# User-guided Interactive Colorization Of Grayscale Images

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

**Ruida Jiang**

Supervisor: Dr. François Pitié

Department of Electronic and Electrical Engineering
Trinity College Dublin

This dissertaton is submitted in partial fulfillment for the degree of
*MSc in Electronic Information Engineering*

Trinity College Dublin July 2025

I would like to dedicate this thesis to my loving parents, my supervisor Professor François Pitié, and to my own efforts throughout this journey.

# Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work. It contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year. I have also completed the Online Tutorial provided by the Library, Trinity College Dublin, on avoiding plagiarism 'Ready, Steady, Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steadywrite`

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Ruida Jiang
July 2025

# Acknowledgements

Firstly, I would like to express my deepest gratitude to my supervisor, Professor François Pitié, for his invaluable guidance, patience, and continuous encouragement throughout this research. His insightful feedback and unwavering support have greatly contributed to the completion of this work.

Secondly, I sincerely thank Professor Anil Kokaram, Professor François Pitié, and other faculty members for the courses they delivered, which have instilled in me both rigor and passion in the fields of video and image engineering, consistently inspiring me throughout this project. I would also like to extend my appreciation to my friend, Yiwei Liu, whose guidance on the tools essential for this project has been instrumental in facilitating my continuous exploration and progress. Additionally, I am grateful to my classmates for their help, companionship, and support throughout my studies, leaving me with cherished memories of our shared experiences.

Finally, my heartfelt thanks go to my parents for their unconditional care, unwavering support, and encouragement in all aspects of my life.

# Abstract

This thesis proposes a user-guided interactive image colorization framework based on a collaborative architecture, where the ColorNet network is responsible for image colorization using grayscale images and sparse user click information, while the EditNet network analyzes current colorization results and predicts the optimal next click position to reduce the number of user interactions and improve colorization quality. Additionally, two gradient estimation strategies, ArgmaxSTE and Gumbel-Softmax, are introduced to achieve efficient local color optimization and more stable global color coordination, respectively. Experimental results demonstrate that the proposed method can achieve high-quality colorization effects with fewer user interactions, significantly superior to random click methods, while effectively mitigating the click position collapse problem during the interaction process.

# Table of contents

# List of figures

# List of tables

# CHAPTER 1

# Introduction

## 1.1 Background and Motivation

Image colorization technology, as an important branch of computer vision, has attracted widespread attention since the early 20th century. From early manual colorization techniques to modern automatic colorization systems based on deep learning, this technology has demonstrated tremendous application value across multiple fields, including historical photo restoration, film remastering, artistic creation, and the entertainment media industry. In particular in the digital age, large amounts of precious historical visual materials need to be revitalized through colorization technology to provide modern audiences with more vivid and realistic visual experiences.

Traditional image colorization methods are mainly based on manual operations, which require professional technicians to spend considerable time making fine adjustments to each pixel. Although this approach can produce high-quality results, its efficiency is extremely low and its costs are high, making it difficult to meet the demands of modern large-scale digital content processing. With the rapid development of deep learning technology[13], automatic colorization methods based on Convolutional Neural Networks (CNN) have begun to emerge. These methods can automatically generate reasonable color information for grayscale images by learning color distribution patterns from large-scale datasets.

However, fully automated colorization methods face a fundamental challenge: the multi-solution problem of color prediction. The same grayscale image may correspond to multiple reasonable color schemes, while purely automated systems

often tend to generate conservative, desaturated results that are difficult to satisfy users' personalized needs and creative expression. To address this problem, user-guided interactive colorization methods have emerged. These methods guide the colorization process by introducing sparse user input (such as color scribbles or clicks), maintaining automated efficiency while providing users with precise control over the final results.

Although user-guided colorization has significant theoretical advantages, existing methods still face a critical issue: the efficiency problem of user interaction. Traditional methods typically require substantial user input to achieve satisfactory colorization effects, which not only increases users' workload but also limits the practical application value of the technology. Particularly in large-scale application scenarios such as video colorization, how to minimize user interaction frequency while maximizing colorization quality has become a core problem that urgently needs to be solved.

## 1.2 Problem Statement

As mentioned previously, color ambiguity is the primary bottleneck for fully automatic methods. When a model faces an uncertain region, in order to minimize expected error, the model's final output is often a weighted average of this probability distribution, which visually typically manifests as a lifeless grayish-brown or faded appearance. This conservative strategy, while avoiding obvious color errors, also sacrifices the image's realism and visual impact, making it difficult to meet high-quality application requirements.

Although user-guided interactive colorization methods can effectively address the limitations of automatic colorization, they also face unique technical challenges. The primary challenge lies in the efficiency problem of user interaction. Traditional scribble-based or click-based interactive colorization methods typically require users to provide substantial input to achieve satisfactory results. For example, existing methods may require users to perform hundreds of click operations to complete the colorization of a complex image, greatly reducing user experience and practical application value.

Another important challenge is the accuracy problem of color propagation. The sparse color information provided by users needs to be propagated to the entire image region through some mechanism, but existing methods are prone to issues such as color bleeding and unclear boundaries during the color propagation process, particularly when processing images with complex segmentation boundaries. This inaccurate color propagation not only produces visually unnatural results but may

also require users to perform additional correction operations, further increasing the interactive burden.

In click-based user-guided colorization methods, the selection of user click positions has a decisive impact on the final colorization effect. However, existing methods typically employ random sampling or simple heuristic rules to determine click positions, and these approaches have universal shortcomings. First, random sampling may lead to important regions being overlooked while unimportant regions are over-sampled, thereby reducing colorization efficiency. Second, simple heuristic rules (such as gradient-based or texture-based sampling) are often overly limited and cannot adapt to the diverse needs of different types of images. Therefore, developing an automated method that can intelligently select click positions is of significant importance for improving the efficiency and quality of interactive colorization.

## 1.3 Research Objectives

Based on the analysis of the above problem analysis, the main objective of this research is to develop an efficient user-guided interactive image colorization system that can significantly reduce the required number of user interactions while ensuring colorization quality.

Specifically, design and implement an end-to-end deep learning framework that contains two core components: ColorNet and EditNet. ColorNet is responsible for performing image colorization tasks based on sparse user clicks, while EditNet is responsible for automatically analyzing current colorization results and predicting the optimal next click position. This collaborative architecture of dual-network aims to achieve an organic combination of user interaction and automatic optimization, thereby improving colorization quality while reducing manual intervention. This research will provide a detailed introduction to ColorNet's network architecture design and training strategy in Chapter 3, elaborate on EditNet's structural design and click position prediction mechanism, and conduct an in-depth discussion of the end-to-end training framework and joint optimization strategy for both networks.

Also, research and implement an algorithm capable of automatically selecting optimal user click positions. This algorithm should be able to intelligently determine the next most valuable click position based on current colorization state and target image features. Through this approach, the system can achieve a maximum colorization improvement with minimal user input. The specific implementation details of the click position optimization algorithm will be elaborated in Chapter 3, while the effectiveness validation and performance analysis of this algorithm will be comprehensively demonstrated in the click distribution analysis in Chapter 4.

Addressing the problem of excessive concentration of click positions in edge and shadow areas in existing methods, we need to develop a mechanism that can ensure a reasonable distribution of click positions. This mechanism should be able to balance the needs of local detail optimization and global color coordination, avoiding neglect of certain important areas while preventing over-processing of unimportant regions. This research will conduct an in-depth analysis of solutions to the click position clustering problem in Chapter 3, and verify the correlation between improved distribution and ground truth through experiments. Related implementation details and algorithmic improvements will be specifically described in Chapters 3 and 4.

Furthermore, this research will systematically compare the performance of two gradient estimation techniques, ArgmaxSTE and Gumbel-Softmax, in interactive colorization tasks. Chapter 3 will provide a detailed introduction to the technical principles and implementation details of these two gradient estimation methods, while Chapter 4 will provide comprehensive quantitative and qualitative comparison results, including training stability analysis and convergence speed comparison, providing important technical references for related research fields. Finally, we will conduct an in-depth discussion of the core findings of this research in Chapter 6, summarize the main contributions of this study, and prospective future research directions.

# CHAPTER 2

Literature Review

## 2.1 Technical Foundations

Image colorization technology, as an important interdisciplinary field of computer vision and graphics, aims to endow grayscale images with reasonable and natural color information. This technology not only holds significant academic value, but also shows enormous commercial potential in practical applications such as digital content creation, historical image restoration, and entertainment media production. The image colorization problem is essentially an ill-posed problem, meaning that the same grayscale image may correspond to multiple reasonable color schemes, making research in this field both challenging and rich with innovative possibilities.

### 2.1.1 Lab Color Space

Modern image colorization technology is primarily based on Lab color space processing, a choice with profound theoretical foundations. The Lab color space consists of three channels: L (lightness), a* (green-red component) and b* (blue-yellow component)[1]. Its greatest advantage lies in completely separating luminance information from chrominance information, allowing the colorization task to focus on predicting chrominance components while keeping the brightness structure of the original image unchanged. In colorization tasks, the L channel directly preserves the brightness information of the original image, while the network only needs to predict the a* and b* chrominance channels. This design not only reduces the complexity of the problem but also ensures complete preservation of spatial structural

information. Furthermore, in practice, the application of the Lab color space makes network training more stable with faster convergence speeds. Additionally, Lab color space possesses perceptual uniformity characteristics, meaning that the Euclidean distance can better reflect human visual perception differences, making optimization based on this space more aligned with human visual characteristics.

### 2.1.2 CNN Colorization

The application of convolutional neural networks in image colorization has undergone a development process from simple regression to complex generative models. Early CNN colorization methods mainly employed end-to-end regression training with relatively simple network architectures, typically including several fully connected and convolutional layers. The main problem with these methods was their tendency to produce desaturated results, as L2 loss functions would drive the network to predict average colors to minimize error.

The introduction of classification methods[25] marked an important breakthrough in CNN colorization technology. By converting continuous color prediction into discrete category prediction, networks can learn richer color distributions. The implementation of this approach typically involves quantization of the Lab color space, mapping continuous ab values to finite color categories. During training, the network learns to predict the probability of each pixel belonging to various color categories, and during inference, the final color values are generated through expectation calculation or sampling methods.

