A comparative study of semi-automatic colorization based on color space using a single reference image

Chen-Yu Li [1], Chih-Ping Yen [2,*], Hong-Ming Kang [3]
[1, 2] Department of Information Management, Central Police University, Taoyuan, Taiwan, R.O.C.
[3] Personnel Office, Central Police University, Taoyuan, Taiwan, R.O.C.
{cyli, peter, kang} @mail.cpu.edu.tw
* Corresponding Author

Abstract

Image colorization is the conversion of black and white or grayscale images into color to enhance the visualization effect and information transmission capabilities of the image. It can be used to colorize historical photos, medical images, satellite photos or microscopic images to improve the accuracy of analysis, or to apply in machine learning models to enhance understanding. Grayscale image coloring is divided into automatic and semi-automatic methods. Automatic coloring relies on deep learning technology to generate results but lacks flexibility. Semi-automatic coloring improves accuracy through user-provided color strokes, reference images or text descriptions, which can make up for the disadvantages of automatic coloring. Addresses deficiencies in special scenarios to provide more natural and accurate results. Therefore, this study uses the semi-automatic coloring algorithm of Reference Images and combines it with image quality evaluation indicators to establish an analysis model to explore and compare the different performances reflected in different color spaces. This provides researchers or practitioners with such methods to choose the most suitable color space. Experimental results show that among the four color models of HSV, YCbCr, Lab* and YIQ, YCbCr has the best coloring effect. It can separate brightness and chrominance more effectively, conform to human visual characteristics and be computationally efficient. In addition, under the YCbCr color model, the PSNR value is 27.0291, indicating that the image quality is good but slightly distorted, and the SSI value is 0.8572, indicating that the image structure is well preserved and is very close to the original image.

Keywords: Semi-automatic colorization, color transfer, image quality assessment, SSI

1. Introduction

The main purpose of image colorization is to convert black and white or grayscale images into color to enhance the visualization and information transmission capabilities of the image. First of all, coloring historical photos can reproduce the real scenes at that time, making historical events more realistic and conducive to the preservation and inheritance of historical culture [1]. For example, many black-and-white photos of wars, social changes, or important figures can be colored to allow viewers to feel the atmosphere of the past era more vividly. Secondly, coloring technology has important applications in the fields of medicine, geography and science. By colorizing medical images, satellite photos, or microscopic images, researchers can more visually identify different regions or structures, thereby improving the accuracy of their analyses [2]. For example, color X-ray or MRI images can help doctors diagnose patients' health problems more accurately. In addition, color images can enhance the understanding of machine learning models, allowing them to perform better in object recognition, classification and scene segmentation tasks [3]. Finally, image colorization is widely used in the film and media industries. Colorization of black and white films not only improves the visual effect, but also makes classic films more accessible to modern audiences and enhances the entertainment value [4].

The coloring methods of grayscale images can be divided into two types: Automatic Colorization and Semi-Automatic Colorization. The difference between the two is whether user intervention and guidance is required [5]. The automatic coloring method relies entirely on data-driven deep learning technology to automatically convert grayscale images into color by learning the distribution and pattern of image colors from a large amount of training data [6-7]. Among them, DeOldify [8] is a well-known open source automatic colorization model that can automatically color black and white photos. The advantage of this type of method is that it is simple and fast to operate and can generate results without manual adjustment, so it is efficient in certain applications. However, the performance of the automatic colorization model depends on the training data, which means that it cannot flexibly respond to the special needs of specific scenarios [9]. Due to limitations of the model training database, automatic coloring cannot accurately handle certain cases or spatiotemporal variations. For example, leaves in the spring should be green and leaves in the fall turn yellow, but automatic coloring may have difficulty distinguishing these cases correctly because it can only rely on general patterns learned from the data.

In contrast, semi-automatic coloring methods improve the accuracy and flexibility of coloring results through user participation. These methods allow the user to provide some guidance during the coloring process, allowing the system to be adjusted to specific needs. Common techniques for semi-automatic coloring methods include colored strokes [10-12], reference images [13-15], and semantic-driven text descriptions [16-17]. For example, users can manually draw some colored strokes on a grayscale image to provide a color reference for the system, or provide a color reference image so that the model can be colored according to the color distribution of this image. In addition, there are text-driven methods based on semantic information. Users can guide the coloring process by inputting text descriptions related to the image content, such as the description "Autumn leaves should be yellow", so that the model can produce results that are more suitable for the situation.

The advantage of semi-automatic coloring is that through user guidance, the system can more accurately reflect the color changes of objects under different environments, seasons, lighting and other conditions, thereby achieving a more natural effect. This makes semi-automatic coloring particularly suitable for situations that require detailed adjustments, such as the restoration of historical photos, artistic creation, or the restoration of certain scenes. Under this method, user intervention can effectively make up for the deficiencies of the automatic coloring system in handling special situations, thereby generating more accurate and expected coloring results.

