SCIENTIFIC CULTURE

www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 2.1, (2026), pp. 858-873
Open Access. Online & Print

DOI: 10.5281/zenodo.122.12673

ADAPTIVE AI LEARNING FOR DATA-EFFICIENT ANIMATION HUE SYNTHESIS

Abeer Ibrahim[1*]

[1]Assistant Professor, Department of Graphic Design, Faculty of Architecture and Design, Al Zaytoonah University of Jordan, Amman, Jordan.

Received: 07/11/2025
Accepted: 22/11/2025

Corresponding Author: Abeer Ibrahim (a.ibrahem@zuj.edu))

ABSTRACT

This study addresses the challenge of anime-style colorization with limited reference data. Conventional approaches depend on either large-scale datasets or extensive manual effort, both of which are impractical for production workflows. To overcome these limitations, a few-shot, patch-based learning framework is introduced, specifically tailored for anime line art. The proposed method integrates anime-specific patch sampling, position embedding to reduce spatial ambiguity, and a continuous learning strategy that leverages newly produced artist-colored samples for incremental refinement. Methods: The approach employs a targeted patch sampling method based on corner detection to extract informative visual features from line drawings. Position embedding (PE) is applied to distinguish visually similar patches by spatial location, while a continuous learning mechanism using Exponential Moving Average (EMA) incrementally enhances model performance. Training and evaluation are conducted on two custom-built datasets derived from professional anime productions: Dataset-A (40 million patches) for simulating continuous training, and Dataset-B (22-shot sequences) for performance validation. Results: Experimental findings demonstrate that the proposed method significantly outperforms state-of-the-art baselines including U-Net, SGA, LAVC, and AnT in both region-wise accuracy and mean Intersection over Union (mIoU), achieving 80.18%—well above the 65% threshold typically required for professional production quality. The system requires minimal input (1–5 reference images), while maintaining efficient per-frame processing (~15 seconds). Conclusions: The study presents a scalable and adaptive framework for anime-style colorization that meets industry demands for automation, speed, and visual consistency. The findings position this method as a production-ready solution for professional animation pipelines.

KEYWORDS: Anime, Colorization, Few-Shot Learning, Position Embedding, Continuous Learning.

1. INTRODUCTION

Anime, a form of limited animation often characterized by hand-drawn or digitally rendered visuals, is known for its stylized characters and exaggerated expressions.

Despite advancements in technology, the colorization of anime line drawings remains a labor-intensive process, requiring artists to manually apply colors to each enclosed region within contour lines. This step, although essential for maintaining the artistic integrity of a production, significantly slows down workflow efficiency.

Traditionally, anime-style colorization has been approached as a region correspondence problem, employing methods such as graph matching or region tracking. More recently, colorization has been reframed as a semantic segmentation task using deep neural networks. However, both methodologies face substantial limitations.

Region-based techniques struggle with computational scalability when applied to complex line drawings, while deep learning models require large datasets for training—resources often constrained by intellectual property restrictions and the uniqueness of artistic styles across anime productions.(Kim,et al.2025)

This study addresses these challenges by proposing a few-shot, patch-based learning approach for anime-style colorization.

The method leverages a novel sampling strategy with position embedding to reduce spatial ambiguity in line art and introduces a continuous learning mechanism that adapts over time using artist-colored frames.

This approach enables the model to be trained from scratch or fine-tuned with only a small number of reference drawings, making it practical and easy to integrate into existing anime production pipelines

The primary contributions of this research include: (a) a patch-based colorization method tailored for anime line art that functions effectively with limited training data, (b) a continuous learning strategy that improves model performance with each use, and (c) evidence that colored indicator lines—common in anime workflows—can significantly enhance both manual and automated colorization processes.

Experimental results demonstrate that this method surpasses state-of-the-art techniques in accuracy and processing speed, offering a scalable solution for modern anime production environments.

In a conventional anime production pipeline (see Fig. 1), the process of colorizing draft line drawings is both labor-intensive and time-consuming. Colorization artists meticulously fill each enclosed region within the contour lines using a designated color from a palette curated by a color director (see

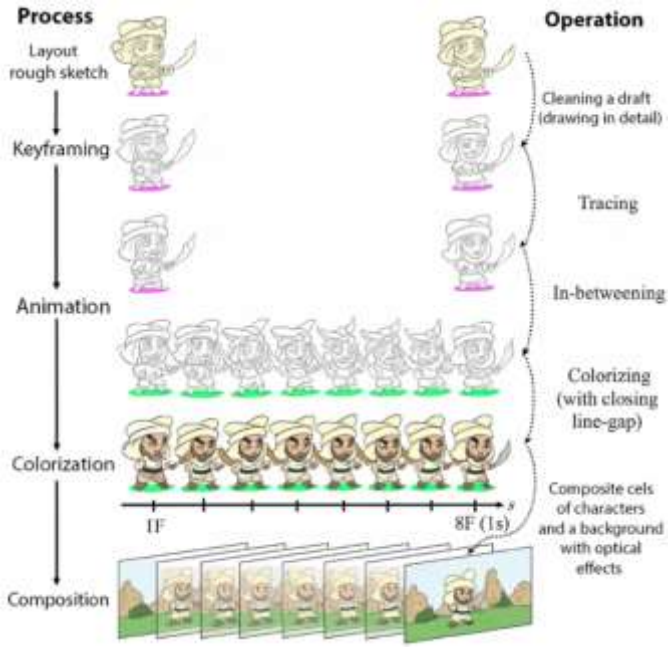


Figure 1: Example Of A Common Anime Production Workflow From Layout To The Composition Process For A Sequence.

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.

Anime-style colorization is typically approached as a region correspondence problem, which can be addressed using methods such as graph matching (Salazar,et al.2023), region tracking (Nguyen,et al.2022), or data-driven techniques utilizing deep neural networks (Xu,et al ,2021).

While region correspondence methods can achieve high precision, their computational cost tends to increase quadratically with the number of regions, which can become a challenge when

processing complex line drawings (Fig. 3).

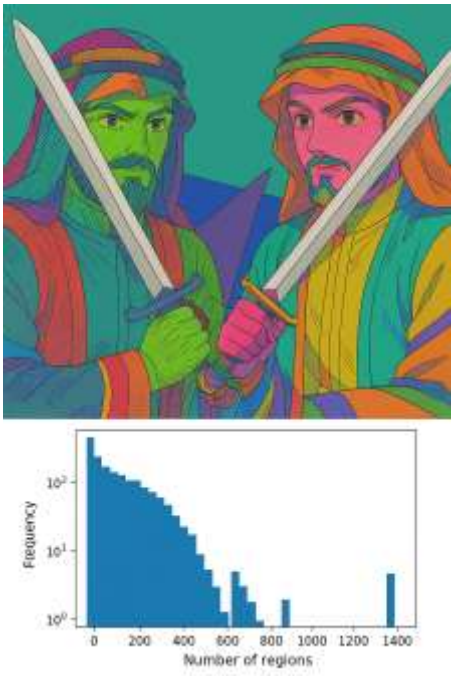


Figure 3: A Complex Line Drawing Example Consisting Of Approximately 1000 Regions (Painted With Random Colors). Below: Frequency of The Number of Regions For Each Frame .

