Aidan Fischer 12/1/2021

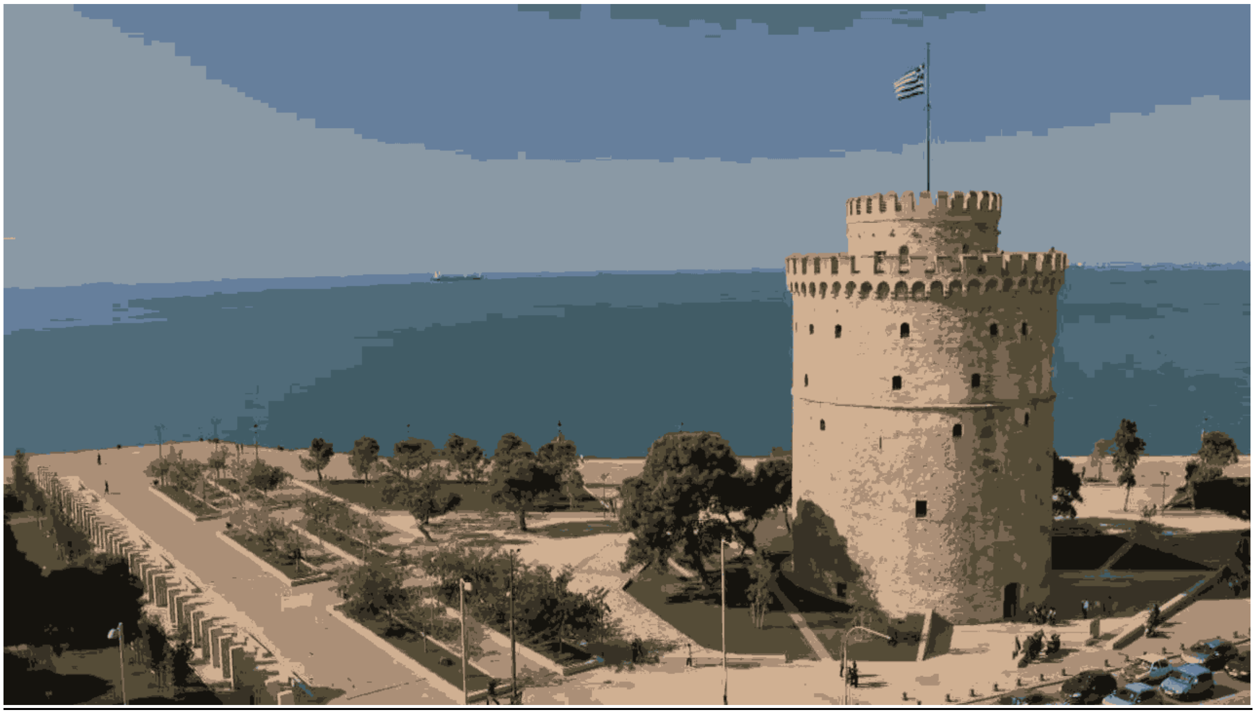
CS558 HW4

I pledge my honor that I have abided by the Stevens Honor System.

