INBETWEEN COLORIZATION USING BOTH KEYFRAMES

DAVID PORTILLA OROZCO
Pereira, Colombia

Abstract: inbetween colorization is one of the latest steps to finish an animation, is time consuming and repetitive. In this paper I use a u-net to attempt solve this problem using both keyframes to feed the net insted of one keyframe. so the net can learn easier the pixel movement, when using one reference the net should understand the lineart to do this task and that is way harder. My net has similar results of state of the art and is way pimplier.

Keywords: line art colorization, inbetween colorization

1) INTRODUCTION

In recent years the Japanese animation industry has considerably increased its profitability, reaching a total value of 18.38 billion euros in 2018, a total increase of 0.9% compared to the last year, as indicated by a report from the AJA (Association of Japanese Animation) [1]. The realization of an anime project involves a great investment both financially and labor. It is estimated to complete a season may have a total cost of around 1.6 million euros [3] [5], in addition, being a mostly purely manual work, the production stages usually become a tedious and labor-intensive job. One of these phases is the coloring of inbetweens, which are the transitional sketches found between keyframes, and which are mostly done by novice animators [4]. These 'sketches' in turn comprise two stages; the drawing and its coloring, however, automating this task would save time and money for the production company, thus reducing the long working hours of the animators and allowing the salary increase, which according to data from 'AssianBoss' [2] It is based on contracts of $ 500 per month, with working days of up to 12 hours a day, thus being a much more precarious and lower paid job than a part-time student in a supermarket [2].

Assuming that coloring an inbetween takes between 1 or 2 minutes, in an animation of 300 shots with 10 of these in each one, it would be 3000 in total, which translates into 50 hours of work per animation episode. The animes usually have between 12 or 24 animation episodes per season, thus assuming between 600 or 1200 hours of work respectively, which when governed under the 40 working hours / legal weeks, equals a total of between 15 or 30 weeks of work. According to Wikipedia.org, in 2020, despite the slowdown caused by the global pandemic, 114 animes were broadcast [6].

There are previous works of IAs that manage to color linearts with the help of some reference [7] and even some that solve the same problem previously raised [8] [9] [10], thus demonstrating with interesting results that this automation is possible. This paper will explore the solution to

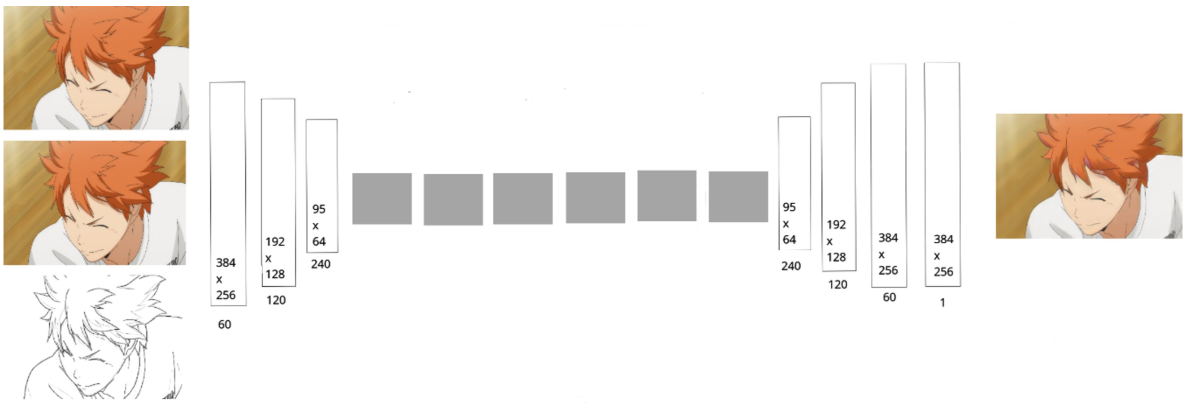

Figure 1: u-net

this problem by developing a simple CNN u-net, avoiding using auto generated frames to avoid amplified errors, using all the available information obtained from two keyframe references instead of just one with a broader dataset in animation styles and preserving the ratio 16: 9.