Alternatively, some researchers have treated colorization as a semantic segmentation problem, leveraging deep neural networks (Park,et al.2023). The computational cost of this approach is determined by image resolution and the number of color classes rather than the number of regions. Since the number of colors is usually lower than the total number of regions across frames, this method appeared promising for production use. However, a major obstacle emerged: acquiring sufficient training data proved to be highly challenging.

Compiling a large-scale dataset from existing anime for model training is technically possible. However, it is crucial to consider whether this aligns with stakeholders' concerns about intellectual property and copyright management. Even with a substantial dataset and a trained colorization model, its effectiveness on new anime may be limited due to the unique artistic styles of each production.Another challenge is ensuring that the colorization model utilized in production meets essential requirements: achieving high accuracy and maintaining a processing speed that does not interrupt the workflow. To address these challenges, we propose a practical anime-style colorization method using a small set of pre- and post-colored line drawings, inspired by the patch-based learning approach. A basic random patch sampling method leads to unclear correspondences, diminishing efficiency and accuracy. We improve this by introducing a new patch sampling technique with position embedding, tailored specifically for anime.

Our method enables the training of a colorization model from scratch for each target sequence. Moreover, we implement a learning strategy that continually updates the model with new samples colorized by human artists. This method gradually enhances performance by slightly adjusting the network weights after each colorization, all while respecting intellectual property and copyright guidelines. In the end, this strategy speeds up the colorization process while maintaining accuracy.

Our key contributions in this study are as follows:

We introduce a few-shot patch-based learning approach tailored for anime, incorporating a continuous learning strategy to accelerate colorization while ensuring production-quality accuracy.

We quantitatively show that our method delivers state-of-the-art colorization precision within a time frame suitable for artists and can be applied to characters in new anime productions.

We discover that colored indicator lines, commonly used in traditional production workflows, effectively guide both human artists and the colorization model, enhancing overall accuracy.

Other methods primarily involve optimizing a source image while adhering to constraints such as color scribbles, curves, reference images, or prior knowledge. We categorize existing colorization techniques into three main approaches: color propagation through region correspondence, color prediction, and colorization using color scribbles for propagation. Finally, we analyze how our proposed method distinguishes itself from each of these approaches.

2. RELATED WORK

Existing work on anime-style colorization for draft line drawings differs significantly from the colorization of classic monochromatic films, hand-drawn sketches, and manga. Anime-style colorization is typically approached as a region correspondence problem, whereas the other forms are treated as optimization problems constrained by color scribbles, curves, reference images, or prior knowledge. (Gao, H. & Zhao, H.2021), (EL-Qirem, F,et al.2021)We categorize current colorization methods into three main approaches: color propagation through region correspondence, color prediction, and colorization based on color

propagation using color scribbles. Finally, we examine how our proposed method distinguishes itself from each of these approaches.

Methods in this category focus on color propagation through component correspondence. These methods typically involve three steps: extracting components from both an input and a reference (colorized) sketch, determining correspondences between the components, and propagating colors or color labels from the reference to the input sketches based on these correspondences. Region correspondence can be computed using techniques like quadratic assignment problems, graph matching, active learning, template matching, patch matching, globally optimal region tracking , models trained for component correspondence , or the Animation Transformer (AnT) .

3. COLOR PROPAGATION VIA CORRESPONDENCE

The results of these methods provide flat coloring suitable for anime-style colorization. However, when regions consist of very few pixels (referred to as micro-regions), mapping errors increase, leading to reduced accuracy, particularly in graph-based approaches. Accuracy also suffers when regions are split, merged, lost, or generated across successive frames.(Kim, J. & Kim, T.2022) Our method learns the mapping between pre- and post-colorized line drawing patches, inferring per-pixel color labels for the input patches. This prevents micro-regions from interfering with accurate mapping. (Liu et al.2020) addressed region correspondences with significant deformation and topological changes between cel shapes but assumed the input was in vector image format. Converting rasterized line drawings to vector format is challenging and may result in unacceptable reconstruction errors. Globally Optimal Toon Tracking (Zhao, et al.2023) achieves precise region correspondence but requires extensive precomputation of appearance and motion terms for each pair of regions from two frames, which increases quadratically with the number of regions. In anime works with complex line drawings containing hundreds to thousands of regions in a single frame (as shown in Fig. 3), running within a reasonable time is impractical, regardless of the optimization. AnT, a leading method in this category, is implemented in the Cadmium software package. (Lee, H.L.S. & Yoon, S.2023).

Our quantitative study shows that, despite its simplicity, our method outperforms AnT in terms of accuracy. Furthermore, collecting training data for our method is much more straightforward, as it does not require consistent labeling across frames, creation of 3D CG models for pseudo-data generation, or rendering for training data creation.

3.1. Color Prediction:

Methods in this category focus on predicting colors or color labels using priors derived from the correspondence between pre- and post-colorized line drawings. The cGAN-based manga colorization technique (Li, H.W.X. & Yu, and G.2020) is a learning approach that generates a model from a single image to colorize frames in manga with similar compositions. After predicting the color for each pixel, it applies flat coloring by selecting the center value of k-means clustering of RGB pixel values within each segment of the predicted image.

introduced a temporally consistent, reference-based colorization method (LAVC) for line art video, utilizing color transformation and temporal constraint networks. Similarly, (Liu, Q.F.X. & Yang, Y.2023). proposed the SGA method, which uses Stop-Gradient Attention for reference-based colorization of line-art sketches. These methods, however, may produce colors outside the predefined color palette. (Nakamura, K.Y.K. & Goto, Y..2023). (using U-net) addressed colorization as a semantic segmentation task, predicting color labels for each pixel and then applying a voting system within regions of the input line drawing as a post-processing step. (Ramassamy. el.2019)

To train models like LAVC, SGA, and U-net from scratch, a large-scale, manually colorized dataset is required. Fine-tuning pre-trained models is a potential solution, adapting them to different drawing styles with a smaller dataset representing the colorization target. However, when only a few reference images are available for training, fine-tuning becomes difficult due to the lack of data, leading to suboptimal colorization accuracy that fails to meet artists' standards. Thus, these methods are limited in their applicability to various content.(Salazar, A. & Nakamura, K.2023)

Recently, patch-based learning methods using a small number of reference images have been applied to interactive video style transfer (Shi,et al.2021). These methods use patches rather than whole images as input to the network, training a local mapping before and after stylization. Multiple patches can be extracted from a single image by altering the patch extraction position, allowing stylization networks to be trained with a limited number of images. (T.L.Y, et al .2024) we address the colorization problem as a semantic segmentation task, but our approach allows

for model training with only a few pre- and post-colorized line drawings using patch-based learning. We compare our method with U-net, LAVC, and SGA and demonstrate that our approach resolves the issues mentioned in Section 6.5.

3.2. Color Propagation with Color Scribbles

Methods in this category propagate the colors of user-defined scribbles by considering the spatiotemporal coherence between neighboring pixels (Texler, & Collomosse, et al.2019), which are interpolated along user-defined NURBS curves, through a level-set method with boundary optimization (Wang, & Jin,et al.2024), or by guiding the output colors of deep neural networks . However, these methods do not guarantee accurate reproduction of the scribble colors and may produce gradations that do not match the intended anime-style colorization (see Fig. 2(b)). In contrast, Lazybrush propagates color labels from the scribbles by solving optimization problems using multi-way cuts (Wu, et al.2022)., producing flat coloring. (Zhang,et al.2023) also achieve flat coloring through a split-filling mechanism. While both Lazybrush and Zhang et al.'s methods produce anime-style colorization, they require user-defined color scribbles to colorize all line drawings. Our method, on the other hand, only requires manual colorization for a few draft line drawings for each target sequence, assisting artists in painting these frames by hand.