**Output Images**

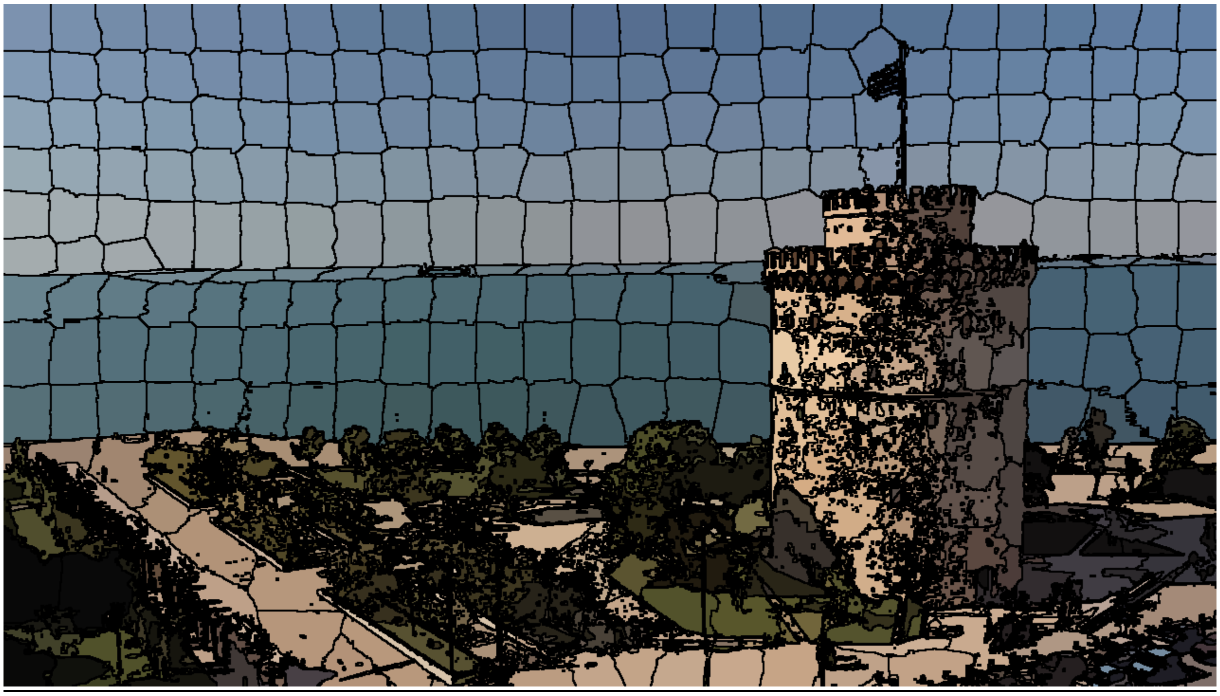
K-Means

white-tower.png

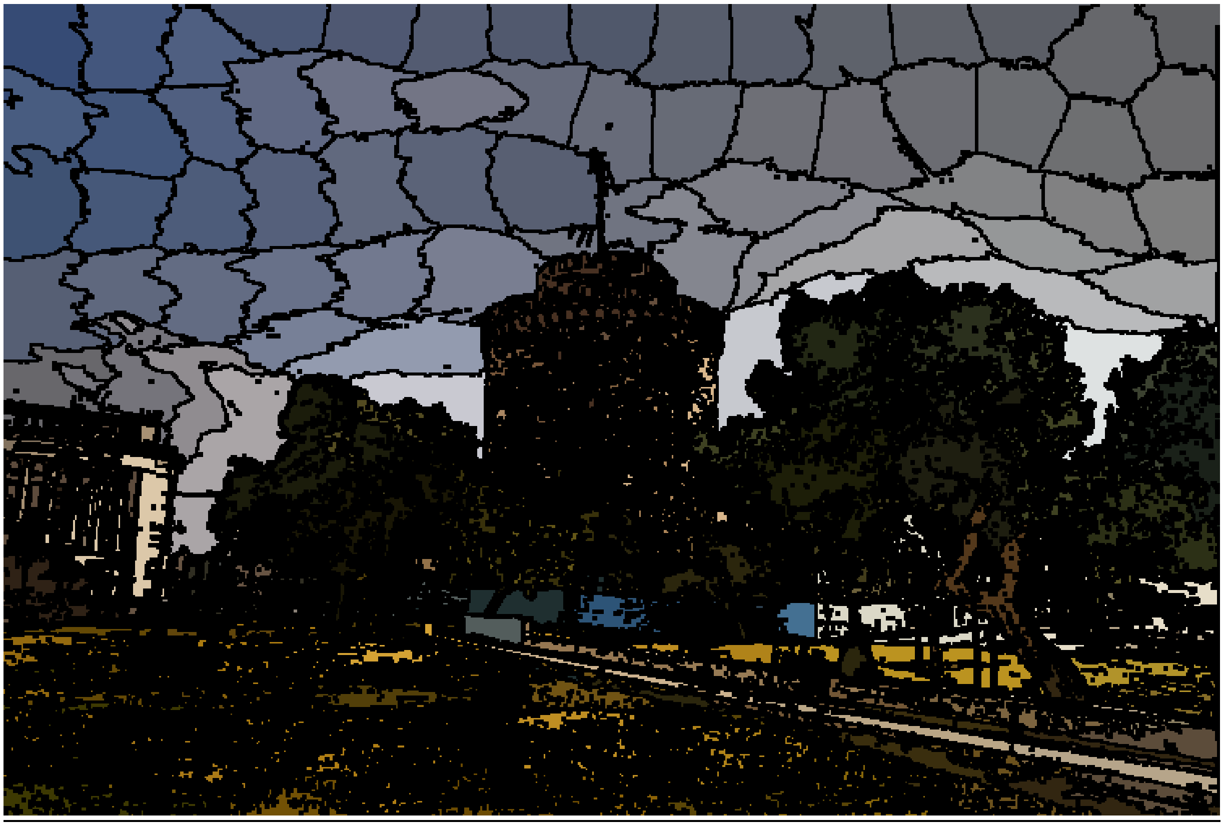


SLIC

white-tower.png

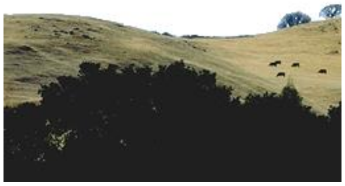


wt_slic.png

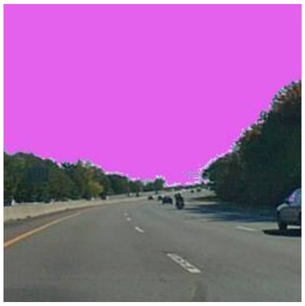


Pixel classification

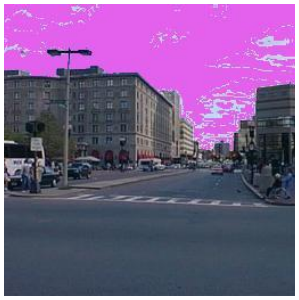
sky_train.jpg - labeled



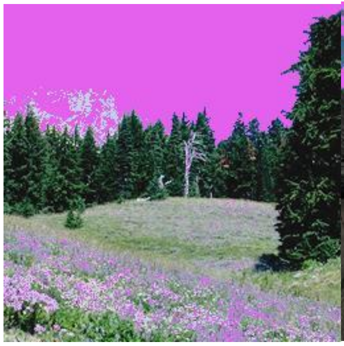
sky_test1.jpg



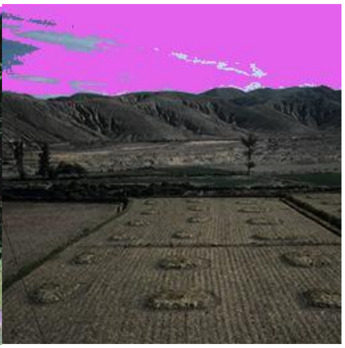
sky_test2.jpg



sky_test3.jpg



sky_test4.jpg



**Implementation Details:**

- Because each problem took a while to run, I only ran one at a time during testing and commented out all other problems, however all have been uncommented in final code submission
- Image.py – wrapper utility around PIL’s Image and ImageOps classes to reduce code duplication. Used in my all my homeworks since I started organizing my code into different modules in HW2
- KMeans.py – implements KMeans for both problem 1 and 3
  - rgb_dst – Determines Euclidean distance between two color triples
  - assign_centroids – Returns a list of quadruplets consisting of (r,g,b,c) where c is the closest centroid assigned to (r,g,b). Set up to allow handling of split training data for problem 3.
  - get_new_centroids – Takes in output from assign_centroids, and calculates the next k centroids from the averages of the assignments. Will ignore centroid assignments of -1, which is a control value used to indicate ignored pixels, implemented for use in problem 3 which splits sky and nonsky pixels into 2 kmeans sets.
  - kmeans_rgb – Runs kmeans on an image, returns kmeans output image as a numpy array.
