1.Algorithm

先 downsampled 再用 Yokoi algo 計算其周圍的4個q , r , s, 在依其周圍的4個q, r, s, 計算自己的connectivity number

2.your principal code fragment & parameters

1 #include <iostream>

2 #include <opencv2/core/core.hpp>

3 #include <opencv2/highgui/highgui.hpp>

>> 4 #include <cv.h>

5 using namespace std;

6 using namespace cv;

7 Mat picInitialSrc(Mat pic, Mat src, int row, int col)

8 {

9 int i,j;

10 for(i=0; i<col; i++)

11 {

12 for(j=0; j<row; j++)

13 {

14 pic.at<unsigned char>(i,j) = src.at<unsigned char>(i,j);

15 }

16 }

17 return pic;

18 }

19

20 int cal(int a[4]) // q=1, r=2, s=3

21 {

22 if(a[0] != a[1])

23 return 3;

24 else if(a[2] == a[0] && a[3] == a[0])

25 return 2;

26 else

27 return 1;

28 }

29

30 int\* hFunction(Mat yokoi64, int i, int j)

31 {

32 int \*f4 = new int[4];//return 4 neborhood

33 int a[4] = {0};//temp

34 for(int k=0; k<4; k++)

35 {

36 a[k] = 0;

37 }

38 a[0] = yokoi64.at<unsigned char>(i,j);

39 if(j<=62)

40 a[1] = yokoi64.at<unsigned char>(i,j+1);

41 if(i>=1 && j<=62)

42 a[2] = yokoi64.at<unsigned char>(i-1,j+1);

43 if(i>=1)

44 a[3] = yokoi64.at<unsigned char>(i-1,j);

45

46 //cout << "0:" << a[0] << "1:" << a[1] << "2:" << a[2] << "3:" << a[3] << endl;

47 f4[0] = cal(a);

48

49 for(int k=1; k<4; k++)

50 {51 a[k] = 0;

52 }

53

54 if(i>=1)

55 a[1] = yokoi64.at<unsigned char>(i-1,j);

56 if(i>=1 && j>=1)

57 a[2] = yokoi64.at<unsigned char>(i-1,j-1);

58 if(j>=1)

59 a[3] = yokoi64.at<unsigned char>(i,j-1);

60

61 //cout << "0:" << a[0] << "1:" << a[1] << "2:" << a[2] << "3:" << a[3] << endl;

62

63 f4[1] = cal(a);

64

65 for(int k=1; k<4; k++)

66 {

67 a[k] = 0;

68 }

69

70 if(j>=1)

71 a[1] = yokoi64.at<unsigned char>(i,j-1);

72 if(i<=62 && j>=1)

73 a[2] = yokoi64.at<unsigned char>(i+1,j-1);

74 if(i<=62)

75 a[3] = yokoi64.at<unsigned char>(i+1,j);

76

77 //cout << "0:" << a[0] << "1:" << a[1] << "2:" << a[2] << "3:" << a[3] << endl;

78

79 f4[2] = cal(a);

80

81

82 for(int k=1; k<4; k++)

83 {

84 a[k] = 0;

85 }

86

87 if(i<=62)

88 a[1] = yokoi64.at<unsigned char>(i+1,j);

89 if(i<=62 && j<=62)

90 a[2] = yokoi64.at<unsigned char>(i+1,j+1);

91 if(j<=62)

92 a[3] = yokoi64.at<unsigned char>(i,j+1);

93

94 //cout << "0:" << a[0] << "1:" << a[1] << "2:" << a[2] << "3:" << a[3] << endl;

95

96 f4[3] = cal(a);

97

98 return f4;

99 }

100

101 void showSavePicture(Mat picture,string windowName,string saveName)

102 {

103 namedWindow( windowName, WINDOW\_AUTOSIZE );

104 imshow( windowName, picture );

105 imwrite( saveName , picture );

106 }

107

108

109 int main(int argc,char\*\* argv )

110 {

111 Mat src;

112 src = imread("lena.bmp",CV\_LOAD\_IMAGE\_GRAYSCALE);

113

114 int i,j,k;

115 int row = src.rows;

116 int col = src.cols;

117

118 int\* h4num;

119 int ans[64][64] = {0};

120

121 Mat yokoi64(64, 64, CV\_8U);

122 for(i=0; i<512; i++)

123 {

124 for(j=0; j<512; j++)

125 {

126 if(i%8==0 && j%8==0)

127 {

128 if(src.at<unsigned char>(i,j) >= 128)

129 yokoi64.at<unsigned char>(i/8,j/8) = 255;

130 else

131 yokoi64.at<unsigned char>(i/8,j/8) = 0;

132 cout << "(" << i << "," << j << ")" << endl;

133 }

134 }

135 }

136

137 for(i=0; i<64; i++)// q=1, r=2, s=3

138 {

139 for(j=0; j<64; j++)

