Multimedia Tools and Applications (2020) 79:27583–27614
https://doi.org/10.1007/s11042-020-09299-2

Color vision deficiency datasets & recoloring evaluation using GANs

Hongsheng Li[1] · Liang Zhang[1] · Xiangdong Zhang[1] · Meili Zhang[1] · Guangming Zhu[1] · Peiyi Shen[1] · Ping Li[2] · Mohammed Bennamoun[3] · Syed Afaq Ali Shah[3]

Received: 23 August 2019 / Revised: 4 June 2020 / Accepted: 6 July 2020 /
Published online: 28 July 2020

Abstract
People with Color Vision Deficiency (CVD) cannot distinguish some color combinations under normal situations. Recoloring becomes a necessary adaptation procedure. In this paper, in order to adaptively find the key color components in an image, we first propose a self-adapting recoloring method with an Improved Octree Quantification Method (IOQM). Second, we design a screening tool of CVD datasets that is used to integrate multiple recoloring methods. Third, a CVD dataset is constructed with the help of our designed screening tool. Our dataset consists of 2313 pairs of training images and 771 pairs of testing images. Fourth, multiple GANs i.e., pix2pix-GAN [1], Cycle-GAN [2], Bicycle-GAN [3] are used for colorblind data conversion. This is the first ever effort in this research area using GANs. Experimental results show that pix2pix-GAN [1] can effectively recolor unrecognizable colors for people with CVD, and we predict that this dataset can provide some help for color blind images recoloring. Datasets and source are available at: https://github.com/doubletry/pix2pix , https://github.com/doubletry/CycleGAN and https://github.com/doubletry/BicycleGAN.

Keywords Color vision deficiency · Recolor · Improved octree quantification method · GAN

1 Introduction

With the explosive growth of multimedia technology, perceiving color accurately is becoming a fundamental necessity for effective visual communication. According to the statistics in [51], there are more than 20 million humans with color vision deficiency (CVD) in the world. Obviously, it is necessary to resolve the CVD problem in order for these people to lead a comfortable daily life. Normal vision is based on the response to photons in three different type of cones which are contained in the retina of the eye, and the peak sensitivities

✉ Liang Zhang
liangzhang@xidian.edu.cn

Extended author information available on the last page of the article.

of these three types are located in the long wavelength (L), middle wavelength (M), and the short wavelength (S) regions of the visible spectrum, respectively [58]. However, people with CVD are divided from partial or complete deficiency of one or more cones. In fact, there are three major types of CVD: anomalous trichromacy, dichromacy and monochromacy. Anomalous trichromacy results from the shift of the peak sensitivity of the cones, which can be divided into protanomaly, deuteranomaly and tritanomaly, according to the shift spectrum of $L-$, $M-$, and $S-$ cones, respectively. Dichromacy is presented when one of the cones is absent, and it can also be divided into protanopia, deuteranopia and tritanopia, respectively. Monochromacy is caused by the lack of all types of cones. In this condition, only brightness variations can be perceived.

In this paper, we focus on protanopia and deuteranopia since they consist of nearly 98% of CVD cases[51]. In Fig. 1, we show an example of how protanopia and deuteranopia patients perceive colors. It can easily be seen that in Fig. 1b and 1c, the significant information that is perceived by people with protanopia and deuteranopia respectively disappears or becomes indistinct. Undoubtedly, CVD will cause considerable inconvenience in people's daily life, because their ability to distinguish colors is useful in many daily activities, such as recognizing road signs, and traffic lights [38]. Therefore, a lot of image processing research has been conducted to improve the color perceptibility for CVD. The existing methods can be categorized into color-to-grey, edge segmentation and recoloring. Color-to-grey methods [3, 43] simply convert color images to greyscale images. Although these methods can maintain the contrast between the different colors and retain the partial information in the color images, they in fact directly discard the color information. Edge segmentation methods [35, 49] detect the edges in the image and assign different colors to different regions crudely. However, the performance of these methods relies heavily on the image segmentation process. Recoloring methods not only reserve the color information but they also enhance the color perceptibility. They firstly find the unrecognizable colors for people with CVD, and then adapt them to other recognizable colors.

Recoloring is a color adaptation procedure for each pixel in the image, which belongs to pixel-level tasks. There are many other pixel-level tasks in the field of computer vision, such as semantic segmentation, normal estimation, deep contour and depth estimation. These tasks can be resolved using deep learning networks. For semantic image segmentation, we want to get the classification result of each pixel in its corresponding position. There are many deep learning networks which deal with pixel-level tasks, such as FCN [52], DeepLab [5, 7–9], and PSPNET [63]. However, using deep learning to train the network requires a large amount of training data. At present, there are many datasets with labels on the pixel level, such as NYUD v2 [53], PASCAL [17], ADE20K [64], and KITTI [24]. It can be seen

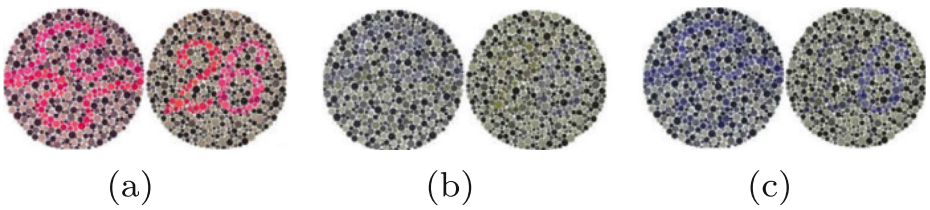

Fig. 1 An example of how protanopia and deuteranopia perceive colors. (**a**) The images that are perceived by people with normal vision. (**b**) The images that are perceived by people with protanopia. (**c**) The images that are perceived by people with deuteranopia

that recoloring can be resolved through deep learning networks. But there is no available dataset for training these networks to enhance the color perceptibility for people with CVD.

For normal people and color blindness, the color space seen is different. We want to convert images in color space of normal people to the counterpart of blindness. This can be regarded as a translation between two domains that are exactly coincident with the functionality of the Generative Adversarial Networks (GANs) [27]. GANs initially used unconditional GAN in image-to-image processing. This approach achieved remarkable results on image manipulation guided by user constraints [65] particularly, but unconditional GAN is uncontrollable and users cannot know what kind of content the generator will produce. Therefore, some researchers have proposed a new method called Conditional GAN [32], to generate images which meet a set of conditions. For example, the pix2pix-GAN [32], Bicycle-GAN [67] and Cycle-GAN [66] applied in this paper are conditional GANs, and experiments have proved that they are superior to other unconditional GANs in terms of complexity and the accuracy of the data generation.

The main contributions of this paper include:

1) A novel self-recoloring method, which can effectively enhance the color perceptibility for people with CVD.
2) A new screening tool for a CVD dataset which integrates multiple recoloring methods.
3) A new CVD dataset using our designed tool, which consists of train set and test set.
4) Extensive experimental tests on multiple GAN frameworks for colorblind data conversion.

The rest of this paper is organized as follows. Section 2 reviews the related works on recoloring, deep learning networks for pixel-level task and some datasets label tools. Section 3 describes the procedure of dataset construction in detail, and introduces the self-recoloring method which is proposed in this paper. Section 4 demonstrates the evaluation of our dataset with some popular GANs to perform the recoloring. Section 5 concludes this paper.

2 Related work

2.1 Recoloring

The various recoloring methods can be categorized into rule-based and optimization-based.

In the case of optimization-based methods, recoloring is performed by minimizing an objective function based on some evaluation metrics. Huang C R [30] and Huang J B [29] obtained the optimal colors by minimizing the difference between the original and the recolored image in the Lab color space. Milic [46, 47] found the optimal choice of discernible color bins for each color in the original image. However, these optimization-based methods are easy to fall into a shallow local minimum. They therefore cannot meet their real-time requirement because of their exhaustive search for the minimum.

On the other hand, the rule-based methods [10, 34, 36, 37, 48] need less time than the optimization-based methods, since they only require a linear computation. There are three factors influencing the performance of the rule-based methods: key color cluster, unrecognizable colors judgment and unrecognizable colors recoloring. Usually, key colors are clustered using a simple method, e.g., K-means, but they have to predefine the number of clusters and obviously the optimal number of cluster is different for different images. Doliotis [10] and Khurge [36] found unrecognizable colors by comparing the difference between the original and the simulated images with a predefined threshold. Kim [34, 37]

found the unrecognizable colors through a predefined red pixel mask for protanopia. However, all these methods involve some predefined parameters, which have a great impact on the performance. When recoloring these unrecognizable colors, Khurge [36] proposed an experience-based matrix, but the matrix cannot guarantee that all the unrecognizable colors are recolored accurately. Doliotis [10] and Kim [37] recolored the unrecognizable colors by multiplying a given matrix iteratively until a stopping iteration condition was satisfied. The stopping iteration condition is also a predefined parameter similar to unrecognizable colors judgment, which cannot guarantee a high contrast between the different colors in the recolored images adaptively. Furthermore, in order to make sure that all the recolored colors are recognizable, these methods recolored all the unrecognizable colors simultaneously. Nevertheless, this requires more iterations.

2.2 Pixelwise prediction tasks

Pixelwise prediction problems can be divided into several levels according to the level of features. For simplicity, we divide them into low-level tasks, mid-level tasks, and high-level tasks.

The low-level tasks, such as edge detection [11, 42, 61] and optical flow [1, 13, 55], are tasks that use low-level features, which are basic features extracted from an image without the need for any shape or spatial relationship. Low-level features contain a little shallow semantic information. By extracting and using the state of each pixel and its neighborhood, it is generally possible to solve low-level tasks. For example, in the edge detection task, by detecting a grayscale change about a pixel, it is possible to determine whether the pixel is on the edge of the object.

The mid-level tasks, such as depth or normal estimation [2, 15, 16, 59], are tasks that use higher-level features which consist of low-level features, which typically include object morphologies and spatial relationships. Compared to low-level tasks, the scope of feature extraction for the mid-level tasks is not limited to the pixel neighborhood, but also needs the position and shape information of the objects. For example, in the depth estimation task, the pose of the object and its spatial position are required to predict the depth for each pixel.

High-level tasks, such as keypoint prediction [4, 60], object detection [19, 26, 28, 31] and semantic segmentation [33, 41, 52, 63], combine multi-layer features to extract the semantic information in the image and predict a semantic label for each pixel. High-level features ignore the shape and the spatial relationship of objects, and mainly contain the common features of a class of objects. Hence they are also called semantic features. For example, in the semantic segmentation task, each pixel of a "cat", category or pose needs to be marked as a "cat". High-level tasks are usually the most difficult pixelwise. To ensure that the algorithm can accurately abstract the semantic information, the algorithm usually requires translation, scale, and rotation invariance. Therefore, the missing pixel position information in the semantic features and the pixelwise annotation required by the task are conflicting. This adds to the difficulty of the high-level tasks.

