Homework #8

ECE661

Name:Chengzhang Zhong

Email:zhongc@purdue.edu

Part 1. Algorithm explanation

Step 1. Local Binary Pattern(LBP) feature extraction

- For an RGB image, convert it into gray scale image with the same method used in previous homework. Now, consider a pixel at the location marked by upper case "**X**" in figure A.

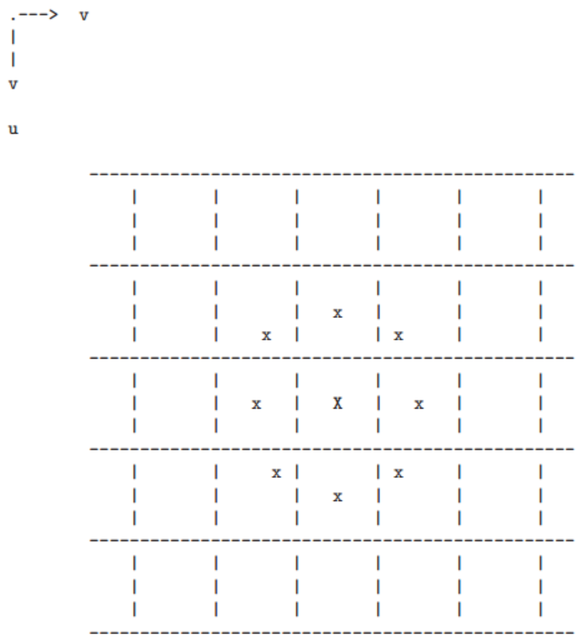


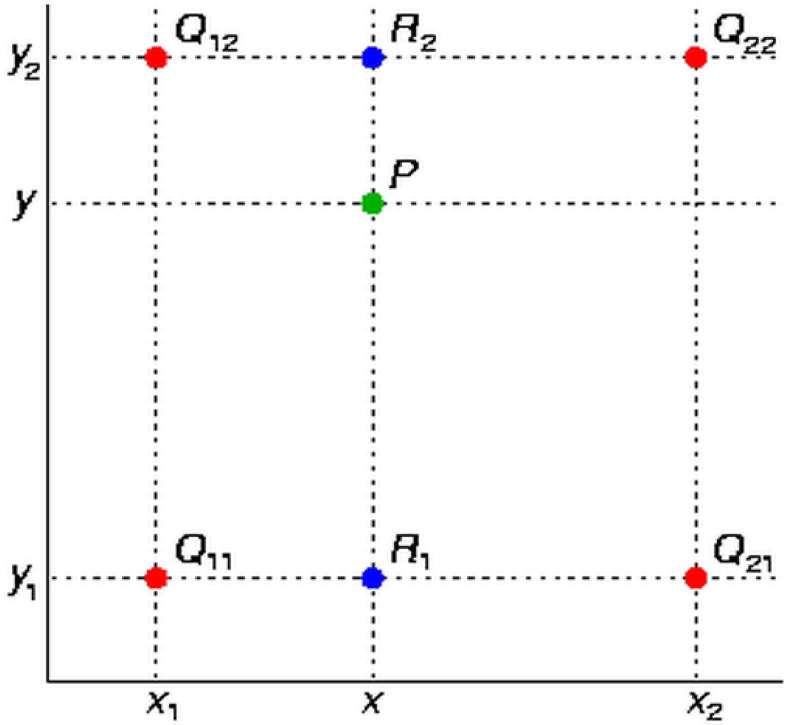
Fig. A

Its neighbors is given by the displacements in directions of **u** and **v.**

$$(\Delta u, \Delta v) = \left(Rcos\left(\frac{2\pi p}{P}\right), Rsin\left(\frac{2\pi p}{P}\right)\right), p = 0,1,2,3,4,5,6,7$$

Where R represents radius of the neighbor circle and P represents total number of neighbors. Each of these neighbor's gray level need to be computed with an appropriate interpolation formula.

- In this case, we can use bilinear interpolation:



where P is the location need to be estimate,

$$f(R_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(R_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

$$f(P) = \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)$$

where $f(P)$ is the final pixel value filled into image plane.

