Continual few-shot patch-based learning for anime-style colorization

Akinobu Maejima
OLM Digital, Inc.
IMAGICA GROUP Inc.
Tokyo, Japan
akinobu.maejima@olm.co.jp

Seitaro Shinagawa
NAIST
Nara, Japan
sei.shinagawa@is.naist.jp

Hiroyuki Kubo
Chiba University
Chiba, Japan
hkubo@chiba-u.jp

Takuya Funatomi
NAIST
Nara, Japan
funatomi@is.naist.jp

Tatsuo Yotsukura
OLM Digital, Inc.
IMAGICA GROUP Inc.
Tokyo, Japan
yotsukura@olm.co.jp

Satoshi Nakamura
NAIST
Nara, Japan
s-nakamura@is.naist.jp

Yasuhiro Mukaigawa
NAIST
Nara, Japan
mukaigawa@is.naist.jp

Abstract

The automatic colorization of anime line drawings is a challenging problem in production pipelines. Recent advances in deep neural networks have addressed this problem; however, collecting many images of colorization targets in novel anime work before the colorization process starts leads to chicken-and-egg problems and has become an obstacle to using them in production pipelines. To overcome this obstacle, we propose a new patch-based learning method for few-shot anime-style colorization. The learning method adopts an efficient patch sampling technique with position embedding according to the characteristics of anime line drawings. We also present a continuous learning strategy that continuously updates our colorization model using new samples colorized by human artists. The advantage of our method is that it can learn our colorization model from scratch or pre-trained weights using only a few pre- and post-colorized line drawings that are created by artists in their usual colorization work. Therefore, our method can be easily incorporated within existing production pipelines. We quantitatively demonstrate that our colorization method outperforms state-of-the-art methods.

Keywords: anime, colorization, few-shot learning, continuous learning strategy

1. Introduction

Anime is a limited animation technique drawn by hand or sometimes using computer graphics. It may use exaggerated and simplified artistic expressions in a character's face and motion that are physically inaccurate. Thus, its production relies entirely on manual labor. Because the number of anime works has increased over the past decade, studios have explored improving their production pipelines while preserving the quality of the work. In a traditional anime production pipeline (see Fig. 1 and Appendix A), the colorization of draft line drawings is a tedious and time-consuming process, in which colorization artists carefully fill each region surrounded by contour lines with a single color (see Fig. 2) specified on a color palette designed by a color director. We refer to this style of colorization as *anime-style colorization*.

The anime-style colorization process is often formulated as a region correspondence problem that is solved using graph matching [8, 18, 12, 11], region tracking [29], or a data-driven approach using deep neural networks [2]. Region correspondence approaches may achieve precise colorization, but the computational cost may increase quadratically with respect to the number of regions. This is sometimes problematic in a production pipeline that needs to process complex line drawings (e.g., see Fig. 3).

As a different approach, several researchers have formulated the colorization problem as a semantic segmentation task using deep neural networks [7, 16]. The computational

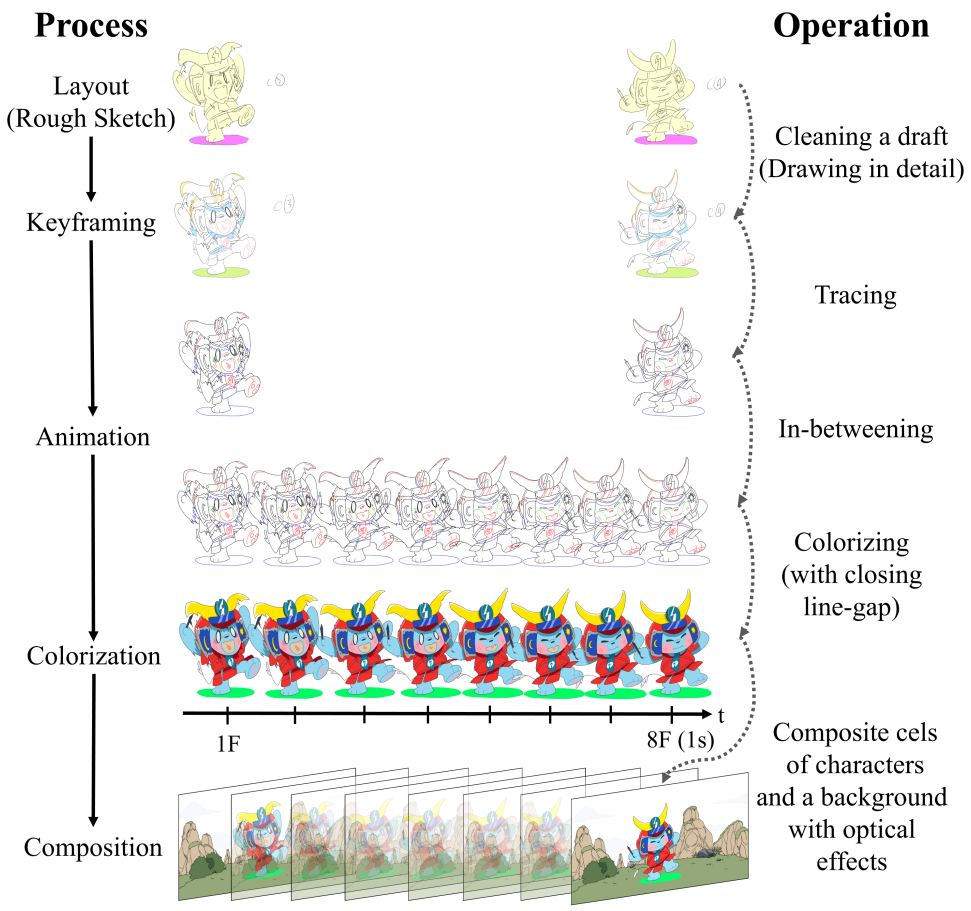
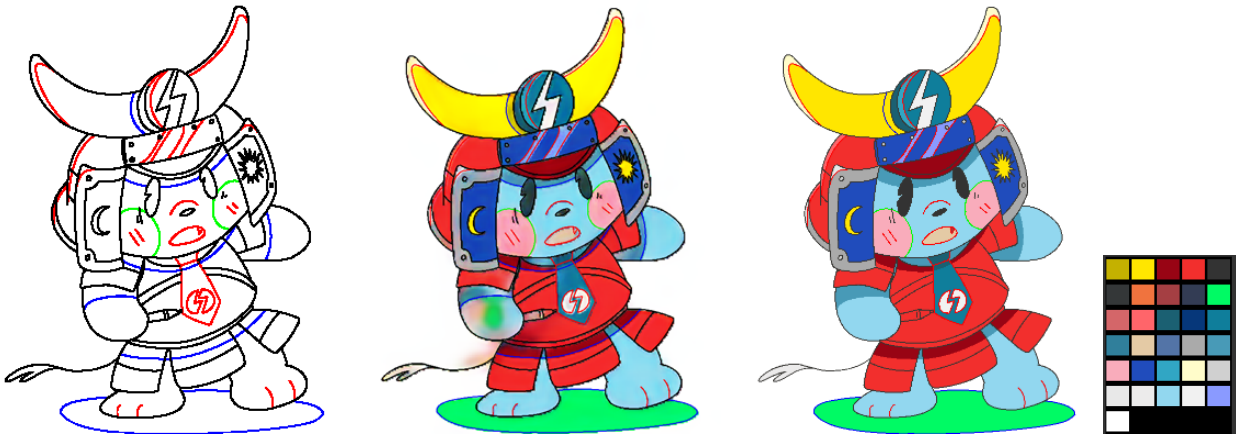


Figure 1: Example of a common anime production workflow from layout to the composition process for a sequence. Images originate from *Deadline* © OLM Asia SDN BHD.

cost of such methods depends on the resolution of the image and the number of colors corresponding to the class labels, not the number of regions. In most cases, the number of colors is less than the number of regions in all frames to be colorized. Therefore, we attempted to use this approach in production, but encountered a substantial issue: *sufficient training data are difficult to collect*

It is technically possible to prepare a large-scale dataset from existing anime works for model training. However, we would need to discuss whether this is acceptable for stakeholders with regard to intellectual property and copyright management. Even if we could obtain a large-scale dataset and train a colorization model, we are not certain that the model would function correctly on novel anime work because the drawing styles of each anime work are quite different. Another issue concerns the requirements for the colorization model used in production: accuracy must match production-ready quality and the overall processing time must be sufficiently short to avoid disturbing the production pipeline. To address the aforementioned issues, we propose a practical anime-style colorization method that uses a few pre- and post-colorized line drawings inspired by the patch-based learning approach [24]. A naive (random) patch sampling method leads to ambiguous correspondence between patches from such line drawings, which is inefficient and causes accuracy degradation in model training. We solve this problem using novel patch sampling with position embedding that is specialized for anime. The advantage of our method is that it is able to learn a colorization model for every target sequence from scratch. We also introduce a learning strategy that continuously updates the colorization model using new samples colorized by human artists. This strategy gradually improves the performance of the model by slightly inheriting current network weights after every colorization process while considering intellectual property and copyright management. As a result, this makes the colorization process faster while maintaining accuracy.

Our contributions in this study can be summarized as fol-



(a) (b) (c) (d)

Figure 2: Different painting styles: (a) line drawing, (b) watercolor painting-style colorization, and (c) anime-style colorization. (d) The anime-style image is colorized with limited colors from a character-specific color palette. Images originate from *Deadline* © OLM Asia SDN BHD.

lows:

- We propose an anime-specific few-shot patch-based learning method with a continuous learning strategy for faster colorization while maintaining production-ready accuracy.
- We quantitatively demonstrate that our method achieves state-of-the-art colorization accuracy within acceptable processing time for artists, and it can be applied to characters that appear in novel anime works.
- We found that colored indicator lines (commonly used in a traditional production pipeline) are useful for providing guidance not only to human artists but also to the colorization model, improving colorization accuracy.

2. Related work

The anime-style colorization of draft line drawings is essentially different from the colorization of classic monochromatic film, hand-drawn sketches, and manga. The anime-style colorization is often formulated as a region correspondence problem, and the others are essentially an optimization problem between a source image and constraints provided by color scribbles, curves, reference images, or priors. In the following, we classify existing colorization methods into three approaches: color propagation via region correspondence, color prediction, and colorization methods based on color propagation with color scribbles. Then we discuss how the proposed method differs from those in each category.

