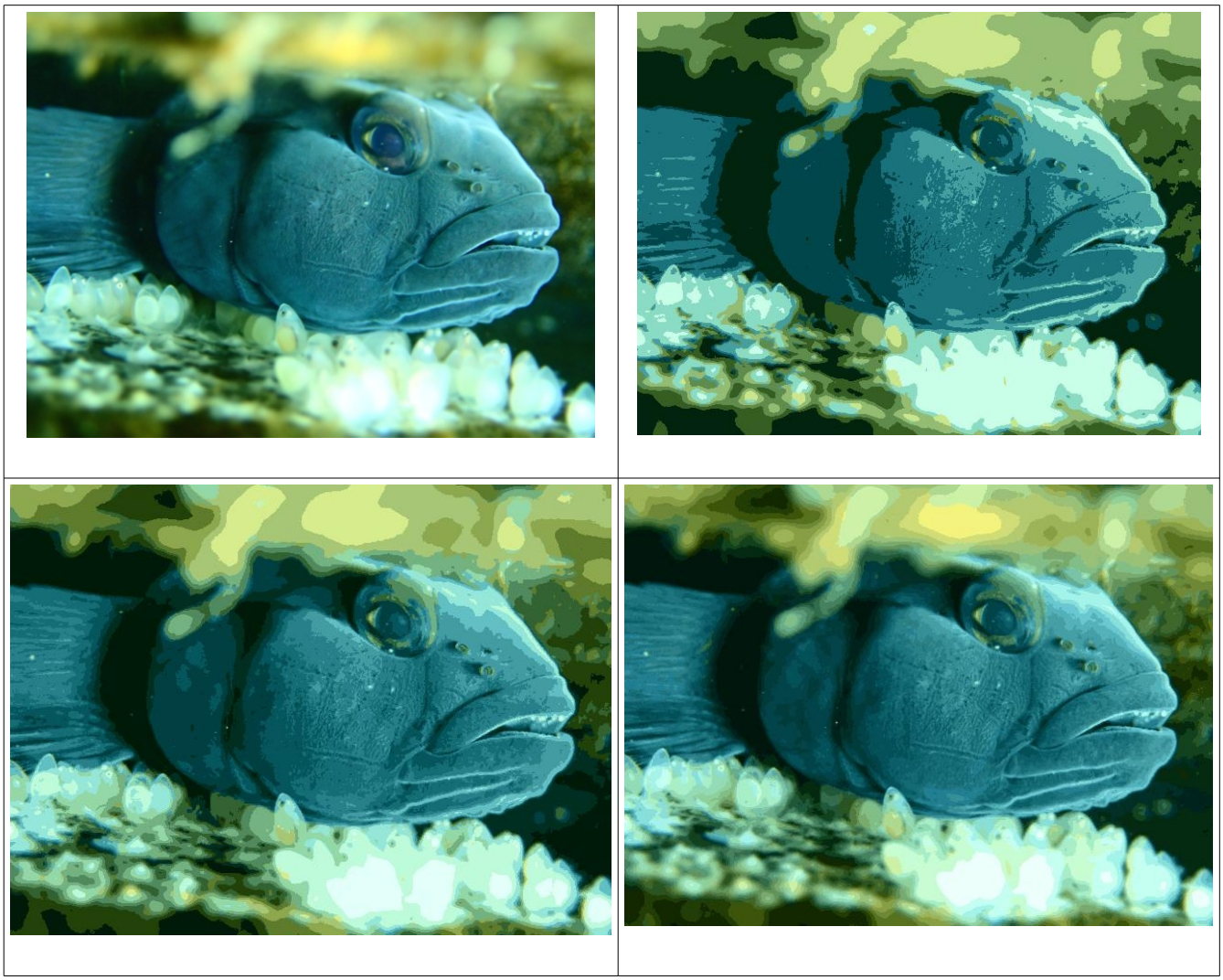
REPORT

Image segmentation using EM. *Use the EM algorithm applied to the mixture of normal distribution model lectured in class to cluster image pixels, then segment the image by mapping each pixel to the cluster center with the highest value of the posterior probability for that pixel.*

- *Segment each of the test images to 10, 20, and 50 segments. You should display these segmented images as images, where each pixel's color is replaced with the mean color of the closest segment*
- *We will identify one special test image. You should segment this to 20 segments using five different start points, and display the result for each case. Is there much variation in the result?The test image is the sunset image*

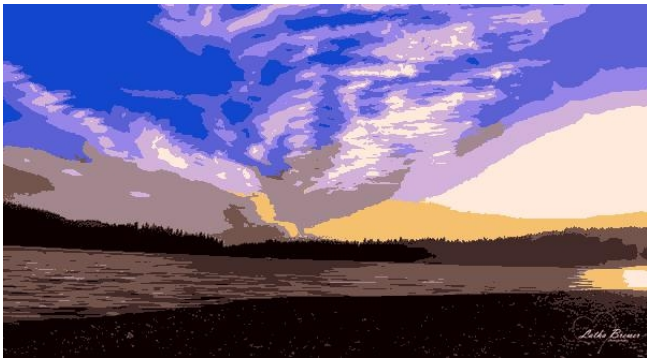
Part 2a. Segment the three test images to 10, 20, and 50 segments.
I have produced three sets of images, one for each of the test images. Top left - original image, top right - 10 segments, bottom left - 20 segments, bottom right - 50 segments. Each of the images from Part 2 (both 2a and 2b) is different from the rest of the similar images which can be better seen if you look at them one after another in a graphic viewer. I have attached the resulting images to this report in case you want to do that.

RobertMixed03.jpg



smallstrelitzia.jpg

smallsunset.jpg



Part 2b. Segment smallsunset.jpg to 20 segments using five different start points. Is there much variation in the result?

Below are 6 images: the original image and segmenting results using five different start points. There are some noticeable variations in the results which can be better seen if you look at the images one after another (the images are attached), but overall there are no drastic variations. You can clearly see that this is the same image, but with different color clusters grouped in a slightly different way. The reason for these minor variations is that the EM algorithm converges to a local minimum, but there may be more than one of them. Actually, they may be a lot of them in a large dataset. Therefore, there is no single best solution, but there are many good ones. If we have points that are located far from all clusters, assigning these points to different clusters (including at the initialization stage which yields different initial cluster centers each time) results in a different set of final cluster centers almost every time the program is run.