Automatic Colorization of Grayscale Image

Aman Kumar   Tapas Agarwal
amankr@iitk.ac.in   tapasag@iitk.ac.in
Advisor : Dr. Amitabha Mukherjee
amit@iitk.ac.in

Abstract

In this project we have described a method of colorizing a gray scale image based on segmented reference image. This method requires considerably less human intervention. The user here just need to provide a colored image which will serve as a reference to the algorithm. The objects and texture content of the reference image must be as similar as it can be to the target image. Although we can never be perfectly sure that matching of objects from target to reference image is absolutely correct. Hence to improve this we partially colorize the given image at those points where we are absolutely sure. Now this partially colored image serves as input to algorithm described in paper "Colorization using optimization" by Levin et al [2]. This method finally gives us an optimized colored image. The approach we have followed in first part is primarily based on paper *Colorization by example* by R. Irony et al[1].

INTRODUCTION

Image colorization is process of recovering color information from given only the luminous intensity information. Mathematically we have to create three dimension RGB data from one dimension intensity data, for each pixels in the target gray image. This information cannot be created on its own, we need to provide some previously learned information.

In the approach towards colorization we are using YCbCr color space. This is a luminance (gray scale) separated uncorrelated color space with Y as luminance component and Cb and Cr provides the color component to the image. The advantage of switching to this color space is that we want to some how retain the gray scale information of the target image. In this colorspace we can say that the Y component of the finally colored image is its current intensity component. Now stating that we only need to estimate the remaining Cb and Cr component of the image.

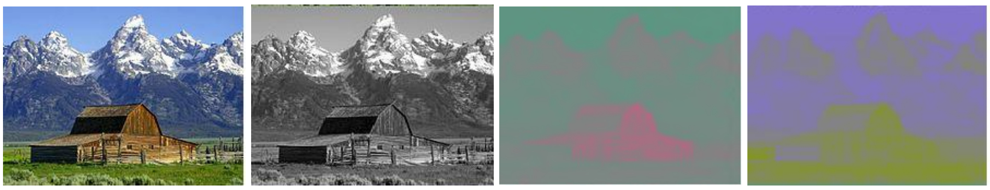


Figure 1: Colored image and its Y, Cb and Cr components *(source : wiki)*

There can sometimes be ambiguity regarding the colorization. for example, we have to colorize a scenery consisting of trees and leaves. The leaves could have been of yellowish or green color depending on season of taking picture but the final color of the leaves will be only be that of reference image.

PREVIOUS WORKS AND APPROACHES

Colorization term was invented for the purpose of coloring black and white movies and photograph to make it more appealing to the viewers. In the initial computer assisted approach towards colorization of videos, people used to manually paint at least one frame of the video and used object tracking techniques to propagate the colors through the frames.
There are two main approaches that are popularly followed :

- First method requires users to give a reference image along with the target image. Then people apply various color transfer mechanism which have been improved over the years. This is the most popular approach in this field.
- In second method, people try to build a collection of various categories of images into a database and then applying various image searching methods to find the most similar image for transferring colors.

The approach we have followed is partly based on first approach and then we uses a new approach to optimize the coloring process to obtain better colorized image.

ALGORITHM AND APPROACH

IMAGE SEGMENTATION

The user provided reference image needs to be divided into some segments differentiating among classes of objects. For this method we use mean shift based segmentation of image. In mean shift segmentation a window around the data point is defined and mean of the data in current window is calculated. Then the center of the window is shifted to the mean point. This process is repeated until convergence.[3]
following are the results of performing mean shift based segmentation process on few images from LabMe dataset [4].

Figure 2: Colored image and its mean shift based segmentation

Training using Textures

Texture features are better than corner point features when we have to match interior regions of the images. There are many texture descriptors but for keeping things simple and efficient we have used Discrete Cosine Transformation Coefficients around a neighborhood of a pixel to obtain texture vector of that particular pixel.
There can be many miss classifications and error nous result if classified in higher dimensions. We need to get into some lower dimension which better describes differences among textures of different segment regions. Hence instead of doing PCA and taking top few vectors we make few changes.

