Image Processing Term Project Report

Group Members

- 110060012 徐竣霆
- 111000223 黃頂軒

1. Abstract

Recoloring grayscale images enhances their utility in areas such as historical restoration and medical imaging. Our report presents two methods using the Oregon Wildlife dataset. The first, inspired by BM3D and K-Nearest Neighbors, was computationally impractical. The second employs background segmentation and histogram transformation to map grayscale to RGB based on consistent color distributions within animal categories. Results demonstrate that the second approach effectively restores color when segmentation and dataset quality are high, highlighting the importance of accurate segmentation and dataset integrity for successful colorization.

2. Background

Recoloring grayscale images, also known as colorization, is a significant area in computer vision and image processing. It involves transforming monochromatic images into colored ones, enhancing their visual appeal and utility across various applications such as historical photo restoration, medical imaging, and creative arts.

Recolorization enhances the interpretability and aesthetic quality of grayscale images. In historical contexts, it brings old photographs to life, providing a more immersive experience. In medical imaging, colorization can highlight specific tissues or abnormalities, aiding in diagnosis and analysis. Additionally, artists and designers use colorization techniques to add creative dimensions to monochromatic works.

3. Method

3.1 Finding Dataset

After considering labeled dataset like CIFAR-10, ImageNet, we finally find a dataset call Oregon Wildlife, with decent resolution, great diversity, and medium size of training (and testing) images. It contains approximately 7000 images, with around 700 images per animal.

3.2 Problem Formulation

By extracting the coloring information from other images, we will be able to restore the color information of the grayscale image. Although the color information might be biased, we believe that we can get the ground-truth information if the size of reference image is diverse and large enough.