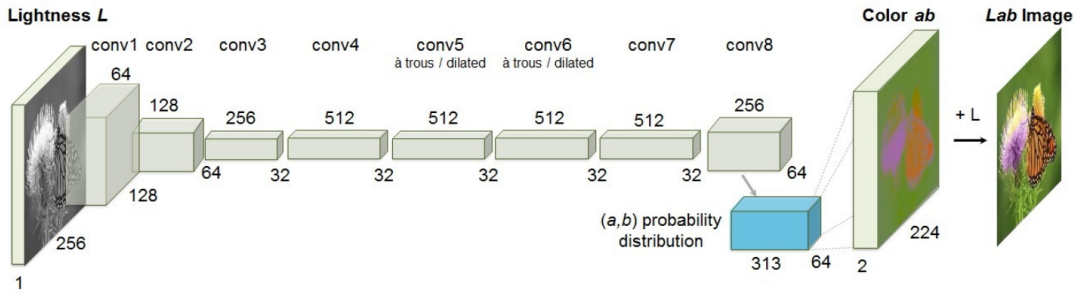



Fig. 2.1 CNN-based image colorization architecture proposed by Zhang et al.[25]. The network predicts color distributions by formulating the colorization task as a classification problem in the Lab color space, significantly improving the color vibrancy and diversity compared to traditional regression-based methods..

The loss function design occupies a central position in CNN colorization. Beyond traditional L1 and L2 losses, modern methods widely adopt advanced loss functions such as perceptual loss, adversarial loss, and feature matching loss. Perceptual loss evaluates the semantic similarity of images by comparing high-level features of

pretrained networks (such as VGG), capable of producing more realistic visual effects. Adversarial loss evaluates the authenticity of generated images through discriminator networks, promoting networks to generate more natural and diverse colors.

### 2.1.3 U-Net Architecture

The choice of deep learning network architecture also has a decisive impact on colorization results. The U-Net architecture[19] demonstrates excellent performance in image colorization tasks due to its unique encoder-decoder structure and the skip connection mechanism. The encoder path extracts high-level semantic features of images through progressive downsampling, while the decoder path recovers spatial resolution through upsampling, and cross-layer skip connections ensure effective transmission of low-level texture and edge details. This design enables the network to understand both the global semantic content of images and to accurately process local detail information, which is crucial to producing high-quality colorization results. The research by Hu et al. (2024)[9] further confirms the advantages of combining the U-Net architecture with the Lab color space. Their experiments demonstrate that U-Net's encoder-decoder structure can exhibit superior performance in multiscale feature fusion and semantic consistency maintenance, which is highly consistent with the design philosophy of ColorNet in this research.

### 2.1.4 Click Iterative Methods

The theoretical foundation of click-iterative methods can be traced back to classical optimization-based colorization algorithms. The optimization framework proposed by Levin et al.[15] is based on a core assumption: spatially adjacent pixels with similar brightness should have similar colors. This assumption can be mathematically expressed through the Laplacian matrix, transforming the colorization problem into a solution of large-scale sparse linear systems. Although this method is mathematically rigorous, it has high computational complexity and difficulty handling high-resolution images.

Modern deep learning-based click iterative methods directly learn the mapping relationship from sparse prompts to complete colorization through end-to-end training[27]. The network architectures of such methods typically include two input branches: one processing grayscale images and another processing sparse color information provided by users. The features of these two branches are fused through attention mechanisms or feature concatenation to generate final colorization results.

The encoding of click information is a key technology for achieving effective iteration. Common encoding methods include: 1) **Heat map encoding**, which encodes

click position and color information as multichannel heat maps[14]; 2) **Attention encoding**, which allows networks to automatically learn how to use click information through attention mechanisms[14]. Modern methods typically adopt learnable encoding strategies, allowing networks to automatically optimize the representation and propagation of click information.

The design of iterative optimization strategies directly affects the efficiency and effectiveness of user interaction. Simple iterative methods run the entire network after each user click, which is simple but inefficient. More advanced methods adopt incremental update strategies, recalculating only image regions affected by new clicks, significantly improving the interactive response speed[27]. Some of the latest methods even introduce predictive interaction, where systems can predict the likely next click positions of users and perform calculations in advance to further reduce response delays[20].

## 2.2 Automatic Colorization Methods

The development of automatic image colorization methods has undergone a fundamental transformation from traditional mathematical optimization to modern deep learning. Early automatic colorization research was primarily based on statistical learning and optimization theory, establishing grayscale-to-color mapping relationships by analyzing statistical characteristics of large numbers of color images. Although these methods could produce reasonable results under specific conditions, they generally suffered from poor adaptability, high computational complexity, and difficulty handling complex scenes.

The introduction of deep learning technology brought revolutionary breakthroughs in automatic colorization. The pioneering work "Colorful Image Colorization" published by Zhang et al.(2016)[25] first redefined the colorization problem as a classification task, fundamentally solving the problem of desaturated results produced by traditional regression methods. The core innovation of this method lies in converting continuous color prediction into discrete category prediction, transforming the colorization problem into predicting probability distributions over 313 color categories for each pixel by quantizing the ab channels in Lab color space. More importantly, this work introduced a class rebalancing mechanism that significantly improved the color saturation and diversity of generated images by adjusting weights of different color categories to overcome the problem of uneven color distribution in natural images.

The work published by Iizuka et al.(2016)[10] in the same year further expanded the technical framework of automatic colorization, proposing the concept of global-

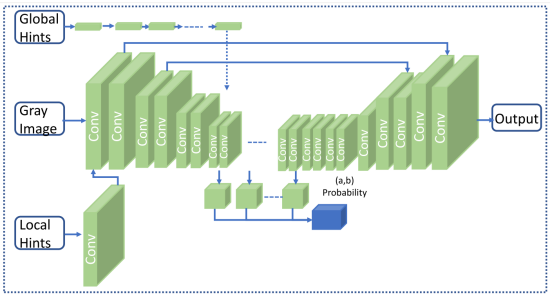
local feature fusion. This method used a dual-branch network architecture to separately extract global semantic features and local texture features of images, then generated final colorization results through a fusion mechanism. This design enabled the network to both understand the overall semantic content of images (such as sky should be blue, grass should be green) and process local detail information, thereby producing more reasonable and natural colorization effects.

The application of Generative Adversarial Networks (GANs)[6] brought new technical possibilities to automatic colorization[21–23]. GAN-based colorization methods can generate more realistic and diverse color results through adversarial training mechanisms. The introduction of discriminator networks makes the generator consider not only pixel-level color accuracy but also ensure that the generated color images possess realism in the overall visual effects[21]. This training mechanism is particularly suitable for handling scenes with multiple reasonable colorization schemes, avoiding the tendency of traditional methods to generate "safe" but uncreative averaged results.

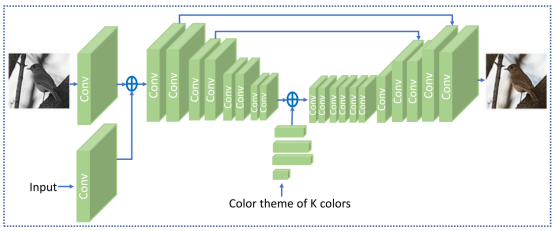
Modern automatic colorization methods increasingly focus on multiscale feature fusion and the utilization of semantic information. Many advanced methods adopt pyramid structures or multi-resolution processing strategies to extract and fuse feature information on different scales[28]. Meanwhile, prior knowledge of advanced visual tasks such as semantic segmentation and object detection is widely applied in colorization systems, guiding color allocation by understanding semantic categories of different objects in images[17]. This semantic understanding-based colorization approach can produce results more aligned with human cognition, such as correctly assigning the corresponding green tones to different types of plants or selecting appropriate color characteristics for objects of different materials.

## 2.3 User-Guided Colorization

User-guided interactive colorization methods represent an important technical shift from complete automation toward user-controllable approaches. These methods guide the colorization process by introducing sparse user input, providing users with precise control capabilities while maintaining automated efficiency. The technical core of user-guided colorization lies in how to effectively propagate local color information provided by users to the entire image region while maintaining spatial consistency and semantic reasonableness of colors.




(a) Real-time User guided Colorization[27]




(b) Interactive Deep Colorization [24]

Fig. 2.2 User-guided colorization architectures: (a) Real-time user-guided colorization architecture with dual-network design for local and global hint processing. [2], and (b) Interactive deep colorization network supporting simultaneous global and local inputs via U-Net architecture. [2]

### 2.3.1 Real-time user-guided colorization

Zhang et al.(2017)[27] established the technical standards for modern interactive colorization. The breakthrough contribution of this system lies in achieving a truly real-time interactive experience, where users can specify colors on grayscale images through simple click operations, and the system can immediately provide high-quality colorization feedback.

The technical architecture of this method employs an innovative dual-network fusion design. The Global Priors Network, trained on large-scale data, can understand the overall semantic content of images and provide fundamental capabilities for automatic colorization. The Local Hints Network specifically processes color hints provided by users, learning how to effectively propagate sparse click information to similar image regions. These two networks are combined through a carefully designed fusion mechanism that maintains the semantic reasonableness of automatic colorization while ensuring precise control effects of user input.

The main network adopts a carefully designed ten-block structure, with each block consisting of two to three convolutional layers, followed by ReLU activation functions and batch normalization processing. The first four blocks of the network extract high-level semantic information by continuously halving the spatial dimensions of the feature tensors while doubling the feature dimensions, while the latter four convolutional blocks restore the spatial resolution in reverse order. The fifth

and sixth blocks employ dilated convolutions (with a dilation factor of 2) to increase the receptive field, while establishing symmetric skip connections between different blocks to recover spatial information. All convolutional layers use 3×3 convolution kernels, with the final layer using 1×1 convolution kernels to map the output of the tenth block to the final result. The loss function combines Huber loss and regression loss, ensuring the accuracy and stability of color prediction.

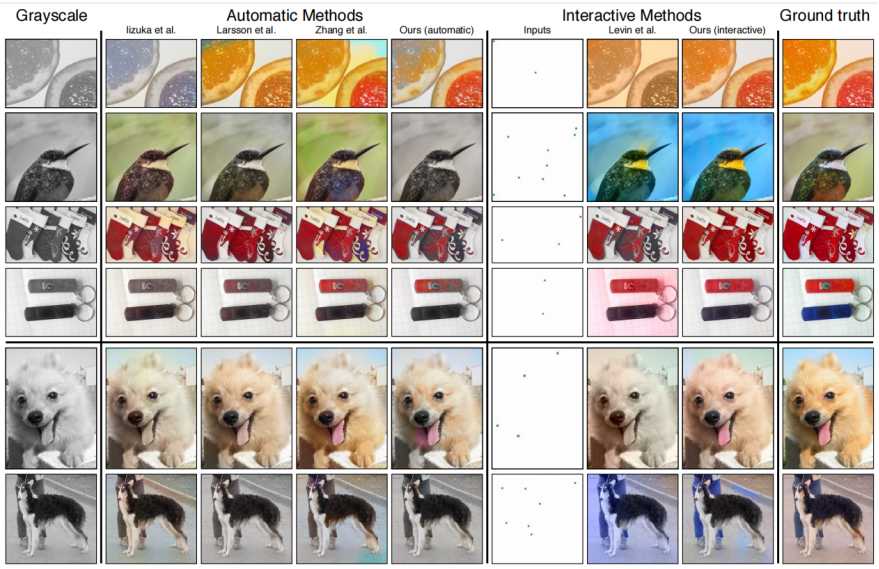



Fig. 2.3 Qualitative comparison between automatic and interactive colorization methods with user guidance [27].