140 {

141

142 int timeForQ = 0;

143 int timeForR = 0;

144 if(yokoi64.at<unsigned char>(i,j) == 255)

145 {

146 h4num = hFunction(yokoi64,i,j);

147 //time ++;

148 //cout << "time:" << time << endl;

149 //for(int i = 0 ; i< 4; i++)

150 //cout << "h4num:" << h4num[i] << endl;

151 for(k=0; k<4; k++)

152 {

153 if(i==0 && j==63)

154 cout << "ans:" << h4num[k] << endl;

155 if(h4num[k] == 1)

156 {

157 timeForQ++;

158 }

159 else if(h4num[k] == 2)

160 timeForR++;

161 }

162 if(timeForR == 4)

163 ans[i][j] = 5;

164 else

165 ans[i][j] = timeForQ;

166

167 }

168 }}

169

170 //h4num = hFunction(yokoi64,0,0);

171

172 //for(int i = 0 ; i< 4; i++)

173 // cout << "h4num:" << h4num[i] << endl;

174

175 for(i=0; i<64; i++)

176 {

177 for(j=0; j<64; j++)

178 {

179 if(yokoi64.at<unsigned char>(i,j) == 0)

180 cout << " ";

181 else

182 cout<< ans[i][j];

183

184 }

185 cout << endl;

186 }

187

188

189 //dila = picInitialSrc(dila,src,row,col);

190

191 //showSavePicture(yokoi64,"yokoi64","yokoi64.bmp");

192

193 waitKey(0);

194 return 0;

195 }

3.resulting images

11111111 12111111111122322221 111111111111 0 0

15555551 115555555511 2 11 11 1155555555511 0

15555551 1 2115555112 21112221 155555555551 21

15555551 1 2 155112 22221511 1555555555511 1

15555551 22 2112 22 121 0 0 15555555555511 0

15555551 1 2 21 2 1 1 15555555555551 0

15555551 12 1 121111 1321 155555555555511

15111551 1322 1155551111 155555555555551

111 1551 1 121555555511 155555555555511

11 1551 21155555511 15511155555511

21 1551 2 15555555111 1551 11555511

1 1551 2 155555555511 1551 115551 1

1551 1121155555555551 1551 15511 12

1551 15555555555555511 1551 1111 111

1551 1 2221155555555555511 1151 11 1151

1551 2 22 1 1555555555555511 151 11111 1551

1551 2 1 11555555555555551 151 115551 11551

1551 2 11555555555555555111511155511 115551

1551 12 11555555555555555555555555551 155551

1551 11 0 221555555555555555555555555112 1155551

1551 111 22 15555555555555555555555551 1 1555551

1551 1511 1 125112111112111555555555111 11555551

1551 15521 1 121 1 11 1 15555555111 0 15555551

1551 1151 132 2 1155555111 0 115555551

1551 151 0 322 115555111 121 155555551

1551 1221 2 1555551 131 1155555551

1551 2 0 1 115555511 1 1155555551

1551 2 0 0 1155555551 0 1 155555551

1551 2 11555555551 21155555551

1551 1 0 115555555551 15555555551

1551 1 11511115555521 1 115555555551

1551 1 1 11111 1155511 2 155555555551

1551 131 111 15111 2 155555555551

1551 121 0 1121 1 111 1 2 1155555555551

1551 11 111 1 221 11 1 2 1555555555551

1551 12 0 1 21 121 11 1111 2 1555555555551

1551 1 12 22 151111111551 2 11555555555551

1551 1 2 1555551115511 1 15555555555551

1551 2 0 0 22 12555551 15551 1 15555555555551

1551 1 1 1555511 11511 2 115555555555551

1551 0 0 21 155551 1 151 2 155555555555551

1551 2 15555112 151 2 155555555555551

1551 1 1 1 1155555511111 2 155555555555551

1551 2 22 111511111212 21155555555555551

1551 0 1 12 151 2 1 15555555111555551

1551 0 0 0 1111 121 155555551 1555551

1551 0 11111111 155555551 1555551

1551 0 115551 155555551 1555511

1551 15551 211111111 155511

11521 1 12 122155511 2 11 115511

1 151 0 1 1 155555111 2111 15511

22 1511 1 15555555111 155111 1511

22 1511 1 15555555551 155551 1151

2 151 0 1 11155555555511 155511 1511

2 1521 0 1 155555555555511 15551 12151

2 151 121 155555555555551 155511 1551

2 1511 0 155555555555551 115551 1511

21 1511 11 155555555555551 111111151

11 151 0 11555555555555511 111511

11 151 15555555555555551 151

11 151 0 115555555555555551 211

11 151 1155555555555555511 1

11 151 0 155555555555555551

11 111 0 1211111111111111111