Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

SmartPaint: a co-creative drawing system based on generative adversarial networks*

Lingyun SUN[†1,2,3], Pei CHEN[†1,2,3], Wei XIANG[†‡1,2,3], Peng CHEN[1,2,3], Wei-yue GAO[1,2,3], Ke-jun ZHANG[1,3]

[1]*Key Laboratory of Design Intelligence and Digital Creativity of Zhejiang Province, Hangzhou 310027, China*

[2]*State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China*

[3]*Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310027, China*

[†]E-mail: sunly@zju.edu.cn; chenpei@zju.edu.cn; wxiang@zju.edu.cn

Received July 30, 2019; Revision accepted Dec. 8, 2019; Crosschecked Dec. 24, 2019

Abstract: Artificial intelligence (AI) has played a significant role in imitating and producing large-scale designs such as e-commerce banners. However, it is less successful at creative and collaborative design outputs. Most humans express their ideas as rough sketches, and lack the professional skills to complete pleasing paintings. Existing AI approaches have failed to convert varied user sketches into artistically beautiful paintings while preserving their semantic concepts. To bridge this gap, we have developed SmartPaint, a co-creative drawing system based on generative adversarial networks (GANs), enabling a machine and a human being to collaborate in cartoon landscape painting. SmartPaint trains a GAN using triples of cartoon images, their corresponding semantic label maps, and edge detection maps. The machine can then simultaneously understand the cartoon style and semantics, along with the spatial relationships among the objects in the landscape images. The trained system receives a sketch as a semantic label map input, and automatically synthesizes its edge map for stable handling of varied sketches. It then outputs a creative and fine painting with the appropriate style corresponding to the human's sketch. Experiments confirmed that the proposed SmartPaint system successfully generates high-quality cartoon paintings.

Key words: Co-creative drawing; Deep learning; Image generation
https://doi.org/10.1631/FITEE.1900386 **CLC number:** TP391

1 Introduction

With recent advances in artificial intelligence (AI), machines can increasingly perform creative tasks such as musical composition (Roberts et al., 2017), writing (Bowman et al., 2016), and design (Zhao et al., 2018). Because it can imitate design and produce large-scale work, AI is commonly employed in the automated design of large-scale forms such as e-commerce banners (Zhang et al., 2017). However, collaborative efforts between AI and humans that support creative expression with varied design outputs have been little investigated. Drawing is among the most important processes for exploring creative ideas. To create a pleasing painting, an artist coherently organizes the visual features of a painting to express his/her intentions and imagination. This creative process requires professional knowledge and advanced artistic skills. To collaborate with humans, a machine must master the domain knowledge of a certain painting. It can then understand the

‡ Corresponding author

* Project supported by the National Science and Technology Innovation 2030 Major Project of the Ministry of Science and Technology of China (No. 2018AAA0100703), the National Natural Science Foundation of China (No. 61672451), the Provincial Key Research and Development Plan of Zhejiang Province, China (No. 2019C03137), the China Postdoctoral Science Foundation (No. 2018M630658), and the Ng Teng Fong Charitable Foundation in the form of ZJU-SUTD IDEA Grant

ORCID: Lingyun SUN, http://orcid.org/0000-0002-5561-0493; Wei XIANG, http://orcid.org/0000-0003-2058-5379

semantic concepts of the user sketch and transform a rough sketch into a vivid painting with a specific artistic style.

To understand a user's input, previous studies identified the subject of the user's drawing by sketch recognition techniques, and then used the information to provide real-time interactive feedbacks such as suggestive instructions (Dixon et al., 2010), similar sketch recommendations (Lee et al., 2011), and improvised responses (Davis et al., 2016a). These approaches attempt to guide and inspire users to achieve their drawing goals step by step, but they only learn the semantics of freehand sketches, without converting the sketches into aesthetically pleasing paintings. Therefore, they cannot simultaneously handle the semantic understanding and stylization of user inputs. Other methods automatically form pleasing paintings with an artistic style from user inputs. For instance, photographs can be transformed into the style of famous paintings (Gatys et al., 2016), or images can be generated from sketch boundaries and sparse color strokes (Liu et al., 2018). Users of these systems can effortlessly obtain fascinating images from photographs or from sufficiently fine sketch boundaries extracted from photographs. However, as these systems cannot cope with variations in the input sketches, they severely limit the users' creativity.

The semantics and artistic style of a painting are inseparable, and the semantics are closely related to the texture and color of each object in the painting. In this study, we introduce SmartPaint, a co-creative drawing system that creates cartoon landscape paintings through machine-human collaborations. To realize SmartPaint, we trained a generative adversarial network (GAN) with 7234 triples of cartoon landscape images and their corresponding semantic label maps and edge detection maps. After model training, the machine can simultaneously understand the cartoon style and semantics, and can identify the spatial relationships among objects in the landscape images. The edge detection maps input to the network enhance the generation quality of our system. To maintain the simplicity of freehand sketching, a user inputs sketches as semantic label maps, and SmartPaint automatically synthesizes edge maps based on the semantics of sketches. The user sketches and synthesized edge maps are then passed to the trained GAN. Several seconds later, GAN outputs cartoon paintings with appropriate colors and textures. By this process, SmartPaint allows both novices and experts to freely input their creative ideas as rough sketches and obtain artistically expressive paintings. Experiments confirmed that the proposed SmartPaint system generates high-quality cartoon paintings. Some of the paintings produced by our system are displayed in Fig. 1.

This study makes the following contributions to the literature. We first introduce SmartPaint, a co-creative drawing system that enables a machine and a human being to cooperatively create cartoon landscape paintings through GANs. Users of this system can freely express their inspiration while guaranteeing high-quality output paintings. The machine simultaneously learns both the semantics and style of a certain type of painting. User inputs are interpreted and transformed into paintings by the computer. Second, we synthesize edge maps based on the semantics of user's input sketches. Our synthesis method enhances the generation quality of paintings and bridges the gap between user inputs and computer's understanding. Third, we propose a set of design principles for developing co-creative machine-human systems.

2 Related work

In creative tasks, a computer colleague collaborates with human users by making independent

Fig. 1 Paintings produced by our system. In each example, we show the input sketch and its generated painting

contributions to a shared creative product (Davis, 2013). In a drawing task, the co-creative agent needs to understand the users' drawing inputs, respond to those inputs, and help users transform their ideas into beautiful paintings.

