Color Constancy through Adjusting for Ambient Light

Freddie Rice

Abstract

Unlike digital cameras, human eyes perceive similar wavelengths as different colors depending on ambient light. Although objects reflect more red light during a sunny day and more blue light on a cloudy day (for example), humans can still determine the true color of objects in a scene. We create a neural network that simulates the work of an eye attempting to remove the ambient light source. Then, we recover the ground truth from the image by analyzing the difference between the original and adjusted images.

1 Introduction

Human perception of color depends on the surrounding colors, as can be demonstrated by known illusory effects. These optical illusions show how human perception differs from that of the simple camera model. Although understanding the eye may not be important for other machine learning tasks, it is inherent to the problem of color constancy. Once solved, the task will provide photographers an automated system for adjusting the color levels such that the resultant images appear to have the same color under neutral lighting conditions.

1.1 Digital Image Pipeline

The light enters the lens of camera and hits a light sensors that detect the strength of the light. These sensors filter out certain colors in an array with a pattern in order to detect specific colors. A common filter is called the bayer filter which encodes red green and blue in a grid-like pattern. The camera then computes a linear combination of these pixels to compute an estimated red, green, and blue colors per location, when in reality these pixel readings are at different locations. To produce a human-viewable image, the raw colors must then be transformed into standard color spaces so that monitors and other viewing platforms can determine the correct colors to emit. We will focus on the raw images since it is most applicable to the problem at hand.

1.2 hGRU Role

hGRU is a new deep learning model developed by Thomas Serre's lab to lower the number of parameters used in processing images by learning long-range spacial dependencies with RNN units that repeatedly apply the same weights. This approach uses biology as an inspiration and attempts to use more computation rather than space while learning. Current deep learning models have too many learnable parameters which leads to overfitting, even on vast amounts of data. HGRU attempts to find a steady state solution to a differential equation where the original image and h_2, a representation of the original image, produce an unchanged h_2 over a large number of timesteps.

Preprint. Work in progress.

2 Dataset

2.1 Grayball

A researcher at SFU Laboratories walked around environments with a video camera, capturing images at discrete intervals. The camera had a grey ball mounted in front of the lens. Since grey colors encoded as RGB should contain the same amount of each, the researchers calculated the true color of the ball. In the end, we decided to train on the Cube dataset since the grayball dataset suffers from preprocessing. The camera has a feature that will automatically select a whitebalance that it thinks is appropriate.

2.2 Cube+

The Cube+ dataset is similar to that of the Grayball, except instead of a ball in front of the camera, they have a cube. Further, the dataset only has 1707 images, however they are less correlated. The images were taken mostly in Croatia, but all with the same Canon 550D camera. The raw camera data is available in this dataset, which allows for our model to make choices on unprocessed data.

3 Model

We tried a few different variants of the model to determine the best approach, but the following aspects of the model did not change. The image is transformed by a single-oppency filter. The filter finds sudden spikes in differences between colors in different directions. In the case of this model, we used the four cardinal directions and 8 different color differences. This transformation takes an image with color mapped in red, green, and blue (values between 0 and 1) to an image mapped in these 8 times 4 = 32 different channels. These image representations are fed through an hGRU with a variable number of timesteps and kernel sizes to produce a final representation h_2. The only differences in the model below is how we choose to interpret / learn for this h_2 value.

3.1 Center column

At first, we started by analyzing the Greyball dataset and had an idea for creating a model that could over time find the hidden h_2 that would properly account for the white balance levels. Effectively, the intuition is that as the hGRU applies its filter, it will learn to push relevant information to the center of the image (since convolutions can learn to shift data quite easily). We split up the channels of this center column and used different parts to code for the proper color levels. We tried max, mean, and a simple fully connected layer, but max worked the best. This method worked very well on the Greyball dataset, but then failed miserably on the Cube+ dataset. Most likely this is due to the limited size of the target space for the ground truths in Greyball dataset. Although it holds more than 11000 images, individual images and their corresponding ground truths are highly correlated. Although the model does not overfit the data, it does overfit *to* the data in the sense that the trained model will not be able to generalize to a larger group of images.

3.2 Adjust Method

Since the center column method failed on the cube dataset, we came up with another method for determining the correct colors. The adjust method teaches the hGRU unit to adjust for the color. Over time, the h_2 should converge to a representation of the input that has adjusted for color (much like how your eyes, while looking at one setting for a while, will adjust to the ambient light). To achieve this we fitted the network with a reconstruction loss to encourage the network to find an h_2 that matched the so-filtered color-corrected image. This auxiliary loss was added to our main angle loss to create the entire loss.

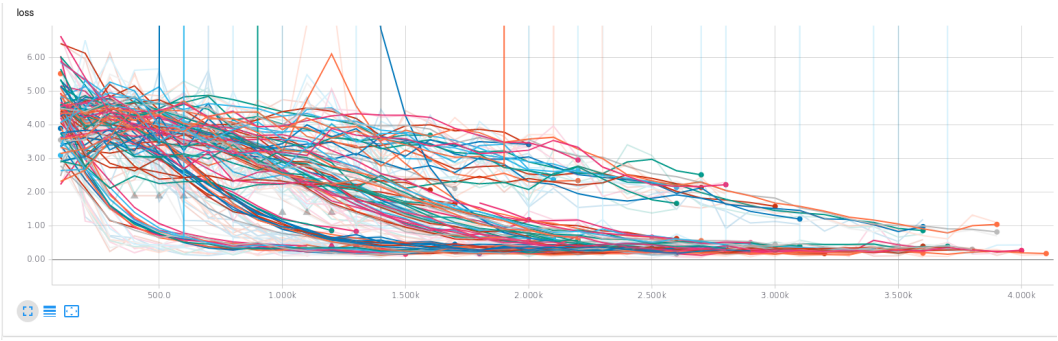
3.3 The Loss

The loss is determined by the cosine distance between the ground truth grey value and the predicted value. In all of the plots I have shown it in degrees rather than radians because it is easier to understand scale. As the loss is only determined by the angle and the multipliers are relative to a

constant green, the network really only predicts 2 values (red relative to green and blue relative to green). I constrained the model appropriately by forcing the output of the model to be two values.