2.1. Color propagation via correspondence

The methods in this category are based on propagating colors via correspondence between components. They essentially consist of three steps. First, they extract components from an input and a reference (colorized) sketch and then determine the correspondences between components. Finally, they propagate colors or color labels from a reference to the input sketches according to the correspondences. Such region correspondence can be computed by solving a quadratic assignment problem or graph matching [8, 18, 12, 11] with active learning [3], template matching [22], patch match [1], globally optimal region tracking [29], using a model trained to induce correspondences between input and reference components [4], or Animation Transformer (AnT) [2].

The results of these methods are flat coloring that satisfies the requirements of anime-style colorization. However, when many regions consist of a few pixels (so-called micro-regions), the number of mapping errors tends to increase, and the accuracy of colorization tends to decrease as a result of the graph matching based approach. Accuracy is also degraded when a region is either split, merged, lost, or generated in successive frames. Our method learns the mapping between pre- and post-colorized line drawing patches, and infers per-pixel color labels for the input line drawing patches. Hence, the aforementioned micro-regions do not interfere with the correct mapping. Liu *et al*. [11] addressed region correspondences involving significant deformation and topological change between two cel shapes. However, they assumed that the input line drawings were stored in vector image format. Vectorizing line drawings in raster image format is not trivial and can contain reconstruc-

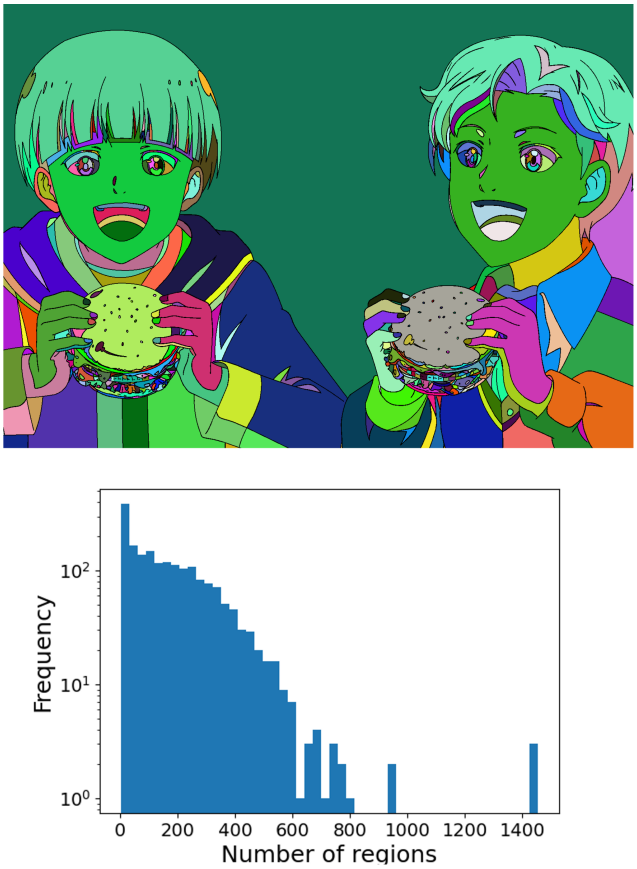


Figure 3: Above: a complex line drawing example consisting of approximately 1000 regions (painted with random colors). Below: frequency of the number of regions for each frame of 1881 sequences in 12 episodes of a TV Anime series. The example drawing above originates from *Restaurant to Another World 2*, © Junpei Inuzuka, IMAGICA INFOS/Restaurant to Another World 2 Project.

tion errors unacceptable to artists. Globally Optimal Toon Tracking (GOTT) [29] achieves highly accurate region correspondence, however, the cost of precomputation of appearance and motion terms for every pair of regions from two different frames increases quadratically with respect to the number of regions. Depending on the style of anime work, it may be necessary to address complex line drawings that consist of hundreds to thousands of regions in a single frame, as in Fig. 3. In such a case, it is difficult to run GOTT within a reasonable time, regardless of the optimization. AnT is a state-of-the-art colorization method in this category and its implementation is available in the *Cadmium* software package (see `https://cadmium.app/`).

We show in our quantitative study that, despite its simplicity, our method achieves better accuracy than AnT. Moreover, collecting training data for our method is much easier. Specifically, it does not require the consistent labeling of segments across frames, the creation of 3D CG models for pseudo-data generation, or any rendering to create training data.

2.2. Color prediction

The methods in this category predict colors or color labels using priors derived from the correspondence between pre- and post-colorized line drawings. cGAN-based manga colorization [6] is a learning method that obtains a model from a single image for colorizing frames in manga with similar compositions. After it predicts the color for each pixel, it achieves flat coloring by selecting the center value of k-means clustering of pixel RGB values within each segment of the predicted image.

Shi *et al*. have proposed a temporally consistent reference-based colorization method (called LAVC) for line art video using color transform and temporal constraint networks [20]. Li *et al*. also have proposed a reference-based colorization method (SGA) for a line-art sketch using Stop-Gradient Attention [10]. These methods may produce colors that do not exist in the color palette. Ramassamy *et al*. [16] and Ishii *et al*. [7] (referred to as U-net) addressed the colorization task as semantic segmentation, and proposed a method that predicts color labels for each pixel and then votes for color labels within regions of an input line drawing as a post process. To train the above models from scratch, LAVC, SGA, and U-net require a large-scale dataset that has already been manually colorized. In general, fine-tuning pre-trained models is a possible workaround for this issue; it adapts the models to different drawing styles using a small-scale dataset representing a colorization target. In this paper, we assume that only a few reference images are available for model training. Under this assumption, it is difficult to fill the gap between drawing styles by fine-tuning due to lack of training data, and then the colorization accuracy does not satisfy the artists' requirements. Thus, such methods are not applicable to a wide range of content.

Recently, patch-based learning methods using a small number of reference images have been proposed and applied to interactive video style transfer [24]. An interesting point is that they provide patches instead of whole images as input to the network so that it trains a local mapping, before and after stylization. Numerous patches can be obtained from a single image by changing the position used to cut out a patch. Thus, the stylization network can be trained using only a few images. Like Ramassamy *et al*. and Ishii *et al*., we address the colorization problem as a semantic segmentation task; however, we make model training possible using only a few pre- and post-colorized line drawings by such patch-based learning.

We compare our method to U-net, LAVC and SGA, and

demonstrate that our proposed learning method can cope with above mentioned issues in Section 6.5.

2.3. Color propagation with color scribbles

The methods in this category also propagate colors of user-defined scribbles by considering spatiotemporal coherency between neighboring pixels [9] that are interpolated along with user-defined NURBS curves [14], a level-set method with boundary optimization between colors [15], or guide the output colors of deep neural networks [28, 17, 5, 27]. These methods do not guarantee the reproduction of colors in the scribbles and can produce gradations that do not achieve anime-style colorization (see Fig. 2(b)). By contrast, Lazybrush propagates the color labels of the scribbles by solving optimization problems with multi-way cuts [23] and produces flat coloring. Zhang *et al*. [26] also support flat-coloring using a split filling mechanism. The last two methods achieve anime-style colorization; however, they require user-defined color scribbles to allow all line drawings to be colorized. Our method requires manual colorization for a few draft line drawings for each target sequence. Thus, these methods will help artists to paint such frames by hand.

3. Terminology

Before we describe the methodology of our method, we first define the terminology we use

Line drawing $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$: a color image that has a white background, black contour lines, and red, green, or blue indicator lines that represent objects commonly used in anime production (see Figs. 1 and 2(a)). W and H are the width and height of the image, respectively. The red, green, and blue lines indicate boundaries where artists need to paint using different colors, for example, highlight or shadow colors. A set of line drawings is denoted by $\mathcal{I}$.

Color palette C : a dictionary that maps color IDs to actual RGB values (see Fig. 2(d)).

Label map $\mathbf{L} \in \mathbb{N}_o^{W \times H}$: an image whose pixel RGB values are replaced with IDs corresponding to RGB values in color palette C. It is derived from a colorized line drawing. A set of label maps is denoted by $\mathcal{L}$.

Patch : a small piece of an image. It can be extracted from an image by cutting out a square of size M around a specified point. We denote a patch by $\mathbf{P} \in \mathbb{R}^{M \times M \times 3}$ and a set of patches by $\mathcal{P}$.

Reference frame(s) are manually colorized by users. Pre- and post-colorized line drawings in reference frames are used for model training. These line drawings are denoted using superscript R by $\mathcal{I}^R$ and $\mathcal{O}^R$, respectively.

4. Anime-specific patch-based learning

To achieve anime-style colorization in a few-shot setting, we consider the colorization problem as a variant of style transfer, and propose a learning method based on a method of patch-based learning for style transfer for video proposed by Texler *et al*. To adopt this in the colorization task, we make two modifications to the network's architecture: (i) We make the generator network predict labels instead of RGB values on patches using the cross-entropy loss function, and (ii) we include position embedding (PE) to reduce location ambiguity between two pairs of line drawing and label patches to distinguish similar line patterns in different locations. The proposed learning method adopts an efficient sampling technique that considers the characteristics of anime line drawings and a continuous learning strategy to improve colorization accuracy and reduce processing time. We introduce each component next.

4.1. Efficient sampling

Anime line drawings have two domain-specific characteristics. First, uncolored line drawings have less color variation than real images because they are drawn using contour lines with few colors representing objects such as anime characters. Second, the color in the colorized line drawing image only changes at the boundary between adjacent regions or junctions with contour lines. Under these conditions, random or regular patch sampling from a line drawing may contain 'blank' patches with no line, particularly from the background and large segments. Such blank patches make both training and prediction difficult because of the one-to-many mapping from them to background or large segment labels. Of course we can sample patches from everywhere on the image and reject those that are blank patches after sampling. However, sampling only meaningful (non-blank) patches is much more efficient than such a trial and error approach.

We propose to determine sampling points based on classical corner detection [19] for contour lines. Then, the sampling points are randomly selected from the detected corners, within the limit of the number of patches N which is specified by the user. Finally, patches are extracted by cutting out square regions around points on pre- and post-colorized line drawings. As a result, our patch-sampling helps to reject difficult samples and leads to successful training and efficient prediction.