Based on the different sensitivities of the human visual system to different color spaces, this study will compare the performance in different color spaces for the classic semi-automatic coloring method using reference images proposed by Tomihisa Welsh et al. [15]. This provides researchers or practitioners with such methods to choose the most suitable color space. Section 2 is Related Works, Section 3 is the model proposed in this study, Section 4 is the experimental results and discussion, and the last section is the conclusion.

2. Related Works

2.1. Color space

Color space is a model or system that uses numerical values or coordinates to represent colors. It provides a method for organizing and describing colors. Its purpose is to reproduce, record and display colors between different devices or systems. Each color space defines a specific range of colors and uses a specific set of parameters to represent color values. The choice of color space affects the display, output, and processing of digital images [18]. The following are some commonly used color spaces:

1) RGB

It is an additive color model used to describe the representation of digital colors. RGB is based on three basic colors—Red, Green, and Blue. Different intensity combinations of these three lights can produce a variety of colors. The value of each color channel is typically between 0 and 255 (or 0 and 1, depending on the representation). The three channels jointly define the final displayed color, which represents Color=(R,G,B). For example, pure red is represented by (255, 0, 0), pure green is represented by (0, 255, 0), pure blue is represented by (0, 0, 255), white is represented by (255, 255, 255), and black is represented as (0, 0, 0),

2) HSV

A color model based on the way humans perceive color, often used in applications such as image processing, design, and color selection. Compared with the RGB color space, HSV more intuitively uses a combination of three parameters: Hue (H), Saturation (S), and Value (V) to reflect color attributes. Hue defines the type of color, such as red, blue, yellow, and is expressed as an angle, ranging from 0° to 360°. For example, 0° is red, 120° is green, and 240° is blue. Saturation represents the purity or intensity of a color, ranging from 0 (gray, no color) to 100% (the purest color), with higher saturation indicating more vivid colors. Value represents the brightness of a color, ranging from 0 to 100%.

3) L*a*b*

It is a color model based on human visual perception and is used to accurately represent colors and compare color differences. Its structure consists of three components: L represents lightness, ranging from 0 (black) to 100 (white), which mainly controls the lightness and darkness of the color. The a-axis represents the color range from green to red, with negative values representing green and positive values representing red. The b-axis represents the color range from blue to yellow, with negative values representing blue and positive values representing yellow. The biggest feature of the L*a*b* color space is that it is device-independent, unlike RGB and CMYK color models that depend on specific devices (such as monitors or printers). This means the L*a*b* color space can maintain better color consistency across different devices and media. Therefore, it is commonly used in color management, image processing, and high-precision printing and color correction, especially in applications that require cross-platform color conversion.

4) YCbCr

It is a color model specifically used for digital image compression, video encoding and broadcasting. It is widely used in JPEG image compression, MPEG video compression and high-definition television standards (such as HDTV). It divides color information into three main components: Y, Cb and Cr. Y represents luma, which is the light and dark information of the picture. Brightness is very important to human vision because the human eye is more sensitive to changes in light and dark than to changes in color. Therefore, the Y component usually preserves most of the image details, and even during the image compression process, it will be retained preferentially to ensure image quality. Cb stands for blue-difference chroma, which is the difference between blue and brightness, and is responsible for the display of blue. Cr represents red-difference chroma, which represents the difference between red and brightness and is responsible for the display of red. The main difference between YCbCr and RGB color spaces is that RGB directly represents the three primary colors of red, green, and blue, while YCbCr separates brightness and chroma, which gives it greater advantages in compression. The human eye is far more sensitive to brightness than chroma, so the chroma resolution can be appropriately reduced while maintaining brightness. This can significantly reduce the file size while maintaining most of the image quality.

5) YIQ

It is a color model used to simulate color television broadcasts. This model separates color images into luminance (Y) and two chrominance components (I and Q) to provide compatibility with early black-and-white televisions while introducing color signals. The three components of YIQ: Y (Luminance) represents the brightness information of the image, which is the same as the black and white TV signal. In this way, the black and white TV can only read the Y component and display a clear black and white image. To the human eye, the luminance component is the most sensitive, so it occupies a dominant position in both color and black-and-white televisions. I (In-phase) represents the color range from orange to blue and is mainly used to provide color information of the image. The human eye is more sensitive to orange and blue variations, so the I component is given a higher bandwidth to deliver these colors. Q (Quadrature) represents the purple to green color range and works with the I component to provide complete color information. Since the human eye is less sensitive to these colors, the bandwidth of the Q component is smaller.

2.2. Transferring color to greyscale images

Tomihisa Welsh et al. [15] proposed a method to transfer the color of a color image to a grayscale image based on color similarity, as shown in Figure 1. This method uses Lab* color space to process image color transfer, and combines the brightness and neighborhood characteristics of pixels to achieve automated coloring effects. The detailed steps are as follows.

