Colorization of Video using Deep Learning

Abstract

Colorization of grayscale videos using a deep learning model consists of two major stages — the colorization of individual frames of the video, and ensuring temporal consistency between the frames to eliminate flickering issues. The model presented in this paper largely uses the local and global features of each frame to obtain the best colorized results. To account for temporal consistency, we introduce short-term temporal loss and long-term temporal loss. As a result of this, two principal requirements are satisfied. Firstly, the quality of colorization on each frame is rich, natural and deep. Secondly, the transition between each frame is smooth and temporally stable.

1. Introduction

The aim of the project is to transform black-and-white videos to their colorized form using deep learning. Current research exists in colorizing images from their black-and-white form, but to achieve the same task on videos proves to be challenging, and has multiple roadblocks. One such issue arises from the fact that a particular object, such as a car can have different natural colors. This may lead to various frames of the video being colored differently. As the colorization is not stable, it leads to inconsistencies and flickering of the video.

The approach to the problem consists of breaking down a dataset of videos into individual frames, colorizing them based on local and global image features. This forms the baseline of the project. Further, to successfully colorize the video as a whole and preserve temporal consistency, the model encompasses a pre-trained FlowNet model which outputs warped image and an occlusion binary mask to obtain short-term and long-term temporal losses.

2. Background/Related Work

Early colorization methods focused on using local user hints in the form of color points or strokes which colorized pixels in space-time that have similar intensities with similar colors [6].

[3] proposes a model based on coloring grayscale images using convolutional neural networks. The approach uses a combination of global image priors, extracted from the whole image, and local image features, computed from small image patches. The global priors act at an image level such as whether or not the image was taken indoors or outdoors, and what time of the day it was, while local features act at the pixel level, representing the texture or subject at a specific location. By combining these features, the semantic information is used to seamlessly color input images which only consist of shades of black and white.

The paper [7] involves transferring the style obtained from a specific input image to an entire video. It uses an existing model from [1] to transfer styles from the reference image to individual frames, which uses the VGG convolutional neural network to obtain high-level image features. Subsequently, a new image with similar neural activations as the content image and feature correlations as the style image is obtained, starting from white noise. The crux of the paper deals with regularizing the transfer of style between video frames. This is dealt with using a multi-pass algorithm that alternatively processes the video using both a forward flow and a backward flow, to produce a coherent video. Additionally, consistency over a long period of the video is maintained by using long term motion estimates.

The work in [4] addresses the problem of temporal inconsistency in videos. The model deals with video inputs which have flickering issues, and stabilizes them to produce temporally consistent videos. The model uses short-term and long-term temporal losses along with perceptual loss from a pre-trained VGG network which maintains perceptual similarity between video frames that are seen in the output and the processed frames. To capture spatial-temporal correlation of the video, the model embeds a convolutional LSTM.

Our proposed model comprises of ideas from [3] as a baseline to convert grayscale video frames into colored ones. As an extension to this model, learnings from [4] are used to model a loss function that maintains optical flow and minimizes temporal fluctuations between frames of the video.

3. Approach

The initial part of the model involves training of the Image ColorNet model with Places205 and ImageNet dataset.

