Journal Pre-proof

A review of image and video colorization: From analogies to deep learning

Shu-Yu Chen, Jia-Qi Zhang, You-You Zhao, Paul L. Rosin,
Yu-Kun Lai, Lin Gao

PII: S2468-502X(22)00038-9
DOI: https://doi.org/10.1016/j.visinf.2022.05.003
Reference: VISINF 141

To appear in: *Visual Informatics*

Received date : 6 December 2021
Revised date : 11 April 2022
Accepted date : 13 May 2022

A review of image and video colorization: From analogies to deep learning

Shu-Yu Chen[a], Jia-Qi Zhang[b], You-You Zhao[c], Paul L. Rosin[d], Yu-Kun Lai[d], Lin Gao[a,e,1]

[a] *Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*
[b] *Beihang University, Beijing, China*
[c] *University of California, Santa Cruz, California, US*
[d] *Cardiff University, Wales, UK*
[e] *University of Chinese Academy of Sciences, Beijing, China*

Abstract

Image colorization is a classic and important topic in computer graphics, where the aim is to add color to a monochromatic input image to produce a colorful result. In this survey, we present the history of colorization research in chronological order and summarize popular algorithms in this field. Early works on colorization mostly focused on developing techniques to improve the colorization quality. In the last few years, researchers have considered more possibilities such as combining colorization with NLP (natural language processing) and focused more on industrial applications. To better control the color, various types of color control are designed, such as providing reference images or color-scribbles. We have created a taxonomy of the colorization methods according to the input type, divided into grayscale, sketch-based and hybrid. The pros and cons are discussed for each algorithm, and they are compared according to their main characteristics. Finally, we discuss how deep learning, and in particular Generative Adversarial Networks (GANs), has changed this field.

Keywords: Image Colorization, Sketch Colorization, Manga Colorization

[1] Corresponding Author.

1. Introduction

Color plays a very important role in the process of human cognition of the world, and rich colors can not only express more information, but also enhance the human visual experience. Image colorization has been a very active research topic in the field of digital image processing, and is an inter-disciplinary area involving disciplines such as Computer Vision, Computer Graphics, Pattern Recognition and Human Computer Interaction. Colorization has been widely used in many fields, such as grayscale photo colorization, old film color restoration, cartoon automatic colorization, etc.

Based on different types of input to be colorized, colorization methods can be divided into two broad categories, one is colorization of grayscale images and black-and-white videos which are ordinary photos without color, and the other is colorization of monochrome art forms, including sketch images (or line art images), manga (or comics) and black-and-white cartoons (or line-art video). See Figure 1 for some examples. We can see that grayscale images contain rich intensity details, while sketch images (or other art forms) only contain relatively sparse details.

Therefore, when processing input images of different categories, researchers usually use different processing methods. For colorization of grayscale images, most methods convert the image in YUV or Lab color space [1, 2], and restore the value of the chrominance channels of the image to be colored based on the similarity of the luminance channel [3]. For black-and-white videos, most models use unsupervised or self-supervised learning from the visual tracking process to track the location of an object in different frames, and link corresponding pixels together, to colorize them based on a user-provided reference photo or based on data-driven deep learning technologies. The colorization of sketch images often involves segmenting it into different regions [4], and based on a learning model a color is assigned to each segment where the color information can come from reference images, users' color scribbles or input text hints. Although grayscale-based colorization methods can be directly used to predict the color value of each pixel in sketch images, they usually do not have good performance due to the lack of texture information. Therefore sketch-based colorization methods are required to propose new solutions for line feature extraction and re-

(a) Grey-scale Image (b) Manga Image (c) Line Art Image (d) Sketch Image

Figure 1: Typical categories of images suitable for colorization.

gion boundary determination, such as studying the temporal and semantic relationships between lines [5, 6, 7].

Generally speaking, traditional colorization methods require a lot of manual interaction and are often sensitive to the parameter settings of the methods. As a result, adding manual interaction and parameter optimization will take a lot of time and effort. Especially in grayscale videos or cartoon film colorization, even a short film usually has thousands of images to process. To improve efficiency, using a DCNN (Deep Convolutional Neural Network) [8] to build up the model or a GAN (Generative Adversarial Network) [9] for training is the most common approach used in recent methods. Both the image colorization effect and efficiency have been greatly improved. There exist both opportunities and challenges, and the development of deep learning technology has brought new directions to the work of image colorization.

Several major challenges remain. Some methods can only be used under certain restrictions, and moreover have some defects. For example, the colorization method can only handle gray-scale images, or the model needs to provide suitable reference color images. Some models need to identify different objects in the image, and then work out appropriate colors, but in particular for sketch image colorization, it is very difficult for the model to understand the sketch image and learn different artistic styles. The existing survey [31] mainly summarizes works performing colorization of grayscale images and the datasets for colorization. However, the task of colorization is not restricted to grayscale images, but also includes manga and sketches. In this paper, we will summa-

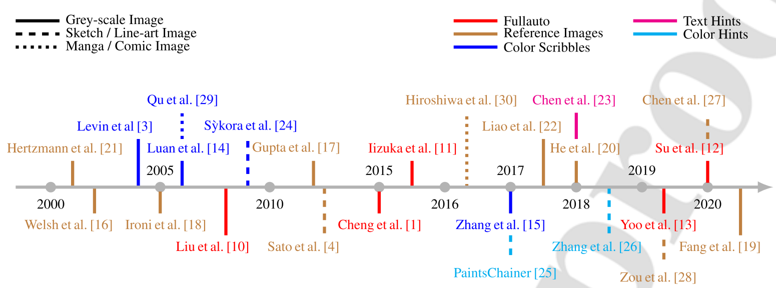

Figure 2: The timeline of image colorization methods. Different line colors represent methods with different types of control, as shown in the upper-right. Different line types represent the different types of input, as indicated in the upper-left. From this timeline, it can be seen that there can be a wide variety in the type of input data, from gray-scale image to manga to sketch, which shows the difficulty of data processing. Compared with gray images, sketch images are sparse and the information available from just lines can be ambiguous. Early approaches were based on reference images, then user interaction was introduced, and finally fully automatic colorization.

rize and discuss different colorization methods from three categories, including their advantages and drawbacks, to give an overview that should be useful for researchers and practitioners.

2. Overview

This survey paper divides existing colorization research work into the following three sections based on the different types of input images to be colored. In Section 3, we mainly introduce the colorization methods for grayscale images, which are further grouped into three subcategories: fully automatic colorization methods, semi-automatic colorization methods based on color strokes or reference images, and text-driven image colorization methods. In Section 4, we focus on methods related to colorization of line-art or sketch images, which are further classified into four subcategories: colorization methods based on color strokes, colorization methods based on reference images, text-driven colorization methods, and synthesis methods from line-art images to real images. In Section 5, we discuss the colorization work of comic or manga images. Finally, we summarize the colorization methods and discuss a possible area for future colorization work in Section 7. Figure 2 shows a timeline of representative methods for image colorization.

3. Grayscale Image Colorization

The color value of each pixel on a grayscale image is between black and white. The grayscale channel can be extracted from color images, but images such as photos taken in the past and much comic art only have grayscale information, and can benefit from colorization. Colorization methods for grayscale images can be divided into two groups according to whether interaction is used, namely automatic colorization methods and semi-automatic colorization methods. In the former group, researchers have used data-driven deep learning technology to automatically colorize grayscale images based on training data [1, 11, 32]. For instance, there is an open source automatic model DeOldify [33] which is free to use. For the second group, colorization methods often take some guidance information from users, e.g., by drawing color strokes [3], providing reference color images [10] or giving an specific color theme[34]. Incorporating user guidance usually increases the efficiency and correctness of the colorization model. It also helps resolve inherent ambiguities for the ill-posed colorization problem (such as tree leaves could be green in spring and yellow in autumn). In addition, in some recent studies, researchers have also studied the use of semantic information to guide image colorization, such as grayscale colorization based on text scripts [35], which is introduced in the last subsection.

3.1. Automatic Grayscale Image Colorization

Different from colorization methods based on guidance, in automatic image colorization methods, researchers can design the model to provide multiple colors for the same pixel to solve the problem of multi-modal colorization of monochrome images. For example, colorization models will generate green or yellow leaf images. In this subsection, we divide the automatic coloring method into two categories based on the diversity of the generated results; one is unimodal colorization in which methods can only generate one result, and the other is multi-modal colorization where methods can generate multiple diverse results, and introduce them respectively.

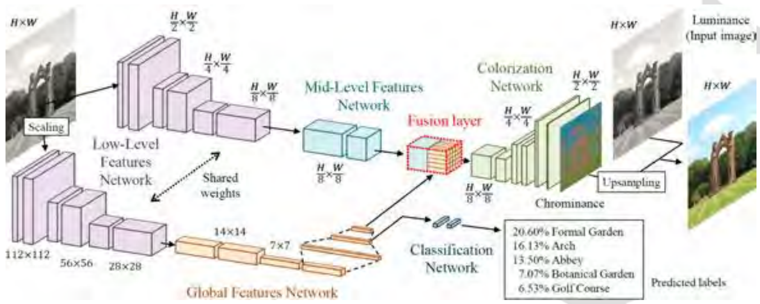

Figure 3: An overview of an end-to-end network for grayscale images colorization [11]. Reproduced with permission from Ref. [11], ©Association for Computing Machinery, Inc. 2016.

3.1.1. Unimodal Colorization

To reduce user interaction, Cheng et al. [1] first propose a fully-automatic colorization method using deep learning with the SUN dataset [36]. Instead of direct taking the grayscale image as input, they take a combination of multi-level features to predicts the U and V channels. However the performance drops when similar reference images are not included from the training data-set. Concurrently, Deshpande et al. [37] improved the learning model for image colorization and learned from examples. This learning model is built upon the LEARCH (Learning to Search) framework [38], and is able to minimize the quadratic objective function defined on the chromaticity maps, comparable to a Gaussian random field.

Larrson et al. [39] proposed an automatic colorization method based on a self-supervised visual representation learning process. The network is built upon the fully convolutional network of VGG-16 [40] with the classification layer removed and a filter layer added. In addition, the model uses skip-layer connections to concatenate the features of different convolutional layers, to provide input to the classification layer which predicts the color histogram of each pixel. Iizuka et al. [11] proposed to learn the global and local features separately from an image and then combine them together for the final colorization process. However, for objects with multiple different colors, the result will most likely produce dominant colors which is learnt in training, like the leaves in green.

Ablations. We validate our design choices by comparing with several alternative options.

n Part

uff validation split	
PSNR ↑	SSIM ↑
27.251	0.938
28.146	0.942
27.959	0.941
28.592	**0.944**

(b) Different Bounding Box Selection

Box Selection	COCOStuff validation split		
	LPIPS ↓	PSNR ↑	SSIM ↑
Select top 8	**0.110**	**28.592**	**0.944**
Random select 8	0.113	28.386	0.943
Select by threshold	0.117	28.139	0.942
G.T. bounding box	0.111	28.470	**0.944**

(c) Different Weighted Sum

Weighted Sum	COCOStuff validation split		
	LPIPS ↓	PSNR ↑	SSIM ↑
Box mask	0.140	26.456	0.932
G.T. mask	0.199	24.243	0.921
Fusion module	**0.110**	**28.592**	**0.944**

eight bounding boxes in terms of by object detector performs best using the ground-truth bounding t with two alternative approaches a mask or using the ground-truth the COCO-Stuff dataset) for fus- potentially overlapping object in- om the full-image network. Using a notable performance boost than is shows the capability of our fu- e challenging scenarios with mul-

rk involves two steps: (1) coloriz- es and outputting the instance fea- stance features into the full-image full-image colorization. Using a 00X 3.30GHz CPU, 32GB mem- 080ti GPU, our average inference nts is 0.187s for an image of res- of two steps takes approximately , while the complexity of step 1 ber of input instances and ranges) to 0.1s (eight instances).

Expert Our results

photographs. The middle column d results by the experts.

(a) Missing detections (b) Superimposed detections

Figure 8. **Failure cases.** (*Left*) our model reverts back to the full-image colorization when a lot of vases are missing in the detection. (*Right*) the fusion module may get confused when there are many superimposed object bounding boxes.

Previous colorization methods usually learn to colorize an entire image, and thus they are often unsuitable for colorizing instances in the image. Su et al. [12] proposed a method to realize instance-aware colorization. They first detect the location of the target objects, and then colorize the object and the overall image respectively. When there is miss detection or overlap detection in the same class instance, the result will be affected, as shown in Fig. 4. Considering the constraints of training data, Yoo et al. [[illegible]] little data. Their whole network consist of memory network and colorization networks. The memory network is trained in unsupervised way to help get the most similar color features that match the input image, and then give [illegible] when given arbitrary grayscale images, but only one corresponding color image can be generated for a grayscale input. A single generated result may not meet user expectations and the user cannot specify local or object-specific colors. Various results could be gener[illegible]tors into the network [illegible].

