Automatic Video Colorization with Deep learning

Simran Pabla and Haripriya Mehta

Abstract

Colorization of classic black and white films is a frequent topic in Hollywood, Bollywood, and other major film industries. Historically, colorization has often done by hand, and more recently, it has been done with the help of digital image processing. However, both processes can take several years, millions of dollars and hundreds of artists and engineers to work on the project. We present an automated method of video colorization via deep learning that builds upon recent work of image colorization by Melas-Kyriazi et al [1] which has been inspired by work of Iizuka et al [2], Zhang et al [3], and Larsson et al [4] in constructing our network. We propose a specific learning rate, a hand-curated dataset and a novel loss function which takes into account the sequential nature of video frames in colorizing a video. Sample film clip colorized with our model can be viewed here.

1. Introduction

With the advent of deep learning neural network, many researchers have sought to colorize images entirely without the aid of humans. Our goal is to investigate whether deep learning can colorize black and white videos in an efficient and more cost-effective manner. We specifically focus on colorizing 30 seconds from the first Bollywood movie to have been colorized, the hit film, Mughal-E-Azam, which was re-released in 2005.

The process of colorizing for the 2005 version of the film, which is similar to many present-day colorization techniques for historical films, involved digitally restoring the original negatives, consulting historians for the color of attire worn during the Mughal period, consulting the actors of the film, assembling a hundred-member team of artists and software engineers to develop software that colorized frames by accepting colors whose hues would match the shade of gray present in the original film, and verifying their work by comparing it to the colors of original costumes worn on the sets. It took more than two years to complete and estimated costs for the project ranged from $280,000 to $1.4 million. Even after undertaking such an arduous task, some critics complained that the colors were psychedelic and unnatural.

We were interested in exploring deep learning alternatives to the laborious, extensive process of colorization. The film industry has already begun to explore deep learning alternatives to their existing digital colorization processes. A film like Mughal-E-Azam, however, poses great complexity. The mirror and light effects in each frame and the sheer detail in the jewelry, clothing, and makeup of each character add layers of difficulty to an already challenging task. Additionally, having been filmed several decades ago, the movie has noisy artifacts that modern films and images lack.

Our approach is based on existing image colorization models and modifying the loss function of the model in such a way that it accounts for the sequential nature of frames in a film. Additionally, we generate a training dataset that includes examples more similar to the video we are trying to colorize than the Places dataset alone would hold. The pre-trained model that we use as an initialization point is pre-trained on the Places dataset, which would not be as intelligent and trained on the kinds of objects and settings that our input contains. Finally, we modified learning rate based on empirical analysis.

2. Related Work

Much work has been done in the field of image colorization. The four critical papers that have driven our research are Iizuka et al, Zhang et al, Larsson et al, and Melas-Kyriazi et al. Each of these papers define unique approaches for image colorization, though there are similarities in how they define the problem. Each of these papers, for example, work in the Lab colorspace instead of the RGB colorspace, where L represents lightness, a represents green-red components and b represents blue-yellow components. The Lab colorspace offers an easier way to separate the grayscale image from the color channels.

More specifically, Iizuka et al. presents a CNN-based network that extracts local low-level, mid-level, and global features from images and fuses this information to colorize images. The global features are derived from classification labels and the local features are derived from small image patches.