2) RELATED WORK

2.1) Inbetweens in realistic videos.

The Depth-Aware Video Frame Interpolation paper [11] is based on applying the AI Optical Flow [13], which creates inbtweens of real-world videos, transforming 60 FPS videos from references at 30 FPS. However, this approach does not have satisfactory results in our field since the assumptions that govern movement in reality are not usually fulfilled in the world of animation. For example, unlike reality, in animation objects usually have a certain deformation when moving since these movements do not necessarily correspond to the expected one governed by the laws of physics and their interpolation is different. On the other hand, in the real world time and the image are related, that is, always for a certain unit of time a certain image is had, while in the world of animation the inbetweens are usually grouped to accentuate the sensation of movement [12] [16].

Or like the paper Deep Line Art Video Colorization with a Few References [9] explains in its introduction:

> Coloring line art videos based on given reference images is challenging. Firstly, unlike real life videos, animation videos do not hold pixel-wise continuity between successive frames. For example, lines corresponding to limbs of an animated character may jump from one shape to another to depict fast motion. As such temporal continuity cannot be directly used to guide the learning process to maintain the color consistency between regions in adjacent frames. (p.1)

Furthermore, in the paper [11] an attempt is made to generate inbetweens when the only 'input' or information supplied is the realistic video keyframes, while for our animation problem we have both the keyframes and the lineart or 'douga', which provides us with information about what will happen between the keyframes.

2.2) Video colorization.

As previously stated, there are other AIs that have already tried to solve the problem, these AIs are: TCVC [8], Line Art Correlation Matching Feature Transfer Network for Automatic Animation Colorization [10] and Deep Line Art Video Colorization with a Few References [9].

TCVC [8] uses as input some reference given by the user and from the result the next frame will be generated, doing this consecutively with the rest of inbtweens. However, this means that, if an error occurs in any of the outputs, when this is the new reference, a 'Snow Ball' effect will be generated and the initial error will be amplified.

The AI [10] uses a single output to color the following reference with a more sophisticated architecture that seeks to relate the characteristics between one frame and another, this being the state of the art to date. However, both the database and the neural network are private code, thus making it impossible to verify their results or a fair comparison.

Finally, in [9] they try to solve the problem with a GAN (Generative Adversarial Network) and providing the residual blocks with additional information with an auxiliary network, and after this a second color coherence network.

3) METHOD

What the previous approaches have in common is that they use a single reference to make the inbetweens that follow it, but taking advantage of the fact that the inbetweens are between keyframes we can assume that these will be full colored when it is in the final phase of the animation.

In addition, certain peculiarities of this problem must be taken into account, such as that the images are paired, that is to say that for each lineart corresponds a color image, and these paired images can be generated in a large quantity, since a single chapter animation has about 15000 unique frames.

The other factor is that the images we want to generate are in sequence and are related to each other in each shot, so as the neural network updates their weights in each batch, I made the animation shots have the length of the batch size, and so on Each shot in the dataset is 8 frames, where the first and the last are used as reference, so there are 6 left, which are the inbetweens and this is the one corresponding to the batch size.

To generate the images use a 'u-net', activation relu, padding = "same", with a 'kernel' (3,3), in the convolutional layers and 6 residual blocks [14] and 'instance normalitation' as shown see in Figure 1, for the 'loss function' use mean absolute error, comparing the generated inbetweens with the real ones.

Finally, each color channel is processed individually and put together at the end. This is possible since the anime has very simple textures. Figure 2.

