Anime Sketch Colorization by Component-based Matching using Deep Appearance Features and Graph Representation

ICPR 2020, January 10th-15th, 2021

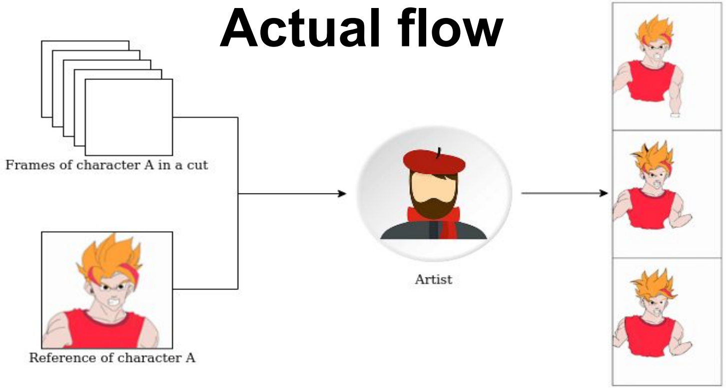
Thien Do*, Van Pham, Anh Nguyen, Trung Dang, Quoc Nguyen, Bach Hoang, Giao Nguyen
Cinnamon AI
Minato, Tokyo 105-0001, Japan
Email: hades@cinnamon.is

cinnamon AI

*Corresponding author

Anime Colorization

Actual flow



Problem

- The cost for artists is expensive
- The artists have to color the image although it is not change too much, so time-consuming

Automatic Colorization

Challenges

- Lack texture information in sketch images
- Complicated objects that require acute coloring
- All pixels in each component must be consistent in output results.

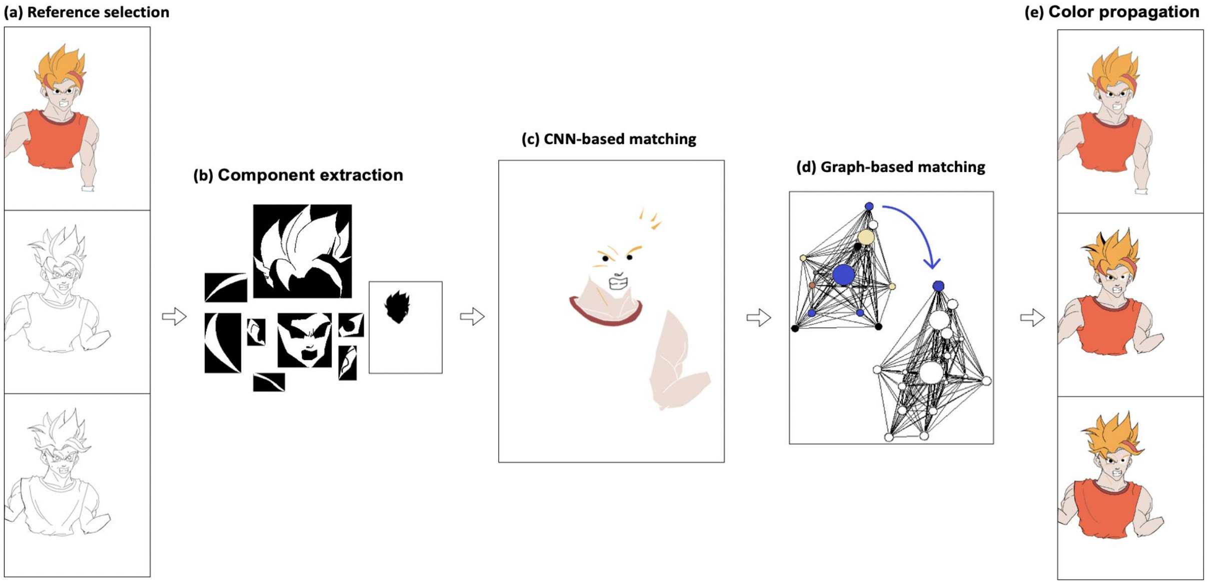
GAN-based Model

There are many models based on GAN architecture designed for automatic colorization:

- Style2paints **[5]** requires human users to identify proper colors and art styles at many different locations so that the model can know how to colorize the input sketch
- Autopainter **[12]** and DeepColor **[13]**, utilize **conditional GAN** to optimize a set of structured losses in the training of the generator so as to generate more coherent color collocations of a sketch
- PaintsChainerV3 **[14]** is a commercial products
- Thasarathan et al. **[15]** and Shi et al. **[16]** propose a GAN with the constraints of temporal and color conditions to improve the colorization

However, these methods produce low-quality results with apparent color mistakes, color bleeding or color distortion when applied on real-world sketches

Automatic Colorization Framework



(a) Colored reference image is the image that has the **largest number of components** in a cut.

