Pattern Recognition Lab2

NAME	WANG Yi	XIAO Tianqi	WANG Zhenqiao
Student ID	3371561	3371477	3371590

1. Task Description

1.1 Import the image tiles into Matlab without rescaling.

Like in lab 1 but with scale now equals 1.

1.2 Use your implementation of assignment 1 to segment your training and testing image with chessboard and SLIC.

Using the implementation function in lab 1, chessboard and SLIC segmentation methods are applied to images separately, resulting in plenty of segments. The ground truth label of each segment is obtained by majority vote. In this step we get 4 segmentation data (chessboard_train, chessboard_test, slic_train and slic_test).

1.3 Train a random forest on the segments of the training image.

The classification is done by the in-build function: “TreeBagger”, with predetermined number of trees, training data and training label.

1.4 Predict segment labels for both images.

The predicted labels of the test image are determined by function “predict”, based on these trees trained before. Images with predicted classes are presented after reshaping the predicted label vector according to segmentation boundaries.

1.5 Experiment with the following parameters

Experiment different number of trees with random forest classification.

1.6 Evaluation

Compare the resulting prediction image with the ground truth image. Calculate the overall accuracy and display the confusion matrix. Evaluate the prediction visually and in terms of accuracy, runtime and confusion matrix. Discuss the impact of the parameters and the preceding unsupervised segmentation.

2. Results and analysis

2.1 original image

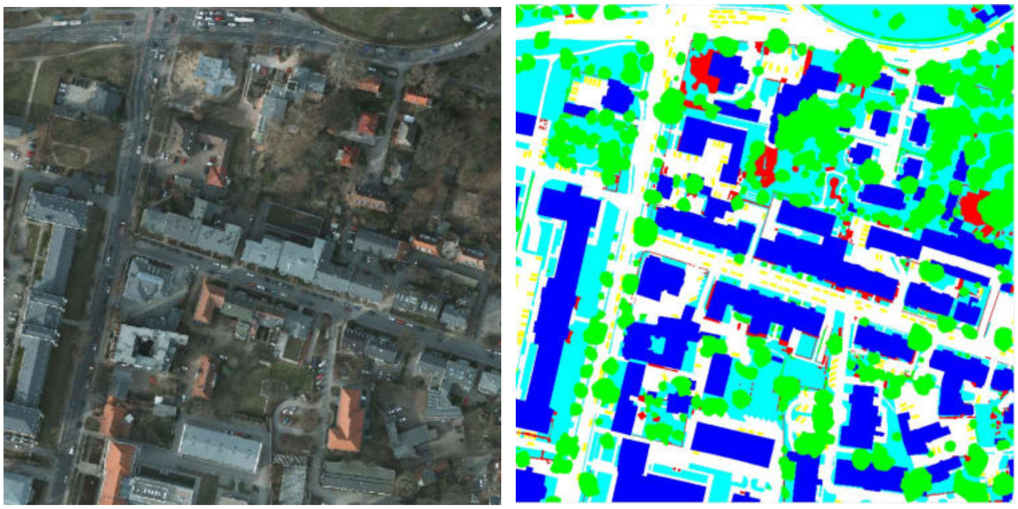


Fig1. Train image and label.