(a) (b) (c)

Figure 1: (a) Reference color image (b) target grayscale image (c) target grayscale image colorization processing [15].

1) Color space conversion

The color image is converted from RGB color space to $L^*a^*b^*$ color space according to Equation 1.

$$\begin{aligned} L^* &= 116 \cdot f(Y/Y_n) - 16 \\ a^* &= 500 \cdot (f(X/X_n) - f(Y/Y_n)) \\ b^* &= 200 \cdot \big(f(Y/Y_n) - f(Z/Z_n)\big), \end{aligned} \quad (1)$$

where L^* represents the brightness (that is, the pixel value of the grayscale image), a^* represents the chroma component from green to red, and b^* represents the chroma component from blue to yellow. X, Y, Z are the tristimulus values converted from the RGB color space, Xn=0.9515, Yn=1.0000, Zn=1.0886 are the tristimulus values of the reference white point, and $f(t)$ is a nonlinear function, such as Equation 2.

$$\begin{cases} t^{1/3} & if\ t > (6/29)^3 \\ (1/3)(29/6)^2 t + (4/29) & otherwise \end{cases} \quad (2)$$

2) Color matching

In the $L^*a^*b^*$ space, the brightness component L^* of the color image is first compared with the brightness of the grayscale image, and then matching pixels are found based on the similarity in brightness. For each pixel in the grayscale image, we find the pixel with similar brightness in the color image and transfer the a^* and b^* components (i.e. chroma) of that color pixel to the grayscale image, as in Equation 3.

$$d_L = \left|L^*_{gray} - L^*_{color}\right|, \quad (3)$$

where L^*_{gray} is the brightness value of the grayscale image, and L^*_{color} is the brightness value of the color image.

3) Neighborhood weighting

To avoid color transfer being too localized, this method considers not only the brightness matching of a single pixel, but also the brightness changes in the neighborhood of that pixel. This ensures that the surrounding pixels also have consistent color transitions when the brightness is similar, as shown in Equation 4.

$$w(p,q) = exp\left(-\frac{d_L^2(p,q)}{2\sigma_L^2}\right), \tag{4}$$

where p is a grayscale image pixel and q is a pixel with similar brightness found in the color image and weighted average based on its neighborhood. σ_L is a parameter that controls the brightness matching sensitivity, and $d_L(p,q)$ is the brightness difference between pixels p and q.

4) Color transfer

According to Equation 5, when the best match is found, the $a*$ and $b*$ components of the matching color pixel are transferred to the grayscale image, thereby obtaining the chroma components of the colored image:

$$\begin{aligned} a^* &= \sum_{q\in\Omega(p)} w(p,q)\cdot a^*(q) \\ b^* &= \sum_{q\in\Omega(p)} w(p,q)\cdot b^*(q) \end{aligned} \tag{5}$$

where $\Omega(p)$ is the set of pixels similar to the grayscale pixel p in the color image.

5) Smoothing

Finally, in order to avoid overly obvious color boundaries due to imprecise brightness matching or inconsistent neighborhood, this method performs smoothing processing to ensure that the transferred color transition is more natural. This is usually achieved through image processing techniques such as Gaussian smoothing.

3. Proposed Research Model

This study proposes the research model in Figure 2 based on the classic semi-automatic coloring method using reference images proposed by Tomihisa Welsh et al. [15] to explore and compare the different efficiencies reflected in different color spaces.

3.1. Describe the model process

Figure 2 illustrates the detailed research process of image colorization based on reference RGB images in different color spaces. The following is a step-by-step explanation, while Algorithm 1 is the detailed virtual code.

1) Real reference image

Provides a source reference image for color information to be used as a color reference for colorizing grayscale images.

2) Convert RGB images to images in different color spaces

The reference RGB color image is converted to other color spaces, such as HSV (hue, saturation, brightness), L*a*b* (lightness, a/b chroma channels), or YUV color space. These color spaces separate luminance and chrominance (color information), which makes it easier to analyze or apply color.