5.6. Colorizing legacy black and white photos

We apply our colorization model to colorize legacy black and white photographs. Figure 7 shows sample results along with manual colorization results by human expert[1].

5.7. Failure modes

We show 2 examples of failure cases in Figure 8. When the instances were not detected, our model reverts back to the full-image colorization network. As a result, our method may produce visible artifacts such as washed-out colors or bleeding across object boundaries.

6. Conclusions

We present a novel instance-aware image colorization. By leveraging an off-the-shelf object detection model to crop out the images, our architecture extracts the feature from our instance branch and full images branch, then we fuse them with our newly proposed fusion module and obtain a better feature map to predict the better results. Through extensive experiments, we show that our work compares favorably against existing methods on three benchmark datasets.

Acknowledgements. The project was funded in part by the Ministry of Science and Technology of Taiwan (108-2218-E-007 -050- and 107-2221-E-007-088-MY3).

[1] bit.ly/color_history_photos

3.1.2. Multi-Modal Colorization

Aiming at solving the problem that colorization requires a lot of user interaction and that the color saturation of colorized images tends to be low, Zhang et al. [2] proposed a fully automatic colorization method that can generate rich and realistic colorization images. This method transforms the colorization task into a self-supervised expression learning task by learning the semantic and texture mapping between the grayscale image and the color image. At the same time, the colorization problem is transformed in a novel manner into a classification task, and a color distribution is predicted for each pixel to solve the multi-modal colorization problem of the image, which maintains the diversity of the colorization results. Zhang et al. [2] were inspired by the simulated annealing method [41] and proposed the operation that works out the annealed mean of a distribution, to estimate the color value of the ab space from the color distribution of each pixel. The value of each pixel in the grayscale image colorization task is not fixed, and the same object in the real world can be colored in different ways.

Unlike [2], Deshpande et al. [42] not only considers the estimation of the color value of each pixel, but also considers the overall spatial continuity of the colorization results. This method uses a variational autoencoder (VAE) to learn the low-dimensional latent variable embedding of the color field, and uses a Mixed Density Network (MDN) to learn a multi-modal model conditioned on the grayscale image. Finally, multiple samples are taken from MDN, and combined with the VAE decoder to obtain multiple colorization results for each sample, so as to provide a rich set of colorization results.

Although the classification model based on color distribution and the generative model based on variational autoencoders can obtain a variety of colorization schemes, the colorization results lack the consistency of the spatial structure and the user controllability of color. Sometimes in the same semantic area, spots of different colors appear in the colorization result. In order to ensure global colorization consistency and user controllability, Messaoud et al. [32] proposed a conditional random field based on VAE and use a Gaussian Conditional Markov Random Field (G-CRF) to capture global image statistics, modeling the output space of the VAE decoder and the encoding of user editing information.

When an image colorization method is directly applied to video colorization, discontinuity will appear. Lei et al. [43] proposed an automatic colorization model for black-and-white video without any user interaction or reference image. This method designs a self-regularization and diversity loss function in order to achieve the consistency and diversity of the grayscale video colorization. The self-regularization loss is mainly composed of a bilateral regularization term and a temporal regularization term, which adds color consistency constraints in the bilateral space of adjacent pixels and corresponding pixels of adjacent frames. Diversity Loss to constrain the multiple generated results to be consistent with real color images. Although the method achieves the generation of multiple colorization results, there are no rich colorization results between different results.

With the rapid development of Transformer [44] in the field of computer vision, Kumar et al. [45] proposed a grayscale colorization network architecture (Colorization Transformer, ColTran) based on Transformer blocks. ColTran is mainly composed of a autoregressive Colorizer, a color upsampler and spatial upsampler. Autoregressive Colorizer enabled color information to be matched to input grayscale images at low resolution, and then the color upsampler and spatial upsampler sampled low resolution color images into high resolution images in a completely parallel way. Based on transformer's better matching ability, this method can provide a variety of colored gray images according to different reference color images.

Compared with the unimodal colorization, multi-modal colorization methods can generate multiple color results for a given grayscale input. Although those automatic methods do not require user interaction, the generated results rely on pretrained network models. The user cannot adjust the generated results, such as the overall colorization style or detail colors, making it difficult to generate the results the user expects.

3.2. Color Strokes based Colorization

In order to solve the problem that automatic methods cannot control the color of the details, some work attempt to take user color strokes and provide an intuitive approach for user control.

Figure 5: Comparison of stroke based colorization. (a-c) Color strokes, strokes alone, colorization result by Levin et al. [3], (d-f) Color strokes, two pixels in each region are labeled, colorization result by Luan et al. [14]. Reproduced with permission from Ref. [3], ©Association for Computing Machinery, Inc. 2004, and Ref. [14], IEEE 2007.

3.2.1. Optimization Colorization

Levin et al. [3] were one of the most important pioneers in the colorization area. In this method, the user needs to mark a grayscale image with color strokes to colorize the image in YUV color space. Then, based on the rule that adjacent pixels have similar intensities and their colors are similar, the method spreads the color of the strokes to the entire image. But when different object colors are diffused and mixed together, there are color bleeding problems in [3]. To solve it, Huang et al. [46] modified the weighting function and proposed an adaptive edge detection algorithm to improve the accuracy of the edges. They use Sobel filters and iterative optimization to improve the edge detection. Further, the colorization method will be more accurate, while at the same time reducing color bleeding issues, and making the image color effect more realistic.

Previous colorization methods based on color strokes, such as Levin's method [3], usually require a lot of manual interaction for a complex scene. To reduce it, Luan et al. [14] proposed a new interactive system that can quickly and easily color grayscale images. This method consists of two stages, the color labeling stage and the color mapping stage. The color labeling stage spreads the marked colors to similar areas by constraining the intensity smoothness and texture similarity of all pixels. The color mapping stage establishes a piece-wise linear mapping in luminance (Y) space according to the scribble's luminance (Y) and chroma (UV) values, and finally the chromatic-

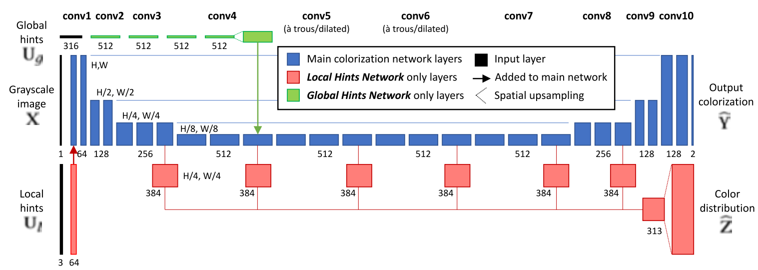

Figure 6: An overall user-guided image colorization network architecture [15]. Reproduced with permission from Ref. [15], ©Association for Computing Machinery, Inc. 2017.

ity values of other pixels are obtained by interpolation. The comparison between Levin et al. [3] and Luan et al. [14] is shown in Fig. 5. It can be seen that [14] is faster and more effective.

For monochrome movies, Yatziv et al. [47] uses intrinsic, gradient weight, and the relationship between colorized point and the nearby spot to colorize in the YCbCr color space. This method first defines the intrinsic (geodesic) distance of any two points to calculate the smoothness between the luminance channel of the two points. Then for any point on the image, the method calculates the shortest intrinsic distance from the point to the known chrominance point, finds multiple chrominance values corresponding to it, and obtains the final chrominance value by blending different chrominance values. But this process involves a large amount of calculation and a complicated process to ensure color quality.

In view of the value of each pixel in the grayscale image, the information contained in the grayscale image is rich, and the optimization method can determine the propagation range of the color according to the relationship between grayscale information or adjacent pixels. However, the computation cost of those methods is high. Therefore, most of them struggle to perform in real time. The emergence of deep neural networks has effectively reduced the time required for the process, and has subsequently inspired more colorization methods based on deep learning.

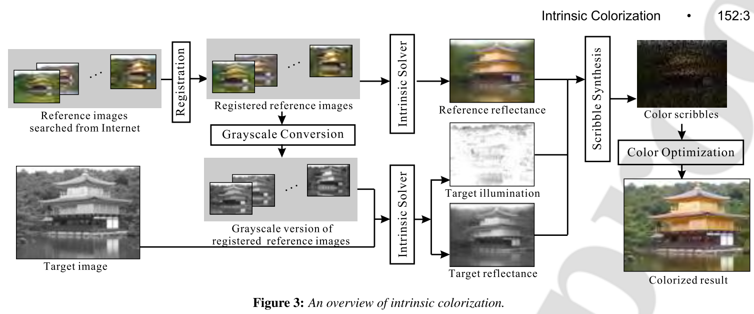

Figure 7: An overview of intrinsic grayscale image colorization with reference images retrieved from the Internet [10]. Reproduced with permission from Ref. [10], ©Association for Computing Machinery, Inc. 2009

3.2.2. Deep Neural Network based Colorization

In the recent work on colorization of grayscale images based on user guidance, researchers began to use Convolutional Neural Networks (CNNs) to learn the mapping relationship between grayscale images and color images with user constraints. Based on deep learning, Zhang et al. [5] proposed a network model that takes user input as guidance for real-time image colorization. The model is mainly divided into three parts: the main colorization network learns to colorize, the local hints network predicts a color probability distribution and the global hints network learn to encodes the input global histogram statistics and average image saturation into the middle part of the main colorization network. By combining the colorization network structure of global and local information, Zhang et al. [5] mapped the input image and user input to the output color image, and finally realized real-time user-guided image colorization. The specific network structure is shown in Fig. 6. Although the network can complete the colorization in real time with only a single forward propagation, it still requires the user to specify a large amount of color hints.

Neural network-based methods effectively improve the speed and quality of image colorization. However, the network training adopts an end-to-end training strategy. As a result, when the input and output are given, the output result cannot be controlled, and the result can only be optimized by editing the input. At the same time, this approach needs to specify the color of the area for each image, which cannot achieve batch colorization work. Therefore, some researchers study the colorization based on reference images.

3.3. Reference Color Image based Colorization

Another approach that balances controllability with user effort is reference based colorization where the user provides reference images with desired color distribution to guide the colorization process. The reference images may be specified directly by the user, retrieved from the Internet or obtained from a large dataset [20]. By referring to the reference image, the colorization results can better satisfy the user's expectations. Although there are substantial overall differences between the images, similarities between the images can still be found in local areas. For example, areas with similar color or texture often also have similarities in structure or lines. Therefore, we can guide the generation of images by finding similarities between the grayscale image and the reference image.

3.3.1. Similarity with Luminance Features

To colorize a grayscale image, these methods need to have one or more reference images, and then use luminance channel mapping with the input image. Hertzmann et al. [21] transferred color information into the input image from analogous regions of the reference. In the work by Welsh et al. [16], the grayscale image only contains one dimensional information, and for a color reference image, its luminance channel can be used to match the grayscale input. So the algorithm converts the reference image into $l\alpha\beta$ color space, and selects a small subset of pixels as a sample. Then the pixels in the grayscale image are scanned in raster order and the best matching part is selected using neighborhood statistics. Welsh et al. [16] described how their model could be applied to a single frame in a video sequence. They used the same colorized target swatch that was used in the first frame to colorize the remainder of the video. This process can effectively solve the problem of color inconsistency. After finding the corresponding pixel, they use the swatch model to produce a vivid colorization effect. In the equation, the error distance $E(N_g, N_s)$ uses the L_2 distance metric between neighborhood N_g in the grayscale image and neighborhood N_s in the colorized image.

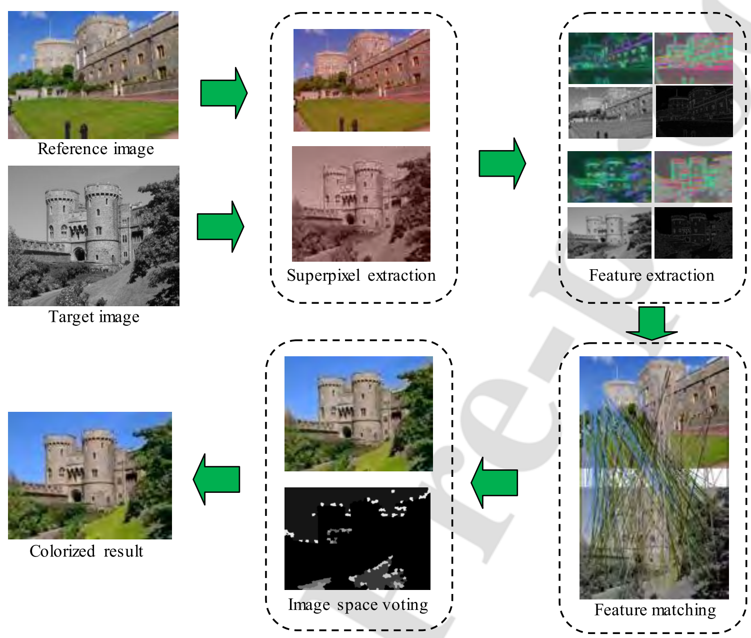

Figure 8: Overview of the colorization method based on matching of similar images [17]. Reproduced with permission from Ref. [17]. ©Association for Computing Machinery, Inc. 2012.

