

# Automatic Comic Generation with Stylistic Multi-page Layouts and Emotion-driven Text Balloon Generation

XIN YANG, ZONGLIANG MA, and LETIAN YU, Dalian University of Technology, China

YING CAO\*, City University of Hong Kong, China

BAOCAI YIN, XIAOPENG WEI, and QIANG ZHANG, Dalian University of Technology, China

RYNISON W.H. LAU, City University of Hong Kong, China

In this paper, we propose a fully automatic system for generating comic books from videos without any human intervention. Given an input video along with its subtitles, our approach first extracts informative keyframes by analyzing the subtitles, and stylizes keyframes into comic-style images. Then, we propose a novel automatic multi-page layout framework, which can allocate the images across multiple pages and synthesize visually interesting layouts based on the rich semantics of the images (e.g., importance and inter-image relation). Finally, as opposed to using the same type of balloon as in previous works, we propose an emotion-aware balloon generation method to create different types of word balloons by analyzing the emotion of subtitles and audios. Our method is able to vary balloon shapes and word sizes in balloons in response to different emotions, leading to more enriched reading experience. Once the balloons are generated, they are placed adjacent to their corresponding speakers via speaker detection. Our results show that our method, without requiring any user inputs, can generate high-quality comic pages with visually rich layouts and balloons. Our user studies also demonstrate that users prefer our generated results over those by state-of-the-art comic generation systems.

CCS Concepts: • **Computing methodologies** → **Computer vision tasks**; *Computer vision*.

Additional Key Words and Phrases: automatic, comic books, keyframes, stylizing, multi-page, layout

## ACM Reference Format:

Xin Yang, Zongliang Ma, Letian Yu, Ying Cao, Baocai Yin, Xiaopeng Wei, Qiang Zhang, and Rynson W.H. Lau. 2020. Automatic Comic Generation with Stylistic Multi-page Layouts and Emotion-driven Text Balloon Generation. 1, 1 (June 2020), 19 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

\*The corresponding author.

This work was supported in part by the National Natural Science Foundation of China under Grant 91748104, Grant 61972067, Grant 61632006, Grant U1811463, Grant U1908214, Grant 61751203, in part by the National Key Research and Development Program of China under Grant 2018AAA0102003, Grant 2018YFC0910506, in part by the Innovation Technology Funding of Dalian (Project No. 2020JJ26GX036).

Authors' addresses: Xin Yang, [xinyang@dlut.edu.cn](mailto:xinyang@dlut.edu.cn); Zongliang Ma, [liangzongma1997@mail.dlut.edu.cn](mailto:liangzongma1997@mail.dlut.edu.cn); Letian Yu, [yuley3012@mail.dlut.edu.cn](mailto:yuley3012@mail.dlut.edu.cn), Dalian University of Technology, Department of Computer Science, 2 Linggong Road, Dalian, Liaoning, 116024, China; Ying Cao, [caoying59@gmail.com](mailto:caoying59@gmail.com), City University of Hong Kong, Hong Kong, China; Baocai Yin, [ybc@dlut.edu.cn](mailto:ybc@dlut.edu.cn); Xiaopeng Wei, [weixp@dlut.edu.cn](mailto:weixp@dlut.edu.cn); Qiang Zhang, [zhangq@dlut.edu.cn](mailto:zhangq@dlut.edu.cn), Dalian University of Technology, Department of Computer Science, 2 Linggong Road, Dalian, Liaoning, 116024, China; Rynson W.H. Lau, [rynson.lau@cityu.edu.hk](mailto:rynson.lau@cityu.edu.hk), City University of Hong Kong, Hong Kong, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2020 Association for Computing Machinery.

XXXX-XXXX/2020/6-ART \$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

## 1 INTRODUCTION

Manga is a popular artwork and information dissemination media, mainly due to its convenient reading form as well as excellent use of storytelling techniques (in drawing, paneling, layout, etc.) for enriched and immersive reading experience. However, traditional manga creation is time-consuming, and it requires professional skills and various content materials. Especially, it is very hard for non-professionals to produce their own manga books. Nowadays, movies or videos are common and popular everywhere with the development of Internet technology, especially the mobile videos uploaded to YouTube per day will take more than 82 years to watch for a man [27]. Such massive amount of videos can be a useful resource for the manga content creation and may save content creation time for designers. Thus, how to convert videos or movies to manga books will be an interesting and significant job. There have been some existing work [2, 11, 20] proposed to edit videos in a manga-like way. Wang *et al.* [24] proposed an automatic schema for turning a movie clip to comics. Chu *et al.* [4] proposed a system to transform image sequence into a comics-based presentation in an optimized way. However, how to automatically convert videos to high-quality manga books with visual richness and excellent storytelling abilities is still a challenging and unsolved problem. The key difficulties are: 1) A fully end-to-end manga generation system to convert videos to manga books is desirable in a fast-forward prototype design, especially for non-professionals. However, existing works either require significant user inputs to give good results or are fully automatic but generate over-simplified results that lack visual richness and expressiveness. 2) The use of diverse word balloons in comics, whose shapes and font sizes vary according to contents, can enhance the expressiveness of a manga book and reinforce the readers' perception of feeling and emotional states of characters [13]. Existing works generally select some pre-defined shapes for balloon generation. This would result in generated outputs failing to express story as faithfully as real comic pages. Thus, how to allocate abundant types of word balloon shape according to the emotion of different character dialogues and the corresponding audio is important. In order to generate multi-page manga books, panels should be assigned to different pages. In [4], visual coherence and reading pace are main factors of their page allocation. Compared with their method, semantic relation of keyframes is considered in our system, which can help to allocate the keyframes with semantic relation on the same page.

Our framework solves the problem of allocating the selected keyframes across pages via a genetic algorithm and then organize the keyframes on each page manga-style layouts. To synthesize professional-looking manga-style layouts, we adopt a data-driven layout approach [2] that learns layout styles from manga data. However, instead of relying upon users to specify inputs, we extend their approach by automatically extracting their inputs from the keyframes, including region of interest, importance rank, inter-frame relation. This makes our layout framework fully automatic without the need for user inputs of any form. Finally, we propose a data-driven, emotion-aware balloon generation model, which can generate different balloon shapes and dynamically adjust font sizes based on the emotion of subtitles and audio. The generated balloons are then placed at the right places by detecting who is speaking and the location of the speaker's mouth. Furthermore, in order to allow users to exert some degrees of control on generated results, we build a user-friendly interface to impose users' constraints towards building more personalized designs and fine tune the results. Our experiments show that the user interface can save users' time, while providing more freedom of creation.

In summary, our contributions are:

- We propose a fully automatic system to generate manga books from videos of arbitrary types (TV series, movies, cartoons). Our system does not need any manual inputs from users, and can generate high-quality manga pages with rich visual effects and expressive storytelling.

- We propose a multi-page layout framework for generating stylistically rich panel layouts across multiple pages jointly, based on rich semantics extracted from video frames automatically.
- We propose an emotion-aware model for balloon generation. Our generated balloons can adapt to the emotion of contents measured by sentiment in subtitles and audios, which could enhance the expressiveness of final results.