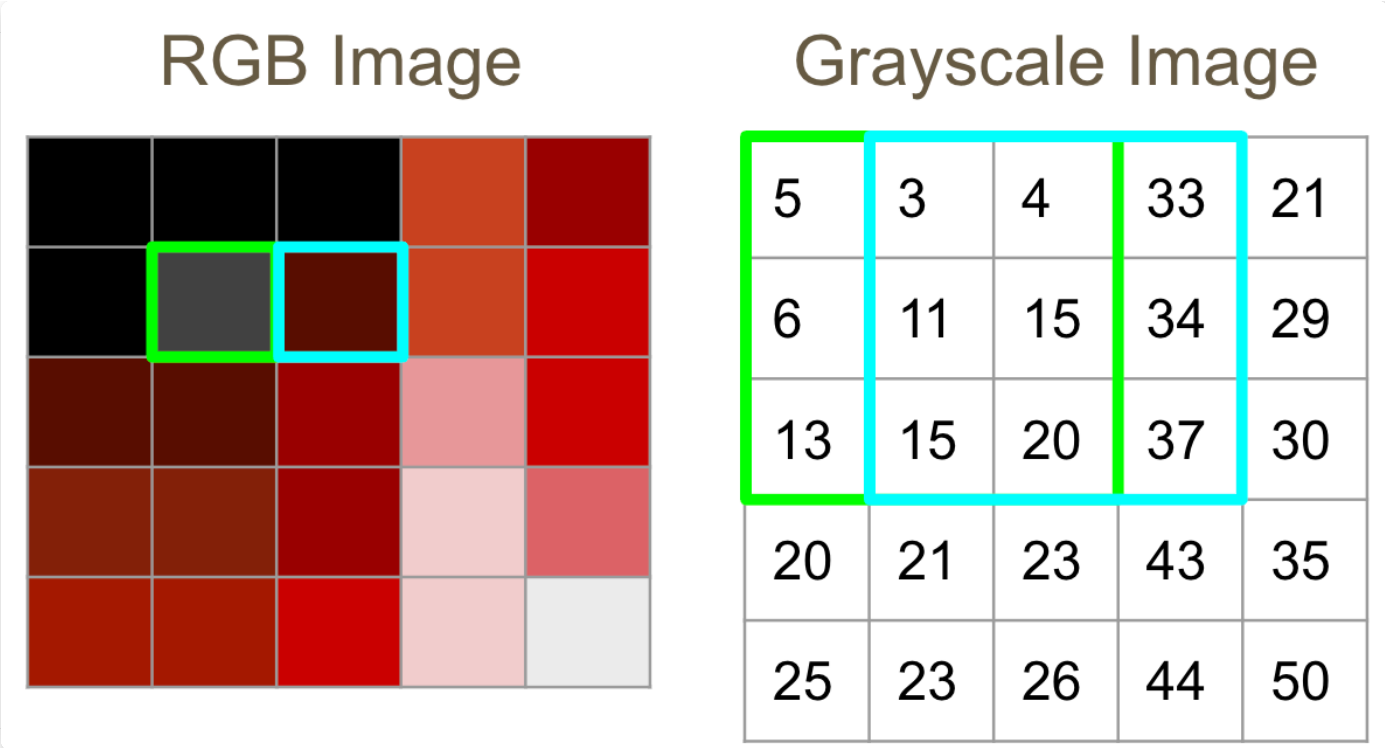
3.3 Our approach

We proposed 2 approaches in our report.

3.3.1 First Approach

This approach is inspired by BM3D. In BM3D, select sections with similar structures and take the average to reduce bias. Our algorithm contains 5 steps:

1. Select a specific type of animal, and split the dataset into training and testing set.
2. For each image in training set, we generate its grayscale image with intensity values ranging from 0~255 using the NTSC formula:
   $\mathrm{Gray} = 0.299 \cdot \mathrm{Red} + 0.587 \cdot \mathrm{Green} + 0.114 \cdot \mathrm{Blue}.$
3. Select a $K \times K$ kernel and treat every $K \times K$ window in the grayscale images from the training set as a K^2-dimensional coordinate, with the corresponding RGB values as the ground truth.
4. Convert the testing set into grayscale using NTSC formula.
5. For each $K \times K$ window in testing set, find some neighbors (number of neighbors can be modified as a parameter) in the K^2-dimensional space using KNN and apply interpolation to obtain the RGB value of the center pixel.



For the above example, the green kernel represents a 9-dimensional coordinate $(5, 3, 4, 6, 11, 15, 13, 15, 20)$ with the ground truth shown in the RGB image.

However this approach hardly work in practice since there are too many points in K^2-dimensional space. A training set of 700 images with 500×500 size have $\approx 10^9$ points. Recolor a image of size 500×500 would take too long, since time complexity of KNN is $O(nd + k)$, where n is the number of points in K^2-dimensional space, d is the number of dimension, and k is the number of neighbors. We have run our program and didn't finish its computation within a week.

3.3.2 Second Approach

Since our first approach takes too much time, we came up with another approach which has a similar idea. Our methodology is that when we consider a specific type of animal, the color distribution should be similar. Although there might be some distortions because of lights, angles, and even noises, we should still get the correct distribution if we have large enough reference sets. Our algorithm can be broken down into 4 steps:

1. Select a specific type of animal set, and remove its background using `rembg` Python package. It use U2-net to extract the background and make it transparent.
2. Calculate the histogram for each of RGB and gray for the animal set. We then compute the average of the RGB and gray histogram to remove bias.
3. Transfer the target image to grayscale, using the NTSC formula mentioned above, and compute its histogram.
4. Calculate 3 histogram transformation function to transform gray histogram in step 3

into each of RGB histogram.

