the challenges of image colorization. This research [83] discusses a denoising diffusion null-space model for image restoration, which can be adapted for image colorization tasks. Lin et al. [48] discuss high-fidelity image colorization techniques using diffusion models to produce vivid and accurate colors.

2.1.2.3 Infrared and Rader-based Image Colorization

Domain-specific networks focus on colorizing images from specialized modalities, such as infrared or radar imagery. The infrared colorization technique [47] utilizes a multi-branch deep convolutional neural network designed to learn luminance and chrominance channels. Initially, the input image undergoes preprocessing and is transformed into a pyramid representation. Each branch of the network is trained independently on a single pyramid level without sharing weights between layers. Subsequently, all branches are merged into a fully connected layer. Each block within the network contains the same number of convolutional layers, with convolution layers using a 3×3 kernel and downsampling performed through a 2×2 pooling layer. Additionally, the authors developed a real-world dataset of road scenes, training the model on 32,000 images and evaluating it on 800 test images.

Wang *et al.* [80] proposed SAR-GAN for colorizing Synthetic Aperture Radar (SAR) images using a cascaded generative adversarial network (GAN) architecture. SAR-GAN comprises two subnets: the speckling subnet and the colorization subnet. The speckling subnet generates noise-free SAR images, while the colorization subnet processes them further to add color. The speckling subnet consists of eight convolutional layers with Batch Normalization and element-wise division within a residual network. The colorization subnet employs an encoder-decoder architecture with eight convolutional layers and skip connections. The entire network is trained using the Adam optimizer [37]. SAR-GAN utilizes a hybrid loss function that combines L1 loss with adversarial loss. The model is evaluated on 88 images from a dataset of 3,392 images.

The author [73] proposes a deep multi-scale convolutional neural network to perform automatically integrated colorization from single-channel near-infrared images to RGB images. Wei et al. [86] introduces a novel infrared colorization algorithm that leverages similarities in high-frequency features between visible and infrared images, achieving colorization through frequency domain feature decoupling and reconstruction.

2.2 Video Colorization

Video colorization extends image colorization techniques to sequential frames, addressing challenges like temporal consistency and motion artifacts. Prominent methods include: Video colorization with hybrid generative adversarial network (VCGAN) [102] works with a two-stage network. The four main parts of VCGAN architecture are the Global feature extractor, placeholder feature extractor, mainstream encoder-decoder, and discriminator. The leading architecture mainstream encoder-decoder adopts from the U-Net architecture. The placeholder feature extractor uses the Resnet50 [23] architecture for short-cut connections. In the middle of the U-net architecture, the global features and placeholder features are combined for the mainstream. In the discriminator section, the author uses a patch discriminator. Both networks are adopted with leaky ReLU activation functions. Moreover, the authors use perceptual loss for the generator. In the first stage, the whole ImageNet dataset [14] was trained for the global features. Furthermore, the second stage utilizes the DAVIS dataset [58] containing 156 videos. Fully Automatic Video Colorization with Self-Regularization and Diversity [44] uses the confidence based refinement network. In the training step, the authors perform the K-nearest neighbor search in the ground truth frame. After, the model gets the most appropriate or high confidence frame and passes it to the following stage network for refinement. The authors proposed a diversity loss for colorization. The main backbone of the architecture is U-net In training, the 1000 pairs of the frame are randomly chosen from the DAVIS [58] dataset. Both networks work in 480p videos and images.

The Deep Exemplar-based Video Colorization [98] presents an end-to-end network for colorization. The network architecture is divided into two subnets: correspondence and colorization. The input of the network is the source image and reference image. The correspondence subnet generated a similarity map and warped the color of the source image. After that, two images are passed through the colorization subnet and generate the color image of the source video frame. The authors use a combination of perceptual loss, contextual loss, adversarial loss, and smoothening loss for optimizing the networks.

2.3 Datasets

Image colorization is a challenging and important task in computer vision, involving the transformation of grayscale images into colorized versions. To achieve this, neural networks are trained using large-scale image databases. These databases play a critical role in the training, validation, and testing of colorization models.

This section discusses some of the most widely used image datasets for colorization tasks, highlighting their characteristics and significance.

2.3.1 COCO-Stuff Dataset

The MS-COCO dataset[49] is a well-known image dataset. The **COCO-Stuff Dataset** [7] is an extension of the primary MS-COCO dataset, specifically designed to include annotations for both common objects and the surrounding context. The inclusion of “stuff” annotations makes this dataset highly versatile for a wide range of computer vision tasks, including image colorization.

- **Size and Diversity:**
  - The dataset contains a total of **164,000 images**, providing a large and diverse set of scenes.
  - It covers **172 categories**, including objects and contextual elements, along with an unlabeled class.
- **Annotations:**
  - Each image is densely annotated, enabling the study of object and scene relationships.
  - The dataset includes pixel-wise segmentation labels, which can aid in tasks that require spatial understanding.
- **Applications in Colorization:**
  - The diversity of scenes and objects in COCO-Stuff makes it an excellent choice for training colorization networks.
  - The wide range of categories ensures that models can generalize well to various real-world scenarios.

2.3.2 Caltech-UCSD Birds 200

The **Caltech-UCSD Birds 200 Dataset** [87] is a bird classification dataset, specifically designed to include annotations for bird species and the surrounding context.

- **Size and Diversity:**
  - The dataset contains a total of **6033 images**, providing a large and diverse set of bird species mostly in North America.
  - It covers **200 categories**, of different species in the dataset.

- **Annotations:**
  - Each image is densely annotated, enabling the study of bird color.
  - The dataset includes various color labels for different parts of the bird, such as beak color, feather color, and head color, among others.
- **Applications in Colorization:**
  - The diversity of different bird colors makes it an excellent choice for training colorization networks.
  - The wide range of categories ensures that models can generalize well to various real-world scenarios.

2.3.3 PASCAL VOC Dataset

The **PASCAL Visual Object Classes (VOC) Dataset** [18] is another popular dataset in computer vision, known for its focus on object detection, segmentation, and classification tasks. It is also frequently used in colorization research due to its high-quality images and well-defined categories.

- **Dataset Size:**
  - The dataset contains approximately **12,000 images**, which are split into training, validation, and test sets.
- **Categories:**
  - There are **20 object categories**, including animals, vehicles, and household items.
- **Annotations:**
  - Each image is annotated with bounding boxes, segmentation masks, and object labels.
  - These annotations provide a rich source of information for tasks like guided colorization.
- **Application in Colorization:**
  - The variety of object categories ensures that models trained on PASCAL VOC can handle diverse image content.

2.3.4 CIFAR Datasets

The **CIFAR (Canadian Institute for Advanced Research) Datasets** [38] are among the most commonly used datasets for image classification and colorization tasks. They are particularly well-suited for training lightweight models due to their small image size.

- **Versions:**
  - **CIFAR-10:** Contains 10 classes, such as airplanes, automobiles, and animals.
  - **CIFAR-100:** Contains 100 classes, offering more granularity and diversity.
- **Dataset Size:**
  - Each version includes **60,000 images** in total: which are split into 50,000 for training and 10,000 for testing.
- **Image Characteristics:**
  - The images are 32×32 **pixels** in size, making them computationally efficient.
  - The dataset is derived from the Tiny Images dataset, ensuring a wide variety of content.
- **Applications in Colorization:**
  - CIFAR datasets are ideal for developing and testing colorization algorithms that operate on low-resolution images.
  - They are also useful for benchmarking model performance due to their standardized splits and extensive use in the research community.

2.3.5 ImageNet ILSVRC2012

The **ImageNet Dataset** [14] is one of the largest and most comprehensive image datasets available, making it a cornerstone of modern computer vision research. The **ILSVRC2012 (ImageNet Large Scale Visual Recognition Challenge 2012)** subset is widely used for training deep learning models, including those for colorization.

- **Dataset Size:**

  - The dataset contains over **1.2 million training images** across **1,000 categories**.
  - The validation set consists of **50,000 images**, providing a robust benchmark for model evaluation.

- **Categories:**
  - The categories span a wide range of objects, animals, and scenes, ensuring comprehensive coverage of real-world scenarios.

- **Image Characteristics:**
  - The images are of high resolution and vary significantly in terms of composition, lighting, and color distribution.

- **Applications in Colorization:**
  - ImageNet's large size and diversity make it an ideal dataset for training colorization networks.
  - Models trained on ImageNet often achieve state-of-the-art performance due to the richness of the data.

2.3.6 DAVIS Dataset

The **DAVIS (Densely Annotated VIdeo Segmentation) Dataset** [58] is primarily designed for video segmentation tasks but is also valuable for colorization research, particularly in the context of temporal consistency.

- **Dataset Composition:**
  - The dataset includes **50 video sequences** with a total of **2,455 annotated frames**.
  - All frames are annotated at **24 frames per second (fps)**.

- **Resolution:**
  - The videos are provided in **full HD resolution** (1920 × 1080), ensuring high-quality inputs.

- **Annotations:**

  - Each frame is densely annotated with segmentation masks, enabling fine-grained analysis.

- **Applications in Colorization:**
  - The temporal aspect of DAVIS makes it ideal for developing and testing video colorization algorithms.
  - High-resolution images allow researchers to study the impact of resolution on colorization quality.

2.4 Evaluation Metrics

Image colorization is a complex and widely researched task in computer vision, involving the transformation of grayscale images into plausible and aesthetically pleasing colorized versions. Evaluating the quality of generated images is crucial for assessing the performance of colorization models. Researchers utilize both quantitative and qualitative metrics to measure the effectiveness of these models. Among the quantitative metrics, *Structural Similarity Index Measure* (SSIM) [84] and *Peak Signal-to-Noise Ratio* (PSNR) are commonly used. Additionally, qualitative evaluation often involves opinion-based user studies to capture subjective perceptions of image quality. Advanced metrics such as *Grayscale-to-Real* (G2R) and *Real-to-Grayscale* (R2G) scores have also gained prominence in colorization research.

This section provides an in-depth exploration of these evaluation methods, elaborating on their mathematical foundations, practical applications, and relevance to the field of image colorization.

2.4.1 Quantitative Metrics

Quantitative metrics provide objective measurements of the quality of colorized images by comparing them to ground truth images. These metrics are based on mathematical formulas and are widely used for benchmarking the performance of colorization models.

2.4.1.1 Structural Similarity Index Measure (SSIM)

The *Structural Similarity Index Measure* (SSIM) [84] is a perceptual metric that quantifies image quality by measuring the similarity between two images. Unlike traditional metrics such as Mean Squared Error (MSE), SSIM considers luminance, contrast, and structural information, which are critical for human visual perception.

Mathematical Definition

The SSIM index between two images x and y is defined as:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.1}$$

where:

- μ_x and μ_y are the mean intensities of x and y.
- σ_x^2 and σ_y^2 are the variances of x and y.
- σ_{xy} is the covariance between x and y.
- C_1 and C_2 are small constants to stabilize the division.

Applications in Colorization

SSIM is extensively used to evaluate the structural fidelity of colorized images. A higher SSIM score indicates that the generated image preserves the structural integrity of the original grayscale image while adding plausible color information.

2.4.1.2 Peak Signal-to-Noise Ratio (PSNR)

The *Peak Signal-to-Noise Ratio* (PSNR) is a widely used metric for measuring the quality of reconstructed images. It quantifies the ratio between the maximum possible signal power and the power of noise that affects the fidelity of the image.

Mathematical Definition

PSNR is defined as:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right) \tag{2.2}$$

where:

- MAX is the maximum possible pixel value of the image (e.g., 255 for 8-bit images).
- MSE is the Mean Squared Error between the ground truth and generated images, calculated as:

  $$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N} \|x_i - y_i\|^2 \tag{2.3}$$

  where x_i and y_i are the colors of the i-th pixel of the ground truth image and generated image, respectively, N is the total number of pixels.

Applications in Colorization

PSNR is used to measure the pixel-wise accuracy of colorized images. Higher PSNR values indicate better reconstruction quality, although this metric does not account for perceptual differences.

2.4.1.3 Frechet Inception Distance (FID)

FID[24] is measured by computing the differences between the representations of features, such as edges and lines, and higher-order phenomena, such as the shapes of eyes or paws that are transformed into an intermediate latent space.

Mathematical Definition

The Frechet Inception Distance (FID) is given by:

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right) \tag{2.4}$$

where μ_r and Σ_r are Mean and Co-variance of the real images.
μ_g and Σ_g are Mean and Co-variance of the Generated images.

Application in Colorization

FID can help to compare the performance of GAN model variations or architectures.

2.4.1.4 Learned Perceptual Image Patch Similarity (LPIPS)

Learned Perceptual Image Patch Similarity (LPIPS) [100] measures the perceptual similarity between two images by comparing their deep feature representations. Given two images x and y, LPIPS computes the distance between their deep feature embeddings extracted from a pre-trained convolutional neural network (e.g., VGG [67], SqueezeNet [28]).

Mathematical Formulation

Let:

- $\phi_l(x)$ and $\phi_l(y)$ be the feature maps from layer l of the chosen deep network.
- w_l be learned weights for each layer.
- N_l be the number of spatial elements in layer l.

LPIPS is defined as:

$$d(x,y) = \sum_l w_l \frac{1}{N_l} \sum_{i=1}^{N_l} \left\| \phi_l^i(x) - \phi_l^i(y) \right\|_2^2 \tag{2.5}$$

where:

- $\|\cdot\|_2^2$ represents the squared Euclidean distance between feature vectors.
- The distance is normalized across spatial dimensions.

The weights w_l are learned based on human perceptual similarity judgments to align LPIPS scores with human perception.

Application of LPIPS in Image Colorization

- **Evaluating Perceptual Quality**: LPIPS is used to assess how realistic and natural the generated colors appear compared to the ground truth.
- **Comparing Models**: Helps compare different colorization models such as CNNs, GANs, diffusion models, and transformers.
- **Ablation Studies**: Used in research to validate the impact of different loss functions or architectures on perceptual quality.

2.4.2 Qualitative Evaluation

While quantitative metrics provide objective measurements, they may not fully capture the perceptual quality of colorized images. Qualitative evaluation involves subjective assessments, often through user studies.

2.4.2.1 Opinion-Based User Studies

In opinion-based studies, human participants are asked to evaluate the quality of colorized images based on criteria such as realism, vibrancy, and naturalness. These studies provide valuable insights into how well the generated images align with human expectations.

Methodology

- Participants are presented with pairs of images: the original color image and the colorized version.
- They rate the images on a Likert scale or choose the image they find more realistic.

- Statistical analysis is performed on the collected data to derive meaningful conclusions.

Challenges

- Subjectivity: Different participants may have varying perceptions of quality.
- Scalability: Conducting large-scale user studies can be time-consuming and resource-intensive.

2.4.2.2 G2R and R2G Scores

Recent advancements in colorization evaluation have introduced metrics such as *Grayscale-to-Real* (G2R) and *Real-to-Grayscale* (R2G) scores. These metrics aim to bridge the gap between quantitative and qualitative evaluations by incorporating perceptual aspects.

Grayscale-to-Real (G2R) Score: The G2R score measures how effectively a colorization model transforms grayscale images into realistic color images. It combines structural similarity with perceptual realism.

Real-to-Grayscale (R2G) Score: The R2G score evaluates the reverse process, assessing how well the colorized image can be converted back to a grayscale version that resembles the original input. This metric emphasizes consistency and reversibility in colorization.

In the next chapter, we explore the transformative role of image colorization, which brings black-and-white images to life while inferring colors in scenarios where traditional methods fall short. We present a novel algorithm that seamlessly converts grayscale images into consistent color compositions by combining convolutional and lambda layers.

CHAPTER 3

λ -Color: Amplifying Long-Range Dependencies for Image Colorization

Colorization of images serves as a transformative tool, imbuing black and white pictures with vitality that mirrors the essence of the captured moment. Beyond merely transitioning aged images into modern color renditions, this process extends its reach to inferring colors for images where conventional color-capturing methods fail. In this work, we introduce a novel algorithm designed to seamlessly convert grayscale images into perceptually consistent color compositions. We have also developed a novel layer by combining convolutional and lambda layers towards image colorization. Our proposed algorithm represents a significant advancement in the field of image colorization, offering a multifaceted solution to enhance visual storytelling and comprehension.

3.1 Introduction

Colorization presents a significant challenge due to the diverse range of colors objects within a scene may possess, influenced by factors like lighting and texture. For instance, skin tones can vary under different lighting conditions, while landscapes may appear distinct based on time or season. To tackle this complexity, researchers have devised various colorization techniques, ranging from manual to automatic and semi-automatic methods. Manual colorization involves adding color by hand using software, like Photoshop, offering control and artistic freedom but demanding time and expertise. Automatic approaches leverage machine learning to predict colors based on patterns learned from extensive datasets. Semi-automatic methods, like scribbler-based techniques, allow user input for finer control. However, automatic colorization encounters challenges such as one-to-many associations, where a grayscale image can have multiple equally plausible colorizations. To address this, strategies like generative adversarial networks (GANs) and self-supervised learning have been proposed. Yet, many automatic methods struggle with color consistency and realism. To enhance results, we propose a novel Lamda net-based algorithm. These aids can guide the colorization process, ensuring more consistent and accurate outcomes. This holistic approach aims to improve colorization's realism, semantic understanding, and overall naturalness, bridging the gap between grayscale input and vibrant, lifelike color output. In this work, we propose a lambda abstraction-based colorization model that takes a grayscale image as input and produces the color components using local and global attention computed using a lambda module.

In this work, our key contribution is to include the long-range interactions without a transformer-based attention model for the colorization task. To the best of our knowledge, this is the first attempt to use lambda abstraction to invoke attention in the colorization process. Our extensive experiments show that our proposed method significantly outperforms the SOTA algorithms.

The remaining sections of this work are structured as follows. Section 3.2 provides an overview of the pipeline and details of the proposed methodology. Experimental results, encompassing dataset description, qualitative findings, comparisons with established methods, and various ablation studies, are presented in Section 3.3. Finally, Section 3.4 concludes the work by summarizing key observations, addressing limitations, and outlining potential future avenues for enhancing the proposed algorithm.

3.2 Proposed Framework

Long-range interactions without explicit attention mechanisms are a key aspect of many recent advancements in neural network architectures. These architectures are designed to capture dependencies between distant elements in the input data without relying on traditional attention mechanisms. Instead, they utilize various techniques to facilitate communication and information exchange across different parts of the network. One approach is to increase the receptive field of convolutional layers by stacking multiple layers or using dilated convolutions. This allows the network to capture information from a broader context without introducing additional parameters or computational overhead. Another technique incorporates recurrent connections between layers, enabling information to propagate across multiple time steps or processing stages. In colorization methods, the LAB color space is commonly employed. This method typically involves taking the "L" channel, which represents the grayscale image, and predicting the "AB" channel to add colorization. This process utilizes the independence of the luminance (L) channel from the chrominance (AB) channel to train efficient colorization algorithms.

3.2.1 Lambda Networks

In the Lambda network [6], we aim to construct a linear function $R^{|k|} \rightarrow R^{|v|}$, denoted by a matrix $\lambda_n \in \mathbb{R}^{|k|\times|v|}$. The lambda layer initially computes keys K and values V through linear projections of the context. Keys are then normalized across context positions using a softmax operation, resulting in normalized keys $\overline{K}$. The λ_n matrix is derived by aggregating the values V using the normalized keys $\overline{K}$ and position embeddings E_n, formulated as:

$$\lambda_n = (\overline{K}^T \cdot V) + (E_n^T \cdot V) \tag{3.1}$$

Where:

- λ_n represents the content lambda and position lambda.

3.2.1.1 Positional Embeddings

Like traditional Transformer models, Lambda Networks incorporate positional embeddings to encode the order or position of elements in the input sequence. However, unlike Transformers, which utilize these embeddings primarily for attention mechanisms, Lambda Networks uses them as the basis for modeling interactions across distant elements in the sequence.

3.2.1.2 Point-wise Transformations

Lambda Networks applies point-wise transformations to the positional embeddings to generate context-aware representations for each element in the sequence. These transformations are applied independently to each element, allowing the model to capture long-range dependencies without the need for pairwise attention computations.

3.2.1.3 Local Interaction Window

To limit the computational complexity of modeling long-range interactions, Lambda Networks introduces a local interaction window. Instead of considering interactions between all pairs of elements in the sequence, the model focuses on a local neighborhood around each element. This windowing strategy helps control the computational cost while still enabling the model to capture global context information.

3.2.1.4 Learnable Basis Functions

Lambda Networks use learnable basis functions to parameterize the point-wise transformations. These basis functions are shared across all elements in the sequence and are optimized during training to capture relevant interactions between positional embeddings.

3.2.1.5 Hierarchical Feature Representation

Lambda Networks are capable of learning hierarchical representations of features in the input sequence. The model can progressively capture higher-level abstractions and dependencies in the data by applying multiple layers of point-wise transformations.

3.2.1.6 Down-Lambda layers

We construct a specialized layer, combining a lambda layer with a convolutional operation. Initially, a convolutional layer with a 3×3 kernel is applied, facilitating down-sampling by a factor of 2. Subsequently, a lambda layer is introduced, augmenting the feature count while concurrently diminishing the image dimensions.

3.2.1.7 Up-Lambda layers

The up-lambda layer consists of a convolutional transpose layer followed by a lambda layer. The convolution transpose layer employs a 2×2 kernel with a stride of 2. This layer receives two inputs: one from the lower dimension and the other from the encoder side. Subsequently, the lambda layer is applied for further processing.

3.2.1.8 Pseudo-code for the Multi-query lambda layer

Here is the pseudo-code for the Multi-query lambda layer. This lambda layer leverages tensor operations for efficient computation in deep learning models, specifically in the context of multi-query attention mechanisms.

```
def lambda_layer(queries, keys, embeddings, values):
    """Multi-query lambda layer."""
    # b: batch, n: input length, m: context length,
    # k: query/key depth, v: value depth,
    # h: number of heads, d: output dimension.

    content_lambda = einsum(softmax(keys), values, 'bmk,bmv->bkv')
    position_lambdas = einsum(embeddings, values, 'nmk,bmv->bnkv')

    content_output = einsum(queries, content_lambda, 'bhnk,bkv->bnhv')
    position_output = einsum(queries, position_lambdas, 'bhnk,bnkv->
                                                          bnhv')

    output = reshape(content_output + position_output, [b, n, d])
    return output
```

Here, The einsum operation represents generalized contractions between tensors of arbitrary dimensions. It is numerically equivalent to broadcasting its inputs to share the union of their dimensions, performing element-wise multiplication and summing across all dimensions not specified in the output.

3.2.2 Generator

The generator architecture is tailored to handle single-channel images sized 256 × 256. Initially, an input convolution layer with a 3 × 3 kernel and 64 channels is applied. Subsequently, four down-lambda layers are employed to gradually increase the dimensional representation of the image by reducing the matrix size. This downsampling process diminishes the matrix size to 16 × 16 by halving it iteratively, thereby enhancing the feature representation. Simultaneously, the number of channels or features is doubled in each down-lambda layer. This augmentation ensures richer feature extraction and better representation learning. Consequently, the generator transforms the input grayscale image into a higher-dimensional feature space, enabling it to capture intricate details and semantic information effectively. This design choice optimizes the generator's capability to generate high-quality colorized outputs while maintaining computational efficiency. Following the height representation of the feature metric, a decoder-like representation is crafted. This involves incorporating four up-lambda layers to generate an enhanced representation of the matrix with a size of 256 × 256. Additionally, features from the

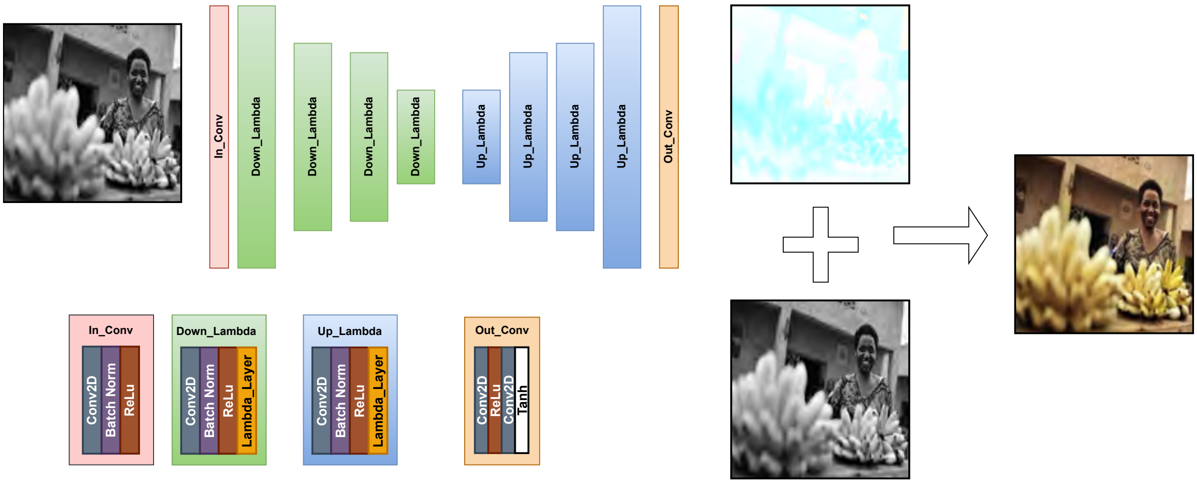


Figure 3.1: Diagram of the proposed network.

same down-lambda layers are utilized to compute the subsequent layer in the up-lambda sequence. Finally, an out convolutional layer is added, configuring the channel count to 2. Please see the detailed overview of the model in Fig 3.1.

3.2.3 Discriminator

To ensure effective local quality detection of colorized images, our colorization task employs a PatchGAN discriminator, denoted as D. This discriminator is pivotal in evaluating the quality of generated colorized images at the patch level, facilitating high-quality single-level generation. Grayscale images (L^i) are paired either with target images (T^i) or estimated images (E^i), where T^i and E^i represent the AB channels of the color image. The combination of (L^i,T^i) is labeled as real, while (L^i,E^i) is labeled as fake, thereby enforcing discrimination on image transitions rather than the images themselves. The Patch discriminator in our model processes a three-channel input dimension of 256 × 256. It consists of three convolution blocks, each containing 64, 128, and 256 filters, respectively, with a filter dimension of 4 × 4. Strides of 2 are employed for the first two convolution blocks, while a stride of 1 × 1 is used for the last two blocks. Batch normalization and leaky ReLU activation follow each convolution layer. Subsequently, a single filter of kernel size 4 × 4 is applied with a stride of 1 to compute the final response. The discriminator's output is the average of these final responses.

3.2.4 Losses

3.2.4.1 MAE Loss

The L_1 loss, also known as the mean absolute error (MAE) loss, measures the absolute differences between corresponding elements of two tensors. It is commonly used as a loss function in regression problems to penalize the magnitude of the errors between predicted and target values. The L_1 loss is calculated as follows:

$$\mathscr{L}_{L_1} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{3.2}$$

where:

- N is the number of samples or elements in the tensors.
- y_i is the true target value or ground truth.
- $\hat{y}_i$ is the predicted value.
- $|\cdot|$ denotes the absolute value.

3.2.4.2 GAN Loss

The GAN loss, used in Generative Adversarial Networks (GANs), is a key component in training the generator and discriminator networks. It comprises two main components: the generator loss ($\mathscr{L}_{GAN}^{G}$) and the discriminator loss ($\mathscr{L}_{GAN}^{D}$).

The generator loss, $\mathscr{L}_{GAN}^{G}$, is computed using binary cross-entropy loss ($\mathscr{L}_{BCE}$), which measures the difference between the discriminator's prediction on generated images and the target label (usually 1, indicating real images). The formulation is as follows:

$$\mathscr{L}_{GAN}^{G} = \mathscr{L}_{BCE}(D(L^i, G(L^i)), 1) \tag{3.3}$$

Similarly, the discriminator loss, $\mathscr{L}_{GAN}^{D}$, involves two binary cross-entropy terms. The first term computes the loss based on the discriminator's prediction on real images (T^i) compared to the real label (1), while the second term evaluates the discriminator's prediction on generated images ($G(L^i, S^i)$) against the fake label (usually 0, indicating fake images). The formulation is given by:

$$\begin{aligned} \mathscr{L}_{GAN}^{D} =& \mathscr{L}_{BCE}(D(L^i, T^i), 1) \\ &+ \mathscr{L}_{BCE}(D(L^i, G(L^i)), 0) \end{aligned} \tag{3.4}$$

In both equations, $\mathscr{L}_{BCE}$ represents the binary cross-entropy loss function.

3.2.5 Training Details

During training, we employed the Adam optimizer with a learning rate of 2e-4, utilizing $\beta 1 = 0.5$ and $\beta 2 = 0.99$. The training process utilized an NVIDIA GeForce RTX 3080 Ti with 12GB of memory. The model comprises 19.84 million parameters, and training was conducted efficiently with these specifications, ensuring optimal convergence and performance.

3.3 Experiments Details, Results and Analysis

3.3.1 Datasets

To generalize a method effectively, evaluating its performance on standard datasets is crucial. In our study, we utilized two prominent datasets for testing: the COCO dataset [7] and a natural color dataset. The COCO dataset comprises 118,000 images, serving as a benchmark for various computer vision tasks due to its diversity and scale. Additionally, we incorporated a natural color dataset [3] consisting of 723 images across 20 different categories, providing a more specific evaluation of color-related tasks. The utilization of these datasets allows for comprehensive assessment across different domains and scenarios. The COCO dataset offers a wide range of real-world scenes and objects, enabling robust evaluation under diverse conditions. Meanwhile, the natural color dataset provides insights into color-related tasks within specific categories, enhancing the method's applicability to real-world scenarios. By testing on these standard datasets, we ensure that the method's performance is validated across various contexts, facilitating its generalization and practical deployment in real-world applications. In the training, we used 108,000 images for training and 10,000 images for testing.

3.3.2 Comparison of results

To ensure the effectiveness of our model, we conducted comparisons with state-of-the-art colorization models. While recent research has focused on diffusion models, which require additional descriptions or language to generate color images, our model offers fully automatic image colorization. Therefore, we specifically compared our results with models that do not rely on any supplementary information. Our comparisons included

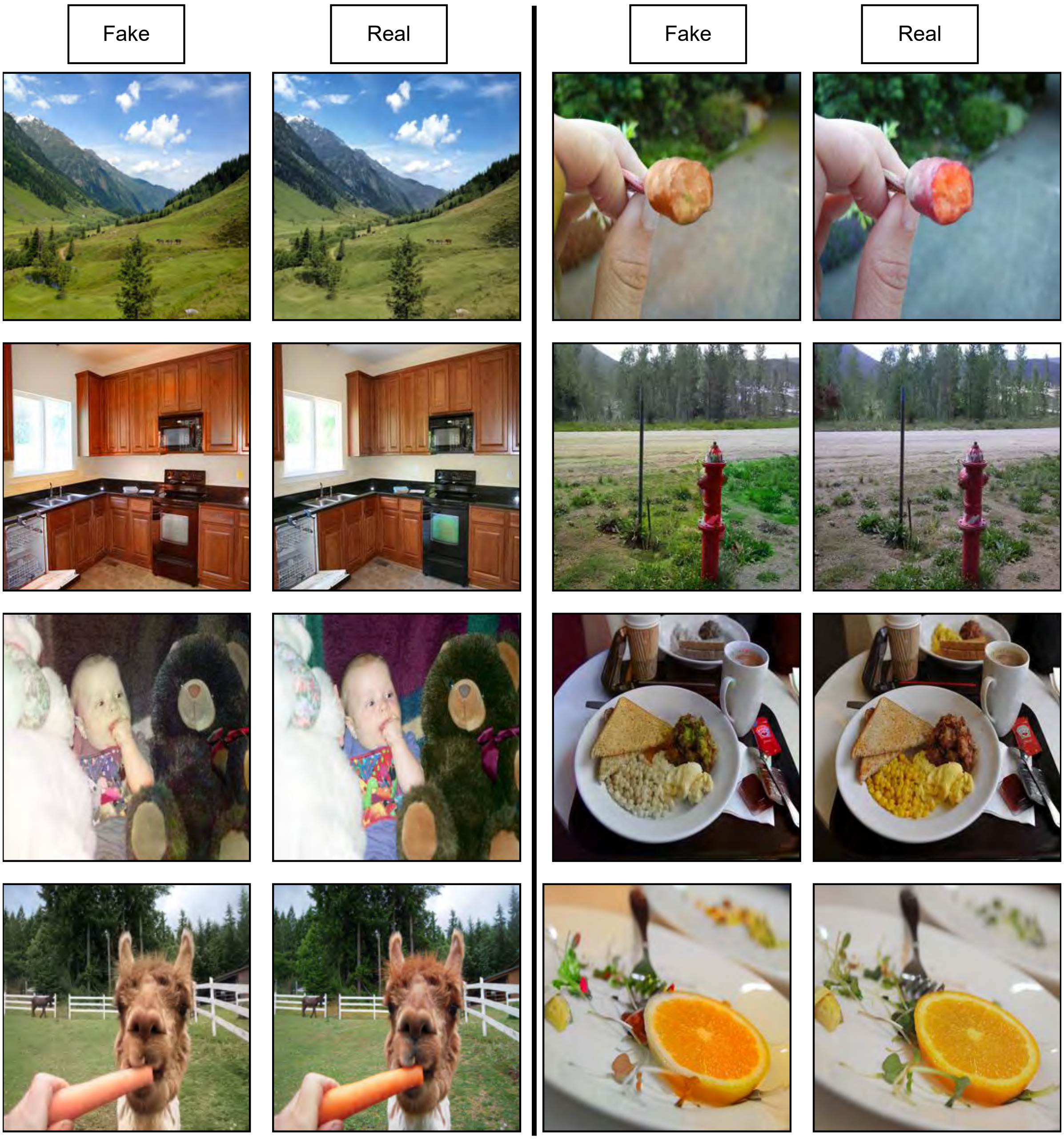


Figure 3.2: Examples of some qualitative results generated from the COCO-Stuff dataset by the proposed framework. Here, "FAKE' means the Images generated by our method, and "REAL" means the Original images in the dataset [Best visible in 300% zoom].

