CS 5785: Homework 4 Write-up

Vianne Gao

27th November 2018

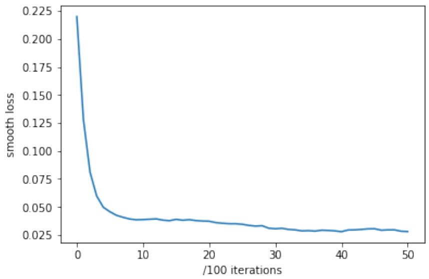
1. Approximating images with neural networks

This question is based on the colored pencil image.

(a) There are 9 layers in total. The first layer has 2 neurons, and is the input layer (x,y coordinates of the pixel), and the last layer has 3 neurons, and is the output layer (r,g,b values of the pixel). The 7 layers in the middle are fully-connected, and each has 20 neurons. They push all ouputs from the previous layer through some function, and output the value to the next layer. The set of all fully-connected layers aim to map the input coordinates to their corresponding output r,g,b values.

(b) The loss shown is a weighted sum of the loss of the current iteration and the previous iterations. The loss for each iteration is L2 distance between output rgb and true rgb, and the loss displayed is the smooth-loss = 0.99*smooth-loss + 0.01*loss.

(c) Loss.



Loss over time. The smooth loss converges around 0.030.

(d) By halving learning rate every 1000 iteration, smooth loss converged to 0.012 after running for 8000 iterations.

(e) Lesion study. Using 4 layers, we were able to achieve acceptable results (20 hidden units in each layer). Using 7 layers, we were able to achieve acceptable results with 13 hidden units in each layer.

(f) Low-rank approximation error.

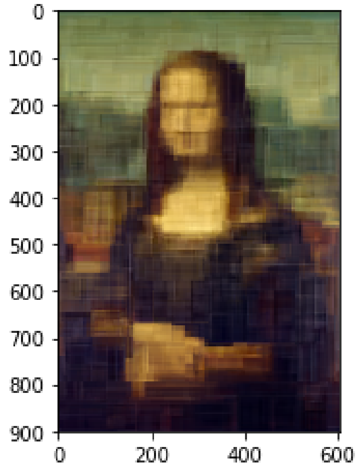
By adding 3 layers to the network, we saw a small improvement in accuracy, but not much. The time taken to complete each iteration increased.

2. Random Forest

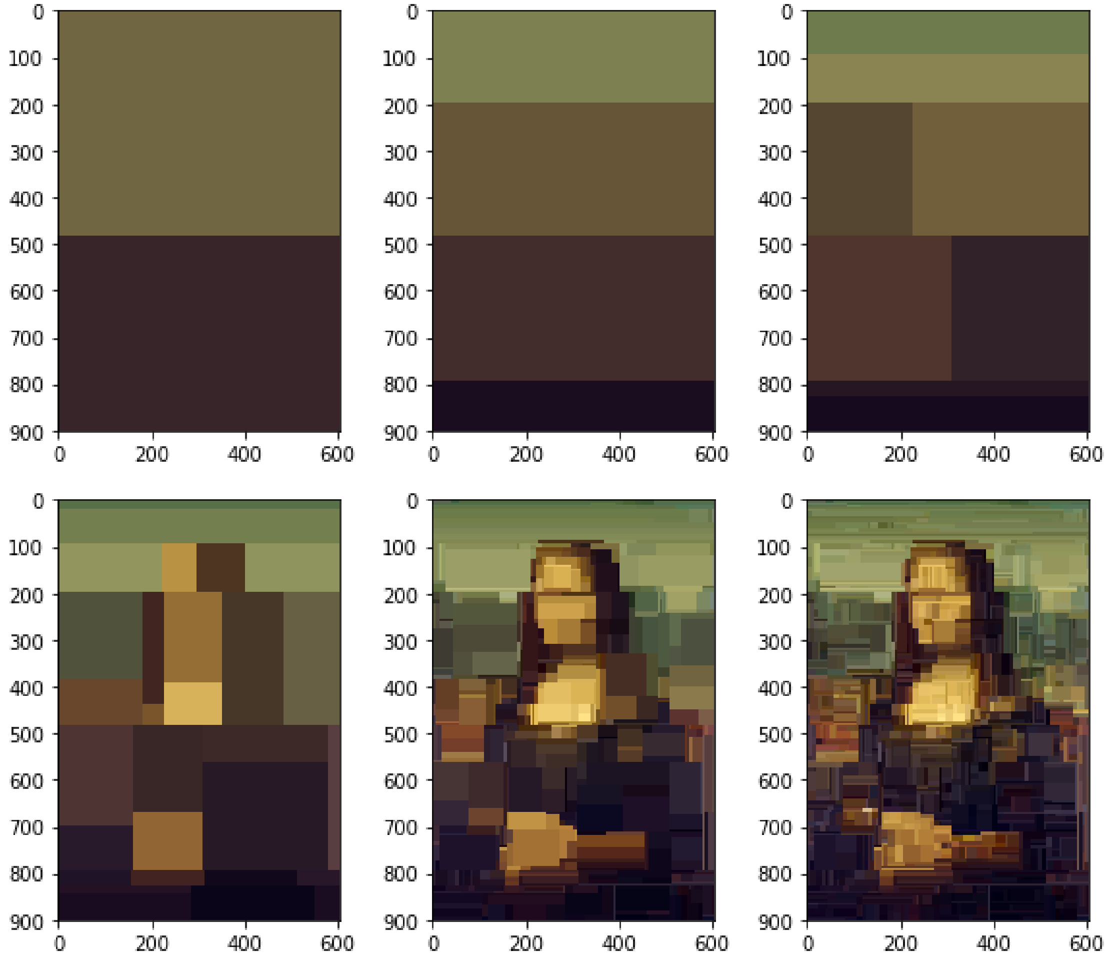
(b) No preprocessing was done to the input data, because our data is discrete and scaling it does not affect the output of any decision trees in the random forest.

