Contents

Table of Figures

PART 1: Data preprocessing & feature generation

HOW TO RUN OUR CODE?

1. *Run trainingExemplars.m (performs feature extraction on training images and saves features in training.mat)*
2. *Run validationExemplars.m (performs feature extraction on validation images and saves features in validation.mat)*
3. *Run main_ver2.m (trains SVM classifier with training.mat, and classifies objects in validation images using validation images features stores in validation.mat. Then performs feature extraction on test images and classifies each object)*

After looking at all the spores (target class) and non-spores (clutter class) in the training images, two attributes that seem to differentiate the classes are:

1) Spores are circular in shape compared to other objects in microscope images. That is, spores tend to have a constant radius around its centroid.
2) Spores tend to have dark colors (except for one immature spore), or at least, a high dark to bright color ratio. In other words, if we looked the histogram of every spore, we would expect to see the majority of the pixel values to fall on the left side (values close to 0 or black) of a histogram.

The following image (training1.tif) highlights these differences. After preprocessing the image, objects that survive some thresholds are assigned a number (in red and blue). Using the truth data provided by the professor, we color coded the labels as blue (spores) and red (non-spores):

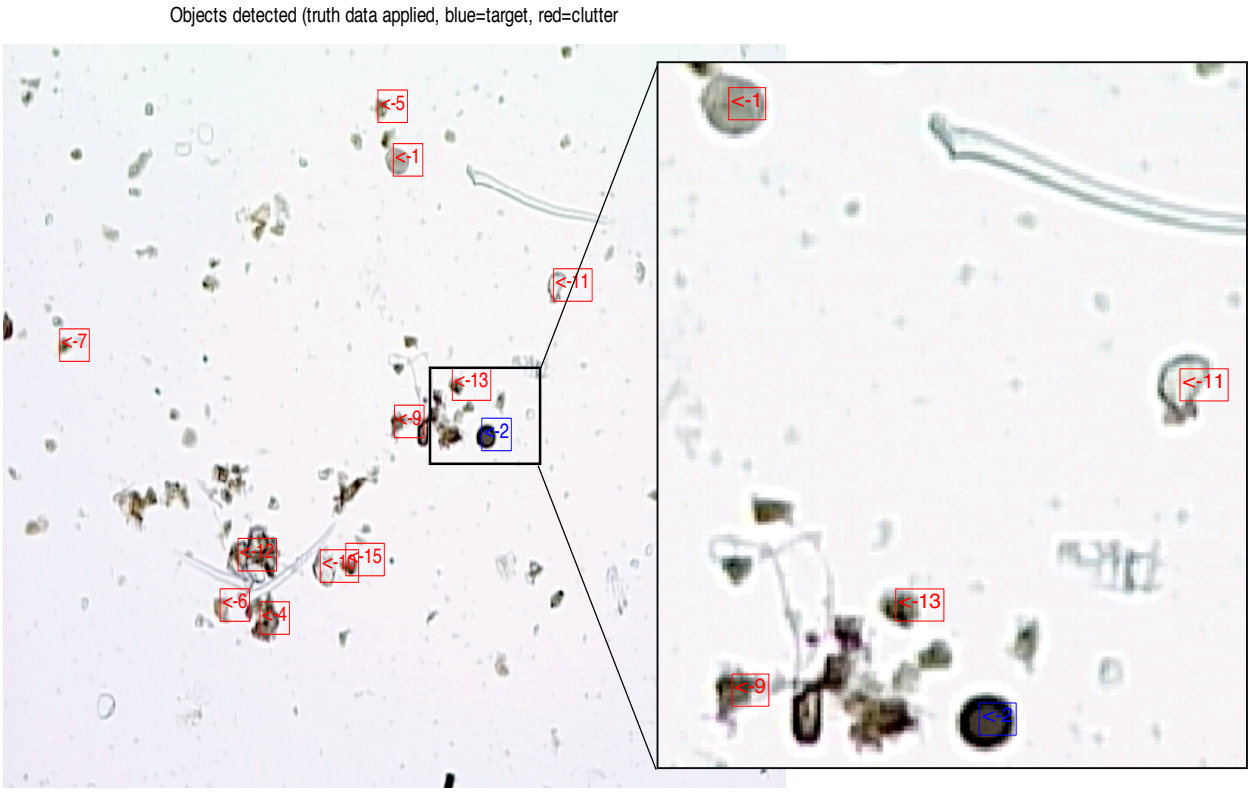

Figure 1. Spore characteristics

Preprocessing of the images consists of creating a threshold black & white version of the image. A second algorithm is then used to further eliminate small objects. The result (shown below) is an image containing the surviving objects:

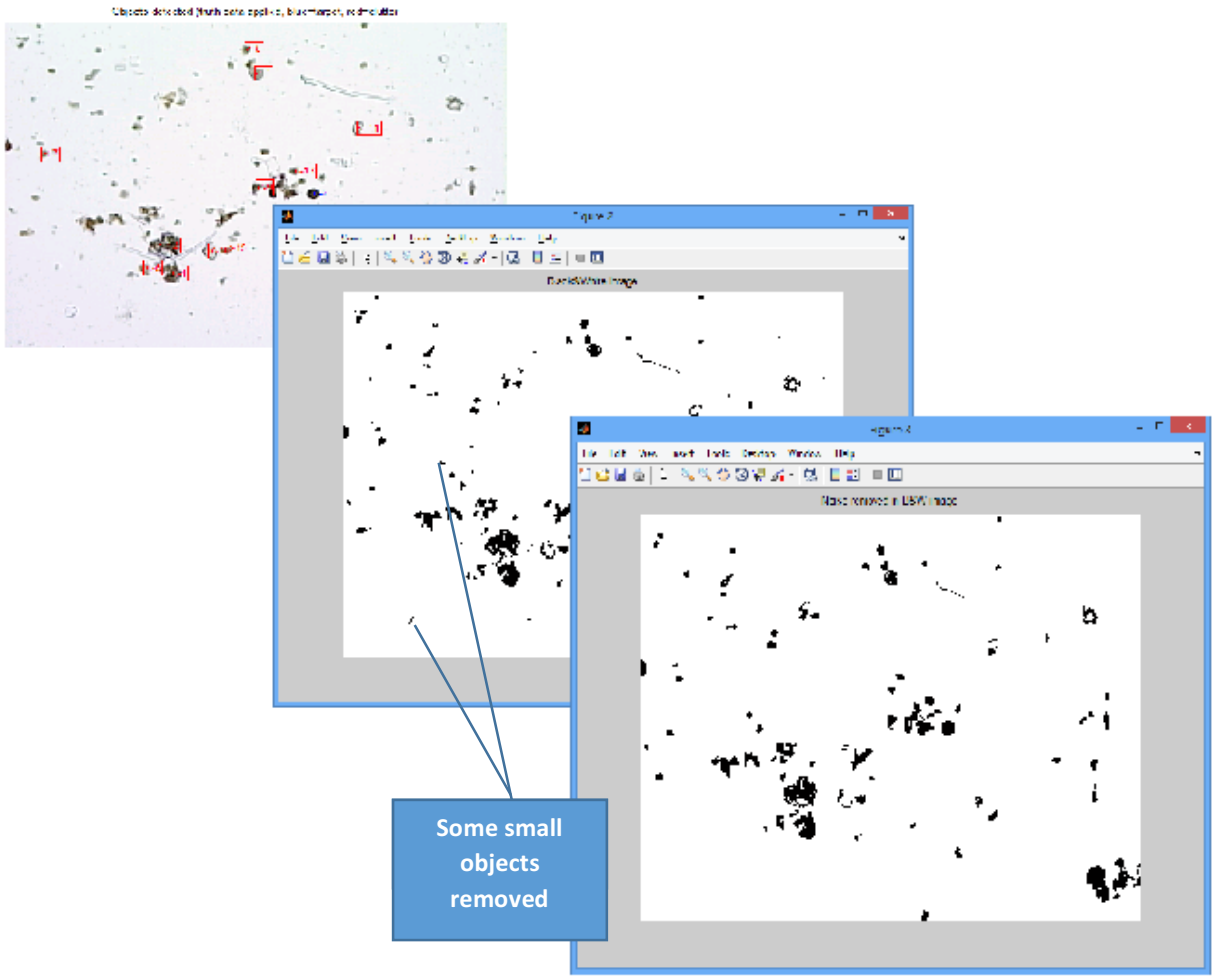

Figure 2 Image processing

The last step in the image preprocessing is done in conjunction with the feature extraction step. Because most spores are circular, we decided to leverage existing algorithms (imfindcircles.m) that had the capability of a) finding circle-like objects on an image and b) extracting attributes associated to them. This algorithm can be set to look for circle-like objects that have specific radius (or a range). After inspecting the training images, we determined a reasonable radius range to look for spores was 6 to 15 pixels. Not only does this algorithm discard objects that are not circle-like, but it also rejects circle-like objects that have a radius outside the specified range.