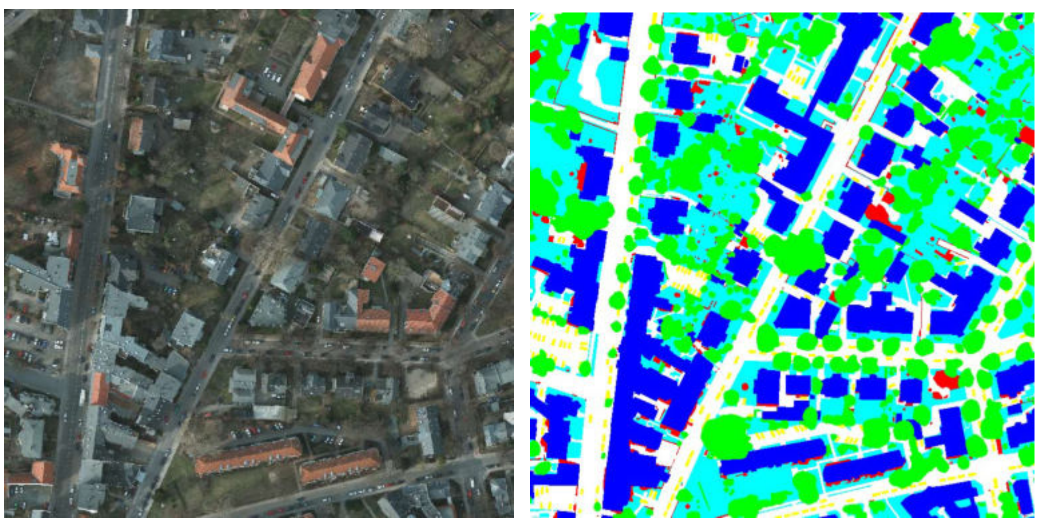


Fig2. Test image and label.