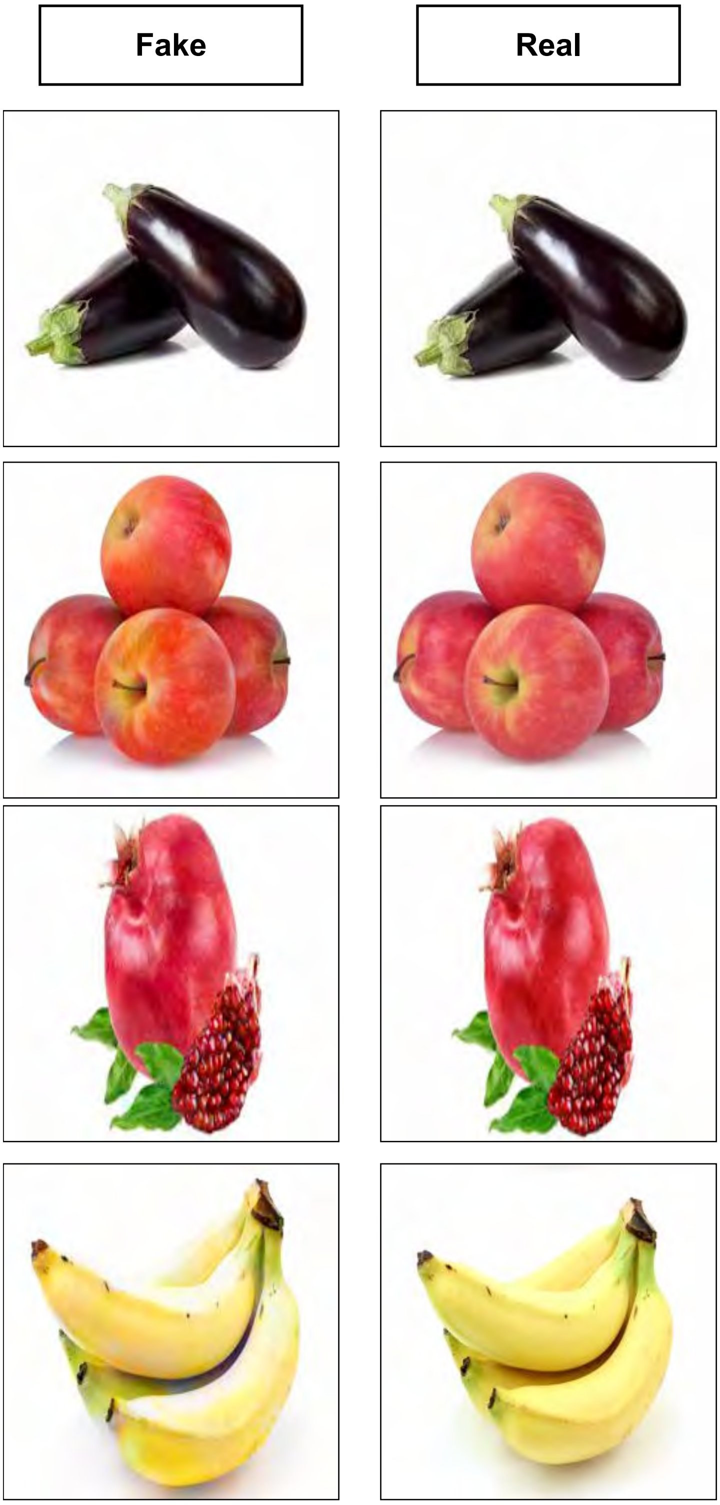


Figure 3.3: Examples of some qualitative results generated from the NCD dataset by the proposed framework. Here, "FAKE' means the Images generated by our method, and "REAL" means the Original images in the dataset [Best visible in 300% zoom].

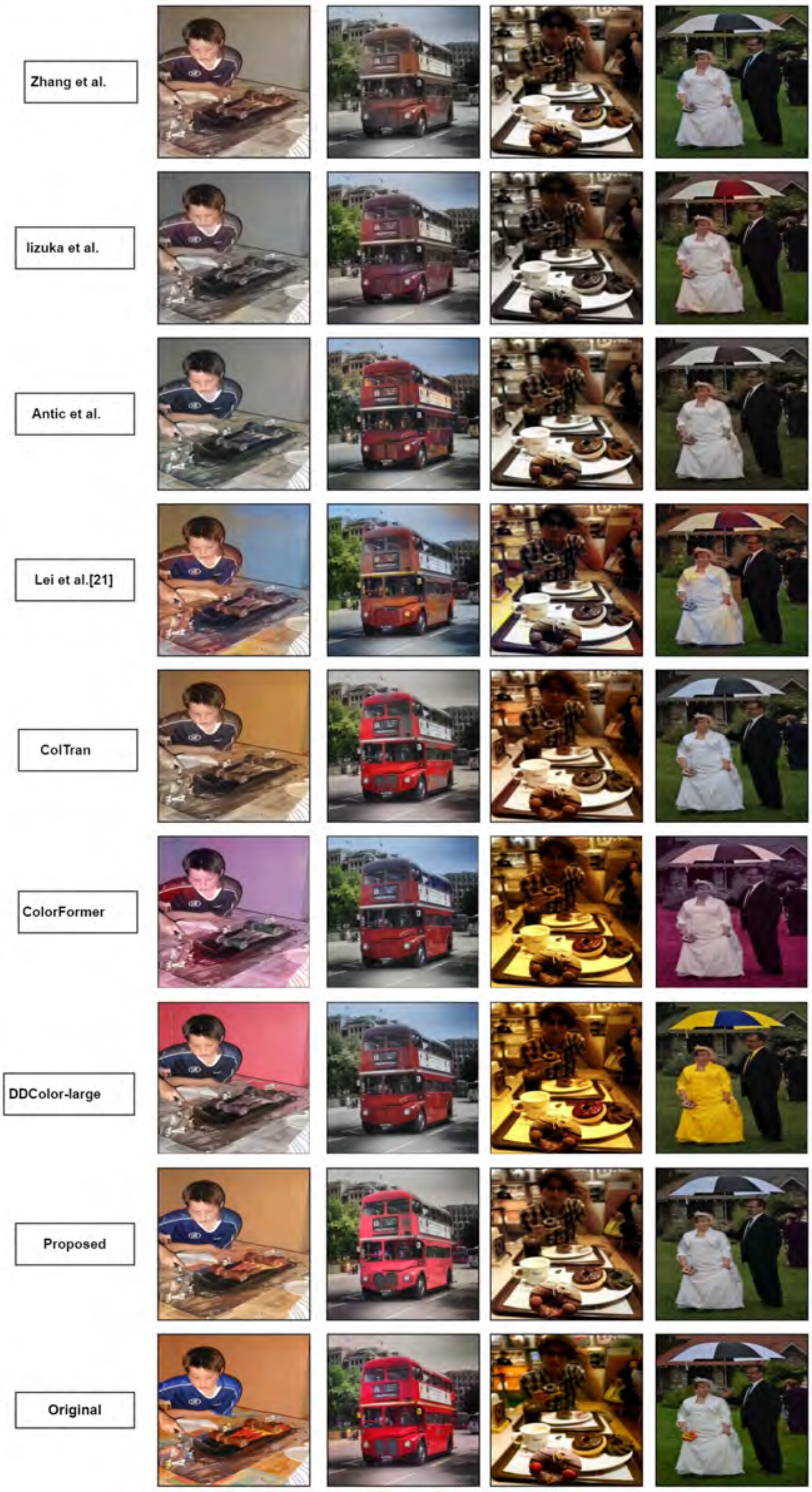


Figure 3.4: Qualitative comparison results of the proposed algorithm with existing colorization algorithms in COCO-Stuff datasets [Best visible in 300% zoom].

Table 3.1: Quantitative comparison of results between the proposed algorithm and existing colorization algorithms on the COCO-Stuff datasets.

	Params.	LPIPS ↓	PSNR ↑	SSIM ↑	FID ↓
Zhang et al. [99]	32.2M	0.234	21.838	0.895	19.17
Iizuka et al. [29]	25.6M	0.185	23.860	0.922	7.63
Antic et al. [2]	63.6M	0.180	23.692	0.920	3.87
Lei et al. [44]	21.6M	0.191	24.588	0.922	12.63
ColTran [40]	74.0M	0.184	23.696	0.922	6.14
ColorFormer [32]	44.8M	0.183	0.882	39.76	1.24
DDColor-large [33]	227.0M	0.190	23.74	0.927	**0.96**
Ours	9.84M	**0.180**	**24.982**	**0.929**	3.62

Table 3.2: Quantitative ablation studies on the COCO-Stuff datasets.

	LPIPS ↓	PSNR ↑	SSIM ↑	FID ↓
Without Lamda	0.190	24.371	0.919	4.46
Self Attention	0.187	24.890	0.918	4.98
Cross Attention	0.183	24.879	0.922	4.08
With Lambda(ours)	**0.180**	**24.982**	**0.929**	**3.62**

benchmarking against models such as Zhang et al.'s [99], Iizuka et al.'s [29], DeOldify [2], Lei et al.'s [44], Kumar et al.'s [40], ColorFormer [32], and DDColor-large [33] . Our method outperforms others across all metrics, including LPIPS [28], SSIM [85], PSNR, and FID [24]. Detailed values can be found in Table 3.1. ,

Furthermore, we compared our visual results with the ground truth for additional insights. It's evident that our method surpasses others in terms of visual fidelity and accuracy. In Figure 3.2, we observe accurate color generation, notably in the first image where the sky and forest hues are faithfully reproduced. In the subsequent row's first image, wooden furniture colors are accurately mimicked, while the fire hydrant's color is more pronounced in the second image. Similarly, in the first image of the fourth row, the camel's color is faithfully rendered, and the second image exhibits a more natural color palette. The Natural colored Dataset results are in Figure 3.3. Overall, our model demonstrates precise color reproduction across various elements, enhancing the fidelity and realism of the generated images.

3.3.3 Ablation

In this study, we investigate the optimal integration of the Lambda network in the image colorization process through a series of ablation studies. We conduct experiments

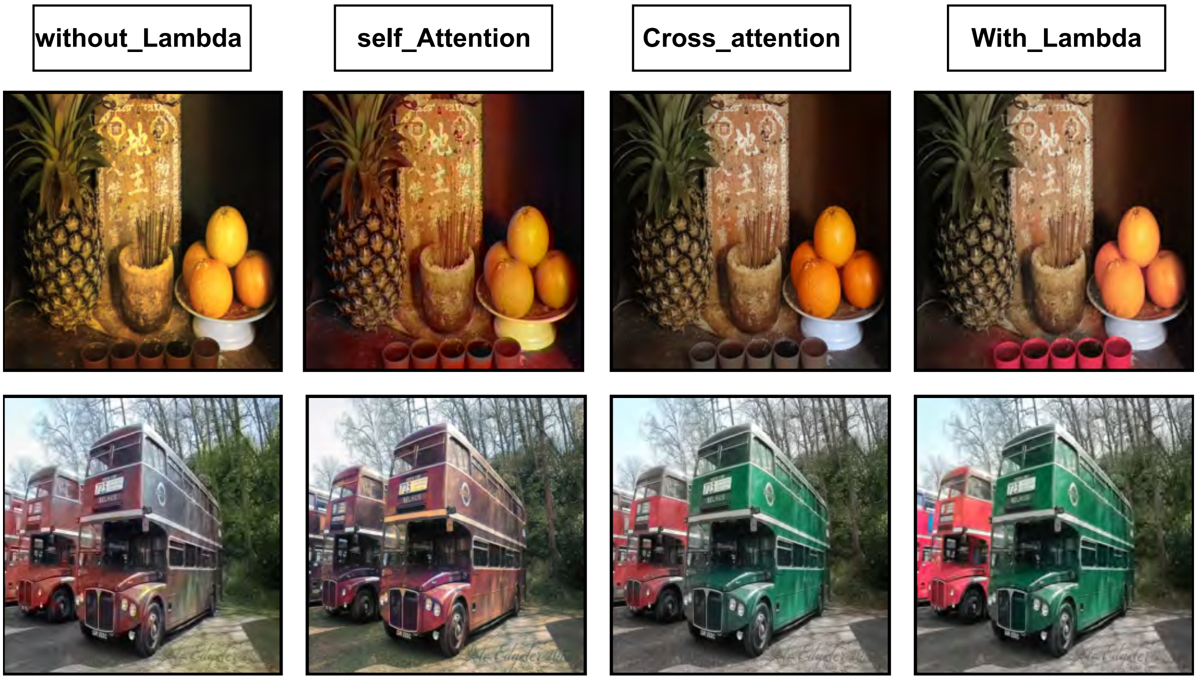


Figure 3.5: Qualitative results of ablation studies on the COCO-Stuff datasets [Best visible in 300% zoom].

comparing different configurations: without the Lambda layer, with self-attention, with cross-attention, and with the Lambda layer. Additionally, we present qualitative results in Figure 3.5 to showcase the impact of these configurations on image quality. Our findings demonstrate that the inclusion of the Lambda layer consistently outperforms across all four metrics evaluated in Table 3.2. This suggests that the Lambda layer plays a crucial role in improving image quality in the context of image colorization tasks. We design a series of experiments to evaluate different configurations of the colorization model: Model without the Lambda layer, Model with the self-attention mechanism, Model with cross-attention mechanism, and Model with the Lambda layer. Our experimental results indicate that the inclusion of the Lambda layer consistently leads to improved performance across all four metrics compared to the other configurations. Additionally, qualitative analysis of the colorized images in Figure 3.5 further supports this finding, demonstrating that the Lambda layer contributes to enhancing image quality in terms of color fidelity and realism.

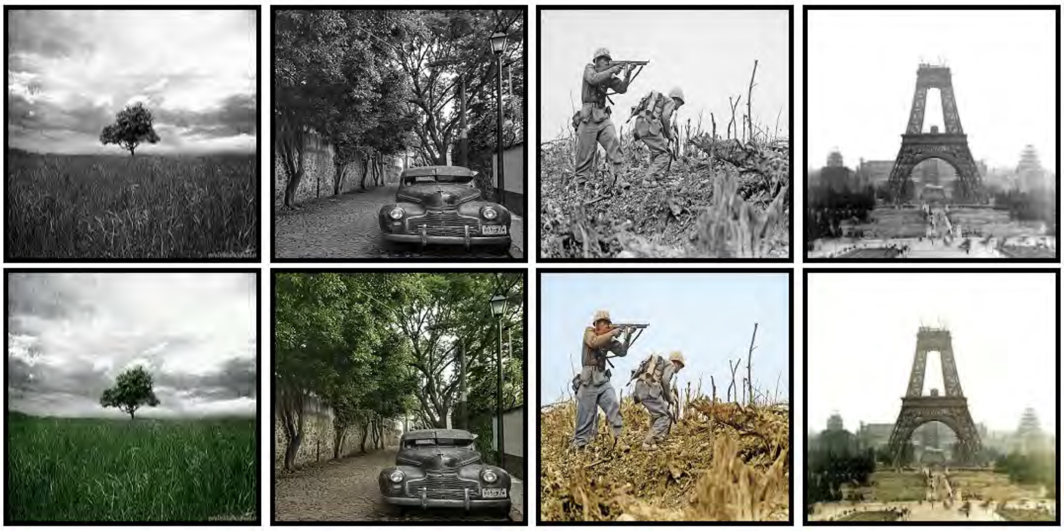


Figure 3.6: Colorized old photos using the proposed method. The top Row is the original one, and the Bottom row is the generated image.

3.3.4 Result on Old Images

We have successfully generated color images from real historical black-and-white photos. Our method effectively colorizes such images, providing a natural and realistic color composition. We have added some colorized old photos in Fig 3.6 to have an idea about the results. Our method has certain limitations when it comes to the varying distribution of luminance channels.

3.3.5 User study

To evaluate the quality of our image generation, we performed a user study using 23 images (18 generated images and 5 real images) from the COCO-Stuff and NCD datasets. The observation test included a mix of generated and real photos, selected and shuffled randomly. Thirty-six people participated in the user study. Our method achieved a score of 68%, indicating that 68% of the generated images are marked wrongly, demonstrating the high realism of our generated images.

3.3.6 Failure Case

We performed some extensive studies on the channel distribution problem in old photos. We also find that the luminance channel mostly affects reduced contrast, noise, and blur problems. Our method performs poorly in the section, as in recent photos. We also

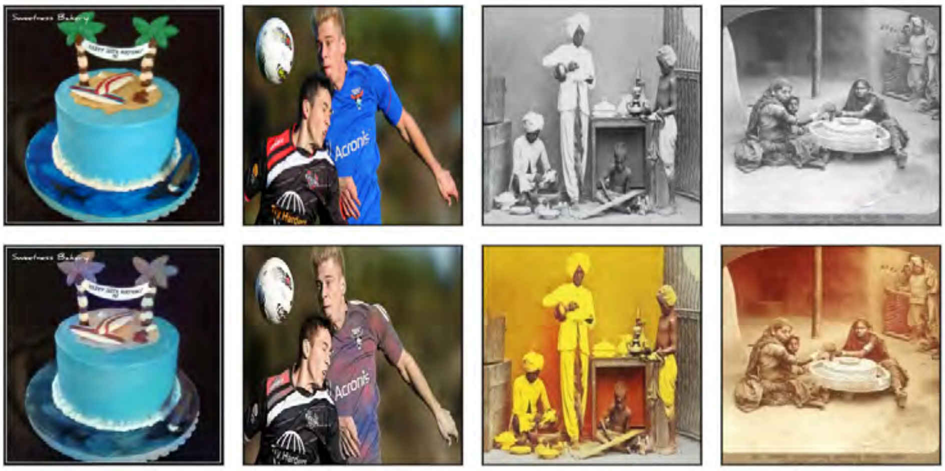


Figure 3.7: Some examples of failure case. The top Row is the original one, and the Bottom row is generated.

included some failure cases caused by the different luminance channel distributions in Fig 3.7. We will make a more prominent effort to try this in the future.

3.4 Conclusion

In summary, our investigation delved into various aspects of image colorization methods. We discussed the utilization of standard datasets such as COCO and natural color datasets for evaluation. Additionally, we compared our model against state-of-the-art techniques, highlighting its superiority across metrics such as LPIPS, SSIM, PSNR, and FID. Visual comparisons against the ground truth further underscored the effectiveness of our approach, particularly in accurately reproducing colors for diverse elements. Furthermore, ablation studies emphasized the significance of incorporating the Lambda layer, showcasing its efficacy in enhancing long-range dependencies crucial for image colorization tasks. Overall, these findings underscore the robustness and efficacy of our proposed method in the realm of image colorization. Further research and development in this direction holds the potential to enhance various applications, including medical imaging, satellite imagery analysis, and artistic rendering, ushering in a new era of image processing capabilities.

In the next chapter, we present an innovative approach for transforming grayscale images into realistic, vibrant representations. Our cross-attention-based generative network utilizes high-dimensional features and a multimodal strategy, combining YOLO (You Only Look Once) feature extraction and mask-based features, to accurately colorize images. We also introduce the Cross Attention-Based Attention (CABA) module to assign suitable colors to different parts of the image to enhance realism.

CHAPTER 4

Color-YOLO: Revolutionizing Image Colorization Potential using YOLO and Cross-Attention Synergy

Colorization can breathe new life into old photographs, films, or artworks, allowing viewers to see them from a fresh perspective and enhancing their visual impact. In our work, we present an innovative method for transforming grayscale images into vibrant, lifelike representations. We introduce a cross-attention-based generative network designed to infuse these images with colors that closely resemble those found in the real world. Our technique leverages high-dimensional features to convey color information through attention mechanisms, as outlined in this research. Our approach adopts a multimodal strategy to address the inherent ambiguity in colorizing images. In the multimodal strategy, the YOLO (You Only Look Once) feature extraction and mask-based features are used to get the best attention and preserve the color features. We also introduce the Cross Attention-Based Attention module (CABA) to intelligently assign suitable colors to different parts of the image, enhancing the realism of the final output.

4.1 Introduction

A primary challenge in colorization arises from the diverse range of colors that objects in a scene can exhibit, influenced by factors such as lighting, texture, and more. Consider, for instance, how a person's skin tone may shift under different lighting conditions or how a landscape can assume distinct appearances based on the time of the day or season. Effectively, colorization algorithms must adeptly encompass these variations, yielding realistic and plausible colorizations. To confront this challenge, researchers have devised an array of colorization techniques, each characterized by its unique architecture, loss functions, and learning strategies. Automatic colorization harnesses machine learning algorithms to predict the colors of a grayscale image. These algorithms undergo training on extensive datasets of colored images, learning associations between specific colors and distinct features or objects. The objective is to create a model capable of accurately predicting colors for an unseen grayscale image based on patterns gleaned from the training data. A prevalent challenge in automatic colorization is the "one-to-many" association problem, where a grayscale image may be linked with multiple equally plausible colorizations. For instance, a black-and-white photo of a sunset might be colored with warm orange and yellow or cool blue and purple, contingent on the artist's interpretation. Researchers have devised various strategies to address these challenges. Some advocate for the use of Generative Adversarial Networks (GANs) [82, 102], where the model learns to generate realistic colorizations by contending with a discriminator distinguishing between real and fake images. Others employ self-supervised learning, training the colorization model to predict the original color image from a degraded or low-resolution version of the same image.

To tackle this issue, we propose an automatic colorization that colors the image using the object and mask-based features. Our method generates more color and more accurate color compared to natural color images. Our key contributions are as follows:

1. Our proposed method is a novel colorization network that uses multi-modal feature attention to color images more accurately.
2. Our method of end-to-end model architecture produces color variations with diverse structures, shapes, and hues.
3. We proposed incorporating additional sources of information, such as semantic segmentation maps and object recognition algorithm features. This augmentation guides the colorization process, fostering more consistent and accurate results.

4.2 Proposed Framework

The proposed method we present operates by taking a grayscale image and an accompanying object image detection mask, and the semantic segmentation-wise high-end feature is used for the image-based cross-attention.

4.2.1 Object Based Feature Extraction

To extract the features of the image, we used the YOLO v5 [103] model for better feature representation. The model uses CSP-Darknet53 [60] as the backbone. Cross-stage Partial network (CSP) is used to overcome the vanishing gradient problem. We used the feature vector from the neck of the architecture. The output of the feature vector size is [512,8,8]. These vectors are pivotal in our generator model, facilitating heightened attention to object color information. This approach ensures a robust representation of image features, enhancing the colorization process with richer contextual understanding. By leveraging the YOLO v5 model, we optimize feature extraction, thereby improving the overall quality and fidelity of colorization results, bolstering the model's capability to accurately capture and reproduce intricate color details within images. We used it on our generator for better attention to object color information.

4.2.2 Masked Based Feature Extraction

In the mask extraction section, we extract the mask information from the Mask R-CNN [22] and create an image that contains identical in size to the originals but with distinctive boundary details. This mask-derived feature is instrumental in enriching object-specific color information, facilitating superior object-colorization accuracy. By leveraging Mask R-CNN, we enhance our ability to discern object boundaries and apply nuanced colorization, resulting in enhanced visual fidelity and realism. This approach ensures that each object receives optimal color treatment, contributing to the overall quality and coherence of the colorization process, thereby enriching the final output with finer details and improved object-color alignment. This mask-based feature helps us to color each object for better color information.

4.2.3 Cross Attention

Cross-attention [19] is not widely used for colorization tasks. We applied the cross attention to different parts of the image to extract relevant information like color, structure, etc. given image patches $X = x_1, x_2 .....x_n$ where n the number of patch and x_i repesents the feature of each image patch. Quarry, key & Value (Q, K, V) are calculated as follows:

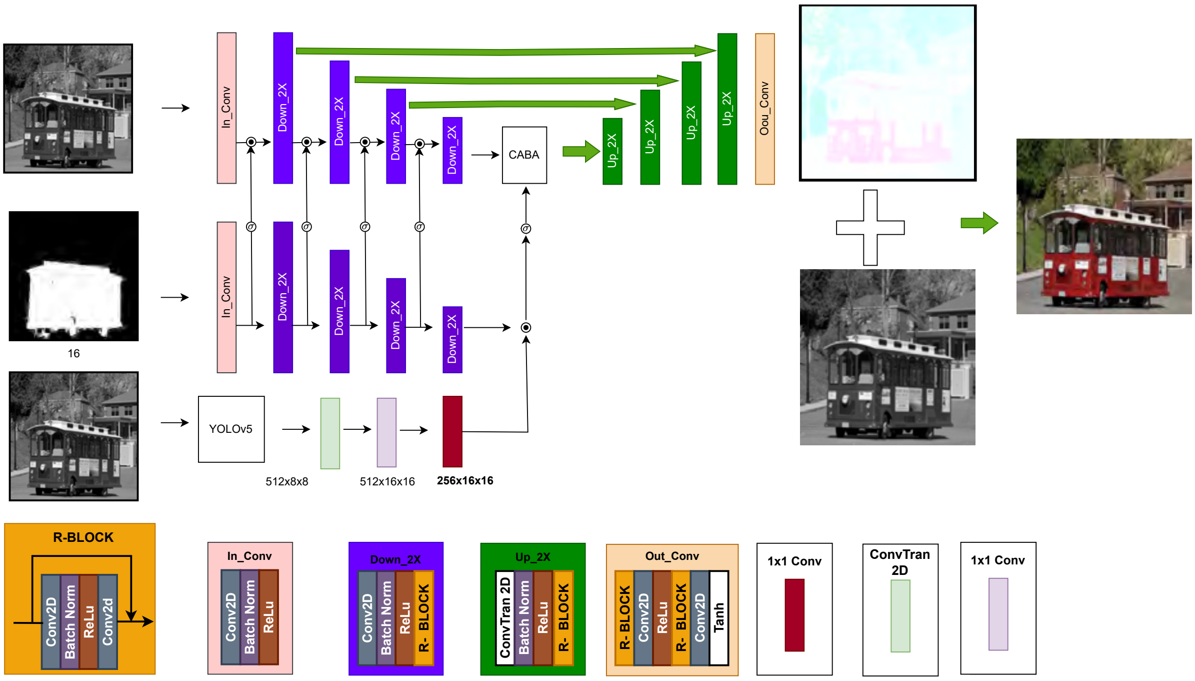


Figure 4.1: Diagram of the proposed network.

$Q = XW_q$
$K = XW_k$
$V = XW_v$
where W_q,W_k and W_v are the weight metrics to be generated from the image patch. d_k is the dimensionality of the key vectors.

The attention is then applied to the patch:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V$$

4.2.4 Generator

The proposed generator architecture comprises three inputs. The first input is the L image (L_i) extracted from the LAB color space. The second input is generated from the Mask R-CNN [22] output(M_i), facilitating the differentiation of colors for each object. Lastly, the third input (O_i) comprises features obtained from the pre-trained YOLO [103] architecture. The dimensions of the network's input sizes are L_i $(256 \times 256 \times 1)$, M_i $(256 \times 256 \times 1)$, and O_i $(512 \times 8 \times 8)$. In the L_i input section, we first apply the In_{convo} module, which contains one convolution, batch norms, and ReLU, respectively. Parallel, we compute the same In_{convo} module for the mask input M_i and get the output from the

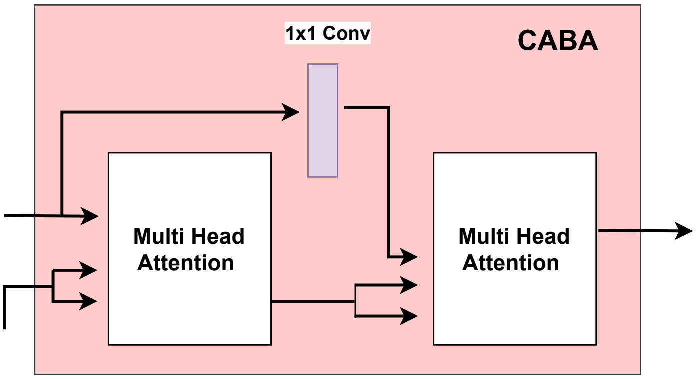


Figure 4.2: Diagram of the Cross Attention-Based Attention module (CABA).

In_{convo} we apply the sigmoid activation on it and multiply with the same size features computed by the L_i input. We apply the same attention with sigmoid up to the height dimensional encoder features. In the encoder section, we used the $Down_2X$ module four times. As shown in Fig. 4.1, The $Down_2X$ contains the one convo2d, Batchnorm, ReLU and one $R-Block$, respectively. $R-Block$ contains the first convolution 2D and BatchNorm, ReLU and last convolution 2D. One skip connection is connected from input to output in the $R-Block$, which preserves the residual connection features to the architecture. The attention of the L_i encoder block is applied from the M_i input block. The four attention modules are used after each $Down_2X$ module.
In the O_i section, the input is computed as the features matrix generated from the Yolo architecture. First, we apply one convolution transposed layer with stride by 2 and 512 filters. The generated the output is (512 × 16 × 16). After that, we use the (1 × 1) convolution block with 256 filters. One convolution layer is used to refine the feature, which contains the 256 filter with point convolution.

After getting the two feature vectors from the M_i and O_i, we multiply with the two vectors to enhance the feature attention. The size of the vector will be [256 × 16 × 16]. We introduce Cross Attention-Based Attention using the feature vector into the high-label features of the encoder architecture of L_i input. After that, we design four UP_2X modules in the decoder section. In the decoder section, a skip connection is connected to get the encoder features from the L_i encoder in each step. The UP_2X contains one transpose convolution followed by BatchNorm, ReLu, and a $R-block$. In the last layer, we add an out convolution block that has six layers: two $R-Block$, two convolution 2D, a Relu layer, and a tanh activation layer. The block diagram of the proposed network is in Fig 4.1.

4.2.5 Cross Attention-Based Attention Module

The proposed Cross Attention-Based Attention module (CABA) module is a cross-attention-based module grounded in multi-head attention mechanisms. The fundamental cross-attention module incorporates three essential components: query, key, and value. We introduced a novel enhancement of point convolution into the key segment. Furthermore, we augmented this layer with an additional color feature, facilitating color-based attention mechanisms. The block diagram of the module is in Fig 4.2.

4.2.6 Discriminator

To create the input for the discriminator, the grayscale input image (L_i) is combined with either a target image (T_i) or an estimated image (E_i), where T_i and E_i are the AB channels of the color image. During the training of the discriminator, the (L_i,T_i) stack is labeled as real, while the (L_i,E_i) stack is labeled as fake. This approach ensures that the generated images are of superior quality and can be used in a variety of applications. By using PatchGan-based discriminators, the modules can evaluate the image quality at the patch level, which allows for greater accuracy in assessing image quality. Additionally, using multiple discriminators further enhances the evaluation process and helps to ensure that the generated images are of the highest possible quality. whereas the GAN loss for the discriminator D_m is defined as

$$\begin{aligned}\mathscr{L}_{GAN}^{D_m} =& \mathscr{L}_{BCE}(D_m(L_i,T_i),1) \\ &+ \mathscr{L}_{BCE}(D_m(L_i,G_m(L_i,M_i,O_i),0)\end{aligned} \tag{4.1}$$

4.2.7 Loss

To train the network, we used four loss types to better visualise the network result. Huber loss, Gan Loss, MAE loss, and SSIM Loss are used to estimate the result.

4.2.7.1 Huber Loss:

Huber loss is used in the network to smooth the MSE loss. The loss function is a piecewise function that acts as the MSE for small errors that are less than or equal to a certain threshold $delta$ and like MAE for larger errors. In the equation, y_i and $\hat{y}_i$ are the pixel values of the original and generated images.

$$\mathscr{L}_{\text{Huber}}(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}\frac{1}{2}(y_i-\hat{y}_i)^2, & \text{if } |y_i-\hat{y}_i| \leq \delta \\ \delta(|y_i-\hat{y}_i|-\frac{1}{2}\delta), & \text{otherwise}\end{cases} \tag{4.2}$$

4.2.7.2 GAN Loss:

We define the GAN loss $\mathscr{L}^{G_m}$ of the generator G_m as

$$\mathscr{L}^{G_m}_{GAN} = \mathscr{L}_{BCE}(D_m(L^i, G(L_i, M_i, O_i)), 1) \tag{4.3}$$

4.2.7.3 MAE Loss:

For overall pixel-level fidelity, we consider the MAE (L_1) loss for the generation process. The L_1 loss for the generator is calculated as:

$$\mathscr{L}^{G}_{1} = \|E^i - T^i\|_1 = \|G_m(L_i, M_i, O_i) - T_i\|_1 \tag{4.4}$$

where G_m indicates the generator of module m that maps L_i and M_i, O_i to E_i.

4.2.7.4 SSIM Loss:

$$\text{SSIM}(T_i, E_i) = \frac{(2\mu_i^T \mu_i^E + c_1)(2\sigma_{(T_i)(E_i)} + c_2)}{((\mu_i^T)^2 + (\mu_i^E)^2 + c_1)((\sigma_i^T)^2 + (\sigma_i^E)^2 + c_2)} \tag{4.5}$$

μ_i^T and μ_i^E are means of T_i and E_i respectively. $\sigma_{(T_i)(E_i)}$ is the covariance between the T_i and E_i. C_1 and C_2 are constants to stabilize the division. This equation calculates the SSIM index, indicating the similarity between two images based on luminance, contrast, and structure.

4.3 Experiments Details, Results and Analysis

4.3.1 Dataset

To evaluate the results, we employed two datasets to understand the proposed colorization method's ability better. The first is MS-COCO [49], and the other is ImageNet [14]. The MS-COCO dataset contains the common scene image, where approximately 190 categories of objects are in the whole dataset. The dataset is mainly used in semantic segmentation and object detection. In contrast, the ImageNet dataset is used for the classification task. The dataset has 1k class labels. We include the two datasets with various color images and all the objects in a state-of-the-art dataset. The MS-COCO data set contains 118k images in the training set and 5k in the validation set. In the ImageNet data set, we used 1.4M images for training and 50k for testing.

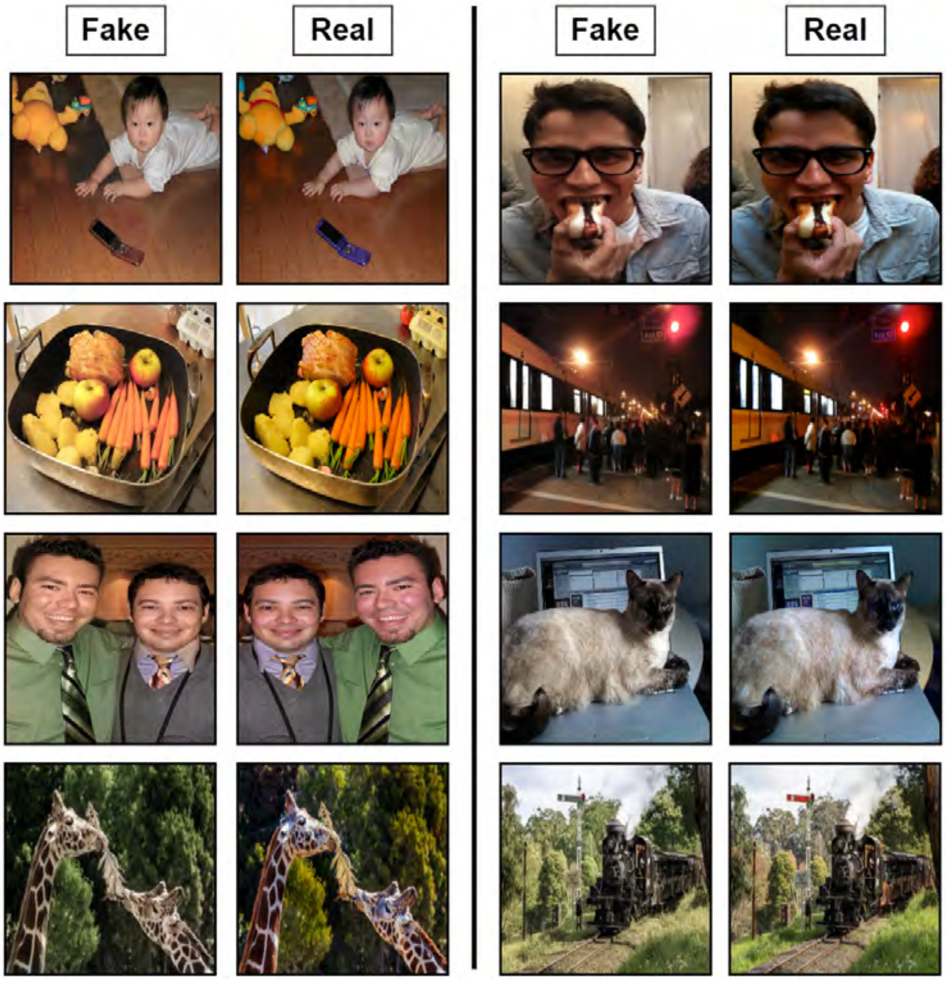


Figure 4.3: Examples of some qualitative results generated from the COCO-Stuff dataset by the proposed framework. Here, "FAKE' means the Images generated by our method, and "REAL" means the Original images in the dataset [Best visible in 300% zoom].