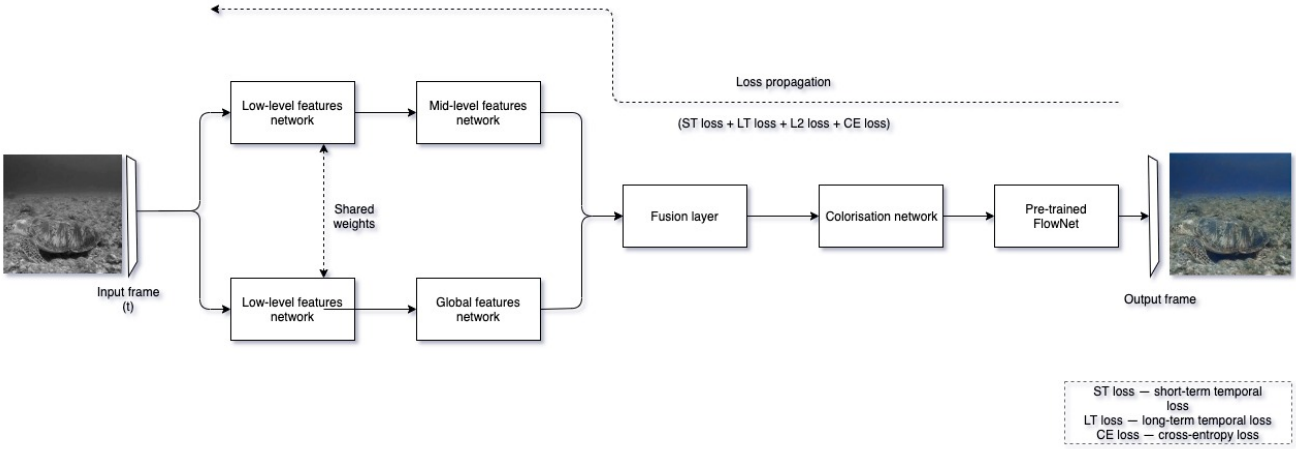

Figure 1: Model Architecture

The model architecture predominantly follows [3]. This architecture is a Convolutional Neural Network with emphasis on local and global features of the image. Local features include ones specific to small patches in the image which represent local texture or object, while global features represent the overarching theme of the image. The second part of the model focuses entirely on ensuring coherency between each frame of the input video and preserving optical flow. The stages involved in each are discussed below.

3.1. Image Colorization

The individual frames are colorized by obtaining low-level, mid-level and global features. The low-level and mid-level features are then combined with the global features in the fusion layer and colorized using the ColorNet. These are described in detail below.

3.1.1 Low-level Features

The local level features are obtained by 6 convolutional layers with increased strides. This essentially reduces the data to half its size and eliminates the need for using pooling layers. The kernel size for convolutional layers are 3x3 and the strides alternate between 2x2 and 1x1, where size of 2 is used when reducing the dimension.

3.1.2 Mid-level Features

The middle level features are obtained by adding 2 additional convolutional layers to Low-level Feature layers with kernel size 2x2.

3.1.3 Global Features

The global level features include layers of low-level feature with 4 additional convolutional layers followed by 3 fully connected layers resulting in a 256-dimensional feature vector.

3.1.4 Colorization Network

A fusion layer is introduced in order to combine local features with global features to obtain a 512-dimensional vector, with a weight of 256x512 and bias of 256x1. The fused features are processed by a set of convolutions and upsampling layers. Alternating these two layers results in an output that is half the size of the original input. Chrominance is computed by a convolutional layer with Sigmoid function. The final color image is obtained by combining this with the input intensity image. The loss function used in training the color network is the mean squared error with cross-entropy loss. The training image is converted to grayscale and transformed into the CIE L*a*b* colorspace. The a*b* components are globally normalized. We compute MSE between scaled ground truth with the computed output, and loss back-propagated through all the layers. To properly learn global features of the image, we calculate the cross-entropy loss between the classes. The cross-entropy loss is back-propagated only to the global features network and shared low-level features network — it doesn't affect colorization network.

3.2. Video Colorization

After building the ColorNet which was the baseline for the project, it was used to colorize the video by splitting the video into frames and colorizing them individually. We compute optical flow between frames using a pre-trained FlowNet model. With the wrapped image and binary mask obtained from FlowNet model, we compute temporal loss between the frames which consists of the short-term temporal loss and long-term temporal loss. We then jointly train

the model with a combination of temporal loss, L2 loss and cross-entropy loss. This loss is minimized in ordered to obtain optimal results.

3.3. Loss

3.3.1 Short-Term Temporal Loss

The short-term loss provides temporal consistency between consecutive frames and is the warping error between the consecutive outputs [4].

$$L_{st} = \sum_{i=2}^{T}\sum_{i=1}^{N} M_{t=>t-1}^{(i)} \|O_t^{(i)} - \hat{O}_1^{(i)}\|_1$$

where $O_t^{(i)}$ represents a vector $\in R^3$ with RGB pixel values of the output O at time t, N is the total number of pixels in a frame. $\hat{O}_1^{(i)}$ is the frame O_{t-1} warped by the optical flow $F_{t=>t-1}$ and $M_{t=>t-1} = e^{\|I_t - \hat{I}_{t-1}\|_2^2}$ is the visibility mask calculated from the warping error between input frames I_t and warped input frame $\hat{I}_{t-1}$.

