Gray to Color

CS 520 Assignment-4

Haotian Xu(hx105), Chaoji Zuo(cz296), Xuenan Wang(xw336)

Spring 2019

Gray to Color

CS 520 Assignment-4

1. Representing The Process

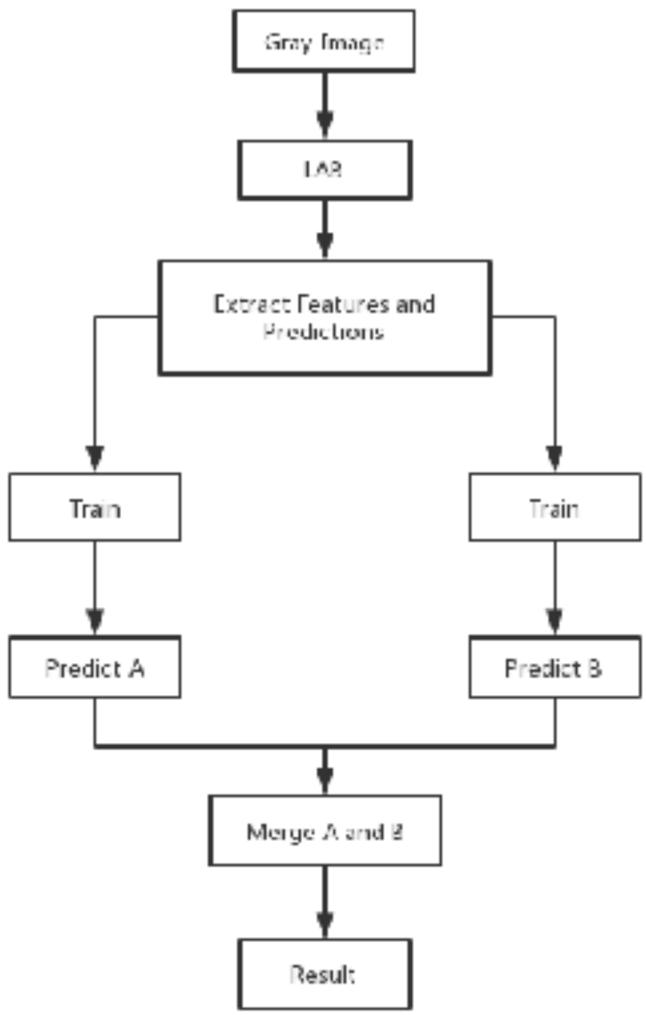


Figure.1 Flow Chart of linear regression approach on LAB color space

2. Data

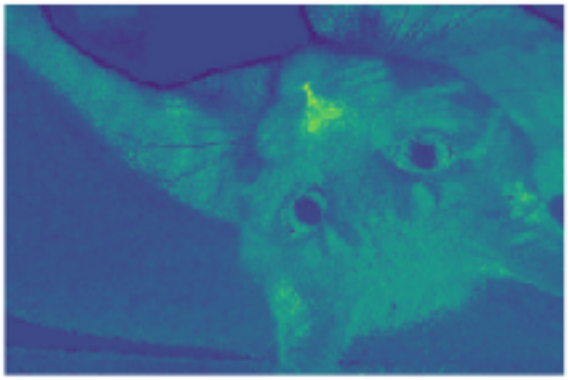
Input data is a RGB image which is actually a 3-D metric. Take Figure 2 for example, channel Red is shown as Figure 3(left), channel Green is shown as Figure 3(middle) and channel Blue is shown as Figure 3(right).

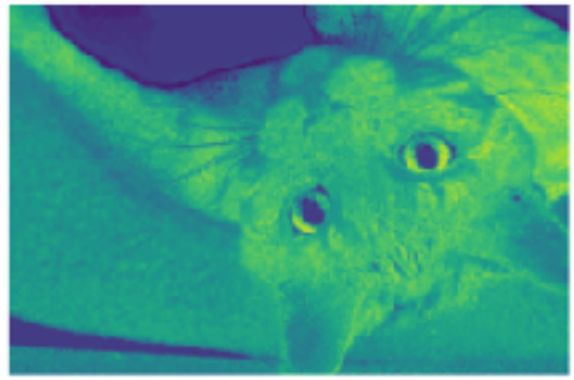
Figure.2 Color Image(Left) v.s. Gray Image(Right)

Figure.3 Red Channel(Left), Green Channel(Middle), Blue Channel(Right)

And we found that if we use LAB color space, we just need to recolor 2 channel a (from green to red) and b (from blue to yellow). In this way, we can do less predication.