### 2.3.2 Interactive deep colorization

The development of interactive deep colorization technology embodies the perfect combination of deep learning and human-computer interaction design. Modern interactive colorization systems not only require technical advancement but also need to meet professional-grade application standards in user experience design. The design philosophy of such systems is to maximize colorization improvement by minimizing user input, achieving the optimal balance between efficiency and quality.

The fusion of multi-input modes represents an important breakthrough in modern interactive colorization systems. Traditional systems often support only a single type of user input, either supporting only global color control or allowing only local color clicking. The innovative method proposed by Xiao et al.[24] broke this limitation, achieving simultaneous support for global and local inputs. This method introduces color themes as global input, allowing users to control the overall color style of images through 3-5 representative colors while retaining precise control

capabilities for local color points. This design enables users to quickly establish the overall tone of images while making precise adjustments to regions of interest.

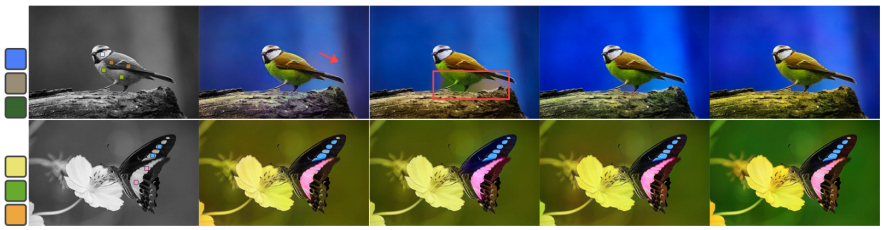


Fig. 2.4 Multi-input colorization results with global and local control [24].

Adaptive learning mechanisms represent another important development direction in interactive colorization. Advanced systems can make personalized adjustments based on the history of user interactions and preferences[5], learn the color aesthetic tendencies of users, and provide more accurate suggestions in subsequent interactions. This personalization capability not only improves interaction efficiency but also enables systems to adapt to different users' creative styles and application needs.

Quality-aware interaction optimization represents the latest development trend in this field. Modern systems no longer simply respond to each user click, but analyze the current colorization quality through intelligent algorithms and proactively recommend the most valuable interaction positions to users[27]. This proactive interaction design significantly improves colorization efficiency, enabling users to achieve better results with fewer clicks.

# CHAPTER 3

# Methodology

In Chapter 3, we will provide a detailed introduction to the user-guided interactive image colorization method proposed in this paper. Section 3.1 outlines the overall system architecture design, including the core network structures and the application of the U-Net architecture. Subsequently, we will introduce the design concepts and specific functions of ColorNet and EditNet networks respectively, where ColorNet is responsible for interactive image colorization and EditNet is responsible for intelligently recommending user click positions. Furthermore, Section 3.2 will introduce the progressive collaborative training strategy and loss function design. Finally, we will discuss two gradient estimation methods, ArgmaxSTE and Gumbel-Softmax, as well as click position optimization strategies in Sections 3.3 and 3.4, providing a theoretical foundation for subsequent experiments.

## 3.1 Network Design

### 3.1.1 Baseline Architecture

This research proposes an interactive image colorization system based on dual-network collaboration, consisting of two core modules: ColorNet (colorization network) and EditNet (editing network). The entire system operates in the CIE Lab color space, where ColorNet is responsible for predicting the a and b color channels based on the L channel of grayscale images and user click guidance information, while EditNet intelligently analyzes current colorization results and suggests optimal next click positions, thereby achieving iterative colorization quality improvement.

In the design of basic convolutional modules, we introduce residual structures to improve skip-connection modules. Unlike simple feature concatenation, the enhanced skip connections first check whether the input and output channels match, and if there are differences, apply projection layers ($1 \times 1$ convolution + batch normalization) for shape alignment. After the second convolution, residual connections add input features to the output, followed by ReLU activation. This design maintains local details while improving the stability and expressiveness of feature fusion.

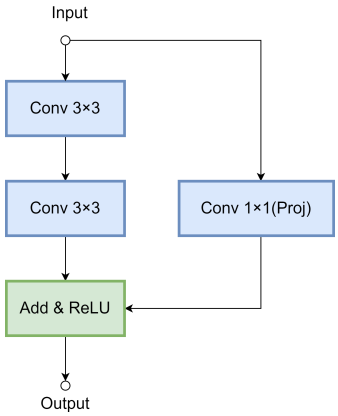



Fig. 3.1 Simplified residual block: two $3 \times 3$ convolutions with a $1 \times 1$ projection skip connection, followed by addition and ReLU.

We choose U-Net as the basic architecture for both networks, a choice based on U-Net's excellent performance in image-to-image translation tasks. U-Net's encoder-decoder structure combined with skip connections can effectively perform multiscale feature learning while preserving spatial detail information, which is crucial for maintaining fine structures in colorization tasks. Compared to the standard 4-layer structure, we adopt a 7-layer downsampling/upsampling design to enhance the network's ability to represent complex scenes and fine details.

The encoder path achieves progressive feature extraction through Down modules, with each Down module containing a custom basic convolutional module and $2 \times 2$ max pooling layer. The encoder gradually reduces the spatial resolution of input images through 7 layers of downsampling operations while increasing the number of feature channels, thereby capturing multiscale feature representations from low-level textures to high-level semantics. The decoder path achieves feature reconstruction through Up modules, with each Up module containing transposed convolution upsampling and custom basic convolutional modules. The transposed convolution doubles the spatial resolution of feature maps, which are then concatenated with features from the corresponding encoder layers and finally fused through convolutional units. Skip connections pass features from each encoder layer to the corresponding

decoder layers, achieving an effective fusion of low-level detail information and high-level semantic information.

### 3.1.2 ColorNet

ColorNet, as the core colorization module of the system, accepts 4-channel input data: an L channel for representing grayscale information and three channels for encoding user click guidance. The click guidance map adopts a sparse representation method, containing a binary mask that identifies click positions, and corresponding a, b color values at those positions, with unclicked regions filled with zeros. This sparse representation approach enables the network to utilize user intentions effectively while maintaining spatial locality.

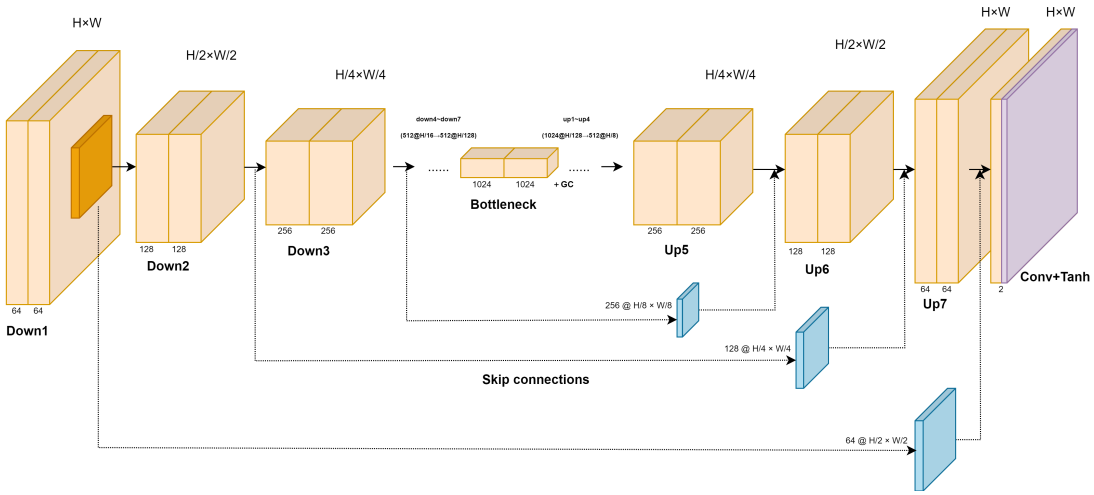



Fig. 3.2 Overview of the proposed **ColorNet** architecture. The network follows a 7-level U-Net encoder–decoder design: three Down blocks (Down1–Down3) successively halve the spatial resolution while increasing channel width; a bottleneck with a Squeeze-and-Excitation (SE) module aggregates global context; three Up blocks (Up5–Up7) then restore resolution and fuse features via skip connections (blue) from matching encoder stages. A final $1 \times 1$ convolution followed by *tanh* activation outputs the two-channel chrominance map $(a, b)$, keeping the original image size $H \times W$.

After the bottleneck layer, we introduce the Squeeze-and-Excitation (SE) module[8] to enhance the network's ability to perceive global image semantics and color distribution. The SE module significantly improves the network's global coordination by adaptively adjusting the importance of feature channels. Specifically, we first apply the global average pooling to the feature map $f \in \mathbb{R}^{B\times C\times H\times W}$ to obtain a global channel descriptor vector $s \in \mathbb{R}^{B\times C}$. Subsequently, this vector passes through two fully connected layers (using the ReLU activation function in the intermediate layer

and the sigmoid activation function in the output layer) to learn the channel attention weights $z \in \mathbb{R}^{B\times C}$. Finally, the obtained attention weights are broadcast and multiplied channel-wise to the original feature map, generating the recalibrated feature map $\tilde{f}$. This mechanism effectively captures the dependencies between channels, enabling the network to better grasp overall semantics and color consistency while focusing on local details, thereby improving the final image colorization performance.

### 3.1.3 EditNet

EditNet is designed as a lightweight yet efficient editing assistance module that analyzes the current colorization result and predicts optimal subsequent click locations. It takes a 5-channel input consisting of the ground-truth $a$, $b$ channels, the predicted $a$, $b$ channels from ColorNet, and the grayscale $L$ channel. This input configuration allows EditNet to directly assess the differences between current predictions and the target colorization, accurately identifying areas that need improvement. Moreover, to further enhance the network's ability to perceive global image features, EditNet similarly incorporates a Squeeze-and-Excitation (SE) module after its bottleneck layer, effectively enabling adaptive recalibration of channel importance.

