Digital Image Processing Experiment 2

Team members and responsibilities:

Xiao Cui (LBP code, RGB histogram code, classification code, final report)
Ying Xu (Gabor code, experiment proposal, final report)

Introduction:

Medical imaging technique is developing fast over the past few decades. Automatic recognition and classification of histopathological images has been a hot field in both research and practice. As hundreds of images could be generated from a medical imaging device per day, it could save physicians' time and labor force by analysing images on the machine.

In this experiment, bovine lung H&E stained images are going to be learnt. Two sets of training data are provided: 15 pictures contain inflammation lung cells and 15 pictures contain normal lung cells. 20 testing pictures are going to be classified based on features from training data. The goal is to automatically recognize pictures with and without inflamed cells.

Methods:

Figure 1 and Figure 2 are used to represent all processes in we did to classify inflammation and normal pictures. Figure 1 is the flowchart of feature extraction in Matlab and Figure 2 is the flowchart of classification in Python.

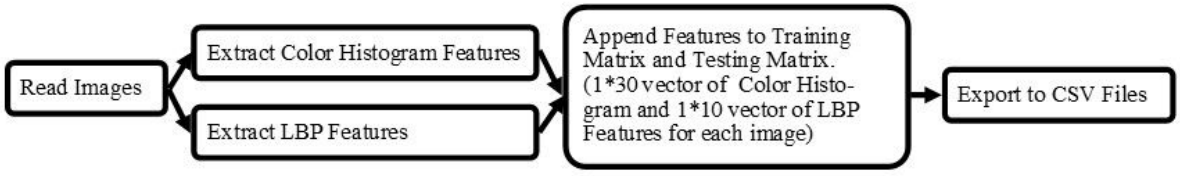

Figure 1. The feature extraction process in matlab

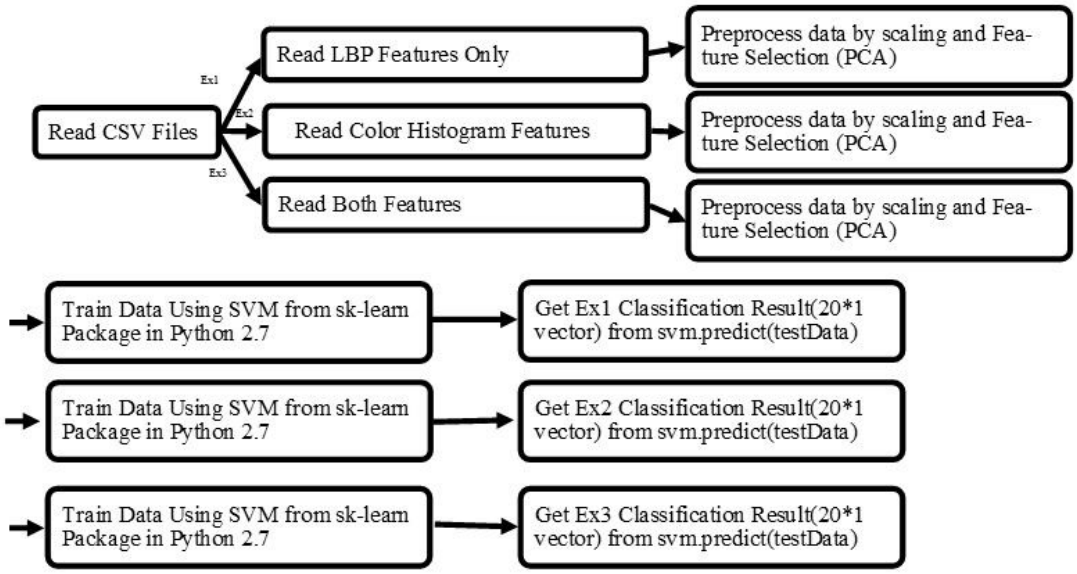

Figure 2. The data preprocessing and Classification in Python

Several different methods are deployed in feature extraction and classification stage. Features such as Color histogram, Gabor features and Local Binary Pattern are extracted in Matlab. Extracted data are collected in .csv files and then support vector machines are done in Python based on those .csv files. Gabor features don't contribute much in the classification stage, therefore the following three features are specifically described.