4 Results

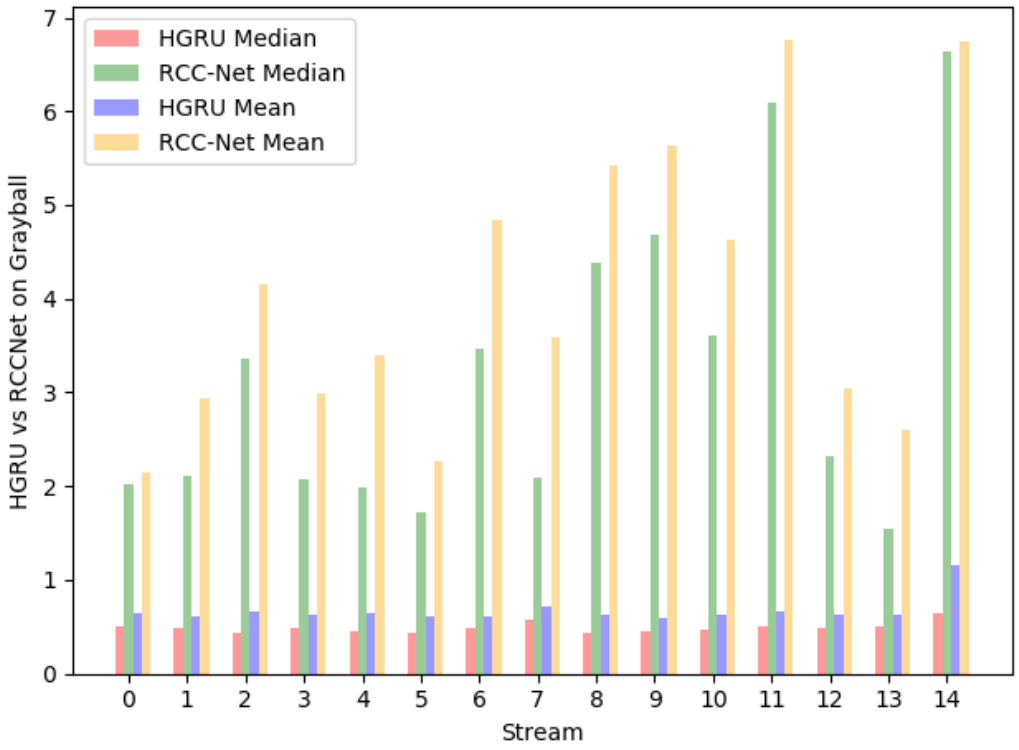
Thanks to Google TPUs I was able to learn a lot about the model in a short amount of time. Here is an example of the model running with a large number of different learning parameters to determine the best model: Further, it allowed me to train multiple of the same models at the same time. This



should not be useful, but unfortunately the loss surface is complicated. As such, initializing the same model with a different set of random weights allowed the model to find a better global minimum.

4.1 Grayball

As discussed earlier, the Grayball dataset was fairly easy. Here is an hGRU we trained with a kernel size of 13, timestep of 6, and max method to as compared to RCC-Net:

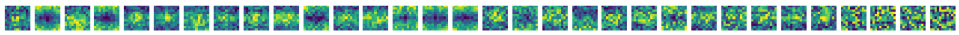


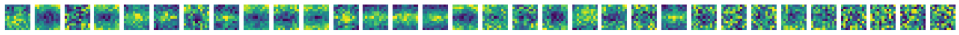
4.2 Cube+

Our model did not perform as well on the cube dataset as we had hoped. With more tuning we could definitely get closer, but as it stands we could only squeeze out a median/mean of around 3 to 4 on the test set, whereas the leading models are able to determine the gt between 1 and 2 degrees. Our best run was a mean of 3.24 and a median of 2.52 degrees.

5 Analysis

Unfortunately, the model does not yet achieve state of the art results. I attribute this to not having found the best approach to optimizing the hGRU unit. The surface of the loss function has a lot of saddle points, so weights are left dangling in awkward middlepoints. Below is a set of weights from the hGRU. The ones on the left have formed into units that learn something structured about the image, whereas the 4 on the right are relatively random in nature. This implies that the hGRU is overfitting the values from these channels.





6 Future Work

Even though the hGRU model does not need as much data as other models, it could still benefit from having more images to look at. I was able to download over 3000 more CR2 raw images from google taken by canon cameras with similar filters. These would help push the reconstruction loss in the right direction and allow us to use larger filters without overfitting the data. Right now the max I could use was 13. With more data I could probably fit the hGRU better.

Further, I would like to determine which of the SO filters are actually useful by doing a hyperparmameter search over the filters (i.e. run hGRU with subsets of the filters). Then we could use lasso regression or some other method to determine which features are actually useful in determining the color constancy problem.

References

[1] Mely David A., Drew Linsley, and Thomas Serre. Opponent surrounds explain diversity of contextualphenomena across visual modalities. *Psychological Review*, 2018.

[2] Drew Linsley, Junkyung Kim, Vijay Veerabadran, and Thomas Serre. Learning long-range spatial dependencies with horizontal gated-recurrent units. *CoRR*, abs/1805.08315, 2018.