## 2 RELATED WORK

### 2.1 Manga Generation

There are already some related works on generating manga or comics. Ryu *et al.* [20] proposed a semi-automated system to generate black/white comic books from an input movie. Keyframes are extracted from a given movie manually. Then a "comic cut converter" based on Mean Shift segmentation and Bilateral Filter is used for stylization. After that, they perform background effect stylization by separating foreground from background and add some effect to the background. Finally, stylized font, speed line, word balloon, and stylized icon are placed to the image by users. However, a problem is that user intervention runs through its entire system, which requires a lot of manual efforts. Wang *et al.* [24] proposed a complete approach to generate comics. Although they claimed that their system is automatic, it still needed a pre-prepared script file with speech contents and speaker identities. Further, they used regular grid-based layouts and a fixed balloon shape, which impairs the visual interest of generated results. Jing *et al.* [11] described a content-aware approach for manga generation and generated layouts by maximizing information contained in a page. Unfortunately, their approach works only for conversational videos (videos with conversation of characters, such as TV series) which make it can't be applied to the videos without conversation, such as vlogs. And the shape of word balloon in their work is fixed which makes it boring for readers to read the manga book. Chu *et al.* [4] designed a system to convert an image sequence to a comics-based presentation by explicitly solving multi-page panel allocation through a labeling optimization. Unfortunately, they only generated regular panel layouts with limited styles by simply matching content importance with pre-defined templates and also used a fixed balloon shape. In summary, despite progress made in generating comics from videos, all prior works either require additional user inputs, or use simplified representation or methods for layout and balloon generation, which cause results to lack visual variety and expressiveness. In contrast, our method is fully automatic and can generate visually rich, multi-page layouts and various balloon shapes that are adaptive to the emotion of speakers.

### 2.2 Keyframe Selection

Keyframe selection is important to generate an interesting manga book. According to our research, there are many existing methods [4, 11, 18, 24] for keyframe selection. Wang *et al.* [24] employed different strategies to extract descriptive keyframes after subshot detection and classification. [11] extracted speaker-key-frames based on speaker detection, and then extracted keyframes from changing scene. Chu *et al.* [4] first extracted keyframes from subshots and then employed a keypoint-based approach to find near-duplicate frames and eliminate redundant keyframes. Qu *et al.* [18] proposed a keyframe extraction method based on the similarity comparison between the undetected frame and the last shot on HSV color space. Different from their work, our keyframe selection method contains two stages: we first divide the video into two kinds of shots: dialogue-shots (shots with corresponding subtitles) and non-dialogue-shots (shots without corresponding subtitles) based

on the information in subtitles. Then, we perform two different strategies for selecting dialogue-keyframes from dialogue-shots and non-dialogue-keyframes from non-dialogue-shots to ensure the fluency of the story and enrich the content of the manga book.

### 2.3 Stylization

In order to generate manga-style images, stylization is performed in our system. Previous works of stylization mainly take two kinds of stylization: black-white stylization and colored stylization. Ryu *et al.* [20] got black-white style images by mean Shift segmentation and Bilateral Filter are combined to get black-white style images. Wang *et al.* [24] applied an image abstraction method to get colored style images by modifying the contrast of visually important features. Both of the two kinds of stylization are used in the work of [11]. Winnemöller *et al.* [26] proposed an advanced approach for stylization with extended Difference-of-Gaussians based on the work of [25]. In recent years, there are some deep learning methods for stylization, (e.g., [7, 8, 12, 16]), however, the stylization module is not the core of our system. Since the approach of [26] is simple and requires no post-processing, we employ it for stylization in our complex system, which we find works well.

### 2.4 Panel Layout

Panel layout is used to present all the panels in a manga style. Early researches, such as [20], only took grid layout which assigned images to rectangular grids. [24] and [23] designed several layout templates and chose one of the templates as the layout according to the matching degree between input panels and the pre-defined templates. For example, [24] represented their template and generated panel list as two numerical sequences and computed the distance of the two numerical sequences to select the best template. Jing *et al.* [11] determined the local structure based on video content and generated the initial layout of a comic page automatically, then they performed layout optimization to get the final panel layout. Cao *et al.* [2] proposed a data-driven method for panel layout. Firstly, they created an initial layout that best fit the input artworks and layout structure model, according to a generative probabilistic framework. Then, the layout and artwork geometries were jointly refined using an efficient optimization procedure, resulting in a professional-looking manga layout.

In our system, we choose the method proposed by [2] for panel layout since it provides various styles learned from different manga series. However, their method is semi-automatic, and needs users to provide ROI (region of interest), importance of each panel, relation of panels and the number of panels presented on each page. Thus, we extend their method by automating the estimation of the required inputs from video frames.

### 2.5 Text Balloon Generation and Placement

Balloon shape is important for manga creation. There are some tools for text balloon generation, such as "Balloonist" (Horlick, 2016), "Comic Book Creator" (Planetwide Games, 2005), and "SuperLame!" (SuperLame, 2015). However, all of these tools require some manual efforts. Preu *et al.* [17] applied textual analysis on a screenplay and extracts information to create two types of balloons: speech balloon and noise balloon (balloon without dialogue and just express the emotion of background). Balloons placement is controlled by a layout algorithm to keep reading order and avoiding occluding speaker's face or exceeding panel boundary. The accuracy of their approach is acceptable, but requires an input screenplay which is not easy to obtain. Hong *et al.* [10] proposed a script-face mapping method to detect who is speaking, and then place the pre-generated word balloon near the speaker and make the balloon tip point to the speaker. However, screenplays of the movies are not easy to get and the tip of balloon may not point to the speaker correctly. Wang *et al.* [24] proposed an improved approach. They used speech recognition to replace script-face



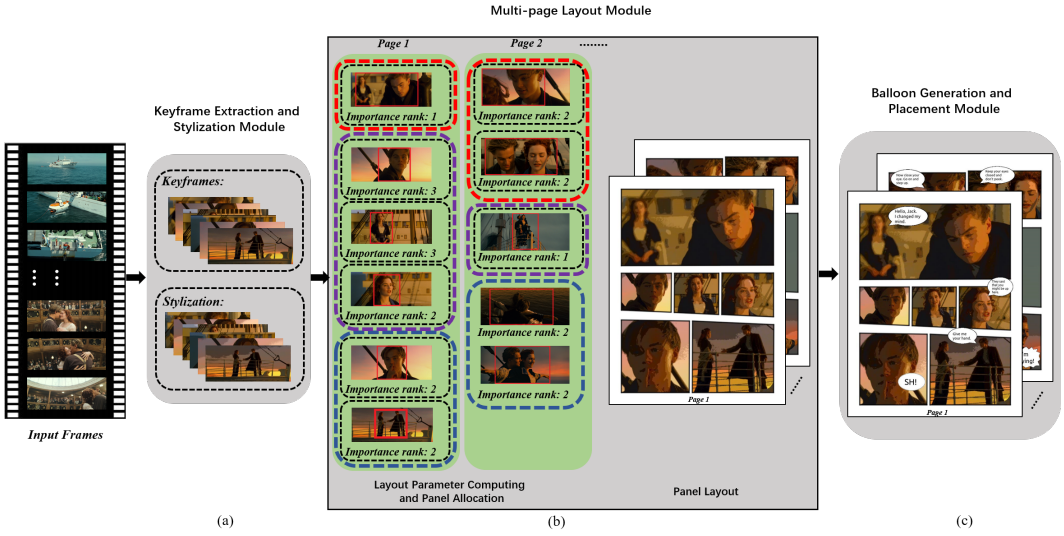


Fig. 1. Overall pipeline of our system. (a): Keyframe Extraction and Stylization. (b): Automatic Multi-Page Layout Framework, red, purple, and green dotted boxes mean different groups. (c): Balloon Generation and Placement. In step (a), we perform keyframe selection and stylization to get the stylized keyframes of the input video frames. In step (b), we first obtain four layout parameters of the frames including region of interest, importance rank, semantic relation, and allocate the frames across different pages. Then, we perform the layout algorithm in [2] for multi-page layout. In step (c), we designed an emotion-aware model for balloon generation and placement.

mapping for speaker recognition to eliminate the need for the screenplay. However, their simplified balloon generation only uses one type of balloon which reduces the variety and expressiveness of generated manga books. Sawada *et al.* [21] chose oval balloons and used feature tracking for speaker detection and word balloon placement. Chu *et al.* [4] proposed an optimized method for balloon generation. Balloon size and balloon shape are two factors they took into consideration. However, they selected the shape of balloons only by comparing several words in a subtitle with pre-prepared words.

