# Speech Bubble Aware Automatic Comic Colorization

**Wei-Hsiang, Shen**
tommyrpg1010@gmail.com

**Yu-Shiang, Tsai**
charlie6513@gmail.com

**Jian-Qi, Pan**
jessepan82@gmail.com


Figure 1: From left to right, input gray-scale image, colorized image, colorized image with speech bubble awareness

## 1 Introduction

Comics have attracted readers from all over the world. Most comics are in black and white, so there is a strong demand for comic colorization techniques. However, colorizing a comic is a time-consuming task that requires hours of works from professional artists.

In this work, we propose a fully automatic comic colorization model that is aware of speech bubbles inside the image, and avoid wrong or inconsistent colorization inside the speech bubbles, as shown in Fig [1].

In summary, in this work we present the following contributions:

- Novel speech bubble segmentation technique
- Automatic comic colorization model with speech bubble awareness
- Open source full implementation of the model in TensorFlow 2.0

## 2 Related Works

Iizuka, et al. [1]. proposed a image colorization model that uses an end-to-end convolutional neural network that fuses local and global features. The global feature networks that consists of fully-connected layers allow the network to learn the global features and thus increase the receptive field of each pixel. However, in our case, we are not able to train the global feature network since we do not have prelabeled classification labels. Also, we would like to point out that comic colorization and image colorization are different problems, as shown in Fig [2]. Gray-scale images of realistic photos contain much more information than gray-scale comic images. Details of how we process the comic image can be viewed in supplementary material.

Furusawa, et al. [2]. proposed a similar model based on the work of Iizuka, et al. [1]. They use character names to train the global feature network. However, they sometimes require user to pin point the color dots, and the system fails when multiple characters appear. Their system is semi-automatic while we would like a fully automatic system.

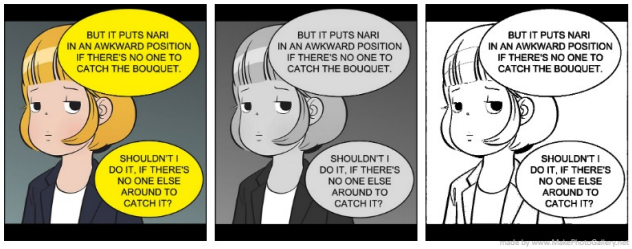


Figure 2: From left to right, color image, gray-scale image of realistic photo image, gray-scale image defined in comic

Kang, et al. [3]. proposed a fully automatic comic colorization system that take care of the color ambiguity problem in the background region and achieve great result. However, we notice that in their work, the color in the speech bubbles in inconsistent and sometimes interfered by color outside the speech bubbles. Therefore, in this project, we aim to solve the colorization inside speech bubbles.

## 3 Methods

Our method consists of three parts, low resolution colorizer, polishing network, and speech bubble segmentation (Fig [3]).

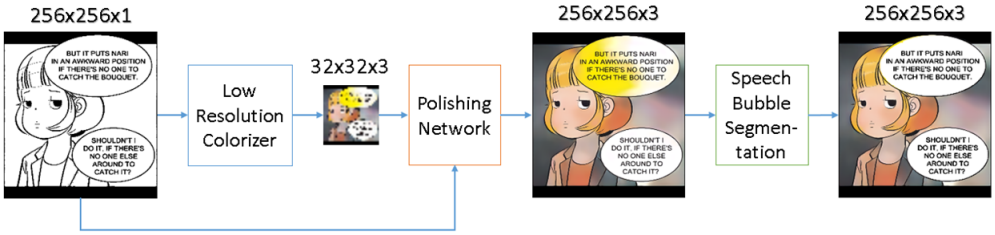


Figure 3: System pipeline

### 3.1 Low Resolution Colorizer

In the work of Furusawa, et al. [2], they pointed out that color is a low spatial frequency information, thus it is sufficient to predict a low resolution chrominance map. We also verify the assumption by downsampling the origianl 256x256 chrominance map to 28x28 , then upsample back to 256x256, and then combine it with the original luminance in Fig [4]. We can see that the resulting colorized images look very similar to the original images. The low resolution colorizer is a fully convolutional neural network (Fig [9]).