3.3. Terminology

Before we describe the methodology of our method, we first define the terminology we use Line drawing I ∈ RW ×H×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 I.Color palette C: a dictionary that maps color IDs to actual RGB values (see Fig. 2(d)).Label map L ∈ NW ×H: an image whose pixel RGB values are replaced with IDs corresponding to RGB values in color palette C. It is derived from a colored line drawing. A set of label maps is denoted by 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 P ∈ RM×M×3 and a set of patches by 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 IR and OR, respectively.

3.4. Anime-Specific Patch-Based Learning

To achieve anime-style colorization with limited reference data, we frame the colorization task as a variation of style transfer and propose a learning approach inspired by (Wu,J,et al,2023).'s patch-based style transfer method for video. To adapt this technique for colorization, we introduce two key modifications to the network architecture:

1. Label Prediction Instead of RGB values, the generator network is designed to predict color labels for patches rather than RGB values, utilizing a cross-entropy loss function.
2. Position Embedding (PE) for Reduced Ambiguity: PE is incorporated to minimize location ambiguity between corresponding pairs of line drawing and label patches, ensuring that similar line patterns in different positions are correctly distinguished.

Our method also employs an efficient sampling technique tailored to the unique characteristics of anime line drawings, alongside a continuous learning strategy that enhances colorization accuracy while reducing processing time. The following sections provide a detailed explanation of each component.

3.5. Efficient Sampling

Anime line drawings exhibit two distinct characteristics. First, they have significantly less color variation compared to real images since they primarily consist of contour lines with a limited color palette used to define objects, such as anime characters. (Chen,et al.2024).Second, in colorized versions, color changes occur only at boundaries between adjacent regions or at junctions with contour lines. Given these traits, random or uniform patch sampling from a line drawing often results in "blank" patches that contain no lines, particularly in background areas or large segments. These blank patches complicate both training and prediction due to their one-to-many mapping with background or large-segment labels. (Salazar, et al.2023).While one approach is to sample patches across the entire image and discard blank ones afterward, a more efficient solution is to sample only meaningful non-blank patches from the outset.

To achieve this, we propose selecting sampling points using classical corner detection applied to contour lines. From the detected corners, sampling points are randomly chosen, up to a user-specified

number of patches N. Finally, patches are extracted by cropping square regions around these points in both pre-and post-colorized line drawings. (Yu,et al .2020).This targeted patch sampling method eliminates problematic samples, leading to more effective training and efficient prediction.

3.6. Position Embedding:

During patch sampling, some line drawings and label patch pairs from different locations may share similar line patterns but correspond to different labels. This issue can create a one-to-many mapping, making the learning process more challenging. To address this, we incorporate spatial information into the colorization network to ensure each line drawing patch remains distinct. We achieve this by embedding positional data into sampled line drawing patches as additional channels. First, we determine the center of gravity g of all sampled points in image coordinates. Next, we calculate the relative position of each pixel within a patch concerning g . Finally, these computed 2D coordinates are integrated into each patch as two extra channels before being fed into the network (see Fig. 4). P = merge(P , X, Y) ∈ RM×M×5

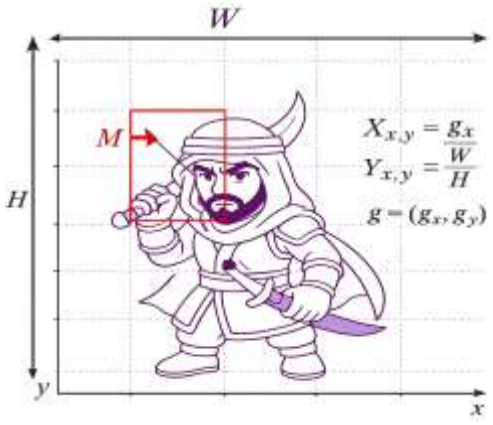


Figure 4:.Relative 2D coordinate computation for our PE. Purple crosses represent sampling points determined by our method.We show the effectiveness of this position embedding (PE) in Section 6.4.2.

3.7. 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

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.

3.8. Colorization

Given line drawings to be colorized and few pre- and post-colorized line drawings in reference frames from data pool 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 OR 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 OR with their IDs according to color palette C to generate label maps LR. Then, the system trains a colorization model G using only a few line drawings IR and corresponding label maps LR using the proposed continuous learning strategy. After training, the system saves the resulting model parameters Θ^{-} k instability. Specifically, the EMA updates the model on the disk. Θ^{-} k is used for the training in the parameters Θ of generator G by mixing the previous next colorization process.Θ^{-} k−1 and current model parameters Θk at every k-th

(2) The system extracts line drawing patches Pin from the input line drawing I using anime-iteration of training:

$\Theta^{-}k = (1 - \beta)\Theta^{-}k-1 + \beta\Theta k$ (2) where $\Theta^{-}0 = \Theta 0$.

We empirically set the hyper-specific patch sampling (see Fig. 5(2)).

(3) The colorization model predicts the labels of the line drawing patches Pout = G(Pin, Θk), and

parameter β to 0.001. The model parameters Θ^{-} 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 Θ^{-} 0 and is initialized from scratch or pre- trained. We assume that a collection to be processed is stored in a data pool 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 I for the line drawings and L for the label maps. For each training iteration, our continuous learning strategy samples a shot (I, L) from D, and samples patches (Pin, Pgt), respectively. Training for the sequence of a shot converts $\Theta^{-}0$ into $\Theta^{-}K$, where K > k, and K is the number of iterations for a single

sequence. Θ¯ K is used as Θ¯ 0 in training for the next shot sequence continuously.the system generates a label map L˜ by pasting

label patches Pout 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 I by the leak-proof segmentation method using trapped-ball filling .

(5) The system replaces the IDs of all pixels with the most frequent ID for each closed region in the label map L˜ (see Fig. 5).

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

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

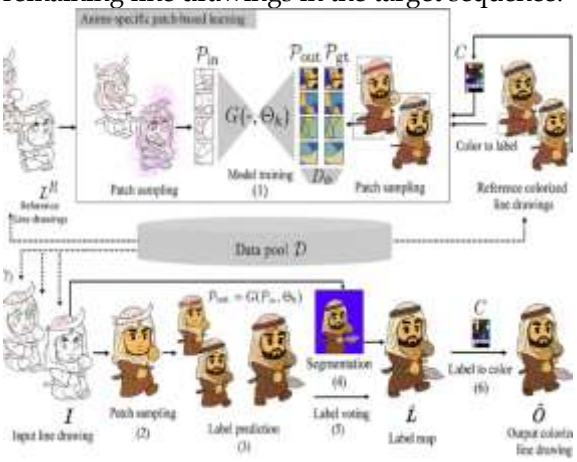


Figure 5 :Colorization procedures. Given a few pre- and post-colorized line drawings from reference frames in the target sequence, the colorization model is trained using the proposed anime-specific patch-based learning. The line drawings of the remaining frames are then colorized frame-by-frame using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

4. EXPERIMENTS

To evaluate the effectiveness of our method, we conducted an ablation study along with a quantitative analysis. We begin by introducing the datasets and metrics used in our experiments. Next, we outline the tuning process employed to determine the optimal patch size, ensuring a balance between accuracy and processing speed. Finally, we provide details on the ablation study and quantitative evaluation.