4.3.2 Qualitative result

To assess the effectiveness of our new method, we produced numerous images using the suggested framework. A selection of these colorized image samples is presented in Figs. 4.3 & 4.4. These results illustrate the capability of our method in efficiently colorizing intricate scenes and adeptly handling scenarios with multiple objects, occlusions, and shadows.

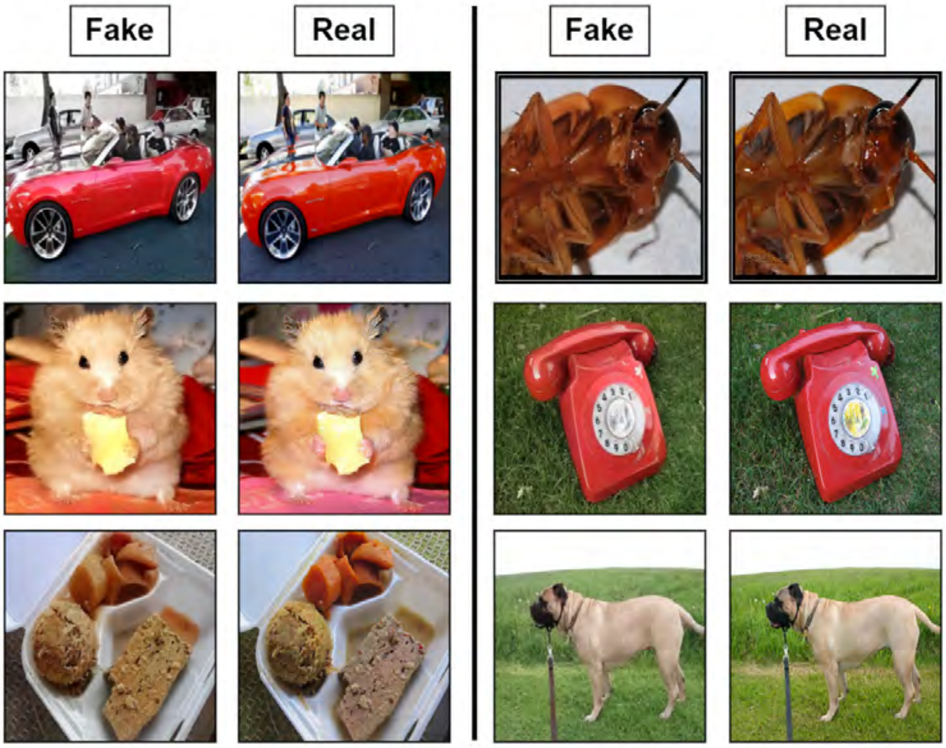


Figure 4.4: Examples of some qualitative results generated from the Image Net dataset by the proposed framework. Here, "FAKE' means the Images generated by our method, and "REAL" means the Original images in the dataset [Best visible in 300% zoom].

4.3.3 Comparison of Results

To affirm the efficacy of our method, we conducted qualitative and quantitative assessments. Table 4.1 compares PSNR, SSIM, the Colorfulness score(CF) [21] and FID [24] metrics between our proposed framework and existing colorization methods such as CIC [99], InsColor [72], DeOldify [2], Wu et al. [92], ColTran [40], CT2 [89], BigColor [35], ColorFormer [32], and DDColor-large [33]. We found that our method attained the best performance on two metrics in the ImageNet data set and two in the cocostuff dataset. In table 4.1, our method exhibits strong quantitative performance across eight metrics values in two different datasets. It secures the top position in four of the eight metrics of two datasets. Additionally, we highlight that our approach outperforms other methods, with the last method, DD color Large [33], achieving the three best results in two datasets. For the quantitative comparison, we have evaluated our method against several state-of-the-art techniques. See in Fig 4.5.

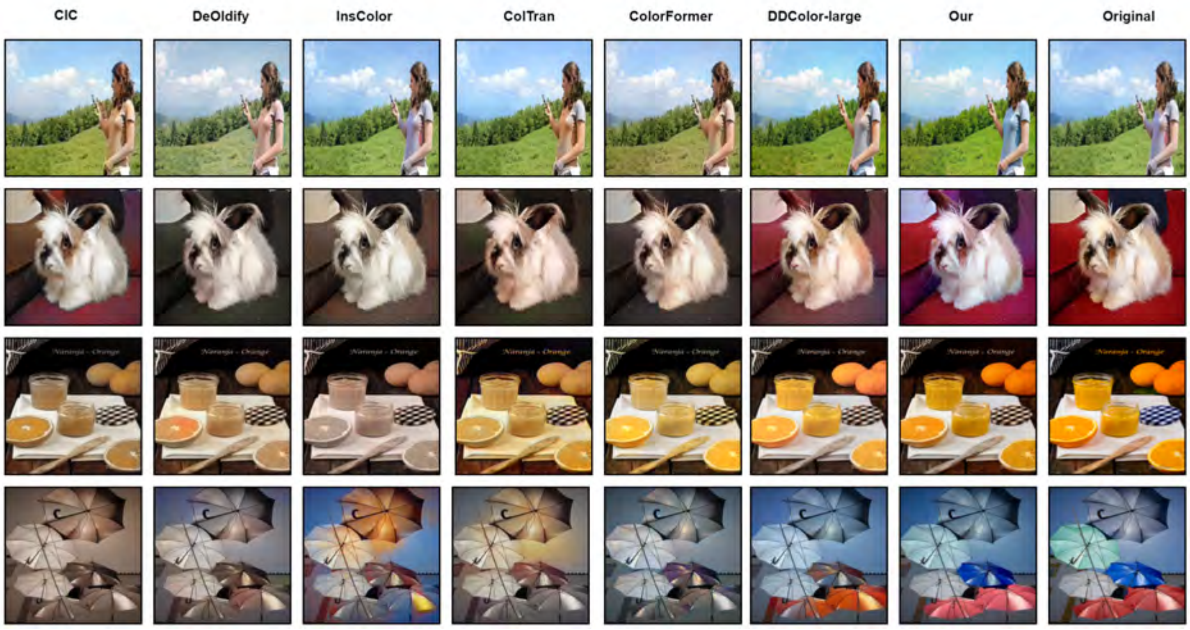


Figure 4.5: Qualitative Comparison results of the proposed algorithm with existing colorization algorithms [Best visible in 300% zoom].

Table 4.1: Comparison results of the proposed algorithm with existing colorization algorithms.

	COCO-Stuff				ImageNet(Val 50k)			
	PSNR ↑	SSIM ↑	CF score ↑	FID↓	PSNR ↑	SSIM ↑	CF score ↑	FID↓
CIC [99]	20.86	0.896	**43.92**	19.17	22.73	0.662	33.84	27.88
InsColor [72]	22.91	0.899	27.05	7.36	23.38	0.673	27.45	13.09
DeOldify [2]	22.97	0.893	22.83	3.87	**24.19**	0.634	24.99	13.09
Wu er.al. [92]	21.81	0.916	35.13	3.62	-	-	-	-
ColTran [40]	22.30	0.930	35.50	6.14	21.72	0.692	36.27	14.94
CT2 [89]	22.93	0.912	39.96	4.59	-	-	-	-
BigColor [35]	21.24	0.889	40.01	1.24	-	-	-	-
ColorFormer [32]	23.00	0.882	39.76	1.24	23.91	0.698	36.34	8.68
DDColor-large [33]	23.74	0.927	38.65	**0.96**	22.85	0.710	**38.38**	**5.18**
Our	**23.86**	**0.931**	35.60	1.20	24.10	**0.715**	35.43	**5.18**

Table 4.2: Quantitative comparison uses of CABA block in the colorization module (COCO-Stuff)

Configuration	SSIM ↑	PSNR ↑	CF score ↑	FID ↓
Without CABA (self attention)	0.880	21.32	31.28	2.98
With CABA 2	**0.931**	**23.86**	**35.60**	**1.20**

4.3.4 Ablation Results

In order to enhance the validation process, an ablation study was conducted to gain deeper insights into the results of the proposed model. Specifically, we explored the impact of the CABA block through two ablation setups: one without the CABA block and the other with the CABA block included. To conduct a quantitative analysis of the CABA module specified in Table 4.3, it is indicated that the CABA module operates efficiently and enhances the network's outcomes. Here's a detailed quantitative comparison in Table 4.2 of the different configurations using YOLO, Mask, and CABA blocks in the colorization module on the COCO-Stuff dataset. The metrics considered are SSIM, PSNR, CF score, and FID. The configuration using YOLO, Mask, and CABA together provides the best performance across all metrics, showing substantial improvements in image quality and color fidelity. In contrast, the baseline without CABA performs the worst. We also validate the uses of CABA blocks using different numbers of CABA blocks used in the network. According to Table 4.4 and Figure Fig 4.6, we can conclude that the number of 2 CABA blocks gives the best result of the proper network.

As shown in Table 4.4, the model achieves the highest performance when evaluated using standard reference metrics such as SSIM, PSNR, and FID. SSIM and PSNR indicate how structurally and visually similar the generated image is to the original, with higher values reflecting better quality. FID measures the perceptual realism of generated images compared to real images, where lower values are considered better. Achieving the best scores in all three metrics suggests that the proposed method not only reconstructs images more accurately but also produces visually realistic outputs. This confirms the effectiveness of the proposed setting in improving both fidelity and perceptual quality in image colorization tasks.

4.3.5 User Study

To assess the quality of our image generation, we conducted human observations on 30 images from the COCO-stuff dataset. Both generated and real images were randomly mixed. The 26 participants were asked to determine whether each image was generated

Table 4.3: Quantitative comparison uses of YOLO, Mask and CABA block in the colorization module (COCO-Stuff)

Configuration	SSIM ↑	PSNR ↑	CF score ↑	FID ↓
YOLO+Mask+Self attention	0.770	19.35	34.52	8.39
YOLO+CABA	0.843	19.97	31.96	5.44
Mask+CABA	0.839	20.32	30.28	3.89
YOLO+Mask+CABA	**0.931**	**23.86**	**35.60**	**1.20**

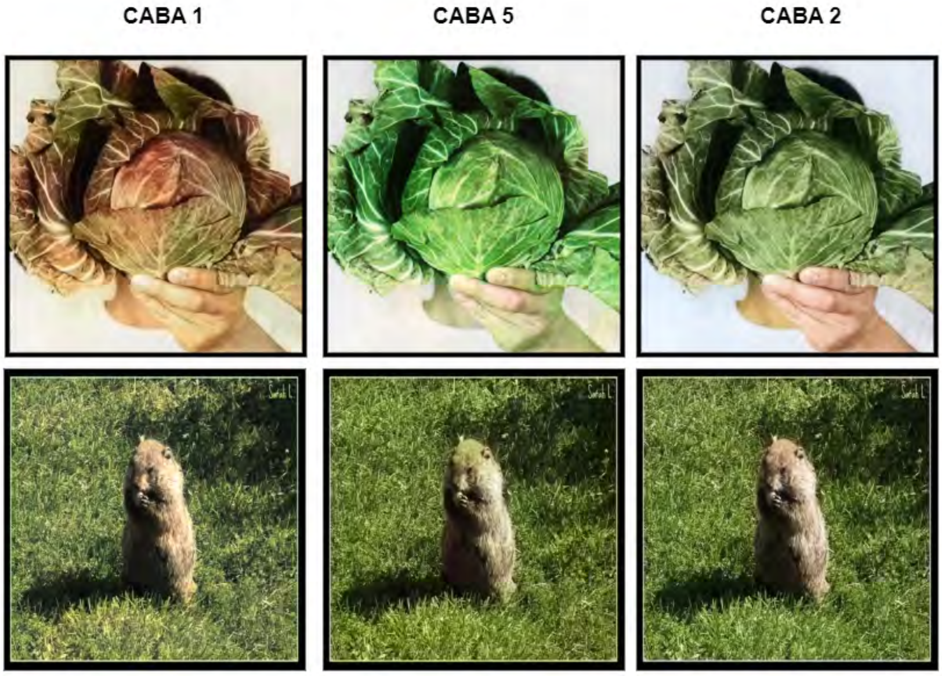


Figure 4.6: Examples of qualitative results of different ablation studies [Best visible in 300% zoom].

Table 4.4: Quantitative comparison among different numbers of CABA in ablation studies (COCO-Stuff)

Configuration	SSIM ↑	PSNR ↑	CF score ↑	FID ↓
CABA 1	0.920	22.99	**37.20**	1.36
CABA 5	0.905	23.13	34.52	1.24
CABA 2	**0.931**	**23.86**	35.60	**1.20**

or real. Our method achieved a percentage score of 71%. This indicates that 71% of the responses needed to be corrected, with participants often mistaking the generated images as real.

4.4 Conclusion

In this work, we introduced an innovative image colorization technique incorporating object detection encoding to enhance the color generation process. Our framework demonstrates superior color accuracy compared to existing algorithms. Notably, our focus lies on CABA conditioning for foreground objects exclusively. While our algorithm surpasses state-of-the-art methods in this context, it exhibits limitations in accurately coloring backgrounds due to the CABA attention primarily emphasizing foreground elements. This issue can be addressed by including detailed color attention for backgrounds. Additionally, we observed discrepancies between the colors generated by our method and the actual ground truths, attributed to the coarse nature of CABA attention. Consequently, our approach may result in less vibrant backgrounds, underscoring the need for comprehensive CABA descriptions in grayscale image colorization endeavors. Future research should aim to develop robust colorization techniques for backgrounds lacking rich CABA attention.

In the next chapter, we explore the integration of textual descriptions as an additional condition, along with the grayscale image, to enhance colorization accuracy. This approach is one of the first to incorporate textual conditioning into the colorization process. A novel deep network is proposed that takes both the grayscale image and its corresponding text description to predict the color gamut, improving color fidelity by leveraging color information from the text.

Chapter 5

Text-Guided Image Colorization

5.1 Introduction

To increase the fidelity in the colorization pipeline, we propose a text-guided colorization pipeline where some color descriptions about the objects present in the grayscale image can be provided as auxiliary conditions to achieve more robust colorized results.
The major contributions of our work are as follows.

- We propose a novel GAN pipeline that exploits textual descriptions as an auxiliary condition.

- We extensively evaluate our framework using both qualitative and quantitative measures. In comparison with the state-of-the-art (SOTA) algorithms, we found that the proposed method generates results with better perceptual quality.

- To the best of our knowledge, this is the first attempt to integrate textual information in the colorization pipeline to improve the quality of generation. The textual color description acts as an additional conditioning to increase the fidelity in the final colorized output.

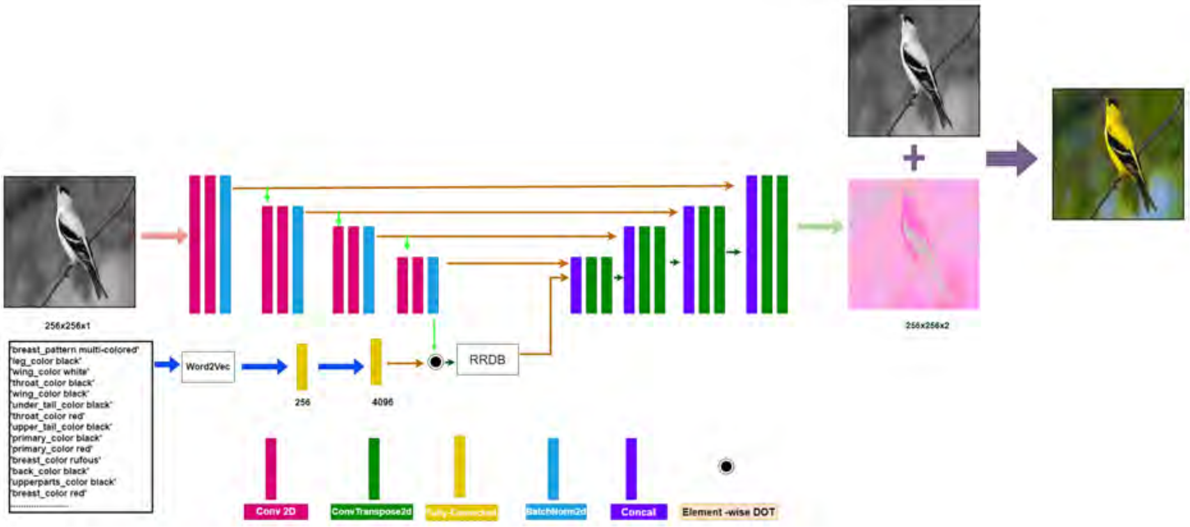


Figure 5.1: The block diagram of the proposed architecture. The network predicts the color components of the image which is combined with the intensity image to produce the final colorized image.

5.2 Proposed Framework

Image colorization aims to generate a color image from a grayscale image. Typically, deep learning tools use RGB images as ground truth for image generation. In the proposed method, RGB images are converted into the CIE LAB color space, where we need to find only the 'A' and 'B' channels instead of three channels of RGB. We converted the input text, containing the color information of the image, to a word vector using the word2vec [54]. The size of the word vector is 256, and the input size of the image is 256×256, which is the 'L' channel of the LAB color space. We add the 'L' channel with the AB channel of the image, which the Generator predicts, to reconstruct a fully colorized image. The discriminator signifies the visual authenticity of the image in a patch-based manner.

5.2.1 Generator

The idea of the proposed Generator is that the text color information is fused with the grayscale image (L^i) at the last downsample step of the network. The input L image is first resized to a fixed size of 256×256. The overall generator has two pathways- an image pathway, through which the image information flows in the network, and the text pathway, through which the text color information flows as a conditional input. Both the pathways finally meets in the Residual in Residual Dense Block (RRDB). For the image

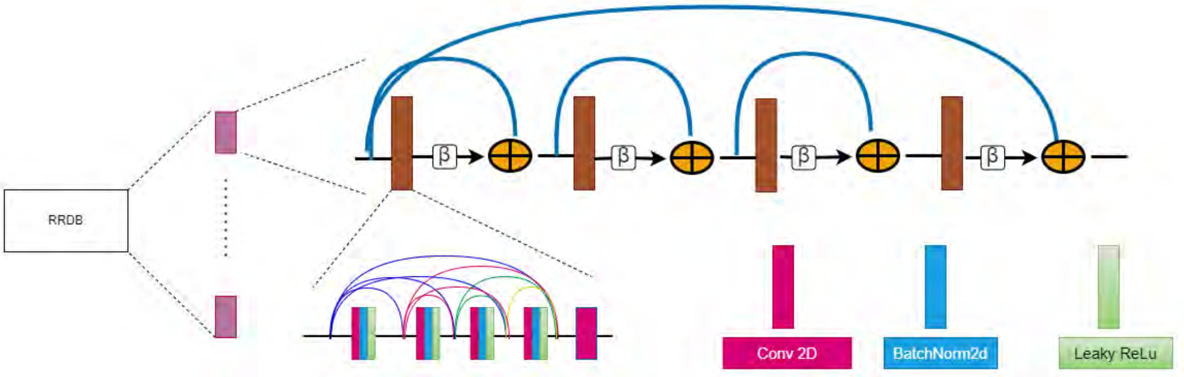


Figure 5.2: The block diagram of the Residual in Residual Dense Block(RRBD) architecture.

path, each resolution level has two convolution layers. The down-sampling follows the last convolution by 2 to move to a new resolution. We use a 3×3 kernel size with 64 filters in each convolution block. After each convolution, we perform batch-normalization [30], and each convolution block has ReLU [56] activation. We also process the text vector(S^i) by two fully connected layers of sizes 256 and 4096. We resize the text features computed by the last fully connected layer to $1 \times 64 \times 64$ and perform an element-wise dot product between the image features and the text features to impose a text-guided conditioning. The text conditioned features are then fed to a Residual in Residual Dense Block (RRBD) [82] before forwarding to the expanding part of the generator. The RRBD block consists of several dense layers with skip connections. The output of each dense block is scaled by β before feeding it to the next dense block. Each Dense block consists of a convolutional layer, followed by BN and leaky ReLU activation with the residual connection. As shown in Fig. 5.2, skip connections are introduced to tackle the problem of a vanishing gradient. The output of the RRBD block is used as input to the convTranspose2d layer with a 64 filter. In the expanding pathway, we have three up-sampling oparations that work in four different resolutions. To increase the feature information in the expanding path, after each up-sampling layer, we concatenate the features available with the same resolution in the contracting path. The convolution blocks in the expanding path are similar to the convolution blocks at the contracting paths, and we decrease the number of filters by two as we move to the higher resolution. At the highest resolution, we apply two filters with kernel size 1×1 to generate the estimated AB channel of the color image. At the end of the proposed network, we compute the color image by adding the generated AB and the input grayscale image(L^i). We illustrate the proposed Generator in Fig 5.1.

5.2.2 Discriminator

For the colorization task, it is required that the discriminator can detect the local quality of a generated colorized image. Thus, we use the PatchGAN [31] Discriminator D to judge the quality of the generated image. The discriminator penalizes the generated structure at the patch level resulting in a high-quality single level generation. We stack the grayscale image (L^i) with either a target image (T^i) or with a estimated image (E^i) where T^i and E^i are the AB channel of the color image. The (L^i,T^i) stack is labeled as real and the (L^i,E^i) stack is labeled as fake. In our model, the Patch discriminator takes a three-channel input dimension 256×256. The discriminator has three convolution blocks with 64, 128 and 256 filters, respectively, in each block with filter dimension 4×4. In the first two convolution blocks, the filter has stride 2, whereas, for the last two blocks, we used stride 1×1. Each convolution layer is followed by batch-normalization [30] and leaky-ReLU [52] activation. After the convolution blocks, we apply one filter of kernel size 4×4 with stride 1 to compute the final response. The average of the final response is the output of the discriminator.

5.2.3 Training

As mentioned in [31], the PatchGAN discriminator focuses more on the high frequency information. Thus to keep the fidelity of low frequency information in the colorized image, we used L_1 loss in the generator G which is calculated as

$$\mathcal{L}_1^G = \|E^i - T^i\|_1 = \|G(L^i, S^i) - T^i\|_1 \tag{5.1}$$

As we have trained the generator in an adversarial manner, we define the adversarial or the GAN loss of the generator and the discriminator as:

$$\mathcal{L}_{GAN}^G = \mathcal{L}_{BCE}(D(L^i, G(L^i, S^i)), 1) \tag{5.2}$$

$$\mathcal{L}_{GAN}^D = \mathcal{L}_{BCE}(D(L^i, T^i), 1) + \mathcal{L}_{BCE}(D(L^i, G(L^i, S^i)), 0)$$

where L_{GAN}^G and L_{GAN}^D denote adversarial Generator loss adversarial discriminator loss, respectively. To increase the visual quality of the image, we use perceptual loss to train the generator.

$$\mathcal{L}_{p_\rho}^G = \frac{1}{h_\rho w_\rho c_\rho} \sum_{x=1}^{h_\rho} \sum_{y=1}^{w_\rho} \sum_{z=1}^{c_\rho} \|\phi_\rho(E^i) - \phi_\rho(T^i)\|_1 \tag{5.3}$$

where $\mathscr{L}^G_{p_\rho}$ is the perceptual loss computed at the ρ^{th} layer, ϕ_ρ is the output from the ρ^{th} layer of a pretrained VGG19 [66] model, and h_ρ, w_ρ and c_ρ are the height, width and the number of channels at that layer, respectively.
The total generator loss $\mathscr{L}^G$ can be defined as

$$\mathscr{L}^G = arg \min_G \max_D \ \lambda_1 \mathscr{L}^G_{GAN} + \lambda_2 \mathscr{L}^G_{P_4} + \lambda_3 \mathscr{L}^G_1 \tag{5.4}$$

We provided a detailed analysis of the roles and effectiveness of L1 loss and the standard Binary Cross Entropy (BCE) loss used in the training of Generative Adversarial Networks (GANs). L1 loss, also known as mean absolute error, is commonly used to enforce pixel-wise similarity between the generated image and the ground truth. It helps preserve image structure and ensures that the generated output does not deviate significantly from the target image. On the other hand, BCE loss is typically used in the discriminator during GAN training to distinguish between real and fake images. It guides the generator to produce more realistic outputs by penalizing differences in probability predictions. This combination is particularly relevant in image colorization tasks, where both structural accuracy and visual authenticity are essential. The experiments and observations presented in this sub-section justify the choice of loss functions in the proposed framework.

5.3 Experiments Details, Results and Analysis

We use the PyTorch framework to build the model, and perform our experiments. We use the Adam [36] optimizer to train both the generator and discriminator up to 350K iterations with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is 1×10^{-4} with a decay of 0. All the leaky-ReLU activations have negative slope coefficients of 0.2. We select $\lambda_1 = 1$, $\lambda_2 = 1$ and $\lambda_3 = 1$.

While training the discriminator D, we concatenate L^i with either T^i or E^i and give that as an input. Both D and G are trained iteratively, *i.e.*, we keep D fixed while training G and vice versa. As the training process of GAN is highly stochastic, we store the network weights at the end of each iteration. At the time of inference, we drop the discriminator network and generate the A,B channels only using the generator network.

'breast_pattern multi-colored'
'leg_color black'
'wing_color white'
'throat_color black'
'wing_color black'
'under_tail_color black'
'throat_color red'
'upper_tail_color black'
'primary_color black'
'primary_color red'
'breast_color rufous'
'back_color black'
'upperparts_color black'
'breast_color red'

'upperparts_color white'
'back_color white'
'belly_color white'
'wing_pattern multi-colored'
'leg_color grey'
'underparts_color white'
'crown_color rufous'
'eye_color black'
'forehead_color rufous'
'underparts_color red'
'tail_pattern multi-colored'
'nape_color red'
'forehead_color red'
'breast_color white'

Figure 5.3: A typical example in our dataset: each sample contains a color image and corresponding color descriptions of the bird. To use the image while training the network, we first convert the color image to LAB color space, and use the 'L' image as the input.

5.3.1 Datasets

To evaluate the performance of our model, we use three popular datasets, Caltech-UCSD Birds 200 [87], MS COCO [49] and Natural color Dataset(NCD) [3].
The Birds dataset contains 6032 bird images with their color information. We split the dataset into two parts (train, test). Total number of images for training is 5032, and the remaining images are used in the test set.A typical example of the dataset is in Fig5.3.

The total number of images in the NCD set is 730 fruit images. We use 600 images for training and the remaining 130 images for testing. We converted the class label to one single color, like the tomato's class is converted into red and used as color information for training and testing.

From the MS COCO [49] dataset, we use 39k images for training and 6225 images for

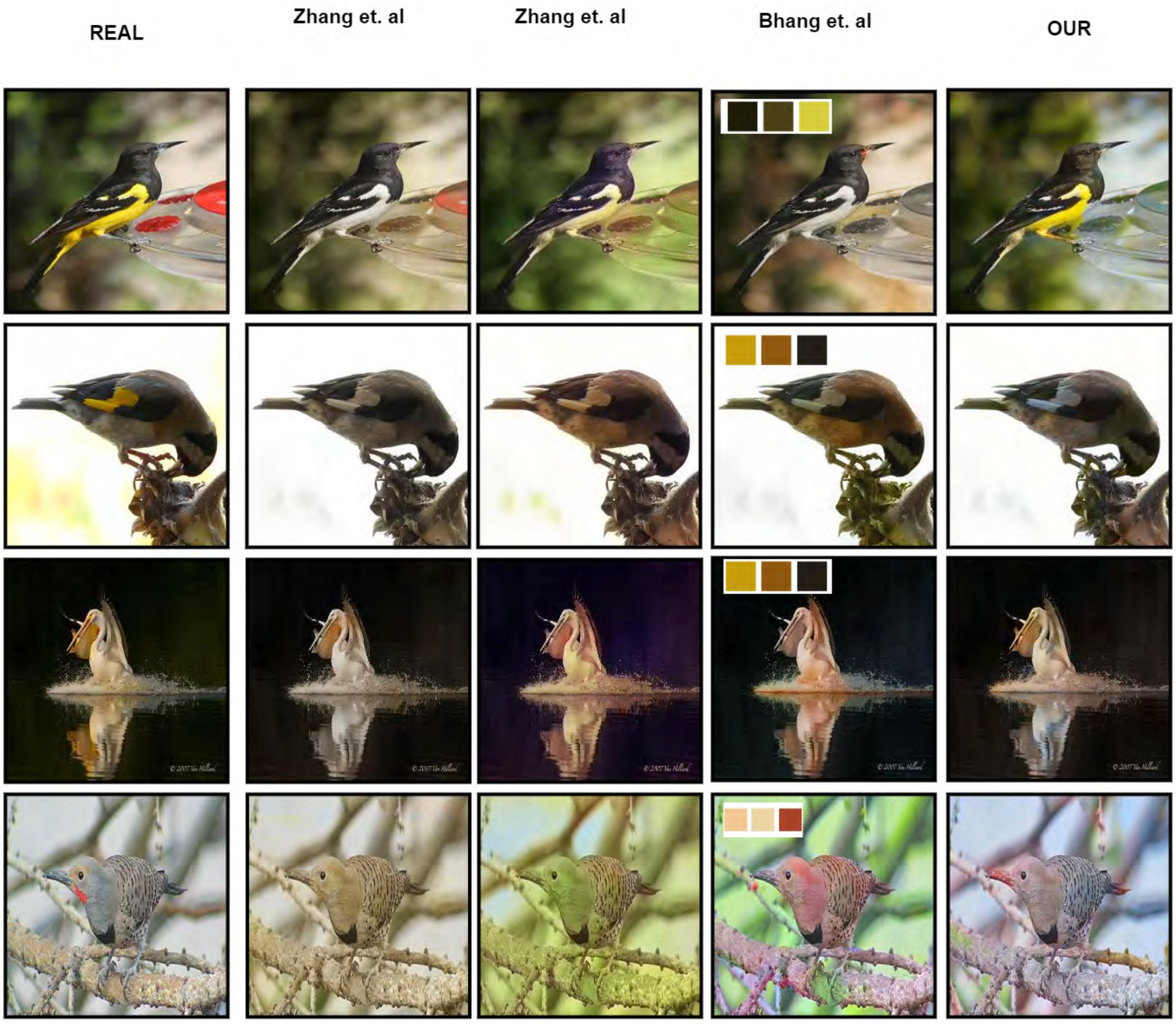


Figure 5.4: Qualitative comparison results: The first column contains ground truth images, the second column, third and fourth columns contain the results generated by the SOTA algorithms, and the last column shows the results generated by the proposed algorithm.

testing. In COCO stuff [7], the text description of the images are available. In each text description, we find the sentence(s) related to color information of an object to provide it as auxiliary information. We collect all such sentences and use it as the final auxiliary information for the respective image.

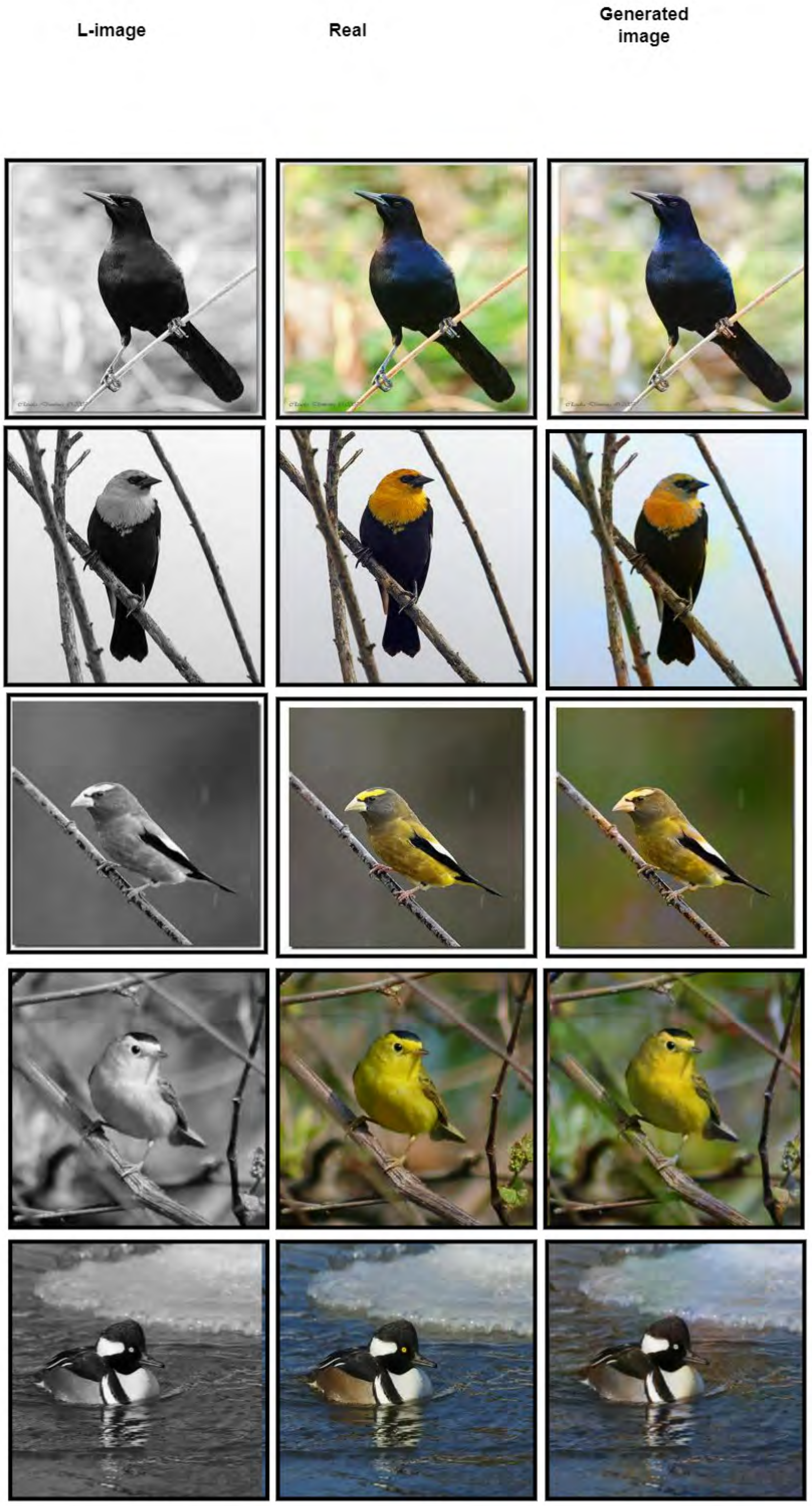


Figure 5.5: Images generated by the proposed algorithm from the Caltech-UCSD Birds 200 [87]: the first column shows the grayscale images, the second column shows the ground truth images and third column shows the colorized outputs of the proposed model.