Comparing to a full resolution colorizer, a low resolution colorizer is more stable to train and converges to a better loss. We tried to train a full resolution colorizer but it failed to produce quality result. The detail architecture can be viewed in supplementary material.

### 3.2 Polishing Network

Since we predict a colorized image in low resolution, we need a super resolution network to upsample it back to the original size. The polishing network can exploit the information of the full resolution gray-scale image, like the network structure used in the work of Iizuka, et al. [1]. However, we let the full resolution gray-scale image to go through some convolution layers, since our gray-scale image is the outline not the true luminance of the output. Our polishing network is a fully convolution network that concatenate the full resolution gray-scale image (Fig [10]).

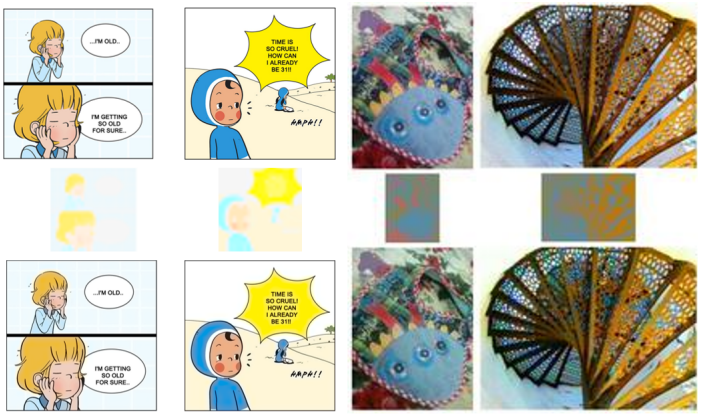

Figure 4: Verification of low resolution chrominance map

### 3.3 Speech Bubble Segmentation

In comics, there are typically many speech bubbles (also called speech balloons) that contain texts inside. The colorizer itself cannot identify the bubbles regions and so it gets confused and try to colorize the bubbles, which cause unwanted artifacts Fig [6](c). We assume that all the speech bubbles should be black and white (it should not be colorized by the colorizer). We develop a novel approach to segment the speech bubbles without needing any form of pre-labeled information.

Our segmentation system comes with five steps (Fig [5]):

1. Text detection using efficient and accurate scene text detector (EAST Zhou et al. [5])
2. Bounding box clustering to eliminate lonely boxes
3. Merge the separated bounding boxes into one big box
4. Flooding filling to segment the speech bubbles
5. Hole filling to eliminate holes caused by text

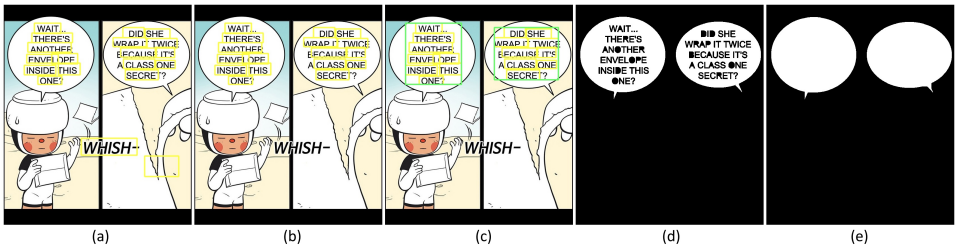


Figure 5: Pipeline of speech bubble segmentation.

An Efficient and Accurate Scene Text Detector (also known as EAST) (Zhou et al. [5]) is a state-of-the-art approach of scene text detection. EAST is robust to detect text in real-world images, which is far more difficult than typical comic text (mostly computer font text). In this project, we use the OpenCV implementation of the EAST and the pretrained weights from the original author.

We then make a bold assumption that many text bounding boxes should be detected if the text is inside a speech bubbles. This assumption showed to work empirically. Bounding boxes that are not clustered (who do not have a near neighbor box) can be eliminated. Without this step, many vocal texts would be mistaken as a speech bubble.

We choose morphological closing operation as our hole filling method. Many other algorithms can also be used (ex. contour models...), but in this project we stick with the morphological opertaion since it seems to work nicely.

## 4 Results

Our results are shown in Fig [6]. By comparing (c) and (f), we can notice that our method generate better colorized result at the speech bubble regions, while the speech bubbles in (c) is inconsistent in color.