Currently, due to the huge increase of the speed brought by GPU computation, convolutional neural network algorithms are widely used for pixelwise prediction tasks, so that multi-scale features from low to high levels can be extracted with one algorithm in a short time. Fusing these features together by adding or concatenating them can result in multi-scale information with both high-level semantic information and low-level location information. This reduces the difficulty and improves the performance of the semantic pixelwise tasks.

Recoloring for the CVD tasks only needs to process the colors, so we can think of it as a low-level task. Through the use of convolutional neural networks, we can quickly get coarse semantic features to guide the learning of the low-level features. For example, adjacent objects of similar color can be distinguished by objects and then recolored with different colors to increase the distinguishability between different objects. This means that tasks, such as recoloring, are no longer low-level tasks since they only need to extract color features, but high-level tasks which require multi-scale features fusion. We hope that our dataset has a higher goal to meet the needs of CVD's high-level tasks.

This paper proposes the task of image-to-image translation, while implemented in a neural network, it includes image generation [20], image editing [65], and representative learning [44]. Currently, GANs [27] have a profound influence in this filed. The GAN framework trains to learn the networks of generator and discriminator. The generator network maps a random vector to the real image, and the discriminator network makes the generated image to be undistinguishable from the real images. Goodfellow et al. [27] introduced the GANs framework, first to generate a picture that is roughly similar to the original image. Since then, many improved methods were proposed, such as pix2pix-GAN[32], Cycle-GAN [66] and Bicycle-GAN [67].

The recently proposed pix2pix-GAN [32] model produced impressive results in the field of Image-to-image translation. With an additional input, conditional GANs is used to generate a perceptually realistic output which has some structural similarity with the input. Cycle-GAN [66] is an improvement based on pix2pix model. Instead of a paired training data, the image is translated between the source domains and target domains and identified by a discriminator. Bicycle-GAN [67] combines cVAE-GAN [39, 40] and cLR-GAN [6, 12, 14] to create diverse outputs for different latent codes by enforcing the one-to-one mapping of the latent code and output.

2.3 Labeling tools

A large quantity of data with labels is required to train a deep learning model. Therefore, some convenient labeling tools are proposed to replace the consuming manual labeling process.

Labelme [50] is a labeling tool for image segmentation. This tool implements the basic labeling work and the function of converting json files to a label image, which can save some information about the object into a json file. LabelImg [56] is a labeling tool for image detection. The functions of the label storage, previous picture and the next picture in this tool are very convenient and practical for users. BRAT [54] is a web-based text annotation tool, which is mainly used for structured annotation of texts. The annotation results generated by BRAT can structure unstructured original texts. Vatic [57] is a video annotation tool, which supports automatic extraction of granular annotation tasks and provides access to the Mechanical Turk platform. The tool uses a simple GUI interface and supports multiple shortcut keys.

3 Datasets construction

This part mainly introduces the construction process of our CVD dataset. It mainly includes four steps: data collection and preprocessing, data recoloring, data filtering, and data statistics. In the data recoloring step, this paper introduces a self-adapting recoloring algorithm and compares it to another state-of-the-art recoloring algorithm.

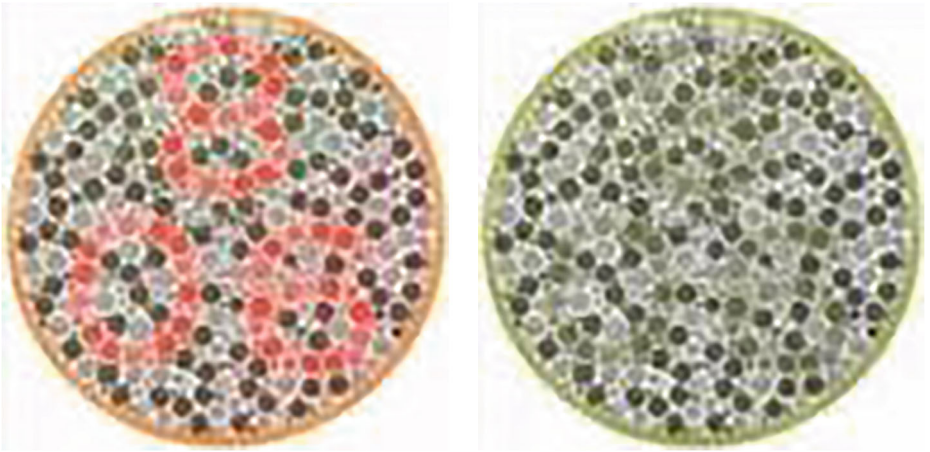

Fig. 2 The left is the original image, and the right is the color blind simulation image. The red and black colors in the image are indistinguishable. Hence, we extract the two colors from the image

3.1 Data collection and preprocessing

The first step for dataset construction is to collect data. In general, image data collection methods include self-photographing and web crawler. To construct our dataset, color blindness images are obtained by website crawler first, and preprocessed to facilitate the subsequent data statistics.

3.1.1 Data collection

This step adopts two approaches in order to get hold of the unrecognized images by CVD. The first phase extracts color pairs that are confused by color blindness from Ishihara plates, as shown in Fig. 2. The second phase searches for indistinguishable color pairs in the gradient chromaticity diagram. As shown in Figs. 3 and 4, color pairs in the red box are not recognized by CVD. In the process of collecting data, we first choose color pairs that are not the same in the original image. However, the CVD regards them as if they are of the same color pairs. Thus they cannot distinguish the original color pairs. The next step is to select the images based on the color pairs through the website.

Through the above two approaches, nine groups of color pairs are obtained. Each group contains at least two colors that are confused by CVD. Figure 5 gives the details of these color pairs. Afterwards it can be set for the hexadecimal value and the weight of each group color on the website[1], and the website can retrieve the relevant images, as shown on Fig. 6. Finally, we download the images using a crawler.[1]

3.1.2 Data preprocessing

The main purpose of this process is to eliminate any irrelevant information in the image, and to filter out interference, noise and to recover useful information. To facilitate the exhaustion of the statistical data, all images are resized to 400 * 300 in this CVD datasets.

[1] website[1]: http://labs.tineye.com/multicolr/

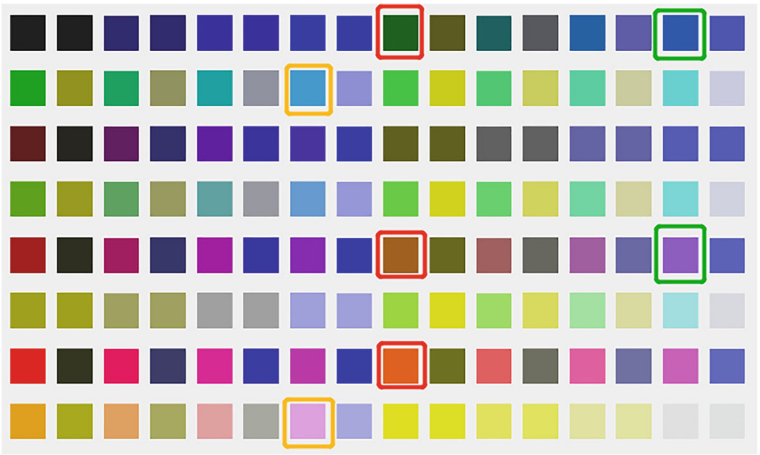

Fig. 3 The chromaticity diagram. The odd columns show the original images, and the even columns show the images perceived by CVD. Three sets of colors are given here, represented by yellow, red, and green boxes, respectively. Each group of colors appears to be of the same color as for in CVD

Fig. 4 A sample of chromaticity diagram. The top row is the original image, and the bottom row is the image seen by CVD

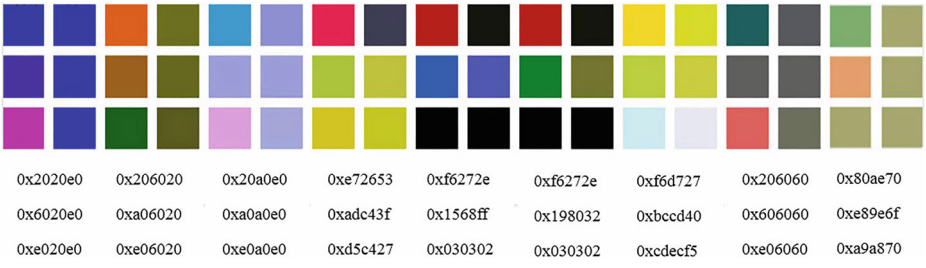

Fig. 5 Odd columns show the original images, and the even columns show the images perceived by CVD. Each column is a set of color pairs. The RGB hexadecimal values for each set of colors are given below the colorness

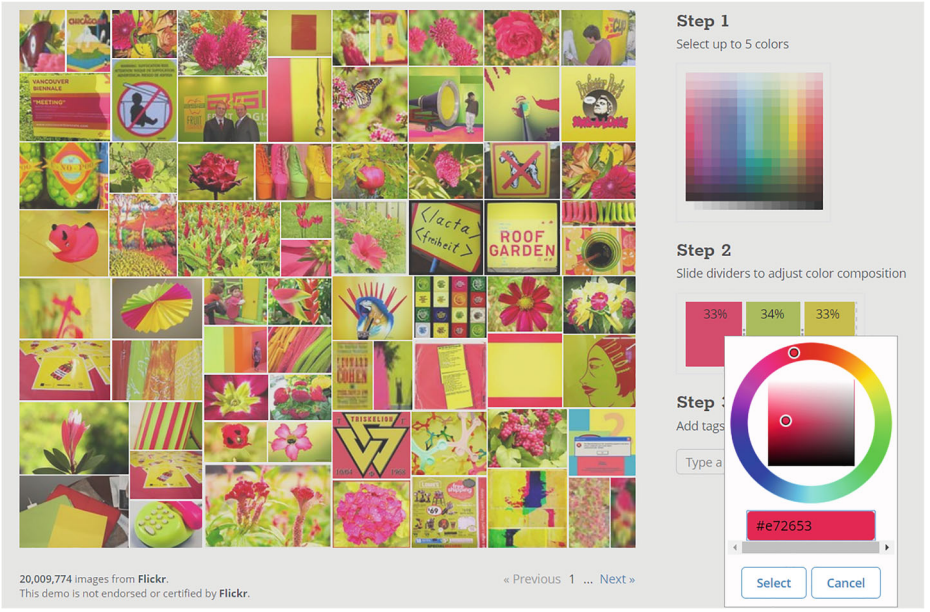

Fig. 6 The interface of the image download website

We downloaded 5,699 images of 400 × 300 pixels in total, which are divided into nine groups, each with 144, 503, 575, 851, 806, 851, 789, 216 and 934 images, respectively.