2.2 Segmentation

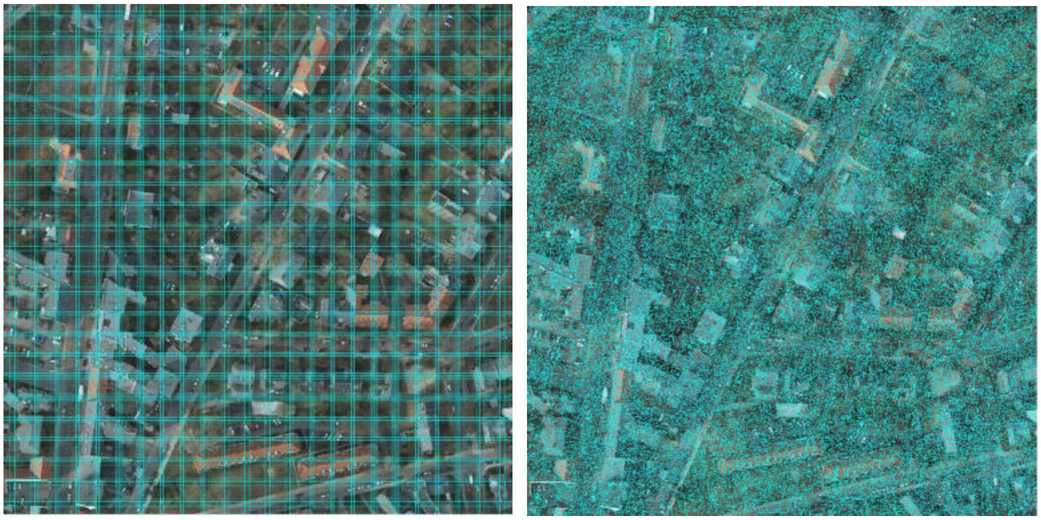


Fig3. Chessboard (30*30) and SLIC (K=37190) segmentation for train image

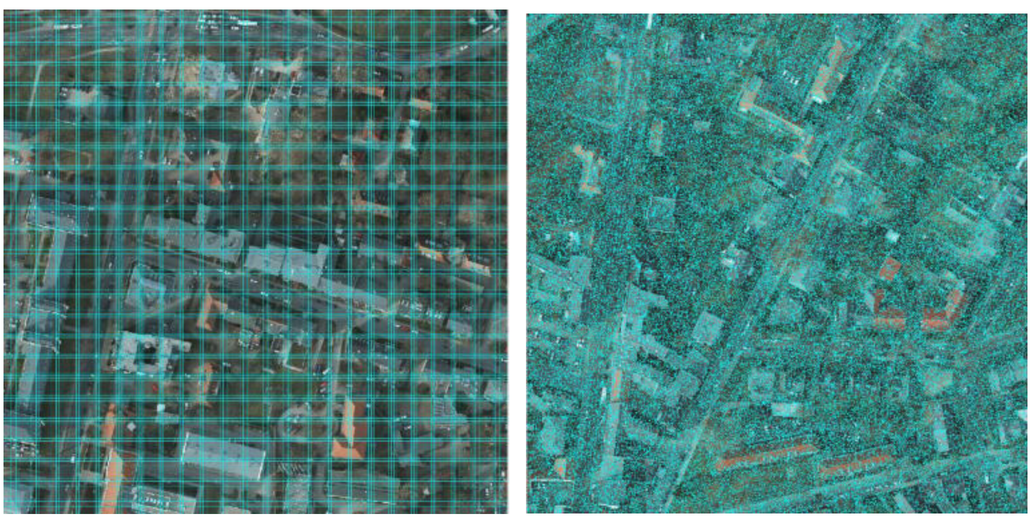


Fig4. Chessboard (30*30) and SLIC (K=37019) segmentation for test image

2.3 Random Forest Classification

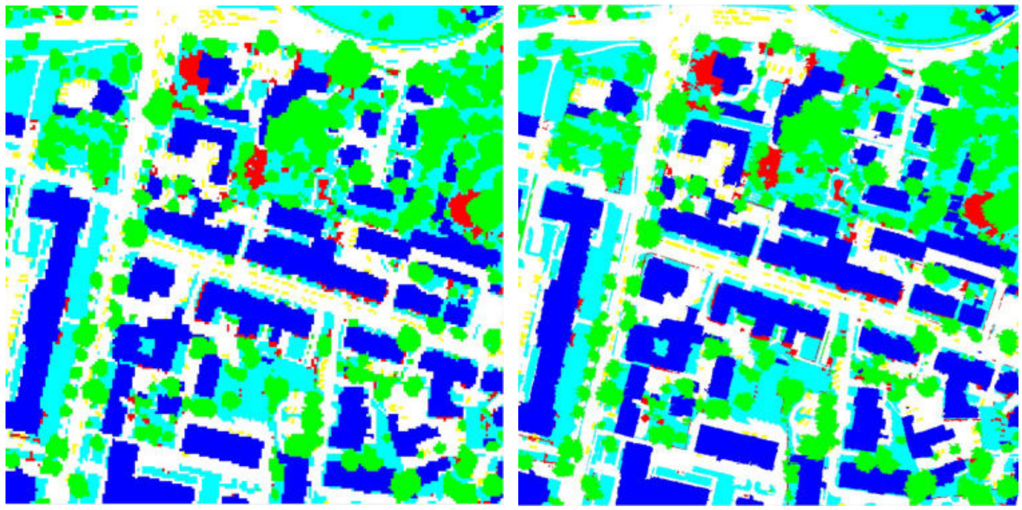


Fig5. Prediction of train image for chessboard (left) and SLIC (right) segmentation (Num_of_trees = 30)

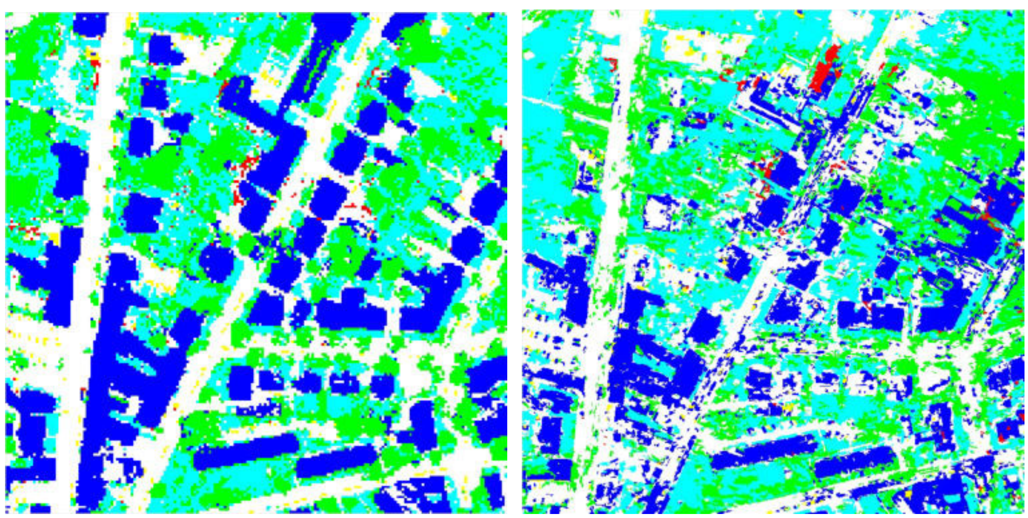


Fig6. Prediction of test image for chessboard (left) and SLIC (right) segmentation (Num_of_trees = 30)