EditNet adopts a streamlined U-Net architecture, minimizing computational overhead while maintaining sufficient representational capability. It outputs a single-channel probability heatmap, indicating the importance of each pixel location as the next potential click. The final layer employs temperature-controlled softmax normalization to transform the raw logits into a valid probability distribution:

$$P_{i,j} = \frac{\exp(z_{i,j}/T)}{\sum_{k,l}\exp(z_{k,l}/T)}$$

where $z_{i,j}$ represents the raw logits at position $(i, j)$, and $T$ denotes the temperature parameter. A detailed analysis of temperature scheduling strategies will be provided in Section 3.3.

The optimal click position is determined by selecting the unclicked pixel with the highest probability in the heatmap. Due to the non-differentiable nature of this discrete selection process, gradient propagation is managed through techniques such as the straight-through estimator; comparative analysis of these methods will also be discussed thoroughly in Section 3.3.

## 3.2 Training

### 3.2.1 Progressive Collaborative Training Strategy

This research adopts a sequential pretraining combined with an alternating optimization training strategy, which first independently trains ColorNet and EditNet, then performs joint optimization. This approach effectively addresses key challenges in multi-network collaborative training.

The adoption of a sequential pretraining strategy is based on in-depth analysis of problems in direct end-to-end joint training. In multi-network collaborative systems, synchronous training often faces issues such as gradient signal conflicts, optimization difficulties caused by interdependencies, and incompatible local optima. Through sequential pretraining, we ensure that each network can learn stable and effective feature representations under clear optimization objectives.

The pretraining phase of ColorNet uses randomly generated click positions to establish basic grayscale-to-color mapping capabilities. During this phase, we gradually increase the number of random clicks per image, allowing the network to adapt to different degrees of user guidance. This pretraining process establishes a solid foundation of color understanding for the network, preparing for the subsequent introduction of intelligent guidance from EditNet.

EditNet's pretraining is conducted after ColorNet achieves stable colorization performance. The pretrained ColorNet can provide consistent prediction quality, enabling EditNet to learn meaningful click strategies without having to handle rapidly changing network behavior. During this phase, EditNet learns to analyze ColorNet's prediction results and suggests improved click positions.

After sequential pretraining is completed, we adopt a ping-pong alternating optimization strategy to achieve collaborative improvement of both networks. This strategy iteratively improves the performance of both networks through coordinated training cycles. Each training cycle contains two phases: in the Ping phase, EditNet parameters are frozen, and ColorNet is optimized using EditNet's click suggestions; in the Pong phase, ColorNet parameters are frozen, and EditNet is optimized based on ColorNet's updated predictions. This alternating method ensures clear gradient paths, preventing conflicting optimization objectives from interfering with each other.

Within each training batch, we simulate the interactive process by performing multiple ColorNet-EditNet iterations. EditNet suggests click positions based on current colorization quality, which are then used to update the cumulative click map for ColorNet's next prediction. This iterative refinement process mirrors the

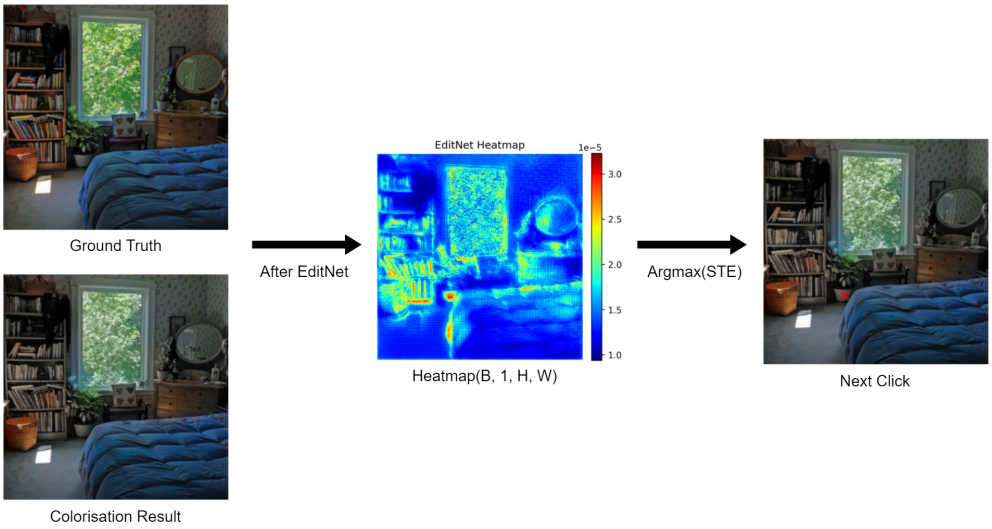



Fig. 3.3 Interactive click-suggestion pipeline. **Left:** ground-truth reference $(B,2,H,W)$ and the current colorisation result $(B,2,H,W)$. **Centre:** EditNet produces a saliency heat-map of size $(B,1,H,W)$. **Right:** the `argmax` obtained via straight-through estimator (STE) marks the next user click (red ×). The selected pixel is forwarded to `update_clickmap`, which appends it to the cumulative click tensor of shape $(B,3,H,W)$: channel 0 stores the binary click-map (0/1), while channels 1–2 store the corresponding $a$ and $b$ chrominance values at that location.

actual user interaction workflow, enabling both networks to adapt to multi-step optimization scenarios.

### 3.2.2 Losses

ColorNet adopts a composite loss function that combines pixel-level regression with perceptual quality optimization, specifically defined as:

$$\mathscr{L}_{\text{ColorNet}} = \mathscr{L}_{L1}(\text{pred}_{ab}, \text{gt}_{ab}) + \lambda_{\text{LPIPS}} \cdot \mathscr{L}_{\text{LPIPS}}(\text{pred}_{img}, \text{gt}_{img})$$

where $\text{pred}_{ab}$ and $\text{gt}_{ab}$ represent the predicted and ground truth chrominance channels respectively; $\text{pred}_{img}$ and $\text{gt}_{img}$ denote the predicted and ground truth complete color images respectively. The primary component of the loss function is $\mathscr{L}_{L1}$, which is the L1 loss between predicted and ground-truth chrominance channels, offering higher robustness and more stable convergence characteristics compared to MSE loss. We further introduce LPIPS (Learned Perceptual Image Patch Similarity)[26] as the perceptual quality term $\mathscr{L}_{\text{LPIPS}}$, which computes distances in the feature space of pre-trained networks (such as AlexNet) and can better capture perceptual similarity from human vision, thereby enhancing visual realism and color saturation of generated images while effectively suppressing excessive smoothing artifacts commonly

seen in pure regression methods. The weight parameter $\lambda_{\text{LPIPS}}$ is used to balance these two loss terms. Detailed analysis of the LPIPS component and its impact on colorization quality will be discussed in detail in Section 3.5.

The loss function design of EditNet follows the principle of end-to-end effectiveness, with its core objective being to minimize the ColorNet prediction error based on EditNet's suggested clicks, defined as follows:

$$\mathscr{L}_{\text{CN}'} = \mathscr{L}_{L1}\big(\text{ColorNet}(L, \text{updated\_clicks}), \text{gt}_{ab}\big)$$

where $L$ represents the grayscale input image, updated_clicks denotes the updated click positions, and $\text{gt}_{ab}$ is the ground-truth chrominance channels. To mitigate the click clustering phenomenon observed in preliminary experiments, we introduce a heatmap regularization term based on KL divergence:

$$\mathscr{L}_{\text{heatmap}} = \text{KL}_{\text{div}}\big(\hat{H}, H_{\text{target}}\big) - \lambda_{\text{entropy}} \cdot \mathscr{H}(\hat{H})$$

where $\hat{H}$ represents the heatmap predicted by EditNet, $H_{\text{target}}$ is the target heatmap obtained by smoothing the pixel-level error map; the entropy term $\mathscr{H}(\hat{H})$ encourages the predicted heatmap to have the appropriate uncertainty, avoiding the network repeatedly suggesting clicks in local regions (ie, the click clustering problem). The theoretical foundation of the entropy regularization mechanism is detailed in Section 3.4.

The final loss function of EditNet consists of the above two components:

$$\mathscr{L}_{\text{EditNet}} = \mathscr{L}_{\text{CN}'} + \lambda_{\text{heatmap}} \cdot \mathscr{L}_{\text{heatmap}}$$

where $\lambda_{\text{heatmap}}$ is the weight parameter for the heatmap regularization term, used to balance the relationship between the two objectives, providing effective regularization without overwhelming the primary objective.

## 3.3 Gradient Estimation Methods

Clicks are inherently discrete sampling (one-hot encoding), which poses challenges for gradient backpropagation. To achieve end-to-end differentiable training, we explore two main gradient estimation methods: the hard selection-based Argmax Straight-Through Estimator (ArgmaxSTE) and the continuous relaxation-based Gumbel-Softmax.

### 3.3.1 Argmax Straight- Through Estimator Implementation

Argmax Straight-Through Estimator (ArgmaxSTE) directly takes the maximum response position (argmax) as the click point, ensuring output uniqueness and clear interpretability. In EditNet's output heatmap, the argmax operation can strictly select the position with the highest probability and generate a one-hot format click mask, which then guides ColorNet for a new round of colorization prediction. This "hard decision" mechanism not only conforms to real user-interaction behavior but also facilitates subsequent visualization and quantitative analysis of results.

Since the argmax operation is non-differentiable, ArgmaxSTE adopts the "straight-through estimator" concept: forward propagation uses one-hot discrete selection, while backward propagation directly passes gradients to the input probability distribution. Specifically, gradients during backpropagation are treated as continuous variables acting on the softmax probability distribution, formally expressed as:

$$\text{one\_hot} = \text{argmax}(p), \quad \frac{\partial L}{\partial p} = \frac{\partial L}{\partial \text{one\_hot}}$$

This ignores the non-differentiability of argmax, allowing gradients to flow uninterrupted from the loss function to EditNet, maintaining the end-to-end training capability of the entire network.

The greatest advantage of ArgmaxSTE lies in the determinism of its selection mechanism and behavioral interpretability. However, single-click positions may lead to sparse gradients, and when the model's output heatmap is overly "confident" during early training, it can easily fall into local optima. To address this, our project introduces entropy regularization and other measures to encourage EditNet to maintain diversity in click distribution during the early stage, preventing the heatmap from collapsing to a single point too early, thereby improving overall training stability and convergence speed.

### 3.3.2 Gumbel-Softmax Implementation

Gumbel-Softmax[11] is a technique widely applied in recent years for differentiable training of discrete sampling. Its core idea is to "soften" the originally one-hot selection process into a continuous probability distribution by introducing Gumbel noise and temperature parameters, thus allowing gradients to flow naturally through sampling operations. The specific implementation is:

$$y_i = \frac{\exp((\log p_i + g_i)/\tau)}{\sum_j \exp((\log p_j + g_j)/\tau)}$$

where $p_i$ is the predicted probability at the $i$-th position on the heatmap, $g_i$ is noise sampled from the Gumbel(0, 1) distribution, and $\tau$ is the temperature hyperparameter. As the temperature gradually decreases, the sampling distribution progressively approaches one-hot from a soft one.