3.2 Data recoloring

In order to make the images obtained above to be correctly distinguished by CVD, all images are fed into two recoloring algorithms to be recolored. One is an optimization-based algorithm proposed by Huang [29], and the other is a self-adapting rule-based recoloring method proposed in this paper (Section 3.2.2). After recoloring, all the undistinguished colors become distinguishable for color blindness.

3.2.1 Optimization-based recoloring algorithm

The optimization-based recoloring algorithm proposed by Huang [29] preserves the visual detail when perceived by CVD. They first extract the representing colors in an image and then find the optimal mapping to maintain the contrast between each pair of these representing colors.

3.2.2 Self-adapting recoloring algorithm

The optimization-based algorithm (Section 3.2.1) changes most of the colors of the original image, in a way that CVD cannot correctly understand the colors that should be known in the original image. Here we propose an efficient rule-based recoloring method for people with CVD. Figure 7 shows the flowchart of our proposed recoloring method.

The exact steps are described as follows:

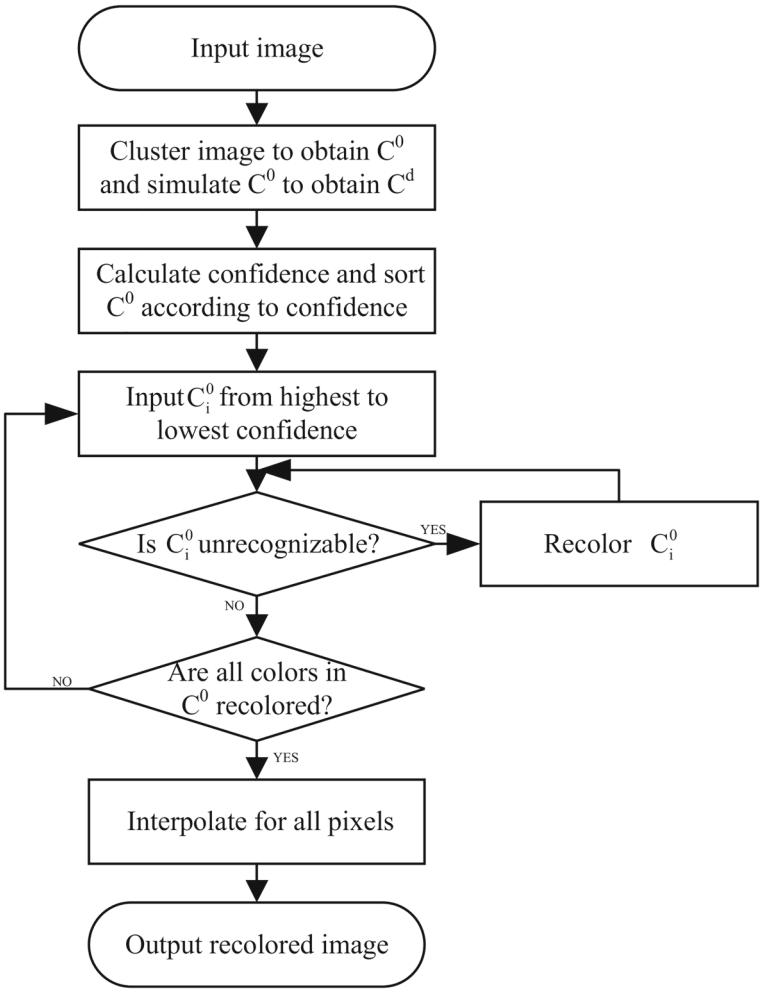

Fig. 7 The flowchart of proposed algorithm

1) Key colors $C^0 = C_1^0, C_2^0, \cdots, C_N^0$ are extracted from an original image with an improved Octree quantization method, and at the same time, the simulated key colors $C^d = C_1^d, C_2^d, \cdots, C_N^d$ are obtained by the well-known color blind simulation algorithm;

2) Color confidences are computed and all key colors in C^0 are ordered from the highest to the lowest confidence, which means C_j^0 has a higher confidence value than C_i^0 when j is smaller than i;

3) From the highest to the lowest confidence, judging whether C_i^0 is unrecognizable by people with CVD by the differences between C_i^0 and the other colors with a higher confidence $C_j^0 (j \leq i)$. If it is unrecognizable, then go to step 4, otherwise go to step 5;

4) An unrecognizable color C_i^0 is recolored to its corresponding recolored color O_i^d by multiplying it with a mapping matrix iteratively, until C_i^0 is not confused with those recolored colors, which have a higher confidence ($O_j^d (j \leq i)$);

5) Judging whether all key colors in C^0 have been recolored. If it is, perform step 6, otherwise go back to step 3;

6) All pixels in the original image are recolored through interpolation, in which the distance between a recolored pixel and its corresponding recolored key color should be equal to the distance before recoloring. Given a pixel P_j in the original image, and its corresponding key color C_i^0, the recolored color R_j for pixel P_j is obtained using (1).

$$R_j = O_i^d + (P_j - C_i^0) \tag{1}$$

An Improved Octree Quantization Method In the existing rule-based recoloring methods [10, 34, 36, 37, 48], the number of cluster has to be predefined before hand. Therefore, it is almost impossible to find an optimal number of clusters that can provide a satisfactory quantization for different images. In order to adaptively obtain the number of clusters according to the image characteristics, we propose an improved Octree quantization method (IOQM).

The basic idea of our proposed IOQM is to describe the RGB color space in a tree structure, in which the root node represents the whole RGB color space, and the other nodes in the Octree represent a small cube of their parent node (each node has eight sub-nodes). If eight sub-nodes are added, the result is equal to the cube denoted by their parent node. Each leaf node stores the dedicated color value and its total pixel count in the image [45]. The proposed IOQM can be performed in three phases:

i Filling the Octree when the number of leaf nodes is less than a predefined value.
ii Reducing the leaf nodes when the number of leaf nodes is greater than a predefined value.
iii Merging the leaf nodes with little pixel counts.

***Phase 1*: Filling the Octree**. For each pixel in the image, its RGB value can be represented by 8-bit binary. The RGB value at each bit can form a 3-bit binary index to determine the sub-node localization at a different depth, which corresponds to the pixel. The RGB value at the 7-th bit determines the sub-node localization at depth 2, and the RGB value at the 6-th bit determines the sub-node localization at depth 3, and so on. Since the binary RGB value has eight bits and every bit is located at different depths, the maximum depth is up to 9. As shown in Fig. 8, for a pixel with an RGB value (81,240,150), the index of the depth 2 is 3 (011), and the index of depth 3 is 6 (110), and so on.

All pixels in the image are sequentially scanned and added to the proposed Octree using the method described above. If the color of a pixel has already been accounted for the Octree, we only need to increase the pixel count of the leaf node and update its corresponding color to a new averaged value. Otherwise, the color of the current pixel will be added into the Octree as a new leaf node. In order to cluster similar colors, the maximum number of leaf nodes, denoted by k, is predefined (256 in our experiment). If the number of leaf nodes has exceeded this value, the number of leaf nodes has to be reduced before filling the current pixel.

Figure 9 shows the results of different values of k. We set the value of k to 64, 128, 256, and 512, respectively. We argue that in the process of nodes reducing (in **Phase 2**), nodes

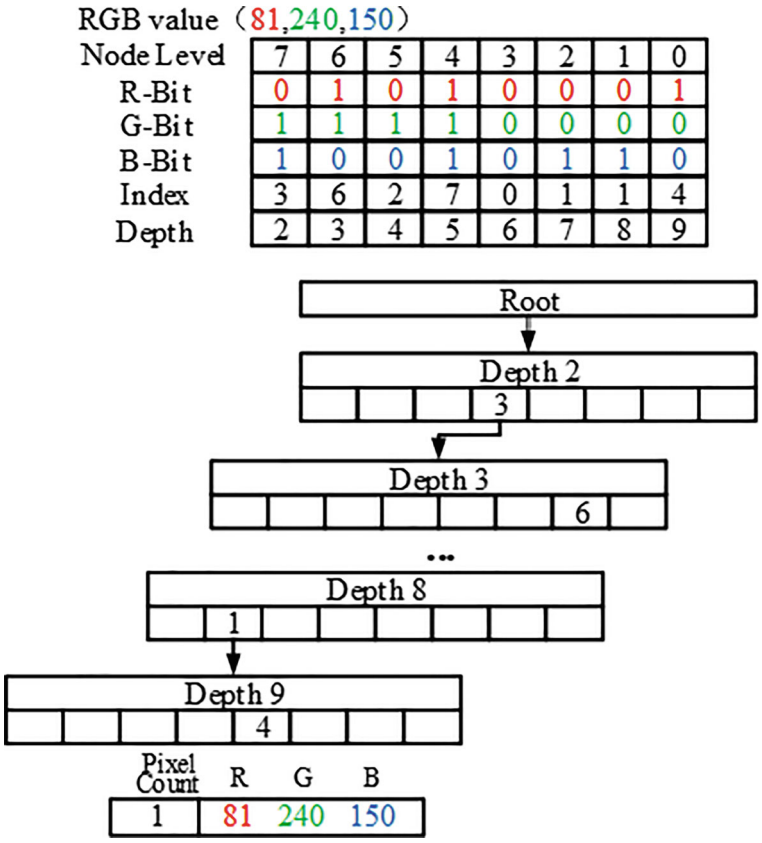

Fig. 8 An example of filling the proposed Octree

with similar colors are merged. And in the subsequent clustering process (in **Phase 3**), the number of clustering categories is less than 64, so the maximum number of nodes set has little effect on the final result. In order to ensure the diversity of node colors and reduce the computational complexity of clustering, we set k to 256 in this experiment.

***Phase 2*: Reducing the number of leaf nodes**. When the number of leaf nodes has exceeded the predefined value k, the Octree will be reduced.

During the reduction, the chosen leaf nodes are merged with their parent node, and their parent node becomes a new leaf node. The pixel count of a new leaf node is the sum of its children, and the dedicated color is the average of its children. For example, let us consider the colors (81,240,150), (109, 204, 170) and (81,241,151), the Octree before reduction is shown in Fig. 10a. Since the leaf nodes have the closest colors, the reduction is done at depth 9. The Octree, after reduction, is shown in Fig. 10b, the $0-th$, $4-th$ and $7-th$ leaf nodes at depth 9 are reduced and their parent node becomes a new leaf node. The pixel count of the new leaf node is the sum of its children ($1+5+5=11$) and the dedicated color is the average of its children ($\frac{80\times1+81\times5+81\times5}{11}\approx 88$).