- First we sample few inter-difference vectors by taking difference vectors of different regions and apply PCA to take top x Eigenvalue vectors.
- Similarly we sample few intra-difference vectors by taking difference vectors of same regions and apply PCA to take bottom y Eigenvalue vectors.

Now we use a combination of these vectors to obtain a matrix T which will project above feature vectors onto a (x+y) dimensional subspace.

Classification

feature vectors for each pixels in target image is obtained using the method described above and the projected onto above obtained sub space using matrix T only. Now we can simply classify pixels of target image using K Nearest Neighbor (Knn) method.
It is correct to assume that there will be many miss classifications. To avoid miss classifications we replace the label of the pixel with the dominant label in its neighborhood. The assumption here is that the miss classified pixel will be surrounded by all different label pixels.
The term dominant label is given by label with highest confidence value as defined below.

$$\mathrm{conf}(p,\ell) = \frac{\sum_{q\in N(p,\ell)} W_q}{\sum_{r\in N(p)} W_r} \quad W_q = \frac{\exp(-D(q,M_q))}{\sum_{r\in N(q)} \exp(-D(r,M_r))}.$$

Here D is euclidean distance between vectors, M_q is best match of pixel q in its feature space. N(p) is the k by k neighborhood of the pixel p.

Figure 3: Some miss classified pixels of target image and their rectification. Colors here represent labels *Source image 1: LabMe dataset [4]*

Colorization

Now we have to determine other two components of classified pixels. If we denote the Cb and Cr component of the image by the term C(p), then C(p) for each pixel p in target image is given by formula :

$$C(p) = \sum_{q \in N(p,\ell)} W_q C(M_q(p))$$

All The terms in this formula are same as described in previous section

Colorization Using Optimization

In this method we colorize the image using "Colorization by Optimization" by Levin et al. It colorizes images using one premise neighbouring pixels in space having similiar intensities should have similar colors .This method required manual annotation of colors to be provided by the user.It is used to mark the region of grayscale image which the user want to colorize.

An Example of how above method works is shown below

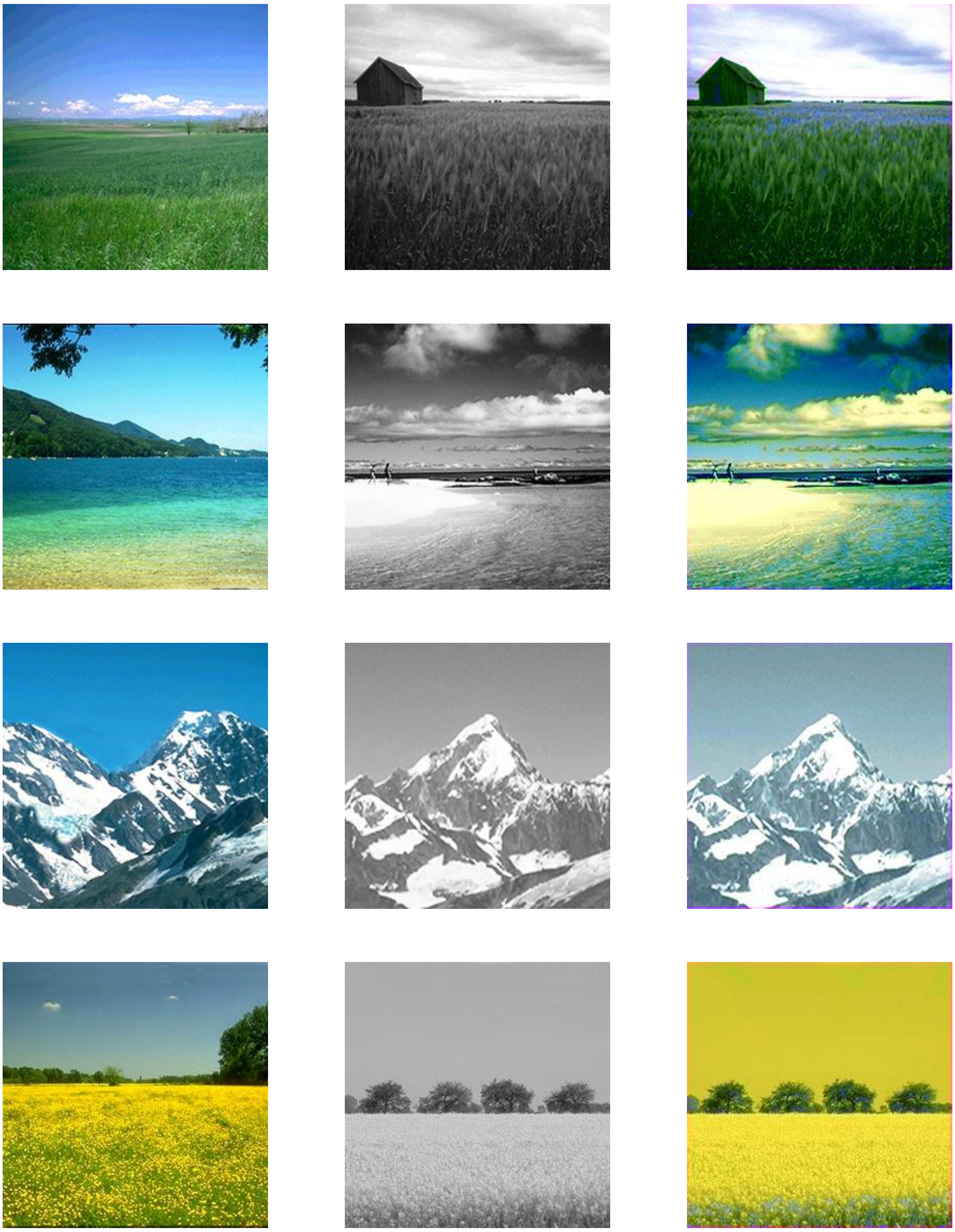
Figure 4: (a)Grayscale image (b) User Marked Image (c) Colorized output
Source: "Colorization using optimization [2]"

