CS 230: Consistent Video Colorization

Gael Colas (colasg), Kevin Lee (kelelee), Rafael Rafailov (rafailov)

Objectives

The goal of our project is to create **consistent colorization of grayscale videos**. This entails two challenges:

- Create **plausible coloring** of each frame
- Ensure **color consistency** between frames

There are existing deep learning architectures, both classical and generative, that tackles the issue of plausible image colorization. But until recently, little work has been done on color propagation using deep learning methods. This is what our project focuses on.

Model

We deploy a **hybrid supervised/C-GAN model**.

- **Generator**
  The conditioning set for the generator for frame t would be a **colored frame c_{t-1} from $t-1$ and the grayscale frame g_t from t**. The goal is to generate a colorized version $\hat{c}_t$ from g_t using the colors of c_{t-1}. The objectives of the generator is given by:

$$\min_{\theta_G} -\alpha \mathbb{E}_{\mathbf{z}}[\log D(G(\mathbf{0}_z|[c_{t-1}, g_t])|c_{t-1})] + \beta||G(\mathbf{0}_z|[c_{t-1}, g_t]) - c_t||_1$$

  We model $\mathbf{z}$ as having zero variance, when conditioned on c_{t-1} and g_t, as we do not want to introduce variability in the coloring. Here α and β are parameter weights between the objectives of the generator network. For we trained two models: $\alpha = 0$, which corresponds to a fully supervised approach and $\alpha = 1$, $\beta = 100$, which corresponds to a hybrid model.

- **Discriminator**
  The discriminator conditioning set consist of c_{t-1}. The model then tries to distinguish between real and fake colorized next frames. Mathematically, the objective is:

$$\max_{\theta_D} \mathbb{E}_{c_t}[\log D(c_t|c_{t-1})] + \mathbb{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{0}_z|[c_{t-1}, g_t])|c_{t-1}))]$$

- **Architecture**
  Our generator model architecture is based around **U-net** and is presented in Figure 1. Each layer in the convolutional part is followed by batch normalization and a Leaky ReLU activation. Each layer in the de-convolutional part is followed by batch normalization and ReLU activation. The discriminator applies a series of convolutions followed by batch normalization. We modify the architectures presented in [1] to consider the expanded conditioning sets.

Figure 1: Generator architecture

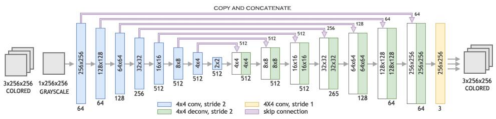

Figure 2: Discriminator architecture

*Figures adapted from [1]

Samples

Figure 3: Comparison of Video Colorizations. From top row to bottom: (a) grayscale ground truth, (b) color ground truth, (c) baseline frame-by-frame colorization, (d) supervised model color propagation, (e) GAN colorization from ground truth reference, (f) GAN color propagation (trained with $dt = 1$), (g) GAN color propagation (trained with $dt = 5$)

Training and Results

- **Training**
  Our original **dataset** was composed of **pairs of consecutive frames extracted from videos**. Our second dataset was composed of frames separated by $dt = 5$. The goal was to prevent the model from directly copying the previous frame colorization. Both models were trained on videos from the 'Baking' category of the Moments in Time dataset, which after processing consisted of about **7,000 samples**. We trained for **10 epochs**.
- **Quantitative evaluation**

Model		sup Gen	cGAN ($dt = 1$)	cGAN ($dt = 5$)
Pixel Accuracy (%)	Train	94	92	84
	Test	78	83	79

Table 1: Pixel Accuracy comparison between the supervised model and different Conditional GAN models

It is important to precise that this was evaluated on the 'color transfer' task: how well the algorithm colorizes a grayscale image given the previous ground truth color frame. These metrics quickly decline when considered over an entire video.

- **Qualitative evaluation**
  Based on samples generated from our test dataset it is hard to distinguish frames that have been colorized from the ground truth, the exception being frames with change in scenery. However, the coloring quickly collapses in videos due to **exponential error propagation**.

Conclusion and next steps

We have trained our model and achieved good color transfer between frames, however we found that the coloring quickly collapses in videos due to exponential error propagation. In order to address this challenge we plan to implement architectures with recurrent convolutional structure, which would allow us to work with entire sequences of images at each pass.

References

[1] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi.
Image colorization using generative adversarial networks.
In *International Conference on Articulated Motion and Deformable Objects*, pages 85–94. Springer, 2018.
https://github.com/ImagingLab/Colorizing-with-GANs.

[2] Richard Zhang, Phillip Isola, and Alexei Efros.
Colorful image colorization.
9907:649–666, 10 2016.
https://github.com/richzhang/colorization.

Codebase

- GitHub: https://github.com/ColasGael/Automatic-Video-Colorization