***Phase 3*: Merging the leaf nodes**. Once all the pixels in the image have been added to the Octree, each leaf node represents a different color, but the predefined value k (256 in our experiment) is not suitable for all images. In order to obtain the optimal number of

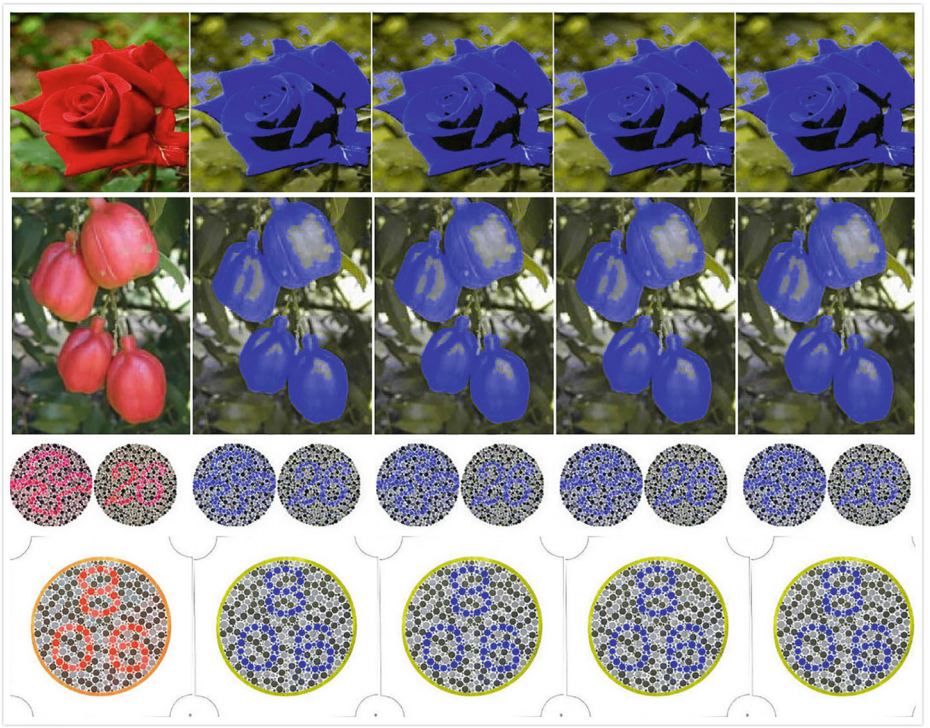

Fig. 9 Comparison when using different k in our experiments. Column 1 represents four different test images. Columns 2 to 5 represent the recolored images with different k ($k = 64$), ($k = 128$), ($k = 256$) and ($k = 512$), respectively

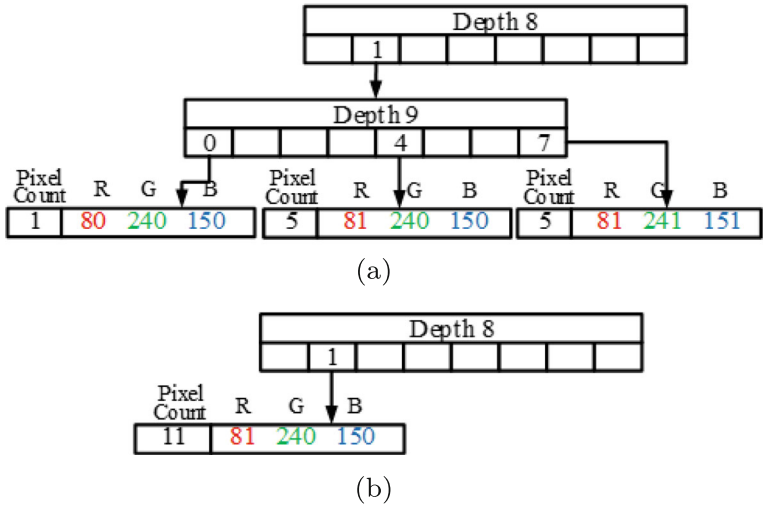

Fig. 10 An example of the reduction of the leaf nodes

clusters adaptively, we merge some leaf nodes according to the image characteristics. If the sum of the pixel count of the leaf nodes with the same parent node is occupied, these leaf nodes will be merged with their parent node, which is similar to the reduction process of the leaf nodes. Finally, the number of leaf nodes is the optimal number of the clusters, and the allocated colors of the leaf nodes are the cluster color.

The comparative results between IOQM and k-means are shown in Fig. 11, where the second row is obtained by IOQM, and the third row is obtained by *k*-means. The number of clusters is set to 8 for all images when using the *k*-means, but the number of clusters for four images is determined adaptively and automatically to be 5, 10, 10 and 7 with IOQM. As we know, *k*-means is a simple clustering method, but the number of clusters has to be predefined manually. However, the optimal number of clusters is different for different images. If the number of clusters is smaller than the optimal number, the different colors will be recolored to the same color, as shown in column 2, orange and red are recolored to blue in row 3, but they are recolored to different colors in row 2. If the number of clusters is larger than the optimal number, the similar colors will be recolored to different colors, as shown in the first and the last columns, red is recolored to two different colors in row 3, but it is recolored correctly in row 2.

Key Color Confidence Calculation In the existing rule-based recoloring methods [10, 34, 36, 37, 48], all unrecognizable colors for people with CVD are judged and recolored simultaneously. This requires more iterations in order to make sure all the unrecognizable colors are recolored accurately. In order to speed up the execution time, we propose a "key color confidence", which is used to judge and recolor an unrecognizable color at a time so that the

Fig. 11 Comparison between IOQM and k-means. The first row represents four different test images. The second row shows the corresponding recolored images obtained by IOQM. The third row shows the corresponding recolored images obtained by *k*-means

recoloring process can be done sequentially. The optimal recolored color can be obtained as long as the current recolored color is not confused with other recolored colors.

Key color confidence is the confidence on whether a color is recognizable by people with CVD. A key color with a higher confidence should be taken as the benchmark for recoloring other key colors with lower confidence. In order to obtain the key color confidence, the color differences $D = D_1, D_2, \cdots, D_N$ between the key colors C^0 and the simulated key colors C^d are first calculated. Then, a key color with a smaller color difference is assigned a higher confidence. Finally, all key colors are ordered from the highest to the lowest confidence. Color differences between C^0 and C^d are calculated by a well-known weighted Euclidean distance, as shown in (2), which approximates to the perceptibility distance of human [21].

$$D_k = \sqrt{\alpha \times \triangle R_k^2 + \beta \times \triangle G_k^2 + (1 - \alpha - \beta) \times \triangle B_k^2} \tag{2}$$

Where $\triangle R_k = C_{k,R}^0 - C_{k,R}^d$, $\triangle G_k = G_{k,R}^0 - G_{k,R}^d$ and $\triangle B_k = B_{k,R}^0 - B_{k,R}^d$ are the differences between three components (red, green and blue) of C_k^0 and C_k^d, and α, β are the weights of $\triangle R_k$ and $\triangle G_k$, respectively.

The experimental results with different weights in (2) are shown in Fig. 12. As described in Section 3, the color confidence and the unrecognizable colors are dependent on the different weights of (2). Since protanopia and deuteranopia are not able to distinguish the red and the green components, α and β in (2) should be set with the same values. The contrast of the recolored images in the last two columns are lower than the second and third columns, because they pay more attention to either the red or the green components in the different computation. Thus, the weights α and β are both set to 0.4 in our next experiments.

In order to speed up the execution time, an unrecognizable color by people with CVD is then judged and recolored from the highest to the lowest confidence, so that the re-coloring process can be done sequentially instead of simultaneously.

Unrecognizable Colors Judgment and Recoloring In the existing rule-based recoloring methods [10, 34, 36, 37, 48], the unrecognizable colors by people with CVD and the stopping iteration condition are determined using some predefined parameters. The contrast between the different colors of the recolored image is heavily dependent on the adjustable predefined parameters. In order to improve the contrast, the color differences, between a color and other colors with higher confidence, are considered as an unrecognizable benchmark for this color. An optimal recolored color for this unrecognizable color is obtained when these corresponding differences in the recolored image have been enhanced compared to those in the original image, which can maximize the contrast between the different colors in the recolored images.

The differences between a key color C_i^0 and the other colors with a higher confidence $C_j^0 (j \leq i)$ are used as benchmark for judging whether C_i^0 is unrecognizable. If there exists a recolored color with a higher confidence $O_j^d (j \leq i)$, and when the difference between O_j^d and C_i^0 is smaller than the corresponding difference between C_j^0 and C_i^0 , we regard C_i^0 as an unrecognizable color. Then the unrecognizable color C_i^0 is recolored to the recolored key color O_i^d according to (3).

$$O_i^d = sim(C_i^0 + \partial * M_t * (C_i^0 - C_i^d)) \tag{3}$$

Where $sim()$ is the well-known color blind simulation function [25], ∂ is a directional parameter that controls the direction of the recoloring and is initially set to 1. If the recolored color O_i^d has exceeded the range of the RGB color space before obtaining an optimal

recolored color, we move to the opposite direction of the recoloring and reduce the movement according to (4). M_t is a mapping matrix, as proposed in [10], with the initial value M_0 as shown in (5), and at the $t - th$ iteration for one unrecognizable color the matrix M_t is obtained from M_{t-1} as shown in (6), where S is a predefined parameter. Same as in [10] and [34], we set S to 0.05.

$$\partial = -0.75 * \partial \tag{4}$$

$$M_0 = \begin{bmatrix} m_t & m_2 & m_3 \\ m_{4,0} & m_5 & m_6 \\ m_{7,0} & m_8 & m_9 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{5}$$

$$M_t = \begin{bmatrix} m_t & m_2 & m_3 \\ m_{4,t} & m_5 & m_6 \\ m_{7,t} & m_8 & m_9 \end{bmatrix} = \begin{bmatrix} m_t & m_2 & m_3 \\ m_{4,t-1} - S & m_5 & m_6 \\ m_{7,t-1} + S & m_8 & m_9 \end{bmatrix} \tag{6}$$

If the differences between O_i^d and one of the colors with a higher confidence $O_j^d (j \leq i)$ in O^d is lower than the corresponding difference between C_i^0 and C_j^0, we still regard O_i^d as an unsatisfactory recolored color for C_i^0. Then we continue to perform recoloring until the differences between O_i^d and $O_i^d (j \leq i)$ are higher than the corresponding values between C_i^0 and C_j^0.