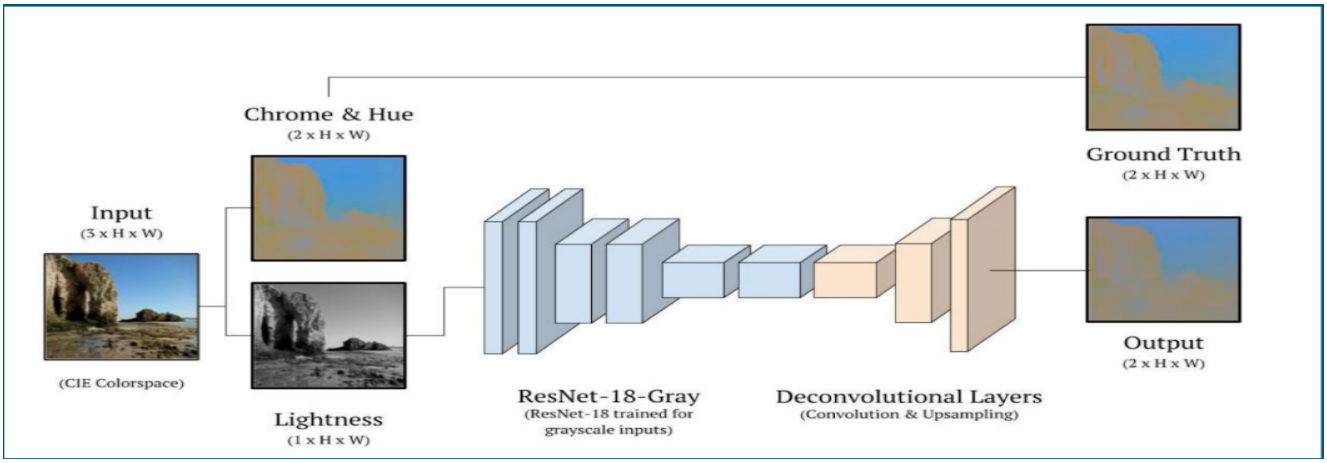

Figure 1. Melas-Kyriazi model. Image is broken into two tensors - one corresponding to L channel (lightness) and a,b channel (green-red and blue-yellow color components). The grayscale "lightness" component is inputted into the ResNet Gray, which outputs a distribution of colors. The output of the model is compared with the original a,b channel tensor. Image borrowed from Melas-Kyriazi's paper.

Larsson et al. explores a variant of Iizuka's approach by utilizing hypercolumns to predict pixel distributions. They modify the traditional VGG network to ensure that it can take 1-channel grayscale images as input and output a distribution over a set of color bins for each pixel in the grayscale image. Unlike Iizuka's approach, they take a continuum of features into account and initialize, but do not continue to use the model in a classification-based manner. As a result, Larsson is not restricted to labeled data.

Zhang et al. represents the image colorization problem as a classification task and utilizes an off-the-shelf VGG network to output a distribution over a set of color bins as Larsson does. Their primary contribution lies in their work on effective loss functions. Due to the multi-modal nature of color prediction, they propose predicting a distribution for each pixel in the grayscale image and utilizing cross-entropy loss to train their network. In order to preserve color vibrancy, they take an annealed mean of the color distribution.

We reasoned that image colorization can be applied to film colorization, and defined our process to involve colorizing individual frames and stitching them together into continuous film.

We began by experimenting with Iizuka's implementation and found that one major advantage was its lack of fully connected layers, which allowed the output to retain the input size was. However, the implementation's training process was reliant on a labeled dataset. If we wanted to train on Bollywood movies that were similar to Mughal-E-Azam, we would be using unlabeled data. As we did not use a labeled dataset, training this implementation from scratch on our data generated poor results.

After producing insufficient results with Iizuka's implementation, we pivoted and took inspiration from Melas-Kyriazi et al. Their approach takes elements of all three papers to produce a new ResNet-based architecture that ultimately proved to be more effective for our problem space. Similar to Larsson's gray-modified VGG network and Zhang's VGG network, Melas-Kyriazi et al. utilizes ResNet-Gray, a modified ResNet architecture that can take grayscale images as input and does not utilize a classification-based network. Like Iizuka's network, Melas-Kyriazi's network examines local features. However, Melas-Kyriazi's network does not examine global features and thus, is not a classification-based network. His network also does not use fully-connected layers, and thus, crops the output.

Additionally, his model is meant for image colorization and not video colorization; thus, when using his model to colorize videos frame by frame, there is a lot of flickering and color inconsistency. One important benefit to using his model was that it was already pretrained on the MIT Places-365 Dataset for 80 epochs.

3. Model

We use the Melas-Kyriazi model as a starting point for our architecture. The model accepts a colored image as input, which may be scaled and center-cropped if necessary. It is then split into a tensor that corresponds to the L channel and a tensor that corresponds to the a,b, channel. The L 1-channel tensor, which is a grayscale image, is fed into a ResNet-Gray model, which is the first six layers of ResNet-18 model pretrained on grayscale images and adapted to take 1-channel images as input. After obtaining the local features from the image through the ResNet-Gray model, the image is upsampled and returns to the cropped inputted size. The predicted output of this model is then compared to the a,b tensor.

Figure 2. Example of frames from hand-curated dataset.

We build upon this model by generating a more suitable dataset for our use case, experimenting with hyperparameter tuning, and reducing error by developing an optimal loss function. These contributions are described in the experiments section.