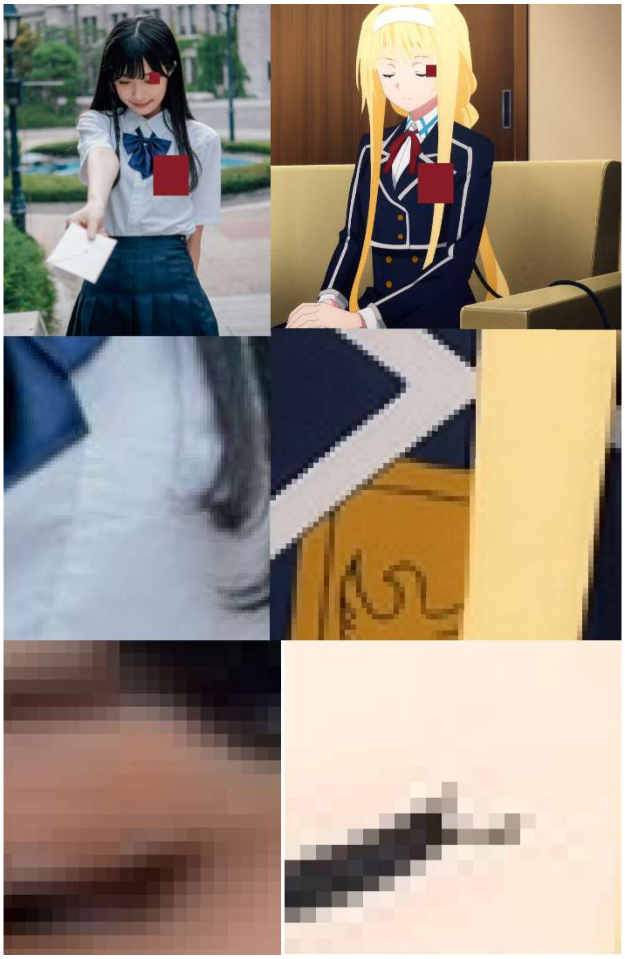
Figure 2. Comparison between textures[26][27]

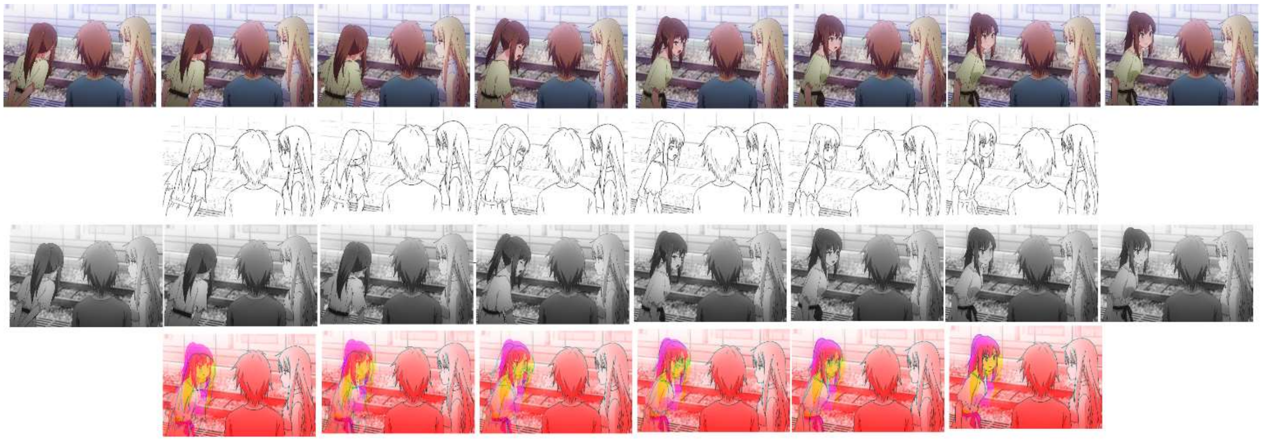

Figure 3: another example using a complete shot, the first and last frames are used as keyframes, in the first row is the real shot, in the second the lineart generated in sketchkeras in the third transformed to black and white, and the fourth is the input that is generated, the black and white boxes that are not keyframes are the ones that will be used as a target in the training phase.