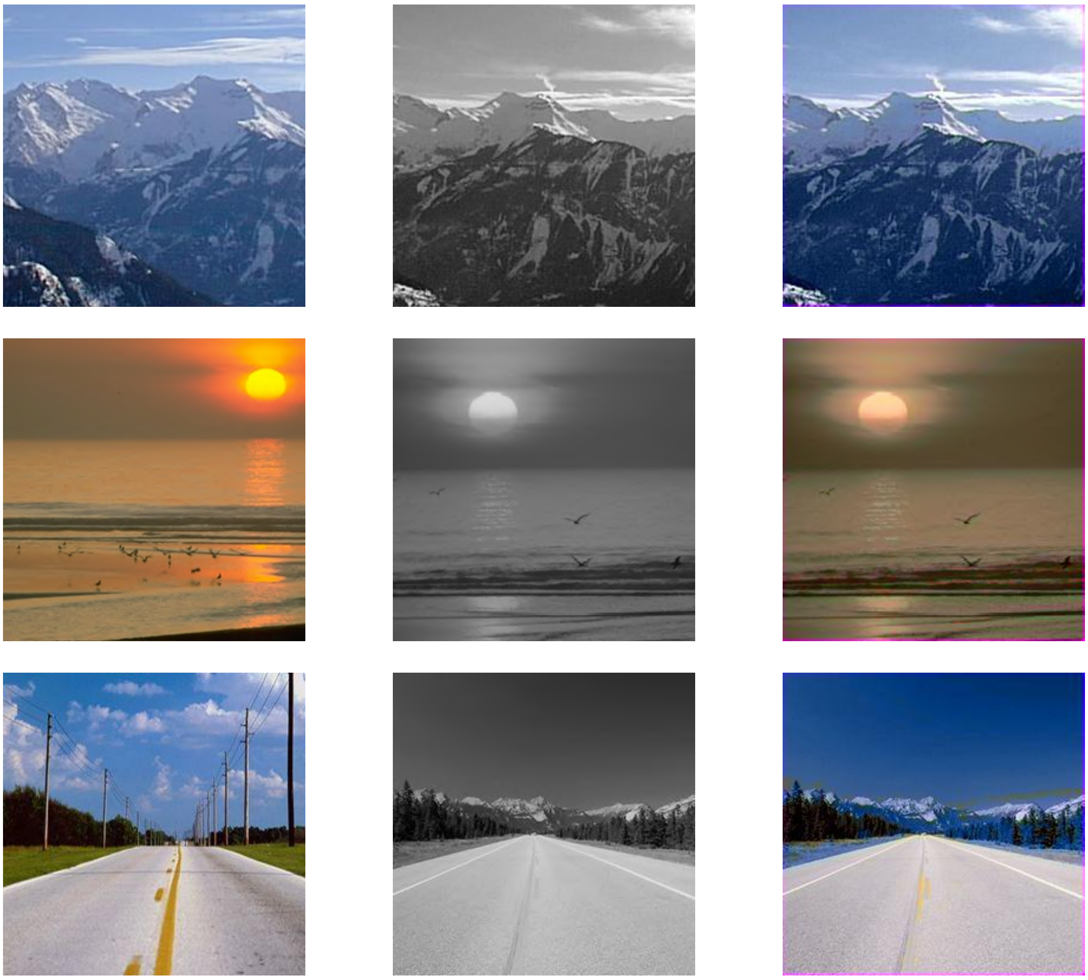
Since we cannot we very sure about perfect classification of pixels we only colorize those patches in the given target image where we are very sure (ie. have high confidence value) Using the above algorithm we can automatically generate scribbles using high confidence level (i.e conf(p,l) > 0.9) and provide the marked image to Levin's optimization algorithm.