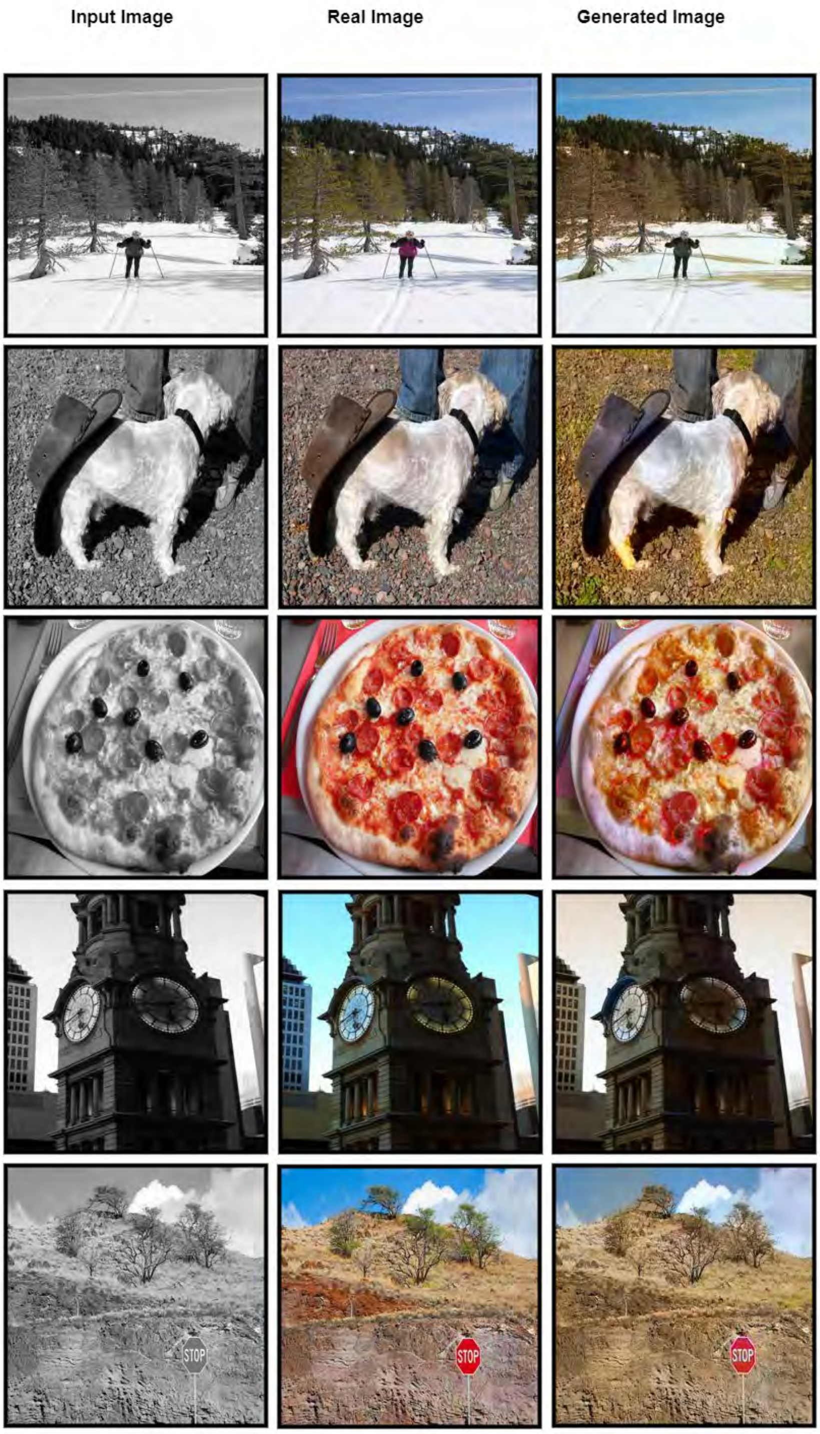


Figure 5.6: Images generated by the proposed algorithm form the MS COCO stuff [7] Dataset: the first column shows the grayscale images, the second column shows the ground truth images and third column shows the colorized outputs of the proposed model.

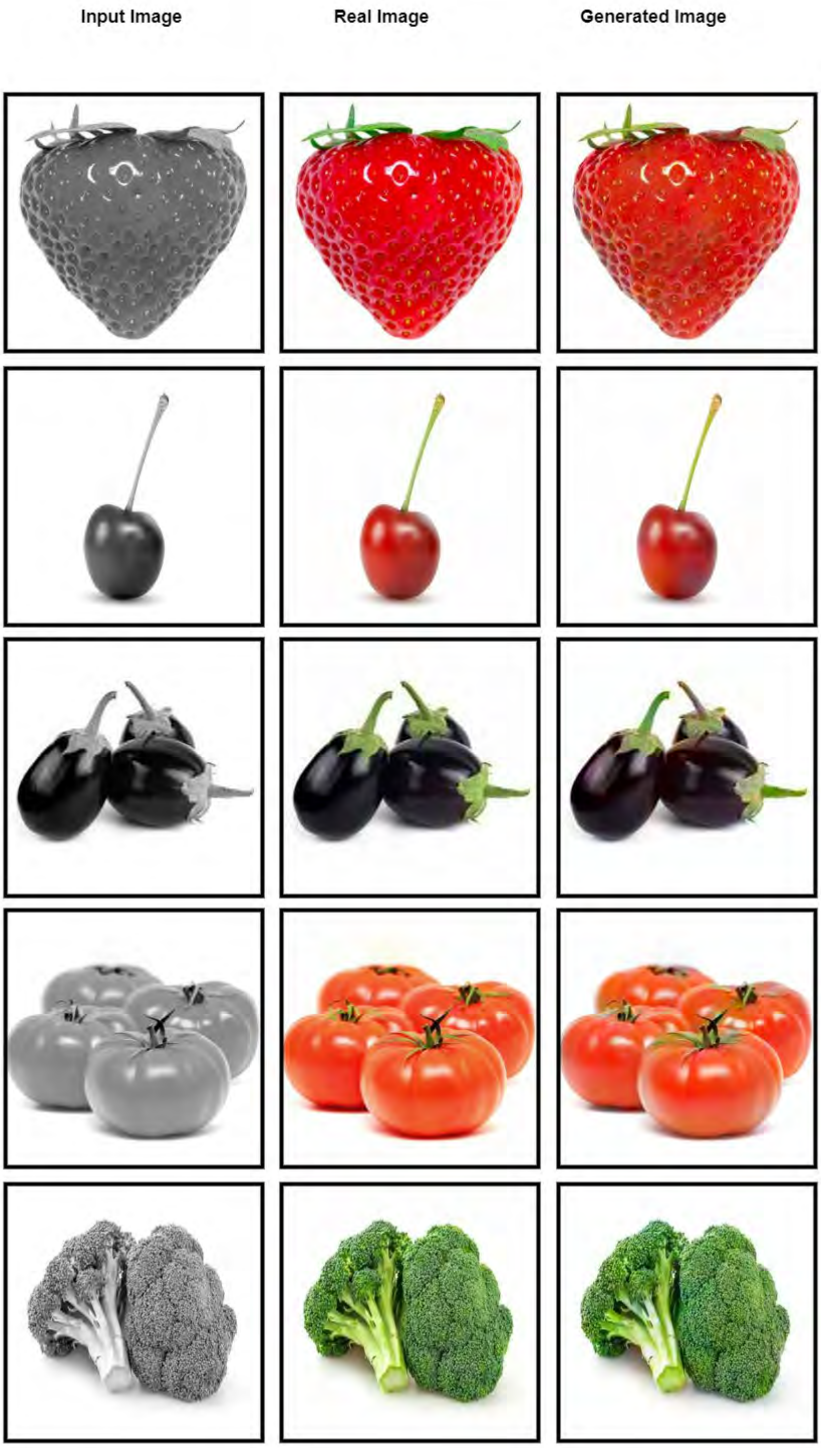


Figure 5.7: Images generated by the proposed algorithm from the NCD [3] Dataset: the first column shows the grayscale images, the second column shows the ground truth images and third column shows the colorized outputs of the proposed model.

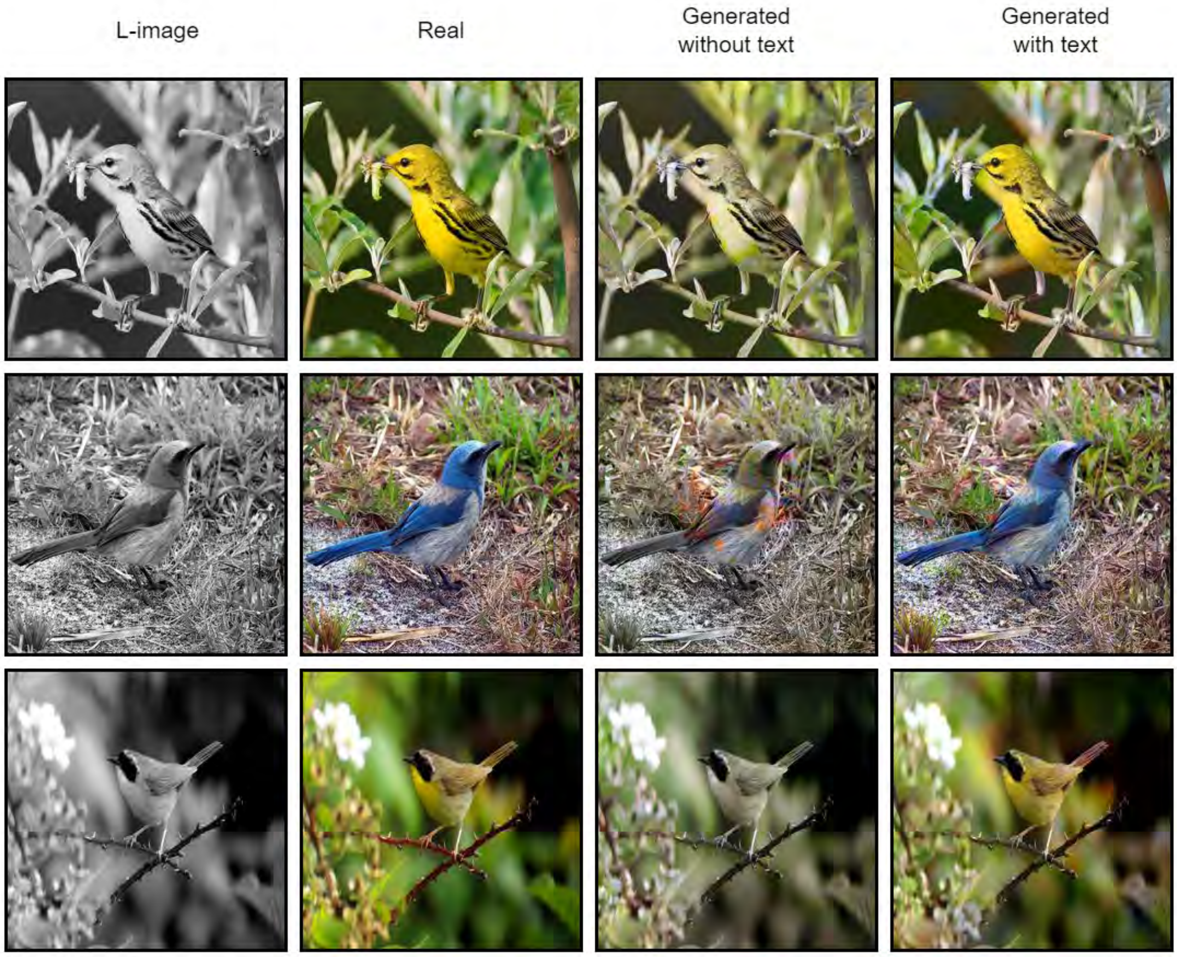


Figure 5.8: Validation of the importance of the textual encoding: first column contains the grayscale images, second column contains the ground truth images, the third and fourth columns show the results generated without and with the textual encoding, respectively.

5.3.2 Experimental Results

To understand the overall performance of the proposed framework, we performed an extensive set of experiments to evaluate the quality of the final colorized images. In Fig. 5.4, we compare our algorithm with [99], [101] and [4]. As shown in the figure, the proposed algorithm colorized the grayscale images with higher fidelity. Although the existing methods have colorized the grayscale images successfully, however the colors are often significantly different to the actual ground truth. The colorized images produced by the SOTA algorithms are also less colorful. As the proposed method utilizes the textual description as auxiliary information, our algorithm generates more realistic and colorful images from the respective grayscale input images. We have generated the

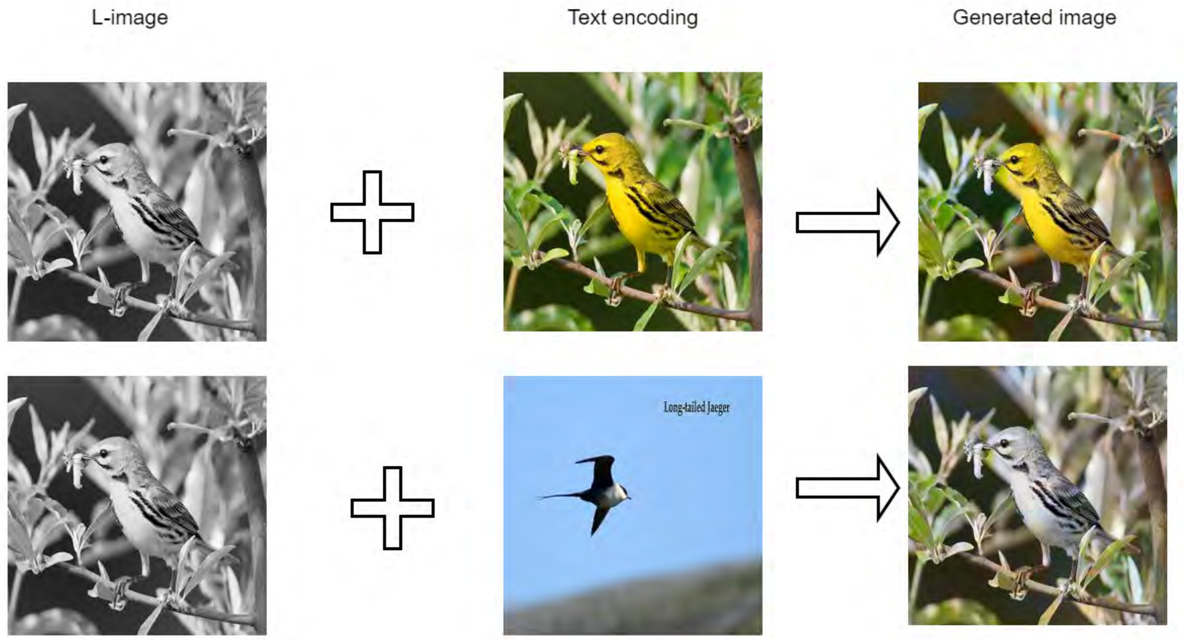


Figure 5.9: Recolorization: The first column shows the grayscale images, the second column shows the images whose textual descriptions are used as conditioning. The third column shows the final colorized images.

color images from three different public datasets. Fig. 5.5 shows image samples from the UCSD Bird dataset [87]. Fig. 5.6 and Fig. 5.7 show samples from the MS COCO [7] dataset and the Natural Color Dataset [3], respectively. To validate the importance of the textual description further, we train a new model without using the textual information. As shown in Fig. 5.8, without the textual conditioning, the proposed pipeline fails to colorize the grayscale images properly. In Fig. 5.9, we also demonstrate that the textual description can be used for the recolorization task. In Fig. 5.9, we have colorized the grayscale image with the actual textual description of the ground truth. In Fig. 5.9, we have kept the grayscale image unchanged and have used the textual description of a different image. It is observed that the proposed framework is able to follow the textual conditioning, and can produce significantly different colorized outputs from the the same grayscale image based on the textual encodings.

To further validate the effectiveness of the proposed model, we evaluate the quality of the generated images using quantitative metrics as well. We use average PSNR, SSIM [85], LPIPS(vgg) [67, 100] and LPIPS(sqz) [28] measures to compare the similarity of the generated images with the ground truth. As shown in Table 5.1, the proposedal.gorithm

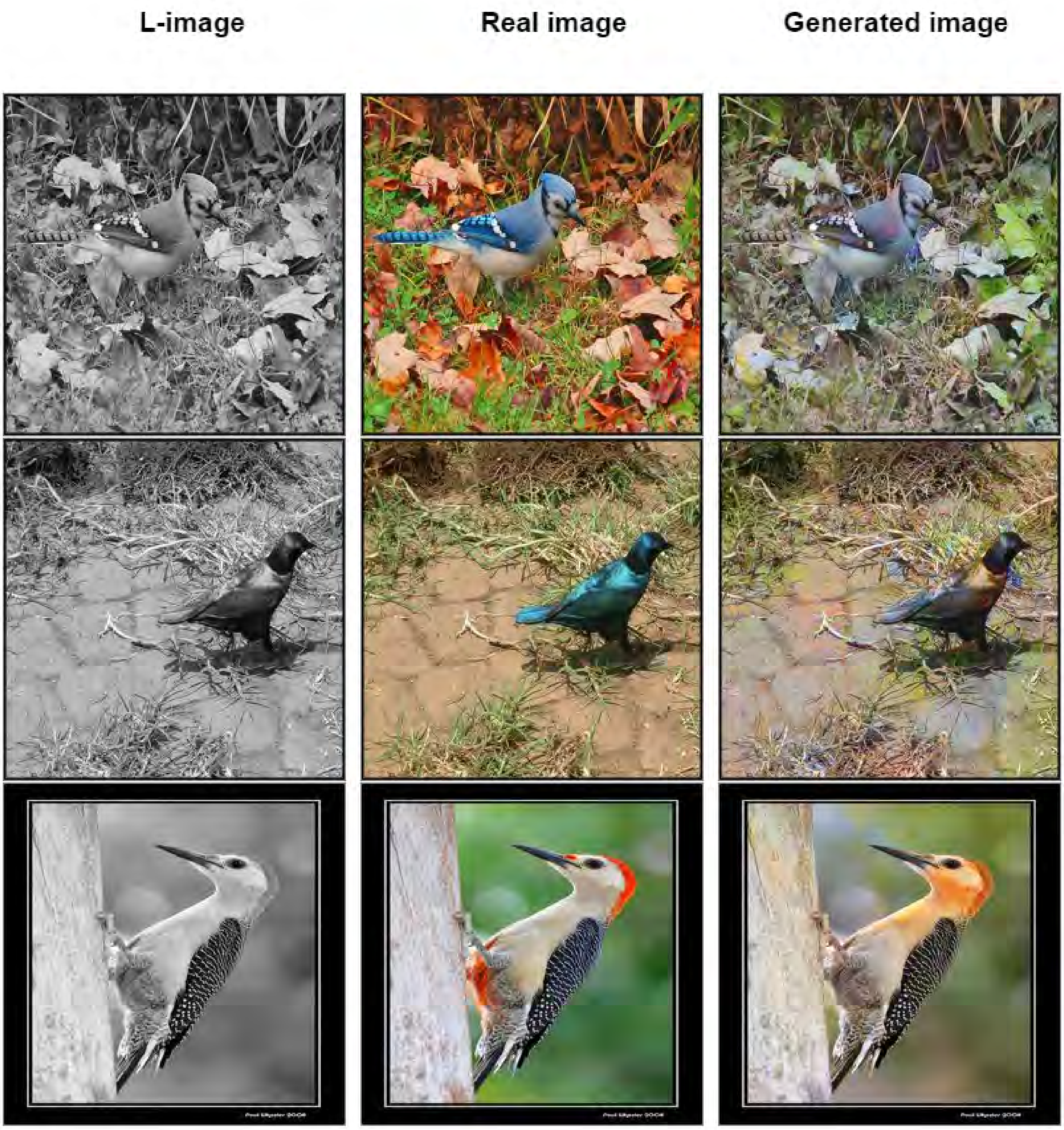


Figure 5.10: Some of the failure cases.

Table 5.1: Quantitative comparison among different colorization methods – Zhang et al. [99], Zhang et al. [101], Bhang et al. [4] and our method.

Method	SSIM ↑	PSNR ↑	LPIPS (vgg) ↓	LPIPS (sqz)↓
Zhang et al. [99]	0.903	22.94	0.253	0.143
Zhang et al. [101]	0.892	22.15	0.231	0.129
Bhang et al. [4]	0.912	22.99	0.228	**0.127**
Our	**0.917**	**23.27**	**0.223**	0.133

outperforms the SOTA algorithms in SSIM, PSNR, LPIPS(vgg) measures.

5.4 Conclusion

In this report, we proposed a novel image colorization algorithm that utilizes textual encoding as auxiliary conditioning in the color generation process. We found that the proposed framework exhibits higher color fidelity compared to the state-of-the-art algorithms. We have also demonstrated that the proposed framework can also be used for recolorization purposes by modulating the textual conditioning. Though our framework has produced superior results, we have considered only textual conditioning of foreground objects in this work. In the given setting, though the proposed algorithm outperforms the SOTA methods, as the textual descriptions mostly depict the foreground objects ignoring the backgrounds; our method exhibits less fidelity for the background colors. However, this discrepancy in the background color is difficult to detect without the ground truth images, which are not available in most of real-world applications. Hence, this problem can be resolved by adding additional color descriptions for the background. We also observed that as the textual descriptions define the colors of the objects coarsely, to fill the gaps, the proposed method generates certain colors that are not there in the respective ground truths. Thus, in certain cases, our method produced a less colorful background which establishes the necessity of a more exhaustive textual description for the grayscale images in the future.

In the next chapter, we discuss integrating textual descriptions as an auxiliary condition with grayscale images to enhance the colorization process. We propose a deep network that takes both the grayscale image and its encoded text description as inputs, predicting the relevant color components for each object. A fusion model then combines the colorized image segments, using the textual information to improve color accuracy and overall quality.

CHAPTER 6

MULTI-MODAL COLORIZATION OF IMAGES USING TEXTUAL DESCRIPTIONS

6.1 Introduction

Colorization techniques differ in many aspects, such as network architecture, loss functions, learning strategies, etc. However, the existing colorization processes [26, 51, 76, 91, 94] mostly follow unconditional generation, where the colors are predicted only from the input grayscale image. In this work, we propose a colorization network that colorizes the image as shown in Fig.6.1, using multi-modal feature attention. In the proposed method, we subdivide the task into two parts to colorize the entire image. At first, we find the mask of each object in the image to associate each object in a scene with an instance. In our multi-modal colorization approach, we consider a textual description of these instances along with the grayscale image in the colorization process. The detected object is colorized by the instance object colorization (IOC) module conditioned over the grayscale image and the corresponding language description. To achieve a superior output, we design the IOC module as a multi-task network that predicts the class of the object instances, and its colorization, to ensure that the IOC module closely learns the association between an object and its color. As we are using text information as an auxiliary condition in the process, the ambiguity in the object-color association is greatly reduced. After the instance-level colorization, we combine all the objects into an image considering their previous spatial support. Finally, we train a network that

takes the partially colorized image generated from the previous stage as input, and the corresponding language description of the entire image (including background and non-object instances like 'sky', 'field' etc.), to generate the final colorized image.
Our contributions are as follows.
- The proposed IOC module utilises instance label image colorization, exploiting object-color associations. To achieve superior performance, we design the IOC module as a multi-task network. To the best of our knowledge, this is the first attempt to design a multi-task network for the colorization task, considering the object-level instances.
- A multi-modal pipeline is proposed that colorizes the image using language information, which is considered as auxiliary conditioning in the colorization process.
- To ensure high fidelity over colors, a novel loss function is proposed that captures the overall Color-Consistency of a scene.

The rest of the work is organised as follows. The overview of the pipeline and the proposed methodology Sec. 6.2, respectively. Sec. 6.3, discusses different experimental results, including the dataset, qualitative results, comparative results with existing methods, different ablation studies, etc. Finally, we conclude the work in Sec. 6.4 discussing the major observations, limitations, and future scope of the proposed algorithm.

6.2 Proposed Framework

The proposed method takes a grayscale image and an image color description as inputs to the network. The textual description contains both object-level descriptions (e.g. 'red ball', 'pink flower', etc.) and the background or non-object descriptions (e.g. 'blue sky', 'green field' etc.). The network predicts two missing color channels in the CIE Lab color space. The proposed method has two sub-stages. The instance object colorization (IOC) module colorizes object-level instances by solving colorization and classification tasks simultaneously, considering relevant object-level textual descriptions only. In the next stage, the entire textual color information is passed through the second network along with the partially colorized image to obtain the fully colorized image.

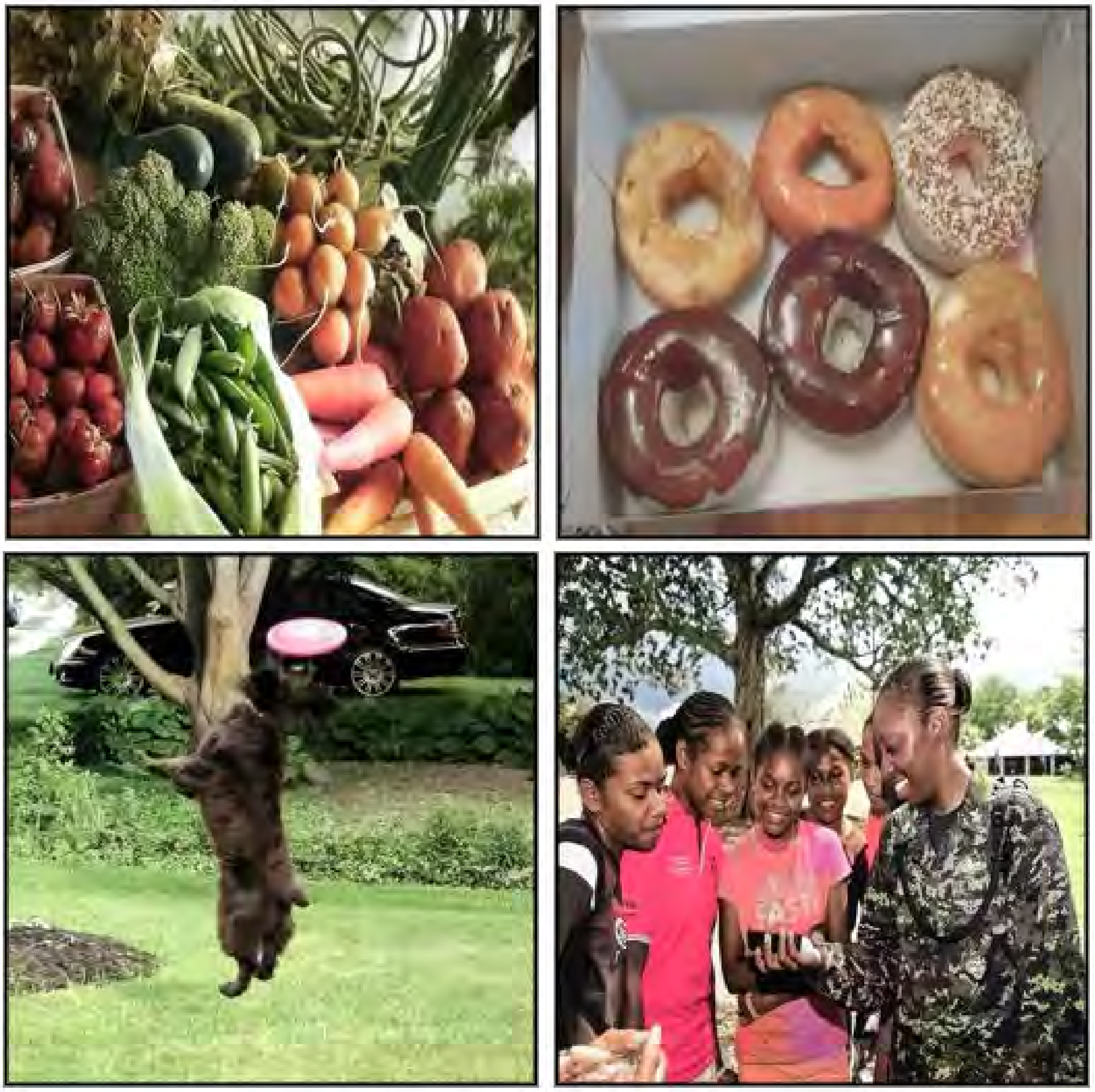


Figure 6.1: Examples of some colorized outputs from our multi-modal colorization approach [Best visible in 300% zoom].

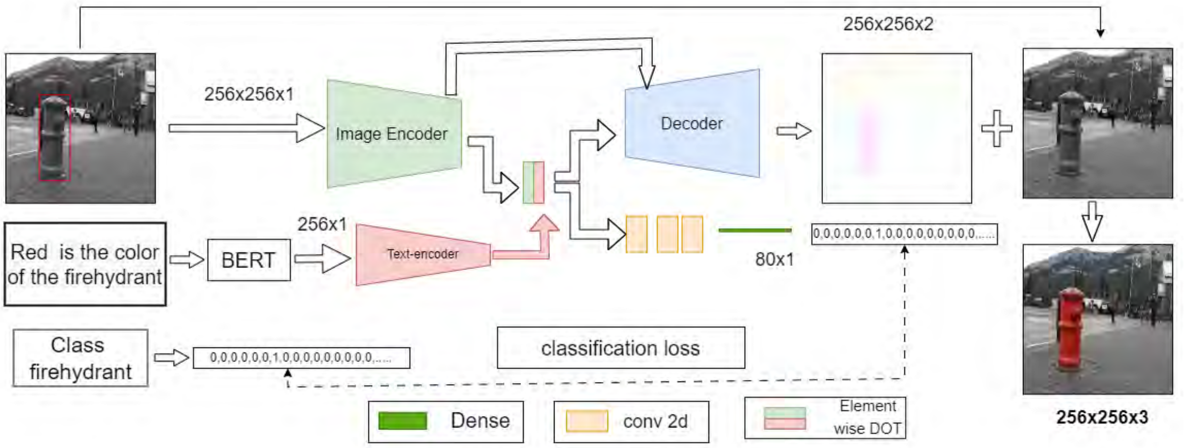


Figure 6.2: Block diagram of the instance object colorization (IOC) module.

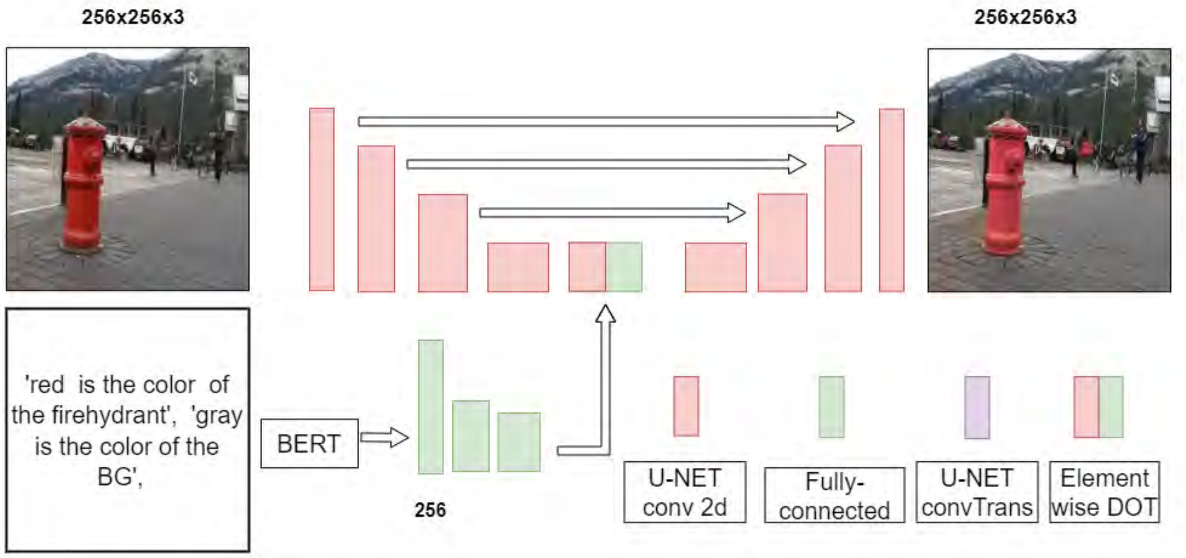


Figure 6.3: A block diagram of the proposed algorithm. The IOC module colorizes all the object instances, whereas the fusion module takes the partially colored image and generates the fully colorized image [Best visible in 300% zoom].

6.2.1 Mask-based Object Detection

For object detection, our method uses the Masked R-CNN [22] network for pixel-accurate object instance masking. After detecting the object's bounding box, we resized the object to 256 × 256 resolution. The actual coordinate or the *support* of the object is stored for the fusion network. The detected object is resized and split into two parts - one is the grayscale image that contains the L channel, and the second one contains the color information of the ‘ab’ channel.

6.2.2 Color Information Encoding

One of the key contributions of our work is to encode the color information of the object instances to increase fidelity in the colorization process. For the color information encoding, we use the BERT [17] model to convert the textual description of an object to a 256×1 dimensional vector $\mathbf{v}_i$. This instance-level encoding is used by our proposed IOC module.

In a complex scene, there might be entities that can be detected as an object by the Masked R-CNN because of their non-object nature (e.g. ‘sky’), or because the classes are simply not included in the Masked R-CNN training (e.g. ‘tiger’). However, it is often probable to get color descriptions of these entities fairly easily (e.g., ‘blue sky’, ‘tiger with yellow and black stripes’). Thus, we encode all the available auxiliary text information associated with the scene using BERT to a 256×1 vector $\mathbf{v}_o$ and pass it to the final fusion module.

6.2.3 Instance Object Colorization (IOC) module

The IOC module uses a modified UNet like structure as the backbone with two inputs. One of the inputs is a $256 \times 256 \times 1$ dimensional grayscale object image detected by the Masked R-CNN, which is passed through the image encoder to generate a feature representation of size $8 \times 8 \times 64$ at the end of the encoder module. The other input to the IOC module is the color encoding $\mathbf{v}_i$ received from the frozen BERT model. The feature representation $\mathbf{v}_i$ is passed through three fully-connected trainable layers of size 256, 1024, and 4096, respectively. The final output of the fully-connected layer is reshaped to size $8 \times 8 \times 64$ and multiplied in an element-wise manner with the output of the image encoder to provide the textual conditioning in the generation pipeline. The text-conditioned feature vector is used for the multi-tasking approach. In one of the output branches of the IOC, we try to reconstruct the image’s color information (‘ab’ channel) using the image decoder. In the other branch of the IOC, we use convolutional layers to classify the object instances among the 80 classes available for the Masked R-CNN module. A schematic of the IOC module is illustrated in Fig.6.2. The loss functions used to train the model are discussed in Sec. 6.2.6. The proposed encoder contains four 2D convolutional layers with ReLU activation. Each of the layers uses a stride of 2 and a batch normalization operation at the end of the layer. In the decoder part, in each layer, we use a 2D up-convolution followed by Batch Normalization and the ReLU activation function. Like the UNet model, we also use the skip connection from the encoder part to the respective decoder with the same resolution. In the label classifier, there are three 2D convolutional layers followed by one dense layer to find the class label prediction.