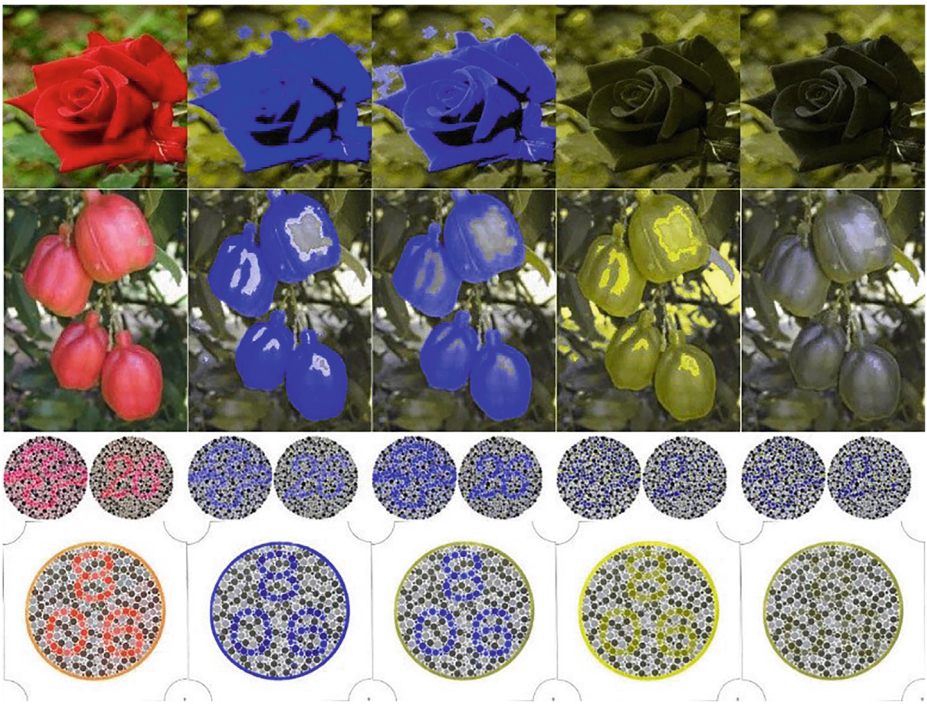

Fig. 12 Comparison when using different weights in (2). Column 1 represents four different test images. Columns 2 to 5 represent the recolored images with weights ($\alpha = 0.2, \beta = 0.2$), ($\alpha = 0.4, \beta = 0.4$), ($\alpha = 0.6, \beta = 0.3$) and ($\alpha = 0.2, \beta = 0.7$), respectively

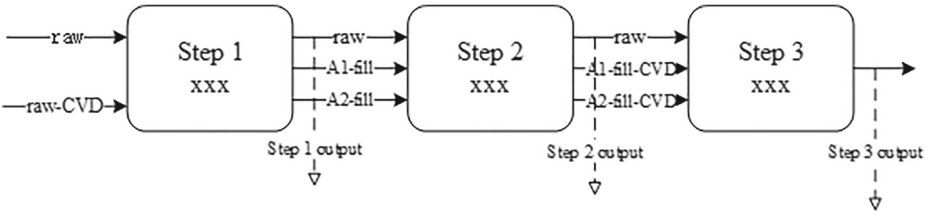

Fig. 13 Flow chart of the Filtering process. Solid arrows indicate input, and dotted arrows indicate output. The output of step 1 are the raw images that cannot be distinguished by CVD. The output of step 2 are the images well filled. Step 3's output is that image that can be distinguished by CVD after coloring, which constructs our datasets

3.3 Data filtering

Based on the above methods of data collection (9 groups of colors) and of two image filling algorithms, each original image, downloaded from the website[2], is referred as raw in the following description. We then generate five different images for each raw image.

First is the color blindness simulation image corresponding to the original image, which is seen by CVD and referred to as raw-CVD.

Second is the image filled by Algorithm 1 proposed by Huang [29] and the image is referred to as $A1\text{-}fill$.

Third is the image filled by the proposed Algorithm 2 in this paper and the image is referred to as $A2\text{-}fill$.

Fourth is the color blind simulation image corresponding to $A1\text{-}fill$, which is seen by CVD and referred to as $A1\text{-}fill$-CVD .

Fifth is the color blind simulation image corresponding to $A2\text{-}fill$, which is seen by CVD and referred to as $A2\text{-}fill$-CVD.

All of the above simulations are generated from website[2]. When inputting an image on this website, it will show a corresponding color blind simulation image.[2]

In order to compare the five images above with the original image quickly, it is necessary to design a tool. The ultimate purpose is to generate our dataset, which contains the original raw image that cannot be distinguished by the CVD and the corresponding fill ($A1\text{-}fill$ or $A2\text{-}fill$) image which can be distinguished by the CVD. The filtering flow chart is given below in Fig. 13.

The arrows in Fig. 13 indicate the input of each step. The output of the previous step serves as the input to the next step.

In the data filtering process, five people with normal color perception are employed to perform the filtering operation at each step and their filtered results are merged. Detailed descriptions of each step are given below.

3.3.1 Image pre-filtering

In order to get more effective data, five individuals are looked for at random to compare each set of our images, which consist of raw, raw-CVD, $A(1,2)\text{-}fill$ ($A1\text{-}fill$ or

[2]website[2]:http://www.color-blindness.com/coblis-color-blindness-simulator/

$A2\text{-}fill$), and $A(1,2)\text{-}fill$-CVD ($A1\text{-}fill$-CVD or $A2\text{-}fill$-CVD). Then a voting strategy is adopted to merge the filtering results of these five individuals.

On the pre-filtering stage, a lot of duplicate and incorrectly recolored images are filtered out. After this operation, there are 2673 sets of images left, each group with 24, 216, 129, 531, 383, 479, 459, 120 and 404 images respectively. Then those images are filtered using our filtering tool (Section 3.2.2).

3.3.2 Step1-recognizable images filtering for CVD

In this step, the raw and the corresponding raw-CVD are the input images, the recognizable images for CVD are filtered out after this comparative process. For an image that is not recognized by CVD, we will draw some bounding boxes on the indistinguishable area. In the following, we introduce the filtering principles, tool usage, and the data statistics, respectively.

Filtering Principles Strictly speaking, the indistinguishability refers to the loss or change of the content in the image, resulting in a reduction of the acquired information. As shown in Fig. 14, on the left is the image seen by normal people, and on the right is the image seen by CVD, it can be seen that the number of 45 in the original image is not perceived by people with CVD. However, the Ishihara test images are usually used for the medical test, and they are not common in daily life. The indiscernibility of the same color can lead to the loss or change of the content in the image. For example, due to the incorrect identification of the orange color next to the yellow flower, color blind people cannot distinguish between the orange flower and the green tree in Fig. 15.

According to this, we use images that are not recognized by CVD as a standard for indistinguishable image pairs. The filtering principle in our dataset is based on the fact that different colors in raw become the same or similar to the colors in the corresponding raw-CVD.

Tool interface and XML file Figure 16 shows the interface of the filtering tool, which includes two radio buttons. The left button represents an image which is unrecognized by

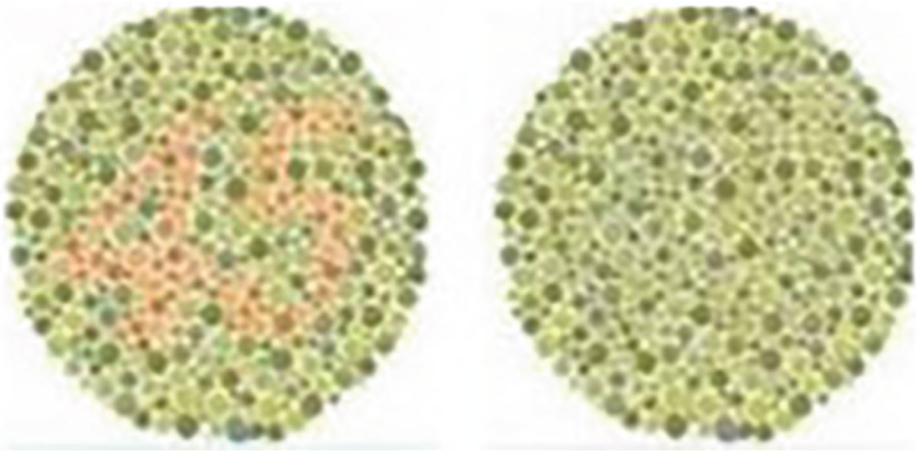

Fig. 14 Ishihara plates, the left is seen by normal people, the right is seen by CVD

Fig. 15 The left is the original image, and the right is the color blind simulation image

CVD, selecting it will set the flag of the image to 1 in the xml file. The right button represents all the colors in the image which can be distinguished by CVD, the flag of the image is set to 0 in the xml file. For undistinguished images, it is required to draw bounding boxes around them. The bounding box is an area which indicates that the pixels in this box are undistinguished by people with CVD. The information of the bounding box is also stored into the xml files which contains the bounding box information, the screener's name and a flag on whether the image has been filtered out.

Statistical results Following the above process, each individual gets an xml file. Then the five xml files are merged using a voting strategy. The process includes two aspects: **First**,

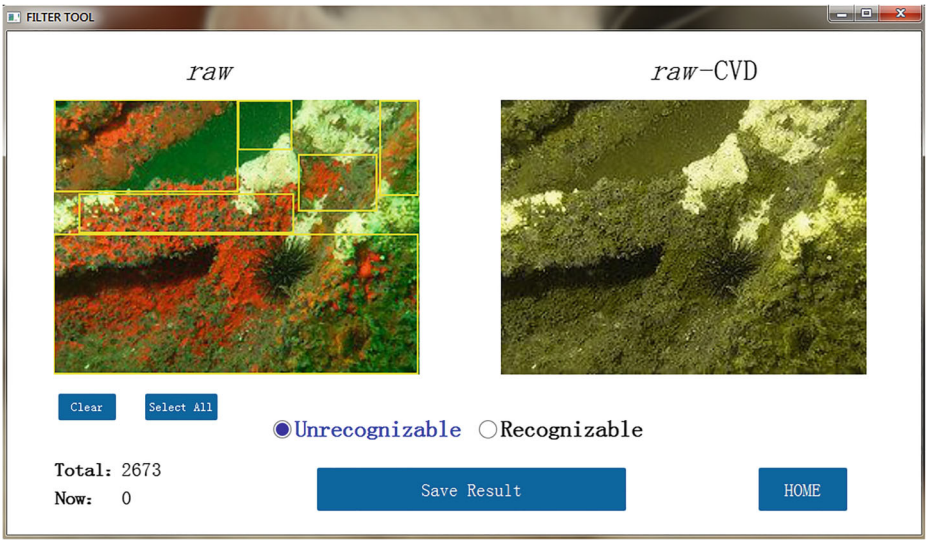

Fig. 16 The interface of step 1

the images will be discarded if more than half individuals consider that this image can be distinguishable by people with CVD. **Second**, the bounding box in the different xml files are merged. The result of the merged bounding boxes is the union of all the bounding boxes in every xml file. According to our statistics, there are 2542 images left after step 1.