  - split_kmeans – takes labeled and unlabeled training images, as well as k, and runs kmeans on the unlabeled image, using labeled as a guide to distinguish between sky and

    nonsky pixels. Centroids are assigned randomly as in kmeans, but it makes sure both sky and nonsky have k centroids. Returns a tuple consisting of the list of final nonsky centroids and the list of final sky centroids after kmeans runs to convergence for both sets.

- SLIC.py – Implements SLIC for problem 2
    - init – Returns the initial centroids. Segmentation is implied in centroid locations.
    - local_shift – implements the local shift operation
    - get_pixel_assignment – gets the centroid closest to a given pixel in xyrgb space at a given location in an image. Ignores centroids over 100 pixels away from the given location.
    - COLOR_SCALING_FACTOR – used to normalize pixel locations to color scale
    - update_centroids – updates centroid locations in xyrgb space
    - output – generates final output image
    - do_SLIC – implements SLIC algorithm
- Classification.py – Implements image classification for problem 3
    - BRIGHT_PINK – Color used to indicate detected sky pixels in output images
    - do_classification – implements classification algorithm. Most of the work is done in split_kmeans in KMeans.py, this just does that then assigns output pixel colors based on classification as sky or non_sky from output of split_kmeans and closest detected centroid. Takes labeled training image, unlabeled training image, and an array of input testing images. Outputs array of output images in same order as input images.