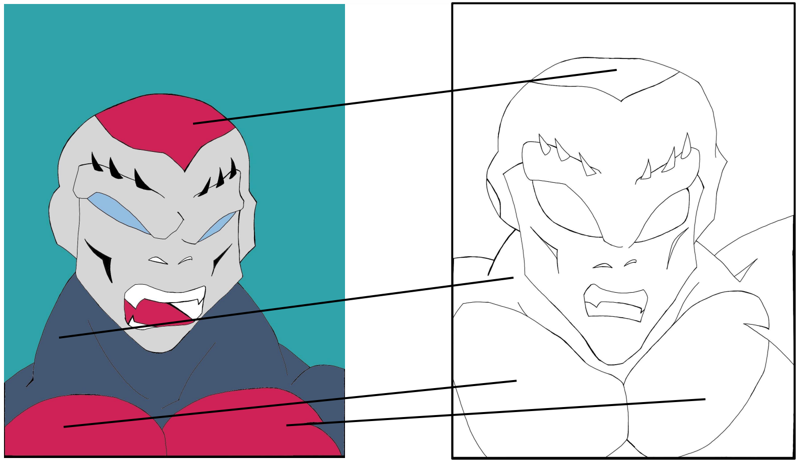
(b) Sketch is segmented into components.

(c) Matching **fixed components** based on features extracted from a ResNet34 backbone

(d) **Deformed components** are matched by **building two graphs** (**[6]**), one from colored image and one sketch.

(e) Colored output becomes the reference colored image to colorize the sketch frame that comes right after or before

The First Matching



$$j^* = \arg\min_j \frac{\|f_{t_i} - f_{r_j}\|_2^2}{max(\|f_{r_i}\|_2^2, \|f_{t_j}\|_2^2)} \quad (1)$$

$$\text{subject to} \begin{cases} D(center_i, center_j) < t_d \\ t_{a_{min}} \leq A(area_i, area_j) \leq t_{a_{max}} \\ \dfrac{\|f_{t_i} - f_{r_j}\|_2^2}{max(\|f_{r_i}\|_2^2, \|f_{t_j}\|_2^2)} < t_f \end{cases} \quad (2)$$

- ft, fr are the features of component in target, reference extracted from ResNet-34 network
- D is Euclidean distance function
- A is the area difference function
- td, ta, tf are thresholds of difference

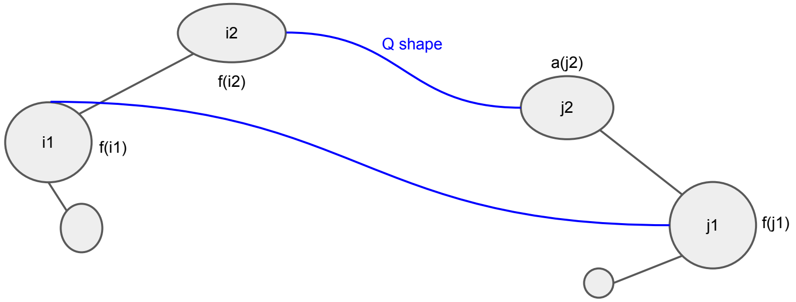
Graph and Affinity Matrix

- Denoting the sketch/target graph as Gt and the reference graph as Gr
- i1, i2 are the index of node in target graph, and j1, j2 are similar in reference graph
- a(i1), d(i1, i2), f(i1) are area of i1, distance between i1 and i2 and feature of i1, respectively

$$Q_{node}(i_1, i_2, j_1, j_2) = Q_{shape}(i_1, j_1).Q_{shape}(i_2, j_2) \quad (6)$$

$$Q_{shape}(i_1, j_1) = \exp\left(-\frac{\gamma}{2}\frac{\|f_{t_{i_1}} - f_{r_{j_1}}\|_2^2}{max(\|f_{t_{i_1}}\|_2^2, \|f_{r_{j_1}}\|_2^2)} +\right) \quad (7)$$