3.1)Data preparation

Given the difficulty of finding a large volume of colored sequence data with their respective douga,I generate this data from the following animations:

- Nisekoi (2014)
- Angel Beats (2010)
- Tonari no Aibotsu Kun (2012)
- Genkan Shojo nozakikun (2011)
- Kokoro Connect (2010)
- Danshi Koukousei no Nichijou (2012)
- Boku no Hero Academia 3rd Season (2018)
- Kaichou wa Maid Sama (2010)
- Toradora (2008)
- Sakuraso no Pet na Kanojo (2012)
- Melancholy of Haruhi Suzumiya (2006)
- Hyuoka (2012)

Use 6 chapters of each for a total of 72 episodes, then use scenedetect [15] to separate animation shots, between 200-400 per episode. In 'shonen' animes (action animes for teenagers) the precision of scenedetec decreases in scenes with a lot of action or a lot of movement, that's why mostly I used school animes (animes that relate the day to day of adolescence in Japan one of the most popular genres currently), then each video extracted images in 2s in groups of 8 (in 1s it means that there are 24 drawings per second, in two it means that there are 12 drawings per second, that is to say that of every 16 frames I take 8 images) for two main reasons. The first is that 12 animation frames per second is the most common way to animate and thus I avoid using repeated frames. The second is to increase the variation of the inbetweens

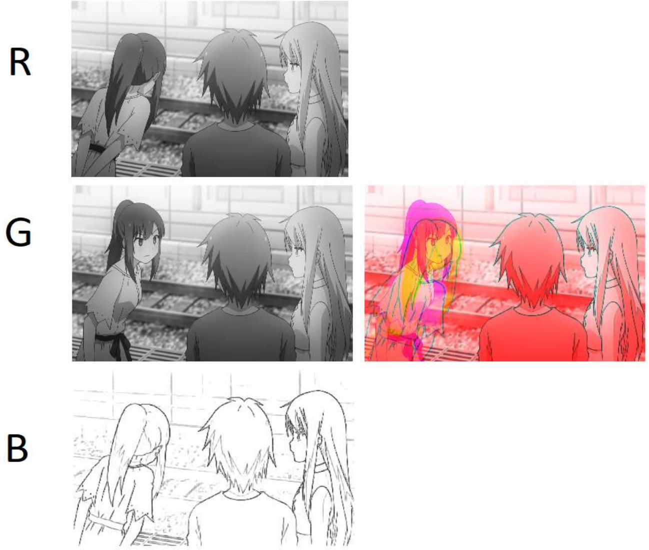

Figure 4: an example of how each of the training images is created, on the left the input separated by channels, and on the right the final result

Then save the images in 256 × 384 pixels to preserve the 16:9 ratio with a total of 839,416 images to filter and just stay with the best which had greater range of motion in the inbetween. In each sequence it took the first frame and compared it with the inbetween using mean absolute error, did the same with the second keyframe and calculated the average of this error, and did the same with each of the 6 inbetweens. If the average error of these 6 inbetweens less than 10, this batch was discarded.

This process left me with a total of 250,000 images in the absence of the last filter, since there is a type of sequence where the camera moves but the image is static and does not contain inbetween, therefore I deleted those sequences manually, thus remaining with 79 048 images, now create a data type for this network, because to train it I had an NVIDIA GTX 960 with 4 GB of VRAM and doing full colored give me out of memory, so I transformed the keyframes to black and white, and the inbetweens to linearts with sketchkeras [17], in each image in the channel RED = keyframe1, GREEN = keyframe2, BLUE = lineart [figure 3 and 4] for a total of 58 286 images.

For validation data filter 1 chapter of Akatsuki no Yona (2014), Kobato (2009) and Love is War (2019).

4) EXPERIMENTS.