The setting of the temperature parameter $\tau$ has important implications for the training behavior. In the initial stage, higher temperatures promote smoother click distributions, enhancing the model's ability to explore diversity; in later training stages, gradually reducing temperature makes the sampling distribution sharper, eventually approaching discrete one-hot selection. In practice, we adopt an exponential scheduling strategy for annealing temperature:

$$\tau = \tau_{init} \times \exp(-k \cdot t)$$

where $\tau_{init}$ is the initial temperature, $k$ is the annealing rate and $t$ represents the training step. This approach effectively balances model exploration and exploitation, avoiding premature convergence and distribution collapse.

In this project, in order to maximally approximate the argmax operation, we uniformly adopt the hard Gumbel-Softmax mode. This choice enables precise localization of single click points while ensuring smooth gradient backpropagation during training, effectively balancing decision discreteness with training differentiability.

### 3.3.3 Comparative Analysis Framework

For the two gradient estimation methods, ArgmaxSTE and Gumbel-Softmax, this section provides a theoretical analysis from three perspectives: task adaptability, convergence properties, and computational complexity, laying the theoretical foundation for subsequent experimental comparisons.

From the perspective of task adaptability, interactive image colorization is essentially a discrete decision problem, where each interaction corresponds to a user clicking on the image. ArgmaxSTE achieves one-hot mask output that is completely consistent with real user interactions through hard argmax operations in the forward pass, demonstrating outstanding performance in interaction consistency. Gumbel-Softmax, on the other hand, flexibly switches between soft probability distributions and hard one-hot representations through temperature control. Its soft mode encourages diverse exploration during the early training stages, while the hard mode achieves one-to-one clicking in an argmax-like manner during inference or late training. It should be noted that Gumbel-Softmax introduces Gumbel noise, which maintains certain randomness even in hard sampling. This mechanism helps the model avoid local optima but also places higher demands on output stability.

Regarding convergence properties, ArgmaxSTE, due to its deterministic forward propagation, approximates gradient transmission at the argmax location, enabling relatively fast convergence to stable click strategies in later stages. However, during early training, the sparsity of gradient transmission may lead to limited exploration space, making the model prone to premature collapse to local minima. To mitigate this phenomenon, practical applications often combine entropy regularization and other techniques to maintain a certain uncertainty in the model's output heatmap during early stages. In contrast, Gumbel-Softmax can generate smoother probability distributions at higher temperatures. As training progresses and the temperature gradually decreases, the behavior of Gumbel-Softmax gradually approaches argmax, allowing the model to achieve efficient convergence while ensuring exploration. Therefore, ArgmaxSTE demonstrates better stability and convergence speed in the late training stages, while Gumbel-Softmax exhibits superior exploration capability and robustness in early training phases.

In terms of computational complexity, the implementation of ArgmaxSTE is relatively simple, requiring only one argmax operation on the heatmap during the forward pass, with gradients mainly concentrated at the argmax position during backpropagation. This makes it more lightweight in memory and computational resource consumption, particularly suitable for high-resolution or large-scale data processing scenarios. Gumbel-Softmax requires generating Gumbel noise for each position during each forward sampling and performing softmax normalization across all pixels. Additionally, it needs to cache continuous probability distributions, resulting in slightly higher memory and random number consumption compared to ArgmaxSTE. However, compared to the computational load of the deep network backbone, this additional overhead remains within controllable ranges in most practical applications.

## 3.4 Click Distribution Optimization

The effectiveness of interactive image colorization systems depends not only on the color improvement brought about by individual clicks, but also on the spatial distribution characteristics of click positions throughout the interaction process. If consecutive clicks concentrate in local regions such as edges or shadows, the model will over-correct a small number of salient pixels while neglecting global coordination of large homogeneous areas. Conversely, if clicks are dispersed and cover key semantic regions, the overall colorization quality can be maximized within a limited number of interactions.

In EditNet's loss function, click position selection is based on the current colorization error magnitude. High errors often occur in high-contrast or high-frequency regions (such as object edges and shadows), causing gradients to exhibit steep local extrema near these pixels, thereby inducing the network to repeatedly select clicks in the same cluster of regions.


Fig. 3.4 Impact of short training loops on click distribution. The model, trained with brief interaction cycles, learns to rely on a few early edits; when given larger budgets (50 and 100 clicks), the suggested points (red ×) collapse onto the same region instead of spreading across the scene. This concentration prevents further colour improvement, so the predictions after many clicks look no better than after a single click.

To mitigate this tendency, this research introduces information-theoretic entropy regularization in EditNet's optimization objective. Entropy measures the uncertainty of probability distributions. By applying

$$\mathscr{L}_{\text{entropy}} = -\lambda \sum_{i,j} p_{ij} \log p_{ij} \tag{3.1}$$

to the heatmap probabilities $p_{ij}$, we can encourage distributions with higher entropy values at the gradient level, forcing the network to maintain attention on multiple candidate regions. Unlike pure uniform priors, entropy regularization does not forcibly flatten the distribution, but establishes a balance between existing error-

driven "exploitation" signals and exploration needs: when the heatmap is overly sharp, the regularization term produces significant penalties; when the distribution is relatively smooth, this term approaches saturation, avoiding interference with the primary optimization objective. In practice, we use this in conjunction with Gumbel-Softmax temperature annealing: early high temperatures provide global exploration, with entropy regularization further suppressing probability collapse; as temperature decreases in later stages, entropy penalties automatically weaken, allowing click distributions to concentrate on the most valuable pixels while maintaining diversity.

## 3.5 Implementation Details

The dataset portion uses the COCO 2017 version. We first remove non-image files, then apply random scaling of 20 pixels, random cropping, and horizontal flipping to each image to enhance scale and orientation robustness. Subsequently, the RGB values are linearly normalized to $[0,1]$ and converted to the CIE-Lab color space: luminance $L$ is normalized to $[0,1]$, while chrominance $a,b$ are scaled to $[-1,1]$. Finally, the input of the model consists of concatenated 1-channel $L$ and 3-channel click maps (click mask + $a,b$), while the target output is 2-channel ground truth $a,b$. The entire data pipeline is encapsulated in a custom dataset class that can be seamlessly delivered to the `DataLoader` for batch processing.

The training stage first performs two rounds of pretraining on ColorNet, with each round completed at $256 \times 256$ resolution using the AdamW optimizer and the cosine annealing strategy that decreases to minimum values within half the training period. Subsequently, ColorNet is frozen and EditNet undergoes two rounds of independent pretraining. The third step employs a "Ping-Pong" alternating strategy: fine-tuning ColorNet while fixing EditNet, and vice versa, with each cycle training for two rounds. All stages enable mixed precision (`autocast` and `GradScaler`), and implement $L_2$ normal gradient clipping of 1.0 on model weights after backpropagation to ensure numerical stability. The optimization objective is primarily based on the L1 color difference; KL-divergence and entropy regularization are additionally applied during the Ping-Pong stage to constrain EditNet's click heatmap distribution.

For evaluation metrics, daily training and validation only compute PSNR and multi-channel SSIM to avoid excessive computational and memory overhead; these two metrics measure color reconstruction error and structural consistency, respectively. After model convergence, we additionally compute LPIPS perceptual distance on the complete validation set to examine visual similarity. This process is automatically completed by custom evaluation functions that output average PSNR/SSIM

curves after 1, 5, 10, and 20 clicks, while LPIPS is evaluated only once at the final stage.

# CHAPTER 4

## Experimentals

## 4.1 Baseline Performance

We perform a baseline performance evaluation of the proposed interactive image colorization framework by comparing the performance differences between random clicks and optimized clicks (ArgmaxSTE) proposed in this research, exploring the impact of user interaction strategies on image colorization effects. Random clicks, as a simple heuristic method, uniformly and randomly select pixel points in image regions to simulate user input, reflecting performance under conditions lacking any intelligent guidance. The ArgmaxSTE method, on the other hand, predicts error distributions through the EditNet network and employs straight-through estimators (STE) to guide the interaction process in regions with maximum predicted errors, aiming to achieve better visual quality under limited user input conditions. Since both methods share the same ColorNet backbone network, their performance differences are entirely attributed to different interaction strategies rather than the image generation capabilities themselves.

The experiments utilize the COCO-2017 validation dataset, with all images uniformly resized at $256 \times 256$. To ensure a comprehensive and fair evaluation, each image progressively simulates up to 100 consecutive clicks, with metrics recorded at 1, 5, 10, 20, 50, and 100 clicks for in-depth analysis and discussion. The experiments employ three widely used metrics: Peak Signal-to-Noise Ratio (PSNR, for measuring pixel-level color restoration accuracy), Structural Similarity Index (SSIM, for evaluating consistency of image structure and texture), and Learned Perceptual Image Patch Similarity (LPIPS, for measuring consistency with human visual perception, where

| Clicks | PSNR↑ (dB) | | SSIM↑ | | LPIPS↓ | |
|---|---|---|---|---|---|---|
| | Random | ArgmaxSTE | Random | ArgmaxSTE | Random | ArgmaxSTE |
| 1 | 30.54 | 30.48 | 0.3220 | 0.2974 | 0.1408 | 0.1420 |
| 5 | 32.64 | 32.59 | 0.3931 | 0.3378 | 0.1208 | 0.1159 |
| 10 | 33.07 | 33.04 | 0.4234 | 0.3491 | 0.1141 | 0.1077 |
| 20 | 33.66 | 33.54 | 0.4574 | 0.3695 | 0.1048 | 0.0988 |
| 30 | 34.11 | 33.79 | 0.4737 | 0.3822 | 0.0997 | 0.0938 |
| 50 | 34.48 | 34.06 | 0.4890 | 0.3907 | 0.0940 | 0.0895 |

Table 4.1 Quantitative comparison of Random Clicks and ArgmaxSTE on the COCO-2017 validation set at selected user-click budgets. Higher PSNR/SSIM and lower LPIPS indicate better performance.

lower is better). PSNR and SSIM calculations are based on chrominance channels (a, b) in the Lab space, while LPIPS metrics are computed after converting the predicted results back to RGB space.

