**Kalbi Zongo ST 599 Project2 Report   
----------------------------------------------------------------------**

**1. Project /data background**

Planktons constitute the foundation of aquatic foods webs and an important species for ecosystem balance. Identifying the number and types of plankton in ocean is one of good measure of ecosystem balance. However, classical methods for identifying planktons are very time consuming and impractical at plankton population level. The Nation Data Science Bowl has recorded thousands of plankton images that are manually impractical to process in order to identify plankton types. The goal of this project is to come up with an automate system to classify planktons into a given category. The data sets consist of a training data set with plankton categories and a test data set without categories. More specifically, I want to build an automate algorithm upon training data set and use it to predict the class of images from the test data set. Concretely, I wander what the class of each image in my test data set is?

**2. 1 Data handling features extraction**

The data consists of images of plankton species and my first attempt is to extract features from those images. Using the Matlab function extractFeatures, which  returns extracted feature vectors, also known as descriptors, and their corresponding locations, I was able to extract 64 features from each images. The function is set to provide by default these 64 features regardless of the size of image it receives. The function has returned around 4000 Nas’ so I decided to remove those rows from the analysis of training the model. When loading data into matlab, a list of all images was created starting from the image with smallest size to the one with largest size, and I couldn’t automatically attach the appropriate label of image in the list of object. Concretely, I lost the exact label of the training data set after processing features extraction. One way of avoiding this issue would be to load images by specie type and extract features associated to that specie and then attach the corresponding specie type. Time did not permit trying this approach of handling data and extract feature without losing labels. I then randomly assign one of the 121 labels to each data point in my training set.

**2.2 Method and results**

Due to the large size of the training data set for few (64) features, I map features space into a larger dimension by applying 10 degrees polynomial transformation of the 64 features. This returns 640 features as predictors to avoid high bias. I implemented one-vs-all classification by training multiple regularized logistic regression classifiers, one for each of the k classes (121) in the training dataset. The regularized parameter used was lambda = .8. After training the one-vs-all classifier, for each input data point, the classifier compute the probability that the data point fall in class k. And the one-vs-all algorithm predicts the class of the input data point by picking the class with the highest probability. The one-vs-all classifier and prediction algorithms have been written for the purpose of this project. On the training data set, the algorithm has performed as poor as 6.79% accuracy rate due to the loss of exact training data set labels. The images below was read from the list of objects and without knowing their specific class.

  

**3. Obstacles**

Features extraction has been very challenging-I start off with Gabor filter and didn’t get it to work. So I switched to FeatureExtract function. However, this function was not able to process around 4 thousands of images in the training set. The results of this function should be taken with cautious. The choice of the regularization parameter lambda was arbitrary. I should have try many values of lambda and pick the one that give smallest cross validation error. Future works would include but not limited to reprocessing features extraction using methods that will guarantee labeling correctly the data points in the feature space, trying others classifiers such as SVM, Random Forrest, and Neural network. The project github directory will be provided by email by Friday 15.

**References**

<http://www.mathworks.com/help/vision/ref/extractfeatures.html>

<https://www.kaggle.com/c/datasciencebowl>