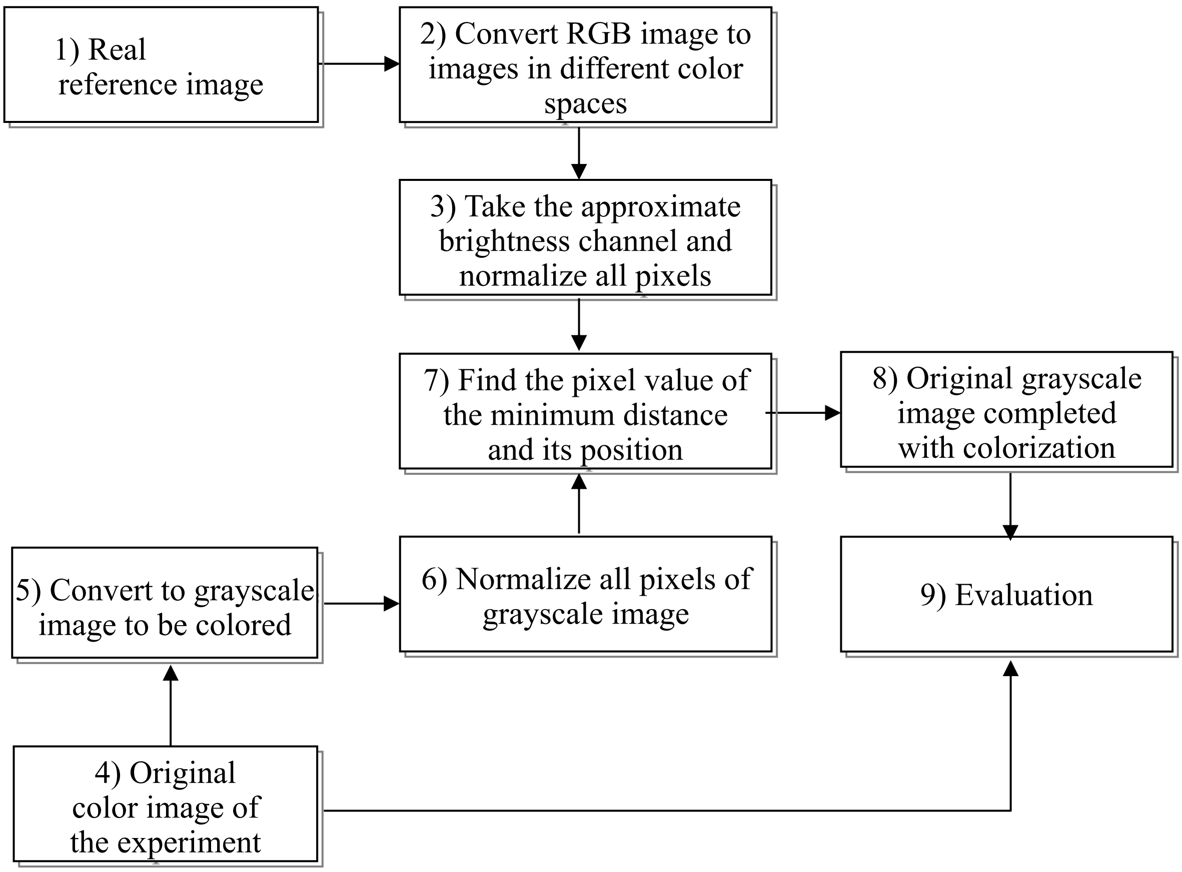


Figure 2: Proposed colorization model in different color spaces

3) Take the approximate brightness channel and normalize all pixels

Extract the brightness or lightness channel from different color spaces, which is the part that represents the grayscale information of the image. Normalize the pixel values in these brightness channels, that is, adjust the pixel values to the range of 0 to 1.

4) Original color image of the experiment

The original color image used for the experiment in this study, its grayscale image was used to recolor, and finally used to verify the effect of the coloring algorithm.

5) Convert to grayscale image to be colored

Convert the experimental color image into a grayscale image, that is, remove the color information in the image and retain only the brightness information for subsequent coloring processes.

6) Normalize all pixels of the grayscale image

Normalize the pixel values in the grayscale image to ensure consistent brightness values in the image. This is done to make the pixel comparison between the grayscale image and the reference image more accurate and consistent.

7) Find the pixel value of the minimum distance and its position

Match the pixel values of the experimental grayscale image with the pixels in the reference image. We use the minimum Euclidean distance to find the closest pixel between the two images, and then record the value and position of the pixel in the image.

8) Original grayscale image completed with colorization

Find the color information in the reference image according to the previous step and transfer the color to the experimental grayscale image.

9) Evaluation

The performance is evaluated between the colored grayscale image and the original color image of the experiment. And use various evaluation metrics of image quality to analyze the experimental results. The relevant evaluation metrics are explained in subsection 3.2.

Algorithm 1: Construct_Target_Color_Image

Input:

R = reference RGB image

I = target grayscale image

Output:

T = final RGB image

1: **function** *Construct_Target_Color_Image*(R, I)

2: R_{YCbCr} = convert R from RGB to YCbCr

R_Y = extract Y channel from R_{YCbCr}
$N_{RY} = normalize(R_Y)$
$N_I = normalize(I)$
for pixel in N_I **do**
    target_value = pixel value in *N*I
    closest_value, *position* = *find_closest*(N_{RY}, *target_value*)
    Cb, *Cr* = extract *CbCr*(R_{YCbCr}, *position*)
    construct T_{YCbCr} pixel with *(target_value*, *Cb*, *Cr*)
end for
T = convert T_{YCbCr} back to RGB color model
return *T*
end function

3.2. Image coloring quality assessment

This study will use various evaluation metrics of image quality to analyze the experimental results of coloring, and understand the different performances reflected in different color spaces.