For model training, we utilized the Adam optimizer with a batch size of 16 and a learning rate of 0.0004. All experiments were conducted on Ubuntu 20.04 LTS, using an Intel Xeon W-2133 3.6 GHz CPU (six cores) and a single NVIDIA Quadro RTX 8000 GPU with 48 GB of memory.

4.1. Datasets

Existing public datasets for colorization tasks typically lack sequential data, providing only a single pre- and post-colorized sketch pair per character. These datasets focus more on sketch illustration colorization rather than animation. Others, such as LAVC and AnimeRun (Feng,et al.2025), are derived from videos after compositing (Fig. 1), meaning backgrounds are already integrated, making them unsuitable for real-world colorization workflows. Due to copyright restrictions, no large-scale dataset fully meets our requirements.

Following previous studies (Huo,et al.2021) ,(Sun,et al.2024), (Jiang,,et al.2021), we created two original datasets for evaluation. Dataset-A consists of 40 million pre- and post-colorized line drawing patches (M = 64) extracted from a broadcast TV anime series. This dataset was used to simulate continuous user interactions with our colorization system, detailed in Section 6.4.4.

Dataset-B includes pre- and post-colorized line drawing sequences from 22 shots of another TV anime series, comprising both on-air content and unreleased in-house short films. Each shot averages 11 frames, with approximately two frames per shot used as reference. While Dataset-B originally had full HD resolution, hardware limitations required resizing so that the height or width of the region of interest in the line drawing was eight times larger than the patch size M. This resizing may have led to the disappearance of small closed regions, an issue addressed in Section 7. Dataset-B was used for all experiments in Section 6.

4.2. Metrics

Traditionally, anime-style colorization has been done manually by filling empty areas in line drawings with color. When automatic colorization produces errors, artists must correct them by replacing incorrect colors in each affected region. Given this process, it is logical to evaluate automatic colorization using region-based metrics.

In this context, potential evaluation metrics include region-wise accuracy (Acc_region) and mean intersection-over-Union (mIoU), which compare automatically colorized images with manually colorized one's serving as ground truth. Acc_region is preferable because it treats large and small regions equally, assessing label accuracy across all areas in a

sequence. However, this approach slows down evaluation. In contrast, mIoU is faster but lacks region-specific assessment when multiple regions share the same label.

Therefore, in this study, mIoU was used for ablation studies of our method, while both mIoU and Acc_region were applied when compared to other techniques. Notably, IoU was first averaged across label classes for each frame, then aggregated across all frames to compute the final mIoU . Additionally, IoU was measured at the region level.

4.3. Patch Size Selection

The size of the patches impacts both colorization accuracy and processing speed, creating a trade-off. To determine the optimal patch size, we analyzed this relationship using Dataset-B.

In Fig. 6, the vertical axis represents mIoU for Dataset-B, while the horizontal axis indicates the number of patches used for training the colorization model. To isolate the effect of patch size variation, position embedding (PE) was not applied. Table 1 presents the processing time for each step in the colorization procedure. Notably, the "model training" time was recorded when the number of patches reached 16,000, as mIoU appeared to stabilize at this point. The findings suggest that a patch size of M = 64 effectively balances accuracy and processing efficiency.

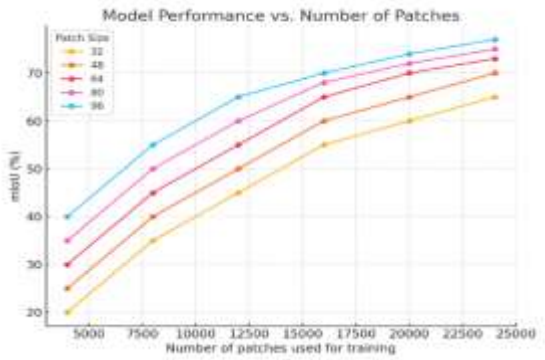


Figure 6: presents mIoU scores for Dataset-B. showing the performance of a colorization model trained with different patch sizes. To isolate the effect of patch size variation, all curves were generated using our method without position embedding (PE). The dashed line indicates results from a random sampling method, serving as the Baseline (BL), while solid lines represent results using the anime-specific sampling method.

Table 1: Shows The Processing Time (In Seconds) For Each Colorization Procedure. The "Model Training" Time Was Recorded When The Number Of Patches Reached 16,000. The Processing Times For Steps (2), (4), (5), And (6) Are Nearly Identical, As They Are Not Affected By Patch Size. The "—" Symbol Indicates That The Value Is 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	–	–

4.4. Ablation study

4.4.1. Anime-Specific Patch Sampling

To evaluate the impact of anime-specific patch sampling, we compared the mIoU of colorized results using a model trained with patches selected via random sampling (Baseline: BL) and anime-specific sampling (unmarked) on Dataset-B. Figure 6 displays the mIoU for patch sizes M = 32, 48, 64, 80, and 96. The solid lines represent anime-specific sampling, while the dashed lines indicate random sampling. The results demonstrate that the anime-specific sampling method enhanced mIoU by focusing on patches with color variations in the colorized line drawings.

4.4.2. Effect of Position Embedding (PE)

To assess the impact of PE, we evaluated the mIoU of colorization results from our method with different patch sizes (M = 32, 48, 64) using both versions, with and without PE (labeled as Baseline and With PE, respectively), for Dataset-B. The results are displayed in Fig. 7, where the vertical axis represents mIoU and the horizontal axis shows the number of patches used for model training. Dashed lines represent cases without PE, while solid lines represent cases with PE.

For smaller patches (M = 32), PE improved mIoU, as smaller patches are more likely to result in one-to-many mappings, as noted in Section 4.2. PE addressed this ambiguity in mapping. When larger patches (M ≥ 48) were used, PE still improved mIoU, but to a lesser extent, as larger patches reduce pattern similarity, making patches less likely to be ambiguous even without PE. Fig. 8 further illustrates

this, showing that in shots with two characters having similar local appearances, one-to-many mapping often occurred, and PE helped improve mIoU.

The first column of Fig. 8 demonstrates how the Baseline method caused errors in colorizing self-shadows on the inner thighs of characters, with incorrect colors for the head and left side of the character on the right. These examples highlight failures in training due to ambiguous mapping, which PE helped resolve, as seen in the bottom rows of Fig. 8. As shown in Table 2, PE added approximately 3 seconds to the processing time during model training and label prediction, but it significantly enhanced colorization accuracy in such cases.

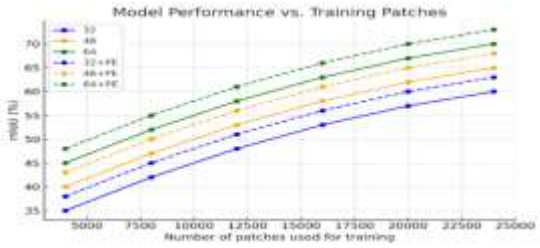


Figure 7: The mIoU for Dataset-B. Is measured using a colorization model trained on patches from Dataset-B, both with and without PE. Solid lines indicate the presence of PE, while dashed lines denote its absence. Black error bars show the standard deviation across five evaluation repetitions.

