Automatic Colorization of Black and White Images

Introduction

Image colorization is the process of adding colors to a grayscale picture or black and white image using a colored image with similar content as a source. Colorization techniques are widely used is astronomy, MRI scans, and black-and-white image restoration. Colorization is fundamentally an ill-posed problem - two objects with different colors can appear the same on grayscale film. Because of this, image colorization algorithms commonly require user input, either in the form of color annotations or a reference image.

We represent images in the YUV color space, rather than the RGB color space. In this color space, Y represents luminance and U and V represent chrominance. The input to our algorithm is the Y channel of a single image, and the output is the predicted U and V channels for the image. We will use images downloaded from Flicker with the tags “Yellowstone” and “Landscape”, to train and test the classifier.

SLIC (Simple Linear Iterative Clustering)

A superpixel can be defined as a group of pixels which have similar characteristics. It is generally color based segmentation. Superpixels can be very helpful for image segmentation. There are many algorithms available to segment superpixels but the one that I am using is state of the art with a low computational overhead.

Simple Linear Iterative Clustering is the state of the art algorithm to segment superpixels which doesn’t require much computational power. In brief, the algorithm clusters pixels in the combined five-dimensional color and image plane space to efficiently generate compact, nearly uniform superpixels.

Image Representation and Training

For our model, instead of predicting color pixel-by-pixel, we predict two real values (U and V channels) for each segment of the image. This allows us to color segments based on image structures.

Additionally, we assume that the U and V channels are independent given an image segment. We will use a general machine learning algorithm to learn the output channels U and V. To extract feature vectors for each of the image segments, we take the square of $10X10$ pixels around each centroid. We then perform a 2D Fast Fourier Transform (FFT) on these squares, giving us our feature vectors $\emptyset(x^{(i)})$. The feature vectors $\emptyset(x^{(i)})$are used as the input, while the average U and V values of the segment $x^{(i)}$are used as the output.

Image Testing and ICM smoothing

To estimate the chrominance values of a test image, we perform segmentation and feature extraction. We then run the two SVRs over the segments, giving us an initial estimate of the color. To smooth out the shading of similar, adjacent superpixels, we model every test image as two Markov Random Fields (MRFs), with one MRF per predicted channel. To minimize the total energy of the MRF, we run Iterated Conditional Modes (ICM) until convergence. Finally, the original Y channel and estimated U and V channels for a target image are converted to the RGB space yielding our final colorization estimate.

References:

[1] Harrison Ho (h2o@stanford.edu) and Varun Ramesh (vramesh2@stanford.edu) Automatic Image Colorization http://cs229.stanford.edu/proj2015/150_report.pdf

[2] Cris Zanoci and Jim Andress December 11, 2015 From Grayscale to Color: Digital Image Colorization using Machine learning http://cs229.stanford.edu/proj2015/163_report.pdf

[3] Machine Learning Course on Coursera created by Stanford University and taught by Andrew Ng. https://www.coursera.org/learn/machine-learning

GROUP MEMBERS: **MENTOR:**

RAHUL CHAUHAN 2K14/CO/090

RAHUL KUMAR 2K14/CO/091

RAJAT VERMA 2K14/CO/093

VAIBHAV SINGH 2K14/CO/137

SONIKA DAHIYA
DEPT. OF COMPUTER SCIENCE, DTU