4.2. Position embedding

In patch sampling, some pairs of line drawing and label patches sampled from different locations may have similar line patterns but different labels. Such data may lead to one-to-many mapping, which makes learning difficult. To prevent this scenario, we provide spatial information to the

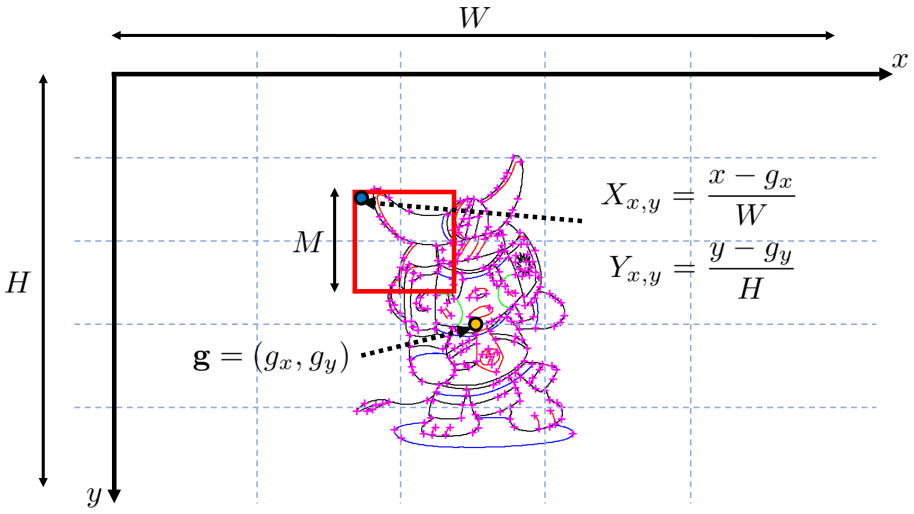


Figure 4: Relative 2D coordinate computation for our PE. Purple crosses represent sampling points determined by our method. Image originates from *Deadline* © OLM Asia SDN BHD.

colorization network to make line drawing patches unique. We achieve this by embedding positional information into sampled line drawing patches as additional channels. We first compute the center of gravity $\mathbf{g}$ of all sampled points in image coordinates and then compute the position of all the pixels in each patch relative to $\mathbf{g}$. Finally, we merge the resulting relative 2D coordinates into each patch as two additional channels before we input the patch into the network (see Fig. 4):

$$\mathbf{P} = \text{merge}(\mathbf{P}, \mathbf{X}, \mathbf{Y}) \in \mathbb{R}^{M \times M \times 5} \tag{1}$$

We show the effectiveness of this position embedding (PE) in Section 6.4.2.

4.3. Learning strategy

Our system needs to continuously learn and colorize line drawings with different visual styles. The uniqueness of drawing styles often becomes noise that renders model training unstable. To achieve stable model training in such a scenario, we introduce the exponential moving average (EMA) method. The EMA can be seen as an ensemble of model parameters during training, and it prevents model training from instability. Specifically, the EMA updates the model parameters Θ of generator G by mixing the previous $\bar{\Theta}_{k-1}$ and current model parameters Θ_k at every k-th iteration of training:

$$\bar{\Theta}_k = (1 - \beta)\bar{\Theta}_{k-1} + \beta\Theta_k, \tag{2}$$

where $\bar{\Theta}_0 = \Theta_0$. We empirically set the hyperparameter β to 0.001. The model parameters $\bar{\Theta}_k$ can be reused, except for the network's last layer where the number of units varies depending on the number of labels in the label maps. Continuous learning begins from $\bar{\Theta}_0$ and is initialized from scratch or pre-trained. We assume that a collection to be processed is stored in a data pool $\mathcal{D}$. The collection consists of a successive sequence of line drawing and label map pairs on each shot. We denote the sequence as a set $\mathcal{I}$ for the line drawings and $\mathcal{L}$ for the label maps. For each training iteration, our continuous learning strategy samples a shot $(\mathcal{I}, \mathcal{L})$ from $\mathcal{D}$, and samples patches $(\mathcal{P}_{\text{in}}, \mathcal{P}_{\text{gt}})$, respectively. Training for the sequence of a shot converts $\bar{\Theta}_0$ into $\bar{\Theta}_K$, where $K > k$, and K is the number of iterations for a single sequence. $\bar{\Theta}_K$ is used as $\bar{\Theta}_0$ in training for the next shot sequence continuously.

Our learning strategy improves colorization accuracy and shortens the processing time when users continue to use our colorization system. We demonstrate the effectiveness of our learning strategy in Section 6.4.4.

5. Colorization

Given line drawings to be colorized and few pre- and post-colorized line drawings in reference frames from data pool $\mathcal{D}$, the proposed colorization system is performed using the following procedure with the anime-specific patch-based learning.

1. Regarding preprocessing, the system extracts colors from the given post-colorized line drawings $\mathcal{O}^R$ and assigns an ID to each extracted color to generate the color palette C for the target sequence. The system also replaces the colors in the post-colorized line drawings $\mathcal{O}^R$ with their IDs according to color palette C to generate label maps $\mathcal{L}^R$. Then, the system trains a colorization model G using only a few line drawings $\mathcal{I}^R$

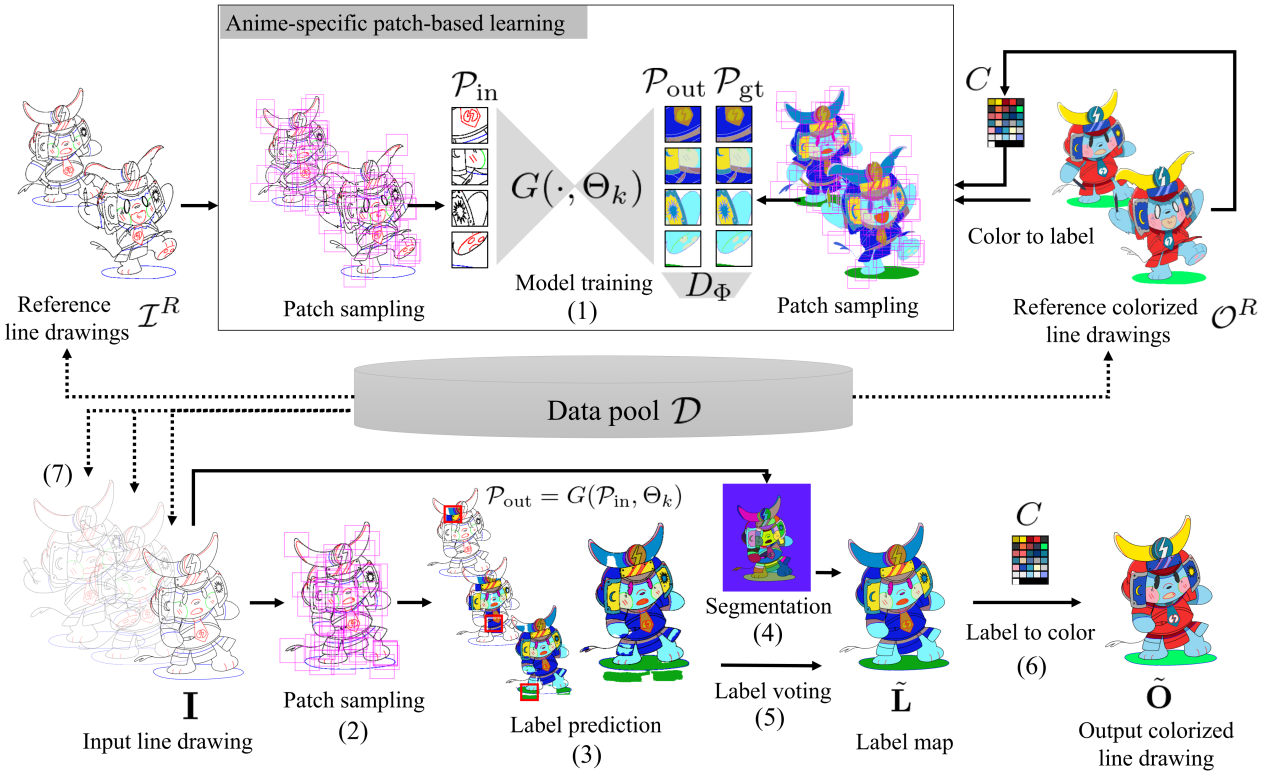


Figure 5: Colorization procedures. Given a few pre- and post-colorized line drawings from reference frames in the target sequence, the colorization model is trained using the proposed anime-specific patch-based learning. The line drawings of the remaining frames are then colorized frame-by-frame using the colorization model. Note that all processes run on the sequence to be colorized. Images originate from *Deadline* © OLM Asia SDN BHD.

and corresponding label maps $\mathcal{L}^R$ using the proposed continuous learning strategy. After training, the system saves the resulting model parameters $\bar{\Theta}_k$ on the disk. $\bar{\Theta}_k$ is used for the training in the next colorization process.

2. The system extracts line drawing patches $\mathcal{P}_{\text{in}}$ from the input line drawing $\mathbf{I}$ using anime-specific patch sampling (see Fig. 5(2)).

3. The colorization model predicts the labels of the line drawing patches $\mathcal{P}_{\text{out}} = G(\mathcal{P}_{\text{in}}, \Theta_k)$, and the system generates a label map $\tilde{\mathbf{L}}$ by pasting label patches $\mathcal{P}_{\text{out}}$ to their original position of the image and selecting the most frequent ID from the IDs that originate from the multiple patches overlapping the pixel (see Fig. 5(3)).

4. The system computes closed regions in the line drawing $\mathbf{I}$ by the leak-proof segmentation method using trapped-ball filling [6] (see Fig. 5(4)).

5. The system replaces the IDs of all pixels with the most frequent ID for each closed region in the label map $\tilde{\mathbf{L}}$ (see Fig. 5(5)).

6. The system replaces the IDs in the resulting label map $\tilde{\mathbf{L}}$ with colors according to color palette C (see Fig. 5(6)) to complete the output colorized line drawing $\tilde{\mathbf{O}}$.

7. The system repeats steps (2)–(6) for the remaining line drawings in the target sequence.

6. Experiments

To demonstrate the effectiveness of our method, we conducted an ablation study and quantitative evaluation. First, we present the datasets and metrics used in the experiments. Then, we describe the tuning we conducted to determine the optimal patch size that balances accuracy and processing speed. Finally, we describe the details of the ablation study and quantitative evaluation. We used the Adam optimizer for model training with a batch size of 16 and learning rate of 0.0004. We ran all experiments under Ubuntu 20.04 LTS, on an Intel Xeon W-2133 3.6 GHz CPU (with six cores), and a single NVIDIA Quadro RTX 8000 48GB GPU.

