Automatic Colorization Of Grayscale Images

Aman Kumar (11070)
Tapas Agarwal (11764)

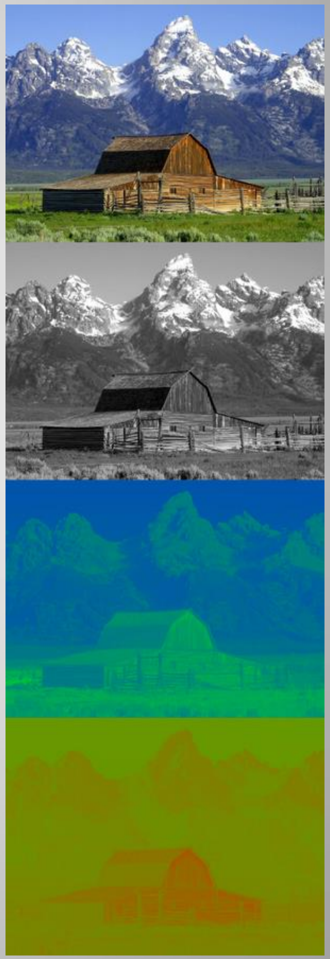
Problem and Motivation

- Given one or more gray-scale image(s) , we want to automatically colorize it using a similar colored image (provided by the user).

- The application of such method is in colorization of old photographs & cinemas, IR images, CCTV cameras, astronomical photography, etc.

YUV Color space

- In this problem, instead of working in RGB color space we will use YUV color space.
- In YUV space , **Y** stands for **Luminance component** of image (gray scale part of the image).
- **U and V** are **Chrominance component** of the image.
- This is an advantage for this problem as we already have Luminance part (Y) in the target image. Hence we only need to determine (U, V) components.



Source : Wikipedia

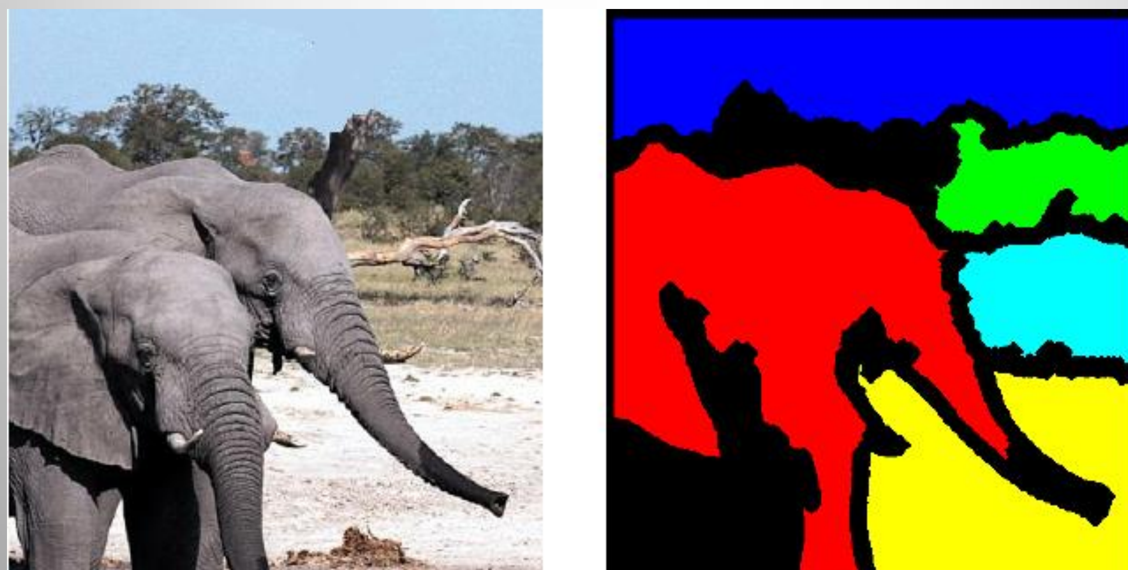
Our Approach

The approach consists of the following main conceptual stages:

- **Segmentation** of reference image.
- **Training** based on feature vector and labelled segment of reference image.
- Segmentation of target image and its **classification** based on above trained model.
- **Colorization** using optimized method.

Segmentation

- Image segmentation is done on the Luminance (Y) channel of the reference image.
- Since the target image (only having Y) is also to be segmented.



Source: Reference 1

Feature Descriptor

- For a reliable classification, training must be based on texture feature of the reference image.
- We will use the Discrete Cosine Transform(DCT) coefficients of a *k* by *k* neighbourhood around the pixel as its feature vector (dimension k*k)
- DCT feature are known to be better texture descriptors which are invariant to Translation and Rotation.

Training Stage

- Intra-difference : difference between vectors of similar segments.
- Inter-difference: difference between vectors of different segments.
- Our classifier must ignore Intra-difference between vectors and must decide on basis of inter-difference.