6.2.4 Fusion Module for Colorization

As shown in Fig.6.3, A fusion module is used to re-colorize the merged image, which is partly colorized by the Instance object colorization module (IOC). During the merging process, we combine the colorized object mask derived from the IOC module. The merging technique incorporates the height probabilities of the object's color features into the resulting merged image. By considering these probabilities, we can effectively address the challenge of overlapping objects. To achieve this, we selectively discard the low-probability object features, allowing us to prioritize and emphasize the objects' more reliable and significant color attributes. This selective approach ensures that the merged image primarily consists of the most probable object color features, resulting in a more accurate representation of the scene. By discarding the low-probability object features, we mitigate the issues caused by overlapping. This means that objects with uncertain or less probable color characteristics have a reduced impact on the final merged image. As a result, the merged image provides clearer and more distinct representations of the objects, enhancing visual quality and aiding in object recognition tasks. This merging technique, incorporating height probabilities of object color features, offers a robust solution for handling overlap problems. It allows us to create visually appealing and reliable merged images that accurately reflect the color attributes of the objects present in the scene. The fusion module uses a modified UNet-like structure as a backbone of two inputs. One input is a merged image, and the second is the text encoding of the input image. The size of the merged image is $256 \times 256 \times 3$, and the text encoding size is 256. In the image encoder, four convolutions followed by BN and ReLU activation are used to generate a feature representation with size $8 \times 8 \times 64$ at the end of the image encoder with down-sampling by 2. In the text encoding pathway, three fully-connected layers with sizes of 256, 1024, 4096 are used to generate the high-dimensional feature representation. After that, we compute the element-wise dot product with the text-based feature vector and image-based feature vector, which gives multi-modal attention to the networks. The decoder is designed employing a UNet-like decoder architecture. Four up-convolution layers followed by BN and ReLU activation are used to generate the final output with up-samplings by 2.

6.2.5 Discriminator

For superior generation quality, both the IOC module and the Fusion module use two separate discriminators to check the quality of the generated outputs. In both modules, a PatchGan-based discriminator is used to judge the generated image. The input size of the discriminator is 256 × 256 × 3. Three convolutional layers, with 64, 128, and 256 filters and ReLU activation, are used to extract the features of the input image. The grayscale input image (L^i) is stacked with either a target image (T^i) or with the estimated image (E^i) where T^i and E^i are the AB channels of the color image. During the training of the discriminator, the (L^i,T^i) stack is labeled as real, and the (L^i,E^i) stack is labeled as fake.

6.2.6 Loss

Both the IOC and the fusion modules are trained independently, considering the same sets of losses. We consider L_i as the input image to module m that has auxiliary text conditioning S_i. We have an estimated image E_i whose ground truth is T_i. For the IOC module, E_i and T_i are the estimated image and the ground truth image of each object instance, respectively, whereas, for the fusion module, E_i and T_i indicate the final estimated image and the actual ground truth of the scene. To train the generator, the proposed method uses three different types of losses- $\mathscr{L}_1$, Perceptual, and Color-Consistency loss.

For overall pixel-level fidelity, we consider the $\mathscr{L}_1$ loss for the generation process. The $\mathscr{L}_1$ loss for the generator is calculated as:

$$\mathscr{L}_1^G = \|E_i - T_i\|_1 = \|G_m(L_i, S_i) - T_i\|_1 \tag{6.1}$$

where G_m indicates the generator of module m that maps L_i and S_i to E_i.
To maintain the high perceptual quality of the generated image, we introduce a perceptual loss term while training the generator G_m. We have calculated the perceptual loss $\mathscr{L}_{per}^{G_m}$ as

$$\mathscr{L}_{per}^G = \|VGG_{L_9}(T_i) - VGG_{L_9}(E_i)\|_1 \tag{6.2}$$

where $VGG_{L_9}(P)$ means that the feature vector is taken from the ninth layer of a pre-trained VGG19 model for a given input P.
Color-Consistency Loss: We observe that the combination of an $\mathscr{L}_1$ loss and the perceptual loss cannot always produce satisfactory results, and there is often a lack of colors in the colorized images. We identified that $\mathscr{L}_1$ loss and perceptual loss often ignore the color consistency of the relatively smaller foreground objects. Minimization of $\mathscr{L}_1$

alone often generates less saturated images. To overcome the problem, we propose a Kullback‚ÄìLeibler (KL)-divergence-based loss function to measure the Color consistency of generated colorized images. We first convert E^i and T^i to their respective RGB color images. Next, we calculated the normalized histogram of each of the color channels for the respective images and defined the Color-Consistency loss (CC Loss) as

$$\mathscr{L}^{G_m}_{Color-Consistency} = \sum_{i=1}^{3} \|D_{kl}(hist_{T_{Ci}}, hist_{E_{Ci}})\| \tag{6.3}$$

where $hist_{X_{Ci}}$ means the normalized histogram of the i^{th} channel of the image (X) and D_{kl} is the KL-divergence.
We define the GAN loss $\mathscr{L}^{G_m}$ of the generator G_m as

$$\mathscr{L}^{G_m}_{GAN} = \mathscr{L}_{BCE}(D_m(L_i, G(L_i, S_i)), 1) \tag{6.4}$$

whereas the GAN loss for the discriminator D_m is defined as

$$\begin{aligned}\mathscr{L}^{D_m}_{GAN} =& \mathscr{L}_{BCE}(D_m(L_i, T_i), 1)\\ &+ \mathscr{L}_{BCE}(D_m(L_i, G_m(L_i, S_i)), 0)\end{aligned} \tag{6.5}$$

where $\mathscr{L}_{BCE}$ is the BCE loss function calculated as

$$\mathscr{L}_{BCE} = (y\log(p) + (1-y)\log(1-p)) \tag{6.6}$$

where y is the true label of a sample and p is the predicted label of the sample.
In the IOC module, we add categorical cross-entropy as class-based object classification loss into the generator.

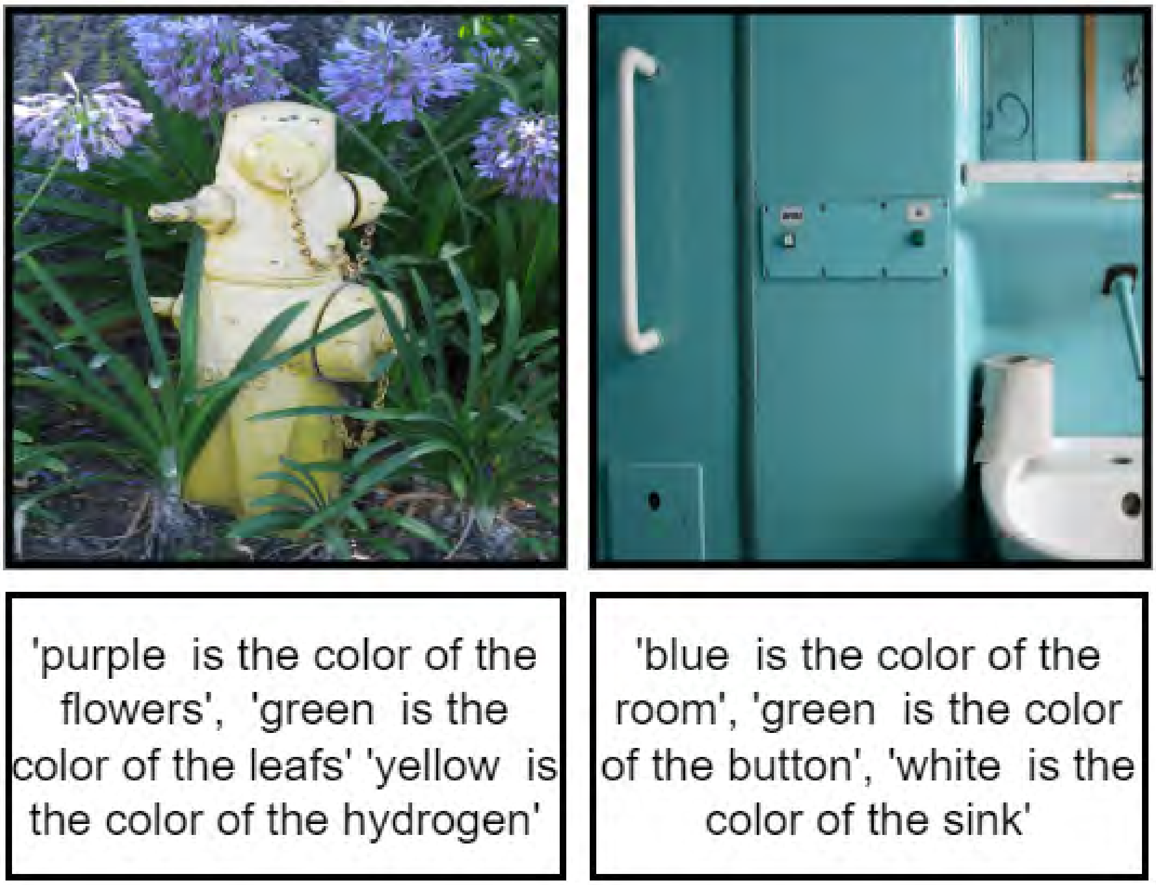
$$\mathscr{L}_{CE} = \sum_{i=1}^{N} y_i.log(\hat{y}_i) \tag{6.7}$$

where N is the total number of classes, y_i and $\hat{y}_i$ are the actual class label of one hot ground truth and predicted label value, respectively. The final loss $\mathscr{L}^{G_{ioc}}$ for the IOC module is calculated as

$$\mathscr{L}^{G_{ioc}} = \lambda_1 L_1^G + \lambda_2 L_{per}^G + \lambda_3 L_{Color-Consistency}^G + \lambda_4 L_{GAN}^G + \lambda_5 L_{CE} \tag{6.8}$$

where λ_i is the weighting factor. In our work, we consider $\lambda_1 = 10$, $\lambda_2 = 1$, $\lambda_3 = 1$, $\lambda_4 = 1$ and $\lambda_5 = 1$.

Though discriminator D_m can be trained directly by minimizing the loss function $\mathscr{L}^{D_m}_{GAN}$, we take a linear combination of the losses associated with the generator to



'purple is the color of the flowers', 'green is the color of the leafs' 'yellow is the color of the hydrogen'

'blue is the color of the room', 'green is the color of the button', 'white is the color of the sink'

Figure 6.4: Examples of some samples of the MS COCO dataset.

calculate the fusion module generator loss $\mathscr{L}^{G_m}$ for training. The final loss $\mathscr{L}^{G_m}$ for the fusion module is calculated as

$$\mathscr{L}^{G_m} = \lambda_1 L_1^G + \lambda_2 L_{per}^G + \lambda_3 L_{Color-Consistency}^G + \lambda_4 L_{GAN}^G \tag{6.9}$$

where λ_i is the weighting factor. In our work, we consider $\lambda_1 = 10$, $\lambda_2 = 1$, $\lambda_3 = 1$ and $\lambda_4 = 1$.

6.3 Experiments Details, Results and Analysis

6.3.1 Dataset

We use the MS-COCO QA [61] dataset for our training and evaluation. The MS-COCO QA dataset contains 42,429 images with color information. We use the color information

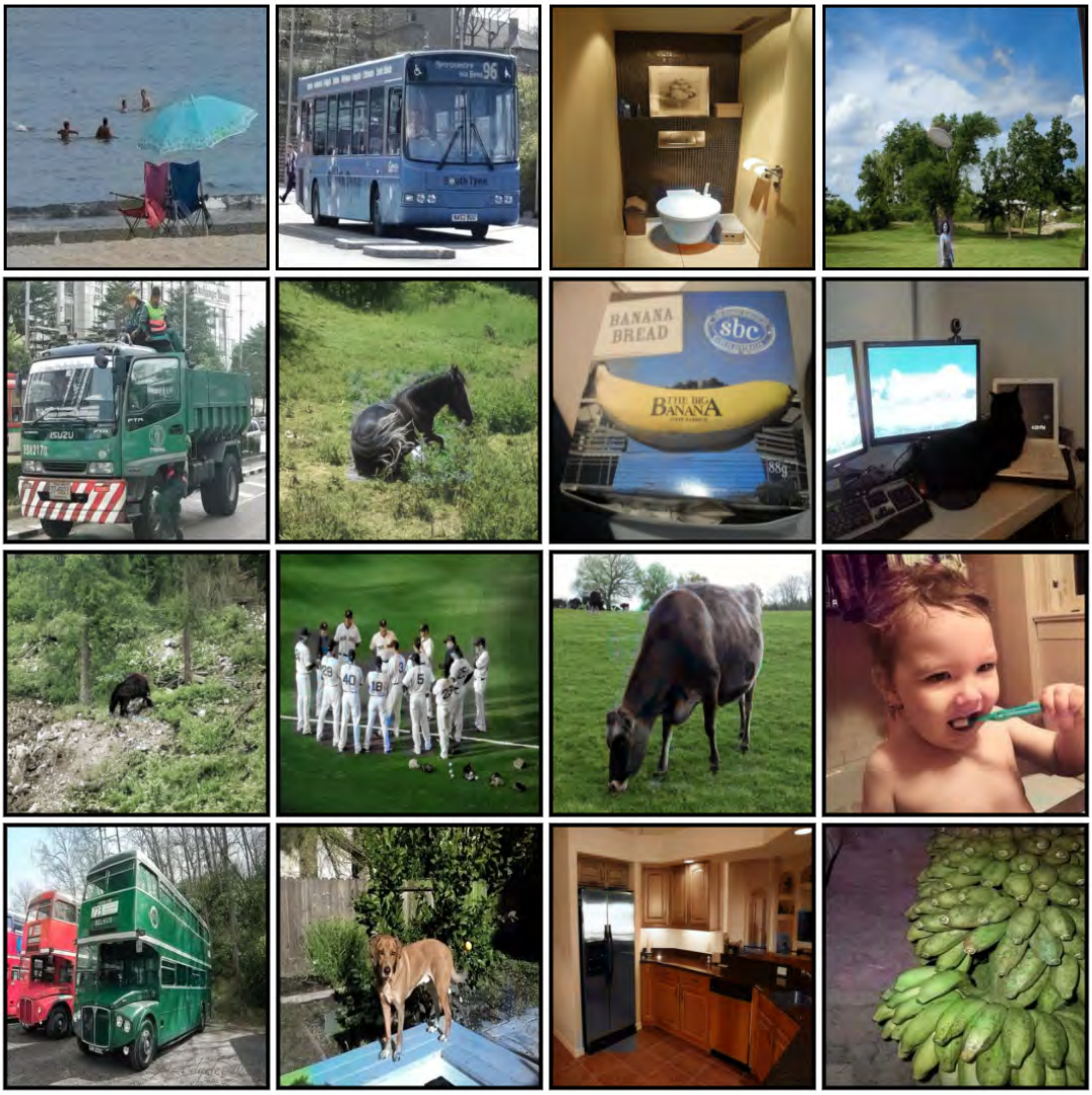


Figure 6.5: Examples of some qualitative results generated by the proposed framework [Best visible in 300% zoom].

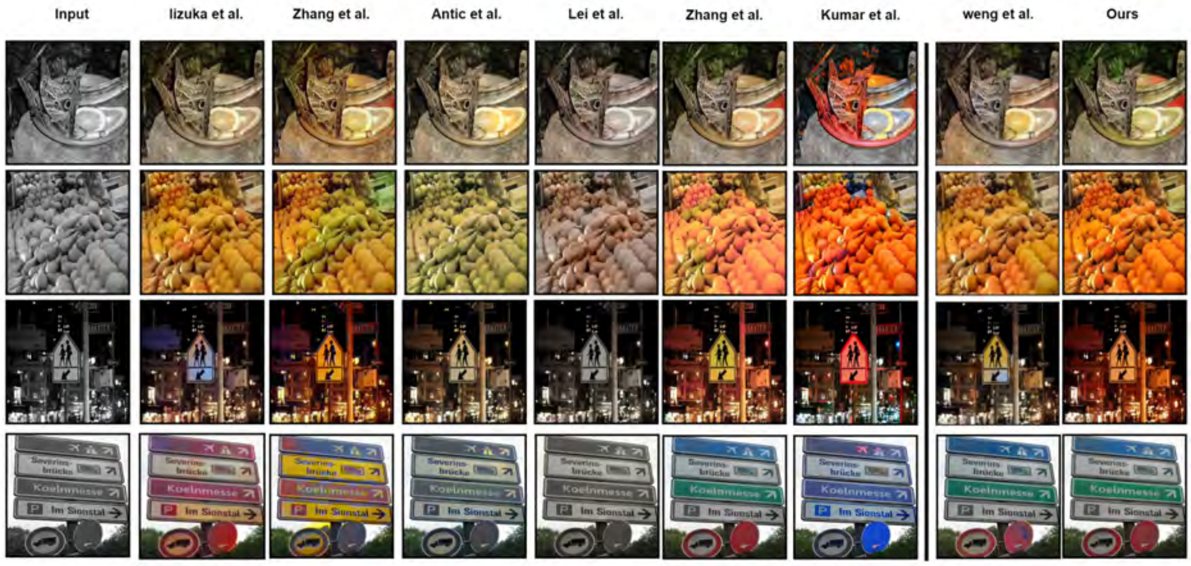


Figure 6.6: Qualitative comparison results: The first column contains input images, columns two to eight contain generated results of the SOTA algorithms, and the last column shows the results generated by the proposed algorithm. SOTA methods are Iizuka et al. [29] Zhang et al. [99] Antic et al. [2] Lei et al. [44] Zhang et al. [101] Kumar et al. [40] and Weng et al. [90] [Best visible in 300% zoom].

as the auxiliary conditioning for the IOC module. A total of 38,136 images are used in training and 4,293 images are used for testing. A few samples of the database are shown in Fig 6.4.

To evaluate the performance of our proposed method, we have generated a large number of images using the proposed framework. Some of the sample colorized results images are shown in Fig 6.5. It is observed that the proposed method is able to colorize complex scene images efficiently. As shown in Fig. 6.5, the model can handle scenes with multiple objects, occlusions, shadows, etc. To further evaluate the quality of the colorized images, we have shown a set of 20 randomly selected images to 41 viewers, where half of the images are natural RGB images and half of the images are colorized using the proposed approach. We asked the viewers to mark whether a displayed image is colorized or not within 5 seconds. In this experiment, we obtained an average accuracy of 53.41%, which is just slightly better than random guessing. This uncontrolled experiment proves that the proposed method colorizes an image, maintaining natural color consistency.

Table 6.1: Comparison results of the proposed algorithm with existing colorization algorithms.

	LPIPS ↓	PSNR ↑	SSIM ↑
Iizuka et al. [29]	0.185	23.860	0.922
Zhang et al. [99]	0.234	21.838	0.895
Antic et al. [2]	0.180	23.692	0.920
Lei et al. [44]	0.191	24.588	0.922
Zhang et al. [101]	0.138	26.823	0.937
Kumar et al. [40]	0.137	26.653	0.937
Weng et al. [90]	0.138	27.329	0.921
Ours	**0.120**	**28.214**	**0.938**

Table 6.2: Comparison results of the proposed algorithm with the existing text-based colorization algorithms. The dataset split of Weng et al. [90] is used to generate the results.

Method	LPIPS ↓	PSNR ↑	SSIM ↑
Manjunatha et al. [53]	0.282	21.055	0.853
Chang et al. [9]	0.159	25.504	0.917
Weng et al. [90]	0.169	24.965	0.917
Ours	**0.128**	**27.230**	**0.933**

6.3.2 Comparison Results

To validate the effectiveness of the proposed method, we evaluate the qualitative and quantitative results as well. In the Table 6.1, we measured LPIPS [67], PSNR and SSIM [85] to compare the proposed framework with the existing image colorization methods. We have considered the algorithms of Iizuka et al. [29], Zhang et al. [99], Antic et al. [2], Lei et al. [44], Zhang et al. [101], Kumar et al. [40] and Weng et al. [90] for comparison. As shown in Table 6.1, the proposed algorithm has outperformed the existing algorithms in terms of LPIPS, PSNR and SSIM scores. Furthermore, we also compared the qualitative results in Fig 6.6.
To further validate, we chose three text-based methods and used the same dataset in [90]. As shown in Table 6.2, the proposed algorithm has performed well in all the scores. We compared the qualitative results in Fig 6.12 and observed that except [90], the colorized images, produced by [53] and [9], do not look natural. The images produced by [90] look over-saturated in color.

It can be seen from the Fig. 6.5 that the proposed algorithm generates perceptually consistent and well-colorized images.

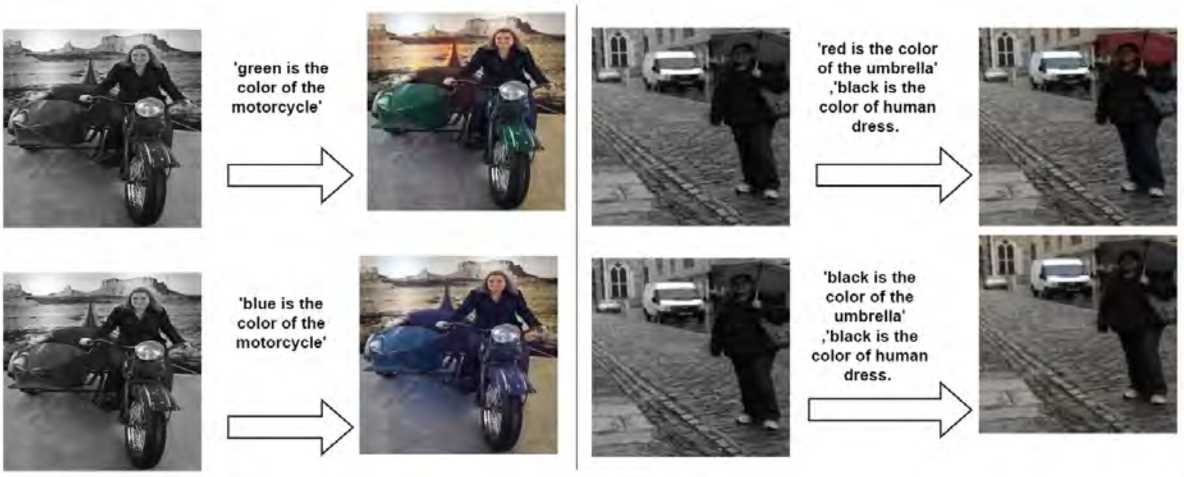

'green is the color of the motorcycle'
'blue is the color of the motorcycle'
'red is the color of the umbrella' ,'black is the color of human dress.
'black is the color of the umbrella' ,'black is the color of human dress.

Figure 6.7: Examples of object re-colorization. [Best visible in 300% zoom]

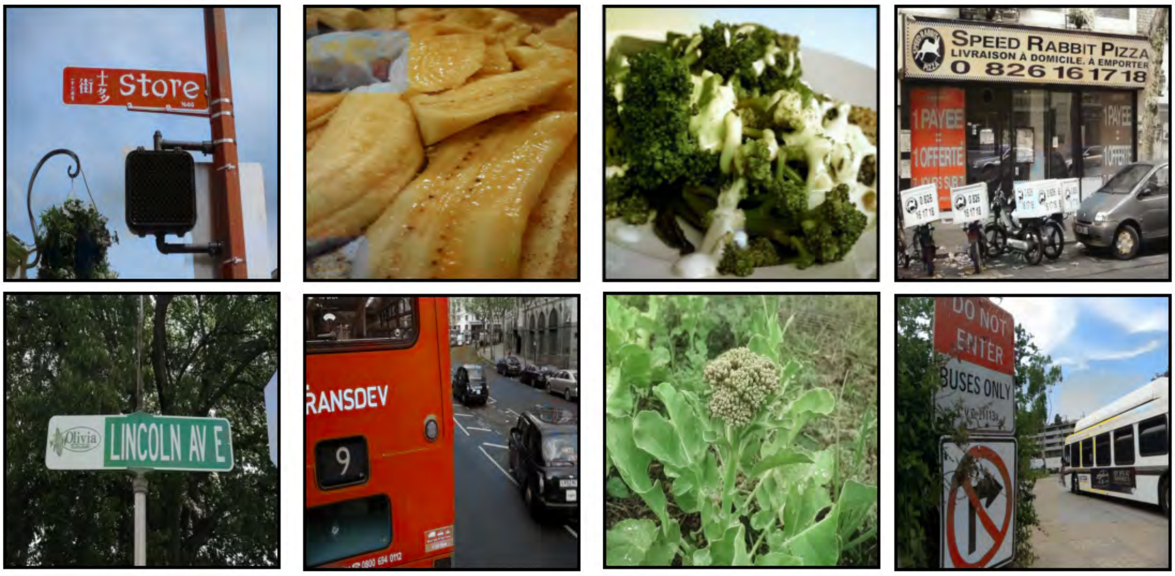


Figure 6.8: Examples of some colorized outputs from our proposed multi- modal colorization approach. [Best visible in 300% zoom].

6.3.3 Qualitative Result

To further evaluate the performance of our proposed method, we have generated a large number of images using the proposed framework. Some of the samples' colorized result images are shown in Fig. 6.8 and Fig. 6.10. We have also compared with the SOTA methods with more examples in Fig. 6.11. To evaluate the performance, we also added qualitative ablation of our method.

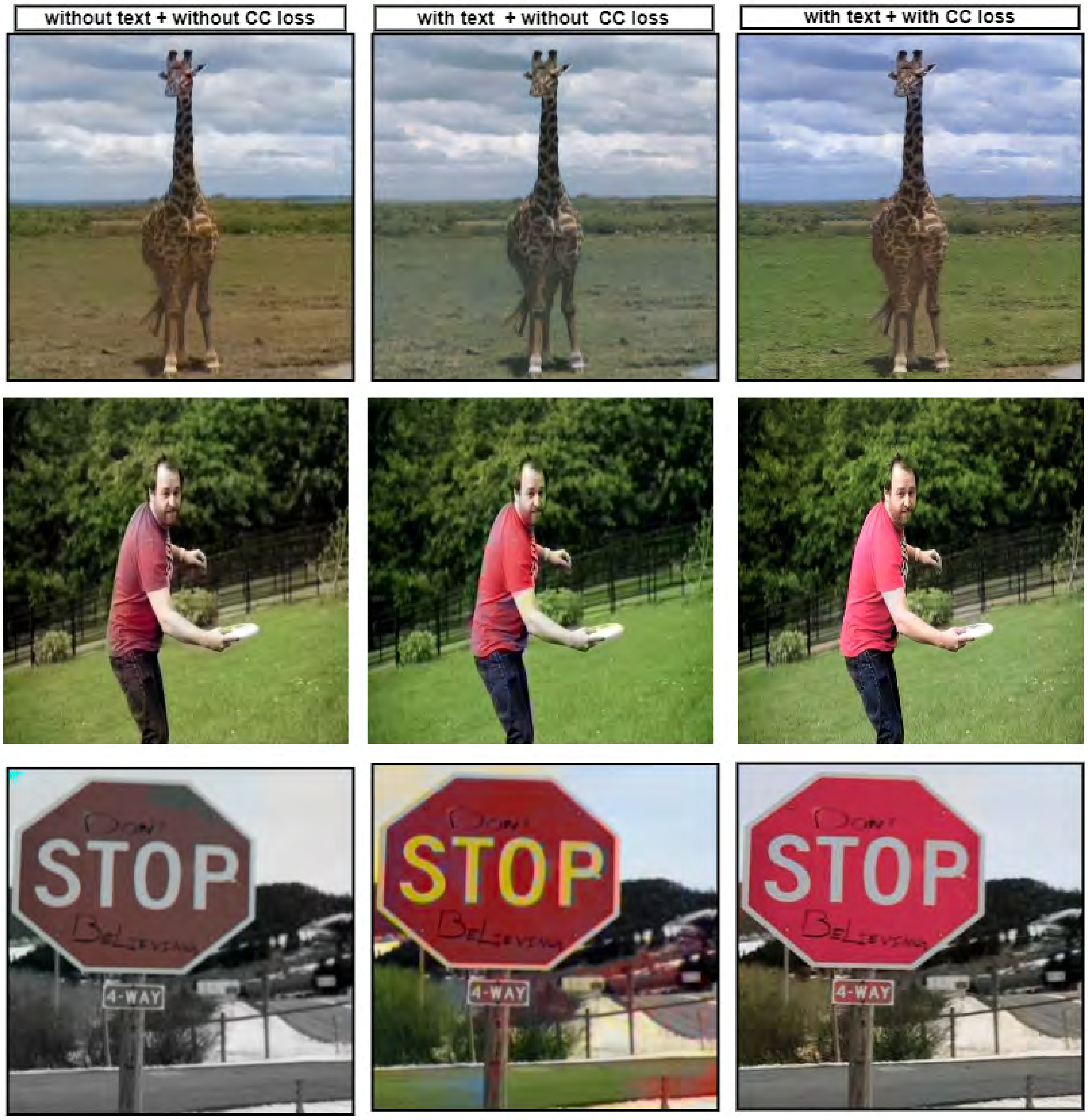


Figure 6.9: Qualitative ablation of our proposed multi- modal colorization approach [Best visible in 300% zoom].

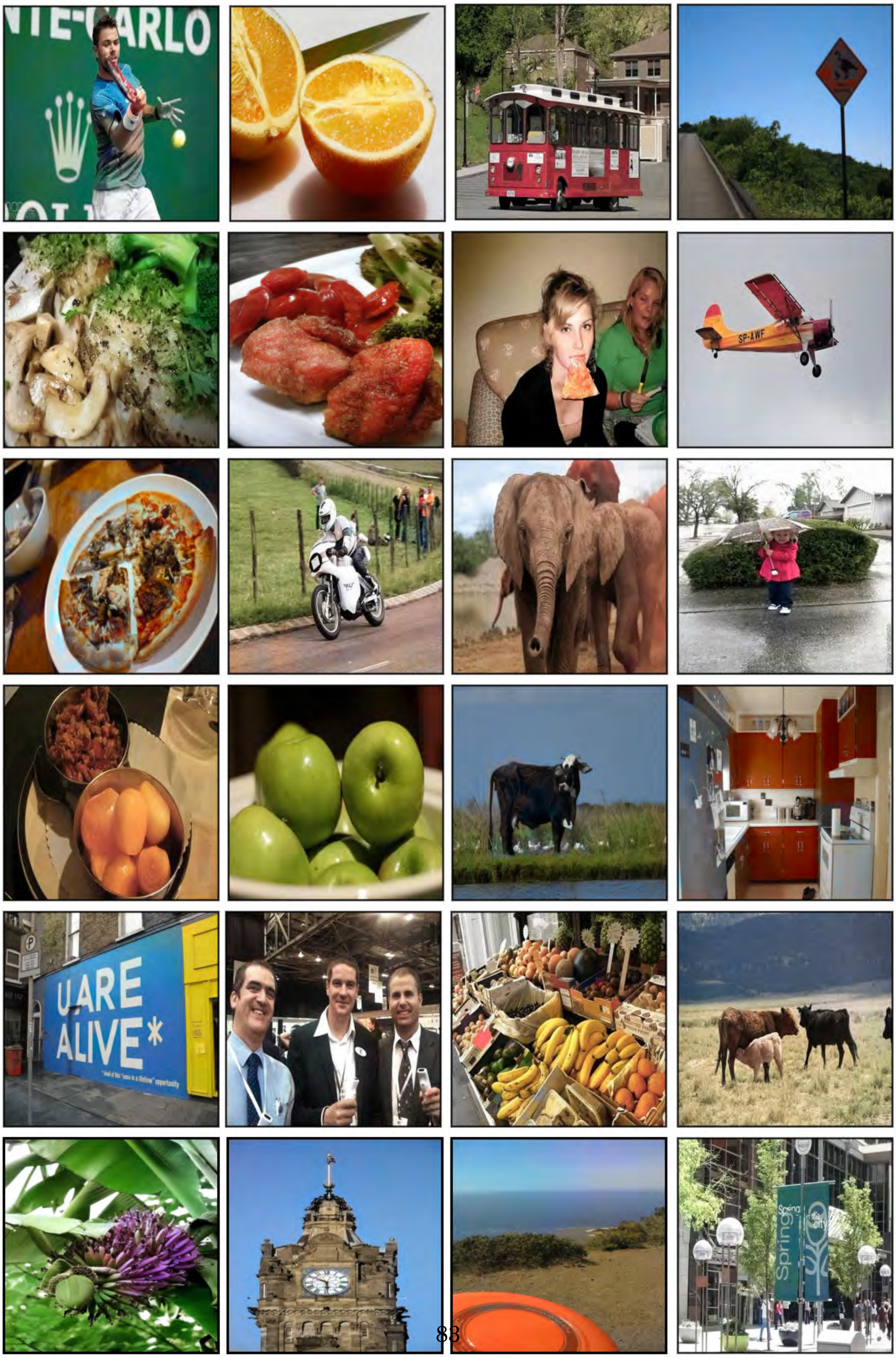


Figure 6.10: Examples of some colorized outputs from our proposed multi- modal colorization approach[Best visible in 300% zoom].

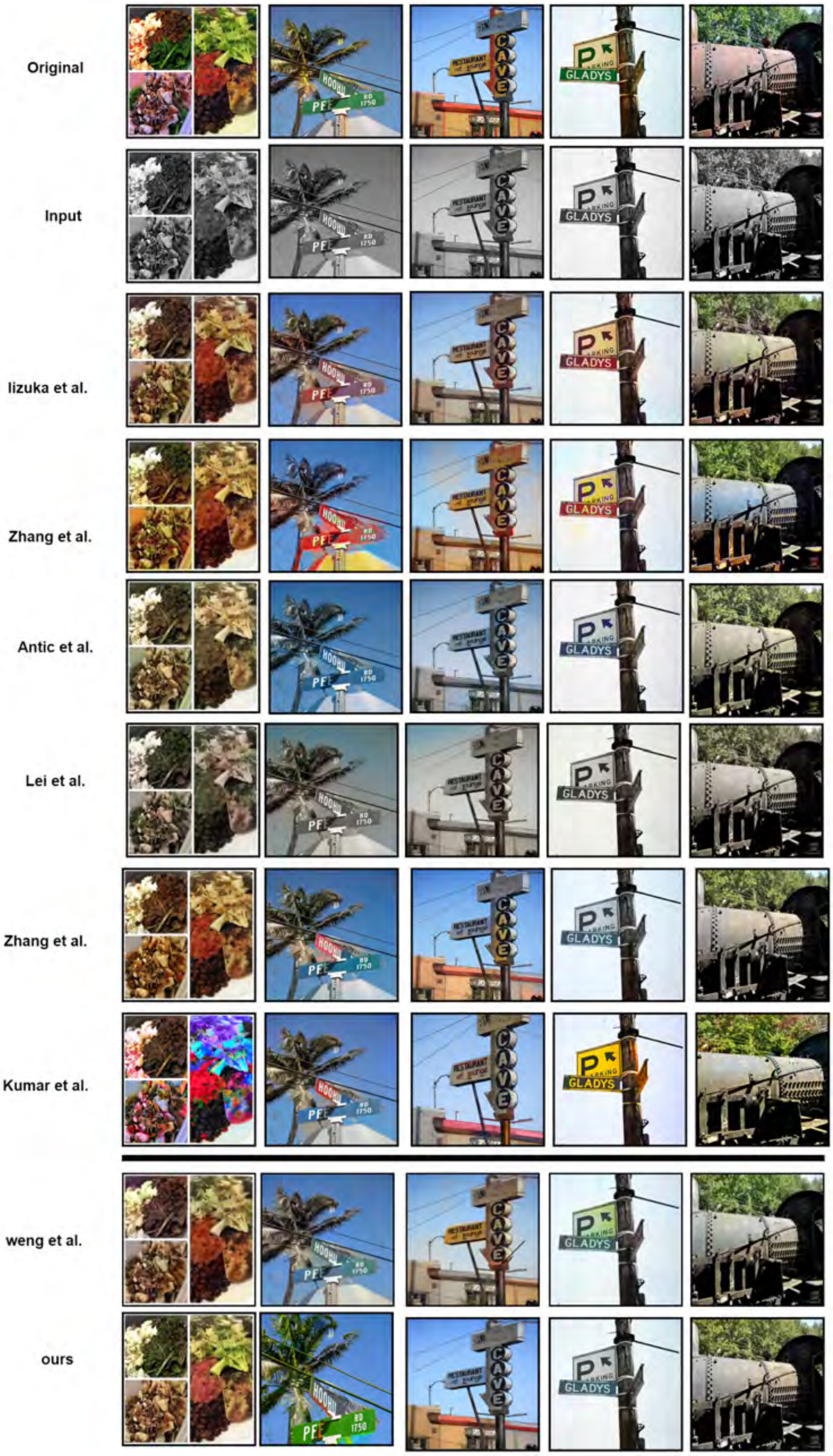


Figure 6.11: Qualitative comparison results: The first row contains original color images, the second-row input images, the third to the ninth row contains generated results of the SOTA algorithms, and the last row shows the results generated by the proposed algorithm. SOTA methods are Iizuka et al. [29], Zhang et al. [99], Antic et al. [2], Lei et al. [44], Zhang et al. [101], Kumar et al. [40], Weng et al. [90] [Best visible in 300% zoom].