6.1. Datasets

Publicly available datasets for evaluating colorization tasks either contain only single pre- and post-colorized sketch pair for each character, not sequences (they are oriented towards colorization of sketch illustration rather than animation) or they are created from videos after the composition process (Fig. 1); the background has already been composited, as in LAVC and AnimeRun [21]. Data from these datasets does not represent a real situation in the colorization process. To the best of our knowledge, there is no large-scale dataset which satisfies our requirements, due to copyright issues. Like previous researchers [2, 3, 20], we constructed two original datasets for evaluation.

Dataset-A consisted of 40 million pre- and post-colorized line drawing patches with $M = 64$ sampled from a TV anime series that had already been broadcast. We used this dataset to simulate users continuously using our colorization system. Its details are described in Section 6.4.4.

Dataset-B consisted of pre- and post-colorized line drawing sequences of 22 shots from another TV anime series comprising on-air and unreleased in-house short movies. The average length of all shots was 11 frames. We used an average of two frames as the references for each shot. Although *Dataset-B* had full HD resolution, because of hardware restrictions, we shrank it so that either the height or width of the region of interest in the line drawing was eight times larger than patch size M. As a result, small closed regions may have disappeared. We discuss how to address this issue in Section 7 We used this dataset for all experiments in Section 6.

6.2. Metrics

Traditional anime-style colorizations have been performed manually by filling blank regions of line drawings with color. If the result of automatic colorization has some errors, the artist should replace the wrong color with the correct color for each region with an error. Thus, it would be natural for the evaluation of automatic colorization to be conducted using a region-level metric.

In this context, region-wise accuracy $\mathrm{Acc}_{\mathrm{region}}$, and mean Intersection-over-Union (mIoU) between automatic and manually colorized images (as the ground truth) are possible candidates for evaluation metrics. $\mathrm{Acc}_{\mathrm{region}}$ is more appropriate because it considers large and small regions equally and evaluates the correctness of the label of the region over all regions in the sequence; however, this makes evaluation slow. By contrast, mIoU can be evaluated faster than $\mathrm{Acc}_{\mathrm{region}}$, but it does not provide a region-wise evaluation in the case of multiple regions that have the same label. Therefore in this study, as evaluation metrics, we used mIoU for ablation studies of our method, and both metrics for comparison with other methods. Note that we averaged IoU over the label classes for each frame, and then averaged the per-frame mIoU over all frames. Additionally, we measured IoU at region level.

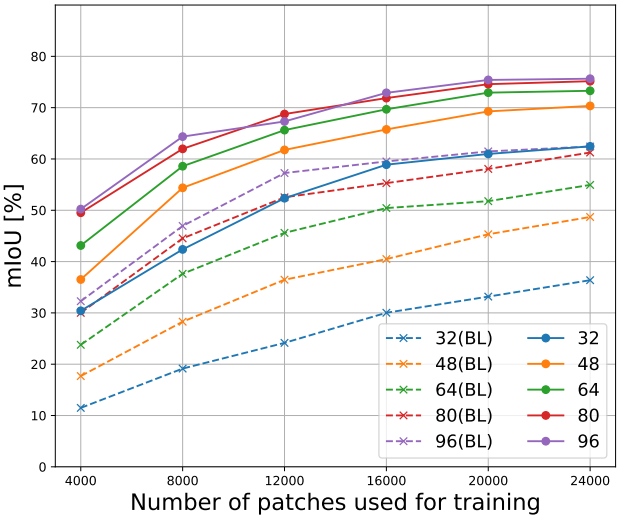


Figure 6: mIoU scores for *Dataset-B* using a colorization model trained using patches of different sizes. Note that, to focus on the impact of patch size variation, all curves were derived using our method without PE. The dashed line represents using a random sampling method as the *Baseline* (BL). Solid lines represent using the anime-specific sampling method for sampling.

6.3. Patch size selection

The patch size affects both the accuracy and speed of colorization, in a trade-off relationship. Therefore, we assessed the relationship to find the optimal patch size using *Dataset-B*.

In Fig. 6, the vertical and horizontal axes represent mIoU for *Dataset-B* and the number of patches used for colorization model training, respectively. Note that, to focus on the influence of patch-size variation, we did not use PE. We also show the processing time for each colorization procedure in Table 1. Note that we measured the 'model training' time when the number of patches reached 16,000 because mIoU seemed to converge at this point. These results indicate that $M = 64$ is a reasonable patch size providing a good balance between accuracy and processing time.

6.4. Ablation study

6.4.1 Anime-specific patch sampling

To confirm the contribution of anime-specific patch sampling, we observed the mIoU of the results colorized by the model trained with patches chosen by random sampling (*Baseline*:BL) and anime-specific sampling (unmarked) for *Dataset-B*. Fig. 6 illustrates the mIoU of the results for the

Table 1: Processing time (s) for each colorization procedure. The 'model training' time was measured when the number of patches reached 16,000. The processing times for (2), (4), (5), and (6) were almost identical as they are independent of patch size. '—' indicates the value was the same as for patch size = 64.

Process / Patch size	32	48	64	80	96
(1) Training	58	60	64	72	83
(1) Training(+PE)	59	63	67	75	85
(2) Patch sampling	—	—	0.02	—	—
(3) Prediction	—	—	5.48	—	—
(3) Prediction(+PE)	—	—	6.00	—	—
(4) Segmentation	—	—	3.80	—	—
(5) Label voting	—	—	0.02	—	—
(6) Label to color	—	—	0.23	—	—

patch sizes $M = 32, 48, 64, 80, 96$. The solid and dashed lines represent anime-specific and random sampling, respectively; they clearly show that the proposed anime-specific sampling method improved mIoU by prioritizing patches with color variations in the colorized line drawings.

6.4.2 Effect of Position embedding (PE)

To confirm the effect of PE, we observed the mIoU of the colorization results using our method with different patch sizes $M = 32, 48, 64$ with and without PE (indicated as *Baseline* and *With PE*, respectively) for *Dataset-B*. The results are shown in Fig. 7. The vertical axis represents mIoU and the horizontal axis represents the number of patches used for model training. Dashed lines represent the cases without PE and solid lines represent the cases with PE. When the patches were small ($M = 32$), PE improved mIoU because smaller patches are likely to lead to one-to-many mappings, as discussed in Section 4.2. PE resolved such ambiguous mapping of the patches. When larger patches ($M \geq 48$) were used, PE also improved mIoU but less significantly because large patches reduce pattern similarity. As a result, patches are likely to be unique without PE. Fig. 12 later shows an example. In the shots that contained two characters with similar local appearances, one-to-many mapping often occurred and PE seemed to improve mIoU. The first column of Fig. 12 shows that the color of self-shadows on the inner thigh of characters switched when the *Baseline* method was used. The colors of the head and the left-hand side of the character on the right are not correct either. These results exemplify failures in training caused by ambiguous mapping. Using *With PE* seemed to resolve that ambiguity, as shown in the bottom rows of Fig. 12. As Table 1 shows, PE requires approximately 3 s more processing time in model training and label prediction. However, it helps to improve colorization accuracy in such cases.

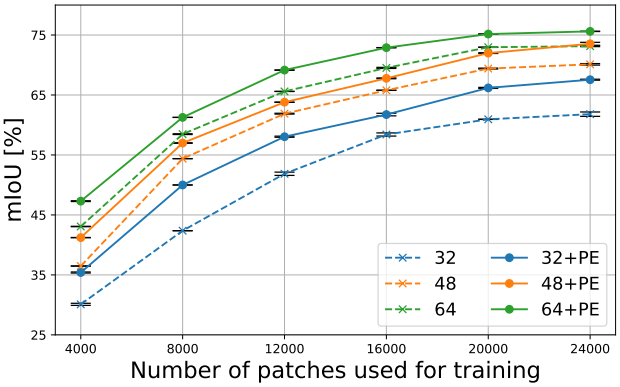


Figure 7: mIoU for *Dataset-B* using the colorization model trained on patches with and without PE on *Dataset-B*. Dashed lines represent no PE and solid lines represent PE. Black error bars represent the standard deviation over five repetitions of the evaluation.

6.4.3 Comparison to image-to-image translation

We addressed the colorization problem as a semantic segmentation task. An alternative approach directly predicts the color of input line drawing patches as per image-to-image translation methods, then the color for each pixel is mapped to the most similar color in the palette. To confirm the validness of our approach, we also compared the mIoU of the colorization results of this method to our method. Note that, the voting process is applied to the resulting image to achieve anime-style colorization, as in our method. This alternative approach is referred as "i2i". In this study, neither method used PE.

Fig. 8 shows the mIoU of the results for patch sizes of $M = 32, 48, 64, 80, 96$. The dashed and solid lines represent i2i and our method, respectively. It clearly shows that our method is more accurate; i2i produces two types of error: prediction error and mapping error to the palette's colors due to color gradations in the prediction result. Instead, our method only produces prediction errors as shown in Fig. 9.

6.4.4 Learning strategy

To confirm the effectiveness of our continuous learning strategy, we evaluated colorization accuracy under the condition that the parameters $\bar{\Theta}$ of the colorization model were updated multiple times using previous colorization processes before the target colorization. Specifically, we first trained colorization models according to our learning strategy using different sizes of patch gallery consisting of 10,000 to 40 million patches from *Dataset-A*. This produced a set of colorization models with $\{\bar{\Theta}_k\}_{625}^{25000}$ with a batch size of 16. Then, we ran the colorization process based

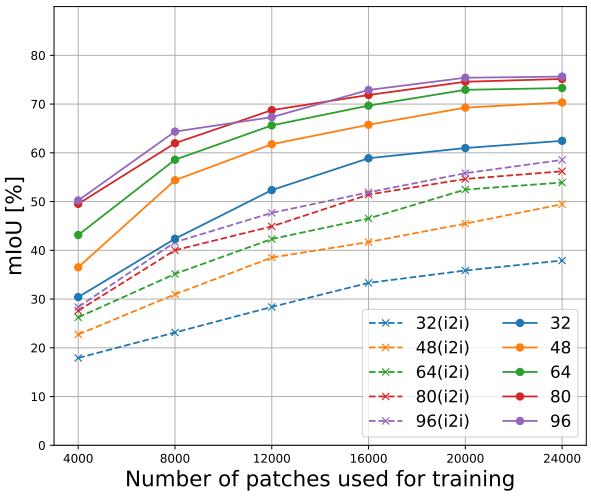


Figure 8: mIoU for *Dataset-B* using i2i and our method. Dashed lines represent i2i and solid lines represent our method.

on the above model parameters $\bar{\Theta}_k$ for each sequence from Dataset-B. Finally, we computed the average of mIoU for each colorization result over Dataset-B and each $\bar{\Theta}_k$. The result is shown in Fig. 10.