2.1 Understanding user inputs

The drawing system should complement a user's drawings in a way that reflects the intent of the user, and hence "makes sense" to the user (Davis et al., 2016b). Previous attempts interpreted freehand drawings through sketch-recognition algorithms, and then offered drawing guidance or recommendations. For example, EyeSeeYou (Cummmings et al., 2012) combines sketch recognition and simple eye-drawing techniques in a six-step procedure that provides accurate eye renditions. The iCanDraw system (Dixon et al., 2010) assists users in drawing human faces by recognizing their sketches and providing instructions and corrective feedbacks. ShadowDraw (Lee et al., 2011) updates the shadow image underlying the current user's strokes, and guides the free drawing of objects. Draw apprentice (Davis et al., 2015, 2016a) collaborates with users in abstract art creations. This method groups and classifies the input sketches of users, and draws objects that resemble the user's most recently drawn object. The system proposed by Karimi et al. (2018) recognizes human sketches and matches them to structurally similar sketches in different categories to improve creativity of outputs. Sketch-RNN (Ha and Eck, 2017) models sketch drawings by recurrent neural networks and generates drawings from unfinished sketches. Sketch2Photo (Chen et al., 2009) and Photosketcher (Eitz et al., 2011) retrieve portions of images from a dataset based on user sketches to composite photorealistic pictures.

By recognizing users' inputs, these methods suggest approaches for improving drawings, or return image results satisfying the specified requirements. All approaches enhance the details of ideas expressed by users. However, creators need a beautification partner that allows free expression through rough sketches while beautifying the drawing into a pleasing painting. Moreover, the above methods recognize only a single stroke or individual objects, and lack an overall understanding of paintings. Therefore, they cannot support the creation of expressive paintings with rich content.

2.2 Turning a sketch into a painting

To automatically convert a user's inputs into paintings with an artistic style, traditional drawing tools rely on physical virtual brush modeling and stroke-based rendering. Chu and Tai (2004) designed a virtual Chinese brush model that creates brushwork through dynamic interactions with a paper surface. Ning et al. (2011) introduced a contour-driven approach which renders user input contours as ink paintings. These rule-based methods usually predefine limited stroke styles and place a set of constraints on the drawing process. Therefore, they are domain-specific and lowly applicable to open-ended problems in drawing.

Recent studies on deep learning have successfully completed a wide range of image synthesis tasks, obtaining images with rich details and various artistic styles within a short period of time. Gatys et al. (2016) proposed the neural style transfer (NST) algorithm, which reproduces oil painting styles on natural images using a convolutional neural network. Later studies applied NST to photorealistic image stylization (Luan et al., 2017), portrait style transfer (Selim et al., 2016), font style transfer (Atarsaikhan et al., 2017), video style transfer (Huang et al., 2017), and other functions. Our work is inspired mainly by neural doodle (Champandard, 2016), an NST-based technique that transforms rough sketches into artworks. In this method, doodles are interpreted as segmentation maps for semantic style transfer. User interaction with neural doodle differs from that of inputting photos. However, like other NST methods, neural doodle does not build an understanding of the semantics or style of paintings, and migrates only the low-level color and texture while omitting the high-level image styles. Consequently, the results of NSTs depend heavily on the chosen style image, and are not aesthetically pleasing for all input styles.

GANs, introduced by Goodfellow et al. (2014), have proved their versatility in automated image generation. A GAN trains two networks, a generator network and a discriminator network. The generator aims to fool the discriminator by generating images that are indistinguishable from real images. The discriminator attempts to classify "real" and "fake" images. The generator and discriminator are jointly trained by solving the following min-max

optimization problem:

$$\min_G \max_D V(G,D) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))], \quad (1)$$

where $\boldsymbol{x}$ is a natural image drawn from the true data distribution $p_{\text{data}}(\boldsymbol{x})$, and $\boldsymbol{z}$ is a latent random vector sampled from a uniform distribution.

Several GAN-based methods provide rapid image generation from input sketches. Zhu et al. (2016) developed an interactive GAN (iGAN) that manipulates images under user constraints. Dekel et al. (2018) and Portenier et al. (2018) presented systems for sketch-based face image editing. Isola et al. (2017) proposed the pix2pix framework, which realistically draws shoes, handbags, building facades, and street scenes from sketches or blocks of color. Later works controlled image generation results by placing additional limitations on the inputs. For example, Xian et al. (2018) guided image synthesis through sketches and texture patches. The Scribbler system developed by Sangkloy et al. (2017) allows users to specify their preferred colors of objects. Zhu et al. (2017) proposed that besides paired image-to-image translation, unpaired image-to-image translation is viable when the paired training data are hard to obtain. Li et al. (2018) further proposed that stacked cycle-consistent adversarial networks can improve the image translation quality by decomposing a complex image translation into multi-stage transformations. However, in sketch-based image generation methods, the input sketches require similar characteristics to the training sketches, which might be creatively prohibitive for users.

Cartoons are an artistic form with wide relevance to daily life. Although creating cartoon stories is appealing to many people, it is very time-consuming and requires artistic skills. Many technologies can remove the redundancy in cartooning work, such as sketch simplification of raster images (Simo-Serra et al., 2016), automatic anime line art colorizing (Ci et al., 2018), and photo cartoonization (Chen et al., 2018).

These methods reduce the complexity and time of obtaining plausible image results, but limit users' inputs to the format of the training data (i.e., photographs or sketch boundaries) without attempting to understand users' expressions. Therefore, expressive and personalized paintings cannot be created by these methods. Our work also builds on image-to-image translation algorithms. User-input understanding and stylization are achieved by allowing the machine to learn the semantics and style of paintings simultaneously, and by synthesizing edge maps based on the understanding of user sketches. The edge maps fill the gap between the user input and machine understanding.

3 System overview and implementation

3.1 System overview

In SmartPaint, humans and machines collaborate in the creation of cartoon landscape paintings. Humans express a creative idea in a rough sketch, and machines turn the sketch into a cartoon painting and return it to the shared canvas. Our system is overviewed in Fig. 2. Human-machine collaboration is realized through three key functionalities: a painting producer, an edge synthesizer, and a reference recommender.

