All the work has been done within the following folder: /home/alexandra/Documents/Image\_Understanding

There are three folders inside:

1. data - this folder mainly contains “surprisingly” just data. Below are the inner folders that might be of interest to you. (I will only describe those that I used and know about)
   1. data/from\_whole\_images\_best\_analysis. I used these (and I also created some of them):
      1. nobins\_reduced\_evalresult\_full\_image.mat
      2. nobins\_reduced\_evalresult\_full\_image\_4algos.mat
      3. nobins\_reduced\_evalresult\_full\_image\_5algos.mat
      4. nobins\_\_reduced\_false\_evalresult\_full\_piece.mat
      5. nobins\_\_reduced\_false\_evalresult\_full\_piece\_4algos.mat
      6. nobins\_\_reduced\_false\_evalresult\_full\_piece\_5algos.mat

First three files are matrices that have names of images, segmentation results or 3, 4 and 5 algorithms, and image features (not CNN features but old features which are brightness levels, sift, gist, etc).

Last three files contain the same thing but for objects.

These files were used in the process of training using old features. Best algorithm is wherever the the field ‘segmentation’ is greater.

* 1. data/JPEGImages - inside you can find object images along with their modifications (you just need the original images themselves, such as 2007\_010032.jpg, 2007\_010033.jpg, etc.)
  2. data/Multi\_Algo\_Final\_Results contains semantic segmentation results by the 5 available (to the date) algorithms in the folders: ale\_val\_cls, cpmc\_val\_cls, long\_shelhamer\_val\_cls, sds\_val\_cls, wdo\_val\_cls.

1. Regions - contains the segmentation results (images) obtained with our algorithm.
2. deep\_learning - this is the folder in which I was mainly working in the second part of my directed study.
   1. deep\_learning/features folder contains CNN features for images. The names are corresponding to image names.
   2. deep\_learning/obj\_features folder contains CNN features for image objects. The names are corresponding to image object names.
   3. deep\_learning/images\_voc2012 folder contains all full images that were available. I just copied them over here to have a quick access to them. You probably won’t need this folder.
   4. filenames.mat - list of images for which all five algorithms had results available (there were 1441 such images in total).
   5. dataset.mat - CNN features for those 1441 images. The CNN feature matrix by Caffe is 4096x10. It was transformed into 40960 x 1 and if all fields in all images were 0, they were skipped. Otherwise, they were copied to the dataset matrix. In that way we got 6470x1441 dataset matrix. (More explanation and coding in analysis\_nn\_cnn.m)
   6. dataset2.mat - CNN features for those 1441 images. This one was obtained by averaging by rows. So each 4096x10 CNN feature matrix was transformed to 4096x1 by averaging and then saved into dataset2 whose final dims are 4096x1441. (More explanation and coding in analysis\_nn\_hypo\_cnn.m)
   7. target.mat and temp\_target.mat contain numbers indicating the best algorithm for each of the available images. temp\_target contains the algorithm numbers and target contains the same thing but in 0/1 format which is required by the neural network toolbox.
   8. voc2012\_mean.mat is the mean image for the available 1441 images in W x H x C with BGR channels; such mean data is required by the Caffe network. This matrix was computed by compute\_mean\_image.m
   9. experiment\_run.m compares performance results from the specified folder (insert the path to the folder that contains our algorithm’s results) and the other 5 available algorithms.
   10. extract\_features.m and extract\_object\_features.m are the scripts that extract CNN features and save them into ./features and ./obj\_features folders respectively. They both use get\_cnn\_features.m (look below)
   11. get\_cnn\_features.m the function that extracts CNN features! The key line is this: features = net.blobs('fc7').get\_data(); It means it gets the data from the 7th last network layer.
   12. contradiction\_check\_accuracy.m - the script that finds the accuracy of contradiction check. It is a separate script and it wasn’t used/needed by any other files.
   13. nn.m - a neural network function generated by the MATLAB neural network toolbox.
   14. analysis\_nn\_cnn.m, analysis\_nn\_cnn\_oldfeat.m, analysis\_nn\_cnn\_stat.m, analysis\_nn\_hypo\_cnn.m, analysis\_nn\_oldfeat, analysis\_svm\_cnn\_oldfeat are the files which ran the experiment itself. I hope that the names will help you to navigate. We only used two ML algos for classification which are either NN or SVM, this is indicated in the names. We used different feature types in the experiment which are CNN features or old features (see section 1-a) or both; this is indicated in the names as well. ‘hypo’ means that hypothesis generation and iterative analysis was involved, ‘stat’ means that the statistics was used if there was found a contradiction. By statistics we mean using the algorithm that statistically works the best among all others for the identified type of object.

Data Creation

Training and testing data

**Generate Features of all images:**

a) to create features from each object run: generate\_data\_textures\_for\_bn\_training\_bb.m to extract features

b) to create features for the whole images run: generate\_data\_for\_whole\_images.m

**To create training data set use two scripts:**

a) evaluate\_data\_from\_full\_img.m to determine best algorithm for each whole image in the folders available

b) evaluate\_data\_from\_full\_img\_to\_bb\_full.m to determine best algorithm for each object in each image in the folders available