Results

Following colorization is performed on manually handpicked similar images from MIT's LabelMe dataset [4] which is an on going project containing images of countryside, coasts, mountains, roads etc. One of these images is converted into gray scale and other one is used to colorize the first one.

Reference Image **Target Image** **Colored Image**





LIMITATIONS

Following are the few observations we have obtained while performing experiments on various images.

- Colorization of target images is crucially dependent on correct segmentation of reference image. hence the type of segmentation method used matters the most. Thus More reliable the segmentation method better is the colorization of the image.
- The above algorithm is significantly rely on the texture based features for matching of objects in reference and target image. Since texture matching is not perfect , assignment of colors is not proper and leakage of colors takes place , even if reference and target images are almost similar.
- The performance of the algorithm decreases for occlusive images.

Conclusions and Future Work

Conclusions

We have discussed above the semi supervised method to colorize a given gray image using a color transfer mechanism. It uses texture features to recognize various objects in the user provided reference image and also describes a color transfer method. To achieve better colorization we generate micro scribbles at high confidence areas and the complete colorization is performed using Levin's *Colorization using optimization* algorithm. The combination of these two method gives highly optimized colorization of images.

Future Work

To better describe texture information of a pixel we can use more advance techniques like Maximum Response filter bank [5]. Also while classification instead of fully relying on texture feature we can also incorporate other features like SIFT , FAST etc so that the method does not fail when the same object has different texture representations.
The method can further extended to color videos by providing reference image to few frames in the videos and subsequent frames can be treated as reference for the next image.
An approach towards fully optimized version can be made by developing an object bank of images. object bank uses various object detection filters to detect objects in a image and stores them effectively. Off course we cannot do this for any general image but we can target for one category of images such as natural images [6].

References

[1] Colorization by Example,R.Irony, D.Cohen-Or, and D.Lischinski, Eurographics symposium on Rendering [2005]

[2] Levin A., Lischinski D., Weiss Y.: Colorization using optimization. ACM Transactions on Graphics 23, 3 ,689âĂŞ694 [2004]

[3] Edison. RIUL, Center for Advanced Information Processing, Rutgers University, [2009] http://coewww.rutgers.edu/riul/research/code/EDISON/

[4] Dataset used (all result Images) : LabelMe , Computational Vision Cognition Laboratory (MIT)

[5] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision, 43(1):29-44, [2001]

[6] Li-Jia Li, Hao Su, Eric P. Xing and Li Fei-Fei. Object Bank: A High-Level Image Representation for Scene Classification and Semantic Feature Sparsification. Proceedings of the Neural Information Processing Systems (NIPS), [2010].