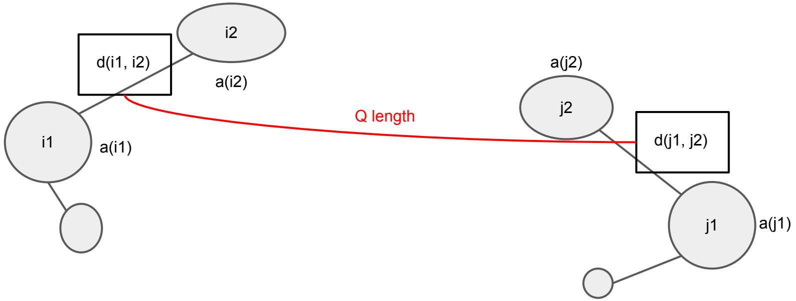
$$Q_{shape}(i_2, j_2) = \exp\left(-\frac{\gamma}{2}\frac{\|f_{t_{i_2}} - f_{r_{j_2}}\|_2^2}{max(\|f_{t_{i_2}}\|_2^2, \|f_{r_{j_2}}\|_2^2)} +\right) \quad (8)$$



$$Q_{edge}(i_1, i_2, j_1, j_2) = Q_{length}.Q_{area} \quad (3)$$

$$Q_{length} = \exp\left(-\alpha\frac{|d_{t_{i_1 i_2}} - d_{r_{j_1 j_2}}|}{max(d_{t_{i_1 i_2}}, d_{r_{j_1 j_2}})}\right) \quad (4)$$

$$Q_{area} = \exp\left(-\beta\frac{|a_{t_{i_1}}.a_{t_{i_2}} - a_{r_{j_1}}.a_{r_{j_2}}|}{max(a_{t_{i_1}}.a_{t_{i_2}}, a_{r_{j_1}}.a_{r_{j_2}})}\right) \quad (5)$$



The Second Matching

Affinity matrix

$$Q(i_1,i_2,j_1,j_2) = Q_{edge}(i_1,i_2,j_1,j_2).Q_{node}(i_1,i_2,j_1,j_2) \tag{9}$$

$$Q = \begin{bmatrix} Q_{1111} & Q_{1112} & Q_{1113} & \cdots & Q_{1211} & \cdots \\ Q_{1121} & Q_{1122} & Q_{1123} & \cdots & Q_{1221} & \cdots \\ Q_{1131} & Q_{1132} & Q_{1133} & \cdots & Q_{1231} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ Q_{2111} & Q_{2112} & Q_{2113} & \cdots & Q_{2211} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \tag{10}$$

Applying the quadratic form

$$\sum_{i_1,i_2,j_1,j_2} Q(i_1,i_2,j_1,j_2) x_{i_1 j_1}.x_{i_2 j_2} = x^T Q x \tag{11}$$

$$x = \begin{bmatrix} x_{i_1 j_1} \\ x_{i_1 j_2} \\ x_{i_1 j 3} \\ \vdots \\ x_{i_2 j_1} \\ x_{i_2 j_2} \\ x_{i_2 j_3} \\ \vdots \end{bmatrix} \tag{12}$$

The optimal solution x*

$$x^* = \arg\max_x x^T Q x \tag{13}$$

$$\text{subject to} \quad Cx = 1, x \in \{0,1\}^{N^2} \tag{14}$$

Experimental Dataset

- The number of sketches in each cut: 9 in the training set, 8 in the test set
- The number of components in one sketch in the training and validation subsets are, respectively, 101 and 125 on average
- The range of the resolution is from(1166×1856) to (2162×4160)

TABLE II: Data summary

Dataset	Num of cut	Sketch in cut	Num of component
Traning	152	9 ± 7	101 ± 58
Validating	9	8 ± 3	125 ± 27