3.1. Dataset

Continuing training on Places dataset, for colorizing Mughal-E-Azam clips would not work as Mughal-E-Azam featured different scenery than the Places dataset and humans. We extended the pretrained model by further training it on a hand-picked set of Bollywood videos that were similar to Mughal-E-Azam in terms of vibrancy and glitter. We aimed to select royal settings which feature the insides of palaces, as the Places dataset lacked this kind of data. We omitted any videos that featured royal settings, but were overall "brownish" or "maroonish" in color as using a mean squared error loss results in dull hues (explained further under 4.2.1) and we did not want to emphasize those colors. Additionally, these Bollywood clips include people, which the Places dataset severely lacks.

To introduce enough variety into our dataset, we decided to only include every fifth frame in our clips. There was little change from frame to frame in the 24 fps clips, so this filtration process allowed more variety in our dataset. Our dataset eventually consisted of over 60,000 frames from Bollywood movies such as Devdas, Prem Ratan Dhan Payo and Jodha Akbar.

Interestingly, while we initially expected that a model solely trained on the Bollywood movie dataset we generated would produce the best results, we were surprised to find that a combination of the Places dataset and the Bollywood dataset was more effective at colorizing the images correctly. We're still not certain why including Places dataset improved the colorization, but we speculate that the model may have benefited from the added diversity with respect to color and scenery.

4. Experiments

Because we don't have an accurate colored example of this film to compare the images against for accuracy analysis, we analyzed the results of our analysis both visually and based on the loss values, as will be shown in each of the experiments below.

Figure 3. From Left to Right Learning Rates 0.1, 0.001, 0.0001. Learning Rate of 0.001 does not have a blue nose.

4.1. Learning rates

We trained on several learning rates and discovered that different rates reduced the error significantly. As is evident in Figure 3, the blue hues that are present on the woman's face reduce significantly when using a learning rate of 0.0001 and nearly vanish when using 0.001. The blue is likely introduced to the image due to the Places dataset, and the optimal learning rate seems to decrease the impact of this dataset and place more emphasis on the skin tones and other details that are heavily represented in the Bollywood dataset. Visually, the results are far superior at learning rates of 0.001 and 0.0001; empirically, however, we found that the loss values did not vary much after 99 epochs of training for each of these models. The learning rates 0.001 and 0.0001 produced loss values ranging from 0.0530 to 0.1214, while the learning rate 0.1 produced loss values ranging from 0.0557 to 0.1217.

4.2. Loss Functions

While previous papers have focused on crafting a loss function that reduces error for colorization for a particular image, video presents an additional dimension for calculating loss. In addition to error calculation for each individual frame, error can be tracked between frames as well.

4.2.1 Individual Loss

We began by exploring loss functions for individual frames. Common loss functions like mean square error pose a challenge due to the multi-modal nature of color distributions.

Figure 4. Left: Output of MSE-trained model. Right: Output of MAE-trained model. Right image has less bleeding and purer colors – ex: feather on woman's head is colored pure white rather than a light reddish-pink.

They heavily penalize mismatched colors without considering context. This is problematic, as we don't have enough information when expanding from one channel (grayscale) to 3 channels – when colorizing a dress, for example, there are several valid colorizations, but only one present in the ground truth image. Penalizing this kind of error across several images with different dresses can cause the model to converge towards a muted brown shade for most objects and scenes.

Melas-Kyriazi et al utilizes mean square error (MSE) despite these concerns, noting that it produced sufficiently vibrant results for their images. We tested different loss functions to determine whether or not MSE would be the optimal solution in our case, as we noted brownish tones in our images that could ideally be substituted by more vibrant colors.

We first tested a mean absolute error loss (MAE) function in place of the MSE. MAE did offer slight improvements, as we found that the colors bled less and were far more pure. Unlike those produced by the MSE-trained model, the images produced by the MAE-trained model are cleaner and, as a result, slightly more vibrant. As can be seen in Figure 4, the white pillars and the feather in the woman's hat aren't tinged with the reddish-pink hues that are present in the MSE-trained model output.