In this work, we consider multiple types of balloon shapes and change both balloon shape and font size in balloons by analyzing the emotion of subtitles and audios. This allows us to produce the results that better express the pacing of stories and the feeling of characters. Then, we perform the speaker detection in [21] for word balloon placement.

### 3 METHODOLOGY

Our key idea is to design our system in a fully automatic manner without any manually specified parameters or constraints. Meanwhile, we optionally introduce user interaction for more personalized design and diversities. As shown in Fig 1, the input of our system is a video with its subtitles file, and there are mainly three modules in our system, detailed as keyframe selection and stylization, multi-page layout generation, and balloon generation and placement. Firstly, representative keyframes are extracted and stylized from the input video in the keyframe processing module. Secondly, we perform an automatic multi-page layout framework to present all the panels in a manga-like way. Finally, we generate word balloons by analyzing the emotion of subtitles

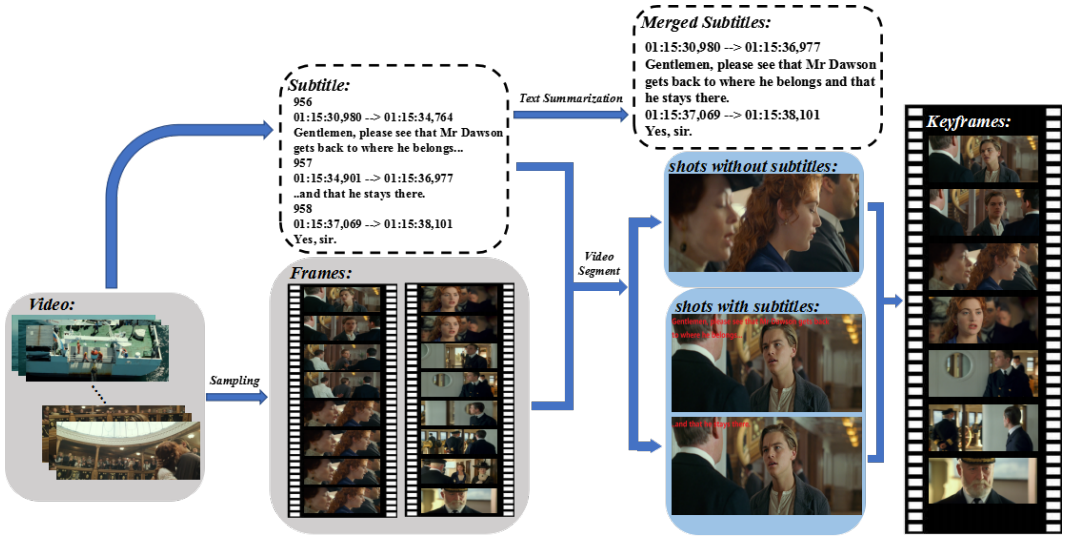


Fig. 2. Pipeline of Keyframe Selection. Firstly, we select one frame every 0.5 second to get frames which can represent the input video. Meanwhile, the subtitles of the video is obtained from the video. Then, with the time duration of the subtitles, we divided the frames into two groups: frames with subtitles and frames without subtitles. For these two groups of frames, we take different strategies to pick up the representative keyframes.

and audios, summarize multiple sentences, then perform lip motion detection to help for balloon placement. We will give more design and implementation details in the following sections.

### 3.1 Keyframe Extraction and Stylization Module

The input of our system is a video and its subtitles which contain dialogues and corresponding start and end time information. We first select one frame every 0.5 seconds from the raw video. These selected frames can represent the raw movie. Then we use time information in subtitles and similarity between two consecutive frames to select informative keyframes. Finally, we perform stylization to convert ordinary images to manga-style images.

**Keyframe Selection.** We make use of the time information for keyframe selection as shown in Fig 2. Firstly, we segment the video into shots using the start and end time of each subtitle. There are two kinds of shots: dialogue-shots (shots with subtitles) and non-dialogue-shots (shots without subtitles). For dialogue-shots, we compute the GIST [15] similarity between two consecutive frames obtained before because the GIST similarity is macroscopic and it is accurate for our work. If the foregoing GIST similarity is smaller, two frames differ more from each other. In our implementation, if the similarity is less than threshold  $\theta_1$  we set, then the latter frame will be selected as a keyframe. If none of the frames corresponding to a subtitle are selected, we just pick the middle frame of the shot as a keyframe. Considering that more than one subtitle may correspond to a consecutive dialogue and the same scene, we compute the GIST similarity between consecutive keyframes obtained before. If the similarity is more than the threshold  $\theta_2$  we set, we regard them as belonging to the same scene. Then, we will just keep one of them as keyframes and merge the subtitles. It is possible that we can select more than one keyframes during one subtitle, and we consider those keyframes to have semantic relations which will be used in multi-page layout. For non-dialogue-shots, we first select the frame most dissimilar to the frames in the shot. Then, in order to reduce

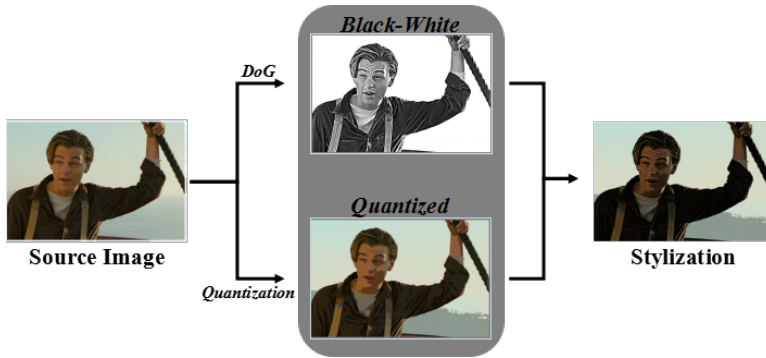


Fig. 3. Pipeline of Stylization. Firstly, we convert the source image to black-white image via the extended DoG [26] and then get quantized image by color quantization. Finally, we combine these two kinds of image to get colored stylization.

redundancy, we compute the GIST similarity between the frame in this shot and the keyframes we selected before. Only if the similarity is less than we set before, this frame can be selected as a keyframe. Finally, the set of subtitles is then grouped by comparing the starting timestamp and keyframe's timestamp. Any subtitle starts within the time interval bounded by the start and end timestamp of a keyframe will be gathered together. In this way, we can extract informative keyframes and get the semantic relevance of some keyframes which is important in multi-page layout.

**Stylization.** As shown in Fig 3, we employ the extended Difference-of-Gaussians approach [26] to convert the source image to black-white image. For colored stylization, firstly, 128 level color quantization is executed to get the quantized image. Then, we get the DoG edges of the image using the algorithm in [26]. Finally, we get colored stylization by combining DoG edges and quantized image.

### 3.2 Multi-page Layout Module

Multi-page layout framework is used to automatically allocate and organize the panels across pages in visually rich layouts. In our framework, we first compute four key factors that are used to guide our multi-page layout generation, including region of interest of keyframes (ROI), importance rank of keyframes, semantic relation between keyframes, and the number of panels on one page. Then, We propose an optimization-based panel allocation method to assign the keyframes into a sequence of pages and use a data-driven manga-style layout synthesis approach to generate the layout for each of the pages.

