Automatic Image Colorization

Varun Ramesh and Harrison Ho
CS229: Machine Learning

Problem Statement

A Grayscale Photo of Yellowstone

The Result of Our Algorithm

Image colorization is the process of adding color to grayscale or sepia images, usually with the intent to modernize them. Automated image colorization is an ill-posed problem, as 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 propose an algorithm to automatically colorize black and white images in restricted circumstances, without requiring any user input at all. Our technique works by training a model on a corpus of images, then using the model to colorize grayscale images of a similar type. We use Support Vector Regressions and Markov Random Fields in our approach.

Approach

We represent images in the YUV colorspace, where Y represents luminance and U and V represent chrominance. Our algorithm attempts to predict the U and V channels of an image, given only the Y channel.

In order to constrain the problem, we segment the images into sections using the SLIC superpixel algorithm. Instead of predicting color pixel-to-pixel, we predict two real values for each segment of the image (the U and V channels), allowing us to color segments based on image structures.

For our models, we train two separate support vector regressions. We perform a Fast Fourier Transform on squares centered around the centroid of each superpixel. This acts as the input feature for our SVRs, with the U and V chrominance channels used as the output. Next, to smooth out shading of adjacent super pixels, we represent the segmented image as a Markov Random Field. The field uses individual potentials between the SVR-predicted chrominance and the true chrominance and joint potentials between the chrominances of neighboring segments. We then run Iterated Conditional Modes over the MRF to minimize the total energy.

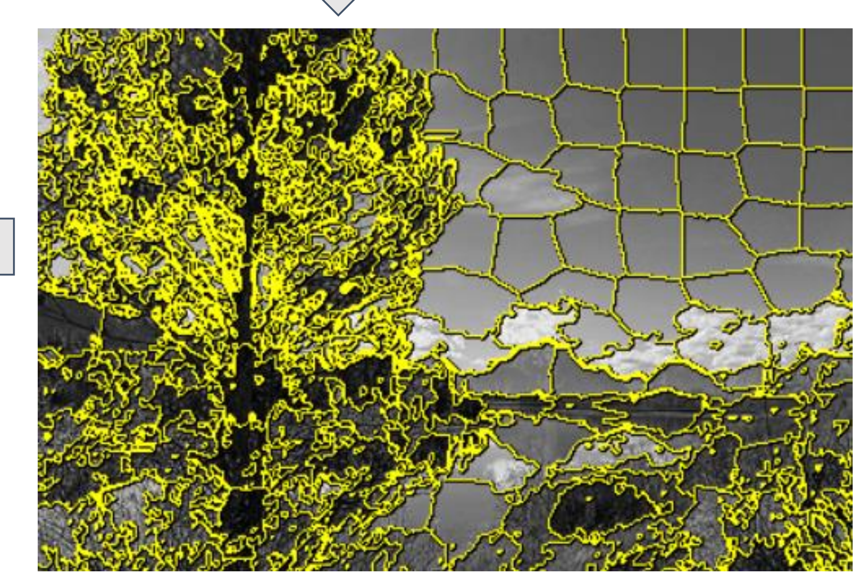
Implementation

To train and test the classifier, we decided to focus on images of Yellowstone National Park. We downloaded photos on Flickr with the tags “yellowstone” and “landscape,” as this gave a good selection of photos that did not include animals, humans, or buildings. Additionally, we split the images into training (98 images) and test (118 images) sets.