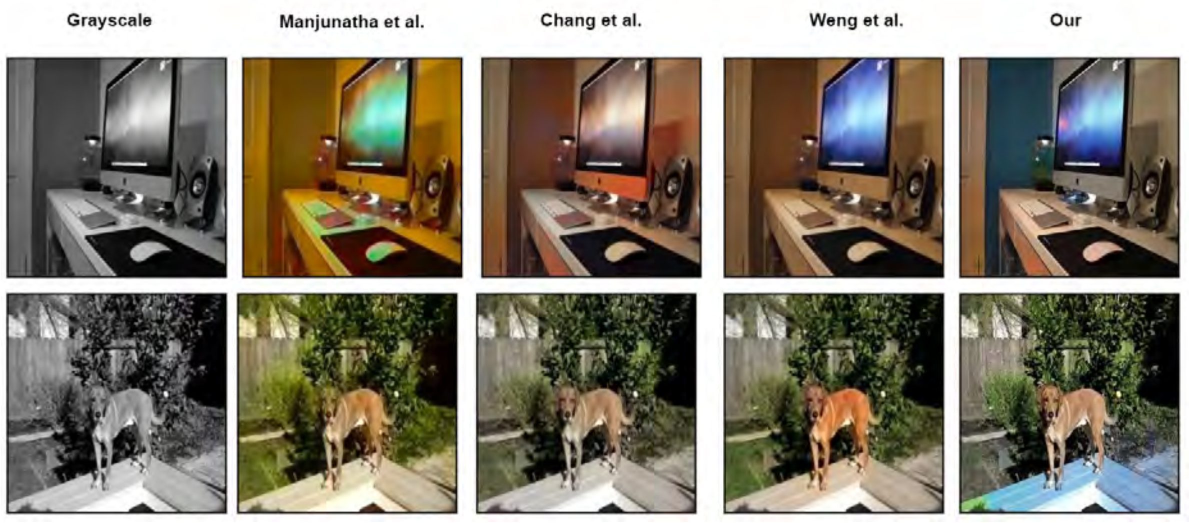


Figure 6.12: Comparison results of the proposed algorithm with the existing text-based colorization algorithms.

6.3.4 Ablation Study

To further validate the contributions of the novel components that are present in our frameworks, we perform a detailed ablation study. We mainly focus on the effectiveness of textual conditioning along with the effect of the proposed Color-Consistency loss term. As shown in Table 6.4, we observe the presence of textual conditioning, which significantly improves all three metrics. Though the proposed Color-Consistency (CC) loss slightly improves the LPIPS score, the improvements in PSNR and SSIM scores are significantly higher when the Color-Consistency loss is incorporated. Our ablation studies show that we achieved the best result when both Color-Consistency loss and textual information were used in the colorization framework. Finally, to validate the usefulness of the proposed algorithm on the overall generation quality, we observe the alignment of the color histograms of the real GT image and the colorized image. As shown in Fig. 6.13, the color histogram of the colorized image closely follows the color histogram of the ground truth. Some examples of ablation results are in figure 6.9.

6.3.5 Re-colorization

Auxiliary conditioning is a valuable technique utilized in the colorization process. By applying this method, we can effectively modify the textual information and achieve the re-colorization of various object instances within a given input image. This approach

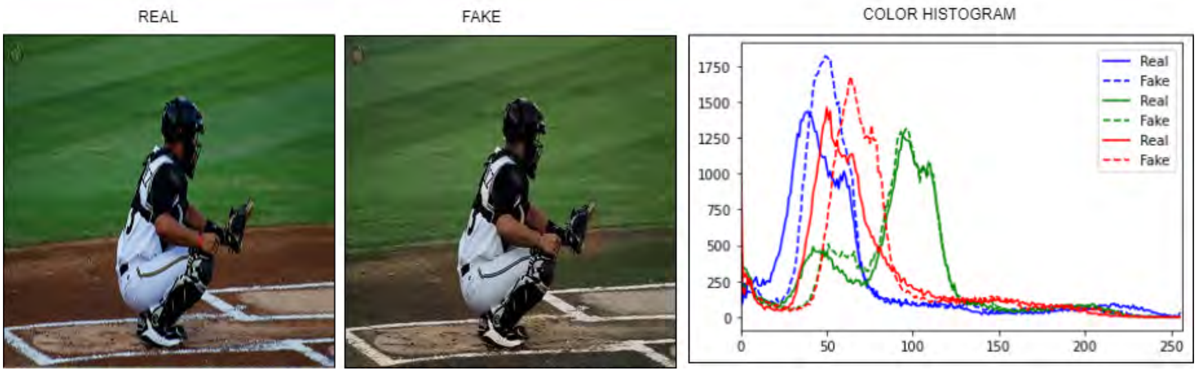


Figure 6.13: Channel wise histogram [Best visible in 300% zoom].

Table 6.3: Ablation results of different components used in our framework. *Text Info* indicates the usage of auxiliary text information and *CC Loss* indicates the proposed Color-Consistency loss.

Text Info	CC Loss	LPIPS ↓	PSNR ↑	SSIM ↑
X	X	0.156	24.230	0.896
X	✓	0.155	24.625	0.912
✓	X	0.130	27.238	0.921
✓	✓	0.120	28.214	0.938

Table 6.4: Ablation results of different components used in our framework. *Configuration* indicates the usage of depth of U-Net, and activation indicates the proposed Architecture.

Configuration	LPIPS ↓	PSNR ↑	SSIM ↑
with sigmoid	0.163	18.326	0.863
Encoder depth 3	0.142	22.214	0.893
Encoder depth 4	0.120	28.214	0.938
Encoder depth 5	0.123	27.988	0.931

allows for a versatile and dynamic colorization process, enhancing the overall flexibility and control of the system. Through auxiliary conditioning, we introduce additional factors or variables that influence the colorization outcome. These factors are typically derived from the textual information provided, such as object labels or descriptions. By modulating these factors, we can selectively alter the colorization of specific object instances while keeping the rest of the image intact. This capability to re-colorize different object instances of the same input image opens up numerous possibilities. It enables us to experiment with various color schemes and aesthetics without the need for extensive manual intervention. By simply adjusting the auxiliary conditioning, we can generate different versions of the colorized image that cater to different preferences or

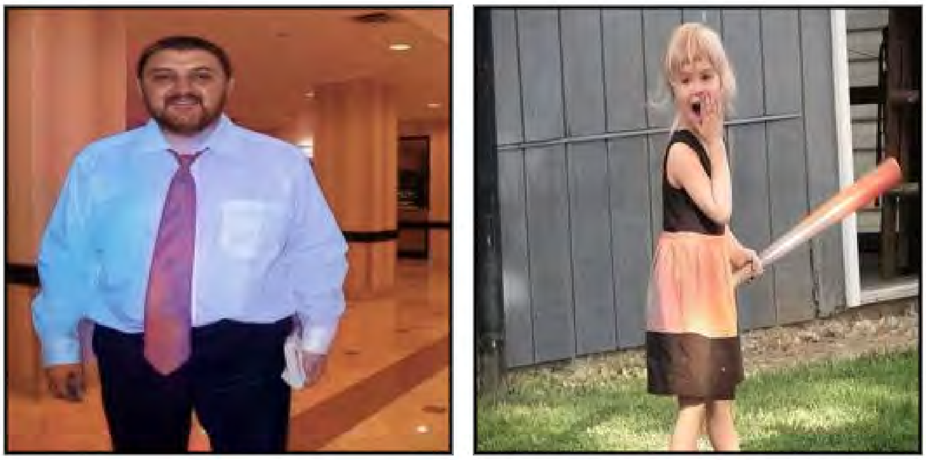


Figure 6.14: Examples of some failure cases of the proposed method.

artistic requirements. Some re-colorization results are shown in Fig. 6.7. As shown in the figure, the proposed methods can seamlessly handle the re-colorization task depending on the textual conditioning.

6.3.6 Failure Cases

In Fig. 6.5, the effectiveness of the proposed algorithm in handling complex scenes with multiple objects is evident. It demonstrates a remarkable performance overall. However, it is important to acknowledge that there are instances where the generated results may fall short of expectations. Upon closer examination, it has been observed that such shortcomings occur more frequently when the available auxiliary textual descriptions of colors are insufficient. The quality of the colorization process heavily relies on the textual information provided as auxiliary conditioning. When there is a lack of detailed or comprehensive descriptions of colors in the input, the algorithm faces challenges in accurately determining the appropriate colorization for different objects within the scene. Consequently, the generated results may not meet the desired level of satisfaction. To address this issue, it becomes crucial to ensure that the input data includes sufficient and precise auxiliary textual descriptions. By providing richer and more comprehensive color-related information, we can enhance the algorithm's ability to produce satisfactory colorizations, even in complex scenarios. Some failure cases are shown in Fig. 6.14. This problem, however, can be mitigated by introducing more color descriptions for the object instances.

6.4 Conclusion

In this work, a novel image colorization approach is proposed that utilises the color information as an auxiliary conditioning of the network. The results of the proposed algorithm are compared with existing image colorization methods, and it can be seen that the proposed method outperforms the existing ones in terms of LPIPS and PSNR metrics. We validated that the object instance-level colorization produces superior results when auxiliary textual conditioning is available. A novel loss function has also been introduced to have more fidelity in the color generation process. In certain cases, it is observed that the proposed method produces less colorful images when the color information is not adequately present in the conditioning texts. This problem can be solved in future work by adding additional text descriptions in the database.

In the next chapter, we present our colorization method, which uses a deep network that takes a grayscale image and its coarse textual description as inputs. By employing a multi-modal feature attention mechanism with the Instance Object Colorization (IOC) module, we enhance colorization accuracy by conditioning on both image and text.

CHAPTER 7

Exploring Auxiliary Conditioning for Image Colorization

7.1 Introduction

One of the main challenges of automatic colorization is the one-to-many association problem, where a grayscale image can be associated with multiple equally plausible colorizations. For example, a black-and-white photo of a sunset can be colorized with warm oranges and yellows but also with cool blues and purples, depending on the artist's

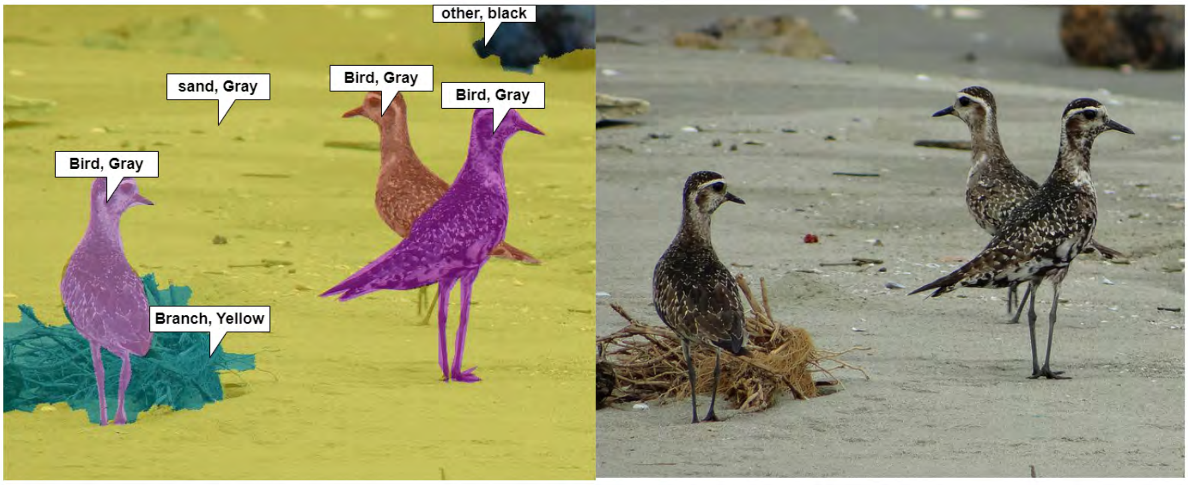


Figure 7.1: Example of an image of the proposed dataset [Best visible in 300% zoom].

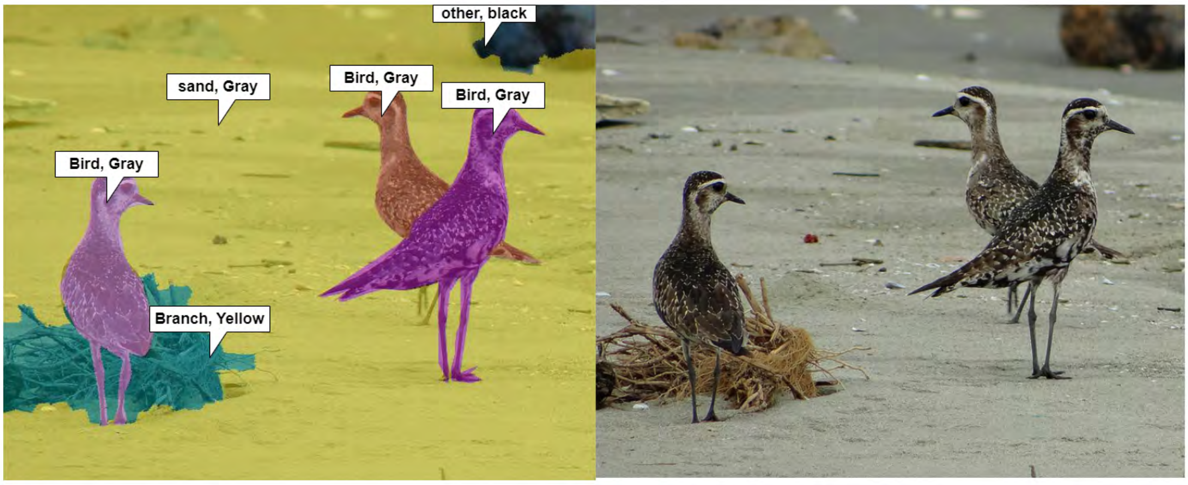


Figure 7.2: Example of an image of the proposed dataset [Best visible in 300% zoom].

interpretation. Researchers have developed various strategies for training colorization models to overcome these challenges. For example, some have proposed using Generative Adversarial Networks (GANs), which can learn to generate realistic colorization by competing against a discriminator that tries to distinguish between real and fake images. Others have used self-supervised learning, where the colorization model is trained to predict the original color image from a degraded or low-resolution version of the same image. Most of the automatic colorization methods lack color consistency, semantic understanding of the color, naturalness, and realism with appropriate object color [9, 12, 34, 93]. To address this problem, we have proposed using additional sources of information, such as semantic segmentation maps or object recognition algorithms, to guide the colorization process and produce more consistent and accurate results.

This work proposes a novel colorization network that uses multi-modal feature attention to colorize images more accurately. Our approach divides the colorization task into two parts. First, we detect the mask of each object in the image to associate it with an instance. Next, we consider a coarse textual description of the instances and the grayscale image in the colorization process. This multimodal approach helps reduce ambiguity in an object-color association.

In this work, our contributions are as follows.

- A multi-modal pipeline is proposed that colorizes the image using language information, which is considered as auxiliary conditioning in the colorization process.
- We also proposed a novel dataset based on the color information of every object of the COCO dataset. We generate color-coded textual information by associating class labels

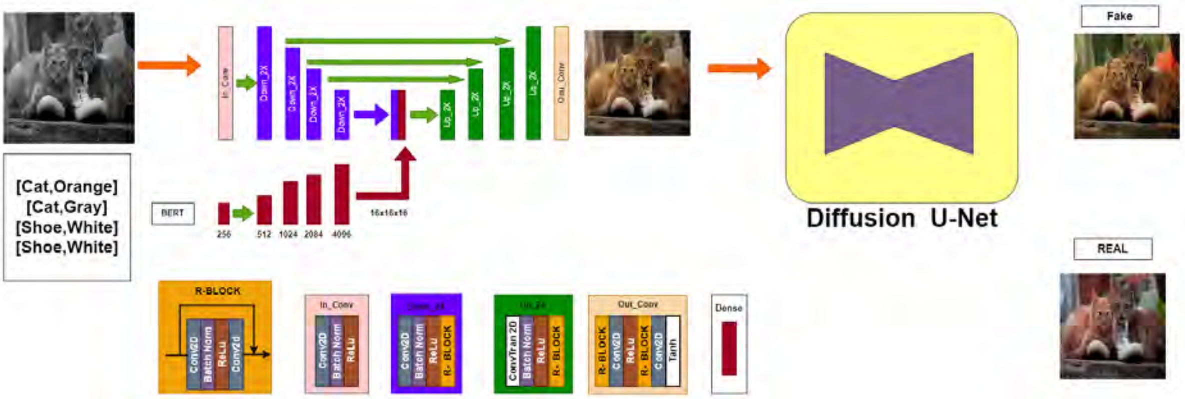


Figure 7.3: Block diagram of the Instance Object Colorization (IOC) module [Best visible in 300% zoom].

with their respective objects in the images.
The remainder of this chapter is structured as follows. In Section 7.2, we delve into the pipeline overview and the proposed methodology. Section 7.3 presents various experimental findings, encompassing qualitative results, comparisons with existing methods, and various ablation studies. Finally, in Section 7.4, we conclude the work by discussing key observations, limitations, and potential future directions for the proposed algorithm.

Despite the availability of several algorithms for image colorization, none of them currently exploit a multi-modal approach at the instance level for colorization. The lack of a proper color dataset has been identified as one of the main challenges for this algorithm. To address this issue, we have developed a new dataset including color descriptions for foreground and background objects. The new dataset is designed to provide detailed and accurate color information for a wide range of objects, from simple shapes to complex structures. This information can be used to improve the accuracy and performance of the algorithm, making it more effective in various applications. The color descriptions in the dataset are based on a rigorous and standardized methodology that ensures consistency and reliability. Each color is described using a combination of color names, numerical values, and other relevant information, such as hue, saturation, and brightness. In addition to providing color descriptions for individual objects, the dataset also includes information about color combinations and contrasts. This allows the algorithm to consider the overall color scheme of an image and make more informed decisions about how to process it. Overall, the new color dataset represents a significant step forward in developing this algorithm. Providing accurate and detailed color information will help overcome one of the algorithm's main challenges and enable it to perform more effectively in a wide range of applications.

7.2 Proposed Framework

The proposed method operates by taking a grayscale image and an accompanying image color description as inputs to the network. The color description encompasses both object-level descriptions, such as "red ball" or "pink flower," as well as background or non-object descriptions, such as "blue sky" or "green field." The objective of the network is to predict the missing two color channels in the CIELAB color space, completing the colorization process. To achieve this, our proposed method consists of two distinct sub-stages. The first stage is the Instance Object Colorization (IOC) module Figure 7.3, which focuses on colorizing object-level instances. In this module, we concurrently tackle the colorization and classification tasks, leveraging relevant object-level textual descriptions exclusively. By considering the textual descriptions associated with each object instance, we aim to enhance the accuracy and specificity of the colorization process. Moving on to the second stage, the entire textual color information is passed through the second network. In this stage, we introduce the partially colorized image along with the textual information. The objective is to obtain a fully colorized image as the network's final output. By incorporating both the textual color information and the partially colorized image, we leverage the complementary nature of these inputs to refine and enhance the colorization results. The overall methodology follows a two-stage process, where the IOC module handles object-level colorization. At the same time, the subsequent network integrates the remaining textual color information with the partially colorized image to produce the final fully colorized output. By dividing the process into two sub-stages and carefully considering object-level descriptions and background information, we aim to achieve accurate and realistic colorization results.

7.2.1 Dataset

We employed the MS-COCO dataset [61] for training and evaluating our models. This dataset comprises 123,000 images. During the training phase, we utilized 118,000 images, while the remaining 5,000 were set aside for testing. We generated valuable color information from each image. We leveraged this color information as auxiliary conditioning for our IOC module.

To generate the color information, we followed a two-step procedure encompassing

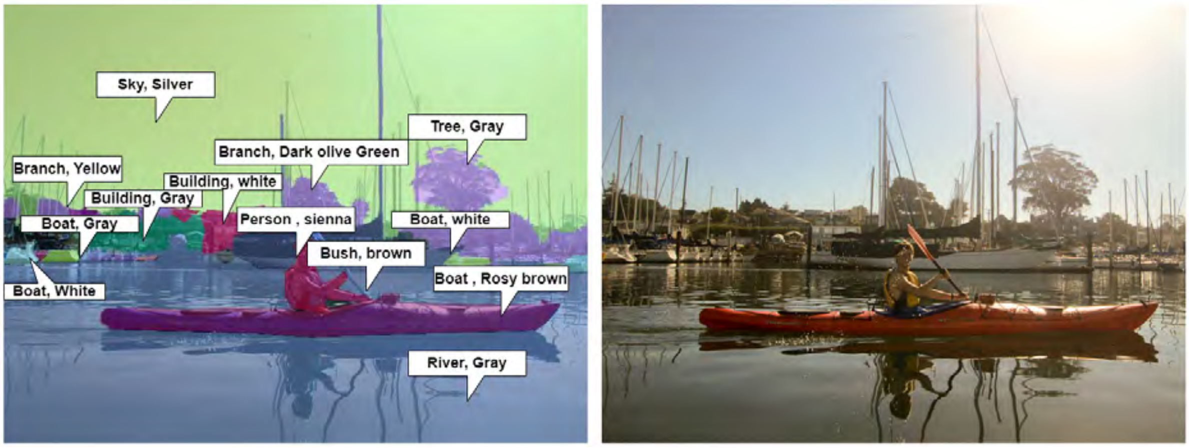


Figure 7.4: Examples of the dataset [Best visible in 300% zoom].

segmentation and color identification. Initially, we utilized the ground truth of the COCO stuff object segmentation boundaries and class labels to extract object and non-object regions. Subsequently, we proceeded to determine the most appropriate or dominant color of each object segment by calculating its distance from a predefined set of 60 colors.

For metric distance, we used the ΔE metric for similar color identification as mentioned in Eq. 7.1 .

$$\Delta E_i = \sqrt{\Delta L_i^2 + \Delta a_i^2 + \Delta b_i^2 + R_T \cdot \Delta C_i \cdot \Delta H_i} \tag{7.1}$$

ΔE_i is the Delta E (i=2000) color difference. $\Delta L_i^2, \Delta a_i^2, \Delta b_i^2$ are differences in lightness, redness-greenness, and yellowness-blueness, respectively. ΔC_i is the difference in chroma. R_T is a rotation function to adjust for differences in hue and chroma. ΔH_i is the difference in hue.

This resulting color information and its corresponding class label were stored as the ground truth for further use. To ensure the accuracy of the color information, we employed human labour to validate it. Trained individuals reviewed each color identification and determined its correctness. If the color information was deemed accurate, it was preserved. However, if any discrepancy or error was found, we selected the appropriate color from the predefined set of 60 colors to rectify it. The dataset encompasses a total of 183 class labels, providing a diverse range of object categories. For a visual representation, you can refer to Figure 7.4, which showcases a selection of samples from our dataset. This illustration offers a glimpse into the rich and varied content present within the dataset.

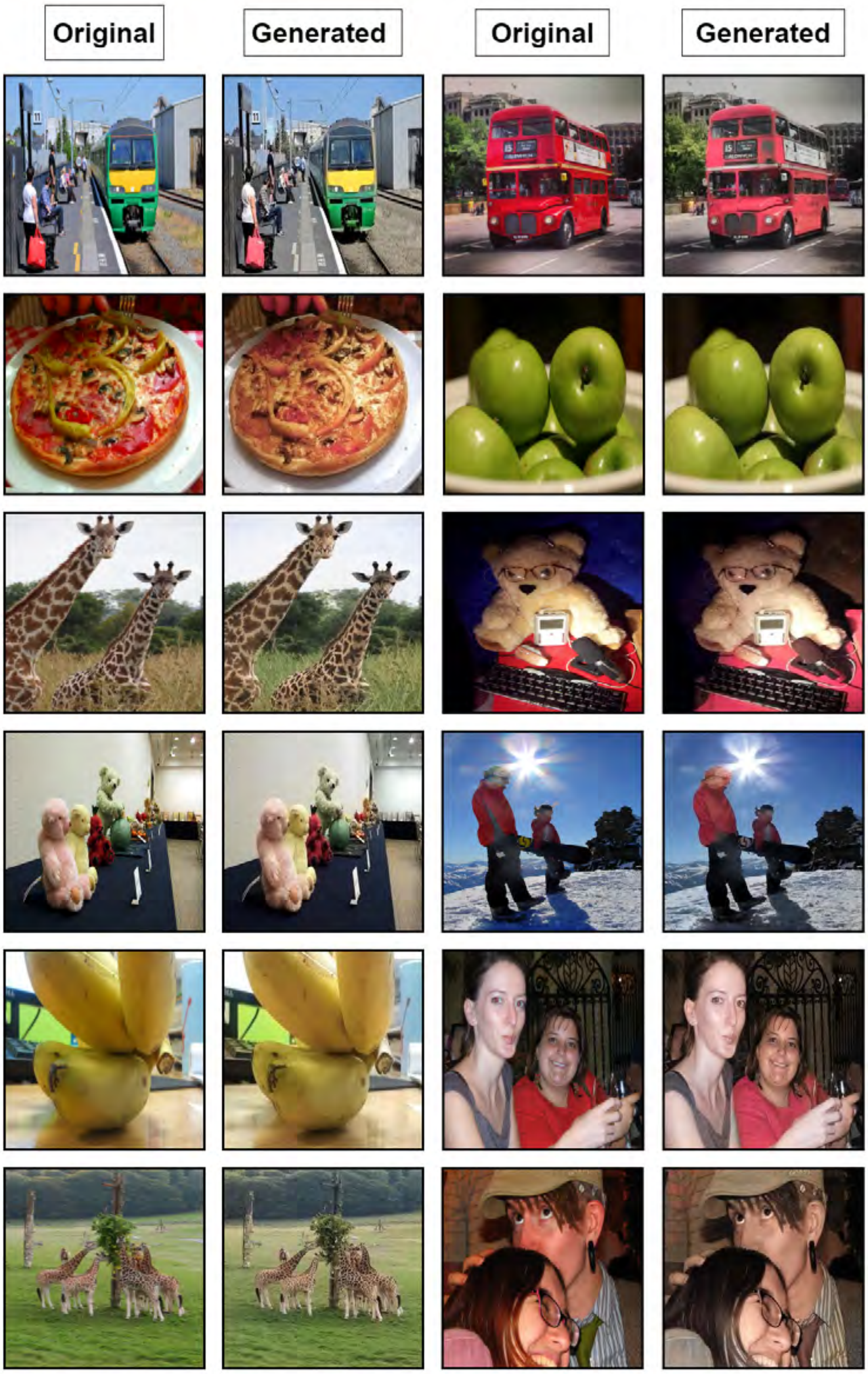


Figure 7.5: Examples of some qualitative results generated by the proposed framework [Best visible in 300% zoom].

7.2.2 Method

7.2.2.1 Color information encoding

Our work makes a significant contribution by incorporating color information encoding for object instances, thereby enhancing the accuracy of the colorization process. To achieve this, we utilize the BERT (Bidirectional Encoder Representations from Transformers) model, as introduced by Devlin et al. [17] in their paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." By leveraging BERT, we transform the textual descriptions of object instances into a compact and informative representation in the form of a 256×1 dimensional vector, denoted as $\mathbf{v}_i$. This instance-level encoding is a crucial component utilized by our proposed Instance-Level Object Colorization (IOC) module. For example, the algorithm might identify the “sky" as an object or encounter classes that were not included during the training phase, such as a “tiger." However, in many cases, it is relatively easy to obtain color descriptions for these entities, such as “blue sky" or “tiger with yellow and black stripes." Taking advantage of this additional information, we encode all available auxiliary text data associated with the scene using BERT, resulting in a 256×1 vector representation denoted as $\mathbf{v}_o$. This encoded information is then passed to the final fusion module of our framework. By incorporating the color information encoding step, we aim to address the challenges posed by complex scenes and the potential inclusion of non-object entities. These challenges may hinder the accuracy of the colorization process and impact the overall fidelity of the output. By leveraging BERT to encode the available textual descriptions, we can effectively capture and represent the color-related characteristics of both object instances and auxiliary scene elements. Our approach capitalizes on the power of BERT, which has proven to be highly effective in natural language processing tasks. By utilizing the same model for encoding both object instances and auxiliary scene information, we ensure consistency and enable a seamless fusion of color-related information across the entire scene. This integrated approach significantly improves the ability of our system to generate high-quality colorization.

7.2.2.2 Image-Object-Color

The Image-Object-Color module employs a modified U-Net-like structure to generate images based on textual input. This framework takes two primary inputs: a grayscale object image and a color encoding obtained from a pre-trained BERT model. Initially, the grayscale object image undergoes processing through an image encoder, resulting in a feature representation of dimensions 8 × 8 × 64. Simultaneously, the color encoding traverses through three fully connected layers with sizes 256, 1024, and 4096, respectively. The output of the final layer is reshaped to align with the dimensions of the image encoder's output, facilitating element-wise multiplication for textual conditioning. This text-conditioned feature vector serves as the foundation for multitasking within the Image-Object-Color module. The Image-Object-Color module branches into two outputs. One branch concentrates on reconstructing the image's color information through an image decoder consisting of four 2D convolutional layers with ReLU activation. Each layer incorporates a stride of 2 and includes batch normalization. Additionally, the decoder integrates 2D up-convolution, followed by Batch Normalization and ReLU activation in each layer, maintaining resolution consistency akin to U-Net. In the alternate output branch, convolutional layers aid in object instance classification among 80 available classes for the Masked R-CNN module. The label classifier comprises three 2D convolutional layers followed by one dense layer to predict class labels. This branch aims to deliver both color reconstruction and object classification capabilities. The Image-Object-Color module's training strategy combines multiple loss functions. For color reconstruction, the mean squared error between predicted and ground truth 'ab' channels serves as the primary loss function. Meanwhile, binary cross-entropy loss quantifies discrepancies between predicted and ground truth class labels for the object classification branch.

7.2.2.3 Object Image Translation Module

Diffusion Models, as discussed in Sohl-Dickstein et al.'s work [69], are probabilistic models designed to capture a data distribution, denoted as $p(x)$, through a gradual denoising process applied to a normally distributed variable. This denoising process, also referred to as the generation process, entails learning the inverse operation of a fixed Markov Chain with a predetermined length, denoted as T [16, 25].

In the context of image synthesis, the most effective models employ a re-weighted variant of the variational lower bound on $p(x)$. This variant closely resembles the concept of denoising score-matching, a crucial technique for achieving successful outcomes. It aids in aligning the generated samples with the target distribution by assigning appropriate weights to different regions of the latent space. Assuming u represents the latent variable, we can define the diffusion process with the following equation:

$$\frac{\partial u}{\partial t} = D\nabla^2 u \tag{7.2}$$

Here, $\frac{\partial u}{\partial t}$ signifies the partial derivative of u with respect to time t, which describes the rate of change of the function u over time. D represents the diffusion coefficient, a constant that governs the rate at which diffusion occurs. The symbol ∇^2 denotes the Laplacian operator, which is the sum of the second partial derivatives of u concerning the spatial variables. The input size of the diffusion U-Net is 256 × 256 × 3. In the U-Net encoder, a depth of 4 labels is used to generate the result better.

7.2.3 Discriminator

In order to guarantee the generation of high-quality outputs, both the IOC module and the Fusion module incorporate separate discriminators. These discriminators are responsible for evaluating the quality of generated images using a PatchGAN-based approach within each module. They operate with a standardized input size of 256 × 256 × 3. To capture the characteristics of the input image, each discriminator utilizes three convolutional layers with 64, 128, and 256 filters, respectively, along with ReLU activation functions.

To create the input for the discriminator, the grayscale input image (L_i) is combined with either a target image (T_i) or an estimated image (E_i), where T_i and E_i are the AB channel of the color image. During the training of the discriminator, the (L_i,T_i) stack is labeled as real, while the (L_i,E_i) stack is labeled as fake.

This approach ensures that the generated images are of superior quality and can be used in a variety of applications. By using PatchGAN-based discriminators, the modules can evaluate the image quality at the patch level, which allows for greater accuracy in assessing image quality. Additionally, using multiple discriminators further enhances the evaluation process and helps to ensure that the generated images are of the highest possible quality.

The use of convolutional layers with different filter sizes allows for the extraction of features at different scales, which is important for generating high-quality images. By combining the grayscale input image with either a target image or an estimated image, the modules can learn to generate color images that are highly accurate and of superior quality.