#### Layout Parameter Computing.

- **Region of Interest (ROI):** We use a class activation mapping algorithm (CAM) for classification [28] to find region of interest on a keyframe image because CAM can output a heatmap of an input image and the heatmap is usually the most attractive region of the input image. Specifically, an image is first input to a five cascaded layers convolutional neural network (CNN). The output features of the last CNN are the input to the global average pooling (GAP) layer. Then, a full connection layer is used to get the heatmap of the input image. As shown in Fig 4, in our implementation, we select the top seven scored heatmaps sorted by the classification score. We get the average of their activation values at the same position of these heatmaps, and convert them into a grayscale image. Finally, we compare the value of

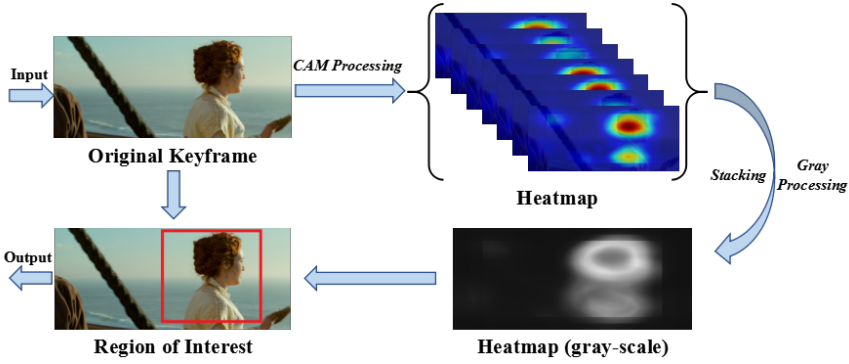


Fig. 4. Processing of extracting the region of interest (ROI). We obtain different heatmaps by CAM processing from original keyframes. Then, we get gray-scale image of heatmap by stacking these heatmap and gray processing. Finally, we get ROI from gray-scale image of heatmap.

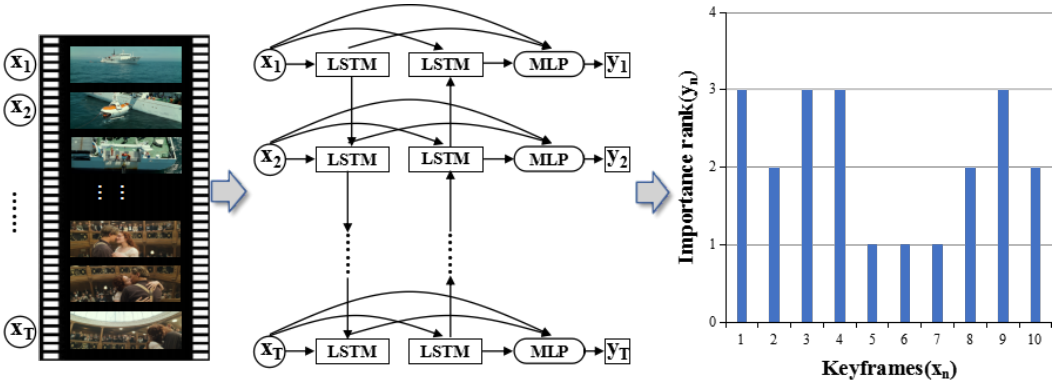


Fig. 5. LSTM Model. The input  $\{x_1, x_2, \dots, x_n\}$  is the keyframes and the output  $\{y_1, y_2, \dots, y_n\}$  is the corresponding importance ranking of each keyframe.

the grayscale image with the threshold  $\theta_3$  we set. If the value of a point is greater than  $\theta_3$ , it is regarded to be within the ROI. And if the value of a point is less than  $\theta_3$ , it is regarded to be outside of the ROI. Then, a minimum bounding box is found by retaining all the within-points and reducing outside-points. The minimum bounding box is the ROI we find.

- **Importance Rank:** In order to get the importance of each keyframe, we take use of the LSTM neural network used for video summarization [27]. As reported in [27], the LSTM model can output frame-level importance scores representing the likelihoods of the frames being selected as a part of summary. The procedure can be summarized as follows: 1) As shown in Fig 5, we input a 1024-dimensions vector obtained by extracting the output of the penultimate layer (pool 5) of the GoogleNet model [22] to the LSTM model because it can modestly improve the performance over the same shallow features (i.e. color histograms, GIST, HOG, dense SIFT) for the task of video summarization. We put the input frames into the official pretrained GoogleNet model and fetch the output feature vectors of the fifth pooling layer; 2) We put the feature vectors into the LSTM pretrained in [27] and the outputs are the importances of the input frames we need.

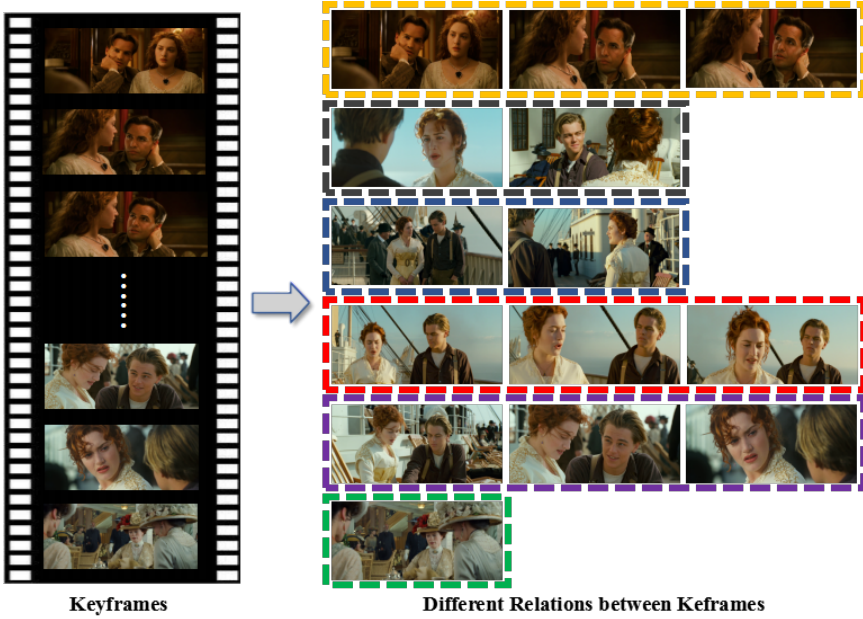


Fig. 6. Different relation between keyframes. Frames in the same box have semantic relation while frames in different boxes do not have semantic relation.

- **Keyframe Relation:** We compute the keyframe relation in keyframe selection stage. As mentioned before, we can select more than one frame in our keyframe selection and it is obvious that the selected keyframes with the same subtitle are semantic related. Note that, the result of the relation among keyframes is sparse because only the frames with the same subtitle can be regarded as semantic related. The result is shown in Fig 6.

**Panel Allocation.** In a manga book, the number of panels on each page is not fixed, different pages have varied number of panels in order to make readers have a better reading experience. In our implementation, we take it as an optimization problem in a global way, and assign all the panels to manga pages together.

Let  $\{f_1, f_2, \dots, f_N\}$  denote the input keyframes where  $N$  is the number of the input keyframes obtained from keyframe selection. Given the total number of pages  $N$  in a manga book, we should generate the number of panels on each page subject to:

$$\begin{cases} n_i \leq N_{\max} \\ n_i \geq N_{\min} \\ \sum_{i=1}^M n_i = N \end{cases}$$

Here,  $n_i$  denotes the number of panels on the  $i$ th page.  $N_{\max}$  is the maximum number of panels on a page and  $N_{\min}$  is the minimum number of panels on a page.  $M$  is the total number of pages, which is user-defined. There are two more constraints we should take into consideration. One is uniformity and the other is the relation among panels. Uniformity means that the panels with high importance rank should be assigned to different pages to avoid having boring pages, i.e., pages that only contain panels of low importance values. The relation among panels means that the panels with semantic relation should be assigned to the same page, and the readers can read it fluently.