1. Painting producer

The painting producer consists of a conditional GAN (Mirza and Osindero, 2014) trained on cartoon images, their corresponding semantic label maps, and edge detection maps. The semantic label maps provide the semantic information in each region of the image, and the edge maps offer the details of each object. Combining both inputs as the network input empirically improves the quality of paintings generated by the computer.

2. Edge synthesizer

To ensure a simple drawing task, the proposed system takes user sketches with different colors as semantic label maps. The edge synthesizer automatically synthesizes edge maps from the user's input sketches. The user-drawn sketches and synthesized edge maps are then input into the trained GAN to generate paintings.

3. Reference recommender

Real-time guidance and visual feedback during the creative process can inspire the user's creativity. Given an arbitrary sketch input, SmartPaint recommends the eight most similar semantic label maps in the dataset as references for users.

Fig. 3a illustrates the main interface of our system. The interface consists of three components:

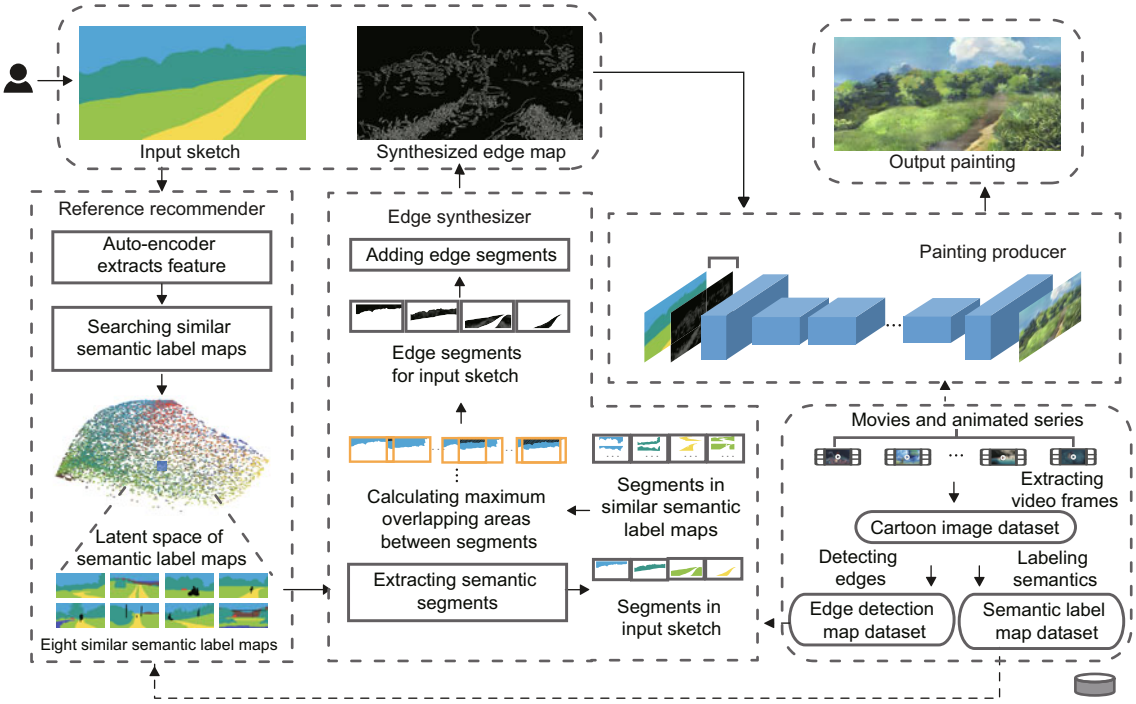

Fig. 2 Overall structure of SmartPaint

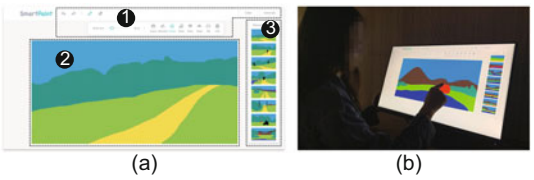

Fig. 3 The system display of SmartPaint: (a) SmartPaint interface (area 1: palette of drawing tools; area 2: shared canvas; area 3: display for the reference images); (b) a user drawing in the system

a palette of drawing tools (area 1), a shared canvas on which humans and machines can draw (area 2), and a display for the reference images (area 3). The drawing palette includes the common functions associated with drawing applications, such as brush selection. When drawing objects, users select from eight kinds of brushes with different colors representing mountains, grass, trees, houses, sky, rivers, roads, and stones. Using these brushes, users draw the outline held in their imagination. There are two modes in our system, i.e., drawing mode for constructing and modifying sketches and generation mode for displaying generated paintings on the canvas. If the user is dissatisfied with the result, he/she can return to the drawing mode to modify the sketches. The system is built in a python-based web framework. Its usage is illustrated in Fig. 3b. In the following subsection, we detail the implementation of our three functionalities.

3.2 System implementation

3.2.1 Painting producer

1. Training data

The training data contain Japanese cartoon images, their corresponding semantic label maps, and edge detection maps. All training images were resized and cropped to 1024×512 pixels. To obtain a set of cartoon images, we extracted frames of cartoon films drawn by Hayao MIYAZAKI and other Japanese films or animations with a similar style. The film and animation data are listed in the Appendix.

Our image data were limited to landscapes; images with obvious characters or artificial objects were discarded. For semantic labeling of the cartoon images, we grouped visual classes in the dataset into nine categories, that is, mountains, grass, trees, houses, sky, rivers, roads, stones, and others (annotated in black); each category was represented by a different color. The edges of the cartoon images were extracted by a standard Canny edge detector (Canny, 1987). Our final dataset consisted

of 7234 data triples, each describing a cartoon image, a semantic label map, and an edge detection map (Fig. 4).