The horizontal axis represents the number of patches processed for the sequence of a shot and the vertical axis represents the mIoU of the colorization. The blue dashed line represents the performance of the baseline model that was trained from scratch (random initialization of the model parameters). The other solid lines represent the performance of the models previously updated using 10,000 to 40 million patches. The labels show the number of patches divided by 10,000 in Fig. 10. The result implies that a gallery of over 0.1 million patches was sufficient for the updates.

Interestingly, increasing the number of patches used for pre-training with Dataset-A helped to increase the update speed for Dataset-B, even though Dataset-A and Dataset-B consisted of sequences from different anime series. A possible reason is that Dataset-A was similar to Dataset-B at the patch level. The model learned to be a generalized feature extractor for line drawing patches; that is, knowledge in Dataset-A was transferred to the update step for Dataset-B. As a result, our learning strategy enabled us to reduce the time to reach production-ready accuracy. We assume a new colorization procedure that artists create some reference frames by hand, provide our colorization system with the reference frames and the target sequence to be colorized, and fix the resulting sequence to complete. Our learning strategy updated the model for every colorization process and the resulting model provided progressively better performance. Because our patch-based learning extracted many patches, even from a single reference, the collection of over 0.1 million patches remained practical.

6.5. Quantitative evaluation

To demonstrate the effectiveness of our method, we colorized the line drawings from all shots in *Dataset-B* using U-net, SGA, LAVC, and AnT, and then compared colorization accuracy for all methods.

To compare our method to U-net, we trained a U-net using pre- and post- colorized line drawings from only the reference frames for each sequence in *Dataset-B*. We used the Adam optimizer with one size of mini-batch and retained the colorization model with the highest validation mIoU during 20,000 epochs.

SGA supports only one reference image for each colorization target. Therefore, we colorized line drawings in a sequence using each reference line drawing in the sequence, then picked the most accurate result frame-by-frame for evaluation. LAVC requires two reference images sandwiching line drawings to be colorized. We used the same reference image as the second reference if we have only a single reference image in a sequence. In this paper, we have assumed that only a few images are available for model training. Under this assumption, it was difficult to train both models from scratch. An alternative option is to use their pre-trained models for the evaluation. However, there is a gap in drawing styles between the dataset they use for pre-training and *Dataset-B*; our line drawings consist of only foreground objects including characters and/or props. Moreover, they came from before the composition process (see Fig. 1), in other words, there is no background. To make the comparison as fair as possible, we attempted to fill those gaps by fine-tuning both pre-trained models using $\mathcal{I}^R$ and $\mathcal{O}^R$ from *Dataset-B* as reference sketches or line art images. Note that, we limited the time for the fine-tuning to be approximately the same time required to train our model. Specifically, as for SGA, we fine-tuned the pre-trained model using 1 to 5 reference images for each sequence in the Dataset-B with 400 epochs. As for LAVC, we fine-tuned the pre-trained model using sequential triplets extracted from reference images in *Dataset-B* with 250 iterations. We set the batch size to 1 in both cases. When reporting results, we add a suffix 'w/ fine-tuning' to a model after fine-tuning. To achieve anime-style colorization, a color for each pixel was mapped to the most similar color in the palette, then the voting process for each closed region was performed as in our method.

Our method had full components with a patch size of $M = 64$ and we trained the model from scratch (Ours from scratch) or previously updated it using 40 million patches (Ours). The baseline method (Baseline) used random sampling instead of the anime-specific sampling in our method. The baseline did not use PE or continuous learning.

The colors of the indicator lines in the line drawings may

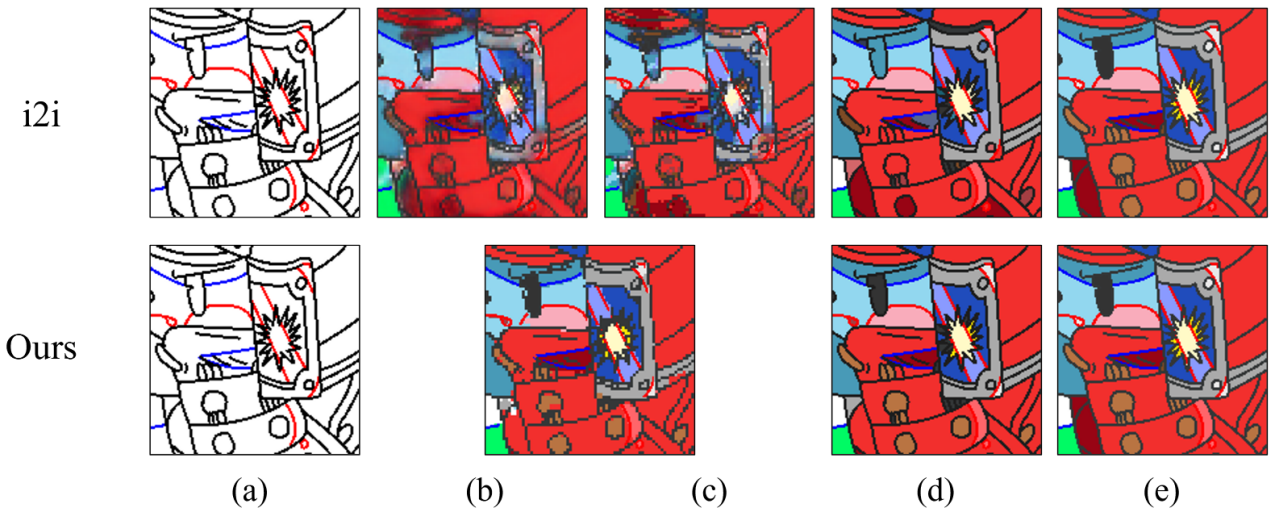


Figure 9: An example of incorrect color mapping to the palette's colors: (a) input line drawing patch, (b) resulting images after prediction, (c) after color mapping, (d) after voting process, and (e) corresponding manual coloring result (ground truth). Above: colorization by i2i, below: colorization by our method. Images originate from *Deadline* © OLM Asia SDN BHD.

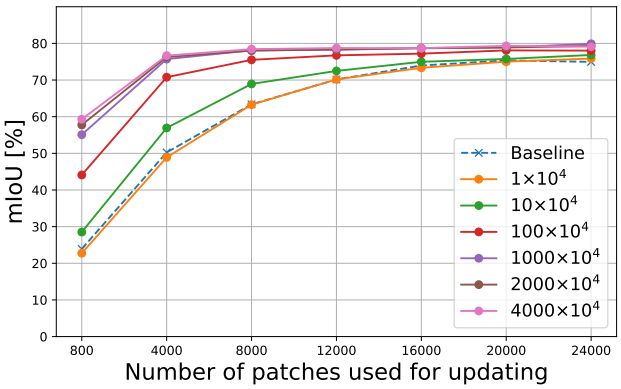


Figure 10: Colorization accuracy for various models continuously updated using our learning strategy and the *Baseline* trained using random initialization. Note that each model was updated using $1 \times 10^4 - 4,000 \times 10^4$ (labeled $1-4000$) patches of pre- and post-colorized line drawing from each shot in *Dataset-B*. The horizontal axis represents the number of patches used for updating.

be a strong guide to the network and will lead to better performance. To confirm the improvements caused by colored indicator lines, we also evaluated variants of both AnT and Ours; the inputs of these methods were the line drawings consisting of black lines only that we generated by replacing colored indicator lines with black. We refer to these results as AnT (Mono) and Ours (Mono). Then we computed the average and standard deviation of region-wise accuracy and mIoU for all methods. The results are shown in Table 2. Additionally, some representative colorization results are shown in Figs. 13–16 with their statistics.

As shown in Table 2 and Figs. 13–16, regardless of fine-tuning, SGA and LAVC were less accurate than our methods. These results indicate that it is difficult for SGA and LAVC to adapt to various styles of line drawings of characters and objects given limited data and time. Our method achieved state-of-the art performance with respect to region-wise accuracy and mIoU while requiring only one to five colorized references and hence, not requiring a large-scale dataset, tedious annotation, and pseudo-data generation for model training.

7. Discussion

We now discuss whether our method demonstrates acceptable performance for professional colorization artists. From the professional artist's point of view, according to the literature [13], borderline acceptable mIoU (which provides a good starting point for manual colorization and reduces manual labor time) and overall processing time are 65% and 200 s, respectively. Our proposed method achieves a higher mIoU (80.18%) in a shorter time (approximately 90 s for model training and 15 s for per-frame automatic painting), exceeding these targets.

Our learning strategy is efficient for the following reasons.

- It updates our model in the background after running every colorization process. Because knowledge is distilled into the network, storing raw data is not required for performance improvements.

Table 2: Region-wise accuracy and mIoU comparison. The numbers in parentheses are the standard deviation of these values. Notes: † Trained using only reference frames. * Returned exactly the same results when inputting line drawings with colored indicators.

Method	$\mathrm{Acc}_{\mathrm{region}}$[%]	mIoU [%]
U-net †	21.09 (±4.43)	27.02 (±5.16)
SGA	9.47 (±3.47)	9.16 (±1.97)
SGA w/ fine-tuning	32.24 (±5.42)	33.10 (±5.26)
LAVC	6.32 (±2.36)	19.36 (±3.77)
LAVC w/ fine-tuning	23.51 (±5.44)	27.07 (±6.41)
AnT	62.27 (±7.95)	71.80 (±10.79)
Baseline	52.41 (±6.07)	56.86 (±8.39)
Ours from scratch	68.24 (±6.89)	75.12 (±7.83)
Ours	70.93 (±6.75)	80.18 (±8.59)
AnT (Mono) *	62.27 (±7.95)	71.80 (±10.79)
Ours (Mono)	66.55 (±8.21)	75.95 (±10.20)

- It exploits knowledge from the latest colorization process: users may use the colorization process in the same anime work over a period of time.