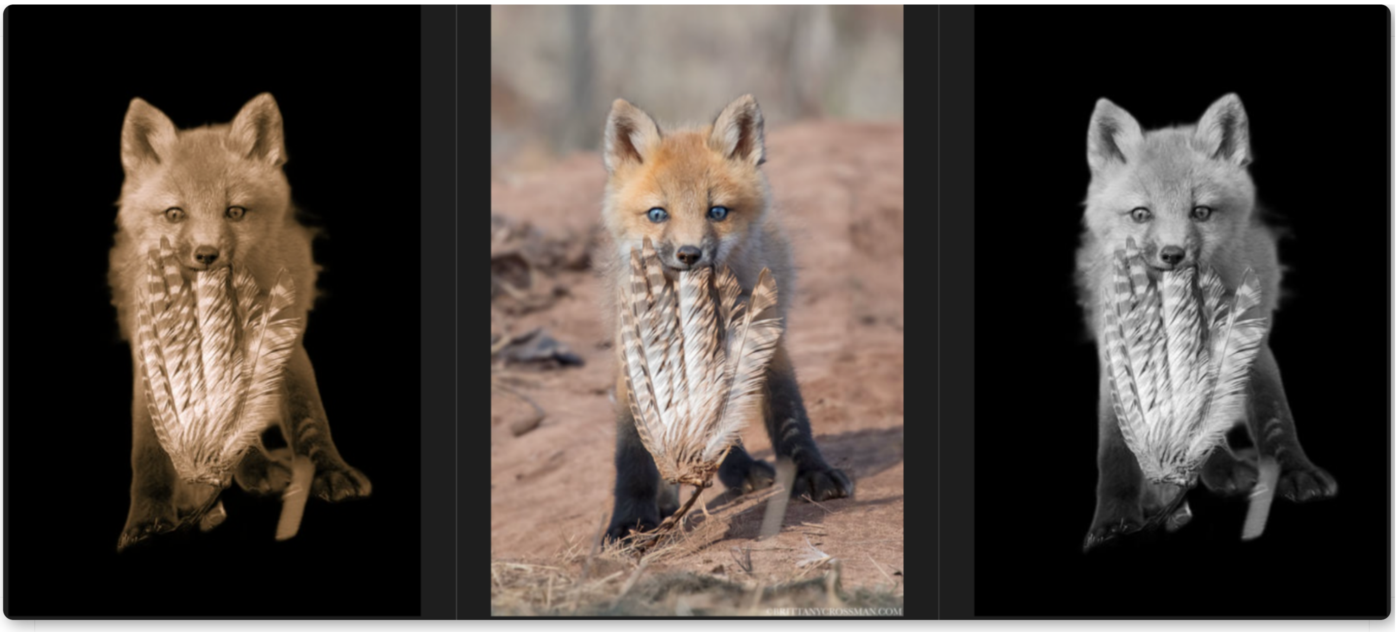
5. Take a grayscale image for testing, apply the histogram transformation from step 4 to RGB histograms, and combine RGB channels to form a recolored image.

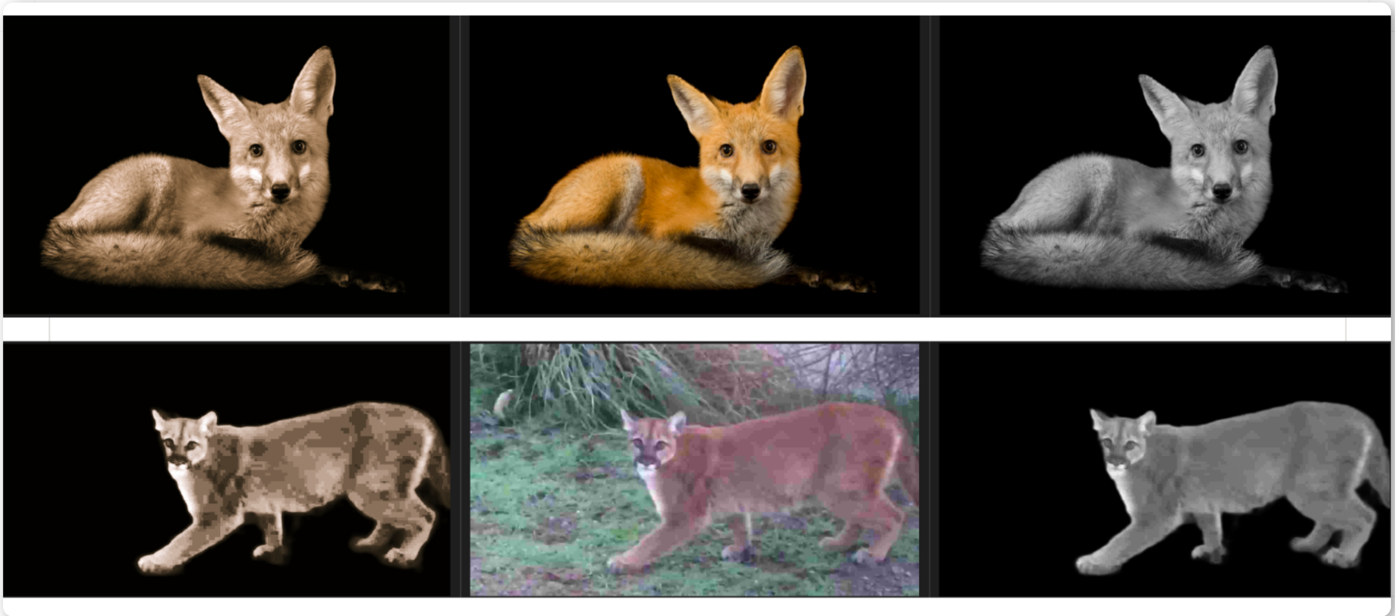
Hence, we suppose we should be able to keep the coloring information by normalizing all the color distribution of reference images. By mapping the grayscale back to RGB channels, we could restore the color information of our image.