- After estimating the gray levels at each neighbor points on the circle, we threshold these gray levels with respect to the gray level value at the pixel at the center of the pixel. If the neighbors gray level value is greater or equal to the value at the center, we set the point equals to 1. Otherwise, the point equals to 0.

- Now, we obtained a binary sequence of length 8 of the neighbors. The next task is to find a way to represent this sequence in a invariant to in-plane rotations of the image. The method is to shift the binary sequence to left by one bit at a time, and convert it into decimal representation. Keep repeating this step until we found the smallest integer value.

  For example:

  if the binary sequence is [1 1 0 1 1 1 1 1], its final representation would be [0 1 1 1 1 1 1 1].

- After find the sequence, we need to design a method to encode it efficiently, the method can be described as follow:

  (a). Each changing from 0 to 1 or from 1 to 0 is counted as a "run".

  (b). The initial "run" number of a sequence equals to 1.

  (c). If "run" larger than two, encode the sequence as P + 1, where P equals to the number of neighbors.

  (d). If "run" equals to two, encode the sequence as the number of "1"s inside the sequence itself.

  (e). If "run" equals to 1 and the first number in sequence equals to 1, encode the sequence as P.

  (f). If "run" equals to 1 and the first number in sequence equals to 0, encode the sequence as 0.

- In the final step, use raster order to search through the whole image with the above procedures and put all the encoded number into a histogram of P+2 bins from 0 to P+1, each bin represents a corresponding encode value. This histogram is the extracted feature from LBP algorithm.

Step 2. NN-Classifier

Now, to classify a test image, we need to build a feature matrix by training and use it for testing.

- The training stage will take images from 4 different categories – building, car, mountain and tree. Extract all the LBP features and store them as a feature matrix.
- The testing stage will extract the LBP feature from the testing image, then compute and compare the Euclidean distance of the test image's feature with every other feature in the training feature matrix. Select the top K smallest Euclidean distance from the training feature matrix as the test result(K = 5 in this case). Each selected feature belongs to an image category, and the most frequently appeared category among these K selected candidates is the result category of the test image.

Step 3. Confusion Matrix

The result confusion matrix is calculated as following. Rows correspond to the actual class labels while columns corresponds to the predicted class labels:

Actual | Predicted	Building	Car	Mountain	Tree
Building	3	0	2	0
Car	2	3	0	0
Mountain	1	0	4	0
Tree	1	0	0	4

Thus, the overall accuracy of the algorithm equals to: $\frac{3+3+4+4}{5*4} = \frac{14}{20} = 70\%$

Part 2. Demonstration for LBP feature vector for one image for one of each class

Figure 1.1 Building image

1	2	3	4	5	6	7	8	9	10
2656	5207	1752	4007	6490	6937	2207	5284	7130	7848

Figure 1.2 LBP feature of building image(not normalized)

Figure 2.1 Car image

1	2	3	4	5	6	7	8	9	10
1769	3340	1406	3101	6209	6034	2321	3597	17359	4277

Figure 2.2 LBP feature of car image(not normalized)

Figure 3.1 Mountain image

1	2	3	4	5	6	7	8	9	10
2926	3319	2173	3126	3674	5707	2812	3624	16292	5691

Figure 3.2 LBP feature of mountain image(not normalized)

Figure 4.1 Tree image

1	2	3	4	5	6	7	8	9	10
4859	4910	3410	4333	5049	3843	3142	4876	5853	9138

Figure 4.2 LBP feature of tree image(not normalized)

Part 3. Performance and observations

As the overall accuracy of 70% indicated in the end of part one, LBP's performance is not as good as the latest detection algorithms. For the performance in each category, tree is one of the class with highest detection accuracy because its vertical texture type.

Building is one of the category with worst detection result, the reason is because each chess board texture type is different from another. Therefore, the accuracy of this class is relatively low.

Moreover, some of the false positives, images from car, tree and mountain which might also contains chess board texture were mistakenly detected as buildings.