Ideally, a dataset of the same artwork should be used to validate the continuous learning strategy (Section 6.4.4). However, the results of our experiment suggest that even models trained with images of different styles can contribute to improving accuracy. It may be possible that the patch-based approach makes the style irrelevant and then enables the procedural generation of patches for model training. This would result in some IP-free data and would improve colorization accuracy from a potentially unlimited amount of generated data.

Nevertheless, our method has the following limitations. Because of hardware restrictions, we shrank the input line drawings (Fig. 11(a)). This may have caused small closed regions to disappear (Fig. 11(b)), thereby resulting in these regions not being colorized. Ishii *et al*. indicated the difficultly for artists to find and correct errors arising if the colorization model outputs incorrect labels for small regions (Fig. 11(c)). They addressed this issue with a strategy that ignored the result of automatic colorization for such small regions and regions with low prediction confidence. In a similar manner, we can address this issue by removing such small regions from the colorization targets using the area of a region as a threshold.

Training did not work well with simple line drawings with few details such as Fig. 11(d). This type of line drawing can be painted faster by hand than using our system. Additionally, it is not easy for users to know how many and which reference frames are required to obtain the best results. Empirically, we first recommend selecting frames that have many details. Then, we suggest painting secondary

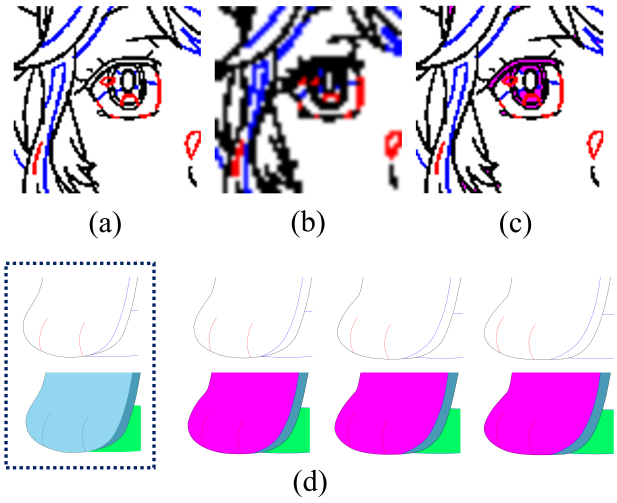


Figure 11: Limitations of our method. Small closed regions (in (a)) are disappear during the shrinking operation satisfy the hardware restriction. (b): after shrinking. Such regions are indicated by painting a noticeable color (e.g. magenta) to indicate artists need to fix them (c). Colorization results for a simple line drawing sequence with error (colored magenta) (d). Reference frames are surrounded by dashed blue boxes. Manual colorization must be much faster than using our method together. Images on the top row originate from *Restaurant to Another World 2*, © Junpei Inuzuka, IMAGICA INFOS/Restaurant to Another World 2 Project.

frames with the highest variation in line topology.

In this paper, we adopted an U-net as a network architecture for proof of concept. Despite using a simple network, our colorization method achieved state-of-the-art performance on the dataset from real anime productions. The performance might be improved by introducing the latest network architectures. We plan to attempt this in future.

8. Conclusions

We have proposed a novel anime-style colorization method based on few-shot patch-based learning. Our continuous learning strategy reduces training time while achieving the desired accuracy. Through a quantitative evaluation, we have demonstrated that our colorization method achieves state-of-the-art performance while being more practical and pipeline-friendly than existing methods.

In future, we hope to make the whole process more interactive. We aim to make it possible for users to suggest reference frames to be painted and to give them immediate feedback about the colorization results of the sequence. This would require achieving on-the-fly iterative training. We are also considering introducing temporal coherency, and, for further efficiency in the colorization process, an automatic

open junction snapping method for line drawings [25].

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. We are grateful to Zekun Li and Prof. Fang Lue Zhang for providing their code and data for comparison. We give thanks to Mohammad Shafiq Bin Md Shawal, Muhammad Mohamad Din Yati, Yap Fei, Raihanah Ayuna Faiz, Kiyoumi Agemura, and Shogo Sakurazawa for testing our colorization system and for providing valuable feedback from the production side. We thanks Ken Anjyo, Marc Salvati, and Alexandre Derouet-Jourdan for reviewing the paper and providing feedback. Finally, we would like to give our thanks and appreciation to Junpei Inuzuka and IMAGICA INFOS for their permission for us to use images from *Restaurant to Another World 2* for research purposes.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28(3), Aug. 2009. 3

[2] E. Casey, V. Pérez, and Z. Li. The animation transformer: Visual correspondence via segment matching. In *Proc. International Conference on Computer Vision (ICCV)*, pages 11323–11332. IEEE, 2021. 1, 3, 8

[3] S.-Y. Chen, J.-Q. Zhang, L. Gao, Y. He, S. Xia, M. Shi, and F.-L. Zhang. Active colorization for cartoon line drawings. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 28(2):1198–1208, 2022. 3, 8

[4] T. D. Q. Dang, T. Do, A. Nguyen, V. Pham, Q. Nguyen, B. Hoang, and G. Nguyen. Correspondence neural network for line art colorization. In *Proc. ACM SIGGRAPH Posters*, SIGGRAPH '20, New York, NY, USA, 2020. Association for Computing Machinery. 3

[5] C. Furusawa, K. Hiroshiba, K. Ogaki, and Y. Odagiri. Comicolorization: Semi-automatic Manga Colorization. In *ACM SIGGRAPH Asia Technical Briefs*, SA '17, New York, NY, USA, 2017. Association for Computing Machinery. 5

[6] P. Hensman and K. Aizawa. cGAN-based Manga Colorization Using a Single Training Image. In *Proc. ICDAR*, volume 3, pages 72–77, Los Alamitos, CA, USA, 2017. IEEE Computer Society. 4, 7

[7] D. Ishii, H. Kubo, S. Shinagawa, A. Maejima, T. Funatomi, S. Nakamura, and Y. Mukaigawa. Confidence-aware practical anime-style colorization. In *ACM SIGGRAPH Talks*, SIGGRAPH '20, New York, NY, USA, 2020. Association for Computing Machinery. 1, 4

[8] Y. Kanamori. Region matching with proxy ellipses for coloring hand-drawn animations. In *ACM SIGGRAPH Asia 2012 Technical Briefs*, SA '12, New York, NY, USA, 2012. Association for Computing Machinery. 1, 3

[9] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics (ToG)*, 23(3):689–694, aug 2004. 5

[10] Z. Li, Z. Geng, Z. Kang, W. Chen, and Y. Yang. Eliminating gradient conflict in reference-based line-art colorization. In *European Conference on Computer Vision*, pages 579–596. Springer, 2022. 4

[11] S. Liu, X. Wang, X. Liu, Z. Wu, and H. S. Seah. Shape correspondence for cel animation based on a shape association graph and spectral matching. *Computational Visual Media*, pages 1–24, 2023. 1, 3

[12] A. Maejima, H. Kubo, T. Funatomi, T. Yotsukura, S. Nakamura, and Y. Mukaigawa. Graph matching based anime colorization with multiple references. In *Proc. ACM SIGGRAPH Posters*, SIGGRAPH '19, New York, NY, USA, 2019. Association for Computing Machinery. 1, 3

[13] A. Maejima, H. Kubo, S. Shinagawa, T. Funatomi, T. Yotsukura, S. Nakamura, and Y. Mukaigawa. Anime character colorization using few-shot learning. In *ACM SIGGRAPH Asia Technical Communications*, SA '21 Technical Communications, New York, NY, USA, 2021. Association for Computing Machinery. 11

[14] A. Orzan, A. Bousseau, P. Barla, H. Winnemöller, J. Thollot, and D. Salesin. Diffusion curves: A vector representation for smooth-shaded images. *Commun. ACM*, 56(7):101–108, jul 2013. 5

[15] Y. Qu, T.-T. Wong, and P.-A. Heng. Manga colorization. In *ACM SIGGRAPH Papers*, SIGGRAPH '06, page 1214–1220, New York, NY, USA, 2006. Association for Computing Machinery. 5

[16] S. Ramassamy, H. Kubo, T. Funatomi, D. Ishii, A. Maejima, S. Nakamura, and Y. Mukaigawa. Pre-and post-processes for automatic colorization using a fully convolutional network. In *ACM SIGGRAPH Asia Posters*, pages 1–2. Association for Computing Machinery, New York, NY, USA, 2018. 1, 4

[17] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6836–6845. IEEE, 2017. 5

[18] K. Sato, Y. Matsui, T. Yamasaki, and K. Aizawa. Reference-based manga colorization by graph correspondence using quadratic programming. In *ACM SIGGRAPH Asia 2014 Technical Briefs*, number Article 15 in SA '14, pages 1–4, New York, NY, USA, Nov. 2014. Association for Computing Machinery. 1, 3

[19] J. Shi and Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600. IEEE, 1994. 5

[20] M. Shi, J.-Q. Zhang, S.-Y. Chen, L. Gao, Y.-K. Lai, and F.-L. Zhang. Reference-based deep line art video colorization. *IEEE Transactions on Visualization and Computer Graphics*, 29(6):2965–2979, 2023. 4, 8

[21] L. Siyao, Y. Li, B. Li, C. Dong, Z. Liu, and C. C. Loy. Animerun: 2d animation visual correspondence from open source 3d movies. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 8

[22] D. Sýkora, J. Buriánek, and J. Žára. Unsupervised colorization of black-and-white cartoons. In *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation*

and Rendering, NPAR '04, page 121–127, New York, NY, USA, 2004. Association for Computing Machinery. 3

[23] D. Sýkora, J. Dingliana, and S. Collins. Lazybrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum*, 28(2):599–608, 2009. 5

[24] O. Texler, D. Futschik, M. Kučera, O. Jamriška, Šárka Sochorová, M. Chai, S. Tulyakov, and D. Sýkora. Interactive video stylization using few-shot patch-based training. *ACM Transactions on Graphics (ToG)*, 39(4):73, 2020. 2, 4

[25] J. Yin, C. Liu, R. Lin, N. Vining, H. Rhodin, and A. Sheffer. Detecting viewer-perceived intended vector sketch connectivity. *ACM Trans. Graph.*, 41(4), jul 2022. 13

[26] L. Zhang, C. Li, E. Simo-Serra, Y. Ji, T.-T. Wong, and C. Liu. User-guided line art flat filling with split filling mechanism. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9889–9898. IEEE, 2021. 5

[27] L. Zhang, C. Li, T.-T. Wong, Y. Ji, and C. Liu. Two-stage Sketch Colorization. *Proc. ACM SIGGRAPH Asia*, 37(6):261:1–261:14, November 2018. 5

[28] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time User-guided Image Colorization with Learned Deep Priors. *ACM Transactions on Graphics (ToG)*, 36(4), July 2017. 5

[29] H. Zhu, X. Liu, T.-T. Wong, and P.-A. Heng. Globally optimal toon tracking. *ACM Transactions on Graphics (ToG)*, 35(4):75:1–75:10, July 2016. 1, 3, 4

Appendix

A. Traditional anime production pipeline

From the storyboard of a new anime work, production follows a pipeline that consists of layout, keyframing, animation, colorization, and composition, as shown in Fig. 1.