Training And Classification

- We use PCA and Projections.
- Randomly sample intra-different vectors, apply PCA , keep eigenvectors corresponding to **Low** values (minimizing intra-difference)
- Randomly sample inter-different vectors, apply PCA , keep eigenvectors corresponding to **High** values. (maximizing inter-difference)
- Project data points onto above space and Use **K Nearest Neighbour** during classification.

Image Space Voting

- There can still be many misclassification . A pixel p in target image, may be surrounded by pixels of different classes.
- To rectify this we replace the label of p with dominant label in N(p). Where N(p) is k*k neighbourhood of p.
- Dominant label is label with highest confidence , conf(p ,l) .

$$\text{conf}(p,\ell) = \frac{\sum_{q\in N(p,\ell)} W_q}{\sum_{r\in N(p)} W_r}.$$

$$W_q = \frac{\exp(-D(q,M_q))}{\sum_{r\in N(q)} \exp(-D(r,M_r))}.$$

Source : reference 1

- Here , D is Euclidian distance between feature vectors and M_q is nearest neighbour of vector q in feature space.

Colorization

- Let C(p) be the Chrominance coordinate (U,V) of a pixel p.
- After Classification each p in target image, the color of p (with label l) is given by

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

Source: reference 1

Optimization

- Since there might be some misclassifications , hence assigning colors to all pixels using above method will not be correct.
- We only assign colors to the pixels whose confidence in their label is sufficiently large

  (conf (p ,l) > threshold) .
- This process is called “micro-scribbling”.

Optimization

- Colorization using Optimization by Anat Levin et al (2004), describes method to colorize gray scale images annotated by user.

Source: reference 2

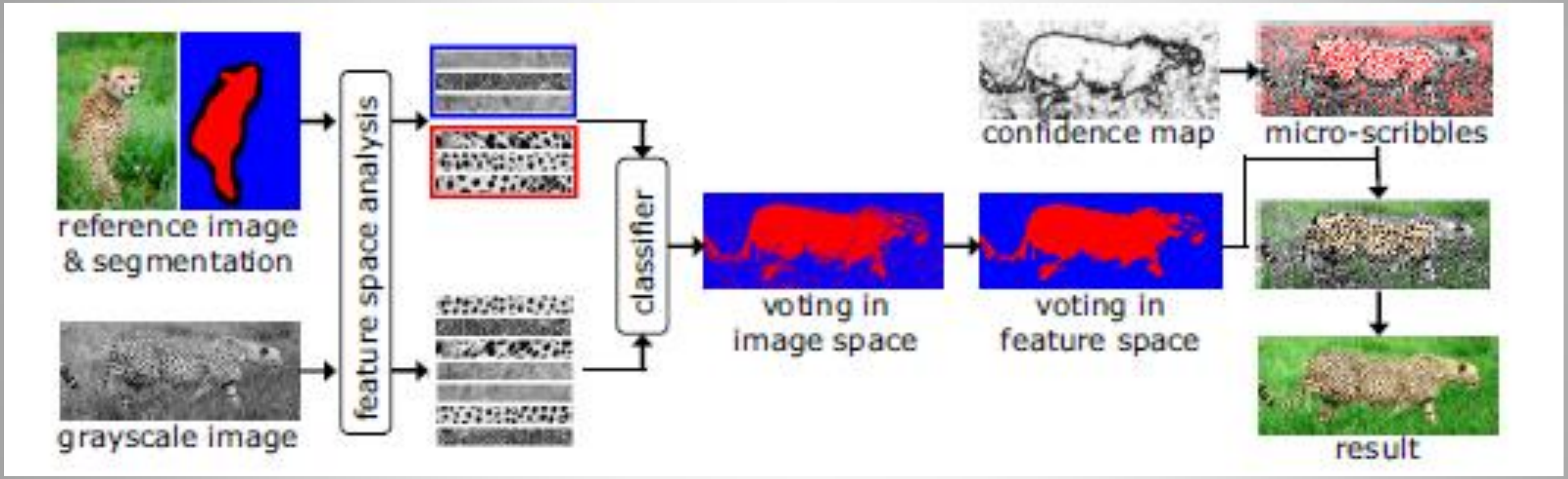
- We feed our “micro-scribbled” image this Levin’s algorithm for better results.

Dataset

- We will use Local IITK copy
  of Berkeley Segmentation Dataset BSD 300
  http://web.cse.iitk.ac.in/users/cs676/data/BSDS300-images.tgz
- It contains test and training data of similar images.