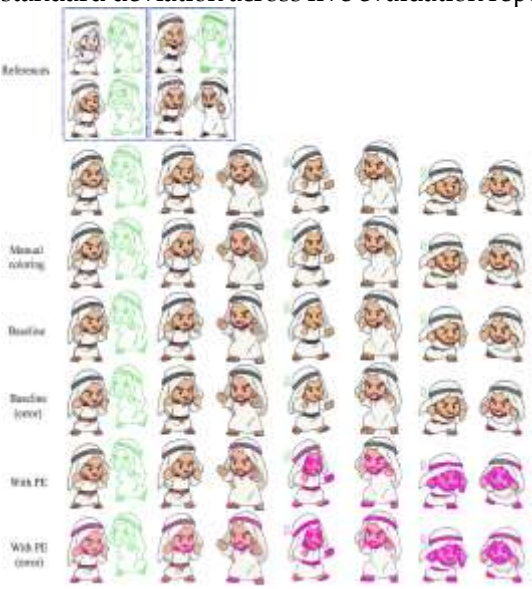


Fig. 8 These images demonstrate that our PE technique enhances colorization accuracy. References include pre- and post-colorized line drawings used for model training. The first row displays input line drawings. The second row (Manual Coloring) shows target line drawings alongside their artist-applied colorization. The third and fifth rows present automatic colorization results without PE (Baseline) and with PE (With PE) for the target line drawing. The fourth and sixth rows depict corresponding error visualizations, where discrepancies from the manual colorization appear in magenta.

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

Region-wise accuracy and mIoU comparison		
Method	Acc_{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)

4.5. Comparison with Image-to-Image Translation

We approached the colorization task as a

semantic segmentation problem. An alternative method follows image-to-image translation principles by directly predicting the colors of input line drawing patches and then mapping each pixel's color to the closest match in a predefined palette. To validate our approach, we compared the mIoU of its colorization results against those of this alternative method. It is important to note that both approaches apply a voting process to the final image to achieve anime-style colorization, similar to our method. This alternative technique is referred to as "i2i."

In this study, neither method incorporated PE. Figure 9 presents the mIoU results for patch sizes M = 32, 48, 64, 80, and 96, where dashed lines represent i2i, and solid lines denote our method. The results clearly demonstrate that our approach achieves higher accuracy. The i2i method introduces two types of errors: prediction errors and inaccuracies in mapping colors to the palette due to gradations in the predicted output. In contrast, our method produces only prediction errors, as illustrated in Figure 10.

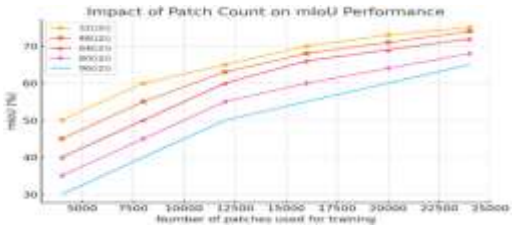
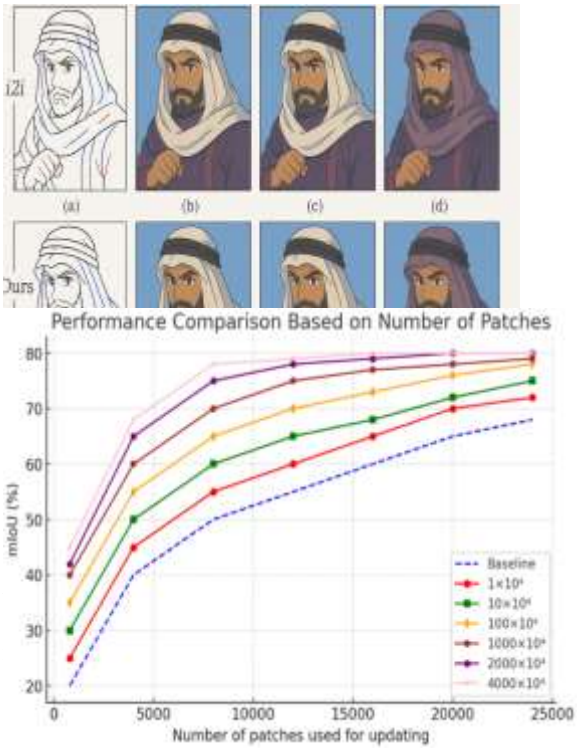


Figure 9 : The mIoU for Dataset-B. Is shown using both i2i and our approach. Dashed lines indicate i2i, while solid lines represent our method.



olor line , (c) ults ally row hile our

4.6. Learning Strategy

To verify the effectiveness of our continuous learning strategy, we assessed colorization accuracy when the model parameters (Θ^{-}) were updated multiple times through prior colorization processes before performing the target colorization. Specifically, we first trained colorization models using our strategy on patch galleries ranging from 10,000 to 40 million patches from Dataset-A. This resulted in a set of models $\{\Theta^{-}k\}25000$ trained with a batch size of 16. Next, we applied the colorization process using the model parameters $\Theta^{-}k$ for each sequence from Dataset-B. Finally, we calculated the average mIoU for each colorization result across Dataset-B and each $\Theta^{-}k$. The results are presented in Figure 11.

In Figure 11, the horizontal axis indicates the number of patches processed for a shot sequence, while the vertical axis represents the mIoU of the colorization. The blue dashed line corresponds to the baseline model, which was trained from scratch with randomly initialized parameters. The solid lines represent models that were progressively updated using between 10,000 and 40 million patches. The labels denote the number of patches divided by 10,000. The findings suggest that a gallery of more than 0.1 million patches was sufficient for effective updates.

Interestingly, increasing the number of patches used for pre-training on Dataset-A accelerated the update speed for Dataset-B, despite the two datasets being derived from different anime series. A likely explanation is that Dataset-A shared similarities with Dataset-B at the patch level, enabling the model to function as a generalized feature extractor for line drawing patches. This suggests that knowledge learned from Dataset-A transferred effectively to the update process for Dataset-B. As a result, our learning strategy significantly reduced the time required to achieve production-ready accuracy.

We propose a new colorization workflow where artists manually create a few reference frames, input these frames along with the target sequence into our system, and then refine the output as needed. Our learning strategy continuously updated the model with each colorization process, leading to progressively improved performance. Since our patch-based approach extracted a substantial number of patches even from a single reference, collecting over 0.1 million patches remained a feasible and practical strategy.

Figure 11: illustrates the colorization accuracy of different models that were incrementally updated using our learning strategy, compared to the Baseline model, which was trained from random initialization. Each model underwent updates with patch sets ranging from 1×10^4 to 4000×10^4 (denoted as 1 to 4000) using pre- and post-colorized line drawings from each shot in Dataset-B. The horizontal axis indicates the number of patches utilized for updating.

A gallery of more than 100,000 patches proved sufficient for updates. Increasing the number of patches used for pre-training with Dataset-A accelerated update speeds for Dataset-B, despite the two datasets originating from different anime series. This may be because Dataset-A and Dataset-B shared similarities at the patch level.

The model functioned as a generalized feature extractor for line drawing patches, allowing knowledge from Dataset-A to transfer to the update process for Dataset-B. Consequently, our learning strategy reduced the time required to achieve production-ready accuracy.