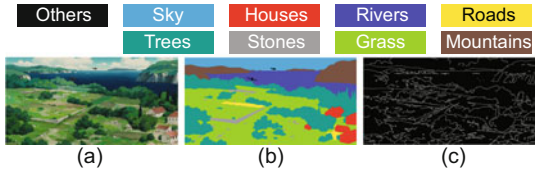

Fig. 4 Examples of our associated data in the training dataset: (a) cartoon image; (b) semantic label map; (c) edge detection map. References to color refer to the online version of this figure

2. Painting generation network

Our generation network was built in the pix2pixHD framework (Wang et al., 2018), which is also a conditional GAN framework for image-to-image translation. pix2pixHD achieves a higher resolution and more accurate image generation than pix2pix does, and uses three discriminators (D_1, D_2, and D_3) for operations on different image scales. Of the two generator networks in pix2pixHD, the global generator framework alone was selected for training our painting generator. The objective of generator G is to convert the semantic label maps and edge detection maps into cartoon images. The edge detection maps are concatenated with the semantic label maps and fed into the generator network. Meanwhile, the objective of discriminator D is to distinguish real images from the translated ones.

pix2pixHD combines GAN loss $\mathcal{L}_{\text{GAN}}$ and discriminator-based feature matching loss $\mathcal{L}_{\text{FM}}$ into the objective function. Güçlütürk et al. (2016) showed that adding a VGG perceptual loss $\mathcal{L}_{\text{con}}$ to the objective function is beneficial for image generation tasks. We also found that perceptual loss improves the generated results in our experiments. The final objective function of our painting generation network is given by

$$\min_{G}\left(\left(\max_{D_1,D_2,D_3}\sum_{k=1}^{3}\mathcal{L}_{\text{GAN}}(G,D_k)\right)+\lambda\left(\sum_{k=1}^{3}\mathcal{L}_{\text{FM}}(G,D_k)+\sum_{k=1}^{3}\mathcal{L}_{\text{con}}(G,D_k)\right)\right), \quad (2)$$

where λ controls the weight of each term.

Our models were trained for 200 epochs on a single GTX 1080 Ti GPU. The runtime was approximately one week.

3.2.2 Reference recommender

Based on user-drawn sketches, we guided the user's creative process by recommending similar semantic label maps to those of the user's sketch. To this end, we computed features of the semantic label maps and compared their similarities. Because sketches are diverse, their semantics cannot be represented by low-dimensional features such as the object shapes.

To determine the similarities of semantic label maps, we trained an auto-encoder to extract high-dimensional features from the images. The architecture of our auto-encoder is illustrated in Fig. 5. The features were extracted through multiple convolutional layers. To ensure the speed and accuracy of feature extraction, we resized all semantic label maps to 512×256 pixels, and converted them into one-channel grayscale images before training the network. The auto-encoder outputs a 4096-dimensional latent vector as semantic label map representation. The 4096-dimensional vector is reduced to a 20-dimensional vector by the principal component analysis (PCA) algorithm, and the Euclidean distance between two 20-dimensional vectors is defined as the difference between two semantic label maps. When a user draws a stroke, the current sketch image is passed into the trained auto-encoder, and the above operation repeats. This process obtains eight similar semantic label maps as the user-drawn sketch in real time. When the user does not draw over the entire canvas, the blank pixels are automatically filled by referring to the nearest colored pixels, as detailed in Algorithm 1.

3.2.3 Edge synthesizer

The synthesized edge map should have cartoon-edge characteristics, should match the semantics of the scene represented by the sketch, and should vary the edges for objects in different scenes. The trained

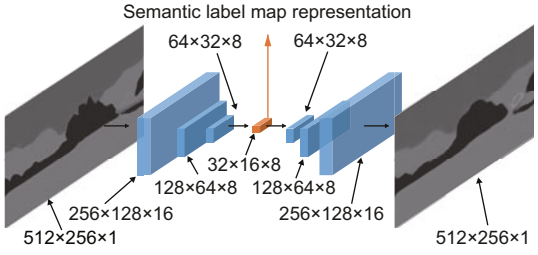

Fig. 5 Structure of the auto-encoder

Algorithm 1 Infilling the blank pixels of the user input

Input: user input sketch I
Output: user sketch without blank pixels I_{result}
1: blanks ← the blank pixels in I
2: **for** each pixel p in blanks **do**
3: Search for the closest colored pixels above, below, to the left, and to the right of p, located at distances of top_{dis}, down_{dis}, left_{dis}, and $\text{right}_{\text{dis}}$ from p, respectively, and with colors of top_{col}, down_{col}, left_{col}, and $\text{right}_{\text{col}}$, respectively
4: p_{col} ← the corresponding color of $\min(\text{top}_{\text{dis}}, \text{down}_{\text{dis}}, \text{left}_{\text{dis}}, \text{right}_{\text{dis}})$
5: **end for**
6: return I_{result}

painting producer outputs a good result only when the input edge map has the same line characteristics as edge maps in the training dataset. This strict requirement on the edge map cannot be satisfied by generative networks, which generate blurred lines and varied line thicknesses. Alternatively, the edge synthesizer can search the photograph that is most consistent with the semantics of the user sketch, and use its edge detection map as the edge map of the sketch. Unlike cartoon edges, which are artificial and regular, photograph edges lack smooth orientations and clear signals along most parts of the curve (Chen et al., 2016). Therefore, we developed a semi-parametric strategy that synthesizes new edge maps for sketches using the edge maps in the training dataset. The strategy provides varied edges for objects in different scenes. For example, if the "tree" brush represents a single large tree in one scene and a dense forest in another, the two scenes will have different edge maps. Using the reference recommender, our strategy obtains eight semantic label maps from the training dataset with the highest similarity to the newly drawn one. The process is detailed below.

First, eight edge maps corresponding to eight similar semantic label maps were input to the edge synthesizer. Second, the semantic segments in the user sketch (T_{label}) and eight similar semantic label maps (S_{label}) were extracted by the shape context algorithm (Belongie et al., 2001). Third, for each segment t in T_{label}, we computed the maximum overlapping area moa_s between it and each segment s in ss, respectively, where ss is a set of segments in S_{label}, and each s in ss has the same semantics as t. Fourth, we selected the maximum value in moa_s, and used the corresponding edge map of s to compute the individual edge map iSE of t. After computing the individual edge map of every segment in the user sketch by the same method, we added all edge segments to obtain the final synthesized edge map. The pseudocode of edge map synthesis is shown in Algorithm 2.

4 Experiment

4.1 Comparison with the original pix2pixHD method

Our approach was compared with the original pix2pixHD method in a qualitative analysis. Fig. 6 demonstrates the effectiveness of our approach for generating various scenes. We trained the original pix2pixHD on our dataset for 200 epochs, using only the cartoon images and their semantic label maps.

