1. Overview of our project

We created a classifier to find swimming pools in images, and evaluated this algorithm with ROC curve.

2. Main ideas of our project

2.1. Classifier

Our pictures are all size of 1500*1100. We divided them with many 20*20 cells.
To get the RGB range, we downloaded several images that contained only single swimming pool. Then we randomly chose pixels and got the RGB values for them. We set the maximum R/G/B value and the minimal value as our boundary (imagine in 3 dimensional RGB space, our boundary is like a box).
After getting the boundary values, we defined the *pooldetect()* function. The codes are showed as below:

```
function sum_spot=pooldetect(pix)
red = (70<pix(:,:,1)).*(pix(:,:,1)<160);
green = (150<pix(:,:,2)).*(pix(:,:,2)<224);
blue = (170<pix(:,:,3)).*(pix(:,:,3)<250);
sum_spot=sum(sum(red.*green.*blue));
end
```

This function is to get the total number of pixels within the RGB boundary in a cell. Then we compared it to the threshold, which means that if the pool pixels' number is less than the threshold, we consider it as negative(no pool) in this cell.
We implemented the classifier on three images.

2.2. Evaluation

We evaluated our classifier by ROC curve. The program was run on three images. Each positive grid was in red and bold line. We regarded the number of red grids as the positive value. The number of red cells that did not cover any part of a pool were regarded as false positive. False negative was the number of cells that failed to mark pools.

We got the total positive and negative values for each image by running the program. After that we counted the false positive and false negative manually, which is time consuming and tedious, and that is why we could only test 3 pictures. The number of threshold values for each image is 23. Finally we added the four positive and negative values of the three images for the same threshold and got the following table and ROC curve(made by excel).

Threshold	Total	TN+FN	FP+TP	FP	FN	TN	TP	FP/(FP+TN)	TP/(TP+FN)
1	12375	12051	324	179	30	12021	145	0.014672131147541	0.828571428571429
2	12375	12116	259	122	38	12078	137	0.01	0.782857142857143
3	12375	12128	247	114	42	12086	133	0.00934426229508197	0.76
4	12375	12149	226	100	49	12100	126	0.00819672131147541	0.72
5	12375	12161	214	91	52	12109	123	0.00745901639344262	0.702857142857143
6	12375	12170	205	84	54	12116	121	0.00688524590163934	0.691428571428571
7	12375	12175	200	82	57	12118	118	0.00672131147540984	0.674285714285714
10	12375	12190	185	70	60	12130	115	0.00573770491803279	0.657142857142857
13	12375	12200	175	65	65	12135	110	0.00532786885245902	0.628571428571429
17	12375	12218	157	49	67	12151	108	0.00401639344262295	0.617142857142857
23	12375	12234	141	39	73	12161	102	0.00319672131147541	0.582857142857143
33	12375	12250	125	30	80	12170	95	0.00245901639344262	0.542857142857143
46	12375	12276	99	14	90	12186	85	0.00114754098360656	0.485714285714286
59	12375	12296	79	12	108	12188	67	0.000983606557377049	0.382857142857143
70	12375	12310	65	10	120	12190	55	0.000819672131147541	0.314285714285714
111	12375	12327	48	5	132	12195	43	0.00040983606557377	0.245714285714286
147	12375	12338	37	4	142	12196	33	0.000327868852459016	0.188571428571429
156	12375	12341	34	4	145	12196	30	0.000327868852459016	0.171428571428571
194	12375	12345	30	4	149	12196	26	0.000327868852459016	0.148571428571429
260	12375	12353	22	4	157	12196	18	0.000327868852459016	0.102857142857143
337	12375	12362	13	2	164	12198	11	0.000163934426229508	0.0628571428571429
372	12375	12367	8	0	167	12200	8	0	0.0457142857142857
386	12375	12370	5	0	170	12200	5	0	0.0285714285714286