Gupta et al. [17] incorporated SIFT features [48] into Welsh's method [16] and created a new method that allows the transfer of color information from a reference image using multiple image features. Gupta et al. [17] use SIFT features to obtain the correspondence between the reference image and the target image for color transfer. So the reference image needs to be a related image in order to obtain the best results. They also proposed image voting for color correction, which checks neighboring superpixels to identify and then correct invalid color assignment, in order to keep the color consistency.

Unlike other methods based on matching between pixels [16, 17], the colorization method proposed by Ironi et al. [18] tries to color images at a higher semantic level.

Based on the method of Levin et al. [3], Ironi et al. [18] expect that the algorithm can

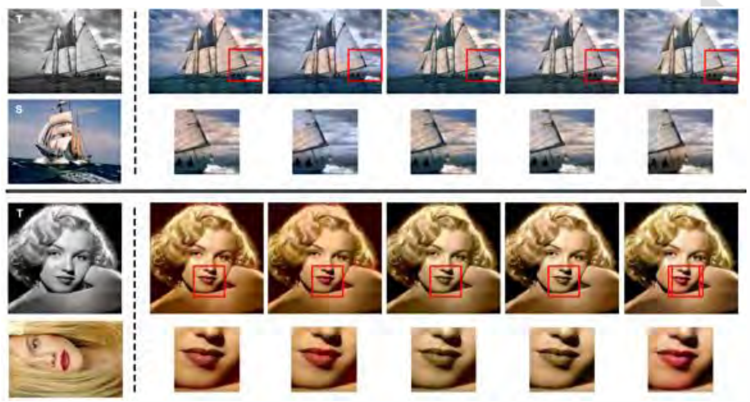

Figure 9: Comparison of the results of colorizing grayscale images. The “T” and “S” of the first column input images are the target and source images. (a) is the result of [19], (b) is the result of [17], (c) is the result of [16], (d) is the result of [49], and (e) is the result of [20]. ©Association for Computing Machinery, Inc. 2019.

automatically place color scribbles, and then use color optimization in [3]. Specifically, the method is mainly divided into four stages.They first train a supervised learning algorithm to build a low-dimensional feature space to discriminate which label the pixel belongs to. Then, they reliably determine the reference color value of each pixel by voting for the nearest neighbors in the feature space. Finally, the color is transferred to neighboring pixels in other spaces and the method of Levin et al. [3] is used for global optimization. Compared with scribbles, this method saves time, and adopts the spatial voting scheme to strengthen the spatial consistency, and has more robust color results than Welsh’s method [16].

Li et al. [50] proposed a new location-aware cross-scale texture matching method to achieve grayscale colorization based on reference images. This method first uses the multi-label graph-cut algorithm to minimize global matching errors and spatial scale variations, and then uses the statistics of up-down relationships in the reference image to correct unreasonable color matches, and finally applies an optimization framework to propagate the high-confidence micro-scribbles to entire image. In the grayscale colorization method based on the reference image, it is very common that the provided reference image and the target image scale are inconsistent, and this method can handle this situation well, and performs well among methods based on texture matching.

The total variation (TV) minimizing denoising model proposed by Rudin et al. [51] is used for image colorization. Kang et al. [52] proposed to use the total variation minimizing colorization model to deal with the problem of image color restoration. This method first minimizes the total variation, and then implements image colorization through weighted harmonic maps. However, this method requires a large number of color scribbles to process complex images. Further, Bugeau et al. [53] proposed a minimization variational formulation modeling which could colorize using a reference image. At the same time, a specific energy function is designed for modeling color selection and spatial consistency constraints. However, this method will produce a halo effect on edges with obvious texture contrast.

Fang et al. [19] proposed a grayscale colorization method based on a reference image, which novelly takes the result of image superpixel segmentation as the target to be processed. The method first uses the Vcells [54] algorithm to segment an image, extracts the features of the segmented blocks, and then uses the method proposed by Gupta et al. [17] to match the reference segmented feature and the target segmented feature. Different from Gupta's method [17], Fang et al. [19] do not use the matched colors as micro-scribbles for color propagation, but instead select a set of candidate colors for each target superpixel. Finally, they used the TV based spatial consistency regularization and non-local self-similarity regularization to determine the most suitable color for each target superpixel from the color candidates. As the comparison shows in Fig. 9, with the same reference image, Welsh et al. [16](c) and Pierre et al. [49](d) are limited by the set of color candidates and cannot match enough correct colors. He et al. [20](e) and Gupta et al. [17](b) obtained more reliable color assignment results, but the results of (b) contain color inconsistency artifacts and those of (e) contain color blurring and color bleeding that appear in tiny objects. Although (a) achieves better colorization results than other methods, there are still incorrect color matching results, such as the hair edges of the characters in the second row.

Instead of specifying reference images by users, these methods achieve automatic

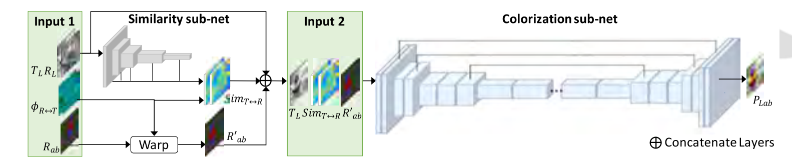

Figure 10: An overview of the architecture of colorization by Deep Exemplar [20]. Reproduced with permission from Ref. [20], ©Association for Computing Machinery, Inc. 2018.

grayscale colorization where the colorization algorithm automatically searched for similar color images on the Internet. Those images are often captured with different poses and illuminations. Given a specified reference image, Irony et al. [18] colorized images by a robust supervised classification scheme, but they cannot handle multiple reference images. Based on [18], Liu et al. [10] proposed an automatic colorization method which could directly search the Internet for multiple color images similar to the target image. It ignored the illumination difference between the grayscale target image and the color reference image. As shown in Fig. 7, after getting the images from Internet, the color reference images are registered by using the SIFT [55] matching algorithm. And then the target illumination, target reflectance, and intrinsic reflectance images of the target scene are extracted from these reference images. Finally, the color is transferred from the above images, and the final result is obtained by combining the illumination component of the target image. Morimoto et al. [56] also used web search for images with similar scene structure, and used the 20 most similar images for colorization. They then used color transfer based on the luminance value to colorize the input image.

The method of using similarity on luminance features has great dependency on the value of each pixel, and can solve the situation that the value of objects between different images is similar. However, when the lighting of the object or the structure changes, it is easy to get the wrong color. However, there is a correlation between the same objects or similar details. By introducing object category analysis or image feature analysis, etc., the color matching between the grayscale image and the reference image can be effectively improved.

3.3.2. Similarity with CNN Features

He et al. [20] proposed for the first time a fully automatic colorization method based on reference images, allowing users to use different reference images to achieve different colorization styles. Its network structure is mainly divided into a similarity sub-network and colorization sub-network, which is shown in Fig. 10. The similarity sub-network helps find the similarity maps between the reference image and the target image. Then the colorization sub-network mainly aligns pixels in the luminance channel based on the similarity sub-network, and then uses the big data learning ability of the colorization sub-network to refine misaligned pixel colors.

Inspired by He et al. [20], Zhang et al. [57] applied the method of deep exemplar-based colorization to grayscale video colorization. Similar to [20], Zhang et al. [57] obtained the dense correspondence between the target image feature and the reference image feature through computing a correlation matrix, and then feed it to a colorization sub-network. In this sub-network, the colorization result of the previous frame will be used as the condition for the current frame colorization. To reduce accumulated propagation errors, reference images are added. Through this recurrent framework, they achieved temporal consistency of video colorization. They further introduced a temporal consistency loss [58] to reduce the color change along the flow trajectory during the video colorization process. Further, Wu et al. [59] propose a method that can generate results with vivid colors by retrieving the matched features. Different from colorization based on reference images, they designed a GAN encoder to generate the color prior in colorization, which allows smooth interpolation between different colors and generates more diverse results.

Vondrick et al. [60] proposed video colorization with self-supervision learning of visual tracking. This method copies colors from a reference frame, and the model needs to use the appropriate region in order to obtain the correct color. And it can also be applied to track people's movement through the video. This model uses a pointing mechanism to solve the problem of inconsistencies in video colorization to a large extent, and maintain the color stability of the frame. However, the pointing mechanism is still not precise enough, and the color edges of the colorization result are sometimes

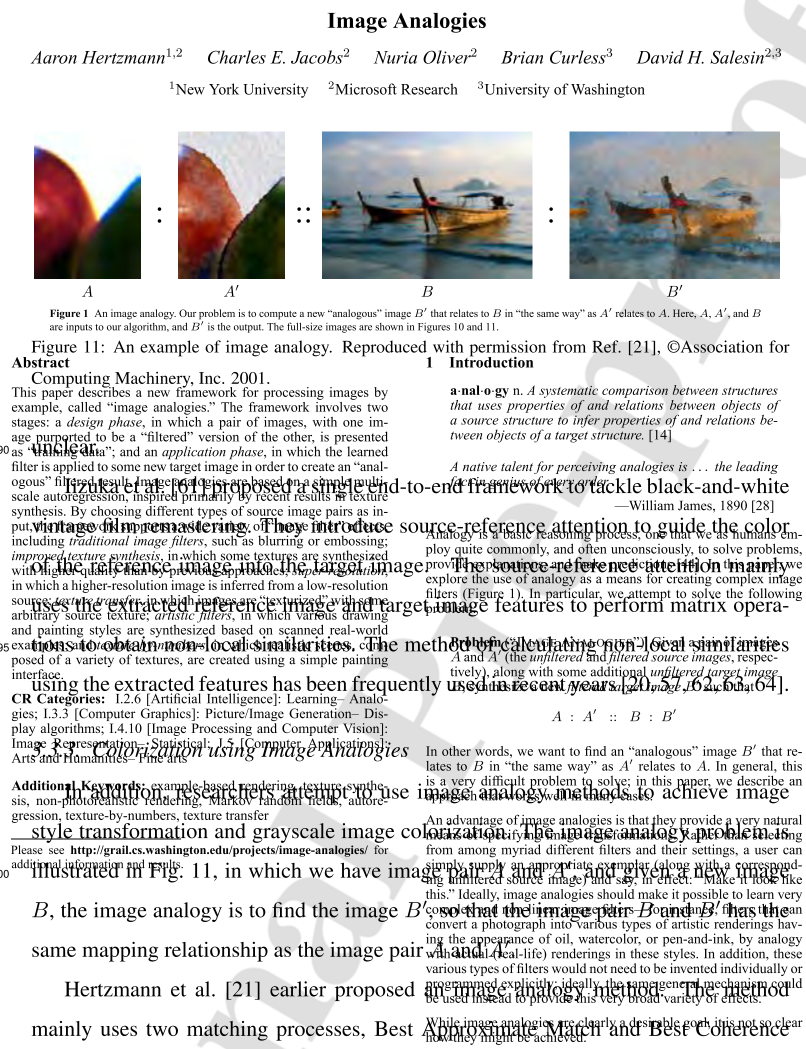

Image Analogies

Aaron Hertzmann[1,2] *Charles E. Jacobs*[2] *Nuria Oliver*[2] *Brian Curless*[3] *David H. Salesin*[2,3]

[1]New York University [2]Microsoft Research [3]University of Washington

A : A' :: B : B'

Figure 1 An image analogy. Our problem is to compute a new "analogous" image B' that relates to B in "the same way" as A' relates to A. Here, A, A', and B are inputs to our algorithm, and B' is the output. The full-size images are shown in Figures 10 and 11.