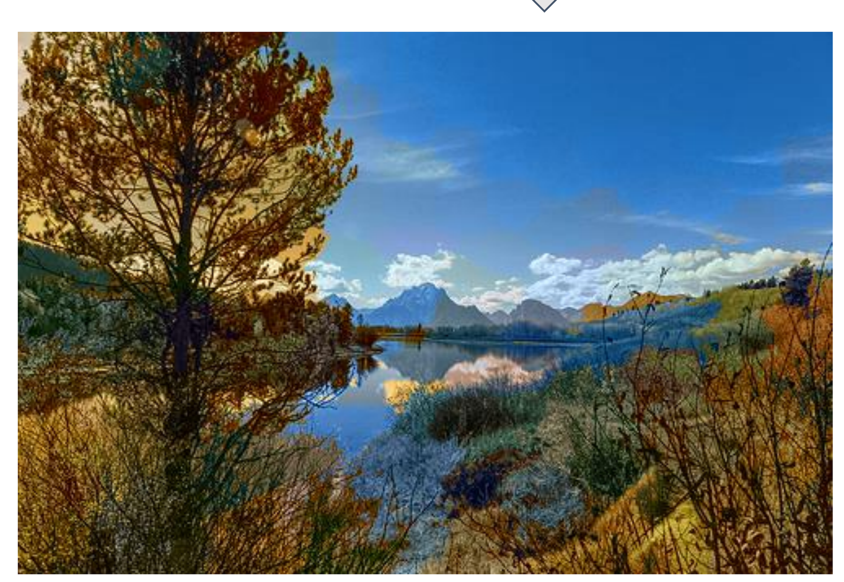
To optimize the algorithm, we evaluated the SVR and ICM over different values of the parameters, and selected the parameters that produced the minimum error. In our formulation we define the error as the average distance between our predicted RGB values and the actual RGB values over the pixels, scaled by a constant.