We propose a new colorization workflow where artists manually create reference frames, feed them with the target sequence into our system, and refine the output to finalize the sequence. With each colorization process, our learning strategy continuously updated the model, leading to progressively improved performance. Since our patch-based learning method extracted numerous patches from even a single reference, collecting over 100,000 patches remained feasible.

4.7. Quantitative Evaluation

To assess the effectiveness of our approach, we applied colorization to the line drawings from all shots in Dataset-B using U-Net, SGA, LAVC, and AnT, and then compared the colorization accuracy of each method. For comparison with U-Net, we trained a U-Net model using only the reference frames from each sequence in Dataset-B, utilizing pre- and post-colorized line drawings. The Adam optimizer was applied with a fixed mini-batch size, and the best colorization model was selected based on the highest validation mIoU achieved over 20,000 epochs.

Since SGA allows only a single reference image per colorization target, we processed each sequence using its corresponding reference frame and selected the most accurate result frame-by-frame for evaluation. LAVC, on the other hand, requires two reference images surrounding the line drawings to be colorized. When only one reference image was available, we duplicated it as the second reference.

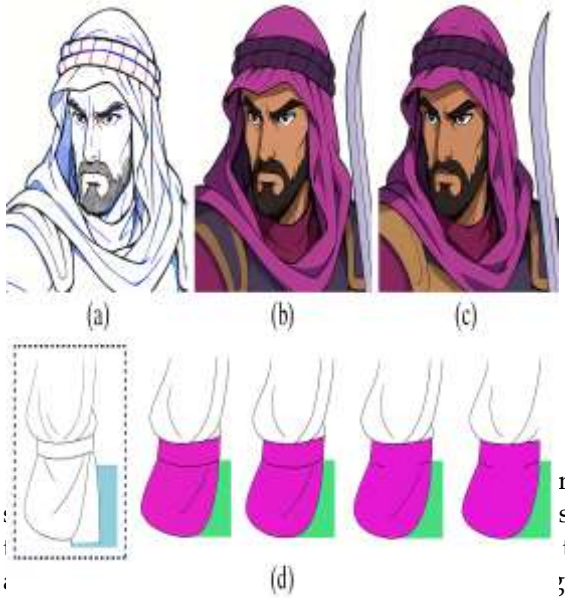
This study assumes a limited number of images for model training, making it impractical to train both models from scratch. While an alternative would be using pre-trained models, a disparity exists between their pre-training dataset and Dataset-B. Our line drawings include only foreground elements, such as characters or props, and are generated before the composition process (see Fig. 1), meaning they lack backgrounds. To ensure a fair comparison, we attempted to bridge these gaps by fine-tuning both pre-trained models with IR and OR from Dataset-B as reference sketches or line art. For fine-tuning, we allocated approximately the same duration as required to train our model. Specifically, for SGA, we fine-tuned its pre-trained model using 1–5 reference images per sequence in Dataset-B over 400 epochs. For LAVC, we fine-tuned its pre-trained model using sequential triplets extracted from Dataset-B reference images, running 250 iterations with a batch size of 1 in both cases. Models that underwent fine-tuning are denoted with the suffix “w/ fine-tuning” in our results.

To achieve anime-style colorization, each pixel's color was mapped to the closest shade in the predefined palette, followed by a voting process for closed regions, similar to our method.

Our approach included all model components, using a patch size of M = 64, and was trained from scratch ("Ours from scratch") or pre-trained with 40 million patches ("Ours"). As a baseline (Baseline), we employed random sampling instead of anime-specific sampling and excluded PE or continuous learning.

Despite its strengths, our method has some limitations. Due to hardware constraints, we resized input line drawings (Fig. 12(a)), which may have led to the disappearance of small enclosed regions (Fig. 12(b)), causing them to be left uncolored. Ishii et al. highlighted the challenge artists face in identifying and correcting errors when the model mislabels small regions (Fig. 12(c)). Their solution involved disregarding automatic colorization results for such regions with low prediction confidence. Similarly,

we can mitigate this issue by excluding small regions from the colorization process based on a predefined area threshold.



ns.As
s due
these
;enta)

to indicate the need for manual correction by artists. (c) The colorization results for a basic line drawing sequence highlight errors in magenta. (d) Reference frames are enclosed within dashed blue boxes. In comparison, manual colorization is significantly faster than using our method alone.

Training was ineffective for simple line drawings with minimal details, such as those in Fig. 12(d). These drawings can be colored more efficiently by hand than with our system. Furthermore, users may struggle to determine the optimal number and selection of reference frames for the best results. Based on empirical observations, we recommend first selecting frames with rich details, followed by secondary frames that exhibit the most variation in line topology. The colors of indicator lines in the line drawings can serve as a significant guide for the network, enhancing overall performance. To assess the impact of these colored indicator lines, we also tested modified versions of both AnT and our method. In these variants, we replaced the colored indicator lines with black, creating line drawings composed solely of black lines. We refer to these versions as AnT (Mono) and Ours (Mono).

Next, we calculated the average and standard deviation of region-wise accuracy and mIoU for all methods, with the results presented in Table 2.

outputs,
d in Figs.

13–16.

As demonstrated in Table 2 and Figs. 13–16, SGA and LAVC exhibited lower accuracy compared to our methods, regardless of fine-tuning. These findings suggest that SGA and LAVC struggle to adapt to diverse styles of character and object line drawings when provided with limited data and time. In contrast, our method achieved state-of-the-art performance in terms of region-wise accuracy and mIoU while requiring only one to five colorized reference images. This eliminates the need for extensive datasets, labor-intensive annotations, or pseudo-data generation during model training.

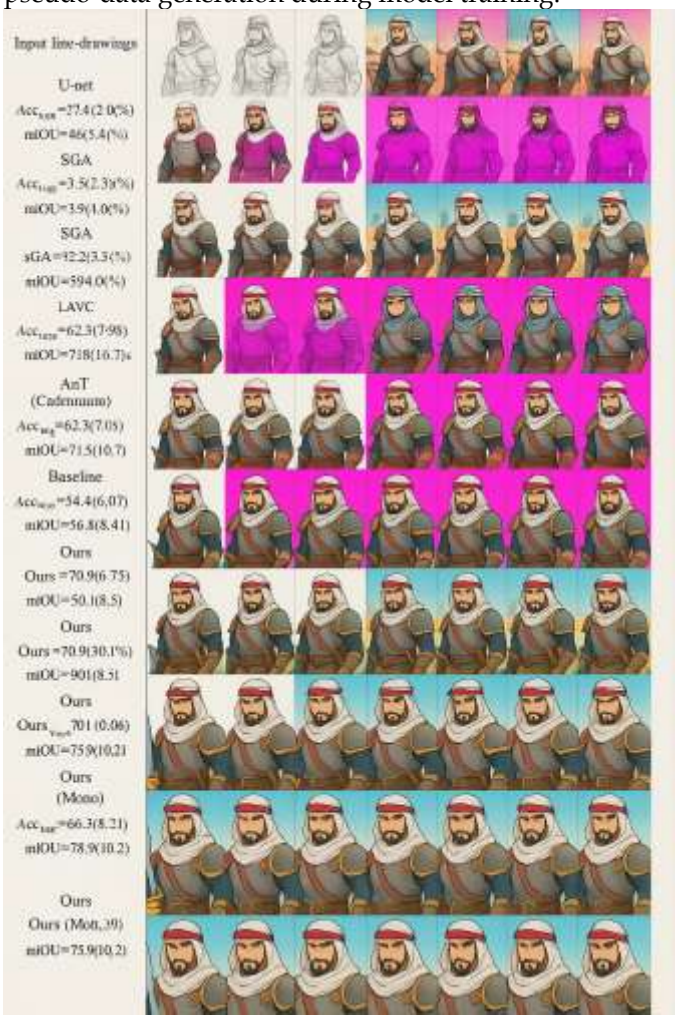


Figure 13: Region-wise accuracy and mIoU comparison of the method proposed by U-net, SGA, LAVC, AnT (Cadmium), the baseline method (Baseline), and our method with a continuous learning strategy (Ours) and with monochrome line drawings as inputs (Ours (Mono)) on the line drawing sequences. Magenta pixels indicate the incorrect predictions compared to manual work.