Color Histogram (RGB):

Color histogram is used because the color intensity of inflammation and normal pictures are different. Normal pictures have a high contrast between cells and background. Cells are dark red to purple and the background is clear. Cells shrink in a thinner shape and have clear boundaries. Inflammation pictures have light pink cells. Those cells usually spread out in the whole picture and the picture doesn't have much clear background.

Color histogram is a representation of the distribution of colors in an image.The color histograms of Red, Green and Blue are extracted from the original images(Example: [countsR,binLocationsR] = imhist(X(:,:,1), 10);). Since 256 * 3 features are too many for machine learning modeling, 256 levels are separated into 10 bins, and each bin contains 3 features. All 10 bins * 3 features are used in classification.

Local Binary Pattern

From observation, the texture of inflammation and normal pictures are different. Gabor filter was first applied to extract those texture information while the classification results are not ideal. Finally Local Binary Pattern (LBP) is chosen since it is widely used in texture feature extraction and machine learning.

LBP is a type of visual descriptor used for classification in computer vision. It is the particular case of the Texture Spectrum model. Pictures need to be convert to grayscale to be

processed, which means the LBP doesn't show any information about color. A 1*10 vector is extracted as LBP features for each image(Example: feature = extractLBPFeatures(A,'Upright',false);).

Combination of color histogram (Red only) and LBP

Since RGB histogram and LBP show slightly different results in the classification and they contain completely different information in pictures, they are combined to see if it will give a better classification result. Only one histogram is used to balance the feature number in LBP. The reason red is chosen is that the most different color in inflammation and normal picture is the intensity of red.