One downside is that our images often have a reddish tint, probably because the RGB color space isn't orthogonally aligned. To fix this, we tried using the Lαβ color space. However, the results were worse. For instance, a deer turned blue after the transformation. We're not sure if this happened because we implemented it incorrectly or if the Lαβ color space just isn't a good fit for our approach.

4. Results

4.1 Some of the great results

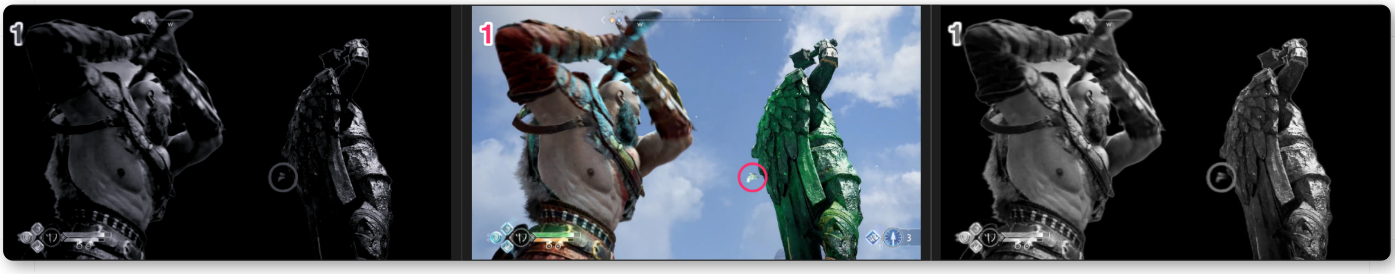


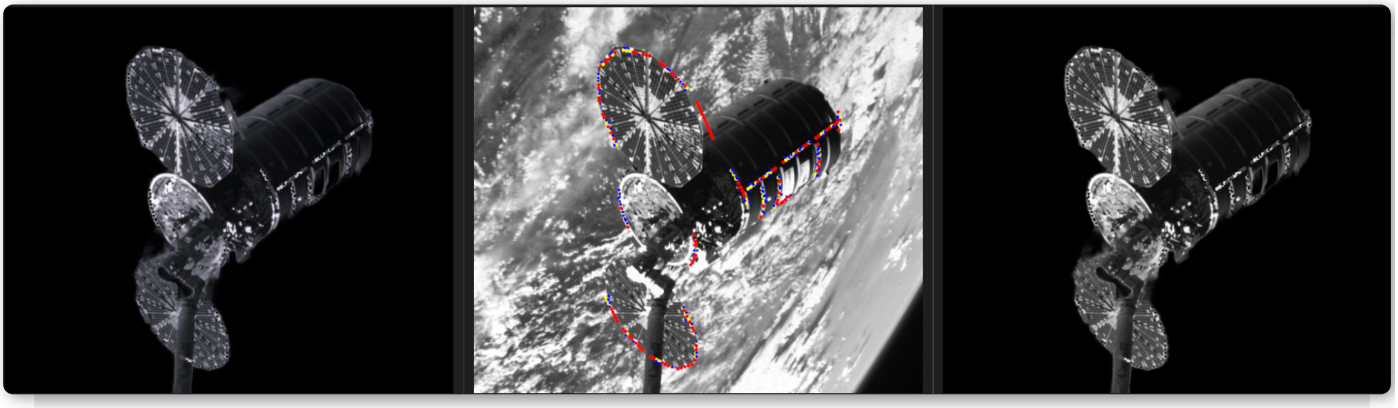


4.2 Some of non-ideal results

The `remove_background()` have not segmented the animal successfully, leading to somehow absurd results.

Raven:





Red Fox:

The dataset contains some strange / miss-labeled image, as a result the information might have been introduced some noise.

5. Conclusion

As demonstrated by the experiments above, the performance of our method depends on the accuracy of image segmentation (background removal) and the quality of the dataset. Although the images appear reddish, we believe that the results significantly enhance the recognition of details. Investigating the reasons behind the reddish tint and the poor performance of the Lαβ color space are areas for future improvement.