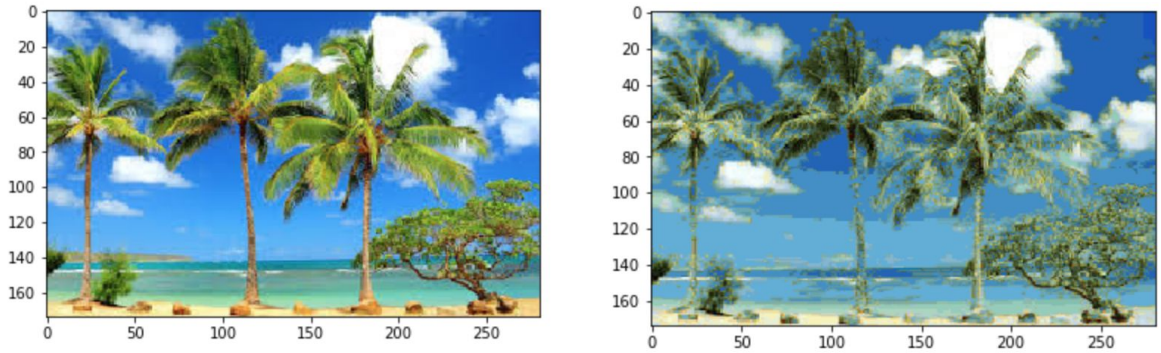
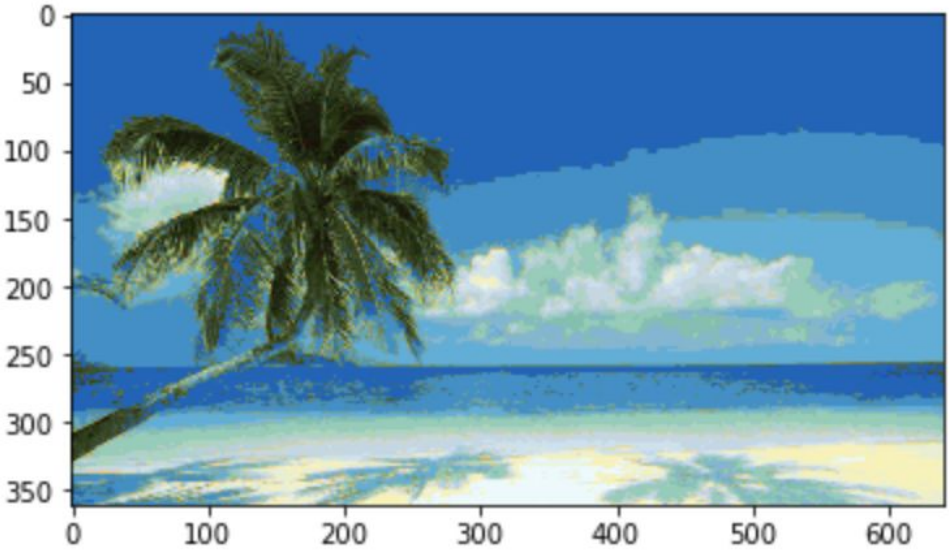
Huy Phan
Prof. Charles Cowan
Introduction to AI CS440
Tuesday, August 15, 2017

Assignment 3



Original image vs Original image converted to grayscale then colored using the learning model.



Testing image colored using the learning model

Quesion 1

Taking the shade values to be integers between 0 and 255, how many possible color values are there? There are 256 posible color values.

How many possible grays? 256 possible grays.

How many possible 3x3 gray pixel patches?

Should a 3x3 pixel patch be enough information to guess the center color, or not enough? Not enough. We need semantic segmentation and object recognition to correctly predict the center color.

Under what contexts might it be enough to try to associate every shade of gray with a specific color, and color an image on a pixel by pixel basis? I don't think there are any contexts we can do that. Converting color image to grayscale image is lossing information. There are many colors that correspond to the same grayscale.