An example of the intermediate results from feature extraction methods above are shown in Figure 3. The data is exported from Matlab and contain features in the combination of red histogram and LBP.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	0.00E+00	5.81E+03	3.03E+04	3.80E+04	5.78E+04	1.36E+05	1.47E+05	9.78E+05	8.00E+00	0.00E+00	0.10999	0.151243	0.047484	0.22294	0.408615	0.366271	0.162282	0.254189	0.623877	0.363016
2	0.00E+00	2.47E+02	7.35E+03	2.51E+04	3.67E+04	7.29E+04	3.65E+05	8.85E+05	1.66E+02	0.00E+00	0.071518	0.109603	0.064474	0.286456	0.66201	0.458792	0.16725	0.181735	0.359077	0.240778
3	0.00E+00	7.20E+01	6.12E+03	2.17E+04	3.44E+04	6.12E+04	2.90E+05	9.78E+05	1.81E+02	0.00E+00	0.075027	0.110919	0.059891	0.278599	0.640795	0.456316	0.166957	0.192673	0.38765	0.258361
4	0.00E+00	3.17E+02	6.32E+03	2.04E+04	2.85E+04	5.37E+04	2.77E+05	1.01E+06	5.10E+01	0.00E+00	0.08641	0.125698	0.057601	0.267589	0.572878	0.44509	0.173169	0.21772	0.464125	0.287603
5	0.00E+00	1.13E+02	4.83E+03	2.45E+04	4.81E+04	7.94E+04	2.85E+05	9.51E+05	4.71E+02	0.00E+00	0.072815	0.10761	0.065406	0.293625	0.653198	0.453921	0.161873	0.175362	0.382216	0.238473
6	0.00E+00	2.53E+02	8.01E+03	2.37E+04	3.44E+04	6.20E+04	2.81E+05	9.83E+05	5.47E+02	0.00E+00	0.070105	0.107002	0.062869	0.286851	0.660012	0.465756	0.165394	0.179846	0.360238	0.235338
7	0.00E+00	1.11E+02	4.87E+03	1.90E+04	3.45E+04	5.95E+04	2.84E+05	9.91E+05	7.90E+01	0.00E+00	0.064815	0.098024	0.055279	0.277614	0.687503	0.460575	0.158097	0.175047	0.341482	0.221425
8	0.00E+00	2.33E+02	7.24E+03	2.49E+04	4.07E+04	6.50E+04	2.93E+05	9.61E+05	5.10E+02	0.00E+00	0.055139	0.08866	0.065958	0.293122	0.72326	0.476174	0.160354	0.149264	0.255667	0.188741
9	0.00E+00	3.36E+02	6.02E+03	2.10E+04	3.57E+04	6.44E+04	3.32E+05	9.33E+05	8.00E+01	0.00E+00	0.070295	0.106416	0.063835	0.286041	0.66041	0.469163	0.168016	0.181187	0.353294	0.236029
10	0.00E+00	2.43E+02	6.51E+03	2.09E+04	3.42E+04	6.09E+04	2.68E+05	1.00E+06	3.10E+02	0.00E+00	0.072185	0.108325	0.058461	0.282501	0.648067	0.46207	0.162675	0.187412	0.380338	0.245126
11	0.00E+00	4.72E+02	1.00E+04	2.54E+04	3.31E+04	7.33E+04	3.39E+05	9.11E+05	1.90E+01	0.00E+00	0.084354	0.123267	0.059388	0.271592	0.583491	0.446272	0.17181	0.211879	0.453306	0.284398
12	0.00E+00	8.11E+02	1.09E+04	2.79E+04	3.57E+04	6.70E+04	2.88E+05	9.62E+05	1.20E+02	0.00E+00	0.080583	0.116292	0.058635	0.271129	0.61493	0.448949	0.163093	0.200019	0.433094	0.262988
13	0.00E+00	3.76E+02	7.73E+03	1.92E+04	2.79E+04	6.11E+04	2.71E+05	1.01E+06	8.00E+00	0.00E+00	0.092041	0.130044	0.045633	0.244134	0.52415	0.421956	0.160656	0.230318	0.537402	0.304131
14	0.00E+00	1.16E+03	1.39E+04	3.10E+04	3.81E+04	6.97E+04	3.28E+05	9.10E+05	2.04E+02	0.00E+00	0.061809	0.094196	0.065092	0.288399	0.708584	0.469921	0.159647	0.159981	0.289833	0.204683