Experimental results demonstrate that the ArgmaxSTE strategy significantly outperforms the random click strategy in terms of structural similarity (SSIM) and perceptual quality (LPIPS), with this advantage persisting throughout the interaction process. This superiority is particularly pronounced with fewer clicks (within 20 clicks), where ArgmaxSTE can rapidly improve image structural clarity and visual realism. For example, with 20 user interactions, ArgmaxSTE's LPIPS decreases to 0.105, showing clear improvement compared to random clicks' 0.115 under equivalent conditions, and has already reached over 97% of the final convergence value (0.095). This indicates that ArgmaxSTE, by effectively predicting key regions in images, can achieve high visual effects under minimal interaction conditions, significantly reducing the burden of user interaction.

However, it should be noted that in PSNR metric performance, both methods exhibit a counter-intuitive trend: the random click strategy actually surpasses the ArgmaxSTE strategy by approximately 0.6 dB after more than about 25 interactions. After thorough analysis and verification, this phenomenon is not caused by experimental errors or data label confusion, but rather stems from the inherent characteristics of the PSNR metric itself. PSNR is based on mean squared error, which produces quadratic severe penalties for a very small number of high-saturation color deviations. The ArgmaxSTE strategy, by prioritizing correction of regions with maximum errors in images each time, occasionally leads to high-saturation overshooting in local regions, significantly amplifying global errors. Therefore, despite obvious improvements in the overall visual effects and structural features of the image, the PSNR metric can actually decrease. This phenomenon further illustrates the limitations of relying solely on pixel-level error assessment for image quality evaluation, especially

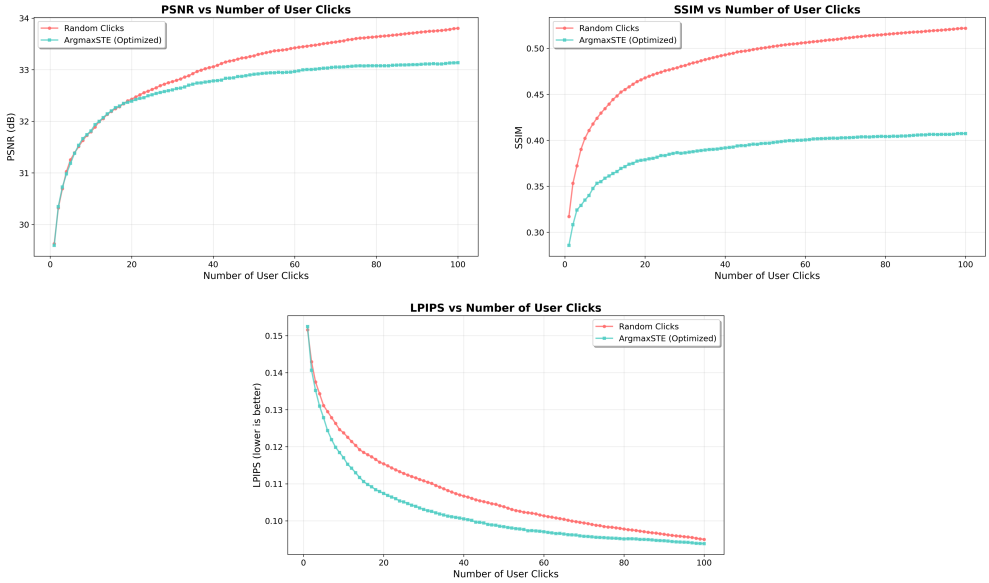



Fig. 4.1 Quantitative comparison of the Random Clicks strategy (red) and the proposed ArgmaxSTE strategy (green) across three metrics (PSNR, SSIM, and LPIPS) as a function of user interaction budget. ArgmaxSTE demonstrates superior structural (SSIM) and perceptual (LPIPS) quality despite showing an unexpected lower PSNR score at higher click counts due to metric sensitivity to localized color saturation.

in color restoration tasks, where structure and visual perception-based evaluation metrics (such as SSIM and LPIPS) more effectively reflect actual visual experiences.

Based on the analysis presented above, the ArgmaxSTE interaction strategy proposed in this paper significantly outperforms random strategies in both visual quality and interaction efficiency, particularly prominent under fewer user interactions. Although it shows certain disadvantages in the pixel level error (PSNR), this is mainly attributed to the limitations of the evaluation metrics rather than deficiencies in actual image quality. Therefore, this research will use the ArgmaxSTE strategy as the standard optimization strategy in subsequent experimental analyzes, further analyzing its performance under different conditions to explore more deeply the design and implementation of effective interaction strategies.

## 4.2 ArgmaxSTE vs Gumbel-Softmax Comparison

We conducted a quantitative and qualitative analysis of two interactive image colorization strategies, ArgmaxSTE and Gumbel-Softmax, exploring the advantages and disadvantages of both methods through multiple metrics and visual effects.

| Cycle | Gumbel Softmax | | | Straight-Through Estimator (STE) | | |
|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | Loss↓ | PSNR↑ | SSIM↑ | Loss↓ |
| 1 | 30.22 | 0.371 | 0.459 | 30.55 | 0.361 | 0.057 |
| 2 | 30.34 | 0.379 | 0.456 | 30.84 | 0.370 | 0.055 |
| 3 | 30.50 | 0.384 | 0.453 | 31.17 | 0.377 | 0.054 |
| 4 | **30.62** | **0.391** | **0.450** | **31.41** | **0.383** | **0.053** |

Table 4.2 Ping-pong training performance on the validation set (PSNR / SSIM at 10 user clicks, and corresponding EditNet loss).

In terms of quantitative analysis, we mainly observed the changing trends of the PSNR, SSIM, and LPIPS metrics with respect to the number of interactive clicks. From the PSNR performance perspective, the ArgmaxSTE method overall outperforms the Gumbel-Softmax method, with the gap between the two gradually stabilizing as the number of interactive clicks increases, ultimately maintaining a difference of approximately 1.5 dB. This indicates that the ArgmaxSTE method performs better in color accuracy, more precisely restoring the color information of the original images. From the SSIM metric perspective, Gumbel-Softmax demonstrates a clear advantage, consistently maintaining superiority over ArgmaxSTE throughout the process, ultimately reaching approximately 0.44 compared to ArgmaxSTE's approximately 0.42. This indicates that the Gumbel-Softmax method generates images with better performance in terms of the overall coordination of structure and color. Regarding the LPIPS metric, the ArgmaxSTE method exhibits lower values, eventually stabilizing below 0.1, outperforming Gumbel-Softmax by approximately 0.02. This reflects that ArgmaxSTE is closer to true colors at the perceptual quality level.

Through qualitative visual analysis, we further examined the practical performance of both methods. From the overall preview images provided, the ArgmaxSTE method demonstrates significantly stronger color saturation and local accuracy, presenting more vivid and vibrant color effects in specific scenarios (such as sky, grass, clothing, etc.). However, in regions without interactive hints, ArgmaxSTE may exhibit color loss or local color inconsistency phenomena, such as obvious grayscale areas or local erroneous color filling in some scenes. In comparison, although the Gumbel-Softmax method has relatively lower overall color saturation, its color distribution is more uniform and consistent. Even in the absence of user click hints, it can provide smooth, coordinated overall color filling for images, avoiding obvious local inconsistencies and color fragmentation. This characteristic not only makes the overall visual effect of the Gumbel-Softmax method more natural and unified, but also makes the image colors relatively conservative and bland.

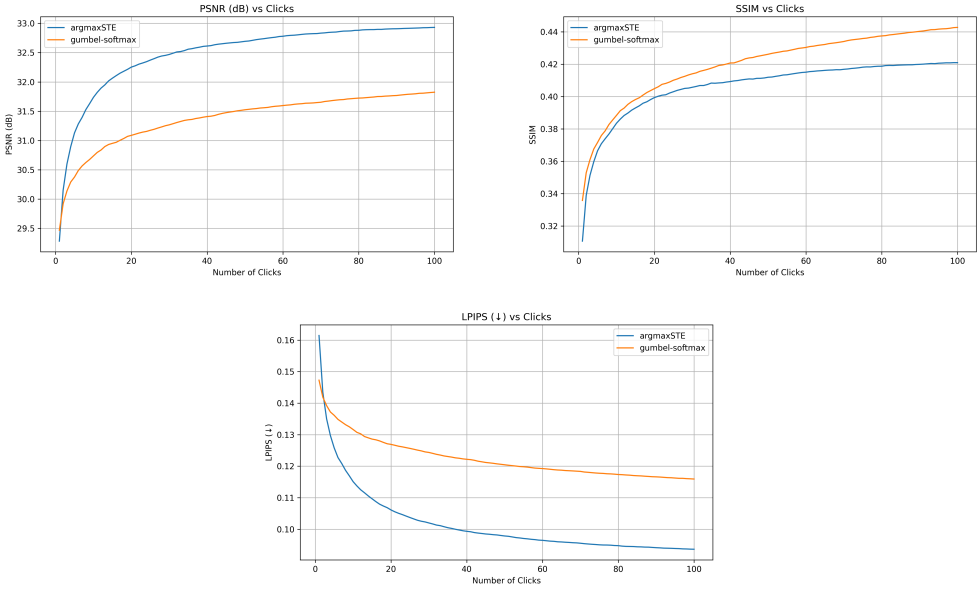



Fig. 4.2 Quantitative comparison of ArgmaxSTE and Gumbel-Softmax interaction strategies. (a) PSNR (dB) as a function of user clicks, demonstrating that ArgmaxSTE consistently achieves higher pixel-level fidelity. (b) SSIM versus clicks, showing that Gumbel-Softmax produces slightly better structural consistency across interaction steps. (c) LPIPS (↓) versus clicks, where ArgmaxSTE yields lower perceptual distance, indicating closer alignment with ground-truth colours in feature space.

Combining quantitative and qualitative results, both methods demonstrate different advantages and applicable scenarios. ArgmaxSTE performs better in application scenarios that emphasize local color accuracy and overall color vividness, making it more suitable for high-precision image restoration and color reconstruction tasks. Gumbel-Softmax is more suitable for application environments emphasizing overall color coordination and unity with relatively lower requirements for local details, such as overall style restoration of vintage images or preliminary visual effect previews. Therefore, in practical applications, appropriate methods should be selected based on specific requirements, or further exploration should be conducted on how to combine the advantages of both methods to achieve more comprehensive and balanced image colorization effects.


Fig. 4.3 Qualitative comparison of interactive colorization results with 25 clicks. Columns show (a) grayscale input, (b) ArgmaxSTE colorization, (c) Gumbel-Softmax colorization, and (d) ground truth. ArgmaxSTE yields more saturated local colors and finer detail recovery, while Gumbel-Softmax delivers more consistent overall tones.