7.2.4 Loss

In our approach, we consider two main loss functions to ensure both pixel-level fidelity and high perceptual quality in the generated images. Firstly, for pixel-level fidelity, we

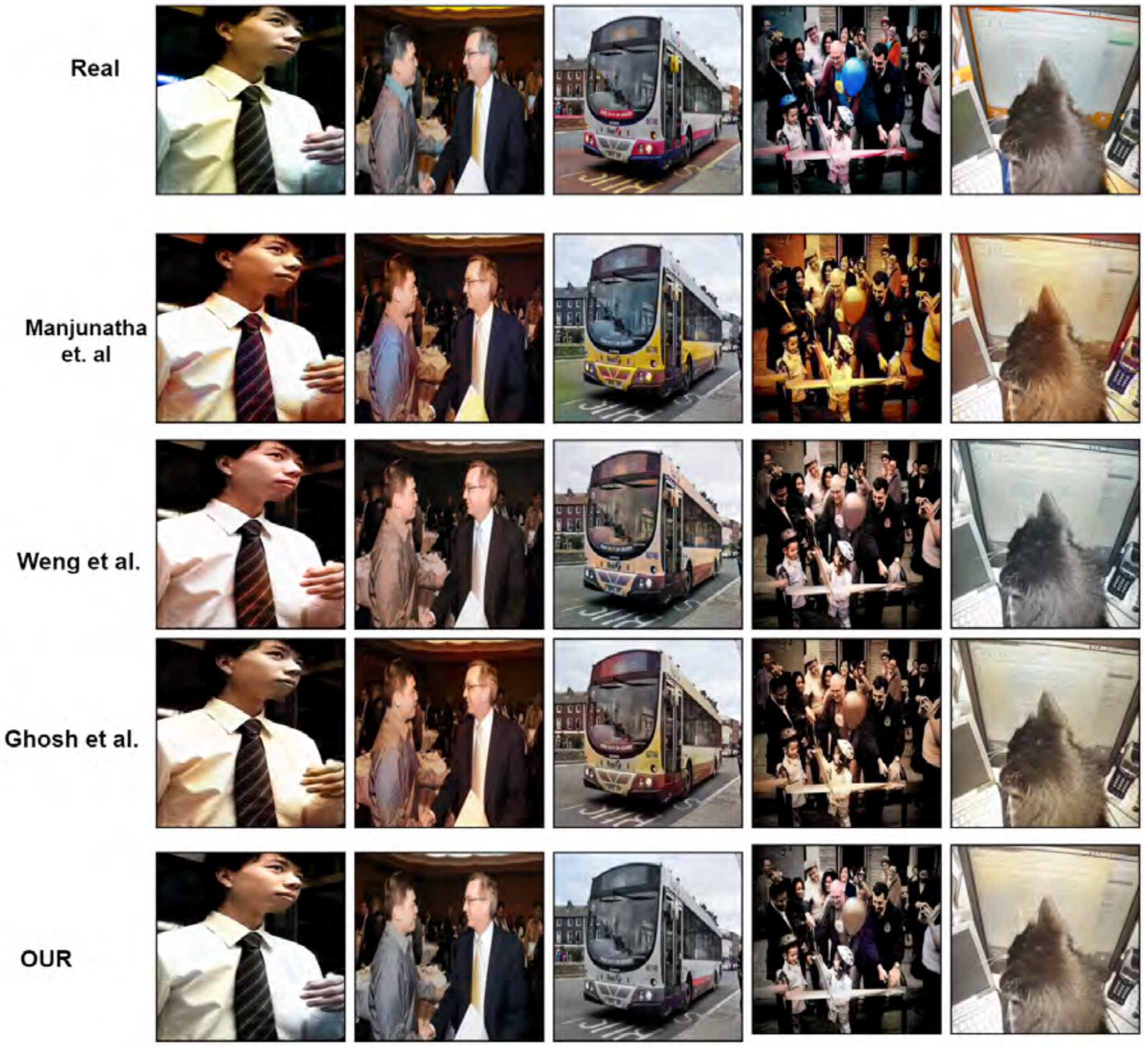


Figure 7.6: Comparison of the outcomes achieved by the proposed algorithm with those of existing text-based colorization algorithms [Optimal visibility at 300% zoom].

employ the L_1 loss, denoted as $\mathscr{L}_1^G$, calculated as:

$$\mathscr{L}_1^G = \|E_i - T_i\|_1 = \|G_m(L_i, S_i) - T_i\|_1 \tag{7.3}$$

Here, G_m represents the generator of module m mapping L_i and S_i to E_i.

Secondly, to maintain perceptual quality, we introduce a perceptual loss term $\mathscr{L}_{per}^{G_m}$ for training the generator G_m. This loss is computed as:

$$\mathscr{L}_{per}^G = \|VGG_{L_9}(T_i) - VGG_{L_9}(E_i)\|_1 \tag{7.4}$$

In this equation, $VGG_{L_9}(P)$ denotes the feature vector extracted from the ninth layer of a pre-trained VGG19 model for input P.

Additionally, for the discriminator D_m, we define the GAN loss as follows:

$$\begin{aligned}\mathscr{L}_{GAN}^{D_m} =& \mathscr{L}_{BCE}(D_m(L_i,T_i),1) \\ &+ \mathscr{L}_{BCE}(D_m(L_i,G_m(L_i,S_i)),0)\end{aligned} \tag{7.5}$$

Here, $\mathscr{L}_{BCE}$ represents the binary cross-entropy loss function, with the discriminator D_m being trained to distinguish between real and generated images.

7.3 Experiments Details, Results and Analysis

Our evaluation of the proposed method involved the generation of an extensive set of images utilizing the framework we developed. To showcase the performance of our method, we carefully selected a subset of these colorized results and presented them in Figure 7.5. The intent behind this selection was to demonstrate the exceptional effectiveness of our approach in efficiently colorizing complex scene images. The outcomes displayed in Figure 7.5 unequivocally affirm the superiority of our proposed method. One of its remarkable strengths lies in its ability to handle intricate scenes that comprise multiple objects, occlusions, shadows, and various other challenging aspects. This capability is particularly noteworthy as scenes with such complexities often pose difficulties for existing colorization methods. Our method demonstrates a high level of efficiency in achieving accurate and visually appealing colorization. By effectively capturing and reproducing the color information, it breathes life into black-and-white images, transforming them into vibrant and realistic representations of the original scenes. This is evident in the diverse examples presented in Figure 7.5, where our model successfully restores colors with impressive fidelity. The underlying framework that powers our method encompasses advanced algorithms and techniques specifically designed to address the challenges posed by complex scene colorization. Leveraging the power of deep learning and sophisticated image analysis, our model excels at understanding the intricate relationships between objects, their surroundings, and the interplay of light and shadow.

7.3.1 User study

Through conducting a user study on colorization, researchers and developers acquire valuable insights into user perceptions and interactions with colorized images. This study facilitates improvements in colorization techniques and tools, aligning them more closely with user expectations and preferences. In our study, we employed 20 colorized images generated by our model and 20 original images. By presenting a mix of these images, we questioned participants to distinguish between the artificially colored images and the genuinely colored ones. The study's outcome demonstrates that our colorization method achieves a remarkable accuracy rate of 63.68 % in misleading or deceiving the user.

7.3.2 Comparison Results

To assess the efficacy of our proposed network, we conducted comprehensive evaluations, both qualitatively and quantitatively, against current state-of-the-art methods. Table 7.1 summarizes these comparisons, illustrating that our approach surpasses existing methods across all metrics. In detail, Table 7.1 provides a comparative analysis of our method against several commonly used techniques in image colorization. Evaluation metrics include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and LPIPS. Notably, our proposed method achieves higher scores in all these metrics, underscoring its superior performance compared to existing techniques. These findings affirm the effectiveness of our approach in generating high-quality, colorized images. Moreover, we conducted a qualitative assessment by visually contrasting the outcomes of our method with those of state-of-the-art techniques. Figure 7.6 illustrates these qualitative results, showcasing side-by-side comparisons of colorized images generated by our approach alongside existing methods. These images demonstrate our method's proficiency in accurately restoring and enhancing color information across diverse image types, including natural scenes, objects, and portraits. Noteworthy strengths of our proposed network include its capacity to handle complex scenes, faithfully reproduce colors, and maintain natural color consistency. The visual appeal and fidelity of the colorized images produced by our method are evident in these comparisons, further affirming the superiority of our approach over state-of-the-art methods.

7.3.3 Ablation Study

To enhance comprehension of our approach, we conducted an ablation study, delving into the intricacies of the encoder network. Our investigation honed in on four specific setups, each representing a unique configuration. Among these, one configuration employes a sigmoid activation function, while the remaining three feature U-Net architectures

Table 7.1: Comparison of the results obtained from the proposed algorithm with those from existing colorization algorithms.

Method	LPIPS ↓	PSNR ↑	SSIM ↑
Manjunatha et al. [53]	0.282	21.055	0.853
Weng et al. [9]	0.194	22.504	0.917
Ghosh et al. [13]	0.174	23.965	0.917
Ours	**0.168**	**24.230**	**0.938**

Table 7.2: Comparison results between the proposed algorithm and existing colorization algorithms (without text).

	LPIPS ↓	PSNR ↑	SSIM ↑
lizuka et al. [29]	0.185	23.860	0.922
Zhang et al. [99]	0.234	21.838	0.895
Antic et al. [2]	0.180	23.692	0.920
Lei et al. [44]	0.191	**24.588**	0.922
Ours(without text)	**0.180**	24.214	**0.929**

Table 7.3: Ablation results of different components used in our framework. *Configuration* indicates the usage of the depth of U-Net, and activation indicates the proposed architecture along with Diffusion NET.

Configuration	LPIPS ↓	PSNR ↑	SSIM ↑
with sigmoid	0.189	17.368	0.858
Encoder depth 3	0.173	20.369	0.867
Encoder depth 4	0.168	24.230	0.938
Encoder depth 5	0.161	23.850	0.912

with varying depths. The results of our rigorous analysis are meticulously outlined in Table 7.3, furnishing a comprehensive summary of the diverse outcomes yielded by these configurations. The primary objective of our ablation study was to dissect the efficacy of different configurations within the diffusion U-Net framework. This methodology, renowned for its versatility and applicability in various domains, warranted a nuanced exploration to discern optimal configurations for specific tasks. Central to our investigation was utilizing distinct configurations characterized by specific architectural elements and parameters. The configuration employing a sigmoid activation function served as a baseline, offering insights into the performance using a conventional activation mechanism. Conversely, the remaining configurations, featuring varying depths within the U-Net architecture, allowed for an in-depth examination of the impact of architectural complexity on performance metrics. The qualitative example of ablation is in Figure. 7.7.

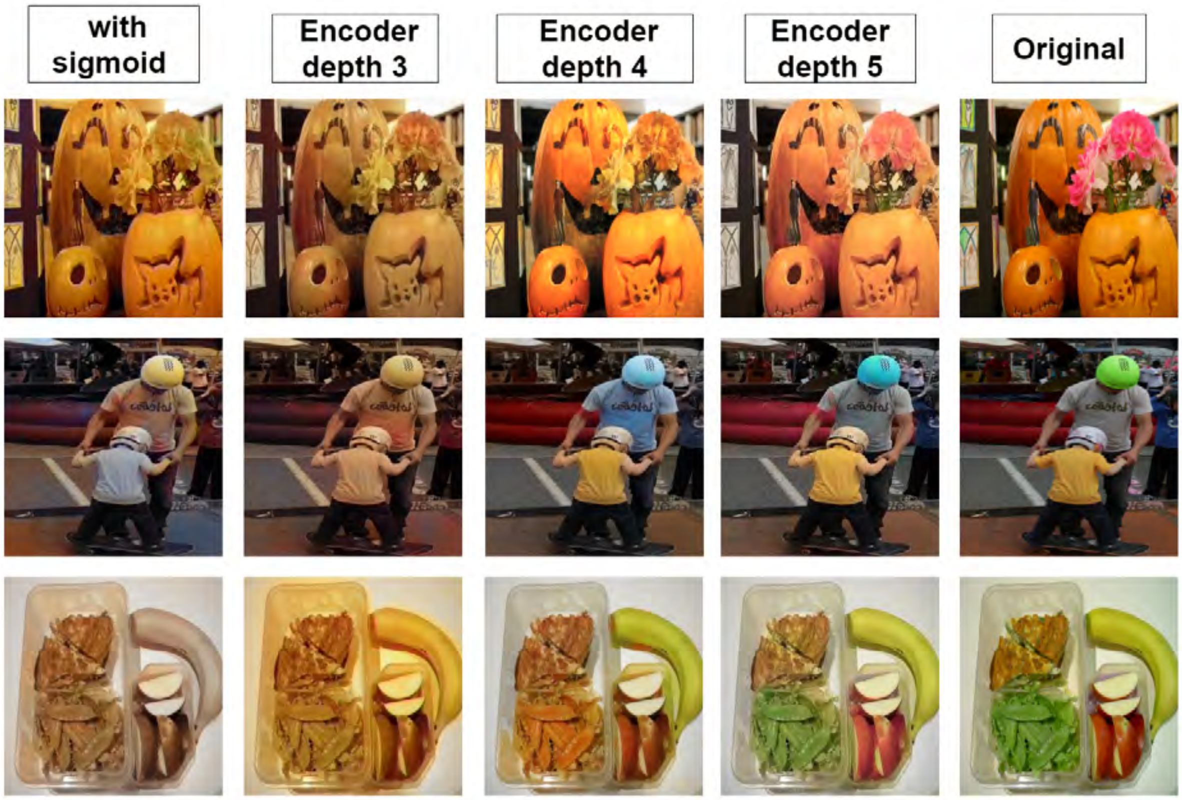


Figure 7.7: Qualitative ablation study of different components used in our framework [Best visible in 300% zoom].

7.4 Conclusion

In this work we have proposed a method for image colorization and the proposed method has demonstrated promising performance and offers several advantages over existing techniques. The evaluation of our method through both qualitative and quantitative comparisons has shown its superiority in terms of color accuracy, natural color consistency, and overall quality of the colorized images. However, it is important to acknowledge the limitations of our method. Despite the significant advancements made, colorization remains a challenging task, and there are certain scenarios where our method may encounter difficulties. For instance, images with intricate textures, fine details, or subtle color variations may produce challenging scenarios for accurately reproducing the original colors. Additionally, the performance of our method can be influenced by the quality and resolution of the input grayscale images. In future research, addressing the aforementioned limitations and challenges would be valuable. Exploring techniques to handle fine-grained details, improving color accuracy in specific scenarios, and optimizing the method's performance on diverse image types could further enhance the capabilities of our approach. Additionally, user studies and feedback can provide insights into the subjective aspects of color perception and user preferences, guiding the development of user-controlled colorization methods.

In the next chapter, we present a novel image colorization method using diffusion models, specifically Schrödinger Bridge image-to-image translation. This approach aims to overcome the limitations of existing diffusion architectures by learning the Stochastic Differential Equation (SDE) to translate random distributions into an image.

CHAPTER 8

IMAGE COLORIZATION USING DIFFUSION BY SOLVING SCHRÖDINGER BRIDGE PROBLEM

8.1 Introduction

In recent years, there has been remarkable progress in the field of generative modeling, leading to the development of various generative models. Notable examples of these models include Generative Adversarial Networks (GANs) [9, 10, 76], which have made significant contributions to the field. Additionally, advancements have been made in diffusion models [25, 26, 70, 71] and stochastic differential equation (SDE) simulations [57], further expanding the capabilities of generative modeling. The Schrödinger Bridge, which falls under the category of Stochastic Differential Equations (SDE), has shown promising outcomes in the realm of image-to-image translation models [41, 70]. Specifically, when dealing with relatively straightforward distributions, certain image-translation diffusion models have achieved optimal results in the task of image-to-image translation. However, the effectiveness of diffusion models is largely unexplored for image colorization tasks. This work explores the application of Schrödinger Bridge diffusion for image colorization by incorporating two key components: adversarial learning and regularization. By leveraging the principles of stochastic differential equations, we attempt to map a grayscale image into the respective color image. This enables us to assign appropriate colors to grayscale images, revitalizing them and enhancing their visual appeal. The proposed method also uses adversarial loss to mitigate the exponentially increasing computational

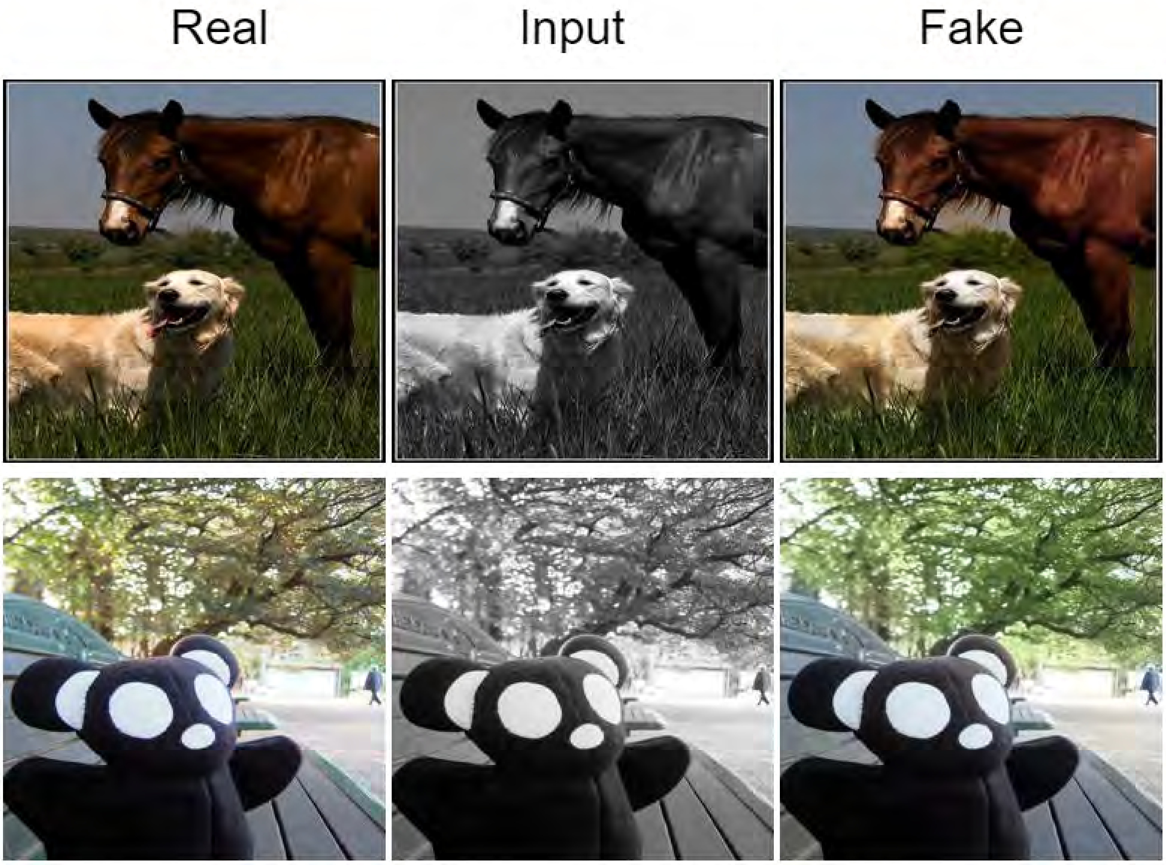


Figure 8.1: Examples of some colorized outputs from our colorization approach. The first column contains the real images, 2nd column contains the grayscale images (input), and the third column generates color images (output) [Best visible in 300% zoom].

complexity due to the high dimensionality of the color. The adversarial loss is computed by training a discriminator network to differentiate between the generated colorized images and real color images. Our method effectively learns high-quality colorization that closely resembles the ground truth through this adversarial interplay.

8.2 Proposed Framework

Colorization involves the process of adding color to black-and-white images. In this particular method, the 'L' channel of an image is used as input. To achieve this, the RGB image is first converted to the LAB color space. The LAB color space separates the image into three channels: 'L' represents the lightness or brightness, while 'A' and 'B' channels contain color information. To generate the final output color RGB images, a

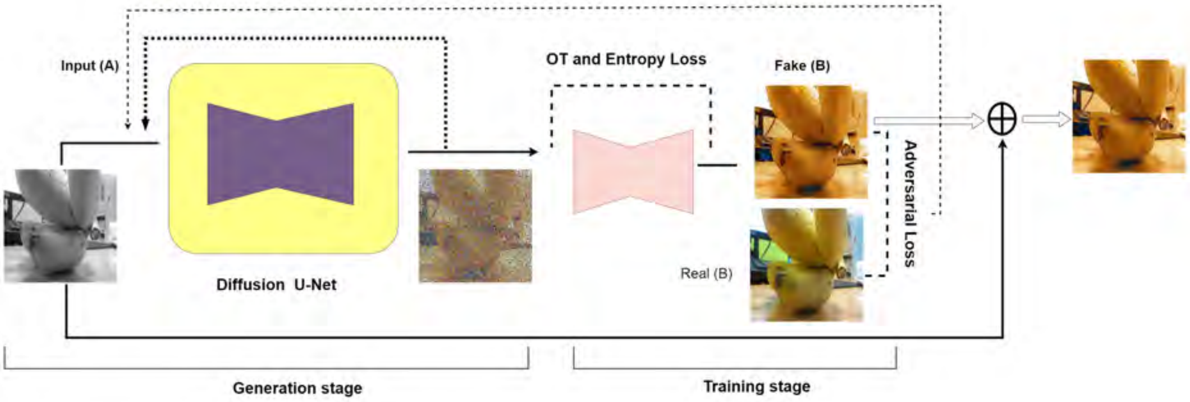


Figure 8.2: The overall architecture of the proposed colorization model.

diffusion architecture is employed that aims to progressively add noise and denoise the channel by solving a Schrödinger bridge problem. The intermediate representation of the process is passed through a generator to estimate the color channels. Finally, the estimated color channels are integrated with the luminance channel to get the final predicted color image. This pipeline ensures that the structural information present in the input grayscale image is intact in the colorization process.

Diffusion Models [69] are probabilistic models that aim to learn a data distribution $p(x)$ through a progressive denoising process of a normally distributed variable. This denoising process, also known as the generation process, involves learning the inverse procedure of a fixed Markov Chain with a length of T [16, 25].In the context of image synthesis, the most effective models utilize a reweighed variant of the variational lower bound on $p(x)$. This variant closely resembles the concept of denoising score-matching, which has been instrumental in achieving successful results. It helps in aligning the generated samples with the target distribution by assigning appropriate weights to different parts of the latent space. Assuming u as the latent variable, we define diffusion

$$\frac{\partial u}{\partial t} = D\nabla^2 u \tag{8.1}$$

$\frac{\partial u}{\partial t}$ represents the partial derivative of u with respect to time t, which represents the rate of change of the function u over time. D is the diffusion coefficient, a constant that determines the rate at which diffusion occurs. ∇^2 denotes the Laplacian operator, which is the sum of the second partial derivatives of u with respect to the spatial variables.

8.2.1 Loss

In this experiment, we employ three distinct loss functions for color generation: Schrödinger Bridge Loss, Adversarial Loss, and Noise Contrastive Estimation.

8.2.2 Schrödinger Bridge Loss

Schrödinger bridges are mathematical constructs that play a crucial role in understanding various probabilistic phenomena. They are particularly useful in situations where there is a need to connect two probability distributions or transition from one distribution to another. Let us assume that we have a grayscale image L which is a sample from the observed distribution X_i. Our target is to generate a color image C which belongs to a target distribution Y_i. In the context of image generation, Schrödinger Bridge Loss [79], also known as the Schrödinger Bridge discrepancy, is a loss function used in optimal transport theory and generative modeling as:

$$\min_{\gamma}\left[\mathbb{E}_{\gamma}[c(X_i,Y_i)]+\lambda\mathbb{E}_{\gamma}\left[D_{\mathrm{KL}}(\gamma||\mu)\right]\right] \tag{8.2}$$

where $\min_{\gamma}$ denotes the minimization over all possible transport plans γ between the source distribution X_i ($L \sim X_i$) and the target distribution Y_i, $\mathbb{E}_{\gamma}$ represents the expectation with respect to the transport plan γ, and $c(X_i,Y_i)$ is the cost function that measures the dissimilarity between the source and target distributions.

$D_{\mathrm{KL}}(\gamma||\mu)$ denotes the Kullback-Leibler divergence between the transport plan γ and a reference distribution μ. Here, λ is a trade-off parameter that controls the importance of the Kullback-Leibler divergence term relative to the cost function.

8.2.3 Adversarial Loss

Adversarial loss refers to the loss function used in Generative Adversarial Networks [20] (GANs), which are a class of deep learning models consisting of a generator and a discriminator network. Adversarial loss aims to train the generator to produce realistic data samples that can deceive the discriminator into classifying them as real. The Adversarial loss consists of two components: the generator loss and the discriminator loss. These components are designed to optimize the generator and discriminator networks simultaneously through a competitive learning process. Markovian discriminator [31] is used in patch labels in the discriminator section.

$$\begin{aligned}\mathscr{L}_{\mathrm{adv}}(G,D) = \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\log D(x)]+\\ \mathbb{E}_{z\sim p_{\mathrm{noise}}(z)}[\log(1-D(G(z)))]\end{aligned} \tag{8.3}$$

$\mathscr{L}_{\text{adv}}(G,D)$ denotes the adversarial loss, which measures the discrepancy between the generated output and the real samples.

$\mathbb{E}_{x\sim p_{\text{data}}(x)}$ represents the expectation over real data samples, where x is drawn from the real data distribution $p_{\text{data}}(x)$. $\mathbb{E}_{z\sim p_{\text{noise}}(z)}$ represents the expectation over noise samples z drawn from a noise distribution $p_{\text{noise}}(z)$, typically a uniform or Gaussian distribution. $G(z)$ represents the generator's output when given a noise sample z. It represents a generated (fake) data sample. In our case, we consider the generated colorized image as the fake sample and real-world color images as real samples to calculate the adversarial loss.

8.2.4 Noise Contrastive Estimation (NCE)

Noise Contrastive Estimation [77] (NCE) is a technique used in machine learning to estimate the parameters of a statistical model, particularly in scenarios where traditional methods are computationally demanding. NCE is commonly employed in natural language processing and other domains where modeling the data distribution directly is challenging. NCE tackles the parameter estimation problem by transforming it into a binary classification task. Rather than explicitly modeling the probability distribution of the data, NCE trains a binary classifier to differentiate between real data samples and artificially generated noise samples.

During training, NCE optimizes the parameters of the binary classifier by maximizing the likelihood of correctly classifying the paired samples. By doing so, it indirectly estimates the parameters of the underlying statistical model. The noise samples serve as a contrasting signal to guide the model towards better parameter estimation.

$$\mathscr{L}_{\text{NCE}} = -\log\left(\frac{e^{s(x,y)}}{e^{s(x,y)} + k \cdot \sum_{y'\in\mathscr{Y}} e^{s(x,y')}}\right) \tag{8.4}$$

$s(x,y)$ is the score function that measures the compatibility between a data sample x and a target distribution y, k is a constant that represents the number of noise samples drawn from a noise distribution, and $\mathscr{Y}$ represents the set of all possible target distributions. In our study, we have incorporated the NCE loss, as proposed in the work of Kim et al. [34].

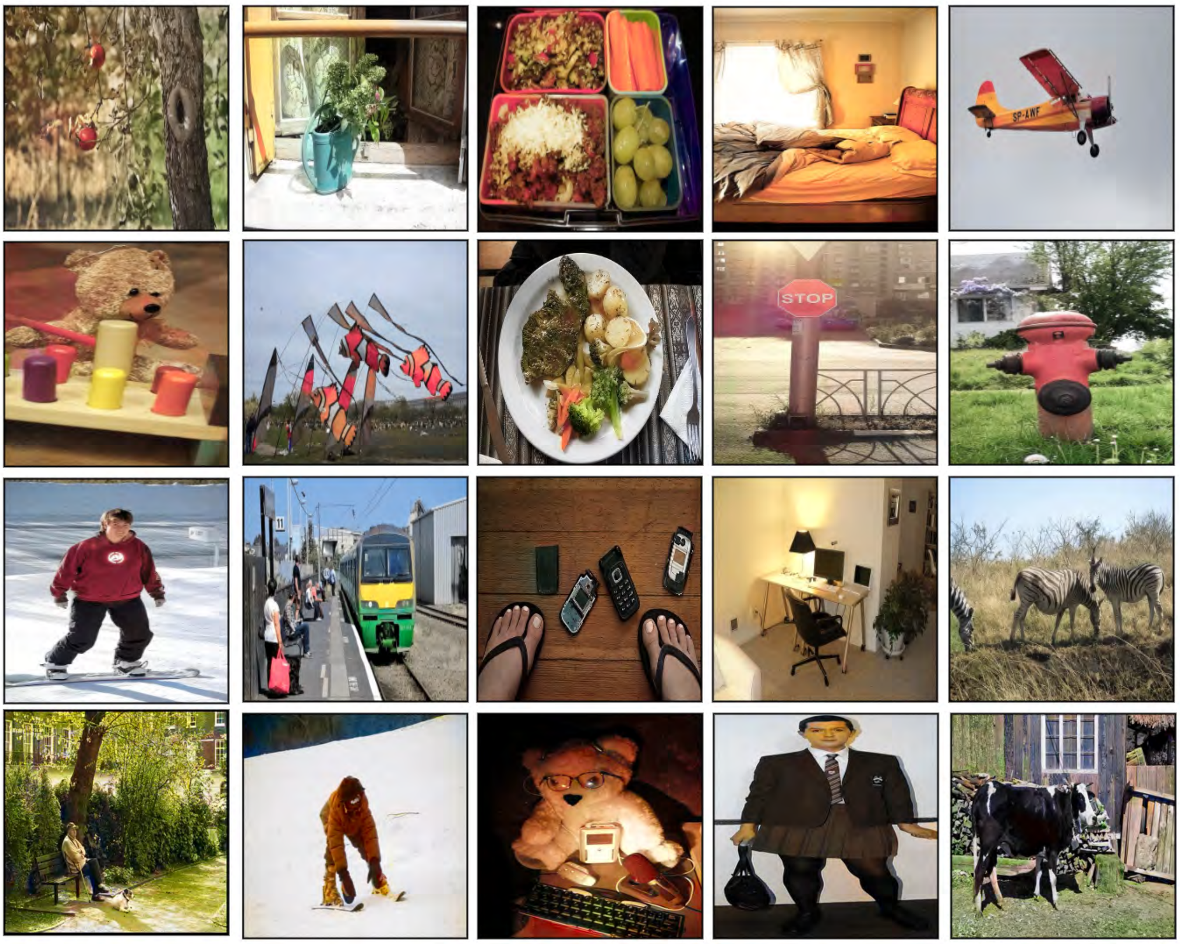


Figure 8.3: Examples of some qualitative results generated by the proposed framework [Best visible in 300% zoom].

8.3 Experiments Details, Results and Analysis

8.3.1 Dataset

We utilized the MS-COCO [49] dataset as our primary resource for generating color images. To train our model effectively, we employed a set of 118,000 images, while a separate collection of 5,000 images was reserved for testing purposes. The MS-COCO dataset served as a valuable asset for our image generation endeavors. Its extensive collection of diverse and high-quality images allowed us to train our model on a wide range of visual patterns, objects, and scenes. By leveraging this dataset, we aimed to enhance the accuracy and versatility of our image generation model. This rigorous training process involved exposing the model to a rich variety of image content, enabling it to learn and capture the intricate details and colors present in real-world scenes.

8.3.2 Quantitative Result

We employed the Adam optimizer to optimize the model, as described by Loshchilov *et al.* [50]. The initial learning rate was set to 0.0002, with momentum values of 0.5 and 0.999. These values allowed for efficient convergence and better optimization of the model parameters. In the diffusion process, we also utilized beta values to control the rate of color mixing. The β min and max values were set to 0.1 and 20.0, respectively. These values determined the amount of color diffusion applied during the colorization process, influencing the smoothness and blending of colors. By employing the PyTorch framework, we were able to leverage its powerful features and extensive library support, which facilitated the implementation of our model architecture. The chosen input size of 256×256 pixels allowed for a balance between computational efficiency and capturing finer details in the images. With 14.1 million parameters in the generator network, our model could learn complex patterns and representations in the colorization process. Adding Gaussian noise during training introduced an element of randomness, promoting robustness and preventing over fitting. The entropy parameter, set to 0.01, was crucial in controlling the level of diffusion and color variation. This parameter played a key role in achieving natural-looking colorization while maintaining structure consistency.

To assess the effectiveness of our approach, we employed four quantitative metrics. The results of these measurements are presented in Table 8.1. Firstly, we utilized the LPIPS metric introduced by Simonyan et al. in 2015 [67]. Additionally, we employed two well-known image quality assessment metrics: PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index), as proposed by Wang et al. [85]. These metrics were employed to compare the performance of our proposed framework against existing methods for image colorization. The LPIPS metric comprehensively evaluates the perceptual similarity between images, considering both low-level visual details and high-level semantic information. PSNR, on the other hand, measures the ratio of the maximum possible power of a signal to the power of corrupting noise, providing insights into the level of noise present in the colorized images. Lastly, SSIM evaluates the structural similarity between the colorized output and the ground truth images, considering luminance, contrast, and structural information. By employing these quantitative metrics, we aimed to objectively compare the performance of our proposed framework with other established image colorization methods. The results obtained from this evaluation provide valuable insights into the effectiveness and superiority of our approach. We compared it with the

Table 8.1: Comparison results of the proposed algorithm with the existing colorization algorithms

	LPIPS ↓	PSNR ↑	SSIM ↑
Vitoria et al. [78]	0.223	22.74	0.871
Kumar et al. [40]	0.210	22.74	0.898
Xia et al [93]	0.236	20.46	0.851
Lui et all [12]	0.239	22.12	0.860
Ours	**0.180**	**23.87**	**0.899**

Table 8.2: Ablation of the proposed algorithm with the loss function

	LPIPS ↓	PSNR ↑	SSIM ↑
DF Loss+NCE loss	0.220	21.84	0.820
NCE Loss+ADV Loss	0.215	22.69	0.887
DF Loss+ADV Loss	0.198	22.36	0.842
DF Loss+ADv Loss +NCE Loss	0.180	23.87	0.899

state-of-the-art algorithms [12, 40, 78, 93].

We conducted thorough comparisons with the state-of-the-art algorithm to provide additional validation for our method. Our findings indicate that our approach surpasses the quantitative results achieved by other methods. See in Table 8.1.
Additionally, we performed a qualitative comparison to demonstrate that our method better preserves the true color and structure compared to alternative approaches. As shown in Figure 8.4, our method produces better perceptual results compared to existing methods.

8.3.3 Ablation

To enhance the validation of our method, we performed ablation studies. These involved altering the number of function evaluations (NF) during the inference step, enabling us to assess their effects on the resulting output qualities. Through this analysis, we gained valuable insights into the impact of varying these parameters, further solidifying the credibility and efficacy of our approach. As shown in Figure 8.5, NF-10 produces the highest performance. For the quantitative analysis see in Tables 8.2 and 8.3.