Experimental Result

- We evaluate our framework and the GAN-based model proposed in **[15]** on the aforementioned dataset
- Metrics: Accuracy computed on component level and pixel level
- Speed evaluated on 4 CPUs and 16GB of memory, GeForce GTX 1050 Ti GPU

TABLE III: Evaluation of traning dataset

Model	Acc-component	Acc-pixel	Speed
GAN [15]	13.52%±2.48	95.96%±5.32	3.78s±1.92
ResNet-34	56.65%±26.15	95.11%±11.95	3.63s±1.53
Our (0.33, 0.33, 0.33)	62.55%±26.36	96.12%±5.71	9.41s±31.05
Our (0.55, 0.15, 0.30)	63.14%±26.52	96.30%±5.52	9.43s±31.11

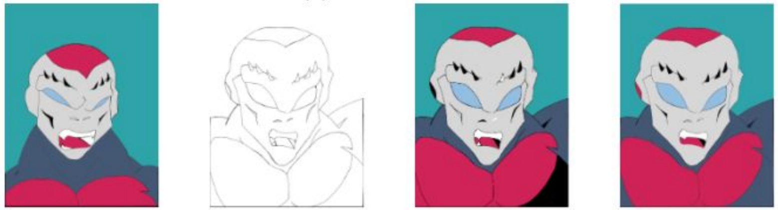
TABLE IV: Evaluation of validating dataset

Model	Acc-component	Acc-pixel	Speed
GAN [15]	8.44%±3.03	89.88%±7.09	3.84s±1.85
ResNet-34	57.48%±20.31	95.42%±4.16	4.80s±1.45
Our (0.33, 0.33, 0.33)	63.88%±18.94	96.49%±3.98	9.25s±20.65
Our (0.55, 0.15, 0.30)	64.63%±20.18	96.39%±4.01	9.56s±20.41

Example output

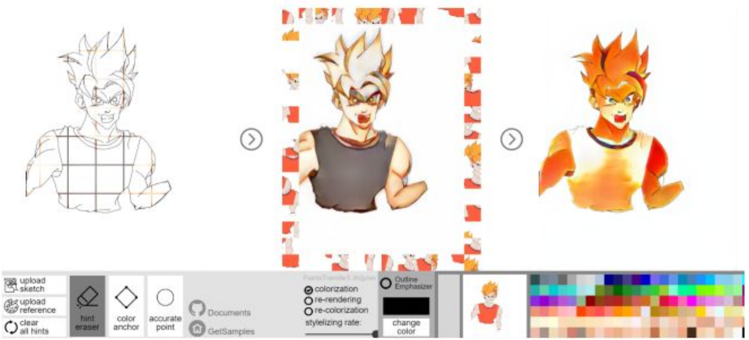
Our model

(a) *Character 1*

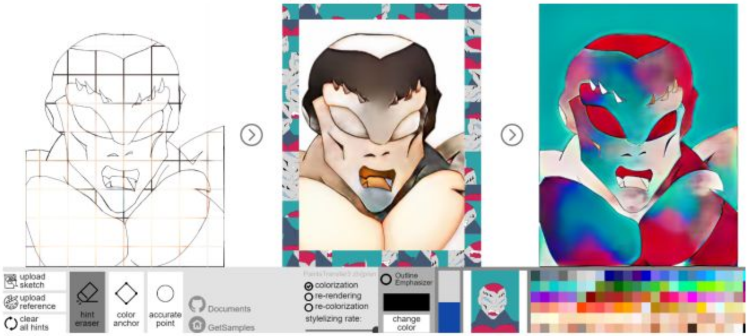


(b) *Character 2*

Style2paints [5]



(a) *Character 1*



(b) *Character 2*

Conclusion

- We introduce a two-stage component-based graph matching algorithm to colorize a series of sketches based on a reference colored image
- After conducting experiments on real-world datasets, we demonstrate that the algorithm can produce high-quality colorized pictures, with high accuracy, good time efficiency and intuitive user experience
- As a result, our method is a promising solution to many high-precision line art colorization tasks in industrial settings
- We will focus our future works on the interpolation of emotions and/or actions of anime characters in a series of scenes to provide more semantic and temporal context for automatic sketch colorization

THANK YOU