1. Rough sketches are drawn by artists for each sequence to determine the pose and layout of objects following the storyboard.

2. Artists clean up rough sketches and add details to create keyframes. Then the keyframes are traced to extract contour lines. In this process, contour lines that are in contact with regions in shadow, highlighted, or in other colors are commonly colored in blue, red, or green. These colored lines are essential for indicating to the artists in charge of colorization which color to use in the above regions of the frame.

3. Artists insert intermediate frames in between the keyframes to achieve smooth motion transitions in the animation.

4. Artists colorize line drawings in the animation. Each region surrounded by contour lines must be filled with a single color (Fig. 2(c)) taken from a color palette designed by the color director; we refer to this style of colorization as *anime-style colorization*. For efficiency, the paint-bucket tools implemented in commercial software such as TV-Paint (https://www.tvpaint.com/v2/wp/?page_id=1224&lang=en) and Clip Studio Paint (https://www.clip-studio.com/clip_site/clipstudiopaint/scenes/animation) are often used to fill closed regions. However, they require airtight regions; hence, unintended gaps in the contours must be corrected in advance.

5. Final animation is composed of the resulting colored animation, backgrounds, and sometimes rendering images by CG.

Traditionally, each process is specialized: the same person is not necessarily in charge of all processes.

In some studios, processes from layout to animation are performed on paper, and the colorization process is performed on a computer using digital scans of the sheets of paper in raster image format. Recently, some studios have adopted a fully digital workflow using line drawings in vector image format. As an intermediate stage, some processes are performed with vector line drawings and then converted to raster image format for the subsequent composition process. To support both traditional hybrid (analog and digital) and a fully digital workflow, we focus on line drawings in raster image format in this study.

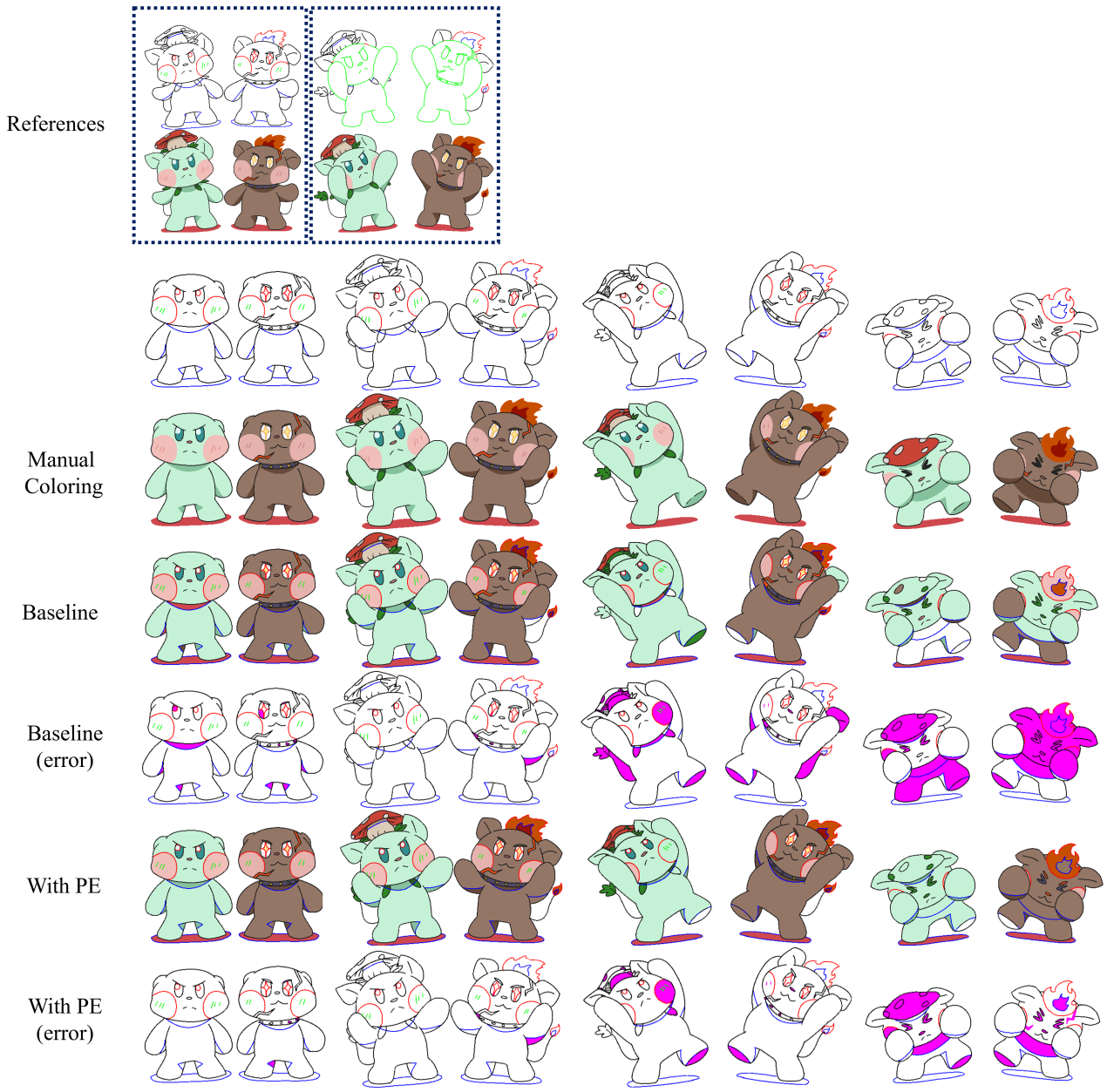


Figure 12: These snapshots illustrate that our PE technique contributes to improving colorization accuracy. References: pre- and post-colorized line drawings used for model training. First row: input line drawings. Second row (Manual Coloring): target line drawings and their colorization by an artist. Third and fifth rows: automatic colorization results without PE (Baseline) and with PE (With PE) for the target line drawing. Fourth and sixth rows: corresponding error visualization in which incorrect colorization compared to manual results is indicated in magenta. Images originate from *OLMA Wonderland* © OLM Asia SDN BHD.

Figure 13: Region-wise accuracy and mIoU comparison of the method proposed by U-net, SGA, LAVC, AnT (Cadmium), the baseline method (Baseline), and our method with a continuous learning strategy (Ours) and with monochrome line drawings as inputs (Ours (Mono)) on the line drawing sequences. Magenta pixels indicate the incorrect predictions compared to manual work. Note that the reference frames surrounded by dashed blue boxes were not counted in the evaluation. Images originate from *Restaurant to Another World 2*, © Junpei Inuzuka, IMAGICA INFOS/Restaurant to Another World 2 Project.