Returning to the feature generation discussion, most spores are dark in color and have a constant radius. The algorithm that finds the circle-like objects computes how “circle-like” an object is. In other words, the more constant the radious of the circle-like object is, the closer this value will be to 1. This is one of the two features that we computed for objects of interest. The range of this feature is 0 to 1 inclusive.

In regard to the darkness observed in spores, once the circle-like object is identified, the algorithm also computes the object's centroid. This centroid, along with the estimated radius, is used to apply a mask to the original tif image. This allows us to compute a histogram and determine how spore-like the gray levels in the object are. We recognize this may be difficult to visualize, so we took an extra effort to explain this with visual aids:

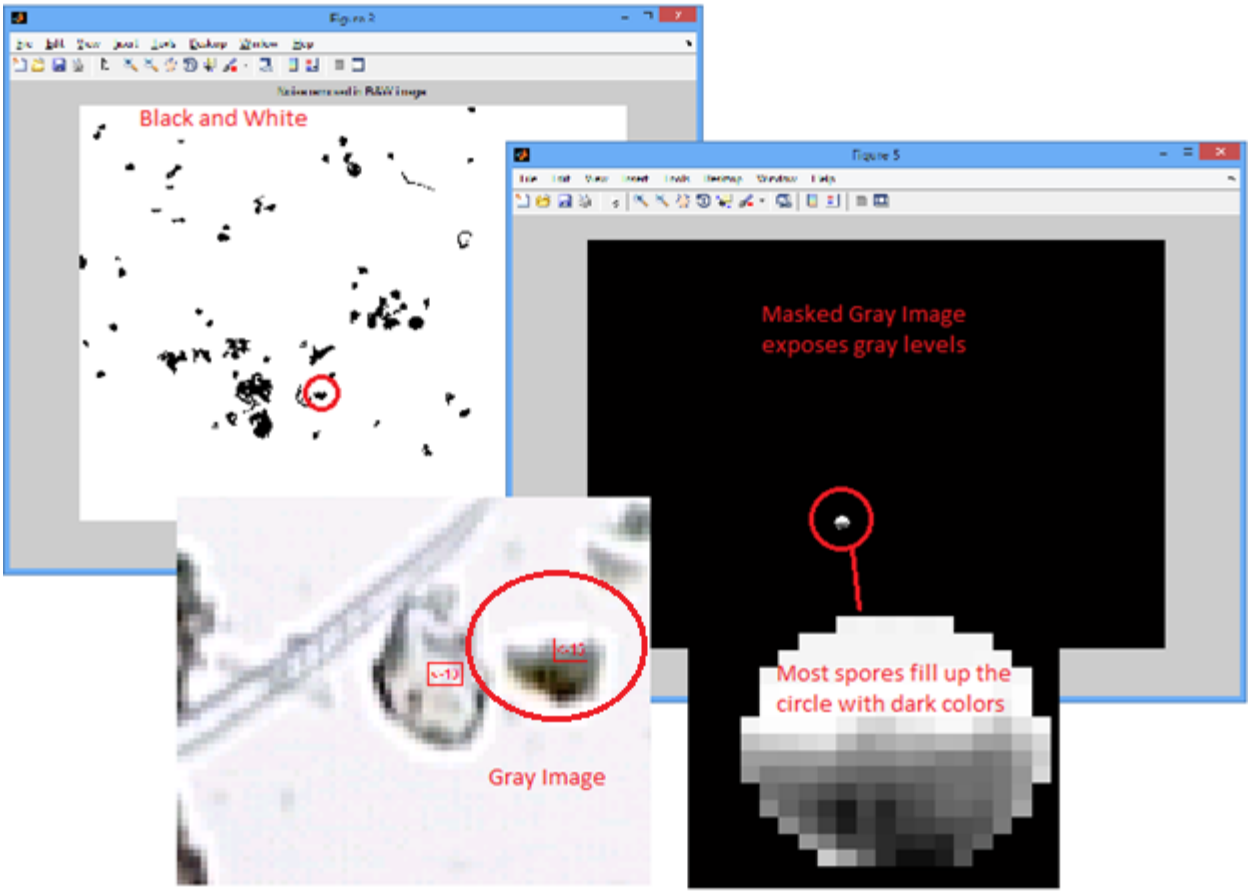

Figure 3. Feature extraction

The first step is running the algorithms that finds circle-like object on the threshold black and white image shown above (top left). This generates several objects of interest, including object #13. Using object 13's computed centroid and radius, a mask is applied to an image (gray level version of original image) exposing only this object. Then, a histogram on the image is computed. We determined that the best way to capture the classes' differences was to define feature #2 as the ratio of dark colors to bright colors. To this effect, we defined dark colors as pixels with values less than 50. Bright colors are defined as pixel colors above 151 (maximum and minimum values are 255 and 0).