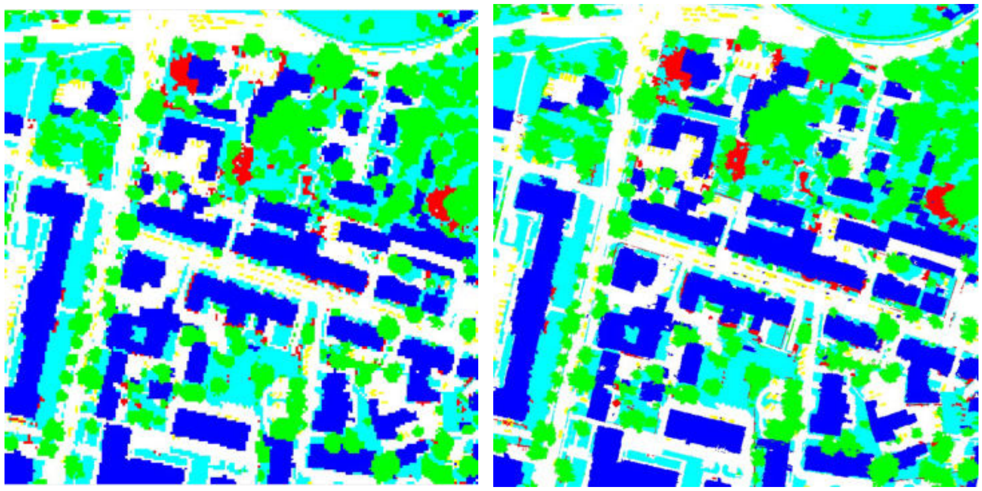


Fig7. Prediction of train image for chessboard (left) and SLIC (right) segmentation (Num_of_trees = 10)

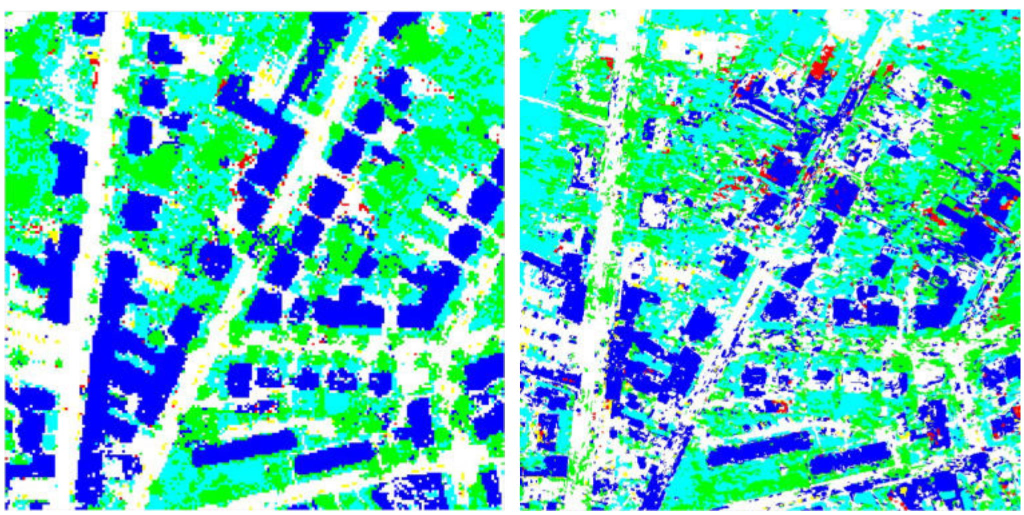


Fig8. Prediction of test image for chessboard (left) and SLIC (right) segmentation (Num_of_trees = 10)

2.4 Accuracy Analysis

2.4.1 Confusion matrix (Nt=30)

Table1. The Confusion Matrix of the training image using Chessboard method.