3.2.1. Mean Square Error (MSE)

Equation (6) is utilized to represent the mean square error (MSE) between the original image and the noisy image. MSE serves as a metric to measure the pixel-wise absolute differences between the two images, providing an assessment of the noise's effect on the original image. This method is crucial for evaluating noise levels and plays a significant role in studies related to image processing and noise reduction [19].

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(I_{ij} - J_{ij})^2}{MN}, \tag{6}$$

where I_{ij} is the original signal, $\hat{I}_{ij}$ indicates the noise added noise to I_{ij}, $(I_{ij} - \hat{I}_{ij})$ is the noise, and *M*, *N* are the image length and width.

3.2.2. Peak Signal to Noise Ratio (PSNR)

Peak signal-to-noise ratio (PSNR) is used to evaluate signal quality by referring to the peak signal value. For an 8-bit grayscale image, the pixel value range is 0 to 255. Regardless of the actual maximum intensity of the pixel, the peak signal value is always 255 [20], as shown in Equation (7). A higher PSNR suggests better signal quality, as the noise has less impact on the clarity of the signal perceived by the human senses.

$$PSNR = 10\ log_{10}\left(\frac{MN\ 255^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}(I_{ij}-\hat{I}_{ij})^2}\right) = 10\ log_{10}\left(\frac{255^2}{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(I_{ij}-\hat{I}_{ij})^2}{MN}}\right), \tag{7}$$

where I_{ij} is the original image, $\hat{I}_{ij}$ indicates the noise added to I_{ij}, $(I_{ij}-\hat{I}_{ij})$ is the noise, and M, N are the image length and width.

3.2.3. Structural similarity Index (SSI)

An index for evaluating the similarity between two images is presented in Equation (8). The Structural Similarity Index (SSI) value is derived from the product of three components: luminance L, contrast C, and structure S of the images being compared. This comprehensive approach allows for a more nuanced comparison of image quality. The calculations for these components are elaborated in Equations (9-11), providing a clear framework for assessing how closely the two images resemble each other based on these key characteristics [20].

$$SSI(i,j) = L(i,j)\, C(i,j)\, S(i,j) \tag{8}$$

$$L(i,j) = \left(\frac{2\mu_i\mu_j + c_1}{\mu_i^2 + u_j^2 + c_1}\right), \tag{9}$$

$$C(i,j) = \left(\frac{2\sigma_i\sigma_j + c_2}{\sigma_i^2 + \sigma_j^2 + c_2}\right), \tag{10}$$

$$S(i,j) = \left(\frac{\mu_{ij} + c_3}{\sigma_i\sigma_j + c_3}\right), \tag{11}$$

where (i, j) represents the pixel coordinates, μ_i, μ_j indicates the average pixel intensity in an region (e.g. 11x11) centered at (i, j), while σ_i^2 and σ_j^2 denote the variances of the pixel values, and μ_{ij} is the cross-covariance. $c_1, c_2,$ and c_3 are constant values.