3.3.2 Long-Term Temporal Loss

Once short-term temporal consistency is accounted for, there is a need to maintain the consistency across frames which are far apart within a given video. This is where we define long-term temporal loss between first output frame and all of the output frames.

$$L_{lt} = \sum_{i=2}^{T}\sum_{i=1}^{N} M_{t=>1}^{(i)} \|O_t^{(i)} - \hat{O}_1^{(i)}\|_1$$

3.3.3 L2 Loss

We add another loss function L_2 to our network to measure the difference between the colored output and the ground truth.

$$L_2 = \|\tilde{x}_t^{ab} - x_t^{ab}\|_2$$

3.3.4 Cross-Entropy Loss

To properly learn global features of the image, we calculate the cross-entropy loss between the classes. The cross-entropy loss is back-propagated only to global features network and shared low-level features network. [3]

$$L_{ce} = -\alpha(y_{l^{class}}^{class} - log(\sum_{i=0}^{N} exp(y_i^{class}))$$

3.3.5 Total Loss

The total loss is the combination of all the other losses described above.

$$L_{total} = \lambda_{st}L_{st} + \lambda_{lt}L_{lt} + \lambda_2 L_2 + \lambda_{ce}L_{ce}$$

	RMSE	FID	Colorfulness
[GT]	0	0	20.3
[5]	-	7.26	10.47
[8]	-	4.02	17.90
[9]	-	8.38	20.16
[2]		4.78	15.63
Ours	8.37	9.42	13.16

Table 1: Performance metrics (where GT is the ground truth).

4. Experiment

Datasets. The image dataset consists of Places205 dataset and a part of ImageNet. Moments in Time dataset from MIT forms the video dataset. A subset of dataset obtained from Videvo was also used as validation data.

Experiments. We built a recurrent structure with a parallelized VideoNet but due to the time and GPU limitations we were unable to train it with a decent batch size (which was set at 4). We finalized on the ColorNet model with newly introduced loss using FlowNet to train the model. The model was trained from scratch for 1 epoch which consisted of about 4000 iterations. The batch size was set at 75 frames which is roughly 2.5s per video. The duration for training one epoch was roughly 4 days.

4.1. Results

We evaluate our results both quantitatively and qualitatively. Quantitatively, we test if model produces realistic color images by measuring Frechet Inception Distance (FID), root mean squared error (RMSE) and colorfulness in Table 1. Colorfulness of the frames are measured using psychophysics metric. Qualitatively, we can see colored video frames which are realistic and vibrant, and elimination of flicker in the final video. The observation from these results is that datasets related to places and landscapes performed well, since our training data was primarily focused on places. The poor images in Figure 4 corresponds to the fact that training data didn't have broader range of images. But we see that temporal consistency between frames is maintained even in the poor results. The FID is more than [5], [8], [9], [2] but the images colored are more natural than [9], [8] and less vibrant than [2].

RMSE. The root mean square error is defined as follows.

$$RMSE = \frac{1}{\sum_{m=1}^{M} N_m} \sum_{t=2}^{T}\sum_{i=1}^{N} \sqrt{\|[y_{\alpha\beta}^{(m)}] - [\hat{y}_{\alpha\beta}^{(m)}]\|^2}$$

where α and β belongs to a and b in Lab color-space.