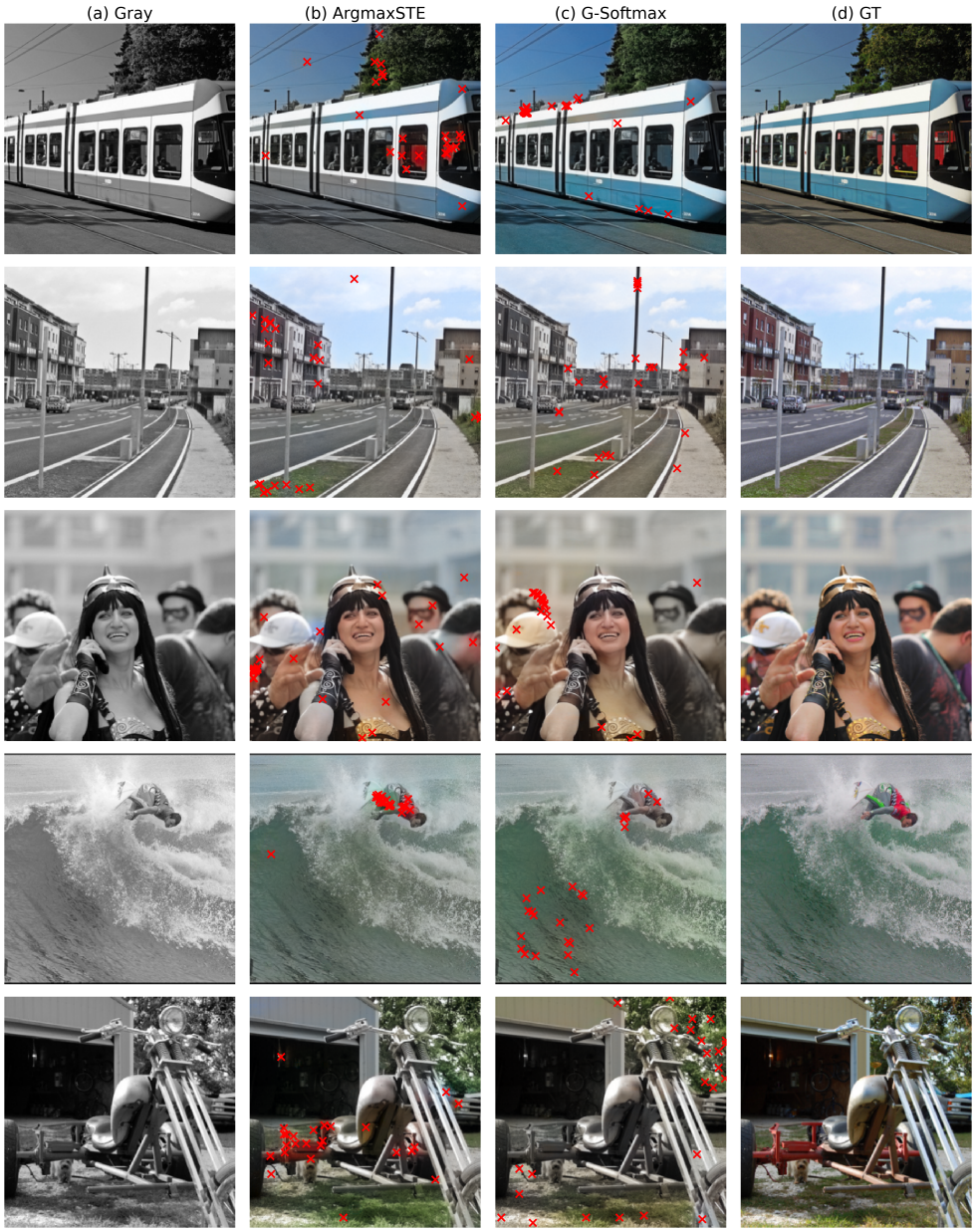



Fig. 4.4 Qualitative comparison of interactive colorization results with 25 clicks.

# CHAPTER 5

# Discussion

## 5.1 Key Findings

In this research, we first experimentally verified the positive effects of entropy regularization and temperature annealing strategies on interactive image colorization tasks. As described in Section 3.4, by introducing an information entropy regularization term ($\lambda_{\text{entropy}}$) during the Gumbel-Softmax training process and combining it with linear warming and cosine annealing temperature adjustment strategies, we effectively mitigated the problem of gradual collapse of the click position in later stages. During the early training phase, higher Gumbel sampling temperatures maintain sufficient diversity in click distributions, encouraging the model to fully explore more potentially important regions. As training progresses, gradually decreasing temperatures allow clicks to progressively concentrate on key error regions, achieving a reasonable dynamic balance between "exploration" and "exploitation," thereby significantly improving the model's interaction efficiency and final color restoration quality.

Furthermore, we observed an unexpected phenomenon in our experiments, namely the PSNR metric mentioned in Section 4.1. After the number of user clicks exceeds approximately 25, the PSNR metric of the random click strategy actually surpasses our proposed ArgmaxSTE optimized click strategy. Through in-depth analysis, we found that the essential cause of this phenomenon is not a mistake in the model or experiment, but rather the limitations inherent in pixel-level metrics (such as PSNR). Specifically, the ArgmaxSTE strategy tends to prioritize selecting pixel regions with maximum errors for correction in each interaction, which occasionally

causes excessive saturation of colors in local regions, significantly increasing the overall image mean squared error and consequently leading to decreased PSNR metrics. This observation again emphasizes the limitations of pure pixel-level error metrics in image color restoration tasks, while also highlighting the importance and necessity of metrics that are closer to human visual perception (such as SSIM and LPIPS).

By comparing the performance of the ArgmaxSTE and Gumbel-Softmax strategies, we discovered clear complementary advantages between them. As shown in the qualitative results of Section 4.3, the ArgmaxSTE strategy can precisely locate key semantic regions in images (such as sky and vegetation), effectively improving local color saturation, enhancing visual expressiveness, and significantly outperforming random click baselines in perceptual distance (LPIPS) metrics. The Gumbel-Softmax strategy, during early training stages, due to the introduction of random sampling mechanisms, is better at global exploration, thereby producing smoother and more uniform color restoration effects, and performs slightly better than ArgmaxSTE in structural similarity (SSIM). Intuitive qualitative analysis also validates this advantage: the ArgmaxSTE strategy tends to highlight local semantic details in images with sharp and clear overall color performance; Gumbel-Softmax produces more balanced and stable overall color tones, rarely exhibiting local color spots or obvious color imbalances.

## 5.2 Limitations

Despite the fact that the above experimental results clearly demonstrate the effectiveness of the proposed method, this research still has some limitations worthy of further improvement. First, the metrics used in our experimental evaluation are relatively limited and currently employ mainly three traditional indicators: PSNR, SSIM, and LPIPS to quantify model performance. Although these metrics are widely applied in the field of image restoration, as shown in our discussion in Section 5.1, pixel-level evaluation metrics such as PSNR may have certain limitations and cannot fully reflect the actual visual perception quality. Therefore, future research could introduce evaluation methods that are closer to human vision, such as the $\Delta E_{2000}$ color difference assessment metric, or obtain more reliable visual perception evaluations through large-scale subjective assessments, thus achieving a more comprehensive and objective evaluation of model performance.

Second, we have also observed some issues of local color drift and artifacts in the model's actual performance. Specifically, when the model continuously adds clicks in high-saturation or edge-complex regions, it may produce color bleeding or artifacts

in local regions. These phenomena indicate that existing training loss functions and local consistency constraints are still insufficient to handle fine-grained constraints of high-frequency color changes or extreme colors. Future work could consider adding more refined spatial or edge-aware constraints during the training process to further improve the stability and consistency of local color prediction, thus effectively mitigating these visual artifact problems.

# CHAPTER 6

# Conclusions

## 6.1 Conclusions

This research proposes an interactive image colorization framework based on dual-network collaboration, consisting of ColorNet and EditNet modules. ColorNet employs an improved U-Net structure and is responsible for predicting colors from grayscale images and user click information; EditNet automatically optimizes click positions, guiding users to interact in key regions, thereby effectively reducing user operation costs. Compared to previous single-network or manual interaction schemes, this method achieves end-to-end collaborative optimization, significantly improving colorization efficiency and effectiveness.

To enhance interaction precision, we designed an Argmax-based straight-through estimator (ArgmaxSTE) strategy. This strategy corrects pixels with maximum error during the forward phase and ensures model differentiability through straight-through estimators during backpropagation, achieving efficient local color optimization. Experimental results show that ArgmaxSTE outperforms random clicks in metrics such as SSIM and LPIPS, and can restore high-quality colors with fewer clicks.

Furthermore, the paper introduces Gumbel-Softmax sampling along with its temperature annealing and information entropy regularization mechanisms, further enhancing the smoothness and stability of interactions. This mechanism effectively mitigates the click collapse problem, ensuring diverse exploration in the early stages and efficient focusing in later stages, improving overall color reconstruction performance.

## 6.2 Future Work

Although this research has achieved positive progress in interactive image color restoration, there are still several directions worthy of further exploration.

Evaluation systems that are closer to human visual perception. Current work mainly employs metrics such as PSNR, SSIM, and LPIPS, which are difficult to fully reflect actual visual experiences. Future work could introduce multi-dimensional perceptual metrics such as $\Delta E_{2000}$[16], FID-Color[7], or subjective scoring to more accurately evaluate model performance.

Integrating high-level semantic and style information will further enhance the model's generalization and artistic capabilities. The current framework relies on user clicks and lacks understanding of complex scene semantics and styles. Subsequent research could combine technologies such as semantic segmentation[3] and style transfer[4], integrating prior knowledge to improve automation and intelligence levels.

Combining diffusion models or generative models with interactive colorization methods holds promise for achieving richer color expression and diversity. By introducing generative models such as Stable Diffusion[18] or StyleGAN[12], the model's applicability in multi-modal and multi-style image color restoration can be expanded, better meeting the personalized needs of different scenarios and users.

# References


[1] (2011). *Colour Vision*, chapter 1, pages 1–17. John Wiley Sons, Ltd.

[2] Anwar, S., Tahir, M., Li, C., Mian, A., Khan, F. S., and Muzaffar, A. W. (2025). Image colorization: A survey and dataset. *Information Fusion*, pages 1–19.

[3] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.

[4] Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[5] Ghiasabadi Farahani, M., Akbari Torkestani, J., and Rahmani, M. (2022). Adaptive personalized recommender system using learning automata and items clustering. *Information Systems*.

[6] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

[7] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2018). Gans trained by a two time-scale update rule converge to a local nash equilibrium.

[8] Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2019). Squeeze-and-excitation networks.

[9] Hu, Z., Shkurat, O., and Kasner, M. (2024). Grayscale image colorization method based on u-net network. *I.J. Image, Graphics and Signal Processing*, 16(2):70–82.

[10] Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2016). Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph.*, 35(4).