This summarizes the two features used by the SVM classifier: 1) circle-likeness of an object, and 2) histogram ratio (or the ratio of number of pixels below 50 and above 151).

After running this feature extraction process on all the training images, we produce the following plot which shows the reparability of the target/clutter classes in the training dataset:

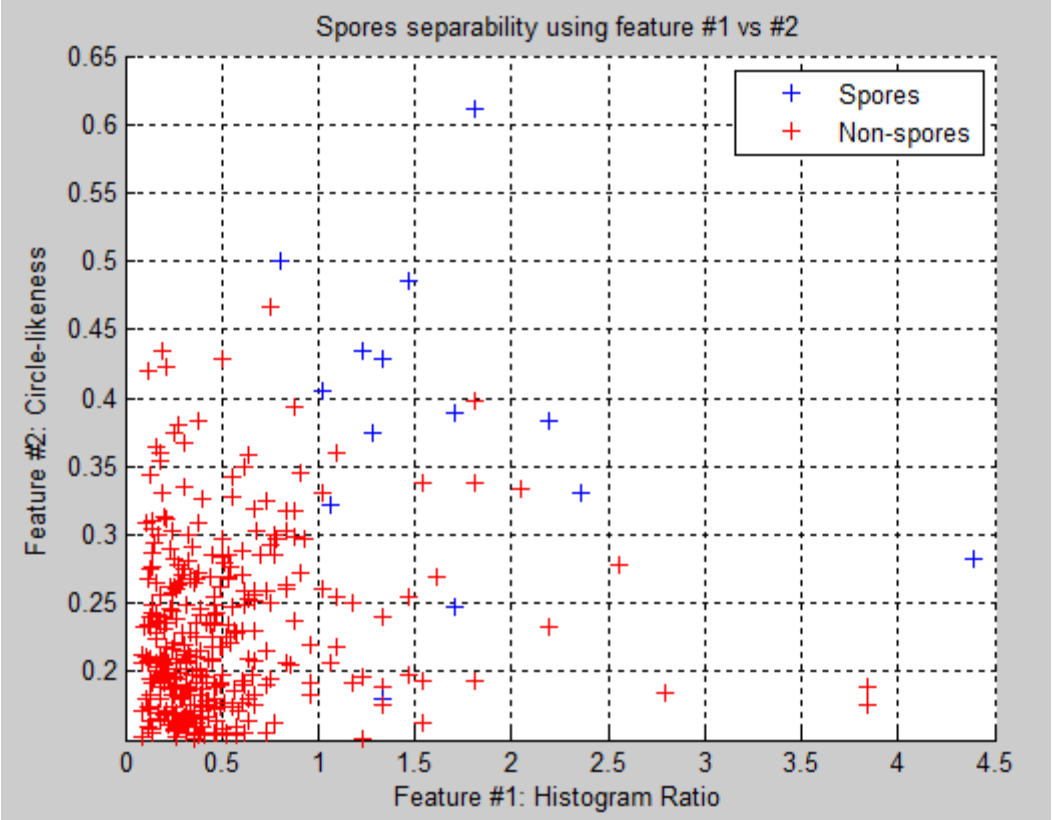

Figure 4 Class separability using features

It can be seen that this is a case where classes overlap. Also, most spores have high circle-likeness values and high histogram ratios when compared to non-spores. Feature #2 is self-explanatory: they are simply not rounded in most cases. Feature #1 is also easy to understand and interpret: most non-spores have a LOW count of dark pixels when compared to bright pixels, just as we had visually seen.

PART 2: SVM training & validation

After extracting the features, a training feature file was generated (training.mat). Using this file we proceeded to train the SVM. But before this step, some important questions had to be answered: what kernel should we use, what smoothing parameter "C" value generates a margin that is acceptable to us, what "method" for the provided Matlab function should we use?

To quote Theodoris and Koutroumbas in Pattern Recognition (4th Ed.) "A major limitation of the SVMs is that up to now there has been no efficient practical method for selecting the best kernel function. This is still an unresolved, yet challenging, research issue".