(c) We chose to regress all three values at once. Only need to scale r,b,g values to be between 0 and 1.

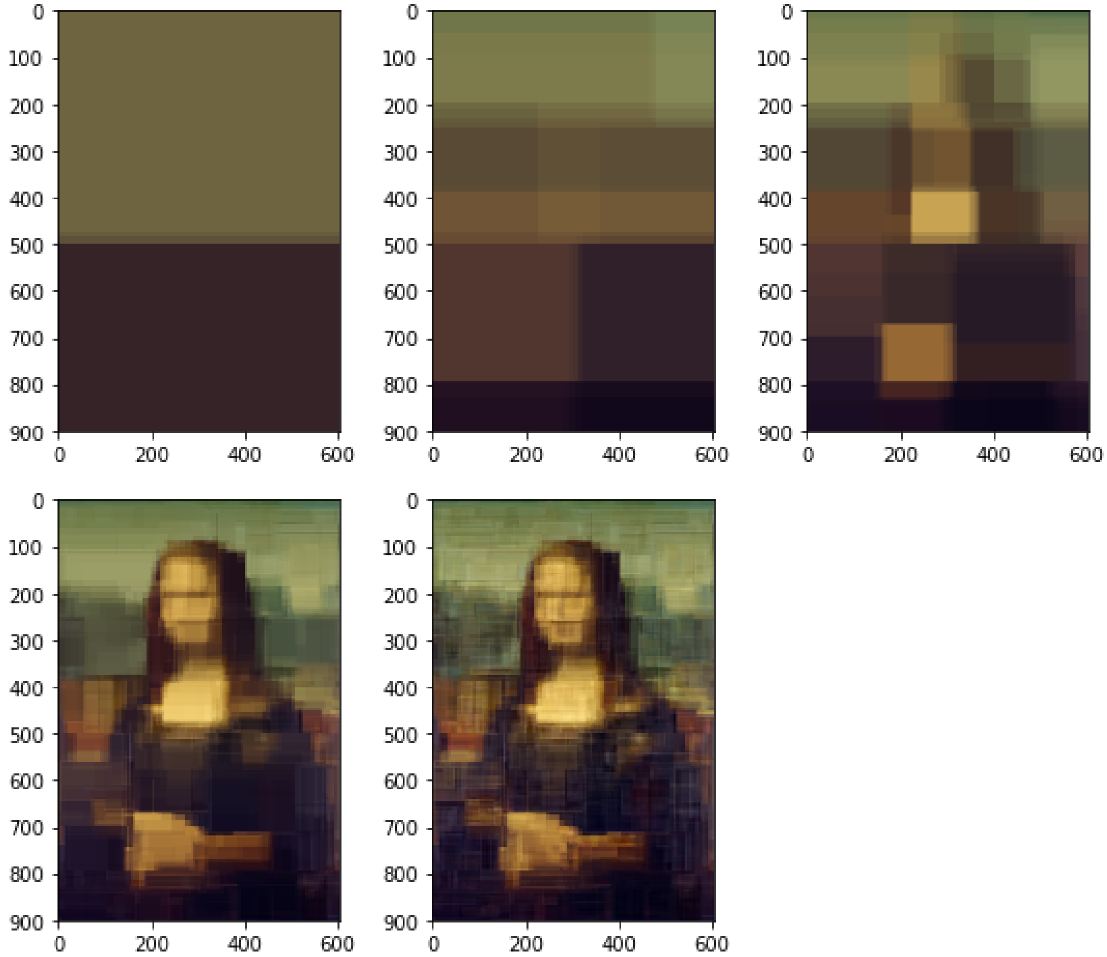
(d) Build image with 10 trees in random forest.



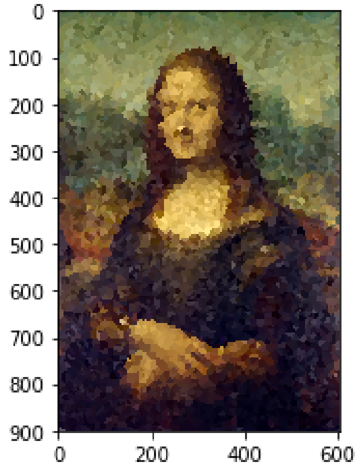
(e) (i)One tree, d layers.



(ii) Seven trees:



(iii) kNN



(f) (i) The decision rule at one of the roots is y ¡= 499.331, where y is the y-coordinate of our input pixel.

(ii) Random Forest:

The output image consists of patches of colors. The patches are rectangular because our decision rules split according to the x-coordinate or the y-coordinate. The colors shown are those that were sampled in the training data.

kNN:

Each pixel is assigned the color of its closest neighbor in the training sample. Since the distance is L2, the decision boundaries are not straight lines but more wriggly.

(iii) Let d be the depth of the tree, there can be at most 2^d color patches.

(iv) Let n be the number of trees in the forest, then there can be at most $n * (2^d)$ choose n number of patches.