Algorithm 2 Edge map synthesis

Input: target semantic label map T_{label} (user sketch), eight similar semantic label maps S_{label}, and their corresponding edge detection maps S_{edge}
Output: synthesized edge map SE for T_{label}
1: Nse ← number of semantics in user sketch
// Label maps are input as matrices. Each pixel takes
// a value in $[1, \text{Nse}]$ with different values representing
// different semantics
2: Using the shape context algorithm to extract semantic segments from T_{label} and S_{label} based on the semantics
3: **for** se ← 1 to Nse **do**
4: ts ← segment set of T_{label} with pixel value se
5: ss ← segment set of S_{label} with pixel value se
6: **for** t in ts **do**
7: M, N ← height and width of t, respectively
8: **for** s in ss **do**
9: moa_s ← the maximum overlapping area between t and s; the offset of s from t is (w, h)
10: **end for**
11: result_area ← (s, w, h), where s is the segment with the maximum moa_s
// compute the individual edge map iSE for t
12: mask ← a matrix $\in (0,1)^{M \times N}$, where the area of t is 1, and the other area is 0
13: iSE ← edge map corresponding to result_area
14: SE ← SE + iSE $\odot$ mask
15: **end for**
16: **end for**
17: return SE

We then drew four sketches of different scenes and processed them independently in pix2pixHD and the proposed method. The images generated by the original pix2pixHD method were blurred and objects were distinguished by their color differences only; almost no detailed textures were depicted. In contrast, our approach generated images with a realistic cartoon style and sharp details (Fig. 6d). It also obtained images with different scenes based on the sketch layouts. For example, it interpreted large areas of trees as forests and small areas of trees as bushes.

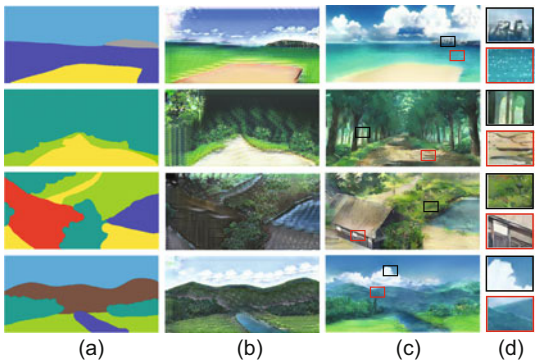

Fig. 6 Visual comparison of our method and the original pix2pixHD method. Given input sketches (a), pix2pixHD (b) and our method (c) generate paintings independently. The right panel (d) shows two magnified insets of (c)

4.2 System evaluation

The SmartPaint system was presented during a public exhibition. Among more than 300 participants who used the system, we invited a subset into our user studies. The usability of SmartPaint was assessed in two user studies with two main goals: (1) to evaluate the performance of SmartPaint in collaborative drawing with both novices and experts, and (2) to explore whether SmartPaint can support artists or designers in real art/design practice. Our evaluation criteria were based on the nine design goals proposed by Benedetti et al. (2014) for the design of art creativity supported tools for novices, and the 15 criteria used by Oh et al. (2018) to understand the user experience of co-creation with AI. Our 10 criteria of a high-performance collaborative system are easy to use, friendly, communicative, flexible, timely, predictable, personalized, satisfying, fun, and effective.

4.2.1 Study 1 methodology

We invited 20 people (10 drawing experts and 10 novices) to participate in study 1. Organized into two sections, this study evaluated whether our system supports creativity of both novices and experts. In the first section, participants drew sketches without subject restrictions. In the second section, the participants were requested to sketch a given text description. Participants were required to create each painting within 3 min. After completing the drawings, the subjects completed a questionnaire related to the evaluation criteria. Each of the 10 questions was answered on a five-point Likert scale, with five being the optimal response. Finally, the reasons for the participants' ratings and other opinions on our system were obtained in interviews.

4.2.2 Study 1 results

Overall, the participants were satisfied with our system. The average score of each item was above three, and five items scored more than four (easy to use, friendly, flexible, timely, and fun). Participants indicated that smearing "color blocks" lowers the difficulty of drawing. Although we set time constraints, both experts and novices created wonderful paintings. The open-ended drawing results, created with no subject restrictions, are shown in Fig. 7, and the drawing results according to the text descriptions are shown in Fig. 8. The experts in both cases created higher-quality paintings than novices, probably because our system was trained on professional drawings extracted from movies, and experts will plan better layouts than novices do even on rough sketches. Our system generated paintings with diverse scenes, such as pastoral, seaside, and forest scenes. Moreover, both experts and novices created paintings with innovative scenes, for example, "Dream island at sea" (Fig. 7d) and "Castle in the sky" (Fig. 7h). Immediate visual feedback encouraged the participants to "try a variety of possibilities quickly" and "preview ideas at the lowest cost." Participants described the generated paintings as "dreamy," "harmonious," "hierarchical," and "like master paintings." However, both novices and experts expected SmartPaint to support more styles and scene types of painting. Participants reported that SmartPaint often returned unexpected results (mean: 3.4, std: 0.821) and they struggled with

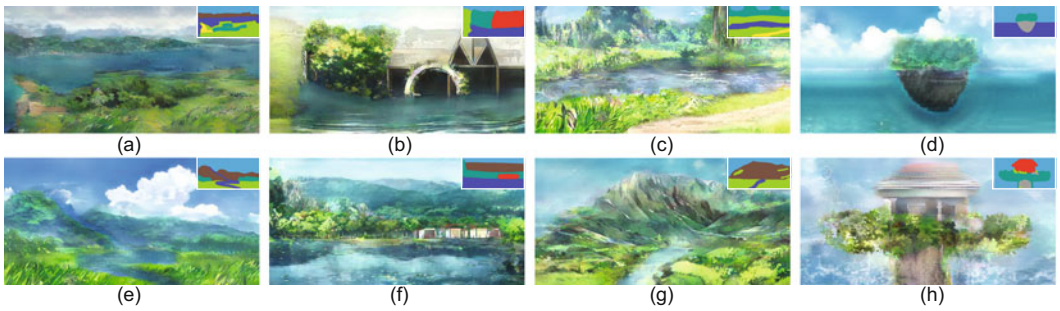

Fig. 7 Paintings from user study 1, created by participants with no restrictions on subject. (a)–(d) are paintings created by novices and (e)–(h) are paintings created by experts. In each example, we show the sketch drawn by the participant and its generated painting