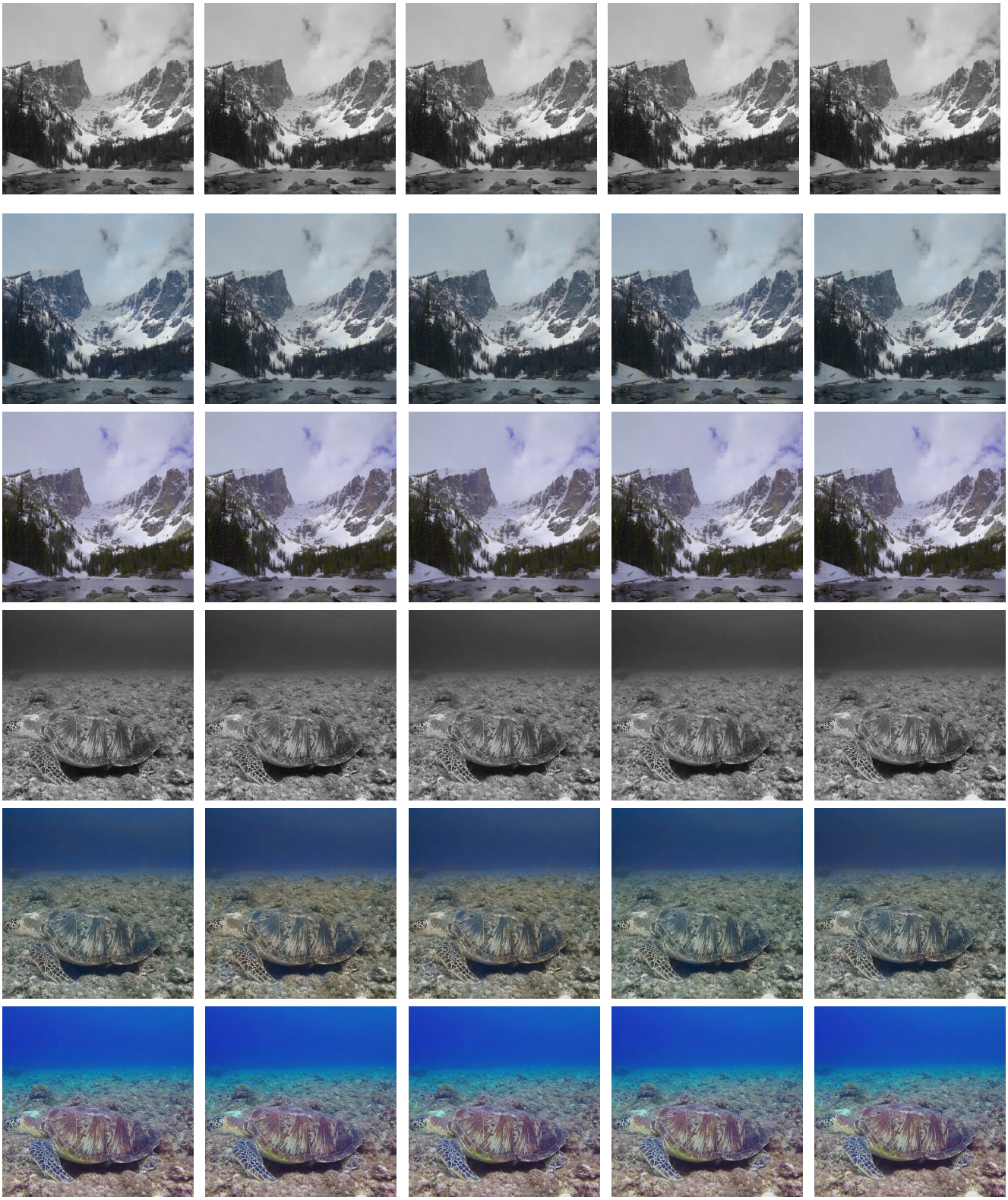

Figure 2: Frame-wise Results (First row is the grayscale input, second is the colored output, third is the ground truth).