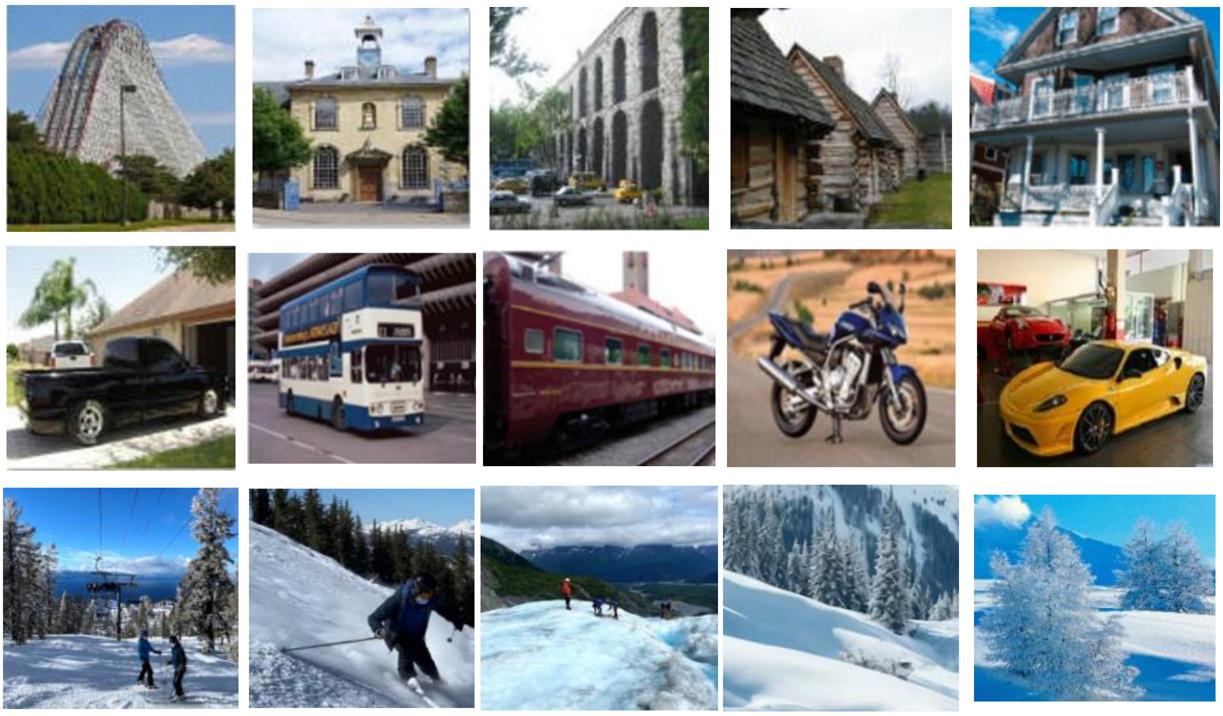
4. Experimental Results and Discussion

4.1. Experimental environment

The experimental setup consists of a personal computer with an Intel Core i5-11400F CPU (4.4 GHz), 16 GB of RAM, and Windows 11 Pro (x64) operating system. All experiments are conducted in the Matlab R2024a development environment, providing sufficient processing power and memory for the necessary computations and analyses.

4.2. Image dataset

We consider 3 image datasets: Places365 [21] is a scene recognition dataset with 10 million images across 434 classes. ImageNet [22] is the Large Scale Visual Recognition Challenge (ILSVRC) 2012-2017 image classification and localization dataset. This dataset covers 1000 object categories and contains 1,281,167 training images, 50,000 validation images, and 100,000 test images. UCI Colorization Dataset [23] is designed for image colorization tasks, the images include National Park, Sunset, Nature, River Images, Sky Images, Village, Barn, etc., a total of 710 images. From the aforementioned data set, 8 categories (i.e. Mountain, Ocean, Forest, Snow scene, Building, Office, Vehicle and Street) of images were randomly selected, with 25 images in each category, totaling 200 images for the experiment, as shown in Figure 3. During the experiment, each image in each category was used as a reference image in turn.



(a)
(b)
(c)

Figure 3: (a) building category (b) vehicle category (c) snow scene category.

4.3. Performance comparison under different color spaces

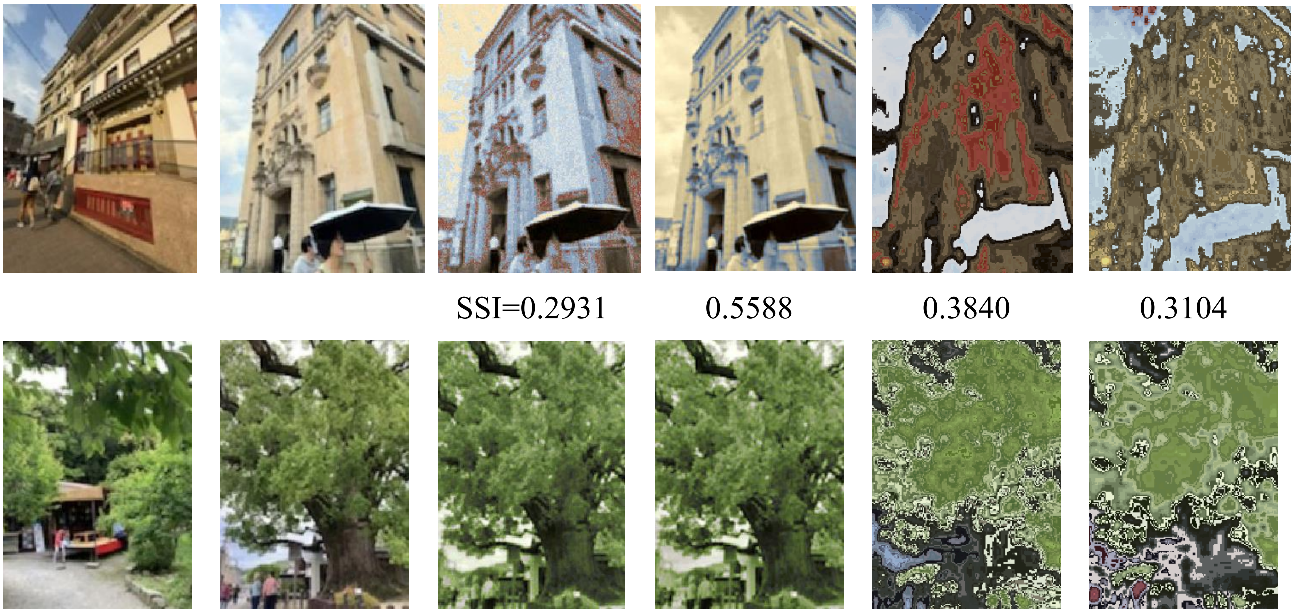
In this research model, the color experimental image is first converted to grayscale, and then colored in different color spaces according to a single reference image, and then the performance in different color spaces is discussed and compared. Figure 4 is the colorization image of the experimental image using the YCbCr color model. Figure 5 is the color image of the experimental image in HSV, YCbCr, L*a*b*, YIQ color space, as well as individual SSI values.