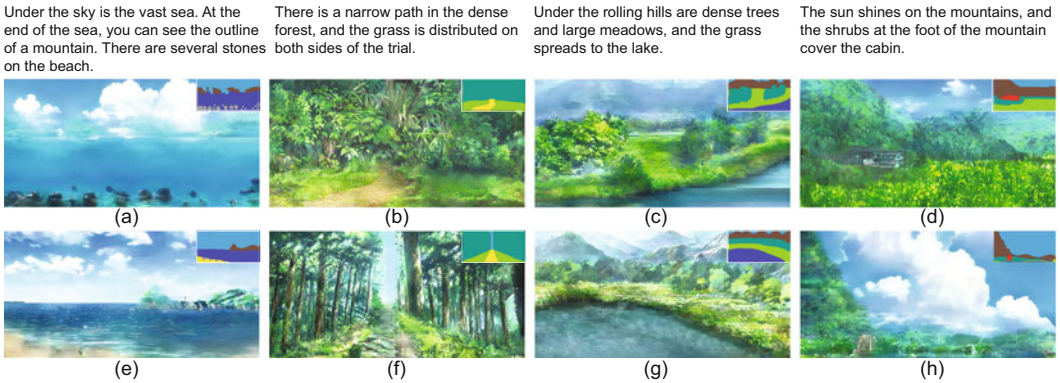

Fig. 8 Paintings created by the participants following text descriptions. (a) and (e), (b) and (f), (c) and (g), and (d) and (h) are pairs of paintings created by participants after reading the same text descriptions. (a)–(d) are created by novices and (e)–(h) are created by experts. In each example, we show the sketch drawn by the participant and its generated painting

understanding its logic. Participants gave mixed responses to this uncertainty. Some participants mentioned that owing to this uncertainty, drawing with the proposed system was a process of inspirational collision that greatly stimulated their creative inspiration. Others insisted that the uncontrollability prevented them from creating the desired effect. Finally, three participants reported a low sense of achievement when co-drawing with SmartPaint, because the system was the main contributor to the paintings.

4.2.3 Study 2 methodology

Our second study explored whether our system can assist designers in real design practice. For this purpose, we invited a professional illustrator to create several paintings with SmartPaint. This illustrator has extensive experience in commercial illustration and has created over one thousand illustrations. The study lasted 4 h. The illustrator was introduced to the proposed system, and after half an hour of familiarization, was instructed to create five paintings. He first worked with SmartPaint to complete the draft, and then optimized the draft in Photoshop. The main operations applied in the optimization were "enhancing image edges," "modifying details," and "adding other elements." The optimization times of each painting (from left to right) are 10, 20, 25, 15, and 15 min, respectively.

4.2.4 Study 2 results

The illustrator believed that SmartPaint was a meaningful attempt to explore the collaborative creation between AI and humans, and expressed surprise at its ability. Fig. 9 shows the paintings created by the illustrator. The first row contains the source paintings created by him and our system, and the second row contains the adjusted paintings after optimization in Photoshop. The illustrator considered that the paintings produced by SmartPaint were

Fig. 9 Paintings produced by a professional illustrator using our system. First row: source paintings created by the illustrator and our system. Second row: adjusted paintings optimized by the illustrator in Photoshop. The optimization times of each painting (from left to right) are 10, 20, 25, 15, and 15 min, respectively

far below his own drawing standards, and that the system was still unsure about some cartoon drawing rules and details. For example, the system failed to understand that in real cartoons, the blue of the sky tends to slowly fade from top to bottom. He pointed out that SmartPaint currently supports only the creation of "normal" paintings, whereas he prefers to create paintings with strong visual impacts and unique perspectives. Therefore, he believed that the current system could not work perfectly with a mature designer, and requires training on more image data.

In the drawing process, the illustrator showed a stronger desire to control the painting results than the participants of study 1. He considered that the unexpected results could free his thinking and provide enlightenment, but introduced uncertainty into his creations. As the illustrator did not know why SmartPaint returned a particular result, he described his cooperation with the system as "temptation." He hoped to exert more dimensional control over the painting; for example, he wished to leave blank spaces, and also designate the tone of the painting, the color of each object, and the perspective of the objects. Moreover, he preferred that computers would finish the tedious work rather than provide inspiration.

5 Discussion

Under our method, the computer learns a large number of images of a particular style, along with their semantic label maps and edge detection maps, and generates specific style images with rich content details. Unlike existing approaches that separately generate and stylize realistic images, our approach produces images that align with a certain style. The aim of our work is to allow human-machine collaborations in drawing tasks and require only the simplest input from humans. To this end, we added the edge detection maps as the network input and trained an auto-encoder to extract the high-dimensional features of semantic label maps, so that we can synthesize edge maps based on the semantics of user input sketches when generating paintings. This method further improves the quality of the generated image and guarantees the freedom and simplicity of the user input.

Our method is suitable for drawing free objects; objects with specific shapes such as houses are poorly generated. Because the uncontrollability and unpredictability of the proposed system sometimes confuse the human collaborators, the system is more suitable for early design explorations. Moreover, because our training dataset comprises frames from Japanese animation movies, the generated images share similar artistic styles. Preparing sufficient finely annotated data for training would extend our method to paintings in other styles. Unlike template-based tools for design, our system supports personalized and innovative creation. Its easy-to-operate interaction helps both novices and experts create unique cartoon stories. Furthermore, our system can quickly generate large quantities of design materials in a specific style, opening new avenues for automatic creative designs such as the design of advertisements and game scenes.