15	0.00E+00	2.30E+01	1.25E+03	9.49E+03	2.59E+04	7.15E+04	5.56E+05	7.28E+05	0.00E+00	0.00E+00	0.114055	0.157992	0.040946	0.213291	0.340784	0.347465	0.166348	0.267537	0.659736	0.378809
16	0.00E+00	0.00E+00	1.10E+01	2.70E+03	2.70E+04	7.06E+04	1.93E+05	8.69E+05	2.31E+05	0.00E+00	0.04074	0.058559	0.054922	0.28115	0.783453	0.467106	0.129473	0.101395	0.199607	0.118217
17	0.00E+00	0.00E+00	9.00E+00	2.07E+03	2.47E+04	7.13E+04	1.84E+05	9.31E+05	1.79E+05	5.00E+00	0.027192	0.044555	0.051587	0.275151	0.811575	0.467731	0.128272	0.074019	0.109158	0.087361
18	0.00E+00	0.00E+00	0.00E+00	2.01E+03	2.14E+04	6.98E+04	1.93E+05	9.93E+05	1.13E+05	1.80E+01	0.02692	0.043932	0.056998	0.275328	0.824846	0.455241	0.124181	0.06061	0.069534	0.08214
19	0.00E+00	0.00E+00	3.00E+01	2.88E+03	1.51E+04	3.43E+04	1.70E+05	1.16E+06	7.27E+03	0.00E+00	0.04754	0.072219	0.057273	0.294081	0.720562	0.495235	0.155691	0.13634	0.268554	0.152787
20	0.00E+00	0.00E+00	0.00E+00	1.21E+03	1.10E+04	3.08E+04	1.82E+05	1.14E+06	2.55E+04	0.00E+00	0.03902	0.060177	0.05291	0.295612	0.763409	0.475964	0.143851	0.115963	0.210821	0.129418
21	0.00E+00	0.00E+00	1.00E+00	3.47E+03	2.46E+04	5.44E+04	2.83E+05	1.02E+06	5.08E+03	0.00E+00	0.033942	0.052885	0.055941	0.293913	0.799001	0.466679	0.141809	0.087773	0.102805	0.109376
22	0.00E+00	0.00E+00	0.00E+00	1.68E+03	2.87E+04	9.23E+04	2.46E+05	8.47E+05	1.77E+05	0.00E+00	0.03253	0.04937	0.053525	0.279958	0.808948	0.460704	0.130211	0.082293	0.123474	0.098211
23	0.00E+00	0.00E+00	1.80E+01	1.84E+03	2.47E+04	7.34E+04	2.07E+05	8.69E+05	2.17E+05	0.00E+00	0.034005	0.052976	0.050867	0.277591	0.796864	0.468208	0.133597	0.09386	0.154809	0.107506
24	0.00E+00	0.00E+00	3.40E+01	2.24E+03	2.44E+04	7.66E+04	2.00E+05	9.13E+05	1.76E+05	1.00E+00	0.045203	0.068862	0.044899	0.279596	0.707848	0.497029	0.144845	0.143765	0.319871	0.144819
25	0.00E+00	0.00E+00	3.08E+02	1.21E+04	4.06E+04	7.96E+04	3.19E+05	9.40E+05	1.18E+03	0.00E+00	0.0489	0.071844	0.041587	0.268113	0.705809	0.505516	0.143529	0.149035	0.313697	0.154607
26	0.00E+00	0.00E+00	5.30E+01	4.66E+03	2.92E+04	6.27E+04	1.85E+05	9.14E+05	1.97E+05	0.00E+00	0.044199	0.064454	0.048416	0.274839	0.740062	0.49216	0.139472	0.127839	0.268684	0.134579
27	0.00E+00	0.00E+00	1.00E+00	1.68E+03	1.56E+04	3.76E+04	1.98E+05	1.13E+06	6.39E+03	0.00E+00	0.042887	0.068092	0.042401	0.270927	0.720408	0.50017	0.146019	0.1429	0.292775	0.147335
28	0.00E+00	0.00E+00	1.00E+00	2.00E+03	1.94E+04	5.12E+04	2.76E+05	1.04E+06	8.34E+03	0.00E+00	0.029614	0.047215	0.052576	0.285177	0.813346	0.457278	0.13201	0.078677	0.095695	0.096927
29	0.00E+00	0.00E+00	1.00E+01	2.73E+03	1.73E+04	4.21E+04	2.06E+05	1.12E+06	7.41E+03	0.00E+00	0.03973	0.062649	0.060637	0.30743	0.764092	0.48343	0.155894	0.108562	0.160608	0.129928
30	0.00E+00	0.00E+00	0.00E+00	1.05E+03	1.09E+04	3.49E+04	1.20E+05	1.05E+06	1.77E+05	1.00E+00	0.040371	0.060064	0.062862	0.309807	0.76939	0.473794	0.154699	0.103632	0.169567	0.122325