(a)
(b)
(c)
(d)

Figure 4: (a) reference image (b) experimental color image (c) experimental graylevel image (d) image (c) colorization with YCbCr color model.

According to subsection 4.2, we randomly selected 8 categories of images from the three data sets of Places365, ImageNet, and UCI, with 25 images in each category, for a total of 200 images for the experiment. Then use MSE, PSNR, and SSI as evaluation indicators and conduct analysis. We take each image in each category as a reference image in turn, and then average all the evaluation results. The final average evaluation results of 8 categories are shown in Table 1. The following is the analysis and discussion of two points.

1) From the observation of MSE, PSNR, and SSI evaluation indicators, it is found that among the four color models of HSV, YCbCr, L*a*b*, and YIQ, YCbCr has the best coloring effect. The main reason why YCbCr has good coloring effect is that it can separate brightness and chroma more effectively, conform to human visual characteristics, and has obvious advantages in calculation efficiency. Although HSV separates chroma and brightness, its hue and saturation performance can easily lead to distortion during compression. Especially under non-linear conversion, color information is easily lost. In addition, although YIQ and LAB also separate brightness and chromaticity, they are not as good as YCbCr in practical applications.



SSI=0.2931 0.5588 0.3840 0.3104

SSI=0.8421 0.8685 0.2863 0.2918

(a) (b) (c) (d) (e) (f)

Figure 5: (a) reference image (b) experimental color image, and colorization with (c) HSV (d) YCbCr (e) L*a*b* (f) YIQ color model, as well as individual SSI values.

2) Analyze PSNR and SSI under YCbCr color model. Generally speaking, a PSNR value above 30 dB is considered a high-quality image, a PSNR value between 25-30 dB indicates medium image quality, and a PSNR value below 25 dB shows obvious image distortion. Therefore, PSNR=27.0291 means that the image has a certain degree of distortion, but the overall quality is still good. SSI takes into account the sensitivity of the human visual system to structure, focusing not only on

pixel-level differences but also on the contrast, brightness and structural characteristics of the image. The closer the SSI value is to 1, the more similar the two images are in structure, and SSI = 0.8572 is a very high score, indicating that the image is very close to the original image in terms of structural preservation. Therefore, SSI performs slightly better than PSNR, showing better preservation of image structure. In summary, both indicators show that the image quality is good. If you pay more attention to pixel-level accuracy and detail retention, PSNR is an important evaluation criterion; but if you emphasize the structural integrity and visual effects of the image, SSI is more representative.

Table 1: Performance comparison under different color spaces based on image quality evaluation

Color models	Evaluation indicator		
	MSE	PSNR	SSI
HSV	430.1280	21.7948	0.7647
YCbCr	**128.8759**	**27.0291**	**0.8572**
L*a*b*	2186.1123	14.7341	0.6492
YIQ	3890.3378	12.2309	0.6370

We also list the evaluation situations of the YCbCr color model for each category: mountain, ocean, forest, snow scene, building, office, vehicle, street, as shown in Table 2. It shows that natural scenery (such as Mountain, Ocean, Forest, Snow scene) has better image coloring effect than man-made scenery (such as Building, Office, Vehicle, Street), and there is a clear difference. This may be because the image colors and structures of natural scenery are softer and regular, while artificial scenery contains more straight lines, angles, complex geometric structures or widely different colors. For example, in the Ocean category, all image scene features, structural complexity, textures and colors are relatively consistent, so the coloring effect is the best, with the smallest MSE, the highest PSNR, and the highest SSI. The differences among all images within the Vehicle category are very large, which can be clearly observed from Figure 3 (b).

Table 2: Performance evaluation of each image category under the YCbCr color model

Image category	Evaluation indicator		
	MSE	PSNR	SSI
Mountain	99.6561	30.1053	0.9175

Ocean	86.4398	31.1159	0.9537
Forest	101.7921	29.9653	0.9114
Snow scene	87.4380	30.5638	0.9298
Building	153.1755	23.0532	0.8155
Office	147.2516	24.4856	0.7823
Vehicle	219.4893	21.0590	0.7181
Street	135.7650	25.8843	0.8296
Average	**128.8759**	**27.0291**	**0.8572**

5. Conclusion

This study analyzes the performance of semi-automatic coloring algorithms in different color spaces, and conducts in-depth comparisons based on image quality evaluation indicators. When coloring algorithms are used to color historical photos, medical images, satellite photos, etc., the choice of color space is crucial to the final effect. Therefore, the experimental results show that the YCbCr color model performs best among the four common color space models: HSV, YCbCr, Lab* and YIQ. The YCbCr model can effectively separate brightness and chromaticity, conforms to the perceptual characteristics of human vision, and has higher computational efficiency. Under the YCbCr color model, the experimentally obtained PSNR value is 27.0291, showing good image quality with only slight distortion, while the SSI value reaches 0.8572, indicating that the image structure is well preserved and is close to the original image. The contribution of this study is to provide an empirical basis for color space selection, which can be used as a reference for researchers and practical practitioners to help them choose the most suitable color space to improve the effect of image coloring.