Nt=30	**Impervious surfaces**	**Building**	**Low Vegetation**	**Tree**	**Car**	**Clutter/ Background**	**User's accuracy**
Impervious surfaces	9715449	147451	362690	141157	148285	39603	0.9205
Building	158566	8424900	122713	40686	1023	19036	0.9610
Low Vegetation	491903	157906	6617397	282060	10440	69645	0.8674
Tree	168650	51473	297215	6990095	7917	18866	0.9278
Car	173778	2201	7064	12680	432315	888	0.6874
Clutter/ Background	92554	32469	116921	32122	1220	610662	0.6893
Producer's accuracy	0.8995	0.9556	0.8795	0.9322	0.7191	0.8049	0.9109

Table2. The Confusion Matrix of the test image using Chessboard method.

Nt=30	**Impervious surfaces**	**Building**	**Low Vegetation**	**Tree**	**Car**	**Clutter/ Background**	**User's accuracy**
Impervious surfaces	7516787	448051	592596	302936	86697	89202	0.8318
Building	519674	7976230	80021	249705	36036	14054	0.8987
Low Vegetation	856682	405110	5286027	1710951	3648	43966	0.6364
Tree	857295	350943	1691532	5380549	16649	13019	0.6475
Car	341583	15231	15682	16529	221156	7619	0.3580
Clutter/ Background	413679	151835	140742	109930	17414	20240	0.0237
Producer's accuracy	0.7115	0.8533	0.6771	0.6924	0.5795	0.1076	0.7334

Table3. The Confusion Matrix of the training image using SLIC method.

Nt=30	**Impervious surfaces**	**Building**	**Low Vegetation**	**Tree**	**Car**	**Clutter/ Background**	**User's accuracy**
Impervious surfaces	9920346	102037	235179	179652	84407	33014	0.9400
Building	90101	8527437	75706	55132	426	18122	0.9727
Low Vegetation	240994	100041	6837738	388738	5850	55990	0.8962
Tree	183377	59539	351194	6905462	10990	23654	0.9165
Car	135165	1276	6638	13474	470860	1513	0.7487
Clutter/ Background	69846	23858	88700	41734	1718	660092	0.7451
Producer's accuracy	0.9324	0.9675	0.9003	0.9105	0.8200	0.8300	0.9256

Table4. The Confusion Matrix of the test image using SLIC method.

Nt=30	**Impervious surfaces**	**Building**	**Low Vegetation**	**Tree**	**Car**	**Clutter/ Background**	**User's accuracy**
Impervious surfaces	5735477	1271086	811595	1109147	29151	79813	0.6347
Building	2927024	5057381	271083	376710	102654	140868	0.5698
Low Vegetation	706368	301829	4803780	2458054	8798	27555	0.5783
Tree	1526534	289289	3386585	3081504	8494	17581	0.3708
Car	195249	247299	43989	19838	105759	5666	0.1712
Clutter/ Background	339476	161305	136146	200945	9465	6503	0.0076
Producer's accuracy	0.5018	0.6901	0.5082	0.4253	0.4001	0.0234	0.5220

We can get a better result of class Impervious surface and building because they have typical characteristics. But for low vegetation and tress, they are similar to each other in the images, so they perform worse. What's more, for cars we have another situation that there are not enough training data and test data to get a reliable data classification. And for Clutter/background, they include so many things, so it is also hard to be classified.

2.4.2 Accuracy

Table5. total accuracy w.r.t. number of N_trees

Algorithms	Chessboard				SLIC			
N_trees	50	30	20	10	50	30	20	10
Overall Accuracy(Training Image)	0.9233	0.9109	0.9100	0.9060	0.9312	0.9256	0.9245	0.9197
Overall Accuracy(Test Image)	0.7220	0.7334	0.7182	0.7172	0.5189	0.5220	0.5174	0.5020
Time (s)	10.11	8.56	5.91	3.18	10.85	9.84	6.72	3.72

As we can see from table 5, the amount of the trees will affect accuracy. However, more trees also mean more computational cost and after a certain number of trees, the improvement is less and less, and even decreases. When number of trees increases to 50, the accuracy of test image does not increase any more, but rather begins to decrease. Generally, the classification has an apparent good result on training image, but not that good with test image. In our case, chessboard segmentation method has a better result, which may come from the data itself, and also the segmentation parameter (here 30*30 for chessboard is already quite small cluster).

