Data Mining Image Classification

1 Classifier

I have used K nearest neighbor classifier.

2 Object

I have classified objects as pool or not pool as a binary classification(1 and 0).

3 Data gathering

I have took two images from Google Maps. One image for training and the other image for testing.

Figure 2: Test Image

4 Procedure

I then put a 50 * 50 cell grid on the images and took 30 points of pools and 30 points of not pools,then I gave these 60 point as our training set and corresponding labels as group. Then I drew a 2D plot(figure 4) to see the relationship between green and blue values of points and a 3D plot(figure 5) to see the relationship of RGB values of pools and not pools. For avoiding over counting success or over counting failure if two or more points are in a cell that is considered not pool I consider that as one failure and vice versa. For drawing ROC curve I calculate true positive and false negative rates as in figure 6. So I will have ROC curve in figure 7.

Figure 4: 2D Plot

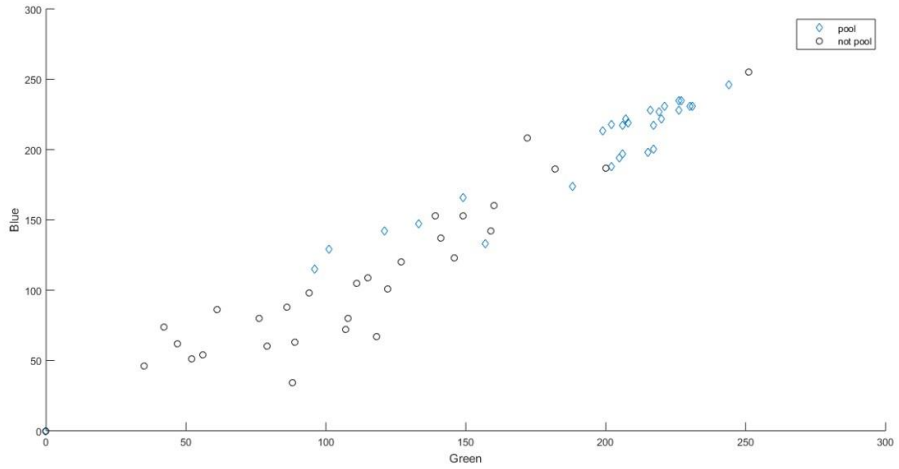


Figure 5: 3D plot

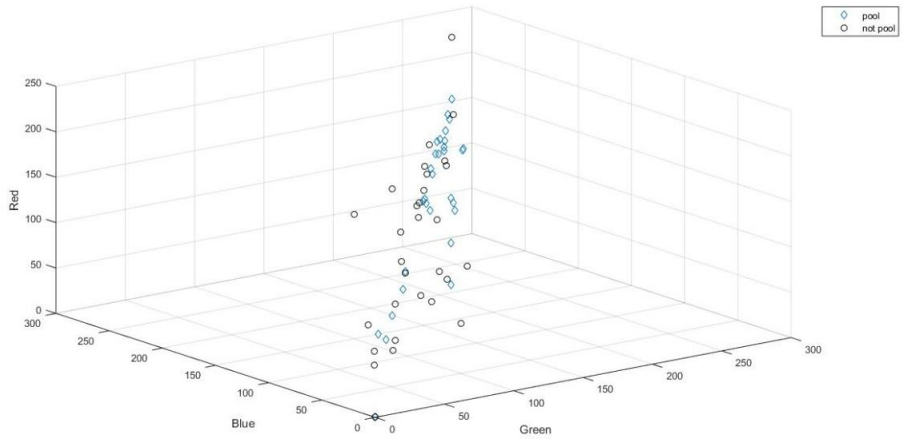


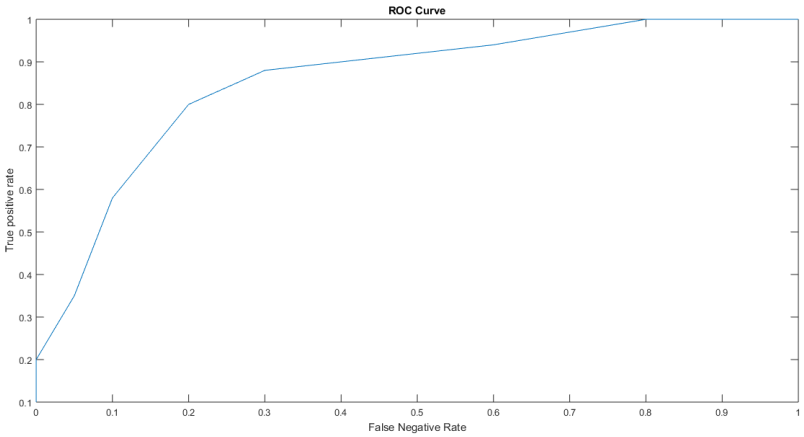
Figure 6: Calculating TPR and FNR

$$\text{sensitivity} = \frac{\text{number of True Positives}}{\text{number of True Positives} + \text{number of False Negatives}}$$

$$\text{false positive rate} = \frac{\text{number of false positives}}{\text{total number of actual negative instances}}$$

		Our Classification Was	
		Pool	*Not Pool*
TRUTH	*Pool*	True Positive	False Negative
	Not Pool	False Positive	True Negative

Figure 7: ROC Curve



1 Matlab Code

```
img1=imread('mypool.png');
img1(50:50:end,:,:)=0;
img1(:,50:50:end,:)=0;
image(img1)

[x_pool,y_pool]=ginput(30);
for i=1:size(x_pool)