3.3.3 Step2-images filtering based on A1-fill or A2-fill

In step2, we feed the remaining 2542 images to proceed to the second round of the filtering process. We use raw and the corresponding $A1\text{-}fill$ and $A2\text{-}fill$ as input. The purpose of this step is to filter out the unsatisfactory recolored images. The filtering principle is introduced and the interface of step2 is shown below.

Filtering principle $A(1,2)\text{-}fill$ are used as the corresponding ground truth for the original image in order to train the deep learning network, that is why that it must be ensured that the content of $A(1,2)\text{-}fill$ is not lost, and the number of the main colors does not change.

The principles of step 2 of the filtering tool are as follows. **First**, compare raw and $A(1,2)\text{-}fill$ to ensure that the content in $A(1,2)\text{-}fill$ is consistent with raw. **Second**, compare raw with $A(1,2)\text{-}fill$ to ensure that the number of primary colors in $A(1,2)\text{-}fill$ is the same as raw. As shown in Fig. 17, the type of color in the image is increased, and this will result in a content that cannot be recognized correctly.

Tool interface and xml file Figure 18 shows the interface of step2, which contains two sets of buttons, corresponding to the $A(1,2)\text{-}fill$ obtained by the above two recoloring algorithms. According to the filtering principle, we compare $A1\text{-}fill$ and $A2\text{-}fill$ with raw respectively to determine whether this image should be filtered out. If the recognizable

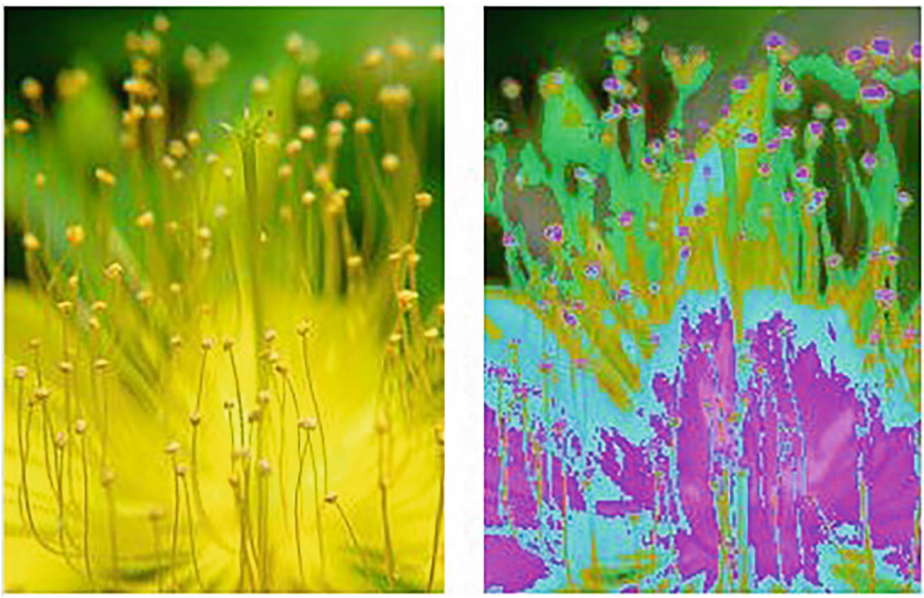

Fig. 17 The left is the original image, and the right is an algorithm-colored image

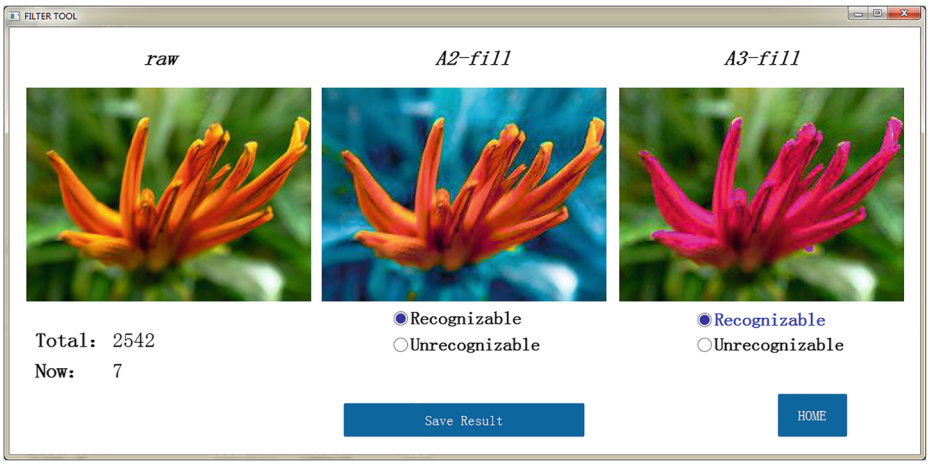

Fig. 18 The interface of step 2

button is selected, $A1\text{-}fill$ or $A2\text{-}fill$ is left and the flag of the image in the xml file is set to 1, otherwise it is set to 0. Different from step 1, this step does not draw a bounding box.

Statistical results Similar to step 1, after the filtering of this step, each individual will get an xml file. Then five xml files are merged by using the strategy of winning and losing. The difference of these xml files are merged, and a new xml is generated. According to our statistics, there are 1959 pairs of raw and $A1\text{-}fill$, and 1329 pairs of raw and $A2\text{-}fill$ left after the filtering of step 2.

3.3.4 Step3-image filtering based on A1-fill-CVD or A2-fill-CVD

In this step, the input is raw and $A1\text{-}fill$-CVD or $A2\text{-}fill$-CVD corresponding to the raw. The purpose of this step is to filter out images that cannot be recognized by CVD after they have been recolored with two recoloring algorithms.

Filtering principle Based on the purpose of this step, the filtering principle is similar to step 1. We use the distinguishable colors pair as the filtering standard. The filtering principle is as follow. By comparing raw and the corresponding $A(1, 2)\text{-}fill$-CVD, we determine whether the color and content in the $A(1, 2)\text{-}fill$-CVD is reduced compared to the raw. An example is shown in Fig. 19. Due to the lack of color information, the content information is reduced.

Tool interface and xml file The tool interface and xml files are similar to step 2.

Statistical results After merging the xml of the five individuals, all the image pairs of the original and the corresponding recolored images are saved in the merged xml file. According to the final statistics, there are 1859 pairs of raw and $A1\text{-}fill$, and 1225 pairs of raw and $A2\text{-}fill$ left.

Fig. 19 The left is the raw image, and the right is $A2\text{-}fill$-CVD

3.4 Data statistics

After the filtering processes above on of Section 3, there are 3084 image pairs left to form our CVD dataset. The proportion of the unrecognized area in the dataset and the corresponding number of images are shown in Fig. 20.

4 Experiments

To evaluate the accuracy of our datasets, we use some popular GAN networks for recoloring. The datasets are divided into a training set and a test set to facilitate the training of the GAN network. The training set has 2313 pairs of images, and the test set has 771 pairs of images. We use the training set to train the GAN networks to generate a certain image recoloring network model, and then use the test set to evaluate the network model. We chose

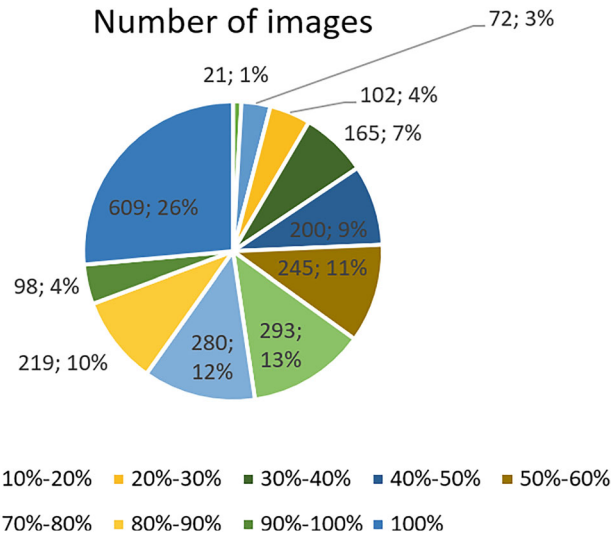

Fig. 20 The proportion of the unrecognized area by CVD in the dataset and the number of images

the popular pix2pix-GAN [32], Cycle-GAN [66] and Bicycle-GAN [67] tests respectively. The experiments proved that pix2pix-GAN [32] produced the best effect in the process of re-coloring the picture compared with the other two GANs.

4.1 GANs

Pix2pix-GAN [32] framework is trained on paired images, which uses a conditional generative adversarial network to learn a mapping from input to output images. Cycle-GAN [66] learns mapping functions between two domains X and Y, including two mappings: G: X→Y and F: Y→X. Bicycle-GAN [67] combines both the cVAE-GAN [39, 40] and cLR-GAN [6, 12, 14] models to enforce the connection between latent encoding and output in both directions jointly and achieved an improved performance. It can produce both diverse and visually appealing results.

4.2 GANs training

During training, for the pix2pix-GAN [32], Cycle-GAN [66] and Bicycle-GAN [67], we set the input image size to 256*256 and train 200 epochs on each model. Other parameters are the default configuration for each network. When testing, the output size of each network is 256*256. For the Bicycle-GAN [67], only one random image is produced for each input image.

4.3 Datasets evaluation

Any method of image processing can lead to the loss of important information and quality in the image. In order to evaluate the effects of different GANs recoloring, we use qualitative evaluation and quantitative evaluations. Quantitative evaluation methods include SSIM and PSNR. This type of evaluation can be used to evaluate recolored images and their corresponding simulated images. Next, we present the results of our qualitative and quantitative evaluations, respectively.

4.3.1 Qualitative results

The qualitative results using several images from our dataset are shown in Fig. 21. The input consists of the original images, our target is the set of images that are recolored using the algorithms mentioned in Section 3.2, pix2pix-GAN [32], Cycle-GAN [66] and Bicycle-GAN [67] are the images recolored through pix2pix-GAN [32], Cycle-GAN [66], Bicycle-GAN [67], respectively. Among these images, we observe that pix2pix-GAN [32] produces the best effect, Cycle-GAN [66] is the worst, and Bicycle-GAN [67] showed significant distortions. Additional results are shown in the Appendix.