Figure 3. Training Data Example in CSV File

Classification

After extracting these features, a supervised machine learning classifier was applied. The Supported Vector Machine (SVM) is chosen as the classifier in this application, as it constructs a set of hyperplanes by features from the training data, and calculated the Euclidean distance between each point on testing data and the hyperplane. The further the point is from the hyperplane, the lower the generation error would have during the classification.

All the classification processes are done in Python with sk-learn package.

Before doing classification, a simple data preprocessing is performed. In data preprocessing part, feature scaling and feature selection are applied. Feature scaling is a method used to standardize the range of independent variables or features of data. It is quite

necessary since the range of values of raw data varies widely. It prevent the classifier from being confused by large data number. Feature selection is the process of selecting a subset of relevant features for use in model construction. Principal component analysis is used as the feature selection method. It helps reduce training time and enhance generalization by reducing overfitting.

Support Vector Machine is chosen as our classifier because it is a well-known and effective tool to do classification in Machine Learning area. Also, the SVM kernel trick allows us to use different parameters to choose the best model. After experiments, we choose 'rbf' kernel and C equals 10.

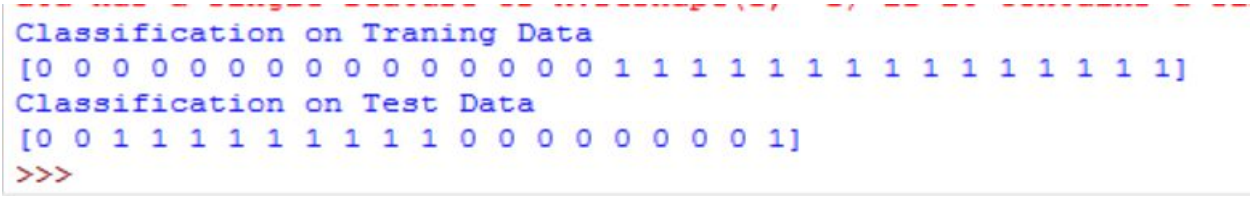

Figure 4. Result Example in Experiment 3

Results:

Three sets of results are generated from three methods. 1 to 20 in the picture name are 0001.jpg to 0020.jpg in the given testing set respectively. In the classification results table, 0 means lung cells in the picture are normal and 1 means lung cells in the picture are inflamed. The classification result of RGB histogram and the combination of LBP and red histogram are consist, whereas LBP is slightly different. Testing pictures with disagreement are picture 0001.jpg, 0002.jpg, 0010.jpg, 0011.jpg and 0020.jpg.

	1	2	3	4	5	6	7	8	9	10
LBP	1	1	1	1	1	1	1	1	1	0
RGB	0	0	1	1	1	1	1	1	1	1
LBP &R	0	0	1	1	1	1	1	1	1	1

(cont.)	11	12	13	14	15	16	17	18	19	20
LBP	0	0	0	0	0	0	0	0	0	0

RGB	1	1	0	0	0	0	0	0	0	1
LBP &R	1	1	0	0	0	0	0	0	0	1

Conclusion and Discussion:

All three methods are able to solve the problem, as we have a series of results for all pictures by all three methods. By using those methods, automatically classification of lung cell images is achieved. The solution is able to be generated to all situation since we keep the same standard for all testing images. To evaluate the quality of our solution, the results are compared with our observation since there is no ground truth.

Picture 0003.jpg to 0009.jpg are inflammation and 0013.jpg to 0019.jpg are normal, which are consistent in all three methods and observation. The challenge comes from picture 0001.jpg, 0002.jpg, 0010.jpg, 0011.jpg and 0020.jpg. We tend to believe picture 0001.jpg, 0002.jpg, 0010.jpg, 0011.jpg and 0020.jpg are all inflammation based on our observation because normal pictures don't have much pink plasma.

The possible misclassification(10, 11, 12, 20) in Experiment 1:

The 10th, 11th, 12th and 20th images are very tricky to identify even with human's eyes. Due to the lack of enough training data, it is hard for our model to classify some tricky images.

The possible misclassification(1, 2) in Experiment 2, 3:

The 1st and 2nd images represent the inflamed cells based on our observation, but our classifier misclassified them into images of normal cells. This is possibly because the color distribution of these two images is not in training dataset. Obviously, these two images are covered with inflamed cells and leaves no blank and we can find some images like in training dataset. It makes it hard to identify.

Possible Future Improvement:

The accuracy is acceptable for now but there is still something we can do in the future to build a better classification model. First, more training data is needed for classification. Training images must be randomly chose from the database to make the model more general. Second, some other features like GLCM and Gabor can be used in feature extraction. Last but not the least, the weight of each feature might be adjusted in classification.

Reference:

Extract Local Binary Pattern (LBP) Features - Matlab extractLBPFeatures. https://www.mathworks.com/help/vision/ref/extractlbpfeatures.html

Histogram plot - Matlab Histogram
https://www.mathworks.com/help/matlab/ref/histogram.html

sklearn.svm.SVC ----- scikit-learn 0.18 Documentation
http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html