Figure.4 a channel(left), b channel(right)





3. Evaluating The Model

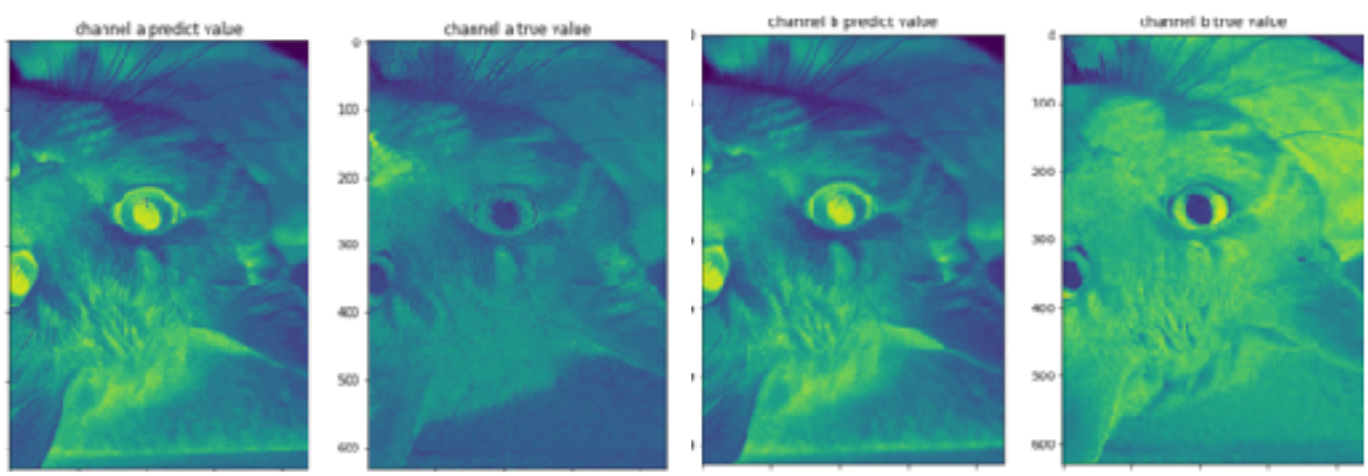
To evaluating the model, the most directly method is watch the output, if a people think the output picture is real, it would be successful. And we also came up with a more mathematical way to evaluating the output, calculate the 2 norm of the subtract of true picture and recolor picture.

And during the linear regression approach, we use R2 score to evaluate the prediction performance.

For the neural network model, we use mean square error to calculate the error during the training.

4. Training The Model

We have different approaches so we have different models, and we have different ways to train those models.



a. Linear Regression approach

Just use the most simple linear regression way, but compare it on the LAB channel and RBG channel.

b. Neural Network approach

We had some trouble in training the neural network using sigmoid function as the activation function, but the gradient always disappear and the weights became 0, even we used the normalized data.

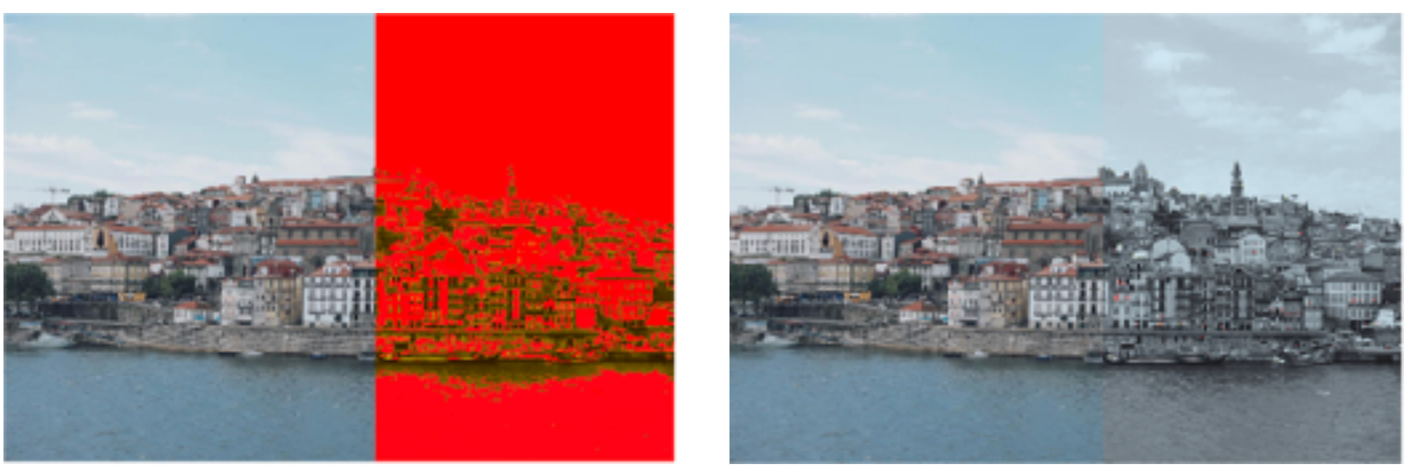
Then we tried to used the tanh function as the activation function. We got a result this time, but the result still seems bad. Usually, all the predicted value would be 1. And we also tried to apply neural network both on LAB and RGB case.