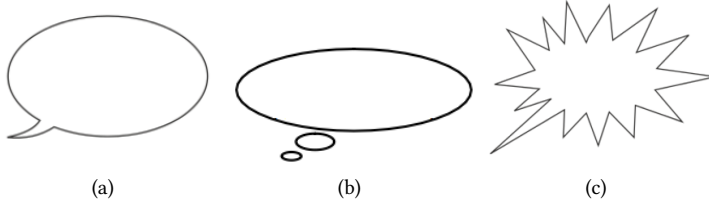


Fig. 7. Popular types of shapes of balloon.(a) rounded balloon. (b) thought balloon. (c) jagged contour balloon.

Considering of all the constraints above, we use genetic algorithm [9] to solve panels allocation next. Let  $\{n_1, n_2, \dots, n_M\}$  denotes the number of panels on each page. In our implementation, we set the maximum number  $N_{max}$  to 9.

The most important part of genetic algorithm is the valuation function. There are 5 parts in our valuation function. The first one is to calculate the absolute difference between each  $n_i$  and  $N_{max}$ , ensuring that  $n_i$  is no more than  $N_{max}$ . The second part is to calculate the absolute difference between each  $n_i$  and  $N_{min}$ , so that  $n_i$  is no less than  $N_{min}$ . The third part is to sum all  $n_i$ , and then calculate the absolute difference between the sum and  $N$ , which can ensure that the number of panels in the manga book is equal to the number of the input frames. The fourth part is to obtain the uniformity by calculating the standard deviation of the maximum importance  $I_i$  on each page. The less the standard deviation is, the better the result is. The fifth part is to count the number of the panels with semantic relation but assigned to different pages. Taking the five parts as a whole, we calculated the weighted value of the five parts, which is shown in Equation (1) below:

$$\min z = \alpha_1 \sum_{i=1}^M |n_i - N_{max}| + \alpha_2 \sum_{i=1}^M |n_i - N_{min}| + \alpha_3 \left| \sum_{i=1}^M n_i - N \right| + \alpha_4 SD + \alpha_5 R \quad (1)$$

$$SD = std(I_1, I_2, \dots, I_M) \quad (2)$$

where  $SD$  means the standard deviation and  $R$  means semantic relation. Obviously, the first three parts have the most strict constraints, thus we assigned a greater coefficient to them. The last two parts have the most relaxed constraints, and we assigned a relatively smaller coefficient to them. Finally, the weighted value of the whole five parts is the valuation of the generation.

**Panel Layout.** Four main layout parameters have been computed by the aforementioned methods, then we input them into an existing panel layout method [2], which provides various layout styles learned from different manga series. It is noteworthy that their original method [2] needs to manually pre-define all the parameters, that is unpractical to an automatic system. Fig 1(b) shows the calculation of layout parameters and the corresponding output layout pages.

### 3.3 Text Balloon Generation and Placement Module

**Balloon Shape Selection.** It is very important to manga content expression that abundant of vivid balloon shape can be selected according to dialogues and emotion in various circumstances. However, existing system works generally only selects basic elliptical speech balloon shape for dialogues, which is sometimes insufficient to fully express a dialogue or an emotion. It is mentioned in [6] that balloon shapes can help readers to capture information which does not have visual form



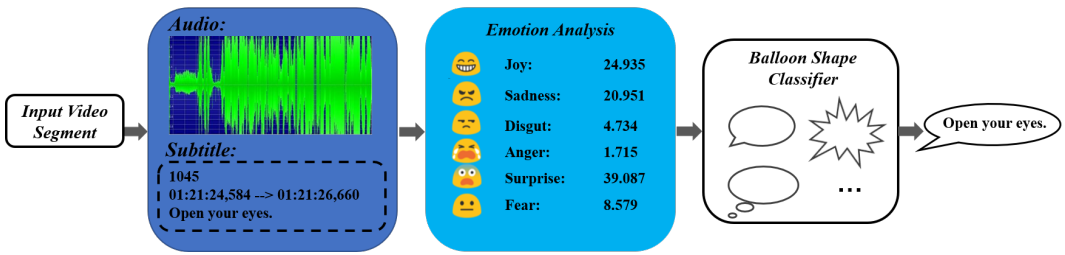


Fig. 8. Emotion-aware Balloon Generation. Given an input video segment, we first get its subtitles and audio. Then, emotion analysis is performed to get the emotion of subtitles and audio. Finally, we feed the emotion values into a balloon shape classifier to predict the balloon shapes of the subtitles.

such as sound. According to the statistic of [6], there are mainly 8 types of shapes of balloon. As shown in Fig 7, the three most commonly used shapes are rounded balloon or rectangular straight balloon, thought balloon, and jagged contour balloon, sorted in descending order. Rounded balloon or rectangular straight balloon is usually used for common speech which makes it most likely to be used in manga creation. Thought balloon is always used to state the thoughts of characters in the panel without speaking it out. A jagged contour balloon is always used when there are verbal conflicts, such as screaming, etc. In our system, we consider the three common balloon shapes.

Instead, we propose an emotion-aware balloon generation method, which can take advantage of videos' audio and subtitles containing emotional information to generate balloons with multiple kinds of shapes. As shown in Fig 8, given a video segment, we determine the shape and size of the balloon for its subtitles. In particular, firstly, we obtain the emotion of subtitles and the corresponding audio by emotion analysis. Then, the emotion of the subtitle and the number of words will determine the size of the word in the balloon. Finally, the shape of word balloon will be selected through a pre-trained classifier which takes as input the emotion of the audio and the subtitle for the panel and predicts the probabilities of different balloon types being selected. For example, if the emotion of the subtitle is plain, we tend to choose the common rounded balloon. If the emotion of the subtitle is strong, the jagged contour balloon should be chosen.

To train our balloon shape classifier, we collect training data that contains three kinds of elements: audio emotion, subtitle emotion, and type of balloon. First of all, we collect some animes and corresponding manga books. For each balloon in a book, we record its type of shape (rounded, thought, or jagged contour) and the text inside it. Then, using the text in the balloon, we can determine the time duration it belongs to in the anime. Meanwhile, we get the audio within this duration and put it into Mixed-Emotion [1] (an open-source emotion analysis tool). The outputs are valence and arousal of this period of sound. Valence is pleasantness (ranging from unhappy to happy) of a stimulus and arousal is the intensity of emotion (ranging from excited to calm). For the subtitles, we put it into an emotion recognizer and the output can represent the emotion of the subtitle. Here, we call the output emotion rank. Finally, we obtain the training data, which contains valence and arousal of the audio, emotion rank of the subtitle, and the shape of the balloon.

We select SVM (Support Vector Machine) as our classifier. We put the valence, arousal, and emotion rank into the SVM. After several iterations, the SVM can learn the mapping relations between the input and the type of balloon shape. When in use, we put the valence and arousal of the audio and the emotion rank of the subtitle into the trained SVM classifier. Then, the output is the type of balloon shape we should use.