Abstract

This paper describes a new framework for processing images by example, called "image analogies." The framework involves two stages: a *design phase*, in which a pair of images, with one image purported to be a "filtered" version of the other, is presented as "training data"; and an *application phase*, in which the learned filter is applied to some new target image in order to create an "analogous" filtered result. Image analogies are based on a simple multi-scale autoregression, inspired primarily by recent results in texture synthesis. By choosing different types of source image pairs as input, the framework supports a wide variety of "image filter" effects, including *traditional image filters*, such as blurring or embossing; *improved texture synthesis*, in which some textures are synthesized with higher quality than by previous approaches; *super-resolution*, in which a higher-resolution image is inferred from a low-resolution source; *texture transfer*, in which images are "texturized" with some arbitrary source texture; *artistic filters*, in which various drawing and painting styles are synthesized based on scanned real-world examples; and *texture-by-numbers*, in which realistic scenes, composed of a variety of textures, are created using a simple painting interface.

CR Categories: I.2.6 [Artificial Intelligence]: Learning– Analogies; I.3.3 [Computer Graphics]: Picture/Image Generation– Display algorithms; I.4.10 [Image Processing and Computer Vision]: Image Representation– Statistical; J.5 [Computer Applications]: Arts and Humanities– Fine arts

Additional Keywords: example-based rendering, texture synthesis, non-photorealistic rendering, Markov random fields, autoregression, texture-by-numbers, texture transfer

Please see **http://grail.cs.washington.edu/projects/image-analogies/** for additional information and results.

1 Introduction

a·nal·o·gy n. *A systematic comparison between structures that uses properties of and relations between objects of a source structure to infer properties of and relations between objects of a target structure.* [14]

A native talent for perceiving analogies is … the leading fact in genius of every order.

—William James, 1890 [28]

Analogy is a basic reasoning process, one that we as humans employ quite commonly, and often unconsciously, to solve problems, provide explanations, and make predictions [44]. In this paper, we explore the use of analogy as a means for creating complex image filters (Figure 1). In particular, we attempt to solve the following problem:

Problem ("IMAGE ANALOGIES"): Given a pair of images A and A' (the *unfiltered* and *filtered source images*, respectively), along with some additional *unfiltered target image* B, synthesize a new *filtered target image* B' such that

$$A : A' :: B : B'$$

In other words, we want to find an "analogous" image B' that relates to B in "the same way" as A' relates to A. In general, this is a very difficult problem to solve; in this paper, we describe an approach that works well in many cases.

An advantage of image analogies is that they provide a very natural means of specifying image transformations. Rather than selecting from among myriad different filters and their settings, a user can simply supply an appropriate exemplar (along with a corresponding unfiltered source image) and say, in effect: "Make it look like this." Ideally, image analogies should make it possible to learn very complex and non-linear image filters—for instance, filters that can convert a photograph into various types of artistic renderings having the appearance of oil, watercolor, or pen-and-ink, by analogy with actual (real-life) renderings in these styles. In addition, these various types of filters would not need to be invented individually or programmed explicitly; ideally, the same general mechanism could be used instead to provide this very broad variety of effects.

While image analogies are clearly a desirable goal, it is not so clear how they might be achieved.

Figure 11: An example of image analogy. Reproduced with permission from Ref. [21], ©Association for Computing Machinery, Inc. 2001.

unclear.

Iizuka et al. [61] proposed a single end-to-end framework to tackle black-and-white vintage film remastering. They introduce source-reference attention to guide the color of the reference image into the target image. The source-reference attention mainly uses the extracted reference image and target image features to perform matrix operations to obtain non-local similarities. The method of calculating non-local similarities using the extracted features has been frequently used in recent years [20, 57, 62, 63, 64].

3.3.3. Colorization using Image Analogies

In addition, researchers attempt to use image analogy methods to achieve image style transformation and grayscale image colorization. The image analogy problem is illustrated in Fig. 11, in which we have image pair A and A', and given a new image B, the image analogy is to find the image B', so that the image pair B and B' has the same mapping relationship as the image pair A and A'.

Hertzmann et al. [21] earlier proposed an image analogy method. The method mainly uses two matching processes, Best Approximate Match and Best Coherence Match. The Best Approximate Match process first uses a Gaussian pyramid to extract the feature information of pixels at different scales, and then uses an approximate-nearest-neighbor (ANN) search to search for the pixel p in the original image A that best matches each pixel q in the target image B. The Best Coherence Match guarantees the spatial consistency of the matching results. This calculation method is mainly derived from the method proposed by Ashikhmin et al. [65]. The specific calculation is as follows:

$$r^{\star} = \underset{r \in N(q)}{\arg\min} ||F_{\ell}(s(r) + (q - r)) - F_{\ell}(q)||^2 \tag{1}$$

where r represents a pixel that has been synthesized in the neighborhood of pixel q in B', $s(r)$ represents the pixel corresponding to r in A', $N(q)$ represents a pixel synthesized in the neighborhood of pixel q, and $F_{\ell}(\cdot)$ represents a neighborhood feature vector of pixel in layer l.

Although the algorithm proposed by Hertzmann et al. [21] can get good results, the algorithm requires pixel-by-pixel matching, which is particularly slow. Later, Liao et al. [22] proposed a new image analogy method using deep learning technology, which has greatly improved the matching speed and effect. The method uses the pre-trained image feature extraction network VGG-19 [40] to extract the 5-layer high-dimensional features of images A and B'. Then the method uses Nearest-Neighbor Field Search (NNFs) to find dense correspondences with bidirectional constraints in each feature layer. Finally, the image features are gradually reconstructed from the roughly corresponding fifth layer to the finely corresponding first layer, and the final generated images A' and B are obtained. The specific pipeline of the method system is shown in Fig. 12.

There are other researchers who use the idea of image analogy. For example, Bénard et al. [66] realized the stylization of animation by extending the image analogy method to create a time-continuous animation sequence. Using the image analogy method, Jamriška et al. [67] used image color, foreground object binary mask, position of SIFT Flow (Scale-invariant feature transform Flow) [48], and foreground object edge information as guidance information, and achieved video stylization.

Affected by the structure of the image and the color range of the image, when the pose and appearance of objects in the image change greatly, it is impossible to obtain good results by using the image analogy method, even if the two images to be compared are images from the same video at different times. In addition, if the color of a part of the new image does not appear in the original image, the algorithm cannot automatically fill this part of the color. At the same time, when the method is

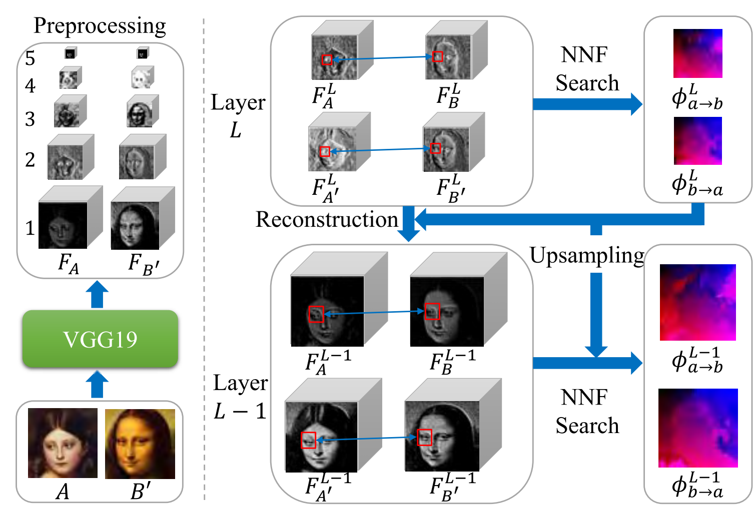

Figure 12: System pipeline of deep image analogy [22]. Reproduced with permission from Ref. [22], ©Association for Computing Machinery, Inc. 2017.

Fig. 4. System pipeline.

applied to sketches and color images, it is difficult to generate reasonable color results corresponding to sketches.

3.4. *Text-based Colorization*

Text can be used to describe the color of objects, the color palette of images, etc. By analyzing the semantics information of color contained in the text, it can be used as a constraint to implement text-based coloring tasks. Using color semantics can effectively improve the color selection in the colorization process. The study [68] establishes upon the image segmentation from natural language expression, which correctly identifies and divides the specified content out of the whole image. Hu's method [68] is based on CNN and Long Short-Term Memory (LSTM) network to process the image and linguistic information, and can be trained end-to-end. Manjunatha et al. [23] were the first to present a language-based colorization method with the help of natural language on Oxford-102 Flowers dataset. They also use the LSTM network to transform the word into a vector through a word matrix and produces a contextual vector. And

together with feature maps are used to reconstruct features of latent images A'/B at the next CNN layer (Section 4.3). The NNFs obtained at the current layer are further upsampled to the next layer as their initialization (Section 4.4). These three steps (NNF search, latent image reconstruction and NNF upsampling) are repeated at each layer, updating correspondences from coarse to fine.

4.1 Preprocessing

Our algorithm starts with precomputing feature maps by a VGG-19 network [Simonyan and Zisserman 2014] that is trained on the ImageNet database [Russakovsky et al. 2015] for object recognition. We obtain the pyramid of feature maps $\{F_A^L\}$ and $\{F_{B'}^L\}$ $(L = 1...5)$ for the input images A and B'. The feature map of each layer is extracted from the *reluL_1* layer. It is a 3D tensor with *width* × *height* × *channel*, and its spatial resolution increases from $L = 5$ to 1, as shown on Fig. 4(*left*).

The features of A' and B are unknown. We estimate them in a coarse-to-fine manner, which needs a good initialization at the coarsest layer ($L = 5$). Here, we let $F_{A'}^5 = F_A^5$ and $F_B^5 = F_{B'}^5$ initially, that is, we view A and A' (also B and B') to be very similar at the top layer where the images have been transformed by the CNN into representations with the actual content, but being invariant to precise appearance (as shown in Fig. 3).

4.2 Nearest-neighbor Field Search

At layer L, we estimate a forward NNF and a reverse NNF; they are represented by mapping functions $\phi^L_{a\to b}$ and $\phi^L_{b\to a}$, respectively.

then the recurrent attentive model combines the image and language features. Finally, get the colorized results from this feature.

Unlike Chen et al. [23] who directly combine the extracted language features and visual features to control the colorization results, Manjunatha et al. [69] apply the feature-wise linear modulation (FILM) [70] structure to language-based colorization with fewer parameters. Since FILM performs feature affine transformation on the output of each convolutional block, only two additional weight matrix parameters are required for each feature map. Instead of putting a specific object into the color form, they use semantic text input to generate the color palette to achieve color palette based on user's text input.

Bahng et al. [35] have a similar focus on this area, instead of putting a specific object into the color form, they use semantic text input to generate the color palette to achieve color palette based on user's text input. This is based on previous related work on color palette design and image editing such as research by Heer et al. [71]. Based on Bahng's model, the user can use both single and multi-word descriptions to create a color palette and colorize the grayscale image. Like Hu's method [68], the network is based on conditional generative adversarial networks (cGANs). Bahng used the palette-and-text (PAT) data set to train the model for predicting color palette parts. The data set contains 10,183 text and five-color palette pairs. The data set is refined by harvesting user custom-made color palettes from community websites. To process raw data from the data set, they use four annotators to vote whether the semantic word matches the color palette.

Text2Color can be divided into two parts, a Text-to-Palette Generation Network (TPN) and Palette-based Colorization Network (PCN). TPN generates a reasonable color palette based on the text input. The objective for the first cGAN is expressed as:

$$\begin{aligned} L_{D_0} = & \mathbb{E}_{y\sim P_{data}}\left[\log D_0(\overline{c}, y)\right] \\ & + \mathbb{E}_{x\sim P_{data}}\left[\log\left(1 - D_0(\overline{c}, \hat{y})\right)\right] \end{aligned} \tag{2}$$

$$L_{G_0} = \mathbb{E}_{x\sim P_{\text{data}}}\left[\log\left(1 - D_0(\overline{c}, \hat{y})\right)\right] \tag{3}$$

For the discriminator D_0 wants to maximize L_{D_0} in opposition to G_0 which wants to minimize L_{G_0}. Vector x and real color palette y are from the data distribution P_{data}.

Bahng found that the Hubor loss is the most effective way to increase color diversity in the color palette. They decide to use Hubor loss to make the generated image closer to ground truth, and they added a Kullback-Leiber divergence regulation term. In this part, they use a conditioning augmentation technique.

$$\hat{y}_i = f\left(s_i\right) \text{ where } s_i = g\left(\hat{y}_{i-1}, c_i, s_{i-1}\right) \tag{4}$$

where s_i is a GRU (Gated Recurrent Unit) hidden state, and i is a time vector, previous generated colors are stored in $\hat{y}_{i-1}$ and the content vector c_i and the previous state stored in s_{i-1} are provided as input. This state is used as input to a fully-connected layer to output the ith color into the palette, and results are the combination of five colors to form a single palette output $\hat{y}$.