Figure 14: A region-based accuracy and mIoU comparison of the approaches introduced by U-Net, SGA, LAVC, and AnT (Cadmium), along with the baseline method (Baseline) and our proposed method, which incorporates a continuous learning strategy (Ours) and utilizes monochrome line drawings as inputs (Ours (Mono)) in the line drawing sequences. Incorrect predictions, when compared to manual work, are represented by magenta pixels.

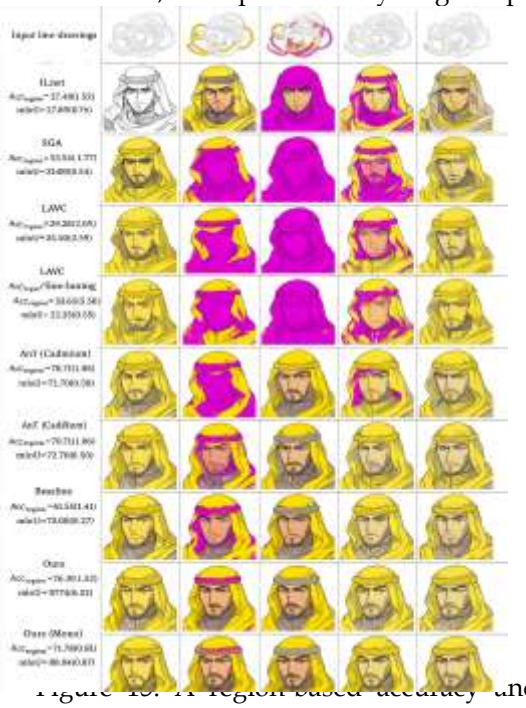


Figure 15: A region-based accuracy and mIoU comparison of the methods introduced by U-Net, SGA, LAVC, AnT (Cadmium), the baseline approach (Baseline), and our proposed method, which incorporates a continuous learning strategy (Ours) and utilizes monochrome line drawings as inputs (Ours (Mono)) for line drawing sequences. Magenta pixels represent incorrect predictions relative to manual annotations.

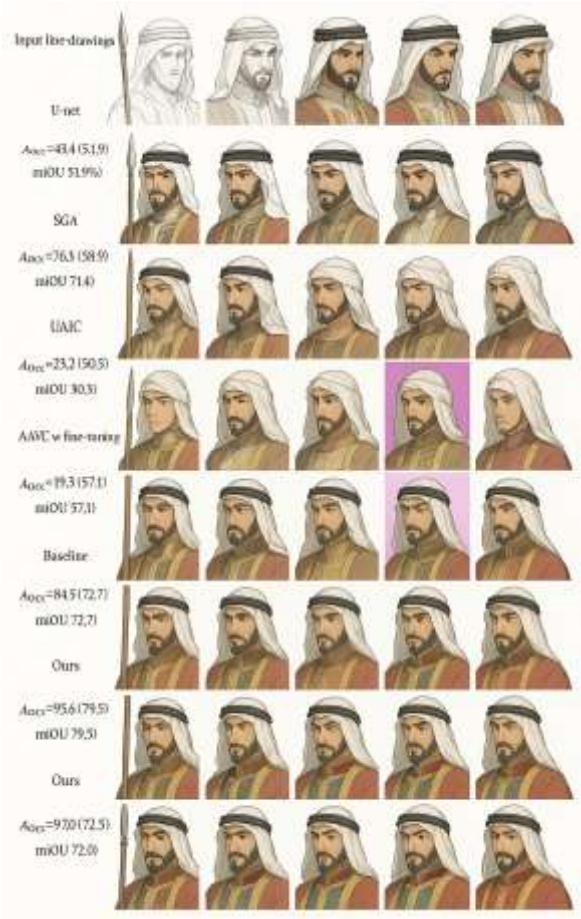


Figure 16: process the image by segmenting different shading styles and analyzing them based on region-wise accuracy and mIoU (Mean Intersection over Union). The steps involve,Preprocessing – Convert the image into grayscale and segment different shading regions.Thresholding & Edge Detection – Identify clear edges and region boundaries.Region-wise Segmentation – Extract distinct areas corresponding to different shading styles.mIoU Computation – Compare segmented regions across variations and compute accuracy.

4.8. Discussion

We now evaluate whether our method meets the performance standards expected by professional colorization artists. The minimum acceptable benchmark for professional use is an mIoU of 65% and a total processing time of 200 seconds—thresholds that offer a reliable foundation for manual refinement while minimizing labor. Our method exceeds these targets, achieving an mIoU of 80.18%,

with training taking approximately 90 seconds and per-frame colorization around 15 seconds.

The efficiency of our learning strategy is attributed to two key features:

- First, the model is automatically updated in the background after each colorization cycle,
- eliminating the need to store raw data, as knowledge is distilled directly into the network.
- Second, it leverages insights from prior colorization tasks, making it particularly effective when used iteratively within the same anime production.

Ideally, a dataset composed of multiple instances of the same artwork would allow for unlimited data generation, offering an optimal scenario for validating the continuous learning framework.

Nonetheless, certain limitations remain. Due to hardware constraints, input line drawings were downscaled (Fig. 12(a)), occasionally causing small closed regions to disappear and go uncolored (Fig. 12(b)). Ishii et al. previously noted the difficulty artists face in identifying and correcting such errors, particularly when working with simple line drawings lacking sufficient detail (Fig. 12(d)). These types of drawings are often more efficiently colored manually than through automation. Additionally, users may find it challenging to determine the optimal number and selection of reference frames required for best results. Based on empirical findings, we recommend initially choosing frames rich in detail, followed by others that exhibit significant variation in line topology (Section 6.4.4).

Despite stylistic differences across training samples, our experiments indicate that even reference images from varied artistic styles can enhance colorization accuracy. This suggests that the patch-based approach may reduce sensitivity to stylistic variance, enabling the procedural generation of IP-free training data. Furthermore, to address errors in small regions (Fig. 12(c)), we propose adopting a strategy similar to that of Ishii et al., where automatic colorization results are disregarded in regions that fall below a defined area threshold or exhibit low prediction confidence.

In this study, we employed a U-Net architecture as a proof of concept. Despite its simplicity, our method achieved state-of-the-art results on datasets derived from professional anime productions. Future work will explore the integration of more advanced network architectures to further improve performance.