**Text Summarization.** It is possible that one keyframe corresponds to more than one sentence in our keyframe selection. If the sentence is too long for the word balloon, it will lead to an

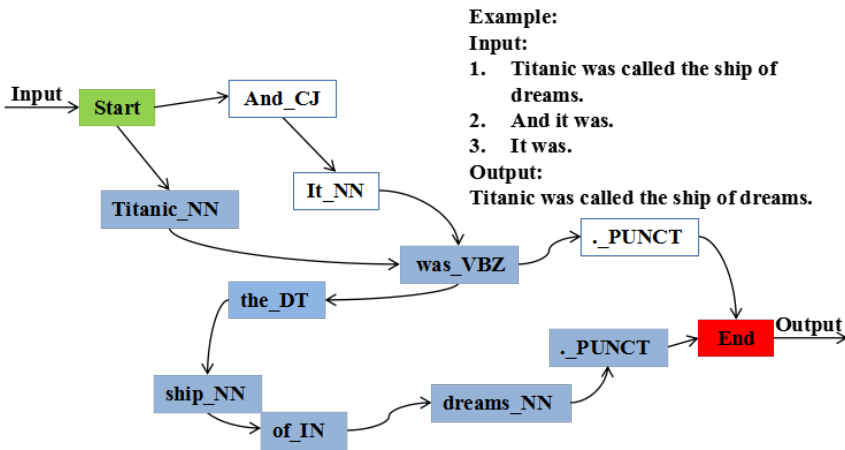


Fig. 9. Example of the word graph with a possible compression path. For instance, the node ‘Titanic\_NN’ in the word graph denotes that the word in the sentence is ‘Titanic’ while ‘NN’ indicates the part-of-speech of the word ‘Titanic’ is a noun. The input is three sentences and finally output the combined result: Titanic was called the ship of dreams. The result is obtained by the path formed by the blue node.

uncomfortable reading experience with repeated sentences and small font sizes. Hence, we use text summarization to merge multiple sentences into one compact sentence, and then render it on the generated balloon. To the best of our knowledge, we are the first to utilize text summarization in a comic generation system. As proposed by [5], the basic idea of multi-sentence compression is to add a set of sentences (excluding punctuations) iteratively to the word graph. Each sentence has a pair of start and end nodes indicating the start and end of the sentence. The first sentence will be simply added to the graph. For the other sentences, a word is mapped into an existing node if they share the same part-of-speech and no word from the sentence has been previously mapped to the node. A new node can be created if there is no possible mapping. Fig 9 displays the word graph obtained from the set of sentences:

- Titanic was called the ship of dreams
- And it was.
- It was.

The node is in the form of a word and part-of-speech pair, separated by an underscore. For simplicity, the edge weight is omitted.

To construct a word graph, words are mapped or created under three conditions:

- Non-stop words without similar candidates in the graph or with an unambiguous mapping.
- Non-stop words with either several possible candidates in the graph or more than one occurrence in the same sentence.
- Stopwords.

Condition 2 and 3 will lead to ambiguous mapping. In this case, the preceding and following words in the sentence and the neighboring nodes in the graph will be the factors affecting the final mapping. Node with a larger overlap in neighboring words, or the one with more words mapped onto it, will be selected. After mapping/creation of nodes, the words in the sentence are connected with directed edges. New nodes or nodes which were not connected before will have an edge weight of one. Edge weights between previously connected nodes will increment by one. The mapped nodes also store a list of id of sentences that contain the word together with the

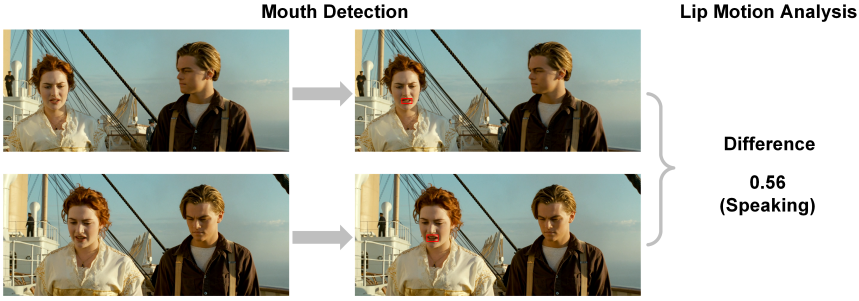


Fig. 10. Pipeline of Speaker Detection.

corresponding position index in the sentences. Once all sentences are added to the graph, the word graph is completed and can be used to find the best compression of input sentences, in other words, the best-compressed sentence. It is done by finding the path among  $K$  shortest paths with lightest average edge weight. Before this, there are some restrictions on how the path weight is allocated. Individual edge weight  $w_{ij}$  is given by Equation (4).

$$association(i, j) = \frac{freq(i) + freq(j)}{\sum_{s \in S} diff(s, i, j) - 1} \quad (3)$$

$$w_{ij} = \frac{association(i, j)}{freq(i) \times freq(j)} \quad (4)$$

where  $freq(i)$  and  $freq(j)$  are the numbers of words mapped to node  $i$  and  $j$ , respectively. Function  $diff(s, i, j)$  refers to the distance between the positions of words  $i$  and  $j$  in sentence  $s$ , defined as Equation (5). Finally,  $K$  shortest path algorithm is implemented to find 50 shortest paths from start node to end node using Equation (4). Paths are rejected if they cannot satisfy the minimum number of words (suggested to be 8) or have no verb node. The remaining paths are ranked by a score that is calculated by normalizing the total path weight over its length. The path with the lightest average edge weight will be the best compression.

$$diff(s, i, j) = \begin{cases} pos(s, i) - pos(s, j), & pos(s, i) < pos(s, j) \\ 0, & otherwise \end{cases} \quad (5)$$

**Balloon Placement.** Most existing work employed speaker detection and lip motion detection to get the location of the speaker in a frame, and then placed the balloon near the speaker it belongs to. Our approach is similar to theirs. As shown in Fig 10, firstly, we detect the mouth of every character in a frame using a face-detector Python library called 'Dlib', which is based on Histogram of Oriented Gradients (HOG) to extract the feature points on a face. 68 feature points will be extracted from one face, of which 48th to 59th depict the outline of the mouth. Then, a lip motion analysis [13] is employed to compute the mean squared difference of the pixel values within the mouth region between two continuous frames. The difference is calculated over a search region around the mouth region in the current frame. Finally, a threshold is set to determine if a character is speaking or not. After we get the location of the speaker, we place the word balloon near the speaker and point the tail of the balloon to the speaker's mouth.

## 4 EXPERIMENTS

In this section, we will introduce the experimental results of our system. Firstly, we compared our results with a state-of-the-art comic generation system to present the superiority and aesthetic

Table 1. Information of Input Movies.

| Movie Name | Clip | Duration | Cost Time | Movie Name    | Clip | Duration | Cost Time |
|------------|------|----------|-----------|---------------|------|----------|-----------|
| Titanic    | 1    | 3.25min  | 5.13min   | The Message   | 1    | 2.15min  | 3.41min   |
|            | 2    | 2.58min  | 4.15min   |               | 2    | 2.56min  | 3.97min   |
|            | 3    | 4.26min  | 6.95min   |               | 3    | 5.15min  | 9.17min   |
|            | 4    | 2.15min  | 5.56min   |               | 4    | 4.15min  | 6.51min   |
| Friends    | 1    | 1.73min  | 2.51min   | Up in the Air | 1    | 5.61min  | 8.64min   |
|            | 2    | 2.72min  | 4.38min   |               | 2    | 4.22min  | 6.68min   |
|            | 3    | 2.15min  | 3.39min   |               | 3    | 4.57min  | 7.13min   |
|            | 4    | 4.79min  | 7.80min   |               | 4    | 3.28min  | 4.89min   |

Table 2. Accuracy of our importance rank detection method. Our learning strategy achieves accurate enough results for our importance ranking task.

| Movie Name | Clip | Number of Frames | Accuracy | Movie Name    | Clip | Number of Frames | Accuracy |
|------------|------|------------------|----------|---------------|------|------------------|----------|
| Titanic    | 1    | 393              | 0.796    | The Message   | 1    | 258              | 0.751    |
|            | 2    | 305              | 0.768    |               | 2    | 308              | 0.838    |
|            | 3    | 511              | 0.714    |               | 3    | 622              | 0.741    |
|            | 4    | 407              | 0.758    |               | 4    | 491              | 0.750    |
| Friends    | 1    | 210              | 0.742    | Up in the Air | 1    | 675              | 0.791    |
|            | 2    | 326              | 0.802    |               | 2    | 334              | 0.783    |
|            | 3    | 258              | 0.763    |               | 3    | 548              | 0.762    |
|            | 4    | 571              | 0.745    |               | 4    | 397              | 0.812    |

in our method. Then, we provide the results and evaluation of individual modules in our system, which include the efficiency of our system, accuracy and recall of importance rank, panel allocation, balloon generation, and balloon placement. Thirdly, a subjective experiment is conducted to evaluate the quality of our results, as compared to a state-of-the-art comic generation system.