Our result color of background region is inconsistent due to the fact we did not implement the background detector proposed by Kang et al. [3]. This is an expected outcome.

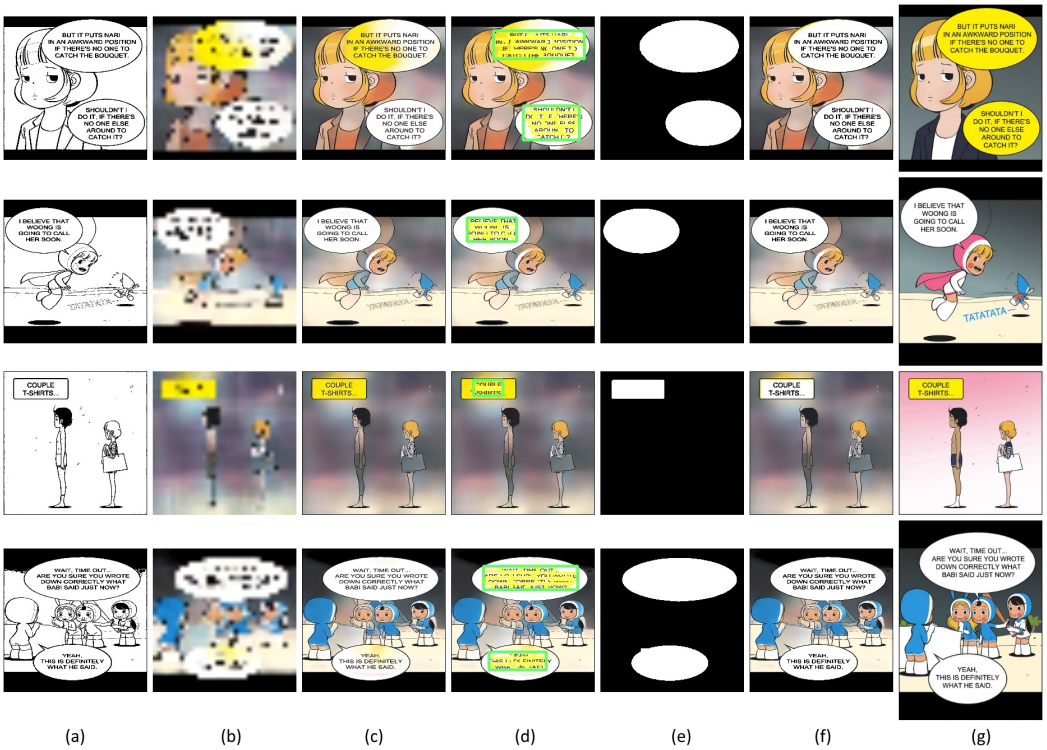


Figure 6: (a) Input gray-scale image, (b) Low-resolution colorized image, (c) Super-resolution color image, (d) Text detection, (e) Speech bubble segmentation, (f) Speech bubble aware colorization, (g) ground truth. Notice the difference in (c) and (f), without speech bubble awareness the colorization is likely to fail in speech bubble regions.

### 4.1 Failure Cases of Speech Bubble Segmentation

Most of the time the speech bubble segmentation would work, but there is some situation that it fails. We analyse the three common cases that it may fail (Fig [7]). (a), there are comic objects and contain texts inside and looks like a speech bubble. (b), two text clusters are too close and are treated as one. (c) Fail text detection by EAST.

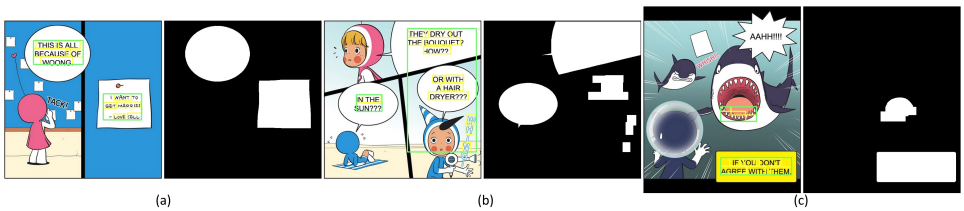


Figure 7: Failure case of speech bubble segmentation

### 4.2 Generalization to other Comics

We also test our model on other comic that contain characters and backgrounds that are never seen by the model (Fig [8]). We can see that the model fail to generalize to unseen characters, which is the typical downfall of neural networks and most learning-based technique.