The text-based method only needs to use the text description to realize the image colorization work, which can be used not only for the colorization of a single image, but also for the colorization of multiple images or videos. However, the text description has certain limitations on the specification of details and the selection of the color range, so it is more suitable for controlling the color palette of the whole image and the color of single object. The enhancement of detail colors can increase the control of detail by introducing color strokes, and researchers can conduct research on multi-modal colorization methods to integrate the advantages of different models.

4. Colorization based on Sketch Images

Sketch images consist of sparse lines, and the information in the images is sparse compared to grayscale images. Colorization of grayscale images tends to use grayscale information from the L channel in Lab space, and it is easier to judge the same semantic area by pixel values, so it is difficult to directly apply to sketch images. Usually, sketch colorization methods are mostly sample-based or need users to provide guidance information, and contain both and automatic colorization option and an interactive user mode, since the input images do not carry texture information. In early research, color hints provided by the user are spread to the entire image, but these colorization methods are limited by the quality of sketch images, the richness of color information, and values of method parameters. In the last five years, most popular models are based on neural networks, such as CNN [8], GANs [9] and U-net architecture [72], which can replace the manual effort of carrying out colorization and can make monochrome images more attractive. In addition, we introduce a special sketch colorization method in the last subsection, which directly generates pictures based on the input sketch images.

4.1. Color Hints or Strokes based Colorization

In an early paper in 2009, Sỳkora et al. [24] described the LazyBrush colorization model which needs users to carefully make color scribbles. They transform the coloring problem of sketch images into an optimization problem, and design an energy function which consists of two main terms: smoothness and data. The smoothness term mainly ensures to hide color discontinuity, and the data term mainly considers the color prompt information added by the user. Although the LazyBrush algorithm performs well in interactive colorization, a long time is needed to calculate the iteration. In order to reduce the algorithmic complexity, Sébastien et al. [73] focus on line art images and do not process black-and-white cartoon or manga, and performs fine analysis on the local geometry of stroke contours. Compared with LazyBrush, their CPU calculation time is reduced by more than 70% in different sizes of test images. Although the algorithm based on optimization is robust and has a high success rate for colorization, methods based on deep networks have natural advantages in reducing user interaction and colorization speed.

In the method based on deep networks, the GAN usually is used a generative model to colorize sketch images, and U-net [72] are instead of the traditional encoder-decoder structure as the GAN's generator. Liu et al. [74] used conditional generative adversarial networks (cGANs) to train the automatic painter model to produce compatible colors for a sketch. Moreover, their architecture allows users to control the color of generated images, which is based on [75] to add color strokes to the input sketch image. Ci et al. [76] proposed a novel conditional adversarial synthesis architecture, which combined with a local feature network. The main branch of the generator is developed based on U-net [72] and four sub-networks, each containing a convolution block to fuse features from skip connection. And the local feature extracted from local feature network is as conditional input of generator and discriminator, to avoid overfitting characteristic of the line art, to help the generator to produce vivid colorful color outputs.

In addition to the end-to-end colorization network, Zhang et al. [26] proposed a two-stage line art colorization network based on color hints (known as style to paint version V3). Each stage of the network consists of a generator based on the U-Net structure and a discriminator, where the color hints marked by the user in the two stages are directly inputted into the network together with the line art image. It is worth noting that the second-stage network uses Inception V1 [77] to extract the features of the simulated color draft generated by the first-stage network and merge them into the intermediate features of the generator. This method can get a good colorization result by adding color hints, but each image requires a lot of user input as color hints.

Moreover, a commercial website PaintsChainer [25] also colorizes sketches based on color hints, which is an automatic colorization model for sketch images. The model provides three different painting styles for the user to choose from, and will produce different results based on the same sketch image. This product is user-friendly for the non-programmer artist and can process sketch images through their online web page with no need to download any software.

Color hints and strokes provide a more convenient interactive tool to support the user to specify the color of the local details of the image. For the colorization of a single image, such methods allow the user to iteratively optimize and produce the desired result. If used on multiple images and videos, there is still a lot of interaction. However, when artists color comics, they often design the clothing and color matching of the characters in advance, and color the content according to the preset image, which has inspired some colorization work based on reference images.

4.2. Reference Color Image based Colorization

Different from color hints based colorization methods, reference based colorization methods match segment shapes or even semantic similarity between reference images and input images to colorize different positions. We divide reference based colorization methods into deep networks and graph correspondence. In the deep network based colorization methods, the color of result only takes the reference image as conditional information, and does not completely depend on it. However, the method based on graph correspondence usually copies the color value of reference images, and color value of result images completely depends on the reference image.

4.2.1. Colorization by Deep Networks

A series of research work of Style to Paint is an important work of applying deep network to color line art images. In style to paint version V1 [78], the model can color sketch images based on the input reference images. The main idea is to use U-net and AC-GAN [79] in a generator for the style image. They redesigned the new network residual U-net [72], and initialized the network with a Gaussian random number. The network shows a stable gradient in the training process and to solve potential noise he adds two additional losses to avoid the vanishing gradient problem. In style to paint version V3 [26], as introduced in Section 4.1, they proposed a two-stage CNN-based colorization framework. Instead of using the reference image, in this model users use the color hint mark to provide colors for the sketch image. In the same year, sun et al. [80] designed a dual conditional generative adversarial network for the colorization problem in icon design. The model can get a good colorization result on simple strokes such as icons, but color bleeding still appears sometimes, and it cannot handle complex sketch images well.

In the work of colorizing line art videos, the videos are usually divided into multiple video sequences. Researchers usually provide one or more reference color images for each video sequence to color the remaining line art frames. Thasarathan et al. [81] proposed a line art video colorization model called automatic Temporally Coherent Video Colorization (TCVC), which extends the image-to-image translation model based on the conditional GAN [82]. They input the line art image and the color image of the previous frame into the generator network for providing color prior information. The discriminator network uses the patch GAN structure proposed by Isola et al. [82], and inputs the line art image and the corresponding color image at the same time to ensure the temporal color consistency.

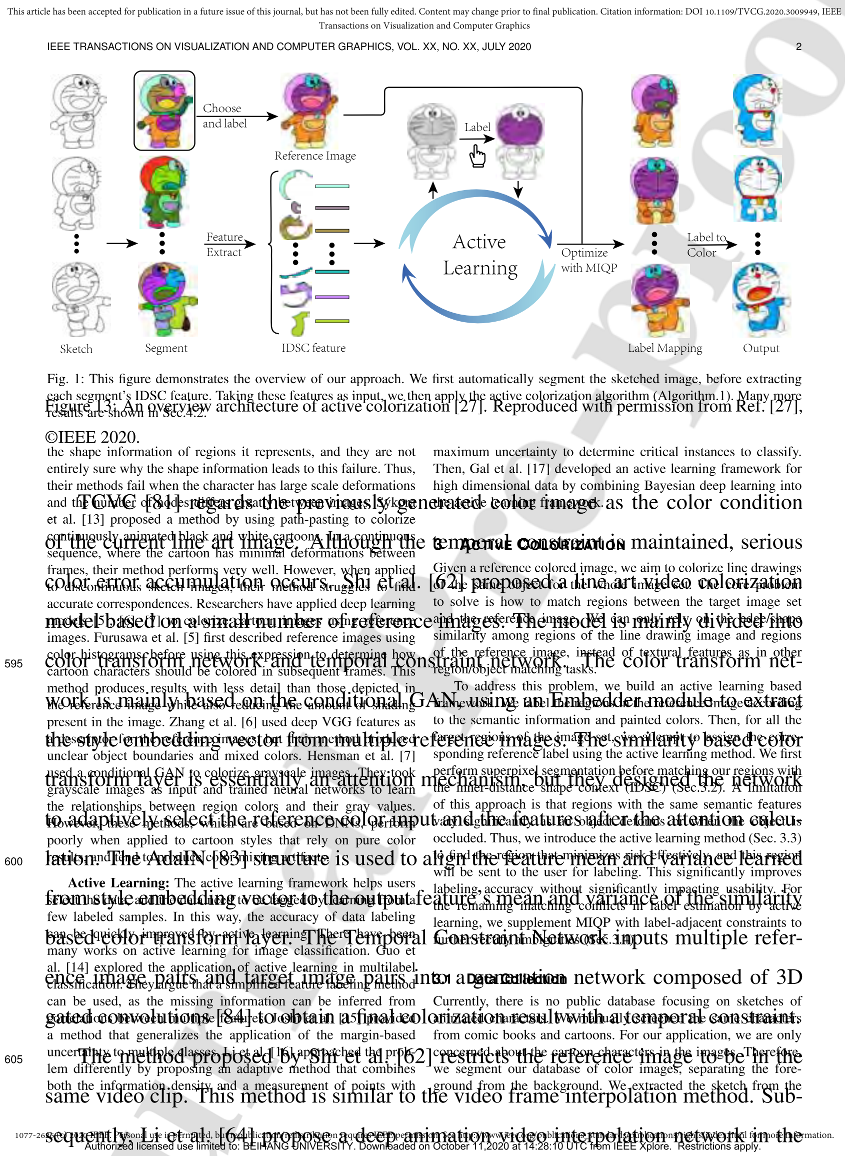

Figure 13: An overview architecture of active colorization [27]. Reproduced with permission from Ref. [27], ©IEEE 2020.

TCVC [81] regards the previously generated color image as the color condition of the current line art image. Although the temporal constraint is maintained, serious color error accumulation occurs. Shi et al. [62] proposed a line art video colorization model based on a small number of reference images. The model is mainly divided into color transform network and temporal constraint network. The color transform network is mainly based on the conditional GAN, using an Embedder module to extract the style embedding vector from multiple reference images. The similarity based color transform layer is essentially an attention mechanism, but they designed the network to adaptively select the reference color input and the features after the attention calculation. The AdaIN [83] structure is used to align the feature mean and variance learned from style embedding vector to the output feature's mean and variance of the similarity based color transform layer. The Temporal Constraint Network inputs multiple reference image pairs and target image pairs into a generation network composed of 3D gated convolutions [84] to obtain a final colorization result with temporal constraint.

The method proposed by Shi et al. [62] restricts the reference image to be in the same video clip. This method is similar to the video frame interpolation method. Subsequently, Li et al. [64] propose a deep animation video interpolation network in the wild. The network also uses high-dimensional features extracted from input images to calculate a similarity matrix and learn the optical flow correspondence between adjacent frames. Compared with Shi et al. [62] directly extracting line art image features, Li

Figure 14: The comparison of reference based colorization method. The leftmost image is the sketch image. (a) is the reference image in different color styles. (b) is the result of [26]. (c) is the result of [86]. (d) is the result of [4] and (e) is the result of [27].

et al. [64] first extract contours which are filled with the "trapped-ball" algorithm [85] to generate color patches, and then are fed into the VGG-19 model to generate color patches features. In addition, after obtaining the bidirectional optical flow correspondence, a recurrent flow refinement network are proposed to optimize the final optical flow through multiple iterations of learning.

Given an arbitrary reference image, such methods can generate color images. Obvious artifacts such as color matching errors and color mixing are prone to occur, especially when the fill is a pure color, as shown in Fig. 14(b). Researchers can consider adding region segmentation information to optimize the colorization results.

4.2.2. Colorization by Graph Correspondence

Since a line art image can be divided into different regions according to lines, the researcher can quickly transfer color from the reference image using the image matching method. Sato et al. [4] proposed a colorization method for line art images based on image matching. First, the method uses the super-pixel method [87] to divide the reference image and the target image into different regions, and then the different regions are used as nodes to compose the graph structure. Then the method uses the area of the node and the length/angle of the vector composed of the centroid of different nodes to calculate the similarity between the two nodes. Finally, according to the reference image and the target image represented by the graph structure, they obtained the final matching result by solving a Quadratic Programming problem, and transferred the color of the reference image to the target image.

The method proposed by Sato et al. [4] needs to specify the number of segmentation regions, and when the lines of the line art image are very complicated, the user needs to manually adjust the segmentation results. In addition, the method calculates the pairwise similarity of nodes, and does not make full use of the shape features of the segmented regions themselves. Chen et al. [27] proposed a colorization method combined with active learning [88]. The overall pipeline of the method is shown in Fig. 13. First, the trapped-ball segmentation [85] was used to automatically segment the line art image, and inner-distance shape context (IDSC) [89] is used to extract the features of the segmented regions. And then, the features of the segmented regions of the reference image and the adjacency relationship between the regions are expressed as a graph structure, and finally the mixed-integer quadratic programming method (MIQP) is used to solve the graph matching problem. As the comparison shows in Figs. 14(d)(e), with the same reference image, Sato et al. [4](d) failed when the number and structure of regions changed. Chen et al. [27](e) produce better results, however, there are still colorization errors in smaller segmented regions (such as the left side of the face).