Unlike the other nets mentioned above, where they used for testing series that were in the training data, so they kept the same drawing and animation style, in my project, to test the network, I used sequence completely new animations to those used for training. I separated them into 8 or 16 frame shots, and I did the whole process to simulate what a real life scenario would be like. Some animation shots do not contain inbetweens or were removed and added at the end for ease of editing. An example of this is illustrated in Figure 4.

After this, I converted it back to 24 frames / second video, in order to better see the final result as a serial, the qualitative results. They can be seen in this video [23], and to see the network work in a broader spectrum of situations use different sequences of short animations and compile them in the following video [24]

From this second video calculate quantitative results that can be seen in the following table, using PSNR and SSIM as metric [18].

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6
PSNR ↑	27.21	26.44	26.17	26.20	26.49	27.45
SSIM ↑	0.8661	0.8571	0.8548	0.8551	0.8572	0.8688

Table 1. PSNR / SSIM result of frame.

5) CONCLUSIONS

The effectiveness of the network is still far from being indistinguishable from the real inbetween, but under certain circumstances it could go unnoticed. However, it is not yet suitable for commercial use, which is a bit disconcerting in the seemingly simple task.

An explanation for this low performance is the quality of the data that, as there is no paired public dataset of original and colored dougas without a background, 2 problems occur, the first is that the douga generated in sketchkeras does not have as much information as a douga original. On the other hand, the background that is added after coloring the douga distracts the learning of the neural network, because to decrease the loss function, the network has to minimize the error of the entire background image included, so it spends part of its learning ability in this task and not only that, but the value of the loss function is wrong for our purpose since we want the error to be only the color of the douga and not the color of the background.

Examples of real 'dougas' [18] [19] [20] [21] [22]

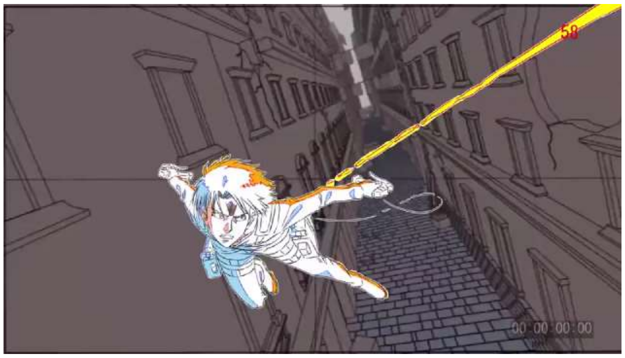

Figure 6: douga from shingeki no kyoujin.

REFERENCES

[1] El Palomitrón. 2021. *Análisis de la industria del anime en Japón en 2019 - El Palomitrón*. [online] Available at: <https://elpalomitron.com/analisis-industria-anime-japon-2019/> [Accessed 5 February 2021].

[2]2019. [video] Available at: <https://www.youtube.com/watch?v=NsnwA9IrqR0> [Accessed 12 February 2021]. minute 7:13

[3]Anime News Network. 2021. *Anime Insiders Share How Much Producing a Season Costs*. [online] Available at: <https://www.animenewsnetwork.com/interest/2015-08-13/anime-insiders-share-how-much-producing-a-season-costs/.91536> [Accessed 12 February 2021]. February 2020

[4]2018. [video] Available at: <https://www.youtube. com / watch? v = Un9_xitSseI> [Accessed 12 February 2021]. minute 6:24, explanation of how the inbetweens coloring workflow is minute 22:00

[5] 2020. [video] Available at: <https://www.youtube.com/watch?v=WEKpdDkn4xk> [Accessed 12 February 2021].

[6] Es.wikipedia.org. 2020. *Anime de 2020*. [online] Available at: <https://es.wikipedia.org/wiki/Categor%C3%ADa:Anime_de_2020> [Accessed 12 February 2021].

[7] lllyasviel. style2paints. https://github.com/lllyasviel/style2paints.