While the MAE loss function penalizes less heavily than MSE, we were interested in investigating other options that might further reduce the muted brown shades in our output images. While not much work has been done in this space as of yet, Fenu et al. [5] proposes a loss function that calculates the error corresponding with pixel color as well as error stemming from colorspace variance.

$$L(Y, Y') = \frac{1}{HW} \sum |Y_{ij} - Y'_{ij}| + \sum |(Y_{ij} - X_{ij})^2 - (Y'_{ij} - X_{ij})^2|$$

The first term in this loss function is simply MAE loss, and the second term refers to colorspace variance. Y refers to the ground truth image, Y' is the predicted image, and X is the grayscale image. This expression normalizes the

Figure 5. From left to right: MSE, MAE, Hybrid. The hybrid model's output clearly has reduced bleeding and cleaner boundaries.

ground truth and predicted pixel values by subtracting their corresponding grayscale pixel values. Ideally, we would have tested different weights for each of these terms to determine the optimal balance between colorspace variation loss and pixel color loss for our hybrid loss function, but with limited time, we chose to not weight these expressions differently.

Applying this hybrid loss function in place of the more traditional MSE or MAE improved our results, as can be seen in Figure 5. In addition to reducing bleeding as MAE did, this hybrid function produces images with a more diverse set of colors. Several similar, yet distinct colors are able to exist next to one another in the images rather than one color being picked for a large chunk of the image. Empirically, we found that the loss values for this hybrid function ranged from 0.0496 to 0.1290, which provides more potential for more accurate results, whereas the MSE and MAE loss function did not produce losses below 0.0550. As a result, we chose to utilize this hybrid loss function in our final model.

4.2.2 Sequential Loss

When tracking loss between frames, we are primarily looking to reduce the "flickering effect", which often occurs when images separately colorized are stitched together. An inconsistency in color between frames causes flickering to occur when these frames are played continuously. To account for this, we decide to implement the LRCN (Long term Recurrent Convolution Networks) by Donahue et al [6] from scratch. Utilizing a LRCN for colorizing videos was discussed in Sahay et al. [7] paper's, but it was unclear

Figure 6. From left to right: Original Black and White, 2005 Colorized Version and Our Output.

how they exactly constructed the LSTM and their loss function. We fed six consecutive output frames for our existing model into an LSTM and used MSE on the output of our original model put through the LSTM and a-b tensor put through the LSTM. Unfortunately, when we trained for a few epochs, we obtained pink images that washed out the original image.

We decided to try something more intuitive as we could not find any other research papers or implementations regarding LRCN and video colorization. As we wanted to minimize variation from frame to frame, we wrote a loss function in which consecutive frames were subtracted from each other. We added it to our original loss function and gave the original loss more weight (0.9999) as it was more important that the frames are colorized properly than they were consistent from frame to frame. Just enforcing consistency could mean that the frames are not properly colorized, but have the same color throughout. Unfortunately, though our frames were consistently colored and losses ranged from 0.02 - 0.04, they were all being colored a metallic brown. None of the frames had any other color. We were limited in time, but we believe that training for additional epochs could potentially also better our results. Thus, we still consider it to be a viable component to our model.

5. Conclusion & Future Work

We determined training a ResNet based model on an appropriate dataset that is similar to the movie that needs to be colorized, choosing an optimal learning rate and a loss function which enforces consecutive frames to be colorized similarly is enough to have substantial results. Our loss values on our model ranged from 0.0495-0.068.

Though colorizing via deep learning gave us an interesting colorized video, it is clear that we have not yet come at a stage where deep learning can completely substitute colorizing software developed by artists and software engineers. Every single object in the film, for example, is not colorized and is not as vibrant as the colorized version of Mughal-E-Azam.

In the future, we would like to modify our network in such a way that it does not crop the video frames, like Iizuka's network, and improve our approach of enforcing similar colors in subsequent frames whether by exploring our sequential loss idea further or implementing an LSTM to retain information about object colors from frame to frame. We also would like to explore data augmentation further – while the Melas-Kyriazi model utilizes data augmentation in the form of horizontal flips, we believe that adding noisy frames to our dataset will be more beneficial, as the original black and white film contains noisy artifacts.

6. References

[1]Luke Melas. https://github.com/lukemelas/Automatic-Image-Colorization, 2018.

[2] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa."Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification". ACM Transaction on Graphics (Proc. of SIGGRAPH), 35(4):110, 2016.

[3] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization, 2016.

[4] G. Larsson, M. Maire, and G. Shakhnarovich. Learning Representations for Automatic Colorization, 2016.

[5] S. Fenu, C. Bagwell. Image Colorization using Residual Networks.

[6] Jeff Donahue and Lisa Anne Hendricks and Sergio Guadarrama and Marcus Rohrbach and Subhashini Venugopalan and Kate Saenko and Trevor Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description, 2014.

[7] Tanvi Sahay and Ashutosh Choudhary. Automatic colorization of videos, 2017.