Question 2

Why might it be a good idea to reduce the color-space or the patch-space, for instance by lumping certain shades of green together into one color? We could change a regression problem to a classification problem. Now we only have to match the reduce color-space to the reduce-patch space

What should a good coloring algorithm do if it was fairly confident a pixel should be green or red, but certainly not blue? Pick the result with the highest probability, either green or red. It should not average the value of green and red, because that gives us a different color.

Question 3

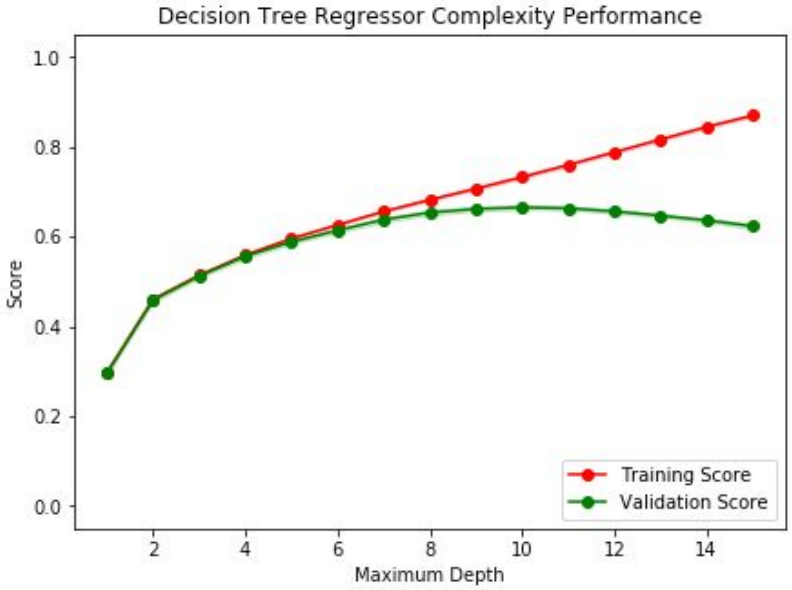
Describe your algorithm, and the design or parameter choices that went into your algorithm. What was your reasoning? What were the difficulties you had to overcome?

After 2 weeks of struggling, I could not implement my algorithm to work with this project. However, I intended to use decision tree to solve this problem. I used the DecisionTreeRegressor from SKlearn.

I chose the max_depth of the trees by looking at the learning curve and testing curve.

When the depth is 1, the model suffer from high bias (undefitting). The the learning and testing score are low, and we can also see that both scores will increase and stay close to each other (not diverge) as the maximum depth increase. Hence, the model is underfitting.

When the depth is 14 the model suffer from high variace (overfitting). The training score keeps getting better while the testing score keeps getting worse. We can see that the curves diverge as the maximum depth increases.



Maximum depth of 10 is best for the model. It's the sweet spot when the 2 curves are just about to diverge.

Question 4

How can you use the provided data to evaluate how good your coloring algorithm is? What kind of validation can you do? Quantify your algorithm's performance as best you can.

I used R2 (coefficient of determination) score to measure the performance of the model. R2 is the proportion of the variance in the dependent variable that is predictable from the independent variable.

To evaluate the model, I used cross reference. I split the given data into training and testing sets. My model performance R2 score is **0.68**

Question 6

Where does your algorithm perform well, where does it perform poorly? How could you (with greater time and computational resources) improve its performance? What parameters or design decisions could you tweak? Would more data on more diverse images help or hinder your algorithm?

Simple to understand and to interpret. Trees can be visualised.

Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values.

The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.

Able to handle both numerical and categorical data.