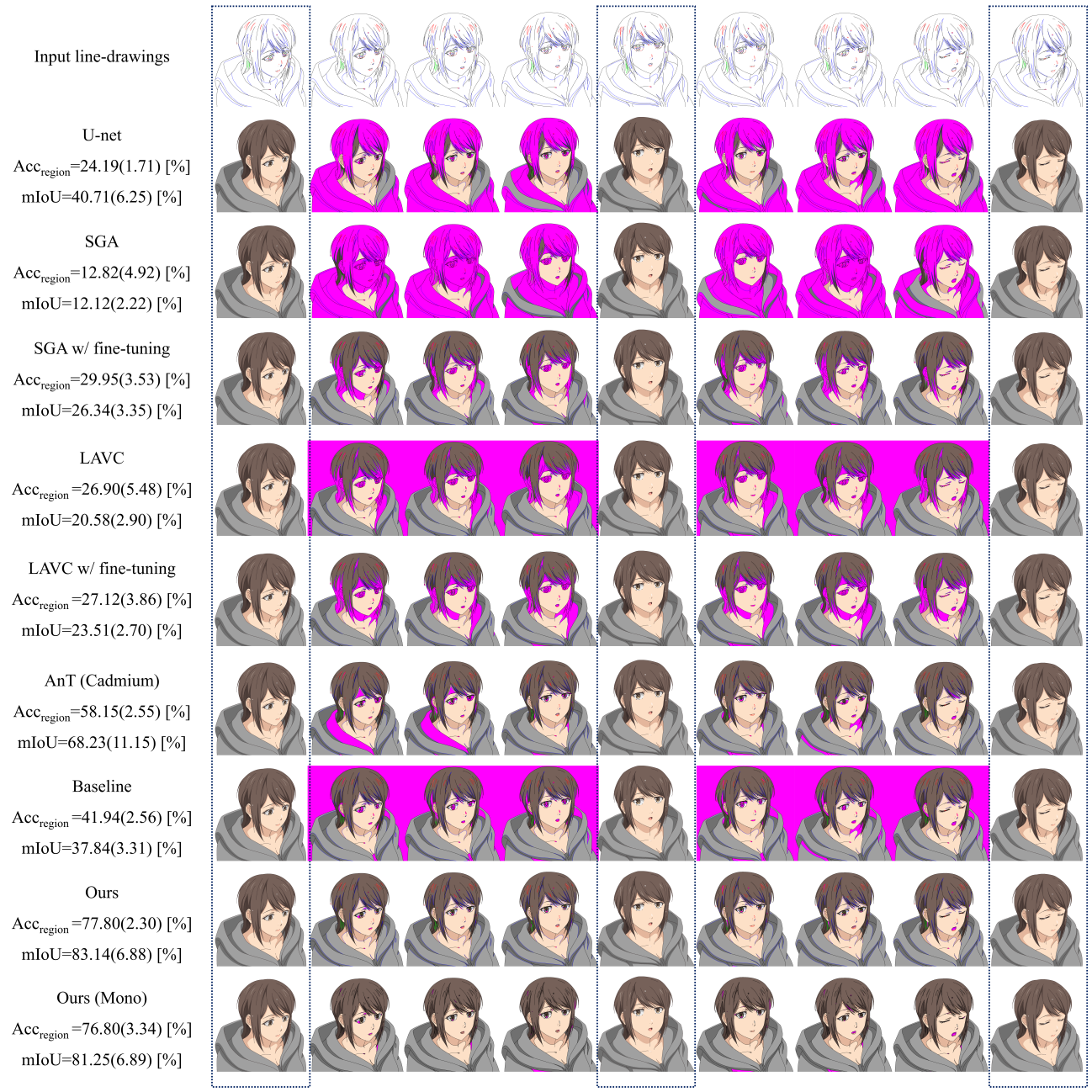


Figure 14: Region-wise accuracy and mIoU comparison of the method proposed by U-net, SGA, LAVC, AnT (Cadmium), the baseline method (Baseline), and our method with a continuous learning strategy (Ours) and with monochrome line drawings as inputs (Ours (Mono)) on the line drawing sequences. Magenta pixels indicate the incorrect predictions compared to manual work. Note that the reference frames surrounded by dashed blue boxes were not counted in the evaluation. Images originate from *Restaurant to Another World 2*, © Junpei Inuzuka, IMAGICA INFOS/Restaurant to Another World 2 Project.

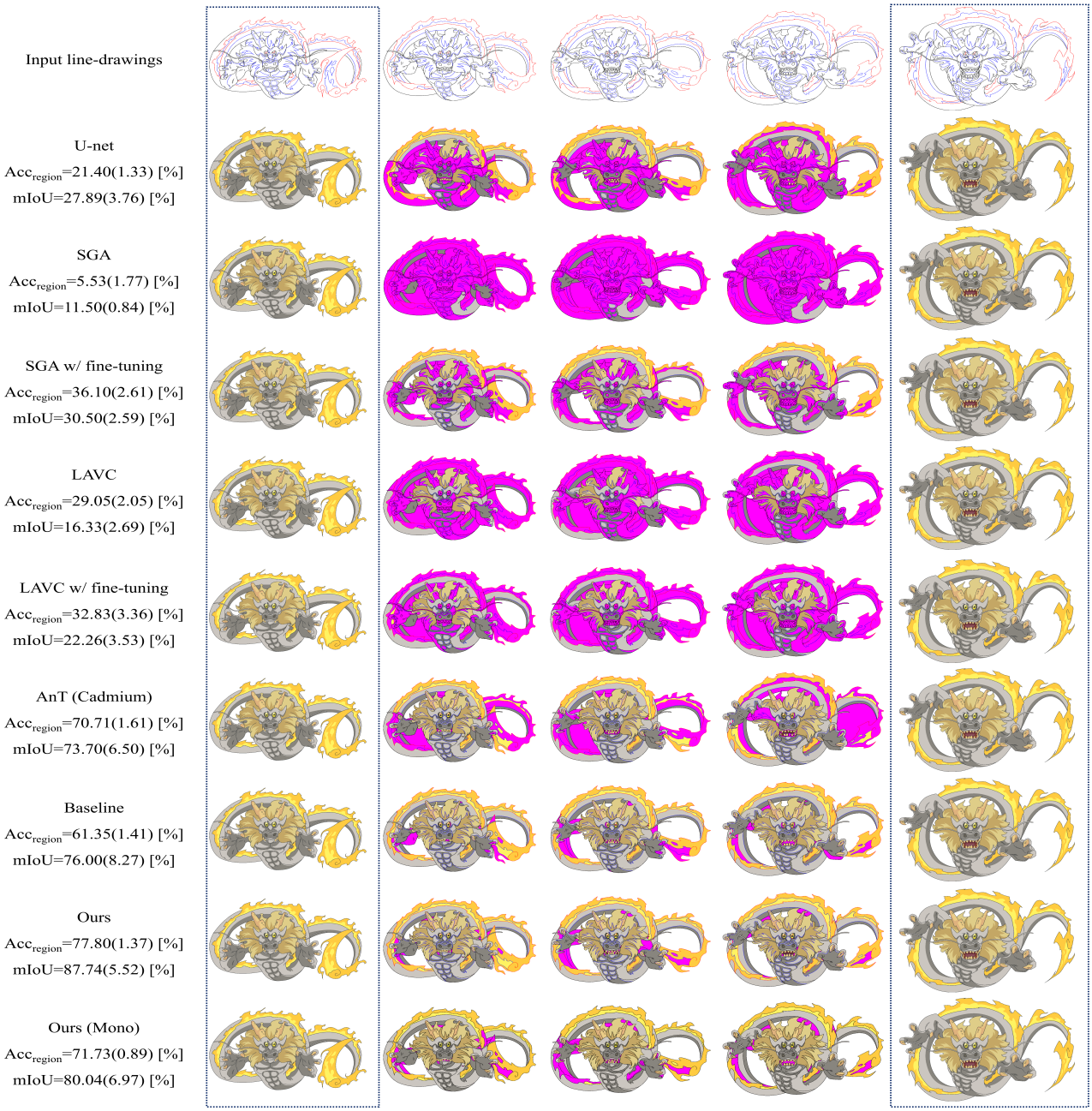


Figure 15: Region-wise accuracy and mIoU comparison of the method proposed by U-net, SGA, LAVC, AnT (Cadmium), the baseline method (Baseline), and our method with a continuous learning strategy (Ours) and with monochrome line drawings as inputs (Ours (Mono)) on the line drawing sequences. Magenta pixels indicate the incorrect predictions compared to manual work. Note that the reference frames surrounded by dashed blue boxes were not counted in the evaluation. Images originate from *Deadline* © OLM Asia SDN BHD.

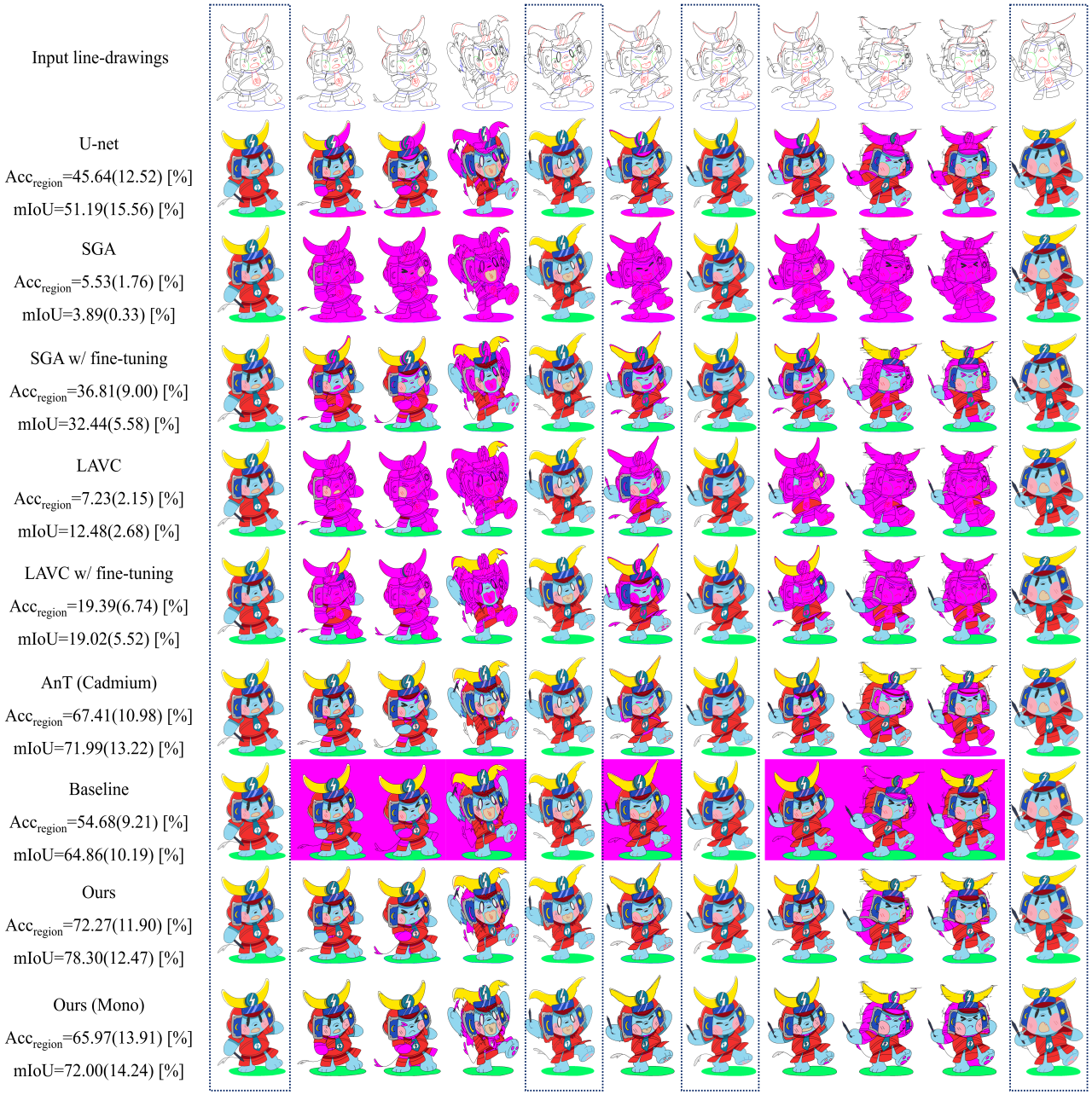


Figure 16: Region-wise accuracy and mIoU comparison of the method proposed by U-net, SGA, LAVC, AnT (Cadmium), the baseline method (Baseline), and our method with a continuous learning strategy (Ours) and with monochrome line drawings as inputs (Ours (Mono)) on the line drawing sequences. Magenta pixels indicate the incorrect predictions compared to manual work. Note that the reference frames surrounded by dashed blue boxes were not counted in the evaluation. Images originate from *Deadline* © OLM Asia SDN BHD.