#### 4.1 Experiment Setting

As shown in Table 1, the inputs come from 16 clips of 4 different movies, including Titanic, The Message, Friends, and Up in the Air. The duration of input videos varies from 2 to 6 minutes and each clip from the four given movies has a subtitle to generate word balloons. For every clip, we record the consumed time to generate a manga book using our system and calculate the mean consumed time.

#### 4.2 Visual Comparison with Prior Work

In this part, we will show some end results of our system and compare them with the results of some existing works [11]. As shown in Fig 11, our work is superior to the other comparison methods in three aspects. Firstly, our system can generate more abundant balloon shapes for word balloon, instead the existing methods only use simplex elliptical word balloon. Secondly, we employ text summarization to merge some related subtitles so that we can ensure the sentence in a word balloon is not too long. Thirdly, we provide fully automatic multi-page layouts by obtaining four important parameters automatically. And the result of our layout behaves reasonable and abundant.



Fig. 11. Comparisons of our method with Content-Aware Video2Comics [11]. (a)-(d) are our results. (e)-(h) are the results of [11]. (a) and (e): Titanic (1997) (20th Century Fox, Paramount Pictures and Lightstorm Entertainment). (b) and (f): The Message (Huayi Brothers). (c) and (g): Friends [Bright/Kauffman/Crane Productions, Warner Bros. Television, NBC and Warner Bros. Television Distribution (worldwide)]. (d) and (h): Up in the Air (DW Studios, The Montecito Picture Company, Rickshaw Productions and Paramount Pictures).

### 4.3 Evaluation of Individual Modules

**Importance Rank.** Importance rank is used to allocate importance to panels on a page. In our implementation, we utilize the neural network proposed in [27] to learn and predict the panel importance. In order to validate the accuracy of this learning approach, we also compare the results with the ground truth that are manually labeled by human. Table 2 shows that our designed learning strategy is accurate enough for this purpose. As for the calculation of accuracy, for each video clip, we asked ten people to mark the importance of the panels ranking from 1 to 4. The manually obtained importance is the ground truth, and then, we compare it with our results to compute the accuracy of our method.

**Panel Allocation.** Panel Allocation is used to allocate keyframes to different pages. To evaluate all these constraints, we employ Equation (1). In our implementation, the coefficient  $\alpha_1$ - $\alpha_5$  are set to 3,3,2,1,1 respectively which can get the best results. To validate the results, we asked ten people to allocate panels manually for each video clip. Each allocation can be represented by a numerical sequence, and we can compare our results with manual obtained allocation numerical sequence to



Table 3. Accuracy of panel allocation method. We compare our results with manually allocated panels and our method achieves reasonable performance.

| Movie Name | Clip | Accuracy | Movie Name    | Clip | Accuracy |
|------------|------|----------|---------------|------|----------|
| Titanic    | 1    | 0.713    | The Message   | 1    | 0.625    |
|            | 2    | 0.654    |               | 2    | 0.580    |
|            | 3    | 0.697    |               | 3    | 0.698    |
|            | 4    | 0.665    |               | 4    | 0.642    |
| Friends    | 1    | 0.640    | Up in the Air | 1    | 0.715    |
|            | 2    | 0.691    |               | 2    | 0.599    |
|            | 3    | 0.593    |               | 3    | 0.625    |
|            | 4    | 0.585    |               | 4    | 0.655    |

Table 4. Accuracy and recall of our panel shape classifier. Among three different kinds of classifiers, support vector machine (SVM) works the best with the highest accuracy and medium recall.

| Model        | Accuracy | Recall |
|--------------|----------|--------|
| SVM          | 0.994    | 0.835  |
| DecisionTree | 0.959    | 0.857  |
| RandonForest | 0.969    | 0.796  |

compute the accuracy. The result is shown in Table 3. Our Panel Allocation Method is accurate and reasonable enough for this module.

**Balloon Shape Selection.** Firstly, for each subtitle, we select a balloon shape for it manually. The ground truth we get in this step is used to train a classifier that will be used for balloon shape selection. Then we get the emotion rank of the subtitles and the valence and arousal score of their corresponding audio with an existing tool "Mixed Emotion". Finally, we put the data we get before into a classifier for training. In our implementation, we employ three different kinds of classifiers: support vector machine (SVM), DecisionTree, and RandomForest. The accuracy and recall are shown in Table 4 and it shows that SVM works best because the accuracy of SVM is highest and the recall of it is medium.

#### 4.4 User Study

We also conduct a user study to further evaluate the effectiveness of our system, and four questions are included in our questionnaire. In our experiments, we recruit 40 participants via Amazon Mechanical Turk to compare our results with those by [11] and evaluate different aspects of the results. The participants are first asked to watch a video and then read the comics generated by either our method or [11], then they answer several questions regarding different aspects of generated comics by rating them using a scale from 1 to 5, with 1 being the worst and 5 being the best. The videos and the corresponding comics are presented in a random order to avoid subjective bias. The aforementioned questions are listed as follows:

- Q1:** To what degree do you think the quality of balloons (in style and position)?
- Q2:** To what degree do you think the visual quality of layouts?
- Q3:** To what degree do you think the ability of comics in expressing the video contents?
- Q4:** To what degree do you think the ability of comics in providing you with engaging reading experience?



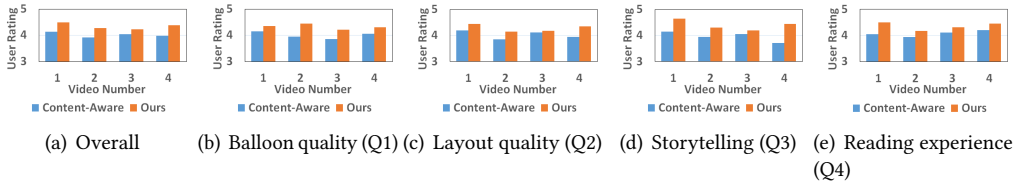


Fig. 12. Results of User Study. The vertical axis is the average rating score of each question. The horizontal axis denotes the video number, 1,2,3,4 stand for the films Titanic (1997) (20th Century Fox, Paramount Pictures and Lightstorm Entertainment), The Message (Huayi Brothers), Friends [Bright/Kauffman/Crane Productions, Warner Bros. Television, NBC and Warner Bros. Television Distribution (worldwide)] and Up in the Air (DW Studios, The Montecito Picture Company, Rickshaw Productions and Paramount Pictures). According to an unpaired t-test, all the preferences are statistically significant ( $p < 0.05$ ).

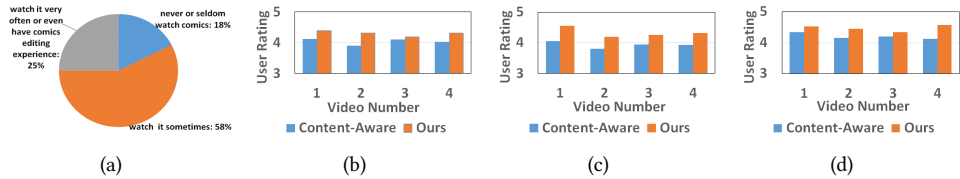


Fig. 13. (a) refers to the information about users' familiarity with comics. (b), (c) and (d) are the average score of each question for the users who never or seldom watch comics, who watch it sometimes and who watch it very often or even have comics editing experiences, respectively.

**Result.** As shown in Fig 12 (a)-(j), for every question and movie, our system performs better than the method [11] (content-aware) no matter whether participants have watched the videos before. According to the readers' feedback (the data can be obtained in the supplementary file), emotion-adaptive varying balloon types and font sizes make our results look visually richer and can better express the stories, while [11] only use one type of word balloon. Also, our layouts are visually diverse and closely resembles real manga styles.