4.3.2 Quantitative results

We used two methods to evaluate the quality of images, Peak-Singnal-to-Noise-Ratio (PSNR) and Structural Similarity (SSIM), which are the most popular methods of image quality assessment. The evaluation results of the two assessment methods for the data generated by several GANs and the data filled by the algorithms mentioned in Section 3.2 are shown in Table 1. Target (the data filled by the algorithms) results in the best overall performance of SSIM, followed by pix2pix-GAN [32], Cycle-GAN [66] and Bicycle-GAN [67].

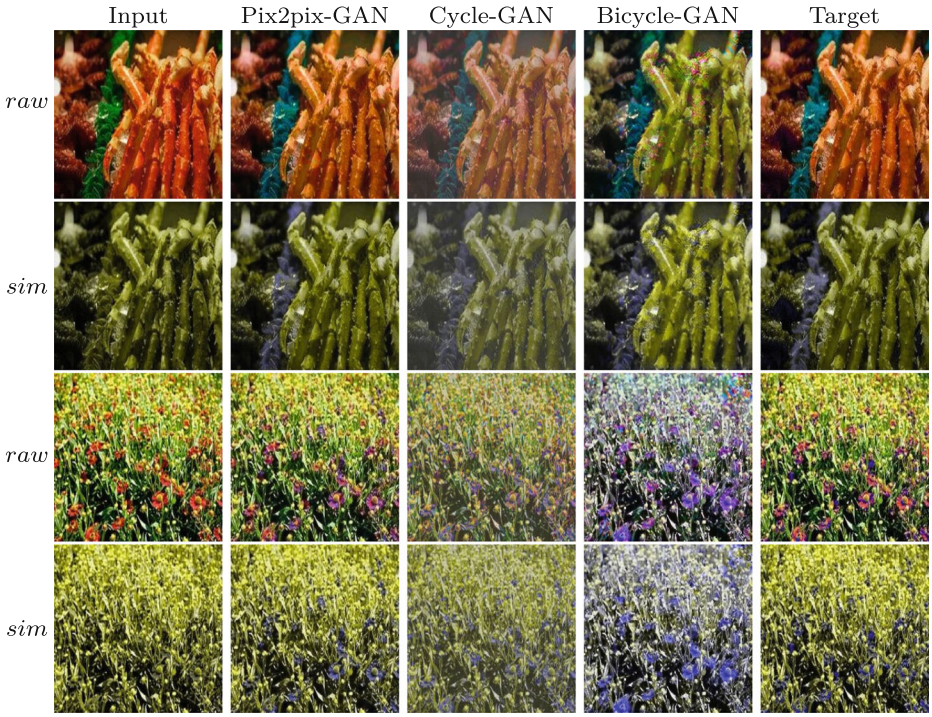

Fig. 21 raw represents the image that normal people can see, sim represents the image seen by CVD. It is observed that pix2pix-GAN produces the best effect, Cycle-GAN produces a worse effect, and Bicycle-GAN shows a significant distortion

These results are consistent with the results described above. However, the evaluation of PSNR is not expected.

SSIM produces an evaluation of the structural similarity between two images. We choose the images filled by the algorithms and generated through GANs, which simulated by color blind simulation to compare with the original images respectively to calculate SSIM. The work we do is to recolor the original image, so that the filled simulation is an image that can be recognized by CVD. The original image seen by CVD is indistinguishable, that is, the contrast of some colors is reduced, resulting in the loss of image content or confusion. Therefore, the closer the simulated image of the filled image is to the original image, the better the filling effect, that is, the higher the SSIM evaluation index. Experiments show that the

Table 1 The evaluation of the data generated by several GAN models and the data filled by the algorithms

Methods	PSNR	SSIM
Pix2pix-GAN	19.5866	0.7469
Cycle-GAN	19.1028	0.7261
Bicycle-GAN	16.1186	0.6384
Target	16.1286	0.7618
Our model	15.2160	0.7568

proposed algorithm fills the best, and the results of the GAN training are also remarkable, which proves that the idea of using deep learning to fill color blind images is practicable.

PSNR is the most widely used objective measure for the evaluation of image quality. In our experiments, we also selected the same data from the above SSIM as input. pix2pix-GAN [32] produces the best evaluation, followed by Cycle-GAN [66], target, Bicycle-GAN [67], and these results were unexpected. Analysis shows that PSNR is an objective criterion to evaluate images but it has certain limitations. In fact, the actual test results show that the PSNR evaluation results are not completely consistent with the visual quality seen by humans. It is possible that a higher PSNR is worse than a lower PSNR. This is due to the fact that the human eye is less sensitive to the spatial frequency, on because the sensitivity of the human eye to the contrast of brightness is higher than that of the color, and the perception of the human eye to a region is affected by the surrounding neighborhood. This leads to differences in PSNR assessment and visual assessments.

5 Conclusion

In this paper, a novel recoloring method is proposed for color vision deficiency. In order to better adapt to the recoloring task, we improved the recoloring methods from four aspects. **First**, a self-adapting recoloring method is proposed. In this method, an improved Octree quantification method is proposed to adaptively obtain the optimal key colors in the original image, which can effectively enhance the color perceptibility for people with CVD. **Second**, a screening tool for CVD datasets was designed to integrate multiple recoloring methods. **Third**, a CVD dataset is constructed. The dataset consists of a training set and a test set. This should be a notable contribution for relevant researchers to make further contributions to the color perception of CVD through the use of deep learning methods. **Fourth**, we use multiple GAN frameworks for colorblind data conversion. Experimental results show that pix2pix-GAN [32] outperforms two other GANs. It is our expectation that this dataset can be helpful for color blind images recoloring with the GANs. In the future, we hope to add prior conditions to the GANs to control the coloring direction of the GANs. For example, for normal vision people, it is easy to identify salient objects in the image, but for color-blind people, it is difficult to do the same thing. From this direction, we can use salient object detection [18, 22, 23, 62] to separate and recolor salient objects in the image from the background, so that color-blind people can distinguish them.

Acknowledgements Supported by National Key R&D Program of China under Grant No. 2019YFB1311600 & Ningbo 2025 Key Project of Science and Technology Innovation (2018B10071).

Appendix A

More recoloring results are shown in this appendix.

References

1. Baker S, Scharstein D, Lewis J, Roth S, Black MJ, Szeliski R (2011) A database and evaluation methodology for optical flow. Int J Comput Vis 92(1):1–31

2. Bansal A, Russell B, Gupta A (2016) Marr revisited: 2d-3d alignment via surface normal prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5965–5974
3. Brettel H, Viénot F, Mollon JD (1997) Computerized simulation of color appearance for dichromats. JOSA A 14(10):2647–2655
4. Cao Z, Simon T, Wei S-E, Sheikh Y (2016) Realtime multi-person 2d pose estimation using part affinity fields, arXiv:1611.08050
5. Chen L-C (2018a) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Patt Anal Mach Intell 40(4):834–848
6. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P (2016) Infogan: Interpretable representation learning by information maximizing generative adversarial nets, NIPS
7. Chen L.-C., Papandreou G, Kokkinos I, Murphy K, Yuille AL (2014) Semantic image segmentation with deep convolutional nets and fully connected crfs, arXiv:1412.7062
8. Chen L.-C., Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation, arXiv:1706.05587
9. Chen L.-C., Zhu Y, Papandreou G, Schroff F, Adam H (2018b) Encoder-decoder with atrous separable convolution for semantic image segmentation, arXiv:1802.02611
10. Doliotis P, Tsekouras G, Anagnostopoulos C-N, Athitsos V (2009) Intelligent modification of colors in digitized paintings for enhancing the visual perception of color-blind viewers. In: IFIP International conference on artificial intelligence applications and innovations. Springer, New York, pp 293–301
11. Dollár P, Zitnick CL (2013) Structured forests for fast edge detection. In: Proceedings of the IEEE international conference on computer vision, pp 1841–1848
12. Donahue J, krähenbühl P, Darrell T (2016) Adversarial feature learning, arXiv:1605.09782
13. Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Van Der Smagt P, Cremers D, Brox T (2015) Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 2758–2766
14. Dumoulin V, Belghazi I, Poole B, Mastropietro O, Lamb A, Arjovsky M, Courville A (2016) Adversarially learned inference, ICLR
15. Eigen D, Fergus R (2015) Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE international conference on computer vision, pp 2650–2658
16. Eigen D, Puhrsch C, Fergus R (2014) Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems, pp 2366–2374
17. Everingham M, Eslami SMA, Gool LJV, Williams CKI, Winn JM, Zisserman A (2015) The pascal visual object classes challenge: a retrospective, Int J Comput Vision 111(1):98–136. [Online]. Available: https://doi.org/10.1007/s11263-014-0733-5
18. Fan D-P, Cheng M-M, Liu J-J, Gao S-H, Hou Q, Borji A (2018) Salient objects in clutter: bringing salient object detection to the foreground. In: Proceedings of the european conference on computer vision (ECCV), pp 186–202
19. Fan D-P, Wang W, Cheng M-M, Shen J (2019) Shifting more attention to video salient object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, conference proceedings, pp 8554–8564
20. Fergus R, Fergus R, Fergus R, Fergus R (2015) Deep generative image models using a laplacian pyramid of adversarial networks. In: International conference on neural information processing systems, pp 1486–1494
21. Fluck D (2006) Coblis - color blindness simulator. [Online]. Available: http://www.color-blindness.com/coblis-color-blindness-simulator/
22. Fu K, Zhao Q, Gu IY-H (2018) Refinet: a deep segmentation assisted refinement network for salient object detection. IEEE Trans Multimed 21(2):457–469
23. Fu K, Zhao Q, Gu IY-H, Yang J (2019) Deepside: a general deep framework for salient object detection. Neurocomputing 356:69–82
24. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on computer vision and pattern recognition (CVPR)
25. Gervautz M, Purgathofer W (1988) A simple method for color quantization: octree quantization. In: New trends in computer graphics. Springer, New York, pp 219–231
26. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
27. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: International conference on neural information processing systems, pp 2672–2680