References

[1] M. R. Joshi, L. Nkenyereye, G. P. Joshi, S. M. R. Islam, M. Abdullah-Al-Wadud, S. Shrestha, “Auto-colorization of historical images using deep convolutional neural networks,” Mathematics, Vol. 8, No. 12, 2020.

[2] E. M. Farella, S. Malek, F. Remondino, “Colorizing the past: Deep learning for the automatic colorization of historical aerial images,” Journal of Imaging, Vol. 8, No. 10, 2022.

[3] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, et al. “Semantic colorization with internet images.” ACM Transactions on Graphics (ToG), Vol. 30, Issue 6, 2011, pp. 1–8.

[4] M. Golyadkin, I. Makarov, “Robust manga page colorization via coloring latent space,” IEEE Access, Vol. 11, October 2023, pp. 111581-111597.

[5] S.-Y. Chen, J.-Q. Zhang, Y.-Y. Zhao, P. L. Rosin, Y.-K. Lai, L. Gao, “A review of image and video colorization: From analogies to deep learning,” Visual Informatics, Vol. 6, Issue 3, Sep. 2022, pp. 51-68.

[6] S. Huang, X. Jin, Q. Jiang, L. Liu, “Deep learning for image colorization: Current and future prospects,” Engineering Applications of Artificial Intelligence, Vol. 114, Sep. 2022.

[7] S.-Y. Chen, J.-Q. Zhang, Y.-Y. Zhao, P. L. Rosin, Y.-K. Lai, L. Gao, “A review of image and video colorization: From analogies to deep learning,” Visual Informatics, Vol. 6, Issue 3, Sep. 2022, pp. 51-68.

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

[9] A. Deshpande, J. Rock, D. Forsyth, “Learning large-scale automatic image colorization,” 2015 IEEE International Conference on Computer Vision (ICCV), Dec. 2015.

[10] L. Yatziv, G. Sapiro, “Fast image and video colorization using chrominance blending,” IEEE Transactions on Image Processing, Vol. 15, No. 5, 2006, pp. 1120-1129.

[11] A. Levin, D. Lischinski, Y. Weiss, “Colorization using optimization,” ACM Transactions on Graphics, Vol. 23, No. 3, 2004, pp. 689–694, http://dx.doi.org/10.1145/1015706.1015780.

[12] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, H.-Y. Shum, “Natural image colorization,” EGSR'07: Proceedings of the 18th Eurographics conference on Rendering Techniques, Eurographics Association, Jun. 2007, pp. 309–320.

[13] M. Shi, J. Q. Zhang, S. Y. Chen, et al., ”Reference-based deep line art video colorization,” IEEE Transactions on Visualization and Computer Graphics, Vol. 29, No. 6, 2023, pp. 2965-2979.

[14] J.-W. Su, H.-K. Chu, J.-B. Huang, “Instance-aware image colorization,” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 7965-7974.

[15] T. Welsh, M. Ashikhmin and K. Mueller, “Transferring color to greyscale images,” ACM Transactions on Graphics, Vol. 21, No. 3, 2002, pp. 277-280. doi:10.1145/566570.566576

[16] S. Ghosh, P. Roy, S. Bhattacharya, U. Pal, M. Blumenstein, “TIC: text-guided image colorization using conditional generative model,” Multimedia Tools and Applications, Vol. 83, 2024, pp. 41121–41136.

[17] X. Zhao, Z. Du, “Video colorization with semantic preservation and consistency,” Computers and Electrical Engineering, Vol. 108, May 2023.

[18] R. C. Gonzalez, R. E. Woods, Digital Image Processing 4/e, Pearson FT Press, 2017.

[19] G. James, D. Witten, T. Hastie, R. Tibshirani. An Introduction to Statistical Learning: with Applications in R (2nd edition), 2021, Springer.

[20] A. Hore, D. Ziou, "Image quality metrics: PSNR vs. SSIM," International Conference on Pattern Recognition, Aug. 2010, pp. 2366-2369.
[21] Github, The Places365-CNNs for Scene Classification. https://github.com/CSAILVision/places365 (accessed 16 Jul. 2024)
[22] ImageNet. https://www.image-net.org/index.php (accessed 18 Jul. 2024)
[23] Kaggle, Image Colorization. https://www.kaggle.com/datasets/mertbozkurt5/image-colorization/data (accessed 16 Jul. 2024)