Due to time constrains and for the purpose of simplicity, we decided to limit our analysis to the linear kernel, and "Platt" method. Looking at the separability plot, it is reasonable to assume that a linear SVM will suffice for this classifying task. Also, the method Platt was used. The parameter we chose to optimize is C. To this end, we ran the training data for different values of C. Note several error types have been computed and are defined after the table:

	C (method: Platt, kernel: linear)							
Value of C	0.2	0.8	1	5	20	200	500	1000
Pd_training	100	100	100	79	79	79	79	79
Pd_validation	92	100	92	58	58	58	58	58
Pfa_training	14	14	12	5	5	5	5	5
Pfa_validation	11	11	11	4	4	4	4	4
Margin	0.69	0.16	0.10	0.004	0.0005	5E-6	8E-7	2E7
Alpha	7	4	4	4	3	3	3	3

Pd (d=detection) = probability that an exemplar is classified as spore and it is a spore (P(1/1))
Pfa (fa=false alarm)= probability that an exemplar is classified as spore, but it is not a spore (P(1/0))

In this case, the goal is to have a high Pd and low Pfa for both the training and validation datasets. The best combined Pd is achieved when C=0.8. The lowest combined Pfa is achieved when C>=5. Since we place more importance in detecting spores, than having false alarms, we chose C=0.8.

The SVM classifier parameters after training are w = [3.2949 1.3279], and b=2.9945.

PART 3: Spore-like objects in validation & test images

The location and discriminant function value of the spore-like objects for each image are listed in the table below:

Spore-like objects									
		Validation Dataset					Test Dataset		
object #	Image #	Location (X, Y)		Discriminant Function value		Image #	Location (X, Y)		Discriminant Function value
1	1	347	315	1.856998989		1	551	25	2.753746894
2	1	300	168	0.476305595		1	541	17	3.713970946
3	1	312	365	0.468462468		2	619	78	1.204317483
4	1	223	103	0.054631897		2	261	101	0.115398088
5	2	509	444	17.75949299		2	536	406	3.122884091
6	2	23	385	2.79151723		2	115	66	8.71671044
7	2	360	324	1.16776647		2	471	160	0.466594382
8	2	454	24	1.018987514		2	543	418	17.62093573
9	2	7	139	1.796393819		2	215	439	0.72728755
10	2	386	44	0.900727474		2	172	212	2.666673711
11	2	514	379	0.754109269		2	99	416	0.394404268
12	2	65	327	0.026757201		3	214	255	17.94205606
13	2	609	3	98.3637867		3	201	381	0.706429205
14	3	138	95	1.249858769		3	25	301	4.107128333
15	3	418	45	6.769696982		3	328	48	0.506355701
16	3	84	320	0.018922948		3	343	54	0.231976575
17	3	47	361	0.203057508		3	116	293	0.367812228
18	3	375	352	1.66751869		3	537	317	4.046439327
19	3	444	16	0.8539144		3	507	476	0.653672925
20	3	442	322	1.270092668		4	478	86	0.173254299
21	3	120	234	0.821431171		4	479	380	1.435234366
22	4	9	184	7.553139359		4	46	466	1.573115659
23	4	435	191	0.441485133		4	330	453	2.168850681
24	4	344	107	5.797607679		4	579	135	0.2372984
25	4	208	335	0.864467651		5	630	432	12.00815966
26	4	571	233	0.48972926		5	241	440	0.05763294
27	4	147	359	0.863148459		5	30	125	1.12882464
28	4	359	24	0.96363158		5	618	105	0.460893829
29	4	67	425	1.747284465		5	47	365	2.19355848
30	4	171	169	0.561017868		5	153	287	1.390933279
31	4	408	346	1.511012357		5	288	73	2.380397371
32	4	538	431	8.536198048		5	153	123	1.068093628

33	4	343	287	0.904263632		5	64	406	1.051139473
34	4	180	40	0.661159014		5	373	253	0.551193615
35	4	193	204	1.685887759		5	193	335	0.517141855
36	4	178	58	0.656799363		5	170	412	0.12813495
37	4	30	404	0.654217669		5	85	188	0.186007678
38	4	349	148	3.21203624		6	559	60	0.426698093
39	4	289	314	2.077288933					
40	4	106	9	5.039008084					
41	4	637	450	2.039096688					
42	4	26	111	5.003690642					
43	4	550	236	1.262464493					
44	5	310	128	0.888219433					
45	5	122	322	1.678525231					
46	5	348	258	1.630361382					
47	5	268	138	0.967971769					
48	5	226	266	48.05663556					
49	5	117	360	2.346835277					
50	5	468	104	0.153894194					