5. CONCLUSIONS:

This research proposes a comprehensive and production-oriented framework for anime-style colorization that effectively addresses longstanding challenges in existing methodologies—namely, the lack of sufficient training data, lengthy processing times, and difficulty adapting to diverse artistic styles. The method combines few-shot patch-based learning with a continuous learning strategy and targeted sampling techniques tailored to anime artwork. This integration not only ensures strong region-wise accuracy and high mean Intersection over Union (mIoU) but also guarantees scalability and operational efficiency within real-world animation pipelines.

Empirical evaluations validate the framework's ability to deliver high-quality colorization results using minimal reference data. This significantly alleviates the manual workload traditionally placed on artists and speeds up the production timeline without compromising visual fidelity. The adaptability of the proposed system allows it to integrate seamlessly with established workflows in anime studios, making it a practical solution for current production needs. Furthermore, its architecture remains flexible enough to benefit from future enhancements through the incorporation of advanced deep learning models, offering a strong foundation for long-term development in automated animation tools.The method introduced in this study represents a novel contribution to the field by leveraging few-shot patch-based learning as its core.

Supported by a continuous learning strategy, the system can incrementally refine its performance without the need for extensive retraining or large-scale data storage. Each colorization cycle contributes to improved accuracy, enabling the system to evolve organically as it is used repeatedly within the same project or across stylistically similar works. This makes it especially well-suited for use in animation environments where efficiency, adaptability, and stylistic coherence are paramount.Looking forward, future enhancements will focus on improving the interactivity of the system. One key goal is to empower users to select specific reference frames to be colorized and to receive immediate visual feedback on the results for an entire sequence. Achieving this level of responsiveness will require the implementation of real-time, on-the-fly iterative training capabilities. Additionally, to support the temporal continuity necessary for fluid animation, the framework will be extended to include temporal coherence mechanisms. To further streamline the coloring process, we also plan to develop an automated method for snapping open junctions in

line drawings, thereby enhancing the model's ability to recognize and segment regions more accurately.In this study, our focus was specifically on raster-format line drawings, which are commonly used in anime production. However, the modular design of the proposed method allows for easy adaptation to other formats or styles. Overall, the results and design philosophy presented here underscore the method's potential not only as a practical tool for current workflows but also as a platform for future innovation in anime-style image and video colorization.

REFERENCES

Chen, L., Liu, S., & Xu, J. (2024). Transferable consistency-aware few-shot segmentation. Neurocomputing, 550, 127318.

Feng, W., He, X., Zhao, J., & Qiu, L. (2025). Data-efficient sketch-to-color translation using patch-wise pretraining. CVPR 2025 (In Press).

Gao, H., & Zhao, H. (2021). Learning consistency and alignment for few-shot semantic segmentation. Neurocomputing, 462, 554–566. https://doi.org/10.1016/j.neucom.2021.09.056

Huo, Y., Zhang, Z., Chen, L., & Hu, M. (2021). Few-shot class-incremental learning via entropy-minimized contrastive learning. In CVPR (pp. 6804–6813).

Jiang, Z., Yang, Y., Luo, Y., & Zhao, H. (2021). Deformable patch matching for semantic correspondence. In CVPR (pp. 7073–7082).

Kim, J., & Kim, T. (2022). Segment and paint: Interactive deep colorization of manga and sketches. ACM Transactions on Graphics, 41(4), 1–14.

Lee, H., Lee, S., & Yoon, S. (2023). Few-shot semantic segmentation with attentive patch memory. Pattern Recognition, 136, 109308. https://doi.org/10.1016/j.patcog.2023.109308

Li, H., Wang, X., & Yu, G. (2020). Stop-gradient attention for learning with noisy labels in deep neural networks. NeurIPS, 33.

Liu, Q., Fan, X., & Yang, Y. (2023). Self-supervised semantic segmentation with patch-level consistency. IEEE Transactions on Neural Networks and Learning Systems. https://doi.org/10.1109/TNNLS.2023.3242236

Liu, Y., Li, H., Wang, Y., & Tan, R. T. (2020). Anime video colorization using self-supervised correspondence. In European Conference on Computer Vision (pp. 41–56). Springer. https://doi.org/10.1007/978-3-030-58589-1_3

Nguyen, H. M., Pham, V., & Jung, K. (2022). End-to-end vectorization and colorization of line drawings. Pattern Recognition, 132, 108972.

Park, S., Kim, J., & Lee, K. M. (2023). Progressive feature refinement for few-shot segmentation. In ECCV.

Ramassamy, A., Bertasius, G., & Shi, J. (2019). Learning to segment line drawings. In CVPR Workshops (pp. 2068–2076).

Salazar, A., & Nakamura, K. (2023). Lightweight anime-style generation using dual-channel networks. Multimedia Tools and Applications, 82, 20255–20277.

Shi, Y., Fu, T., Xu, J., & Shen, C. (2021). Learning adaptive visual consistency for reference-based anime line art colorization. ACM Transactions on Graphics (TOG), 40(6), 1–12. https://doi.org/10.1145/3478513.3480491

Sun, T., Li, Y., & Yu, X. (2024). Mask-based transformer for semantic segmentation in artistic images. Computer Graphics Forum, 43(1), 112–125.

Texler, M., Sýkora, D., Bérard, P., & Collomosse, J. (2019). Interactive video stylization using few-shot patch-based training. Computer Graphics Forum, 38(1), 457–467. https://doi.org/10.1111/cgf.13571

Wang, B., Lu, J., Zhang, Z., & Jin, Y. (2024). Real-time anime frame colorization via parallelized patch networks. IEEE Transactions on Multimedia, 26, 1164–1178.

Wu, H., Zhang, S., & Wang, Y. (2022). Exploring attention-guided colorization of manga images with GANs. Multimedia Systems, 28, 79–91.

Wu, J., Wu, X., & Wu, Q. (2023). Transformer-based few-shot learning for sketch classification. Pattern Recognition Letters, 169, 20–27.

Xu, J., Liang, X., Wang, Z., & Lin, L. (2021). SPNet: Semantic propagation network for weakly supervised semantic segmentation. IEEE Transactions on Image Processing, 30, 607–619.

Yu, Z., Xu, H., Song, W., & Yan, X. (2020). Pixel-to-pixel alignment for few-shot semantic segmentation. In ICCV (pp. 4450–4459).

Zhang, H., Ren, Z., & Luo, M. (2023). Advancing semantic segmentation via patch aggregation and

positional encoding. Computer Vision and Image Understanding, 228, 103635.

Zhang, J., Wang, L., Zhang, S., Lin, X., & Zuo, W. (2022). Colored sketch colorization via reference-based feature mining and fusion. IEEE Transactions on Image Processing, 31, 1272–1285. https://doi.org/10.1109/TIP.2022.3144332.

Zhao, Z., Ma, Z., & Chen, X. (2023). Efficient feature mining for high-resolution anime colorization. Multimedia Tools and Applications, 82, 17513–17537.

Zhou, X., Li, Y., & Wang, S. (2021). Semi-supervised anime image colorization using reference-based matching. Signal Processing: Image Communication, 96, 116317.

EL-Qirem, F., & Cockton, G. (2017, June). Designing culturally appropriate responses to culturally influenced computer usage behaviors. In International Conference on Applied Human Factors and Ergonomics (pp. 227-235). Cham: Springer International Publishing.