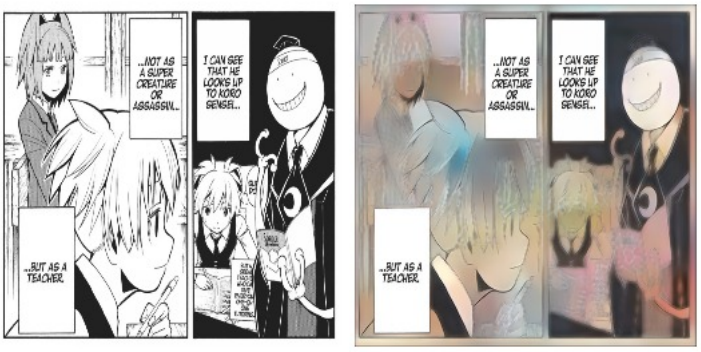


Figure 8: The model fails to generalize to unseen characters

## 5 Conclusion and Future Works

In this project, we proposed a novel speech bubble segmentation technique and combine it with comic colorization. From the result, it show improvement in the speech bubble regions comparing the work of Kang, et al.[3].

The proposed system has some straightforward extensions:

1. Our polishing network behaves like a super resolution network. Adverserial loss training is proven to be beneficial to generate realistic super resolution images (Ledig et al. [6]). Also, adverserial loss could let the model to generate more realistic colorization images as well.
2. As discussed in section 4.1, our method of speech bubble segmentation may have some failures due to non-adaptive parameters choice of flood filling and morphological operation. Therefore, we suggest using a neural network dedicate to segment the speech bubbles. Previous research showed great result on image segmentation (Ronneberger et al. [7]). We believe after some user interaction to eliminate wrong segmentations of our algorithm, a neural network can perform better since the feature of speech bubbles is obvious.

## References

[1] Satoshi Iizuka, Edgar Simo-Serra & Hiroshi Ishikawa. (2016) "Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification".

[2] Chie Furusawa, Kazutuki Hiroshiba, Keisuke Ogaki & Yuri Odagiri. (2017) "Comicolorization: Semi-Automatic Manga Colorization".

[3] Sungmin Kang, Jaegul Choo & Jaehyuk Chang. (2017) "Consistent Comic Colorization with Pixel-wise Background Classification".

[4] Sergio Guadarrama, Ryan Dahl, David Bieber, Mohammad Norouzi, Jonathon Shlens & Kevin Murphy. (2017) "PixColor: Pixel Recursive Colorization".

[5] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He & Jiajun Liang. (2017) "EAST: An Efficient and Accurate Scene Text Detector".

[6] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang & Wenzhe Shi. (2017) "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network".

[7] Olaf Ronneberger, Philipp Fischer & Thomas Brox. (2015) "U-Net: Convolutional Networks for Biomedical Image Segmentation".

## Supplementary Material

All code can be viewed at GitHub: "https://github.com/Rabbit1010/Speech-Bubble-Aware-Automatic-Comic-Colorization"

### Dataset Preparation

The dataset and preprocess methods we used is the same as the work of Kang et al. [3]. The dataset we used is the comic Yumi's Cell (English version), from the first episode to episode 238. However, they use a train validation split ratio of 95% and 5%, while we use a ratio of 80% and 20%.

### Low Resolution Colorizer

While in the work of Kang et al. [3], they also have a low resolution colorizer and polishing network, we use a simpler architecture for our work. Our low resolution colorizer is a simple sequential fully convolutional neural network (Fig [9]).