The colorization method based on the graph correspondence needs to segment the image into sub-regions in advance, and construct the graph structure according to the adjacency relationship between sub-regions. Some methods introduce constraints such as shape similarity during optimization. But such methods suffer from segmentation mistakes and are only suitable for some simple-structured images. When the number of image areas is too large and the structure changes, the effect is slightly worse, such as the case where there are too many lines in the hair area.

4.3. Text Hints based Method

Text based colorization methods not only learn the correspondence between the color and the semantic regions of images, but also the correspondence between the text and the color needs to be determined. Kim [90] purposed using text tags to colorize sketch images. Users can easily color line art images based on their input color variant tag (CVT) from which the generator produces a color result. The CVT module first extracts the features of input text information, and then merges the features to the output features through the SECat (Squeeze and Excitation with Concatenation) module.

For the first time, Zou et al. [28] proposed a language-based interactive colorization system for scene sketches. Users can color the foreground objects and background sequentially through text scripts. They proposed a new instance matching model, which uses the DeepLab-v2 [91] network to extract sketch features, and the interactive model mLSTM (multimodal Long Short-Term Memory) [92] is also added to the generator network, which can achieve the joint modeling of text script and images. Zou et al. [28] also designed a foreground colorization network and a background colorization network to facilitate the processing of foreground objects and background areas with different image characteristics. It is worth noting that since the current research on semantic understanding in natural language processing is still in the preliminary stage, the input text of text based colorization methods is more similar to a text control instruction, and more accurate text understanding and color matching are still Need further research.

4.4. Sketch to Image Synthesis

The synthesis of sketch to image is a kind of research related to the colorization of line art images. Different from the line art colorization, the sketch to image synthesis method does not strictly provide color values at different positions, but extracts the semantic features of the lines in sketch image and compares them with the existing images in the data set. After that, a new image is synthesized by fusing different matching results. Therefore, the method of sketch to image usually requires an amount of data to be matched.

4.4.1. Image synthesis by Internet Search

Chen et al. [93] proposed a system Sketch2Photo that can automatically synthesize realistic images from sketches with text labels. Sketches will be divided into background and multiple scene items, and then match the initial set of candidate images on the Internet based on text labels and lines. For the background image, the candidate images with inconsistent content and cluttered regions are removed. Inspired by Ben-Haim et al. [94], they used a clustering algorithm to filter the content consistency of the image, and count the number of regions covered by the convex hull of all scene items to determine whether the image is uncluttered or not. And then, after discarding candidate background images with salient areas and complex backgrounds, they use the grab-cut algorithm [95] to segment the expanded areas of the target object from the image. According to the shape feature and the clustering algorithm, the images with inconsistent shapes and inconsistent content are eliminated. Finally, they used the proposed hybrid method to fuse the background and scene item candidate images.

Later, Chen et al. [96] proposed a method that can quickly build a large-scale human image database. Inspired by Sketch2Photo [93], they use a human detection algorithm [97] to extract images containing humans from the Internet. Then they filter out the algorithm-unfriendly images, and segment the foreground and background of the image. The final human image is organized by action semantics and clothes attributes, and the user can retrieve images of the corresponding posture through the outline. In addition, Chen et al. [96] demonstrated the use of this data to generate multi-frame personalized content image synthesis programs.

By extracting and assembling content from existing data, reasonable image results can be obtained. However, since the generated results depend on the existing data content, the image content outside the dataset cannot be generated. The emergence of generative networks solves this problem, training on existing datasets to generate new results in addition to existing ones. Therefore, the methods based on generative networks for image synthesis have become more popular.

4.4.2. Image synthesis by Generative Networks

Sangkloy et al. [75] proposed a sketch-to-photo architecture, similar to an image-to-image translation network [98]. They are the first to use a feed-forward architecture model that can generate realistic images based on imperfect sketches. The architecture presented colorization in three different domains: faces, bedrooms and cars. Chen et al. [99] proposed a fully automatic model to synthesize realistic images from humandrawn sketches. The model takes the GAN model as the basic structure and proposes the Mask Residual Unit (MRU). The convolutional features of the previous layer and an additional image are fed into the MRU, then the MRU dynamically determines the final output features of the network by calculating an internal mask. This is conducive to generating results that are similar in content information to the input image and generating higher quality results.

Different from the previous image synthesis methods based on sketches and colors [93, 98], Xian et al. [100] proposed an image synthesis method based on sketches and textures. This method realizes the fine-grained control of the synthesized image based on the texture image. For this purpose, they mainly design two GAN network structures to produce preliminary synthesis results and fine-tuned textures of synthesized images respectively. In addition, a novel local texture loss $\mathcal{L}_t$ was proposed. The loss function first randomly samples n patches of $s \times s$ size from the generated result $G(x)$ and the input texture image I_t, and then calculates its Local Adversarial Loss $\mathcal{L}_{adv}$, Local Style Loss $\mathcal{L}_s$ and Pixel Loss $\mathcal{L}_p$. The specific calculation is as follows:

$$\mathcal{L}_t = \mathcal{L}_s + w_p\mathcal{L}_p + w_{adv}\mathcal{L}_{adv} \tag{5}$$

where $\mathcal{L}_s$ use Gram matrix-based style loss and $\mathcal{L}_p$ use L2 pixel loss. $\mathcal{L}_{adv}$ is defined as follows:

$$\mathcal{L}_{adv} = -\sum_i (D_{txt}(h(G(x_i), R_i), h(I_t, R_i)) - 1)^2 \tag{6}$$

where D_{txt} is a local texture discriminator, $h(x, R)$ represents cropping a patch from the segmentation mask R of image x.

Inspired by [101], Chen et al. [102] proposed a local-global network for sketch-based image synthesis. They take the facial structure into consideration and design a manifold projection to deal with rough/in-complete sketch. For easier use by ordinary users with little drawing skill, they design an interface with shadow-guided on the drawing board like ShadowDraw [103]. They further proposed a disentanglement framework [104] which could disentangle the geometry and the appearance features from facial images. With this framework, it is possible to edit the appearance by

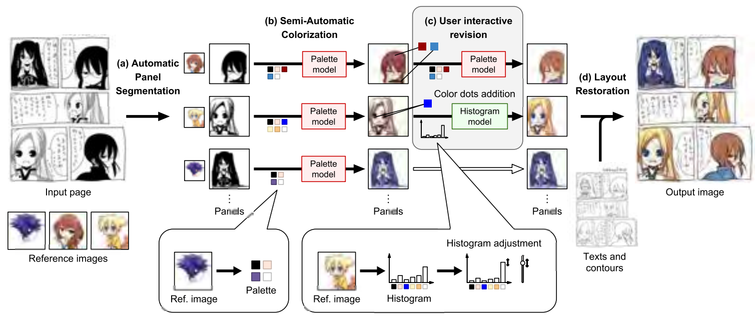

Figure 15: The overall pipeline of the semi-automatic manga colorization [30]. Reproduced with permission from Ref. [30], ©Association for Computing Machinery, Inc. 2017.

changing the appearance image and edit the geometry by using a sketch, for example, adding wrinkles or changing the color of hair.

The image synthesis method based on generative networks provides a new solution for sketch-to-image synthesis. However, due to the variety of styles of hand-drawn sketches by users, how to map sketches to reasonable results is a challenging research problem. Analysis of the semantic information of the lines in the sketch can give guidance when generating the image, thereby improving the correspondence between the image and the sketch.

5. Hybrid Method - Cartoon and Manga Image Colorization

In this Section, we will introduce some hybrid methods, for which the input is a black-and-white cartoon or manga. Those images contain a clear sketch-based line art, and also carry some black-and-white shading information. In manga, the colorization process has an additional pattern colorization process. Colorizing black and white manga can make them more attractive, and using semi-automatic methods as a helper can increase the speed of drawing.

Early work by Sỳkora et al. [105] carried out research on the colorization of black-and-white cartoons. They divided each frame into two parts: the static background and dynamic foreground, which were colorized separately. To separate frames into two parts, they used the Laplacian-of-Gaussian mask to detect outlines. When realizing continuous frame colorization, they used a probabilistic reasoning scheme to calculate the similar region and neighborhood relationship to process more video frames. Finally, to improve the quality they added color modulation, composition, and a technique to remove dust spots in order to improve the appearance of the final image. In this way, the image edges are sharp and can adapt to complicated drawing. The whole process is semi-automatic.

For manga colorization, Qu et al. [29] proposed a manga colorization method based on user scribbles. In this model, they use a novel texture-based level set method for segmentation. The model provides two modes of propagation for segmentation, pattern-continuous and intensity continuous propagation. Users can easily alter those two methods in different steps. The model can identify the hatching and screening effects that are used in traditional paper comics. After confirming the segment region, they use three different colorization methods for various conditions, such as color replacement, stroke-preserving colorization, and pattern to shading.

Hensman et al. [86] used cGAN and post-processing to colorize manga images, reducing the degree of user interaction and producing better results. This method can be divided into the following steps: screentone removal, segmentation, color selection, saturation increase, color quantization, and generation of shading. Model training is based on the corresponding grey-scale image and colorized reference image as a single image pair. Based on the model parameters obtained from the training of a single image pair, the model can colorize manga images similar to the reference image. However, when the character's clothing becomes complex, the model cannot achieve the correct correspondences between different frames. The results will not only contain artifacts, but also are not as colorful as the reference image, as shown in Fig. 14(c).

Hiroshiwa et al. [30] connected the common manga colorization problem and was able to color the same character at once. The overall pipeline of the semi-automatic manga colorization method is shown in Fig. 15. Users need to provide the reference image and the corresponding color image together as input. A segmentation step is based on [106], and semi-automatic colorization is based on the CNN architecture, using an encoder-decoder network with some refinement to improve the performance.

After the palette model, the model produces the draft colorization result, and the user needs to carry out interactive revision of the details to ensure the correctness of the output result. During revision users can choose either color dots or a histogram to adjust the image.

Sketch images contain sparse lines, and grayscale images represent image information such as shading and texture. The manga image may contain one or more styles, such as region boundary, narrow structure, specific region representations, etc. This leads to a conflict in the understanding of image content. Distinguishing the texture of an object or the specific representation in the image is the key to improving the colorization of the manga.

6. Assessment of Colorization

Researchers typically evaluate different colorization methods based on quantitative and qualitative aspects. In quantitative evaluation, researchers assume the existence of unique ground truth and provide unique ground truth, which facilitates simple analysis of colorization results. Then researchers typically use root mean square error (RMSE) [37, 39], peak signal to noise ratio (PSNR) [39, 1, 32], or structural similarity index measurement (SSIM) [32, 27, 62] image evaluation indicators, comparing the results of different colorization methods.

However, the widely used indicators such as PSNR and SSIM do not fully match human perception. In recent years, researchers have generally extracted the deep features of images to compare the perception similarity between images. A common measurement is to use the Inception Score (IS) [107] to evaluate the quality of the generated images. Given a desired image set, we could use the Frechet Inception Distance score (FID) [108, 45] to measure the difference between the distribution of the real images and the distribution of the generated images. Zhang et al. [109] systematically analyzed the unreasonable effectiveness of deep features as a perceptual metric, and proposed the Learned Perceptual Image Patch Similarity (LPIPS) metric which better captures image perception similarity. LPIPS has been used to evaluate colorization by Yoo et al. [13], and this has been shown to be an effective colorization evaluation mea-

Input Category	Paper	Year	Condition	Paper	Year	Condition
Gray-scale image	[21]	2001	R	[16]	2002	R
	[3]	2004	S	[46]	2005	R S H
	[18]	2005	R	[47]	2006	S
	[14]	2007	S	[10]	2008	A R
	[56]	2009	A R	[17]	2012	R
	[1]	2015	A	[37]	2015	A
	[11]	2016	A	[39]	2016	A
	[2]	2016	A	[22]	2017	R
	[15]	2017	H	[42]	2017	A
	[23]	2018	H	[69]	2018	H
	[35]	2018	H	[32]	2018	A
	[60]	2018	R	[20]	2018	R
	[67]	2019	R	[57]	2019	R
	[43]	2019	A	[13]	2019	A
	[50]	2019	R	[61]	2019	R
	[19]	2020	R	[12]	2020	A R
Sketch image	[24]	2009	S	[93]	2009	R
	[96]	2013	R	[4]	2014	R
	[25]	2017	A S	[78]	2017	R
	[74]	2017	A S	[75]	2017	S T
	[100]	2018	R	[26]	2018	S H
	[76]	2018	A S	[90]	2019	H T
	[28]	2019	T	[80]	2019	R
	[81]	2019	R	[62]	2020	R
	[27]	2020	R	[102]	2020	A
	[104]	2021	A R	[64]	2021	A R
Manga image	[105]	2005	R	[29]	2006	S
	[30]	2017	R H	[86]	2017	A

Table 1: Summary for different categories of image colorization method. A marks fully-automatic colorization. R, S, H and T marks various types of color control, which are reference images, color-scribbles, color hints and text hints. Methods with a light blue background are neural network-based approaches.

sure by Žeger et al. [110]. In addition, researchers will also compare the colorization time of different methods for different resolution images [10, 11] and the computing resources, such as network parameter size [2, 99].