We try to further analyze the results of our subjective evaluations statistically. Using an unpaired t-test, we find that, for each of Q1-Q4, there is a statistically significant difference in subjects choosing our method over the method [11] (all  $p$ -value  $< 0.05$ ). This is expected, as our method uses multi-page semantically rich panel layouts and emotion-aware word balloons together, and thus has the ability to generate a higher quality of balloons (Q1), design more attractive layouts (Q2), present better visual contents (Q3), and all together make our manga presentation more natural and delightful (Q4).

We also analyze the responses of different types of users based on their familiarity with comics. As shown by Fig 13(a), we analyze our results of three types of users: 1) those who know nothing about comics or seldom read comics, 2) those who sometimes read comics and 3) those who often read comics or even have comics editing experience. The average rating scores are summarized in Fig 13(b)-(d). Our methods performs better in each group of users. The unpaired t-test demonstrates that, there are statistically significant differences (all  $p$ -value  $< 0.05$ ) for each group.

## 5 CONCLUSION

In this paper, we have presented a system that is capable of generating high-quality comics without any user inputs. At the core of our system is a multi-page layout framework that jointly organizes multiple pages in visually rich layouts using rich semantics from videos, and an emotion-aware balloon generation method that can create a variety of balloon shapes and font sizes according to emotion contained in subtitles and audio. Our experiments demonstrate that our system can synthesize more expressive and engaging comics, in comparison to a state-of-the-art comic generation system. Although our system has been shown to achieve promising results, it is still subject to several limitations. For example, the keyframe selection is not accurate enough. In some cases, the selected keyframes are similar to each other, which would certainly introduce redundancy into generated comics. It would be interesting to develop an end-to-end neural network for keyframe selection, which is left for future work. Furthermore, it is time-consuming to generate comics from a video without any subtitles. That is because speech recognition, which is required for subtitle extraction, is often prone to errors, and thus many manual efforts are needed to refine the recognition results. We believe that, with the advent of more sophisticated speech recognition technique, this problem can be alleviated greatly. What's more, inspired by many existing methods [3, 14, 19] which can generate image sequences given a story with multiple sentences, it is possible to produce comic books from textual stories and we are interested to extend our method to leverage textual information to help generating manga.

## REFERENCES

- [1] P. Buitelaar, I. D. Wood, S. Negi, M. Arcan, J. P. McCrae, A. Abele, C. Robin, V. Andryushechkin, H. Ziad, H. Sagha, M. Schmitt, B. W. Schuller, J. F. Sánchez-Rada, C. A. Iglesias, C. Navarro, A. Giefer, N. Heise, V. Masucci, F. A. Danza, C. Caterino, P. Smrž, M. Hradiš, F. Povolný, M. Klimeš, P. Matějka, and G. Tummarello. 2018. MixedEmotions: An Open-Source Toolbox for Multimodal Emotion Analysis. *IEEE Transactions on Multimedia* 20, 9 (2018), 2454–2465.
- [2] Ying Cao, Antoni B. Chan, and Rynson W. H. Lau. 2012. Automatic stylistic manga layout. *Acm Transactions on Graphics* 31, 6 (2012), 1–10.
- [3] Shizhe Chen, Bei Liu, Jianlong Fu, Ruihua Song, Qin Jin, Pingping Lin, Xiaoyu Qi, Chunting Wang, and Jin Zhou. 2019. Neural Storyboard Artist: Visualizing Stories with Coherent Image Sequences. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2236–2244.
- [4] Wei Ta Chu, Chia Hsiang Yu, and Hsin Han Wang. 2015. Optimized Comics-Based Storytelling for Temporal Image Sequences. *IEEE Transactions on Multimedia* 17, 2 (2015), 201–215.
- [5] Katja Filippova. 2013. Multi-sentence compression: finding shortest paths in word graphs. In *COLING 2010, International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*. 322–330.
- [6] C. Forceville, T. Veale, and K. Feyaerts. 2010. *Balloonics: the visuals of balloons in comics*. 232–236 pages.
- [7] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. (2016), 2414–2423.
- [8] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. 2018. Arbitrary Style Transfer with Deep Feature Reshuffle. (2018), 8222–8231.
- [9] Goldberg D E Harik G R, Lobo F G. 1999. The compact genetic algorithm. In *Evolutionary Computation IEEE Transactions*. 287–297.
- [10] Richang Hong, Xiao Tong Yuan, Mengdi Xu, Meng Wang, Shuicheng Yan, and Tat Seng Chua. 2010. Movie2Comics:a feast of multimedia artwork. In *International Conference on Multimedia 2010, Firenze, Italy, October*. 611–614.
- [11] Guangmei Jing, Yongtao Hu, Yanwen Guo, Yizhou Yu, and Wenping Wang. 2015. Content-Aware Video2Comics With Manga-Style Layout. *IEEE Transactions on Multimedia* 17, 12 (2015), 2122–2133.
- [12] Justin Johnson, Alexandre Alahi, and Li Feifei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. (2016), 694–711.
- [13] David Kurlander, Tim Skelly, and David Salesin. 1996. Comic Chat. In *Conference on Computer Graphics and Interactive Techniques*. 225–236.
- [14] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. 2019. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6329–6338.

- [15] Aude Oliva and Antonio Torralba. 2001. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision* 42, 3 (2001), 145–175.
- [16] Dae Young Park and Kwang Hee Lee. 2019. Arbitrary Style Transfer With Style-Attentional Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [17] Jacqueline Preu. 2007. From movie to comic, informed by the screenplay. In *ACM SIGGRAPH*. 99.
- [18] Zhong Qu, Lidan Lin, Tengfei Gao, and Yongkun Wang. 2013. An Improved Keyframe Extraction Method Based on HSV Colour Space. *Journal of Software* 8, 7 (2013).
- [19] Hareesh Ravi, Lezi Wang, Carlos Muniz, Leonid Sigal, Dimitris Metaxas, and Mubbasir Kapadia. 2018. Show me a story: Towards coherent neural story illustration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7613–7621.
- [20] Dong Sung Ryu, Soo Hyun Park, Jeong Won Lee, Do Hoon Lee, and Hwan Gue Cho. 2008. CINETOON: A Semi-automated System for Rendering Black/White Comic Books from Video Streams. In *IEEE International Conference on Computer and Information Technology Workshops*. 336–341.
- [21] Tomoya Sawada, Masahiro Toyoura, and Xiaoyang Mao. 2013. Film Comic Generation with Eye Tracking. In *International Conference on Multimedia Modeling*. 467–478.
- [22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1–9.
- [23] Masahiro Toyoura, Mamoru Kunihiro, and Xiaoyang Mao. 2012. Film Comic Reflecting Camera-Works. In *Advances in Multimedia Modeling - International Conference, MMM 2012, Klagenfurt, Austria, January 4-6, 2012. Proceedings*. 406–417.
- [24] Meng Wang, Richang Hong, Xiao Tong Yuan, Shuicheng Yan, and Tat Seng Chua. 2012. Movie2Comics: Towards a Lively Video Content Presentation. *IEEE Transactions on Multimedia* 14, 3 (2012), 858–870.
- [25] Holger Winnemöller. 2011. XDoG: advanced image stylization with eXtended Difference-of-Gaussians. In *ACM Siggraph/eurographics Symposium on Non-Photorealistic Animation and Rendering*. 147–156.
- [26] Holger Winnemöller, Jan Eric Kyprianidis, and Sven C. Olsen. 2012. XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization. *Computers & Graphics* 36, 6 (2012), 740–753.
- [27] Ke Zhang, Wei Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video Summarization with Long Short-Term Memory. (2016), 766–782.
- [28] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning Deep Features for Discriminative Localization. In *Computer Vision and Pattern Recognition*. 2921–2929.