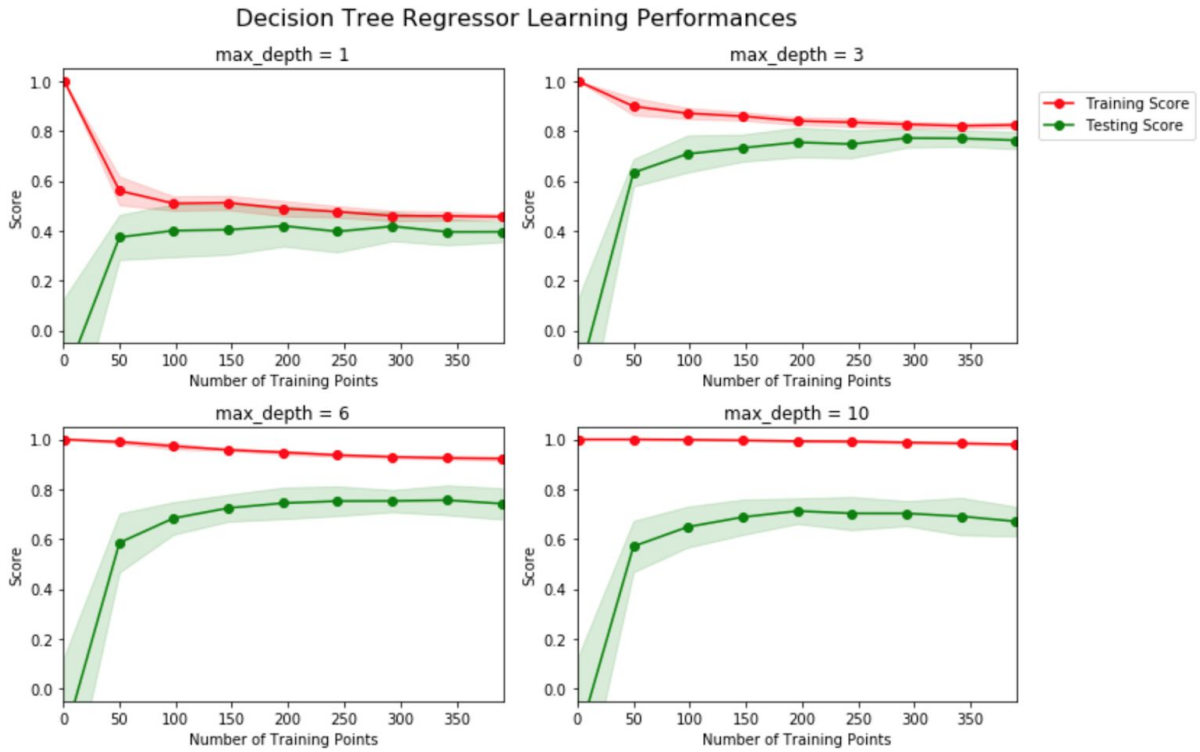
Decision-tree learners can create over-complex trees that do not generalise the data well. As we can see when the max_depth is 14. Decision trees can be unstable because small variations in

the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble.

I think I haved used the most suitable hyper-paremeters for this model. In order to improve performance, I have to use other type of algorithm like support vector machine or neural networks.

More data on more diverse images would hinder this algorithm. We can see that more training data didn't improve the testing score because it has converged to a particular values.

Moreover, more data on diverse images would decrease the performance because for a paricular patches of 3x3 pixel there are more corresponding RGB pixel to match. If we train only on images of beaches it would increase the performance, but if we train on more diverse images, the performance will be reduce.



Question 7

What would happen if you tried to color a grayscale image of a tiger, after training your algorithm on pictures of palm trees?

As I mention before the performance will be decrease. More data on diverse images would decrease the performance because for a paricular patches of 3x3 pixel there are more corresponding RGB pixel to match. If we train only on images of beaches it would increase the performance, but if we train on more diverse images, the performance will be reduce.

Bonus Question

Write your program to take in an image, convert it to grayscale, train your algorithm, and then use the algorithm to color example grayscale images. How does your algorithm do, visually?

It does pretty well with images of the same category.