After analyzing the user studies, we inferred four guidelines for the development of an AI co-creative system. In general, humans wish to dominate the creative process, and aided by the machine, to achieve their creative goals quickly and effectively. Therefore, the co-creative system should first provide more communication routes between the machine and humans. When working with human partners, we express our ideas in various ways, using

sketches, language descriptions, and bodily expressions. Similarly, users cooperating with a machine prefer to communicate their ideas to the machine through multi-channel inputs. For example, users wish to control the brush stroke during the drawing process, informing the machine on where to apply a denser texture. If users could express their ideas more clearly, the machine could return results that more closely match their expectations. Second, the interactions should enable humans to understand the logic of the machine. As highlighted in the user studies, the machine often returns unexpected results. On one hand, these unexpected results can surprise the users and inspire their creativity. On the other hand, they cause confusion and hinder the users' creativity. If users understand the logic of the machine, they can better collaborate and negotiate with the machine. Third, we should increase the human ownership of the outputs. Although some users of our system enjoyed collaborating with the machine, the low ownership of paintings reduced their sense of achievement. Owing to the automatic generation capabilities of the machine, users can quickly obtain amazing image results, but lack opportunities to participate in the creative process. Therefore, the machine automation must be carefully balanced against user engagement. Fourth, we should let the machine adapt to the humans' own drawing or design habits. For example, many users do not fill the canvas with their drawings. Such adaptation would help our algorithm capture the user's intentions. An AI co-creative system should support flexible and natural interactions, allowing people to create higher-quality works in familiar ways.

6 Conclusions

We presented a co-creative drawing system, in which computers and humans collaborate to create cartoon landscape paintings. The paintings are generated quickly and require the minimum interaction and painting expertise. The computer masters the domain knowledge of cartoon landscape paintings by learning thousands of cartoon images and their semantic label maps and edge detection maps. The trained machine can easily understand the user inputs and produce high-quality paintings. Regardless of what is drawn by users, the computer builds on the input sketches and makes its own contribution to the final result. The generation quality of SmartPaint was evaluated in comparison experiments. The results demonstrated that our method produces more vivid cartoon paintings than the original pix2pixHD method does. The usability of SmartPaint was then evaluated in two user studies. The results showed that SmartPaint can assist creative drawing by both novice users and professional users, and reduced the number of cumbersome steps for designers, enabling them to fully realize their inspirations. We conclude with four guidelines for building a cooperative machine-human system: (1) increasing the number of communication channels between the machine and humans, (2) improving human understanding of the logic of the machine, (3) increasing the human ownership of the creative outputs, and (4) letting the machine adapt to the drawing and design habits of humans.

We believe that the learning and generating capabilities of deep learning technologies open new opportunities for human-machine collaborations in creative tasks. Combining the intelligences of machines and humans is a worthy undertaking.

Compliance with ethics guidelines

Lingyun SUN, Pei CHEN, Wei XIANG, Peng CHEN, Wei-yue GAO, and Ke-jun ZHANG declare that they have no conflict of interest.

References

Atarsaikhan G, Iwana BK, Narusawa A, et al., 2017. Neural font style transfer. Proc 14th IAPR Int Conf on Document Analysis and Recognition, p.51-56. https://doi.org/10.1109/ICDAR.2017.328

Belongie S, Malik J, Puzicha J, 2001. Shape context: a new descriptor for shape matching and object recognition. Proc 13th Int Conf on Neural Information Processing Systems, p.798-804.

Benedetti L, Winnemöller H, Corsini M, et al., 2014. Painting with Bob: assisted creativity for novices. Proc 27th Annual ACM Symp on User Interface Software and Technology, p.419-428. https://doi.org/10.1145/2642918.2647415

Bowman SR, Vilnis L, Vinyals O, et al., 2016. Generating sentences from a continuous space. Proc 20th SIGNLL Conf on Computational Natural Language Learning, p.10-21. https://doi.org/10.18653/v1/K16-1002

Canny J, 1987. A computational approach to edge detection. In: Fischler MA, Firschein O (Eds.), Readings in Computer Vision: Issues, Problem, Principles, and Paradigms. Elsevier, Amsterdam, p.184-203. https://doi.org/10.1016/B978-0-08-051581-6.50024-6

Champandard AJ, 2016. Semantic style transfer and turning

two-bit doodles into fine artworks. https://arxiv.org/abs/1603.01768

Chen C, Lin JC, Liao MH, et al., 2016. Learning to detect salient curves of cartoon images based on composition rules. Proc 11th Int Conf on Computer Science & Education, p.808-813. https://doi.org/10.1109/ICCSE.2016.7581686

Chen T, Cheng MM, Tan P, et al., 2009. Sketch2Photo: Internet image montage. *ACM Trans Graph*, 28(5), Article 124. https://doi.org/10.1145/1618452.1618470

Chen Y, Lai YK, Liu YJ, 2018. CartoonGAN: generative adversarial networks for photo cartoonization. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9465-9474. https://doi.org/10.1109/CVPR.2018.00986

Chu NSH, Tai CL, 2004. Real-time painting with an expressive virtual Chinese brush. *IEEE Comput Graph Appl*, 24(5):76-85. https://doi.org/10.1109/MCG.2004.37

Ci YZ, Ma XZ, Wang ZH, et al., 2018. User-guided deep anime line art colorization with conditional adversarial networks. Proc 26th ACM Int Conf on Multimedia, p.1536-1544. https://doi.org/10.1145/3240508.3240661

Cummmings D, Vides F, Hammond T, 2012. I don't believe my eyes! Geometric sketch recognition for a computer art tutorial. Proc Int Symp on Sketch-Based Interfaces and Modeling, p.97-106. https://doi.org/10.2312/SBM/SBM12/097-106

Davis NM, 2013. Human-computer co-creativity: blending human and computational creativity. Proc 9th Artificial Intelligence and Interactive Digital Entertainment Conf, p.9-12.

Davis NM, Hsiao CP, Singh KY, et al., 2015. Drawing apprentice: an enactive co-creative agent for artistic collaboration. Proc ACM SIGCHI Conf on Creativity and Cognition, p.185-186. https://doi.org/10.1145/2757226.2764555

Davis NM, Hsiao CP, Singh KY, et al., 2016a. Co-creative drawing agent with object recognition. Proc 12th Artificial Intelligence and Interactive Digital Entertainment Conf, p.9-15.