28. He K, Gkioxari G, Dollár P., Girshick R (2017) Mask r-cnn. In: Computer vision (ICCV), 2017 IEEE international conference on. IEEE, pp 2980–2988
29. Huang JB, Chen CS, Jen TC, Wang SJ (2009) Image recolorization for the colorblind. In: IEEE International conference on acoustics, speech and signal processing, pp 1161–1164
30. Huang CR, Chiu KC, Chen CS (2010) Key color priority based image recoloring for dichromats. Adv Multimed Inform Process - PCM 2010 6298:637–647
31. Huang L, Yang Y, Deng Y, Yu Y (2015) Densebox: Unifying landmark localization with end to end object detection, arXiv:1509.04874
32. Isola P, Zhu J-Y, Zhou T, Efros AA (2016) Image-to-image translation with conditional adversarial networks. CVPR 5967–5976
33. Jégou S., Drozdzal M, Vazquez D, Romero A, Bengio Y (2017) The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation. In: Computer vision and pattern recognition workshops (CVPRW), 2017 IEEE Conference on. IEEE, pp 1175–1183
34. Jeong J-Y, Kim H-J, Wang T-S, Yoon Y-J, Ko S-J (2011) An efficient re-coloring method with information preserving for the color-blind. IEEE Transa Consumer Electron 57(4)
35. Katsuhiro N, Manami T, Hiroshi S, Hiroshi O, Mu S, Atsushi H, Isao M, Shin'Ichi I, Nobuyuki F, Kazunori K (2016) A way of color image processing for the colorblinds. Bull Hiroshima Mercant Marine College 38
36. Khurge DS, Peshwani B (2015) Modifying image appearance to improve information content for color blind viewers. In: Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on. IEEE, pp 611–614
37. Kim H-J, Jeong J-Y, Yoon Y-J, Kim Y-H, Ko S-J (2012) Color modification for color-blind viewers using the dynamic color transformation. In: Consumer electronics (ICCE), 2012, IEEE international conference on. IEEE, pp 602–603
38. Kim YK, Kim KW, Yang X (2007) Real time traffic light recognition system for color vision deficiencies. In: Mechatronics and automation, 2007. ICMA 2007. International conference on IEEE, pp 76–81
39. Kingma DP, Welling M (2013) Auto-encoding variational bayes, ICLR
40. Larsen ABL, Larochelle H, Winther O (2015) Autoencoding beyond pixels using a learned similarity metric. ICML 1558–1566
41. Lu X, Wang W, Ma C, Shen J, Shao L, Porikli F (2019) See more, know more: unsupervised video object segmentation with co-attention siamese networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, conference proceedings, pp 3623–3632
42. Martin DR, Fowlkes CC, Malik J (2004) Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans Patt Anal Mach Intell 26(5):530–549
43. Martin CE, Keller J, Rogers SK, Kabrinsky M (2000) Color blindness and a color human visual system model. IEEE Trans Syst Man Cyber - Part A: Syst Humans 30(4):494–500
44. Mathieu M, Zhao J, Sprechmann P, Ramesh A, Lecun Y (2016) Disentangling factors of variation in deep representations using adversarial training. NIPS 5040–5048
45. Maurer CR, Qi R, Raghavan V (2003) A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. IEEE Trans Patt Anal Mach Intell 25(2):265–270
46. Milić N., Belhadj F, Dragoljub N (2015) The customized daltonization method using discernible colour bins. In: Colour and visual computing symposium (CVCS), 2015. IEEE, pp 1–6
47. Milić N, Hoffmann M, Tómács T, Novaković D, Milosavljević B (2015) A content-dependent naturalness-preserving daltonization method for dichromatic and anomalous trichromatic color vision deficiencies. J Imaging Sci Technol 59(1):10 504–1
48. Orii H, Kawano H, Maeda H, Kouda T (2014) Color conversion algorithm for color blindness using self-organizing map. In: Soft computing and intelligent systems (SCIS), 2014 joint 7th international conference on and advanced intelligent systems (ISIS), 15th international symposium on. IEEE, pp 910–913
49. Rasche K, Geist R, Westall J (2005) Detail preserving reproduction of color images for monochromats and dichromats. IEEE Comput Graph Appl 25(3):22–30
50. Russell BC, Torralba A, Murphy KP, Freeman WT (2008) Labelme: a database and web-based tool for image annotation. Int J Comput Vision 77(1-3):157–173
51. Sharpe LT, Stockman A, Jägle H, Nathans J (1999) Opsin genes, cone photopigments, color vision, and color blindness. Color vision: From genes to perception, 351
52. Shelhamer E, long J, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 3431–3440
53. Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor segmentation and support inference from rgbd images. In: European conference on computer vision. Springer, New York, pp 746–760

54. Stenetorp P, Pyysalo S, Topić G, Ohta T, Ananiadou S, Tsujii J (2012) Brat : a web-based tool for NLP-assisted text annotation. In: Proceedings of the demonstrations session at EACL 2012. Avignon France: Association for computational linguistics
55. Teney D, Hebert M (2016) Learning to extract motion from videos in convolutional neural networks. In: Asian conference on computer vision. Springer, New York, pp 412–428
56. Tzutalin (2015) Labelimg. Git code. [Online]. Available: https://github.com/tzutalin/labelImg
57. Vondrick C, Patterson D, Ramanan D (2013) Efficiently scaling up crowdsourced video annotation. Int J Comput Vis 101(1):184–204
58. Wandell BA (1995) Foundations of vision. Sinauer Associates Sunderland, MA, vol 8
59. Wang X, Fouhey D, Gupta A (2015) Designing deep networks for surface normal estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 539–547
60. Wei S-E, Ramakrishna V, Kanade T, Sheikh Y (2016) Convolutional pose machines. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4724–4732
61. Xie S, Tu Z (2015) Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision, pp 1395–1403
62. Zhao J-X, Liu J-J, Fan D-P, Cao Y, Yang J, Cheng M-M (2019) Egnet: Edge guidance network for salient object detection. In: Proceedings of the IEEE international conference on computer vision, conference proceedings, pp 8779–8788
63. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2881–2890
64. Zhou B, Zhao H, Puig X, Fidler S, Barriuso A, Torralba A (2017) Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition
65. Zhu J-Y, Krähenbühl P, Shechtman E, Efros AA (2016) Generative visual manipulation on the natural image manifold. In: European conference on computer vision. Springer, New York, pp 597–613
66. Zhu J-Y, Park T, Isola P (2017) Unpaired image-to-image translation using cycle-consistent adversarial networkss. In: Computer vision (ICCV), 2017 IEEE international conference on, pp 2242–2251
67. Zhu J-Y, Zhang R, Pathak D, Darrell T, Efros AA, Wang O, Shechtman (2017) Toward multimodal image-to-image translation. In: Advances in neural information processing systems

Hongsheng Li is a student of software engineering, and now he is studying for a Ph.D at Xidian University. His research interest is deep learning, generative adversarial network for recoloring and action/gesture recognition. He is working on combining the traditional statistics-based method and the GAN network to ensure the color balance of the generated images while enabling the color-blind people to distinguish more easily confused colors. His current work in action recognition involves design of novel basic block to efficiently extract correlation-based spatiotemporal feature. This work has been shown to be effective in improving the accuracy of action recognition without adding additional parameters.

Liang Zhang received the Ph.D. degree in instrument science and technology from Zhejiang University, China, in September 2009. In September 2009, he joined the School of Software, Xidian University, where he is currently an associate Professor and the Director of the Embedded Technology and Vision Processing Research Center. He has published more than 40 academic papers in peer-reviewed international journals and conferences. His research interests lie in the areas of multicore embedded systems, computer vision, deep learning, simultaneous localization and mapping (SLAM), human robot interaction, image processing.

Xiangdong Zhang received his M.S. and Ph.D. degrees in school of telecommunication engineering from the Xidian University, China, in 1995 and 1998, respectively. He is currently an associate Professor with the School of Telecommunication Engineering, Xidian University. His research interests are intelligent transportation system (ITS), image processing, and computer vision.

Meili Zhang is now studying for M.S. degree at Xidian University. Her research interest is color vision deficiency.

Guangming Zhu received the Ph.D. degree in instrument science and technology from Zhejiang University, China, in March 2015. He is currently associate Professor at Computer Science and Technology, Xidian University. His major research fields are information fusion, human action/gesture recognition, scene recognition, and deep learning.

Peiyi Shen received the Ph.D. degree from Xidian University, in 1999, and the Ph.D. degree from the MTRC, Computer Science, University of Bath. He was a Research Officer with the MTRC, Computer Science, University of Bath, under the supervision of Prof. P. Willis, and a Research Fellow with CVSSP, University of Surrey, under the supervision of Prof. A. Hilton. He was with Agilent Technologies, USA, U.K., Malaysia, and Singapore, from 2000 to 2003. He is currently a Professor at the Computer Science and Technology, Xidian University. His research interests are in computer vision, volume visualization.

Mohammed Bennamoun lectured in robotics at Queenars, and then joined QUT in 1993 as an associate lecturer. He then became a lecturer in 1996 and a senior lecturer in 1998 at QUT. In January 2003, he joined The University of Western Australia as an associate professor. He was also the director of a research center from 1998lC2002. He is the coauthor of the book Object Recognition: Fundamentals and Case Studies (Springer-Verlag, 2001). He has published close to 100 journals and 250 conference publications. His areas of interest include control theory, robotics, obstacle avoidance, object recognition, artificial neural networks, signal/image processing, and computer vision. He is currently a W/Professor at the School of Computer Science and Software Engineering at The University of Western Australia.

Syed Afaq Ali Shah obtained his Ph.D. from the University of Western Australia in the area of computer vision and machine learning. He is currently working as a research associate in school of computer science and software engineering, the University of Western Australia, Crawley, Australia. He has been awarded arStart Something Prize for Research Impact through Enterprise as for 3D facial analysis project funded by the Australian Research Council. His research interests include deep learning, 3D object/ face recognition, 3D modelling, and image processing.

Affiliations

Hongsheng Li[1] · Liang Zhang[1] · Xiangdong Zhang[1] · Meili Zhang[1] · Guangming Zhu[1] · Peiyi Shen[1] · Ping Li[2] · Mohammed Bennamoun[3] · Syed Afaq Ali Shah[3]

Hongsheng Li
hsli@stu.xidian.edu.cn

Xiangdong Zhang
xdchen@mail.xidian.edu.cn

Meili Zhang
1334625816@qq.com

Guangming Zhu
gmzhu@xidian.edu.cn

Peiyi Shen
pyshen@xidian.edu.cn

Ping Li
pli@bnc.org.cn

Mohammed Bennamoun
mohammed.bennamoun@uwa.edu.au

Syed Afaq Ali Shah
afaq.shah@uwa.edu.au

1 Xidian University, Xi'an, China

2 Shanghai National Engineering Research Center for Broadband Networks & Applications, Shanghai, China

3 The University of Western Australia, Perth, Australia