5. Assessing The Final Project

a. Linear Regression approach

Our linear regression approach had a bad performance, it's final 2-norm of true picture and recolor picture is huge. But we still find some interesting things. The performance on LAB color space is obviously better than the performance on RGB color space, which means recolor in LAB space may be a great choice for recolor problem. And we also tried some other linear model such as bayesian regression and ridge regression. But the result did not have a significant change.

What's more, our model is good at recolor single color pictures, but really bad at complex color pictures. Actually, the RGB would show better result than LAB in complex pictures because the recolor became wired.



Left: LAB recolor picture, Right: RGB recolor picture

Additionally, our LAB linear regression model would always make the picture yellow, it just like another kind of "grey". And the RGB linear regression model would just keep the picture in grey. But if we use the real R channel and real B channel, the result would be pretty great. So we had a suppose that the G channel can be great predicted by linear regression.

Left: LAB recolor picture. Right: RGB recolor picture with true r and b.

Left: Linear regression on LAB, Right: Linear regression on RGB

b. Neural Network approach

We also tried the neural network approach, but the result is not great as I mentioned in the model training part. The RGB case just get some strange result. And the LAB case just get some "yellow" result.
An interesting thing we found was that thought the recolor pictures looks wired for us, but theirs 2-norm with the true picture are much lower than the regression case, especially the RGB case. So we thought that maybe 2-norm is not a good value to evaluate the result, since the R, G, B channel may have some interaction. And in regression case, we just take them as independent value. So it may cause a lot of error.

left: Neural Network in LAB, right: Neural Network in RGB.

c. further thoughts

We had the idea of using convolutional neural network to solve this problem, we think it can dig out more useful features from the pictures by applying different convolutional layer, but it's a little bit hard for us to program that by ourselves.

Other Thoughts:

For coloring an image, we as humans will first recognize the which kind of objects that are represented in the image then do the coloring. Therefore, it may help to do a better job if computer first recognize these images.