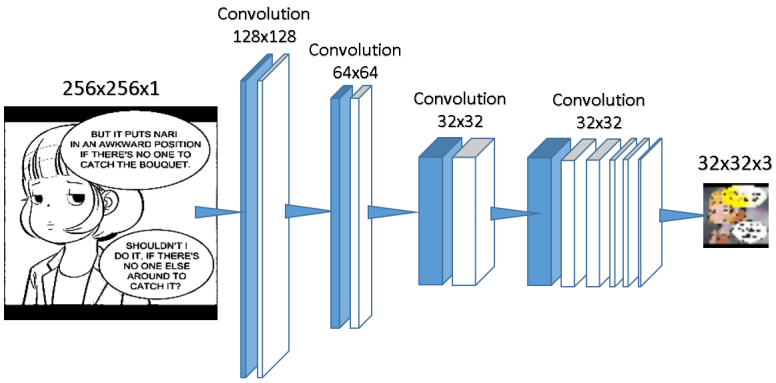


Figure 9: Low resolution colorizer

Table 1: Architecture of low resolution colorizer

| Layer | Features number | Stride |
|---|---|---|
| Conv2D | 64 | (2,2) |
| Conv2D | 128 | (1,1) |
| Conv2D | 128 | (2,2) |
| Conv2D | 256 | (1,1) |
| Conv2D | 256 | (2,2) |
| Conv2D | 512 | (1,1) |
| Conv2D | 512 | (1,1) |
| Conv2D | 256 | (1,1) |
| Conv2D | 128 | (1,1) |
| Conv2D | 64 | (1,1) |
| Conv2D | 32 | (1,1) |
| Conv2D | 3 | (1,1) |

Each convolution layer is followed by a batch normalization layers then a ReLu activation function. All filters are 3x3 size, and padding are used to maintain image resolution during convolution. MSE loss and Adam optimizer is used for training.

### Polishing Network

The polishing network is an autoencoder-like neural network (Fig [10]). Each convolution layer is followed by a batch normalization layers then an activation function. Leaky ReLu activation with

parameter = 0.2 is used for the encoder part, ReLu activation is used for all other convolution layers. All filters are 3x3 size, and padding are used to maintain image resolution during convolution. MSE loss and Adam optimizer is used for training.

For concatenating layers, see Fig [10] for more details.

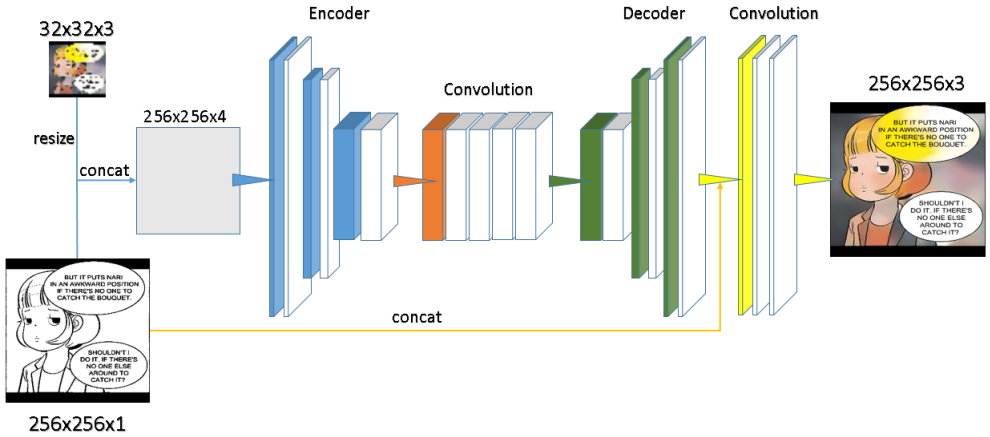


Figure 10: Polishing network

Table 2: Architecture of polishing network

| Layer | Features number | Stride |
|---|---|---|
| Conv2D | 64 | (2,2) |
| Conv2D | 128 | (1,1) |
| Conv2D | 128 | (2,2) |
| Conv2D | 256 | (1,1) |
| Conv2D | 256 | (2,2) |
| Conv2D | 512 | (1,1) |
| Conv2D | 512 | (1,1) |
| Conv2D | 512 | (1,1) |
| Conv2D | 512 | (1,1) |
| Conv2D | 512 | (1,1) |
| Conv2D | 512 | (1,1) |
| UpSampling2D | | |
| Conv2D | 512 | (1,1) |
| Conv2D | 256 | (1,1) |
| UpSampling2D | | |
| Conv2D | 256 | (1,1) |
| Conv2D | 128 | (1,1) |
| UpSampling2D | | |
| Conv2D | 128 | (1,1) |
| Conv2D | 64 | (1,1) |
| Conv2D | 3 | (1,1) |