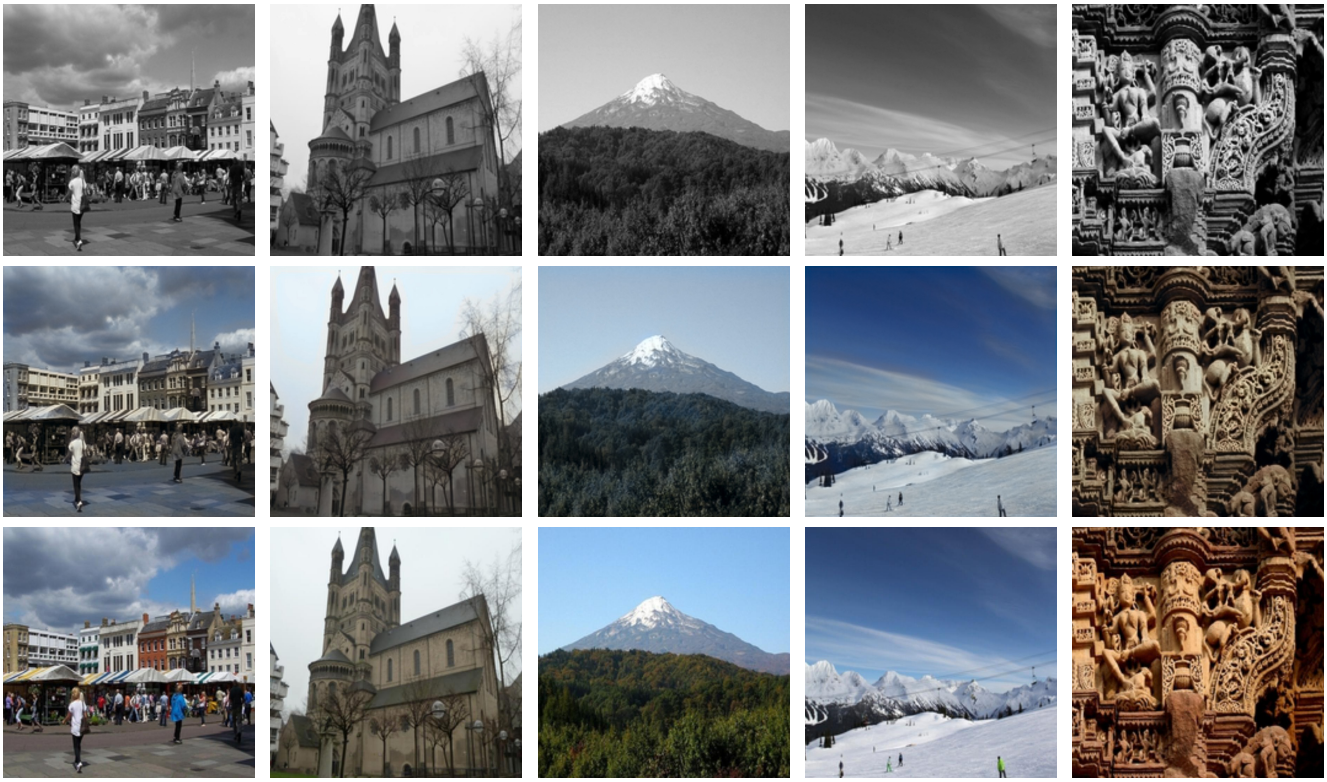

Figure 3: Image-wise Results (First row is the grayscale input, second is the colored output, third is the ground truth).

Figure 4: Failure modes (First row is the grayscale input, second is the colored output, third is the ground truth).

FID. The frechet inception distance is given by,

$$d^2((m, C), (m_w, C_w)) = \|m - m_w\|_2^2 + Tr(C + C_w - 2(CC_w)^{1/2})$$

where m corresponds to feature-wise mean and C corresponds to covariance matrix.

5. Conclusion

Given the time, constraints the model we trained works well for landscapes and scenic places. With the temporal loss, the flickering issue was reduced, thereby maintaining temporal consistency between the frames. The model performs poorly on people as the dataset consisted of limited number of images with people, which can be improved by training on a wider range of images. The model can be significantly improved by running on a cluster with more computing power for a larger number of epochs. The recurrent VideoNet built can be improvised to be GPU efficient to run on a larger batch size. Hyperparameters set to model loss can be fine-tuned for better convergence of the network.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style, 2015.

[2] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan. Deep exemplar-based colorization, 2018.

[3] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016)*, 35(4):110, 2016.

[4] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang. Learning blind video temporal consistency, 2018.

[5] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization, 2016.

[6] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, Aug. 2004.

[7] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos. *Pattern Recognition*, page 26–36, 2016.

[8] B. Zhang, M. He, J. Liao, P. V. Sander, L. Yuan, A. Bermak, and D. Chen. Deep exemplar-based video colorization, 2019.

[9] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization, 2016.