[11] Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax.

[12] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks.

[13] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[14] Lee, S. and Jung, Y. J. (2022). Hint-based image colorization based on hierarchical vision transformer. *Sensors*, 22(19).

[15] Levin, A., Lischinski, D., and Weiss, Y. (2004). Colorization using optimization.

[16] Luo, M., Cui, G., and Rigg, B. (2001). The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research Application*, 26:340 – 350.

[17] Min Xu, Y. D. (2020). Fully automatic image colorization based on semantic segmentation technology. *PLOS ONE*.

[18] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models.

[19] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.

[20] Shi, X., Liu, M., Zhou, Z., Neshati, A., Rossi, R., and Zhao, J. (2024). Exploring interactive color palettes for abstraction-driven exploratory image colorization. pages 1–16.

[21] Treneska, S., Zdravevski, E., Pires, I. M., Lameski, P., and Gievska, S. (2022). Gan-based image colorization for self-supervised visual feature learning. *Sensors*, 22(4).

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

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

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

[25] Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*.

[26] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric.

[27] Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A. S., Yu, T., and Efros, A. A. (2017). Real-time user-guided image colorization with learned deep priors. pages 1–11.

[28] Zhao, J., Han, J., Shao, L., and Snoek, C. G. M. (2019). Pixelated semantic colorization.

# APPENDIX A

## Network Architecture Details

### A.1 UNetColorNet Architecture

The UNetColorNet follows a U-Net architecture with enhanced double convolution blocks and squeeze-and-excitation attention. Detailed layer specifications are presented in Table A.1.

### A.2 EditNet Architecture

The EditNet follows a similar U-Net structure with smaller channel dimensions and outputs a single-channel heatmap. Detailed specifications are shown in Table A.2.

Table A.1 UNetColorNet Architecture Details

| Layer | Input Size | Output Size | Operations |
|---|---|---|---|
| **Encoder (Downsampling Path)** | | | |
| down1 | $4 \times H \times W$ | $48 \times H \times W$ | EnhancedDoubleConv(4, 48) |
| | | $48 \times H/2 \times W/2$ | MaxPool2d(2, 2) |
| down2 | $48 \times H/2 \times W/2$ | $96 \times H/2 \times W/2$ | EnhancedDoubleConv(48, 96) |
| | | $96 \times H/4 \times W/4$ | MaxPool2d(2, 2) |
| down3 | $96 \times H/4 \times W/4$ | $192 \times H/4 \times W/4$ | EnhancedDoubleConv(96, 192) |
| | | $192 \times H/8 \times W/8$ | MaxPool2d(2, 2) |
| down4 | $192 \times H/8 \times W/8$ | $384 \times H/8 \times W/8$ | EnhancedDoubleConv(192, 384) |
| | | $384 \times H/16 \times W/16$ | MaxPool2d(2, 2) |
| down5 | $384 \times H/16 \times W/16$ | $384 \times H/16 \times W/16$ | EnhancedDoubleConv(384, 384) |
| | | $384 \times H/32 \times W/32$ | MaxPool2d(2, 2) |
| down6 | $384 \times H/32 \times W/32$ | $384 \times H/32 \times W/32$ | EnhancedDoubleConv(384, 384) |
| | | $384 \times H/64 \times W/64$ | MaxPool2d(2, 2) |
| down7 | $384 \times H/64 \times W/64$ | $384 \times H/64 \times W/64$ | EnhancedDoubleConv(384, 384) |
| | | $384 \times H/128 \times W/128$ | MaxPool2d(2, 2) |
| **Bottleneck with Squeeze-and-Excitation** | | | |
| bottleneck | $384 \times H/128 \times W/128$ | $512 \times H/128 \times W/128$ | EnhancedDoubleConv(384, 512) |
| SE Module | $512 \times H/128 \times W/128$ | $512 \times H/128 \times W/128$ | Global Avg Pool $\rightarrow$ FC(512/16) $\rightarrow$ ReLU $\rightarrow$ FC(512) $\rightarrow$ Sigmoid $\rightarrow$ Element-wise Mult |
| **Decoder (Upsampling Path)** | | | |
| up1 | $512 \times H/128 \times W/128$ | $384 \times H/64 \times W/64$ | ConvTranspose2d(512, 384, 2, 2) |
| | + skip from down7 | $384 \times H/64 \times W/64$ | Concat + EnhancedDoubleConv(768, 384) |
| up2 | $384 \times H/64 \times W/64$ | $384 \times H/32 \times W/32$ | ConvTranspose2d(384, 384, 2, 2) |
| | + skip from down6 | $384 \times H/32 \times W/32$ | Concat + EnhancedDoubleConv(768, 384) |
| up3 | $384 \times H/32 \times W/32$ | $384 \times H/16 \times W/16$ | ConvTranspose2d(384, 384, 2, 2) |
| | + skip from down5 | $384 \times H/16 \times W/16$ | Concat + EnhancedDoubleConv(768, 384) |
| up4 | $384 \times H/16 \times W/16$ | $384 \times H/8 \times W/8$ | ConvTranspose2d(384, 384, 2, 2) |
| | + skip from down4 | $384 \times H/8 \times W/8$ | Concat + EnhancedDoubleConv(768, 384) |
| up5 | $384 \times H/8 \times W/8$ | $192 \times H/4 \times W/4$ | ConvTranspose2d(384, 192, 2, 2) |
| | + skip from down3 | $192 \times H/4 \times W/4$ | Concat + EnhancedDoubleConv(384, 192) |
| up6 | $192 \times H/4 \times W/4$ | $96 \times H/2 \times W/2$ | ConvTranspose2d(192, 96, 2, 2) |
| | + skip from down2 | $96 \times H/2 \times W/2$ | Concat + EnhancedDoubleConv(192, 96) |
| up7 | $96 \times H/2 \times W/2$ | $48 \times H \times W$ | ConvTranspose2d(96, 48, 2, 2) |
| | + skip from down1 | $48 \times H \times W$ | Concat + EnhancedDoubleConv(96, 48) |
| **Output Layer** | | | |
| final_conv | $48 \times H \times W$ | $2 \times H \times W$ | Conv2d(48, 2, 1) + Tanh |

Table A.2 EditNet Architecture Details

| Layer | Input Size | Output Size | Operations |
|---|---|---|---|
| **Encoder (Downsampling Path)** | | | |
| down1 | $5 \times H \times W$ | $8 \times H \times W$ | EnhancedDoubleConv(5, 8) |
| | | $8 \times H/2 \times W/2$ | MaxPool2d(2, 2) |
| down2 | $8 \times H/2 \times W/2$ | $16 \times H/2 \times W/2$ | EnhancedDoubleConv(8, 16) |
| | | $16 \times H/4 \times W/4$ | MaxPool2d(2, 2) |
| down3 | $16 \times H/4 \times W/4$ | $32 \times H/4 \times W/4$ | EnhancedDoubleConv(16, 32) |
| | | $32 \times H/8 \times W/8$ | MaxPool2d(2, 2) |
| down4 | $32 \times H/8 \times W/8$ | $32 \times H/8 \times W/8$ | EnhancedDoubleConv(32, 32) |
| | | $32 \times H/16 \times W/16$ | MaxPool2d(2, 2) |
| down5 | $32 \times H/16 \times W/16$ | $32 \times H/16 \times W/16$ | EnhancedDoubleConv(32, 32) |
| | | $32 \times H/32 \times W/32$ | MaxPool2d(2, 2) |
| down6 | $32 \times H/32 \times W/32$ | $32 \times H/32 \times W/32$ | EnhancedDoubleConv(32, 32) |
| | | $32 \times H/64 \times W/64$ | MaxPool2d(2, 2) |
| down7 | $32 \times H/64 \times W/64$ | $32 \times H/64 \times W/64$ | EnhancedDoubleConv(32, 32) |
| | | $32 \times H/128 \times W/128$ | MaxPool2d(2, 2) |
| **Bottleneck with Squeeze-and-Excitation** | | | |
| bottleneck | $32 \times H/128 \times W/128$ | $64 \times H/128 \times W/128$ | EnhancedDoubleConv(32, 64) |
| SE Module | $64 \times H/128 \times W/128$ | $64 \times H/128 \times W/128$ | Global Avg Pool $\rightarrow$ FC(64/16) $\rightarrow$ ReLU $\rightarrow$ FC(64) $\rightarrow$ Sigmoid $\rightarrow$ Element-wise Mult |
| **Decoder (Upsampling Path)** | | | |
| up1 | $64 \times H/128 \times W/128$ | $32 \times H/64 \times W/64$ | ConvTranspose2d(64, 32, 2, 2) |
| | + skip from down7 | $32 \times H/64 \times W/64$ | Concat + EnhancedDoubleConv(64, 32) |
| up2 | $32 \times H/64 \times W/64$ | $32 \times H/32 \times W/32$ | ConvTranspose2d(32, 32, 2, 2) |
| | + skip from down6 | $32 \times H/32 \times W/32$ | Concat + EnhancedDoubleConv(64, 32) |
| up3 | $32 \times H/32 \times W/32$ | $32 \times H/16 \times W/16$ | ConvTranspose2d(32, 32, 2, 2) |
| | + skip from down5 | $32 \times H/16 \times W/16$ | Concat + EnhancedDoubleConv(64, 32) |
| up4 | $32 \times H/16 \times W/16$ | $32 \times H/8 \times W/8$ | ConvTranspose2d(32, 32, 2, 2) |
| | + skip from down4 | $32 \times H/8 \times W/8$ | Concat + EnhancedDoubleConv(64, 32) |
| up5 | $32 \times H/8 \times W/8$ | $32 \times H/4 \times W/4$ | ConvTranspose2d(32, 32, 2, 2) |
| | + skip from down3 | $32 \times H/4 \times W/4$ | Concat + EnhancedDoubleConv(64, 32) |
| up6 | $32 \times H/4 \times W/4$ | $16 \times H/2 \times W/2$ | ConvTranspose2d(32, 16, 2, 2) |
| | + skip from down2 | $16 \times H/2 \times W/2$ | Concat + EnhancedDoubleConv(32, 16) |
| up7 | $16 \times H/2 \times W/2$ | $8 \times H \times W$ | ConvTranspose2d(16, 8, 2, 2) |
| | + skip from down1 | $8 \times H \times W$ | Concat + EnhancedDoubleConv(16, 8) |
| **Output Layer** | | | |
| final_conv | $8 \times H \times W$ | $1 \times H \times W$ | Conv2d(8, 1, 1) |
| softmax | $1 \times H \times W$ | $1 \times H \times W$ | Flatten $\rightarrow$ Softmax($T$) $\rightarrow$ Reshape |