In the qualitative comparison of different colorization methods, researchers usually use user study to evaluate, and then use box plots, radar plots, or bar graphs to display the results in the paper. Specifically, in the user study design scheme, the researcher will design multiple questions for different colorization results for users to score and evaluate. These questions mainly include: 1.The authenticity of the colored image [17]: They will mix colored images with real (natural) color images, allowing users to score the authenticity of the image; 2. Whether the colored result obeys the hint color or reference image color [35, 50, 13]: participants will be asked to evaluate the color consistency between the coloring result and the reference image or palette; 3. Color segmentation [26, 90]: Whether the coloring method can accurately identify the colors that should be used in different areas to prevent color bleeding and fusing problems; 4. Coloring time [26]: In interactive colorization methods the researcher will compare the interactive coloring time needed for users to complete the colorization tasks; 5. Quality of the coloring results: Whether the coloring method can produce reasonable results [50, 90]. In addition, researchers not only display user evaluation results, but also use analysis of variance to determine whether there are significant differences between different evaluation indicators, and perform statistical analysis on user survey results [80, 50, 102].

Moreover, existing indicators for quantitative evaluation of colorization results are inaccurate, including PSNR, SSIM, LPIPS, and qualitative comparison can better evaluate the pros and cons of different methods. However, due to the lack of a unified public dataset, especially in the field of line art colorization task, qualitative evaluation cannot fairly compare the capabilities of different colorization methods. Researchers should be committed to open datasets of different coloring tasks, and then use multiple quantitative indicators and qualitative comparisons to evaluate different methods under the same dataset, so as to evaluate the pros and cons of different methods more reasonably.

7. Conclusions

In this survey, we summarize many different methods in image colorization and related areas. Based on the user interaction, we divided grayscale colorization methods into three categories: fully-automatic colorization, semi-automatic colorization and language based colorization. For sketch images, most methods combine the fully-automatic colorization models with user guidance together, as simple editing can improve the result and make it closer to their expectations. For each area, they are pursuing different goals to meet the different demands, and with deep learning, huge progress has been made in recent years.

Exploring the connection between colorization for grayscale images and for sketch images, we found that there are many similarities, and the color processing techniques are the same. The common goal of grayscale colorization is to achieve the most realistic color which should be the same as the ground truth, so there are many methods that focus on how to fill in the one or more right colors for each pixel in the image. Usually some object colors are not fixed to one color in the real-world such as balloons and the color of a dress. And for sketch image colorization, the methods require accuracy which is achieved by image segmentation, and prevents the color bleeding issue. Most anime and manga character colors are unknown, so using the 'wrong' color will not be a big issue in sketch colorization. We summarize the models in Table 1, which list the grayscale image and sketch image colorization methods based on the publication year respectively. It can be seen from Table. 1 that before 2016 most research was carried out on grayscale images, and thereafter research has increased on sketch images Moreover, with the development of neural networks, the latest methods tend be to more automatic and less interactive. The two main problems of the current research are how to further improve the quality of generation while also reducing interactions, and how to accurately control the boundaries of colorization areas in a convenient way, such as text or sketch.

Developments in this area are closely connected with segmentation, semantic and style transfer study. Regardless of whether it is to improve the quality of image colorization or interactively control image colorization, researchers need to propose more accurate image region segmentation and semantic matching methods to determine the boundaries of coloring regions, so as to avoid color overflow and cross-coloring in colorized images. The current research trend is to use machine learning to solve problems instead of determining everything by hand. However the parameters of the network are hard to determine based on lack of information, and they still do not provide a good solution for all cases. At the same time, the current deep learning-based methods cannot precisely control the colorization result regardless of whether it is processing grayscale images or line art images, and lacks an effective and precise method to control interaction. This is mainly reflected in the fact that the interactive colorization methods based on hints, sketch or text can only provide color prior information for the region, but cannot control the region boundary or the propagation of color prior information.

In the existing methods, interactive control methods include simple color feature fusion, intermediate network output feature normalization, and color palette control after quantization of the reference image. These feature control methods need to be further improved. In the future, researchers may consider using the feature fusion method of the transformer model [44] or the CLIP (Contrastive Language-Image Pre-training) model [111]. In addition, the information contained in the sketch is sparse and ambiguous. The semantic analysis of the lines in the sketch can effectively reduce the ambiguity of the lines and further improve the quality of the image. To reach it, we need to refocus on the interpretability of the neural network structure, and have a deeper understanding of how color is formed in the colorization process. There are many commercial applications of colorization, and these can be further extended by exploring new formulations and solutions.

References

[1] Z. Cheng, Q. Yang, B. Sheng, Deep colorization, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 415–423.

[2] R. Zhang, P. Isola, A. A. Efros, Colorful image colorization, in: European conference on computer vision, Springer, 2016, pp. 649–666.

[3] A. Levin, D. Lischinski, Y. Weiss, Colorization using optimization, ACM Trans. Graph. 23 (3) (2004) 689–694. doi:10.1145/1015706.1015780.
URL https://doi.org/10.1145/1015706.1015780

[4] K. Sato, Y. Matsui, T. Yamasaki, K. Aizawa, Reference-based manga colorization by graph correspondence using quadratic programming, in: SIGGRAPH Asia 2014 Technical Briefs, SA '14, Association for Computing Machinery, New York, NY, USA, 2014. doi:10.1145/2669024.2669037.
URL https://doi.org/10.1145/2669024.2669037

[5] J. Zhang, Y. Chen, L. Li, H. Fu, C.-L. Tai, Context-based sketch classification, in: Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering, 2018, pp. 1–10.

[6] L. Yang, J. Zhuang, H. Fu, X. Wei, K. Zhou, Y. Zheng, Sketchgnn: Semantic sketch segmentation with graph neural networks, ACM Transactions on Graphics (TOG) 40 (3) (2021) 1–13.

[7] C. Ge, H. Sun, Y.-Z. Song, Z. Ma, J. Liao, Exploring local detail perception for scene sketch semantic segmentation, IEEE Transactions on Image Processing (2022).

[8] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[10] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, P.-A. Heng, Intrinsic colorization, in: ACM SIGGRAPH Asia 2008 Papers, SIGGRAPH Asia '08, Association for Computing Machinery, New York, NY, USA, 2008. doi:10.

1145/1457515.1409105.
URL https://doi.org/10.1145/1457515.1409105

[11] S. Iizuka, E. Simo-Serra, H. Ishikawa, 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) (Jul. 2016). doi:10.1145/2897824.2925974.
URL https://doi.org/10.1145/2897824.2925974

[12] J.-W. Su, H.-K. Chu, J.-B. Huang, Instance-aware image colorization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7968–7977.

[13] S. Yoo, H. Bahng, S. Chung, J. Lee, J. Chang, J. Choo, Coloring with limited data: Few-shot colorization via memory augmented networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[14] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, H.-Y. Shum, Natural image colorization, EGSR'07, Eurographics Association, Goslar, DEU, 2007, p. 309–320.

[15] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, A. A. Efros, Real-time user-guided image colorization with learned deep priors, arXiv preprint arXiv:1705.02999 (2017).

[16] T. Welsh, M. Ashikhmin, K. Mueller, Transferring color to greyscale images, ACM Trans. Graph. 21 (3) (2002) 277–280. doi:10.1145/566654.566576.
URL https://doi.org/10.1145/566654.566576

[17] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, H. Zhiyong, Image colorization using similar images, in: Proceedings of the 20th ACM international conference on Multimedia, ACM, 2012, pp. 369–378.

[18] R. Ironi, D. Cohen-Or, D. Lischinski, Colorization by example., in: Rendering Techniques, Citeseer, 2005, pp. 201–210.

[19] F. Fang, T. Wang, T. Zeng, G. Zhang, A superpixel-based variational model for image colorization, IEEE Transactions on Visualization and Computer Graphics 26 (10) (2020) 2931–2943.

[20] M. He, D. Chen, J. Liao, P. V. Sander, L. Yuan, Deep exemplar-based colorization, ACM Trans. Graph. 37 (4) (Jul. 2018). `doi:10.1145/3197517.3201365`.
URL `https://doi.org/10.1145/3197517.3201365`

[21] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, D. H. Salesin, Image analogies, in: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM, 2001, pp. 327–340.

[22] J. Liao, Y. Yao, L. Yuan, G. Hua, S. B. Kang, Visual attribute transfer through deep image analogy, ACM Trans. Graph. 36 (4) (Jul. 2017). `doi:10.1145/3072959.3073683`.
URL `https://doi.org/10.1145/3072959.3073683`

[23] J. Chen, Y. Shen, J. Gao, J. Liu, X. Liu, Language-based image editing with recurrent attentive models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8721–8729.

[24] D. Sỳkora, J. Dingliana, S. Collins, Lazybrush: Flexible painting tool for hand-drawn cartoons, in: Computer Graphics Forum, Vol. 28, Wiley Online Library, 2009, pp. 599–608.

[25] T. Yonetsuji, Paintschainer. github. com/pfnet, PaintsChainer 1 (2017) 2.

[26] L. Zhang, C. Li, T.-T. Wong, Y. Ji, C. Liu, Two-stage sketch colorization, in: SIGGRAPH Asia 2018 Technical Papers, ACM, 2018, p. 261.

[27] S.-Y. Chen, J.-Q. Zhang, L. Gao, Y. He, S. Xia, M. Shi, F.-L. Zhang, Active colorization for cartoon line drawings, IEEE Transactions on Visualization and

Computer Graphics 28 (2) (2022) 1198–1208. doi:10.1109/TVCG.2020.3009949.

[28] C. Zou, H. Mo, C. Gao, R. Du, H. Fu, Language-based colorization of scene sketches, ACM Trans. Graph. 38 (6) (Nov. 2019). doi:10.1145/3355089.3356561.
URL https://doi.org/10.1145/3355089.3356561

[29] Y. Qu, T.-T. Wong, P.-A. Heng, Manga colorization, in: ACM SIGGRAPH 2006 Papers, SIGGRAPH '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 1214–1220. doi:10.1145/1179352.1142017.
URL https://doi.org/10.1145/1179352.1142017

[30] C. Furusawa, K. Hiroshiba, K. Ogaki, Y. Odagiri, Comicolorization: Semi-automatic manga colorization, in: SIGGRAPH Asia 2017 Technical Briefs, SA '17, Association for Computing Machinery, New York, NY, USA, 2017. doi:10.1145/3145749.3149430.
URL https://doi.org/10.1145/3145749.3149430

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

[32] S. Messaoud, D. Forsyth, A. G. Schwing, Structural consistency and controllability for diverse colorization, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 596–612.

[33] J. Antic, Deoldify, https://deoldify.ai/.

[34] X.-H. Wang, J. Jia, H.-Y. Liao, L.-H. Cai, Affective image colorization, Journal of Computer Science and Technology 27 (6) (2012) 1119–1128.

[35] H. Bahng, S. Yoo, W. Cho, D. Keetae Park, Z. Wu, X. Ma, J. Choo, Coloring with words: Guiding image colorization through text-based palette generation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 431–447.

[36] G. Patterson, J. Hays, Sun attribute database: Discovering, annotating, and recognizing scene attributes, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2751–2758. doi:10.1109/CVPR.2012.6247998.

[37] A. Deshpande, J. Rock, D. Forsyth, Learning large-scale automatic image colorization, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 567–575.

[38] N. D. Ratliff, D. Silver, J. A. Bagnell, Learning to search: Functional gradient techniques for imitation learning, Autonomous Robots 27 (1) (2009) 25–53.

[39] G. Larsson, M. Maire, G. Shakhnarovich, Learning representations for automatic colorization, in: European Conference on Computer Vision, Springer, 2016, pp. 577–593.

[40] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[41] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, science 220 (4598) (1983) 671–680.