[8]Harrish Thasarathan, Kamyar Nazeri, and Mehran Ebrahimi. Automatictemporally coherent video colorization.CoRR, abs/1904.09527, 2020.

[9] Min Shi and Jia-Qi Zhang and Shu-Yu Chen and Lin Gao and Yu-Kun Lai and Fang-Lue Zhang (2020) Deep Line Art Video Colorization with a Few References arXiv:2003.10685

[10]Zhang Qian and Wang Bo and Wen Wei and Li Hai and Liu Jun Hui (2020) Line Art Correlation Matching Feature Transfer Network for Automatic Animation Colorization arXiv:2004.06718

[11] Wenbo Bao and Wei-Sheng Lai and Chao Ma and Xiaoyun Zhang and Zhiyong Gao and Ming-Hsuan Yang (2019) Depth-Aware Video Frame Interpolation arXiv:1904.00830

[12] n.d. *inbetweening explination, in the anime shiro bako*. [video] Available at: <https://streamable.com/qe8sh> [Accessed 12 February 2021].

[13] Deqing Sun and Xiaodong Yang and Ming-Yu Liu and Jan Kautz PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume arXiv:1709.02371

[14]Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun (2015) Deep Residual Learning for Image Recognition https://arxiv.org/pdf/1512.03385.pdf

[15]GitHub. 2021. *Breakthrough/PySceneDetect*. [online] Available at: <https://github.com/Breakthrough/PySceneDetect> [Accessed 12 February 2021].

[16] APLattanzi, 2019. *AKIRA: The 24 Frames-Per-Second Myth*. [video] Available at: <https://www.youtube.com/watch?v=YtYpif-dLjI> [Accessed 12 February 2021]. because there are 1s and 2s minute 3:03 spacing minute 4:49

[17]lllyasviel. sketchkeras. https://github.com/lllyasviel/sketchKeras, 2018.

[18] Cvnote.ddlee.cc. 2019. *PSNR and SSIM Metric: Python Implementation - CV Notes*. [online] Available at: <https://cvnote.ddlee.cc/2019/09/12/psnr-ssim-python> [Accessed 12 February 2021].

[19] 2019. *Animation Education - What is a "Douga"?*. [video] Available at: <https://www.youtube.com/watch?v=IrmZIc7sDv4> [Accessed 12 February 2021]. minute 1:36

[20] 2020. *Que es el Genga?*. [video] Available at: <https://www.youtube.com/watch?v=HD-2lKv1M0M> [Accessed 12 February 2021]. minute 0:20.

[21] 2019. *Key animation: Levi vs Kenny scene anime Attack Titan - animation Arifumi Imai*. [video] Available at: <https://www.youtube.com/watch?v=4TyYKZmcx6c> [Accessed 12 February 2021].

[22] 2019. [video] Available at: <https://twitter.com/WIT_STUDIO/status/1205449789040672772> [Accessed 12 February 2021].

[23] 2019. *View from the Summit | Haikyu!!*. [video] Available at: <https://www.youtube.com/watch?v=qIdm1GFImRQ> [Accessed 12 February 2021].

[24] 2021[video] Available at: <https://drive.google.com/drive/folders/1PfLBpxPzkipXkPuRJZISHBfmKKFxQPc9?usp=sharing>

[25] 2019. Así se hace Dr. STONE | La creación de un anime [video] Available at: <https://www.youtube.com/watch?v=10hyFva4VnU&ab_channel=CrunchyrollenEspa%C3%B1ol>[Accessed 12 February 2021] minute 7:50.

[26] [image] Available at: <https://i.pinimg.com/564x/e5/39/17/e5391777b428b40270426388e3c1b3cd.jpg>

[27]image from the anime sword art online [image] Available at: <https://animesolution.com/wp-content/uploads/2020/09/Sword-Art-Online-Alicization-War-of-Underworld-23_01.33_2020.09.19_13.05.36_stitch.jpg>