Davis NM, Hsiao CP, Yashraj Singh K, et al., 2016b. Empirically studying participatory sense-making in abstract drawing with a co-creative cognitive agent. Proc 21st Int Conf on Intelligent User Interfaces, p.196-207. https://doi.org/10.1145/2856767.2856795

Dekel T, Gan C, Krishnan D, et al., 2018. Sparse, smart contours to represent and edit images. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3511-3520. https://doi.org/10.1109/CVPR.2018.00370

Dixon D, Prasad M, Hammond T, 2010. iCanDraw: using sketch recognition and corrective feedback to assist a user in drawing human faces. Proc SIGCHI Conf on Human Factors in Computing Systems, p.897-906. https://doi.org/10.1145/1753326.1753459

Eitz M, Richter R, Hildebrand K, et al., 2011. Photosketcher: interactive sketch-based image synthesis. *IEEE Comput Graph Appl*, 31(6):56-66. https://doi.org/10.1109/MCG.2011.67

Gatys LA, Ecker AS, Bethge M, 2016. Image style transfer using convolutional neural networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.2414-2423. https://doi.org/10.1109/CVPR.2016.265

Goodfellow I, Pouget-Abadie J, Mirza M, et al., 2014. Generative adversarial nets. Proc 27th Int Conf on Neural Information Processing Systems, p.2672-2680.

Güçlütürk Y, Güçlü U, van Lier R, et al., 2016. Convolutional sketch inversion. European Conf on Computer Vision, p.810-824. https://doi.org/10.1007/978-3-319-46604-0_56

Ha D, Eck D, 2017. A neural representation of sketch drawings. https://arxiv.org/abs/1704.03477

Huang HZ, Wang H, Luo WH, et al., 2017. Real-time neural style transfer for videos. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.783-791. https://doi.org/10.1109/CVPR.2017.745

Isola P, Zhu JY, Zhou TH, et al., 2017. Image-to-image translation with conditional adversarial networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1125-1134. https://doi.org/10.1109/CVPR.2017.632

Karimi P, Davis N, Grace K, et al., 2018. Deep learning for identifying potential conceptual shifts for co-creative drawing. https://arxiv.org/abs/1801.00723

Lee YJ, Zitnick CL, Cohen MF, 2011. ShadowDraw: realtime user guidance for freehand drawing. *ACM Trans Graph*, 30(4), Article 27. https://doi.org/10.1145/2010324.1964922

Li MJ, Huang HZ, Ma L, et al., 2018. Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks. Proc 15th European Conf on Computer Vision, p.186-201. https://doi.org/10.1007/978-3-030-01240-3_12

Liu YF, Qin ZC, Wan T, et al., 2018. Auto-painter: cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks. *Neurocomputing*, 311:78-87. https://doi.org/10.1016/j.neucom.2018.05.045

Luan FJ, Paris S, Shechtman E, et al., 2017. Deep photo style transfer. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4990-4998. https://doi.org/10.1109/CVPR.2017.740

Mirza M, Osindero S, 2014. Conditional generative adversarial nets. https://arxiv.org/abs/1411.1784

Ning X, Laga H, Saito S, et al., 2011. Contour-driven Sumi-e rendering of real photos. *Comput Graph*, 35(1):122-134. https://doi.org/10.1016/j.cag.2010.11.017

Oh C, Song J, Choi J, et al., 2018. I lead, you help but only with enough details: understanding user experience of co-creation with artificial intelligence. Proc CHI Conf on Human Factors in Computing Systems, Article 649. https://doi.org/10.1145/3173574.3174223

Portenier T, Hu QY, Szabó A, et al., 2018. Faceshop: deep sketch-based face image editing. *ACM Trans Graph*, 37(4), Article 99. https://doi.org/10.1145/3197517.3201393

Roberts A, Engel J, Eck D, 2017. Hierarchical variational autoencoders for music. Workshop on Machine Learning for Creativity and Design, NIPS.

Sangkloy P, Lu JW, Fang C, et al., 2017. Scribbler: controlling deep image synthesis with sketch and color. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.5400-5409. https://doi.org/10.1109/cvpr.2017.723

Selim A, Elgharib M, Doyle L, 2016. Painting style transfer for head portraits using convolutional neural networks. *ACM Trans Graph*, 35(4), Article 129. https://doi.org/10.1145/2897824.2925968

Simo-Serra E, Iizuka S, Sasaki K, et al., 2016. Learning to simplify: fully convolutional networks for rough sketch cleanup. *ACM Trans Graph*, 35(4), Article 121. https://doi.org/10.1145/2897824.2925972

Wang TC, Liu MY, Zhu JY, et al., 2018. High-resolution image synthesis and semantic manipulation with conditional GANs. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.8798-8807. https://doi.org/10.1109/CVPR.2018.00917

Xian WQ, Sangkloy P, Agrawal V, et al., 2018. TextureGAN: controlling deep image synthesis with texture patches. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.8456-8465. https://doi.org/10.1109/cvpr.2018.00882

Zhang YK, Hu KK, Ren PR, et al., 2017. Layout style modeling for automating banner design. Proc Thematic Workshops of ACM Multimedia, p.451-459. https://doi.org/10.1145/3126686.3126718

Zhao NX, Cao Y, Lau RWH, 2018. What characterizes personalities of graphic designs? *ACM Trans Graph*, 37(4), Article 116. https://doi.org/10.1145/3197517.3201355

Zhu JY, Krähenbühl P, Shechtman E, et al., 2016. Generative visual manipulation on the natural image manifold. Proc 14th European Conf on Computer Vision, p.597-613. https://doi.org/10.1007/978-3-319-46454-1_36

Zhu JY, Park T, Isola P, et al., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proc IEEE Int Conf on Computer Vision, p.2223-2232. https://doi.org/10.1109/ICCV.2017.244

Appendix: Japanese films and animations used in the training data

1. Castle in the Sky
2. The Secret World of Arrietty
3. Howl's Moving Castle
4. Princess Mononoke
5. My Neighbor Totoro
6. Porco Rosso
7. The Wind Rises
8. From up on Poppy Hill
9. Tales from Earthsea
10. Only Yesterday
11. Wolf Children
12. Summer Wars
13. The Place Promised in Our Early Days
14. Children Who Chase Lost Voices
15. Hanasaku Iroha
16. Ao Haru Ride
17. Your Lie in April
18. Anohana: the Flower We Saw That Day
19. Natsume's Book of Friends
20. The Garden of Words
21. Silver Spoon
22. Sakura Quest
23. Hotarubi no Mori e
24. Non Non Biyori
25. Barakamon
26. Violet Evergarde