3. Answer of Questions

3.1 Explain and discuss the differences between Bagging and Boosting.

Bagging and Boosting are both executed by sampling with replacement, which means that training data sets are generated from the original data. Each data point has the same probability to be chosen in Bagging, while data points are given different weight in Boosting.

Weak classifiers are built up independently in Bagging. In Boosting, previous classifiers contribute to current ones and their weights are redistributed.

In Bagging the final strong classifier is obtained by averaging weak ones. However, Boosting assigns higher weights to the classifier with better classification capability.

3.2 What is the main idea behind Random Forest?

learnable

Random forest consists of CART and Bagging. It is a collection of decision trees and combines them to get stronger classifiers. Data sets of decision trees are generated by

Bagging.

3.3 What is controlled by the depth of the tree? Can you make it arbitrarily deep?

The deeper the tree, the more branches it has. The depth cannot be arbitrarily large, because it may cause over-fitting.

3.4 What is controlled by the number of trees within a Random Forest?

In general, more trees improve the accuracy. However, it is not helpful beyond a certain range and causes more computational cost.

3.5 How could you improve the classification accuracy?

Add more data

Choose a more suitable classifier

Increase the dimension of feature space

Preprocess like segmentation

3.6 What represents the entropy?

Entropy measures the disorder of images, i.e. the information included in the image

3.7 Imagine you are given a two-class classification problem which splits results in an entropy of zero? Make a sketch and draw the histograms for the sake of clarity and an example calculation resulting in an entropy of zero.

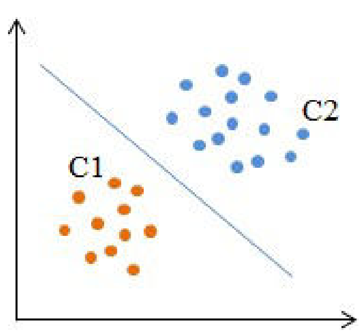
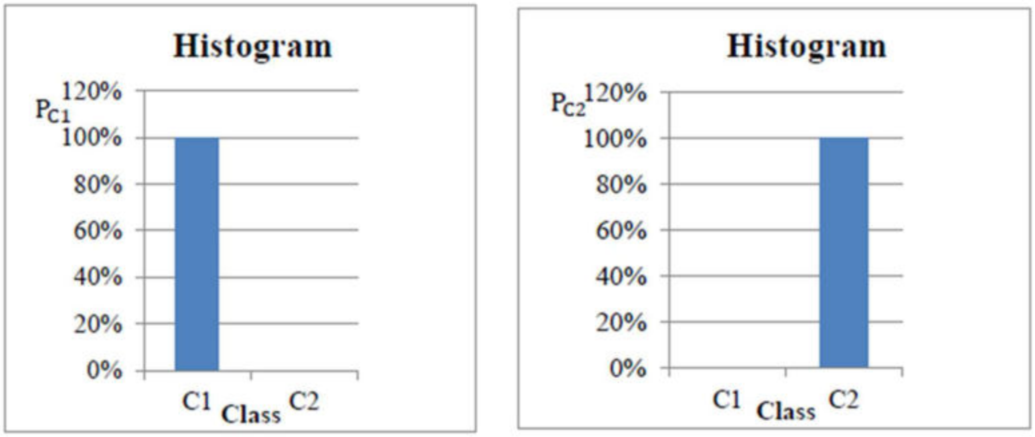


Fig9. Two-class Classification



√

Fig10. Histograms of two classes

3.8 Imagine you are given a two-class classification problem. What happens to the entropy when the probability of one class becomes

a. Zero?

b. Equal to the other class probability?

When the probability of one class becomes zero, the entropy is also zero. √

When the probability of one class becomes equal to the other class probability, the entropy is 1. √