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 tackle 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/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 colored version $c_t$ from $g_t$ using the colors of $c_{t-1}$. The objectives of the generator is given by:

  $$\min_{G} - \alpha \mathbb{E}_{D \sim q_D}[D(G(0, (c_{t-1}, g_t))) + \beta (D(G(0, (c_{t-1}, g_t))) - \epsilon)]$$

  We model $\alpha$ as having zero variance, which conditioned on $c_{t-1}$ and $g_t$, as we do not want to introduce variability in the coloring. Here $\alpha$ and $\beta$ 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 colored next frames. Mathematically, the objective is:

  $$\max_{D} \mathbb{E}_{c \sim q_c}[D(c_{t-1})] + \mathbb{E}_{D \sim q_D}[1 - D(G(0, (c_{t-1}, g_t)))]$$

  *Figures adapted from [1]*

Sample Architecture

![Sample Architecture](image)

*Figure 1: Generator architecture*

*Figure 2: Discriminator architecture*

Comparison of Video Colorizations

![Comparison of Video Colorizations](image)

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

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 $\Delta t = 5$. The goal was to prevent the model from directly copying the previous frame colorization. Both models were trained on videos from the "Biking" category of the Moments in Time dataset, which after processing consisted of about 7,000 samples. We trained for 10 epochs.

- **Quantitative evaluation**
<table>
<thead>
<tr>
<th>Model</th>
<th>Test Gen/GAN ($\alpha = 1$)</th>
<th>Test GAN ($\alpha = 5$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Ac.</td>
<td>79</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Test Gen/GAN ($\alpha = 1$)</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Test GAN ($\alpha = 5$)</td>
<td>79</td>
</tr>
</tbody>
</table>

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 colors a grayscale image given the previous ground truth color frame. These metrics quickly decline when considering 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


Codebase

- GitHub: https://github.com/ColorAd/GAN/Automatic-Video-Colorization