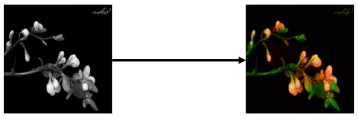
Real Time Video Colorization Using Color Priors

Sneha Venkataramana

Goal

My motivation is to enable colorizing black and white videos in a way that the user has a say in the process. This will be done by letting the user manually provide a color scheme (as detailed as they wish) and then letting my model do the work of transferring the scheme to the video. The model will be in such a way that it operates even with no user input. In addition to colorization, I also wish to provide the user the ability to style the video to their wish.



Poster Walkthrough :
https://youtu.be/68ObBN6_XrE

Data

- 50,000 images of size 32x32 in the CIFAR-10 dataset
  - Used in the initial stages in order to validate the method
  - Model trained here is hard to evaluate because of image size
- 50,000 images of size 256x256 in the Imagenet dataset

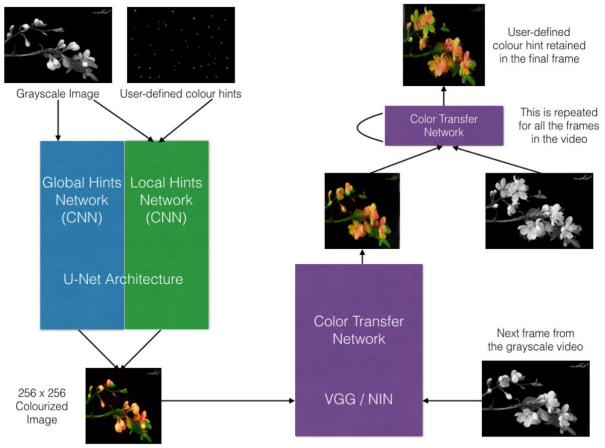
References

Real-Time User-Guided Image Colorization with Learned Deep Priors (2017) - Zhang, Richard and Zhu, Jun-Yan and Isola, Phillip and Geng, Xinyang and Lin, Angela S and Yu, Tianhe and Efros, Alexei A

Colorful Image Colorization (2016) - Zhang, Richard and Isola, Phillip and Efros, Alexei A

A Neural Algorithm of Artistic Style (2015) - Leon A. Gatys and Alexander S. Ecker and Matthias Bethge

Models



- A two-tower U-net model for colorization and VGG-based model for color transfer was used
- Used squeeze convolution filters in the colorization model to reduce the number of parameters
- I modified the loss function on the color transfer network by adding an additional regularization term that helped generalizing the color scheme
- Final loss function: $\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2}\sum_{i,j}(F^l_{ij} - P^l_{ij})^2 + |F^l_{ij} - P^l_{ij}|$, $\mathcal{L}_{style}(\vec{a}, \vec{x}) = \frac{1}{2}\sum_{l=0}^{L} w_l E_l$

Features

- My input image is re-sized to 256x256 to fit memory constraints
- The other input is a user-given hint, which is converted to a 256x256 RGB image
- The network represents the 3-channel hint in 2-channels
- Ground truth is a color image which was de-colorized for training

Results

The results were evaluated objectively using PSNR and qualitatively as well. All the variations above were tested with 15-20 points of color hints.

Method	Avg. PSNR
Only Colorization	11.46 ± 0.35
Colorization + Transfer	12.57 ± 0.13
(2) + loss fn. change	13.04 ± 0.18
(3) + param compression	12.55 ± 0.24

Overall, the third method that combined both models was both qualitatively and from a metric perspective, the best candidate.

Discussion

I tried multiple variations of combining the two models:

1. Only colorization model
2. Colorization model and the color transfer model with no changes
3. Colorization model and the color transfer model with updated loss function
4. Colorization model with reduced parameters and the color transfer model with updated loss function

Approach (1) had the drawback that the color hints for the first frame may not be applicable to the 100th frame. Approach (2) was quite improved, but could have been better on relaxing the weights to reference image. Approach (3) worked better on generalizing the color scheme between frames. The final variation (4) was a test to see if reducing the parameters in the network had a strong effect on the outcome - the motivation here is that the color transfer is rather slow and could be perhaps sped up - this had reasonable results as the difference in the quality of the generated images was low (<10% PSNR loss).

Future Work

I would like to explore the work of He, et.al. on Deep Exemplar based Colorization where this network is augmented with a similarity-sub net which showed very realistic colorizations in their paper. This would be done by modifying the colorization network to have an additional parallel path.

Another interesting problem I would have liked to work on, is to train this network end-end instead of two separate networks. This could be done by formulating a joint loss function for the learning step after connecting the two networks, where we try to minimize (i) the color difference between two adjacent frames, (ii) the distance from the best predicted pixel color and the given user color.