[42] A. Deshpande, J. Lu, M.-C. Yeh, M. J. Chong, D. Forsyth, Learning diverse image colorization, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017. doi:10.1109/cvpr.2017.307.
URL https://doi.org/10.1109%2Fcvpr.2017.307

[43] C. Lei, Q. Chen, Fully automatic video colorization with self-regularization and diversity, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3753–3761.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[45] M. Kumar, D. Weissenborn, N. Kalchbrenner, Colorization transformer, in: International Conference on Learning Representations, 2020.

[46] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, J.-L. Wu, An adaptive edge detection based colorization algorithm and its applications, in: Proceedings of the 13th annual ACM international conference on Multimedia, ACM, 2005, pp. 351–354.

[47] L. Yatziv, G. Sapiro, Fast image and video colorization using chrominance blending, IEEE transactions on image processing 15 (5) (2006) 1120–1129.

[48] C. Liu, J. Yuen, A. Torralba, SIFT flow: Dense correspondence across scenes and its applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 978–994.

[49] F. Pierre, J.-F. Aujol, A. Bugeau, N. Papadakis, V.-T. Ta, Luminance-chrominance model for image colorization, SIAM Journal on Imaging Sciences 8 (1) (2015) 536–563.

[50] B. Li, Y. Lai, M. John, P. L. Rosin, Automatic example-based image colorization using location-aware cross-scale matching, IEEE Transactions on Image Processing 28 (9) (2019) 4606–4619.

[51] L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D: nonlinear phenomena 60 (1-4) (1992) 259–268.

[52] S. H. Kang, R. March, Variational models for image colorization via chromaticity and brightness decomposition, IEEE Transactions on Image Processing 16 (9) (2007) 2251–2261.

[53] A. Bugeau, V. Ta, N. Papadakis, Variational exemplar-based image colorization, IEEE Transactions on Image Processing 23 (1) (2014) 298–307.

[54] J. Wang, X. Wang, Vcells: Simple and efficient superpixels using edge-weighted centroidal voronoi tessellations, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (6) (2012) 1241–1247.

[55] D. G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110. doi:10.1023/B:VISI.0000029664.99615.94.
URL https://doi.org/10.1023/B:VISI.0000029664.99615.94

[56] Y. Morimoto, Y. Taguchi, T. Naemura, Automatic colorization of grayscale images using multiple images on the web, in: SIGGRAPH '09: Posters, SIGGRAPH '09, Association for Computing Machinery, New York, NY, USA, 2009. doi:10.1145/1599301.1599333.
URL https://doi.org/10.1145/1599301.1599333

[57] B. Zhang, M. He, J. Liao, P. V. Sander, L. Yuan, A. Bermak, D. Chen, Deep exemplar-based video colorization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8052–8061.

[58] D. Chen, J. Liao, L. Yuan, N. Yu, G. Hua, Coherent online video style transfer, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1105–1114.

[59] Y. Wu, X. Wang, Y. Li, H. Zhang, X. Zhao, Y. Shan, Towards vivid and diverse image colorization with generative color prior, in: International Conference on Computer Vision (ICCV), 2021.

[60] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, K. Murphy, Tracking emerges by colorizing videos, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 391–408.

[61] S. Iizuka, E. Simo-Serra, Deepremaster: Temporal source-reference attention networks for comprehensive video enhancement, ACM Trans. Graph. 38 (6) (Nov. 2019). doi:10.1145/3355089.3356570.
URL https://doi.org/10.1145/3355089.3356570

[62] M. Shi, J.-Q. Zhang, S.-Y. Chen, L. Gao, Y. Lai, F.-L. Zhang, Reference-based

deep line art video colorization, IEEE Transactions on Visualization and Computer Graphics (2022).

[63] J. Lee, E. Kim, Y. Lee, D. Kim, J. Chang, J. Choo, Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5801–5810.

[64] L. Siyao, S. Zhao, W. Yu, W. Sun, D. Metaxas, C. C. Loy, Z. Liu, Deep animation video interpolation in the wild, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6587–6595.

[65] M. Ashikhmin, Synthesizing natural textures, in: Proceedings of the 2001 Symposium on Interactive 3D Graphics, I3D '01, Association for Computing Machinery, New York, NY, USA, 2001, p. 217–226. doi:10.1145/364338.364405.
URL https://doi.org/10.1145/364338.364405

[66] P. Bénard, F. Cole, M. Kass, I. Mordatch, J. Hegarty, M. S. Senn, K. Fleischer, D. Pesare, K. Breeden, Stylizing animation by example, ACM Trans. Graph. 32 (4) (Jul. 2013). doi:10.1145/2461912.2461929.
URL https://doi.org/10.1145/2461912.2461929

[67] O. Jamriška, v. Sochorová, O. Texler, M. Lukáč, J. Fišer, J. Lu, E. Shechtman, D. Sýkora, Stylizing video by example, ACM Trans. Graph. 38 (4) (Jul. 2019). doi:10.1145/3306346.3323006.
URL https://doi.org/10.1145/3306346.3323006

[68] R. Hu, M. Rohrbach, T. Darrell, Segmentation from natural language expressions, in: European Conference on Computer Vision, Springer, 2016, pp. 108–124.

[69] V. Manjunatha, M. Iyyer, J. Boyd-Graber, L. Davis, Learning to color from language, in: Proceedings of the 2018 Conference of the North American Chapter

of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 764–769. doi:10.18653/v1/N18-2120. URL https://www.aclweb.org/anthology/N18-2120

[70] E. Perez, F. Strub, H. de Vries, V. Dumoulin, A. C. Courville, Film: Visual reasoning with a general conditioning layer, in: AAAI, 2018.

[71] J. Heer, M. Stone, Color naming models for color selection, image editing and palette design, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2012, pp. 1007–1016.

[72] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[73] S. Fourey, D. Tschumperlé, D. Revoy, A fast and efficient semi-guided algorithm for flat coloring line-arts, in: Proceedings of the Conference on Vision, Modeling, and Visualization, EG VMV '18, Eurographics Association, Goslar, DEU, 2018, p. 1–9. doi:10.2312/vmv.20181247. URL https://doi.org/10.2312/vmv.20181247

[74] Y. Liu, Z. Qin, Z. Luo, H. Wang, Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks, arXiv preprint arXiv:1705.01908 (2017).

[75] P. Sangkloy, J. Lu, C. Fang, F. Yu, J. Hays, Scribbler: Controlling deep image synthesis with sketch and color, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5400–5409.

[76] Y. Ci, X. Ma, Z. Wang, H. Li, Z. Luo, User-guided deep anime line art colorization with conditional adversarial networks, in: 2018 ACM Multimedia Conference on Multimedia Conference, ACM, 2018, pp. 1536–1544.

[77] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[78] L. Zhang, Y. Ji, X. Lin, C. Liu, Style transfer for anime sketches with enhanced residual U-net and auxiliary classifier GAN, in: 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2017, pp. 506–511.

[79] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier GANs, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 2642–2651.

[80] T.-H. Sun, C.-H. Lai, S.-K. Wong, Y.-S. Wang, Adversarial colorization of icons based on contour and color conditions, in: Proceedings of the 27th ACM International Conference on Multimedia, MM ’19, Association for Computing Machinery, New York, NY, USA, 2019, p. 683–691. doi:10.1145/3343031.3351041.
URL https://doi.org/10.1145/3343031.3351041

[81] H. Thasarathan, K. Nazeri, M. Ebrahimi, Automatic temporally coherent video colorization (2019). arXiv:1904.09527.

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

[83] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.

[84] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, W. Hsu, Free-form video inpainting with 3d gated convolution and temporal PatchGAN, In Proceedings of the International Conference on Computer Vision (ICCV) (2019).

[85] S.-H. Zhang, T. Chen, Y.-F. Zhang, S.-M. Hu, R. R. Martin, Vectorizing cartoon animations, IEEE Transactions on Visualization and Computer Graphics 15 (4) (2009) 618–629.

[86] P. Hensman, K. Aizawa, cGAN-based manga colorization using a single training image, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Vol. 3, IEEE, 2017, pp. 72–77.

[87] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.

[88] X. Zhu, J. Lafferty, Z. Ghahramani, Combining active learning and semi-supervised learning using gaussian fields and harmonic functions, in: ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining, Vol. 3, 2003.

[89] H. Ling, D. W. Jacobs, Shape classification using the inner-distance, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 286–299. doi:10.1109/TPAMI.2007.41.
URL https://doi.org/10.1109/TPAMI.2007.41

[90] H. Kim, H. Y. Jhoo, E. Park, S. Yoo, Tag2pix: Line art colorization using text tag with secat and changing loss, arXiv preprint arXiv:1908.05840 (2019).

[91] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2018) 834–848.

[92] C. Liu, Z. Lin, X. Shen, J. Yang, X. Lu, A. Yuille, Recurrent multimodal interaction for referring image segmentation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

[93] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, S.-M. Hu, Sketch2photo: Internet image montage, ACM Trans. Graph. 28 (5) (2009) 1–10. doi:10.1145/

1618452.1618470.
URL https://doi.org/10.1145/1618452.1618470

[94] N. Ben-Haim, B. Babenko, S. Belongie, Improving web-based image search via content based clustering, in: 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), IEEE, 2006, pp. 106–106.

[95] C. Rother, V. Kolmogorov, A. Blake, GrabCut: Interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (3) (2004) 309–314. doi:10.1145/1015706.1015720.
URL https://doi.org/10.1145/1015706.1015720

[96] T. Chen, P. Tan, L.-Q. Ma, M.-M. Cheng, A. Shamir, S.-M. Hu, PoseShop: Human image database construction and personalized content synthesis, IEEE Transactions on Visualization and Computer Graphics 19 (5) (2013) 824–837. doi:10.1109/TVCG.2012.148.
URL http://dx.doi.org/10.1109/TVCG.2012.148

[97] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[98] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134.

[99] W. Chen, J. Hays, SketchyGAN: Towards diverse and realistic sketch to image synthesis, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 9416–9425.

[100] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, J. Hays, TextureGAN: Controlling deep image synthesis with texture patches, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8456–8465.

[101] L. Gao, J. Yang, T. Wu, Y.-J. Yuan, H. Fu, Y.-K. Lai, H. Zhang, SDM-NET: Deep generative network for structured deformable mesh, ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019) 38 (6) (2019) 243:1–243:15.

[102] S.-Y. Chen, W. Su, L. Gao, S. Xia, H. Fu, DeepFaceDrawing: Deep generation of face images from sketches, ACM Trans. Graph. 39 (4) (Jul. 2020). `doi:10.1145/3386569.3392386`.
URL `https://doi.org/10.1145/3386569.3392386`

[103] Y. J. Lee, C. L. Zitnick, M. F. Cohen, Shadowdraw: Real-time user guidance for freehand drawing, in: ACM SIGGRAPH 2011 Papers, SIGGRAPH '11, Association for Computing Machinery, New York, NY, USA, 2011. `doi:10.1145/1964921.1964922`.
URL `https://doi.org/10.1145/1964921.1964922`

[104] S.-Y. Chen, F.-L. Liu, Y.-K. Lai, P. L. Rosin, C. Li, H. Fu, L. Gao, DeepFaceEditing: Deep face generation and editing with disentangled geometry and appearance control, ACM Trans. Graph. 40 (4) (Jul. 2021). `doi:10.1145/3450626.3459760`.
URL `https://doi.org/10.1145/3450626.3459760`

[105] D. Sỳkora, J. Buriánek, J. Žára, Colorization of black-and-white cartoons, Image and Vision Computing 23 (9) (2005) 767–782.

[106] D. Ishii, K. Kawamura, H. Watanabe, A study on control parameters of frame separation method for comic images, Tech. rep., IEICE technical report, The Institute of Electronics, Information and Communication Engineers (2009).

[107] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, Advances in neural information processing systems 29 (2016) 2234–2242.

[108] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs

trained by a two time-scale update rule converge to a local Nash equilibrium, Advances in neural information processing systems 30 (2017).

[109] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 586–595.

[110] I. Žeger, S. Grgic, J. Vuković, G. Šišul, Grayscale image colorization methods: Overview and evaluation, IEEE Access 9 (2021) 113326–113346. doi:10.1109/ACCESS.2021.3104515.

[111] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.

Author Contributions Section

Shu-Yu Chen	Writing - Original Draft
Jia-Qi Zhang	Writing - Original Draft
You-You Zhao	Writing - Original Draft
Paul L. Rosin	Writing - Review & Editing
Yu-Kun Lai	Writing - Original Draft
Lin Gao	Writing - Review & Editing and Project administration

Declaration of Interest Statement

Declaration of interests

☐The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Shu-Yu Chen reports financial support was provided by Foundation for Innovative Research Groups of the National Natural Science Foundation of China. Lin Gao reports financial support was provided by Royal Society Newton Advanced Fellowship.