red(i)=img1(round(y_pool(i)),round(x_pool(i)),1)
green(i)=img1(round(y_pool(i)),round(x_pool(i)),2)
blue(i)=img1(round(y_pool(i)),round(x_pool(i)),3)
end

red= transpose(red);
green=transpose(green);
blue= transpose(blue);

[x_not_pool,y_not_pool]=ginput(30);
for i=1:size(x_pool)

redn(i)=img1(round(y_not_pool(i)),round(x_not_pool(i)),1)
greenn(i)=img1(round(y_not_pool(i)),round(x_not_pool(i)),2)
bluen(i)=img1(round(y_not_pool(i)),round(x_not_pool(i)),3)
end

redn= transpose(redn);
greenn=transpose(greenn);
bluen= transpose(bluen);
img2=imread('img2.png');
img2(50:50:end,:,:)=0;
img2(:,50:50:end,:)=0;
image(img2)

[x_sample,y_sample]=ginput;
for i=1:size(x_sample)

reds(i)=img2(round(y_sample(i)),round(x_sample(i)),1)
greens(i)=img2(round(y_sample(i)),round(x_sample(i)),2)
blues(i)=img2(round(y_sample(i)),round(x_sample(i)),3)
end
reds= transpose(reds);
greens=transpose(greens);
blues= transpose(blues);
A=[red,green,blue];
B=[redn,greenn,bluen];
training=vertcat(A,B);
sample=[reds,greens,blues];
```

```
for i=1:30
    group1(i)=1;
    group2(i)=0;
end
group1=transpose(group1);
group2=transpose(group2);
group=vertcat(group1,group2);

% 2D Plot
scatter(green, blue,'d'); hold on;
      scatter(greenn,bluen,'k'),xlabel('Green'),
      ylabel('Blue'),legend('pool','not pool')
scatter3(green, blue,red,'d'); hold on;
      scatter3(greenn,bluen,redn,'k'),xlabel('Green'),   ylabel('Blue'),
      zlabel('Red'),legend('pool','not pool')


class = knnclassify(sample, training, group)

% ROC curve
a1=[0,0,0.0500,0.1000,0.2000,0.3000,0.4000,0.5000,0.6000,0.8000,1.0000];
a2=[0.1000,0.2000,0.3500,0.5800,0.8000,0.8800,0.9000,0.9200,0.9400,1.0000,1.0000];
plot(a1,a2)
hold on
xlabel('False Negative Rate'), ylabel('True positive rate'), title('ROC
     Curve')
```