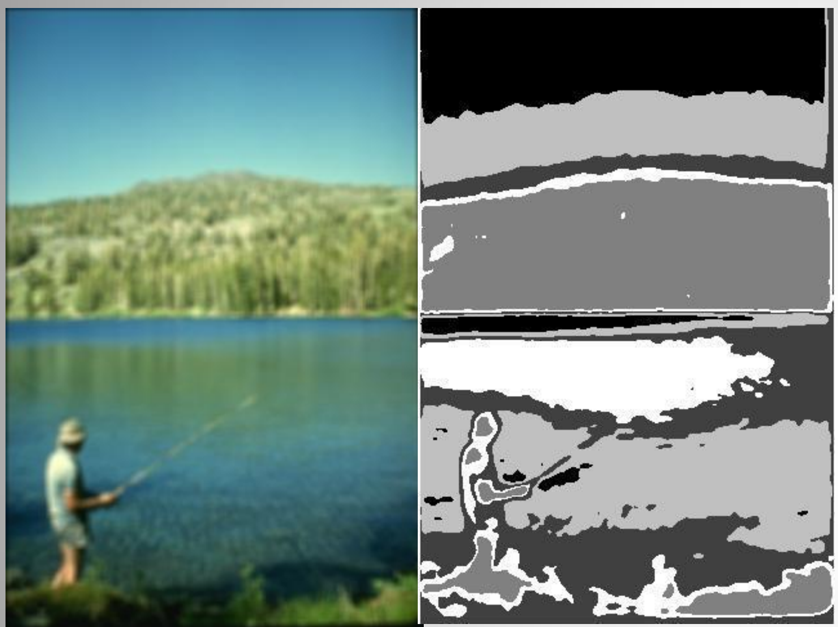
- We will also collect some similar images manually.

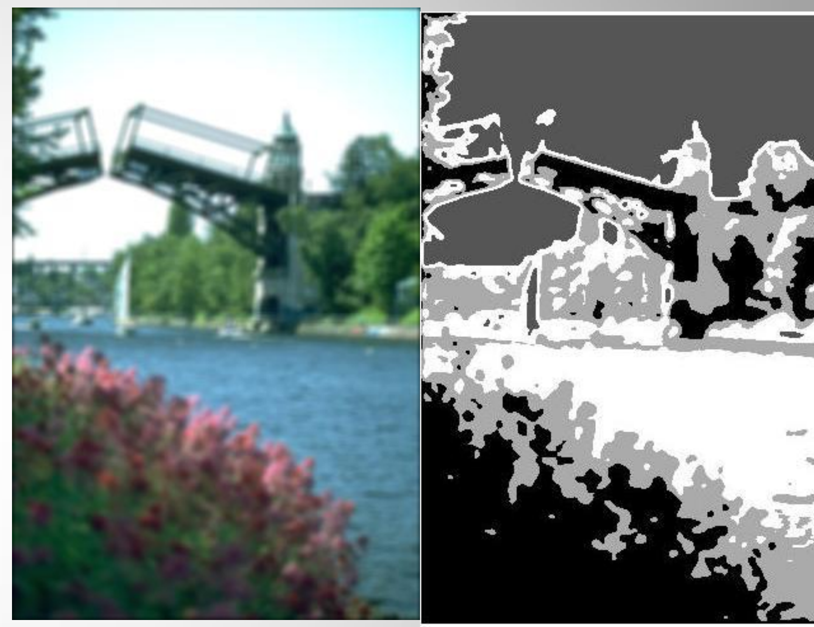
Overview of the method



Source: reference 1

Till Now





Source of colored images : Berkeley Segmentation Dataset BSD 300

We have segmented the images and obtained the Texture feature vectors of the reference and target image in 5x5 neighbourhood of a pixel.

References

1. Colorization by Example,
   R.Irony, D.CohenOr, and D.Lischinski, Eurographics symposiu m on Rendering (2005)
2. Colorization using Optimization, -
   Anat Levin, D.Lischinski, Yair Weiss,SIGGRA(2004)
3. Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation
   Siddheswar Ray and Rose H. Turi (1999)
4. Patch based Image Colorization. -
   A Bugeau and V T Ta. Pattern Recognition (ICPR), 2012.
5. Code for colorization using optimization [2] available at:
   http://www.cs.huji.ac.il/~yweiss/Colorization/

Any Questions ?

Algorithm Of Colorization Using Optimization

- This method colorizes user provided annotated gray-scale image or a video clip with few annotated frames.
- $Y(r)$, $U(r)$, $V(r)$ denote YUV component of pixel (r) at (x,y) at time t.
- Colorization of a pixel r is transformed into minimization of following quantity.

$$J(U) = \sum_{\mathbf{r}} \left(U(\mathbf{r}) - \sum_{\mathbf{s} \in N(\mathbf{r})} w_{\mathbf{rs}} U(\mathbf{s}) \right)^2$$

$$w_{\mathbf{rs}} \propto 1 + \frac{1}{\sigma_{\mathbf{r}}^2} (Y(\mathbf{r}) - \mu_{\mathbf{r}})(Y(\mathbf{s}) - \mu_{\mathbf{r}})$$

Source: reference 2