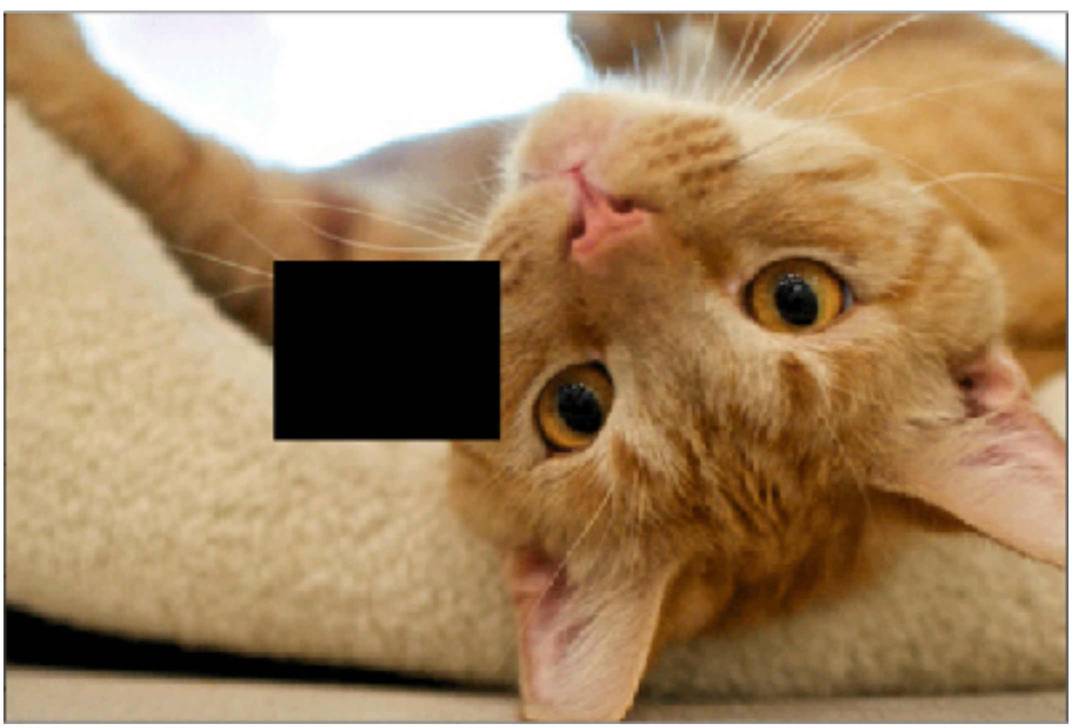
The reason we think image recognition might work is that it will provide more features value and increase the accuracy for the coloring. We first take grey scale and RGB channel value as our feature vectors. If we import another feature vector which would be edges, then we basically doing the recognition and color at the same time. However, edges are not like color features. If we take the binary image, we might result it a bunch white area and black area; therefore, we also need to do image sharpening.

In image recognition-coloring, it might be better if we keep the constancy of our image objects, which means we should take those objects with common colors and shape. For example, a tree would be a great train data, but cars might be not.

Now we have gray scale, binary value, and edge orientation, we can training those feature value separate and then integrate them together which is the same process as above.

6. Bonus

First, to present the hole, we set the value in the center of the picture to 0, it would looks like a black hole in the picture.

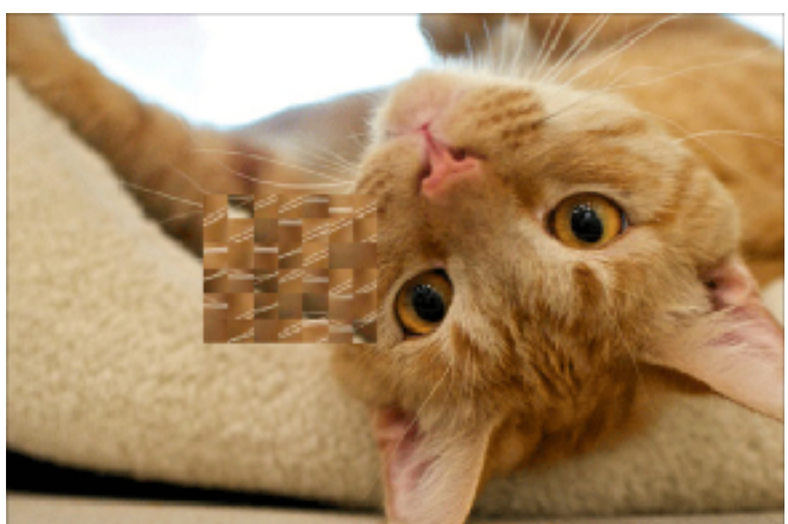


And we also used the 2 norm to evaluate the performance of our model.
To solve this black hole problem, the first idea we got is to use the neighbor area to present the value in the hole.

Then we got such result:
It just put some meaningless blocks in the picture.

Then we tried to enlarge the window of replacing hole.
It still looks wired, but much better. And the 2-norm of this way is smaller than the smaller case.

Also in this problem, which neighbor to use is important, if we choice a similar neighbor, the performance would be a little better.



So to balance the window size and the neighbor is important to black hole problem. And if the black hole is really small, sometime we really can get a good result, like the picture below.

In summary, black hole problem is hard to quantify, because the black hole shape and position is random, which would have a huge influence on the final result. But generally we though using the neighbor to present the hole is a possible solution in some way.

Contribution:

Chaoji Zuo: linear regression approach, neural network approach, bonus.
Haotian Xu: classification approach.
Xuenan Wang: report writing, data processing.