Algorithm Flowchart

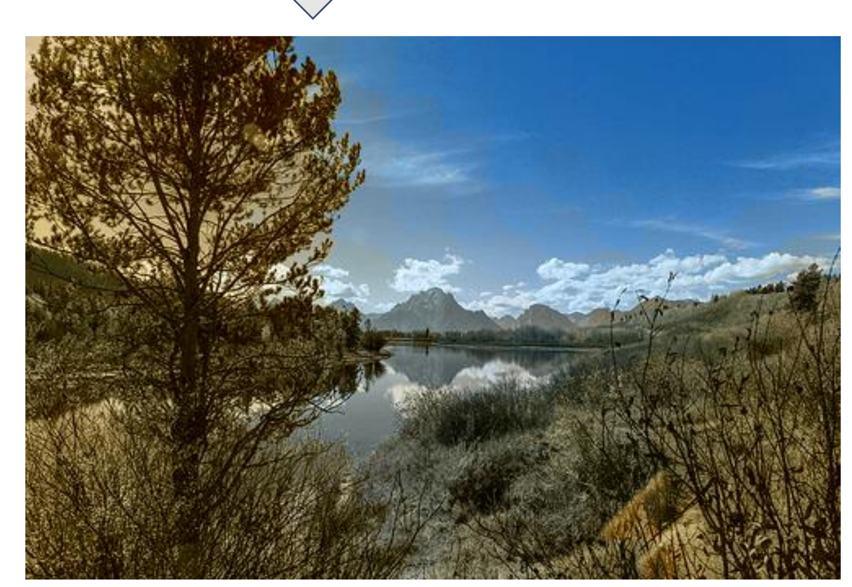
Original Grayscale Image



SLIC Segments

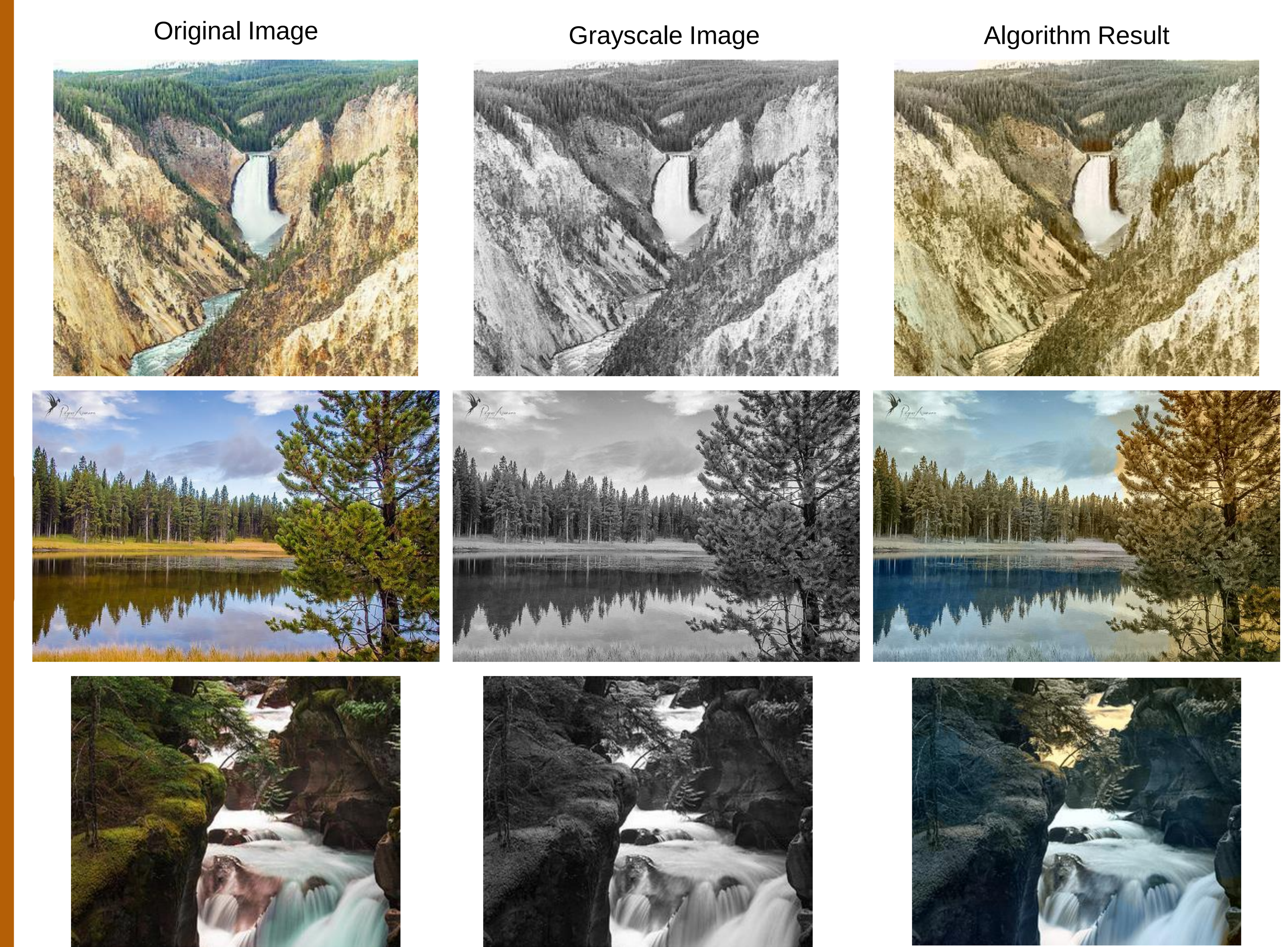


SVR Classification Results



ICM Smoothing

Results and Analysis



Our current approach performs reasonably well, given the variety of colors and textures in the training set. The algorithm successfully shades environmental features differently; as an example, the trees and their reflection in the second image are shaded different colors. The MRF improves the colorization considerably, decreasing the average error 19.30% from 0.5488 to 0.4429.

However, our algorithm could use improvement. The algorithm does not reproduce brown or yellow colors well, and tends to produce images which are excessively blue. This may be a result of overfitting in the training set.

Future Steps

- We currently use the YUV color space to separate luminance and chrominance. Alternate color spaces exist that may do this more effectively (CIELAB).
- Using the Fourier Transform allows us to understand the texture of each superpixel, but is unable to capture global features, such as an entire geyser. We may wish to use convolutional neural networks to include global features within our model and provide more accurate shadings of large structures.
- In order to colorize more image types than can be represented in a single model, we can construct a decision tree, where each leaf is represented by a different colorization model.