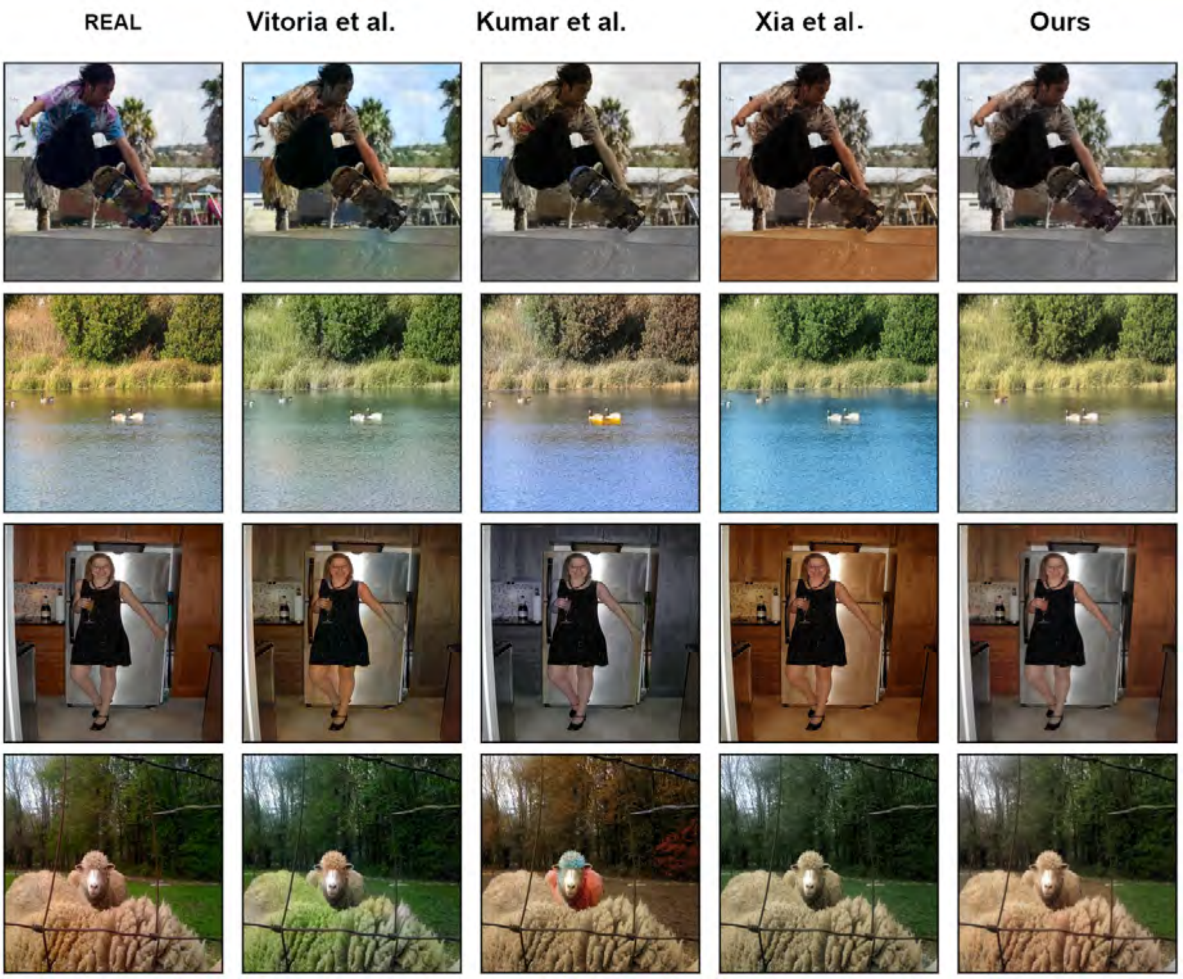


Figure 8.4: The qualitative comparison results are presented as follows: The first column comprises Real color images, while the second to fourth columns depict the generated results of the state-of-the-art (SOTA) algorithms. The last column illustrates the results generated by the proposed algorithm. SOTA methods are Vitoria et al. [78] Kumar et al. [40] Xia et al [93] [Best visible in 300% zoom].

Table 8.3: Ablation of the proposed algorithm with the number of function evaluations.

	LPIPS ↓	PSNR ↑	SSIM ↑
NF-1	0.220	21.84	0.820
NF-2	0.215	22.69	0.887
NF-3	0.198	22.36	0.842
NF-10	0.180	23.87	0.899

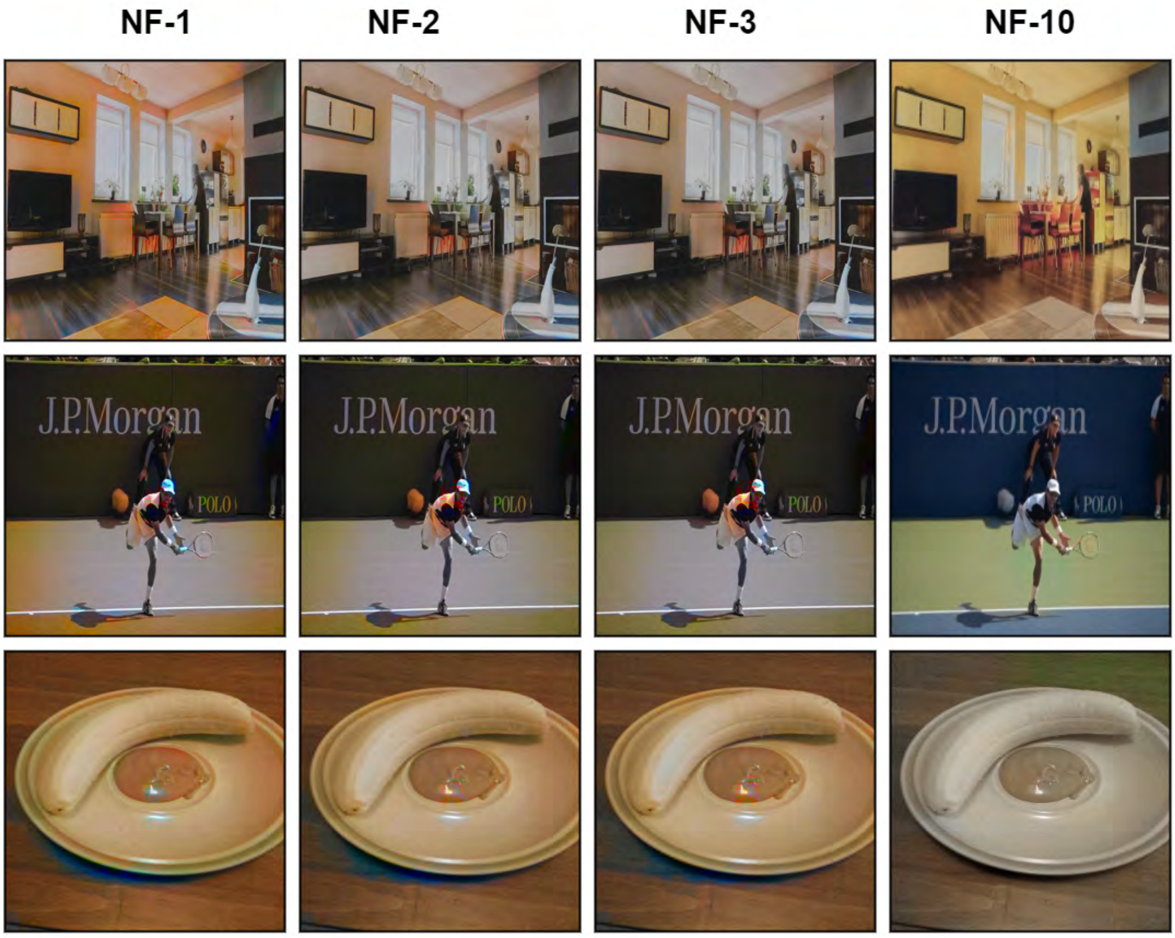


Figure 8.5: Ablation results of the different number of function evaluations (NF) during the inference step used in our framework [Best visible in 300% zoom].

8.3.4 Limitation

Figure 8.3 showcases the evident effectiveness of the proposed algorithm in successfully handling complex scenes with multiple objects, demonstrating remarkable performance overall. However, it is important to acknowledge that there are instances where the generated results may not meet expectations. As shown in Figure 8.6, the proposed technique may produce sub-optimal results in a few cases as the color consistency is not maintained.

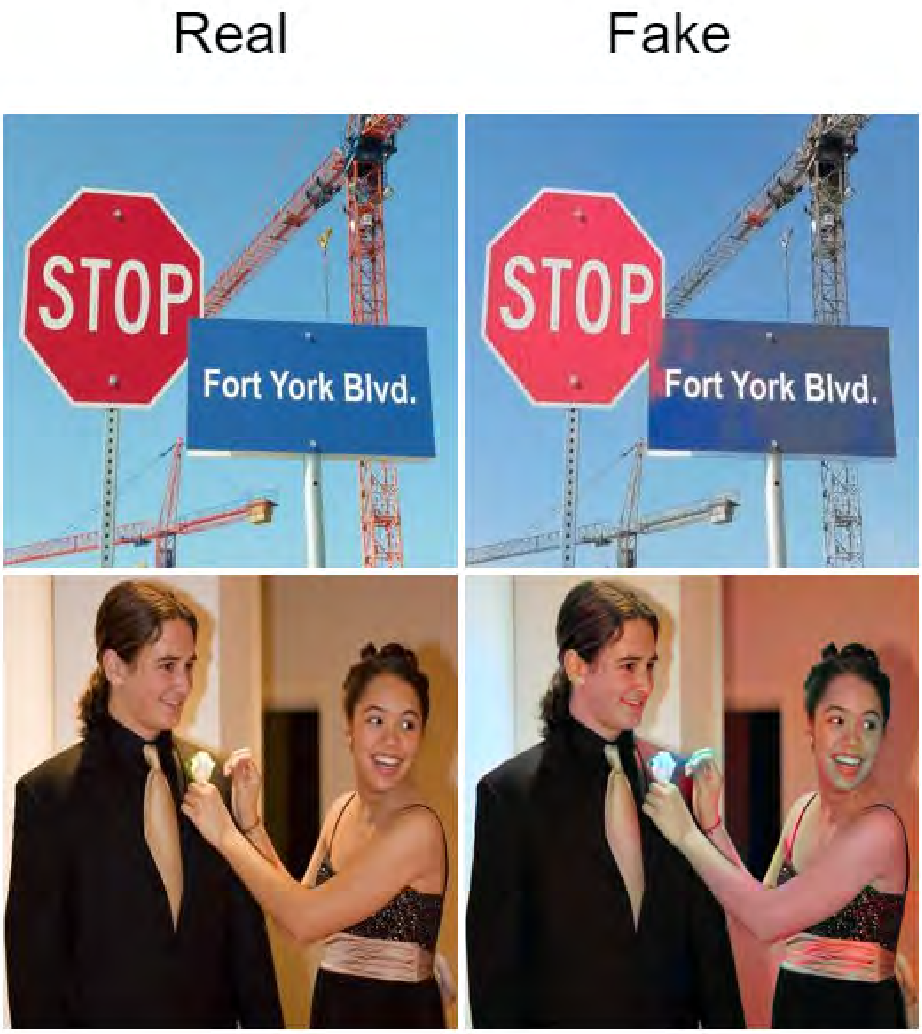


Figure 8.6: Examples of images that demonstrate a lack of proper colorization using the proposed method. The left images are real and the right images are generated [Best visible in 300% zoom].

8.4 Conclusion

This report presents a novel diffusion-based colorization method, which aims to produce natural and visually consistent colors while preserving the structure of the image. To achieve this, we trained our model using a paired approach, where we provided paired samples of grayscale and color images during the training process. Moving forward, one of our future goals is to explore the potential of unpaired colorization, which involves colorizing images without requiring corresponding paired examples. We anticipate improving the generalization and flexibility of our method by incorporating unpaired colorization techniques. This would enable our method to handle a broader range of images and scenarios, expanding its applicability and enhancing its performance in real-world use cases.

In the next chapter, we will discuss the future directions for image colorization. Additionally, we will provide a conclusion summarizing the key findings. Finally, we will outline potential areas for further research.

CHAPTER 9

CONCLUSION

This thesis presents significant advancements in the field of image colorization by introducing a novel approach that leverages text-based object information to enhance colorization accuracy and contextual relevance. Traditional image colorization methods have long relied on generalized techniques that often lack semantic depth, leading to results that may misrepresent real-world object colors. In contrast, this thesis explores the integration of textual descriptions as a means to provide auxiliary guidance for the colorization process, marking a step forward in achieving more realistic and context-aware outcomes.

9.1 Summary of Contributions

This thesis proposes several novel models towards colorization and the key contributions of the research are as follows.

- **A Long-Range Lambda Mechanism:** A long-range lambda mechanism has been proposed to preserve historical images in image colorization. Our key contribution is to include the long-range interactions without a transformer-based attention model for the colorization task. To the best of our knowledge, this is the first attempt to use lambda abstraction to invoke attention in the colorization process. Extensive experiments show that our proposed method significantly outperforms the SOTA algorithms.

- **Multi-Modal Feature Attention:** Our proposed method is a novel colorization network that uses multi-modal feature attention to color images more accurately. Our method of end-to-end model architecture produces color variations with diverse structures, shapes, and hues. We proposed incorporating additional sources of information, such as semantic segmentation maps and object recognition algorithm features. This augmentation guides the colorization process, fostering more consistent and accurate results.

- **Text-Guided Image Colorization:** A text-guided image colorization approach has been proposed in this study. Here a novel GAN pipeline is proposed that exploits textual descriptions as an auxiliary condition. We extensively evaluate our framework using qualitative and quantitative measures. In comparison with the state-of-the-art (SOTA) algorithms, it is found that the proposed method generates results with better perceptual quality. This is the first attempt to integrate textual information into an end-to-end colorization pipeline to improve the quality of generation. The textual color description acts as additional conditioning to increase the fidelity in the final colorized output.

- **Novel Dataset and Baseline Method:** This study explores colorization methods by conditioning on textual color information and introduces both a novel dataset and a baseline method for evaluation. A multi-modal pipeline is proposed that colorizes the image using language information, which is considered as an auxiliary conditioning in the colorization process. We also proposed a novel dataset based on the color information of every object of the COCO dataset. We generate color-coded textual information by associating class labels with their respective objects in the images.

- **Instance-Based Object Colorization:** An instance-based object colorization method has been developed in this study. The proposed IOC (Instance Object colorization) module utilizes instance label image colorization, exploiting object-color associations. To achieve superior performance, we design the IOC module as a multi-task network. To the best of our knowledge, this is the first attempt to design a multi-task network for the colorization task, considering the object-level instances. A multi-modal pipeline is proposed that colorizes the image using language information, which is considered auxiliary conditioning in the colorization process. To ensure high fidelity over colors, a novel loss function is proposed that captures the overall color consistency of a scene.

- **Diffusion Using Schrodinger Bridge Problem:** Here, we propose image colorization using diffusion by solving the Schrodinger Bridge Problem. This work explores the application of Schrodinger Bridge diffusion for image colorization by incorporating two key components: adversarial learning and regularization. By leveraging the principles of stochastic differential equations, we attempt to map a grayscale image into the respective color image. This enables us to assign appropriate colors to grayscale images, revitalizing them and enhancing their visual appeal. The proposed method also uses adversarial loss to mitigate the exponentially increasing computational complexity due to the high dimensionality of the color.

9.2 Challenges and Limitations

Despite its promising results, the proposed approaches have some challenges. In chapter 3, many challenges arise from differences in luminance channel distributions. We plan to make a greater effort to address this in the future. The primary challenge in this chapter is luminance control. The models discussed in chapters 4 rely heavily on quality of object recognition data. Thus, accurate identification and extraction of object-level information are critical for the success of the colorization process. In cases where object recognition fails or provides ambiguous results, these models' performance may degrade, leading to less accurate color assignments. Another challenge lies in handling ambiguous or vague textual descriptions in chapters 5, 6, 7. For example, descriptions such as beautiful scenery or bright objects provide limited guidance for colorization, making it difficult for the model to determine appropriate colors. Similarly, scenes with overlapping or occluded objects present additional complexities, as the model must disentangle the relationships between objects and their corresponding textual cues. In Chapter 8, one of the challenges is unpaired image colorization, which we addressed using only paired data.

9.3 Future Directions

To address these challenges and further enhance the models' capabilities, several avenues for future research can be explored:

- **Luminance Control:** Variations in luminance channel distributions pose significant challenges for colorization. Future efforts should focus on developing more

Table 9.1: Comparison table of Pros and Cons of the methods proposed in the thesis.

Proposed Method	**Pros**	**Cons**
Lambda-Color: Amplifying Long-Range Dependencies for Image Colorization (Chapter -3)	Captures long-range interactions without relying on transformer-based attention models. First known use of lambda abstraction to guide attention in the colorization process. End-to-end architecture captures diverse color variations in structures and shapes. Incorporates semantic segmentation maps and object features for enhanced colorization.	Poor performance in areas with reduced contrast, noise, or blur, especially in recent photos. Less vibrant backgrounds suggest a need for more comprehensive grayscale-to-color mappings.
COLOR-YOLO: Revolutionizing Image Colorization with YOLO and Cross-Attention Synergy(Chapter -4)	Proposes a novel network combining YOLO-based object recognition and cross-attention for improved accuracy. End-to-end architecture captures varied structures, shapes, and hues. Integrates semantic segmentation and object features to guide consistent and accurate colorization	May result in less vibrant or inconsistent backgrounds. Highlights the need for detailed CABA (context-aware background attention) in grayscale scenarios.
Text-Guided Image Colorization (Chapter -5)	Proposes a novel GAN pipeline using textual descriptions as auxiliary input. Demonstrates both qualitative and quantitative improvements. First to use textual conditioning to boost fidelity in colorized outputs.	When textual descriptions are coarse or vague, the model may generate inaccurate \or hallucinated colors.
Multi-Modal Colorization using Textual Descriptions (Chapter -6)	Introduces the IOC module to handle object-level instance colorization via a multi-task network. Multi-modal design leverages language as auxiliary input. Proposes a novel loss function to enforce scene-wide color consistency.	Output quality highly depends on the precision and richness of the textual input. Poorly described or ambiguous inputs reduce effectiveness.
Exploring Auxiliary Conditioning for Image Colorization (Chapter -7)	Multi-modal pipeline utilizes language as an additional condition. Introduces a novel dataset linking COCO object class labels to color information.	Faces challenges in preserving fine textures and subtle variations. Sensitive to grayscale input quality and resolution.
Image Colorization using Diffusion via Schrodinger Bridge Problem (Chapter -8)	Employs adversarial training to closely mimic ground truth color distributions. Demonstrates strong high-fidelity colorization in many cases.	Inconsistent colorization in some scenarios; color fidelity not always maintained. Occasional sub-optimal results shown in qualitative evaluations.

robust techniques for luminance normalization and control, ensuring more consistent and visually appealing results.

- **Improved Object Recognition:** Enhancing the prepossessing steps to ensure more accurate and reliable object recognitions an important step for colorization. This could involve the integration of advanced object detection models or the use of multi-modal approaches that combine visual and textual data.
- **Handling Ambiguity:** Developing mechanisms to interpret and disambiguate vague textual descriptions is crucial for text based colorization . This may be handled by including new contextual embeddings or attention mechanisms that prioritize relevant information within the text.
- **Scalability and Generalization:** Extending the models' applicability to larger and more diverse datasets, so that it can generalize effectively across a wide range of scenes and object types.
- **Real-Time Colorization:** Exploring techniques to optimize the model for real-time applications, enabling their use in interactive systems or live video processing.
- **Unpaired Image Colorization:** Current approaches predominantly rely on paired data, which limits their applicability to real-world unpaired scenarios. Future research should explore domain adaptation, self-supervised learning, or cycle-consistent generative models to improve unpaired image colorization performance while preserving color consistency and realism.

Finally, we believe that our work will inspire further advancements in image colorization to enhance the realism of visual data.

Bibliography

[1] Y. Aksoy, T. O. Aydin, M. Pollefeys, and A. Smolić, *Interactive high-quality green-screen keying via color unmixing*, ACM Trans. Graph., 35 (2016), pp. 152:1–152:12.

[2] J. Antic., *A deep learning based project for colorizing and restoring old images (and video!). https://github.com/jantic/deoldify,*, 2019.

[3] S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan, and A. W. Muzaffar, *Image colorization: A survey and dataset*, arXiv preprint arXiv:2008.10774, (2020).

[4] H. Bahng, S. Yoo, W. Cho, D. K. Park, Z. Wu, X. Ma, and J. Choo, *Coloring with words: Guiding image colorization through text-based palette generation*, in ECCV, 2018.

[5] R. Bastos, W. C. Wynn, and A. Lastra, *Run-time glossy surface self-transfer processing*, MULTIMEDIA, 2013.

[6] I. Bello, *Lambdanetworks: Modeling long-range interactions without attention*, in International Conference on Learning Representations, 2021.

[7] H. Caesar, J. R. R. Uijlings, and V. Ferrari, *Coco-stuff: Thing and stuff classes in context*, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2018), pp. 1209–1218.

[8] F. M. Carlucci, P. Russo, and B. Caputo, $(de)^2co$*: Deep depth colorization*, IEEE Robotics and Automation Letters, (2018).

[9] Z. Chang, S. Weng, Y. Li, S. Li, and B. Shi, *L-coder: Language-based colorization with color-object decoupling transformer*, in ECCV, 2022.

[10] Z. Cheng, Q. Yang, and B. Sheng, *Deep colorization*, 2015 IEEE International Conference on Computer Vision (ICCV), (2015), pp. 415–423.

[11] Y. Cho, J. Lee, S. Yang, J. Kim, Y. Park, H. Lee, M. A. Khan, D. Kim, and J. Choo, *Guiding Users to Where to Give Color Hints for Efficient Interactive Sketch Colorization via Unsupervised Region Prioritization* , in 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Los Alamitos, CA, USA, Jan. 2023, IEEE Computer Society, pp. 1818–1827.

[12] Hanyuan Liu and Jinbo Xing and Minshan Xie and Chengze Li and Tien-Tsin Wong, *Improved Diffusion-based Image Colorization via Piggybacked Models*, ArXiv, abs/2304.11105 (2023).

[13] Subhankar Ghosh and Prasun Roy and Saumik Bhattacharya and Umapada Pal and Michael Blumenstein, *TIC: text-guided image colorization using conditional generative model*, Multimedia Tools and Applications, (2023).

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, 2009 IEEE Conference on Computer Vision and Pattern Recognition, (2009), pp. 248–255.

[15] A. Deshpande, J. Rock, and D. Forsyth, *Learning large-scale automatic image colorization*, in 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 567–575.

[16] A. Deshpande, J. Rock, and D. A. Forsyth, *Learning large-scale automatic image colorization*, 2015 IEEE International Conference on Computer Vision (ICCV), (2015), pp. 567–575.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, ArXiv, abs/1810.04805 (2019).

[18] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, *The pascal visual object classes challenge: A retrospective*, International Journal of Computer Vision, 111 (2014), pp. 98–136.

[19] M. Gheini, X. Ren, and J. May, *Cross-attention is all you need: Adapting pretrained transformers for machine translation*, in Conference on Empirical Methods in Natural Language Processing, 2021.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, in The Conference on Neural Information Processing Systems (NIPS), 2014.

[21] D. Hasler and S. Süsstrunk, *Measuring colorfulness in natural images*, in IS&T/SPIE Electronic Imaging, 2003.

[22] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, *Mask r-cnn*, 2017 IEEE International Conference on Computer Vision (ICCV), (2017), pp. 2980–2988.

[23] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), pp. 770–778.

[24] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *Gans trained by a two time-scale update rule converge to a local nash equilibrium*, in Neural Information Processing Systems, 2017.

[25] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, NeurIPS, abs/2006.11239 (2020).

[26] S. Huang, X. Jin, Q. Jiang, and L. Liu, *Deep learning for image colorization: Current and future prospects*, Eng. Appl. Artif. Intell., 114 (2022), p. 105006.

[27] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, *An adaptive edge detection based colorization algorithm and its applications*, in MULTIMEDIA '05, 2005.

[28] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size*, arXiv preprint arXiv:1602.07360, (2016).

[29] S. Iizuka, E. Simo-Serra, and H. Ishikawa, *Let there be color!*, ACM Transactions on Graphics (TOG), 35 (2016), pp. 1 – 11.

[30] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, ArXiv, abs/1502.03167 (2015).

[31] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 5967–5976.

[32] X. JI, B. JIANG, D. LUO, G. TAO, W. CHU, Z. XIE, C. WANG, AND Y. TAI, *Colorformer: Image colorization via color memory assisted hybrid-attention transformer*, in European Conference on Computer Vision, 2022.

[33] X. KANG, T. YANG, W. OUYANG, P. REN, L. LI, AND X. XIE, *Ddcolor: Towards photo-realistic image colorization via dual decoders*, 2023 IEEE/CVF International Conference on Computer Vision (ICCV), (2022), pp. 328–338.

[34] B. KIM, G. KWON, K. KIM, AND J.-C. YE, *Unpaired image-to-image translation via neural schrödinger bridge*, ArXiv, abs/2305.15086 (2023).

[35] G.-Y. KIM, K. KANG, S. H. KIM, H. LEE, S. KIM, J. KIM, S.-H. BAEK, AND S. CHO, *Bigcolor: Colorization using a generative color prior for natural images*, in European Conference on Computer Vision, 2022.

[36] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[37] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, CoRR, abs/1412.6980 (2015).

[38] A. KRIZHEVSKY, *Learning multiple layers of features from tiny images*, 2009.

[39] A. KRIZHEVSKY, I. SUTSKEVER, AND G. E. HINTON, *Imagenet classification with deep convolutional neural networks*, Communications of the ACM, 60 (2012), pp. 84 – 90.

[40] M. KUMAR, D. WEISSENBORN, AND N. KALCHBRENNER, *Colorization transformer*, in International Conference on Learning Representations, vol. abs/2102.04432, 2021.

[41] G. KWON AND J.-C. YE, *Diffusion-based image translation using disentangled style and content representation*, ArXiv, abs/2209.15264 (2022).

[42] G. LARSSON, M. MAIRE, AND G. SHAKHNAROVICH, *Colorization as a proxy task for visual understanding*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 840–849.

[43] G. LEE, S. SHIN, T. NA, AND S. S. WOO, *Real-time user-guided adaptive colorization with vision transformer*, in 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2024, pp. 473–482.

[44] C. Lei and Q. Chen, *Fully automatic video colorization with self-regularization and diversity*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (2019), pp. 3748–3756.

[45] A. Levin, D. Lischinski, and Y. Weiss, *Colorization using optimization*, in SIGGRAPH 2004, 2004.

[46] Z. Liang, Z. Li, S. Zhou, C. Li, and C. C. Loy, *Control color: Multimodal diffusion-based interactive image colorization*, arXiv preprint arXiv:2402.10855, (2024).

[47] M. Limmer and H. P. A. Lensch, *Infrared colorization using deep convolutional neural networks*, 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), (2016), pp. 61–68.

[48] J. Lin, P. Xiao, Y. Wang, R. Zhang, and X. Zeng, *Diffcolor: Toward high fidelity text-guided image colorization with diffusion models*, arXiv preprint arXiv:2308.01655, (2023).

[49] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context*, in ECCV, 2014.

[50] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, in International Conference on Learning Representations, 2017.

[51] F. Luo, Y. Li, G. Zeng, P. Peng, G. Wang, and Y. Li, *Thermal infrared image colorization for nighttime driving scenes with top-down guided attention*, IEEE Transactions on Intelligent Transportation Systems, 23 (2022), pp. 15808–15823.

[52] A. L. Maas, A. Y. Hannun, A. Y. Ng, et al., *Rectifier nonlinearities improve neural network acoustic models*, in Proc. icml, vol. 30, Atlanta, GA, 2013, p. 3.

[53] V. Manjunatha, M. Iyyer, J. L. Boyd-Graber, and L. S. Davis, *Learning to color from language*, in NAACL, 2018.

[54] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, in ICLR, 2013.

[55] S. Mo, M. Cho, and J. Shin, *Instance-aware image-to-image translation*, in International Conference on Learning Representations, 2019.

[56] V. Nair and G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in ICML 2010, 2010, pp. 807–814.

[57] É. Pardoux and S. Peng, *Adapted solution of a backward stochastic differential equation*, Systems & Control Letters, 14 (1990), pp. 55–61.

[58] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. H. Gross, and A. Sorkine-Hornung, *A benchmark dataset and evaluation methodology for video object segmentation*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), pp. 724–732.

[59] N.-G. Pham, V.-H. Duong, T.-H. L. Tong, H.-N. Tran, and P.-H. Vo, *Image colorization with dif-edunet: A diffusion-based approach*, in International Conference on Intelligent Systems and Data Science, Springer, 2024, pp. 213–224.

[60] J. Redmon and A. Farhadi, *Yolov3: An incremental improvement*, ArXiv, abs/1804.02767 (2018).

[61] M. Ren, R. Kiros, and R. S. Zemel, *Exploring models and data for image question answering*, in NIPS, 2015.

[62] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in MICCAI, 2015.

[63] C. Saharia, W. Chan, H. Chang, C. A. Lee, J. Ho, T. Salimans, D. J. Fleet, and M. Norouzi, *Palette: Image-to-image diffusion models*, ACM SIGGRAPH 2022 Conference Proceedings, (2021).

[64] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, *Scribbler: Controlling deep image synthesis with sketch and color*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 6836–6845.

[65] Z. Shen, M. Huang, J. Shi, X. Xue, and T. Huang, *Towards instance-level image-to-image translation*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (2019), pp. 3678–3687.

[66] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, CoRR, abs/1409.1556 (2015).

[67] K. SIMONYAN AND A. ZISSERMAN, *Very deep convolutional networks for large-scale image recognition*, in The International Conference on Learning Representations (ICLR), 2015.

[68] K. K. SINGH, U. OJHA, AND Y. J. LEE, *Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (2018), pp. 6483–6492.

[69] J. N. SOHL-DICKSTEIN, E. A. WEISS, N. MAHESWARANATHAN, AND S. GANGULI, *Deep unsupervised learning using nonequilibrium thermodynamics*, ICML, abs/1503.03585 (2015).

[70] J. SONG, C. MENG, AND S. ERMON, *Denoising diffusion implicit models*, ICLR, abs/2010.02502 (2021).

[71] Y. SONG, J. N. SOHL-DICKSTEIN, D. P. KINGMA, A. KUMAR, S. ERMON, AND B. POOLE, *Score-based generative modeling through stochastic differential equations*, ICLR, abs/2011.13456 (2021).

[72] J.-W. SU, H. KUO CHU, AND J.-B. HUANG, *Instance-aware image colorization*, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (2020), pp. 7965–7974.

[73] P. L. SUÁREZ, A. D. SAPPA, AND B. X. VINTIMILLA, *Infrared image colorization based on a triplet dcgan architecture*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 18–23.

[74] C. SZEGEDY, W. LIU, Y. JIA, P. SERMANET, S. E. REED, D. ANGUELOV, D. ERHAN, V. VANHOUCKE, AND A. RABINOVICH, *Going deeper with convolutions*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2015), pp. 1–9.

[75] E. TOLA, V. LEPETIT, AND P. V. FUA, *A fast local descriptor for dense matching*, 2008 IEEE Conference on Computer Vision and Pattern Recognition, (2008), pp. 1–8.

[76] S. TRENESKA, E. ZDRAVEVSKI, I. PIRES, P. LAMESKI, AND S. GIEVSKA, *Gan-based image colorization for self-supervised visual feature learning*, Sensors (Basel, Switzerland), 22 (2022).

[77] A. van den Oord, Y. Li, and O. Vinyals, *Representation learning with contrastive predictive coding*, ArXiv, abs/1807.03748 (2018).

[78] P. Vitoria, L. Raad, and C. Ballester, *Chromagan: Adversarial picture colorization with semantic class distribution*, 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), (2019), pp. 2434–2443.

[79] G. Wang, Y. Jiao, Q. Xu, Y. Wang, and C. Yang, *Deep generative learning via schrödinger bridge*, in International Conference on Machine Learning, 2021.

[80] P. Wang and V. M. Patel, *Generating high quality visible images from sar images using cnns*, 2018 IEEE Radar Conference (RadarConf18), (2018), pp. 0570–0575.

[81] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, *High-resolution image synthesis and semantic manipulation with conditional gans*, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2017), pp. 8798–8807.

[82] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, *Esrgan: Enhanced super-resolution generative adversarial networks*, in The European Conference on Computer Vision Workshops (ECCVW), September 2018.

[83] Y. Wang, J. Yu, and J. Zhang, *Zero-shot image restoration using denoising diffusion null-space model*, arXiv preprint arXiv:2212.00490, (2022).

[84] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE Transactions on Image Processing, 13 (2004), pp. 600–612.

[85] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Transactions on Image Processing (TIP), (2004).

[86] C. Wei, H. Chen, L. Bai, J. Han, and X. Chen, *Infrared colorization with cross-modality zero-shot learning*, Neurocomputing, 579 (2024), p. 127449.

[87] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, *Caltech-UCSD Birds 200*, Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.

[88] Q. Wen, J. Gao, X. Song, L. Sun, and J. Tan, *Robusttrend: A huber loss with a combined first and second order difference regularization for time series trend filtering*, arXiv preprint arXiv:1906.03751, (2019).

[89] S. Weng, J. Sun, Y. Li, S. Li, and B. Shi, *Ct2: Colorization transformer via color tokens*, in European Conference on Computer Vision, 2022.

[90] S. Weng, H. Wu, Z. Chang, J. Tang, S. Li, and B. Shi, *L-code: Language-based colorization using color-object decoupled conditions*, in AAAI, 2022.

[91] D. N. B. H. Wu, J. Gan, J. Zhou, J. Wang, and W. Gao, *Fine,Äêgrained semantic ethnic costume high,Äêresolution image colorization with conditional gan*, International Journal of Intelligent Systems, 37 (2022), pp. 2952 – 2968.

[92] Y. Wu, X. Wang, Y. Li, H. Zhang, X. Zhao, and Y. Shan, *Towards vivid and diverse image colorization with generative color prior*, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), (2021), pp. 14357–14366.

[93] M. Xia, W. Hu, T.-T. Wong, and J. Wang, *Disentangled image colorization via global anchors*, ACM Transactions on Graphics (TOG), 41 (2022), pp. 1 – 13.

[94] Y. Xiao, A. Jiang, C. Liu, and M. Wang, *Semantic,Äêaware automatic image colorization via unpaired cycle,Äêconsistent self,Äêsupervised network*, International Journal of Intelligent Systems, 37 (2022), pp. 1222 – 1238.

[95] Y. Xiao, P. Zhou, and Y. Zheng, *Interactive deep colorization using simultaneous global and local inputs*, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), (2019), pp. 1887–1891.

[96] J. xiong Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, *Sun database: Large-scale scene recognition from abbey to zoo*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2010), pp. 3485–3492.

[97] N. Zabari, A. Azulay, A. Gorkor, T. Halperin, and O. Fried, *Diffusing colors: Image colorization with text guided diffusion*, in SIGGRAPH Asia 2023 Conference Papers, 2023, pp. 1–11.

[98] B. Zhang, M. He, J. Liao, P. V. Sander, L. Yuan, A. Bermak, and D. Chen, *Deep exemplar-based video colorization*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (2019), pp. 8044–8053.

[99] R. Zhang, P. Isola, and A. A. Efros, *Colorful image colorization*, in ECCV, 2016.

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

[101] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, *Real-time user-guided image colorization with learned deep priors*, ACM Transactions on Graphics (TOG), 36 (2017), pp. 1 – 11.

[102] Y. Zhao, L.-M. Po, W. Y. Yu, Y. A. U. Rehman, M. Liu, Y. Zhang, and W. Ou, *Vcgan: Video colorization with hybrid generative adversarial network*, ArXiv, abs/2104.12357 (2021).

[103] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, *Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios*, 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), (2021), pp. 2778–2788.