Table 2.1. The table of total positive and negative values

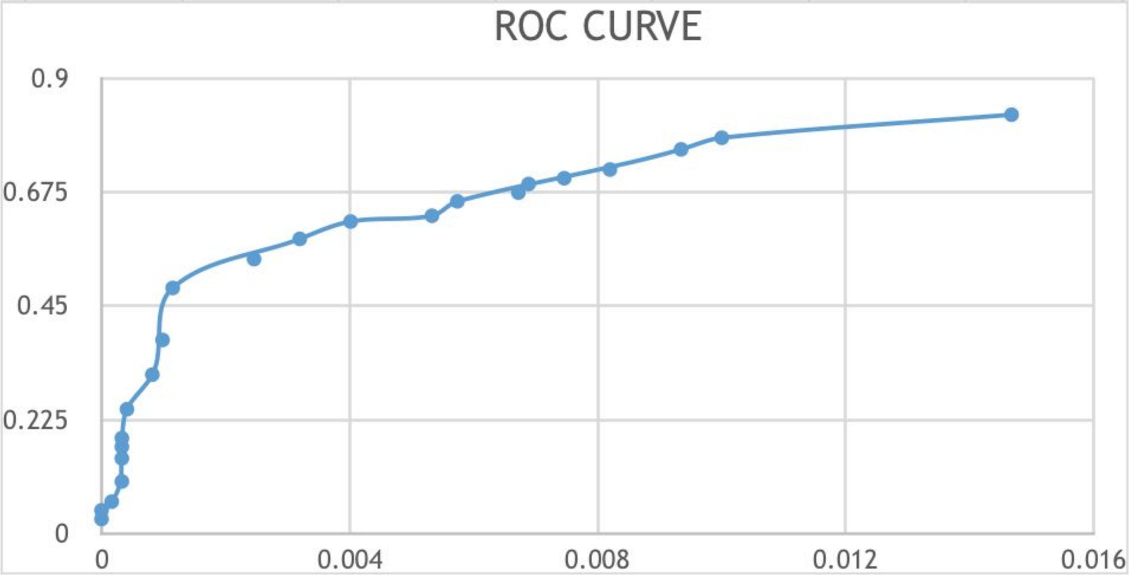

Figure 2.1. ROC curve

3. Observation

What we actually do is create 2 classifiers and then combined them to one that detects pools. This classifier did a good job since it reaches the 83% True Positive Rate with a relatively low, 1.4%, False Positive Rate.
Since we actually created 2 classifiers to do the job and the ROC curve is only based on the threshold for the sum of pool-identified-point in the cell, the True Positive Rate might not able to reach the 100%. But generally speaking, the second classifier did a good job while reaching almost 90% with a relatively low False Positive Rate.

4. Some thoughts

There are several reasons why the True Positive Rate stops at the 83%. One is that there were several parameters that we did not consider and instead we set them as fixed values, such as the size of the cell and the pictures’ scale. And another reason is that due to the coarsely RGB selection in the *pooldetect()*, and due to the slight change of pools' color under different measure scales in google map, the cuboid boundary is actually a bad selection for the classifier.
To further improve the classifier, our idea is to change the RGB selection method. To be more accurate, we can gather different pool pictures under different measure scales (as we said before, the pool color would slightly change in the satellite map with the change of measure scale) and compute the possibilities for the occurrence of RGB points in these images. And then we decide a minimum possibility with which we consider whether a RGB point is in the pool color range. That is, when the RGB point occurs in a bigger possibility than the minimum value, we consider it is in the pool color range.

5. Appendix

pool_proj.m:

```
function Output=proj_pool(picture, thred_hold)
%%picture='testX.png'
%%cell 20*20
Cell_yes=0;
cell_origin=zeros(20,20);
cell_origin(4:17,4:17)=ones(14,14);
pix = imread(picture);
pix_new=im2double(pix);
[x,y,~]=size(pix);
Total_cell=x*y/400;
Cell_no=Total_cell;
for i=0:1:x/20-1
  for j=0:1:y/20-1
    if pooldetect(pix(i*20+1:(i+1)*20,j*20+1:(j+1)*20,:)) >= thred_hold
      Cell_yes = Cell_yes+1;
      pix_new(i*20+1:(i+1)*20,j*20+1:(j+1)*20,2)=pix_new(i*20+1:(i+1)*20,j*20+1:(j+1)*20,2).*cell_origin;
      pix_new(i*20+1:(i+1)*20,j*20+1:(j+1)*20,3)=pix_new(i*20+1:(i+1)*20,j*20+1:(j+1)*20,3).*cell_origin;
      Cell_no=Cell_no-1;
    end
  end
end
imshow(pix_new);
Output=[Total_cell, Cell_no, Cell_yes];
end

function sum_spot=pooldetect(pix)
red = (70<pix(:,:,1)).*(pix(:,:,1)<160);
green = (150<pix(:,:,2)).*(pix(:,:,2)<224);
blue = (170<pix(:,:,3)).*(pix(:,:,3)<250);
sum_spot=sum(sum(red.*green.*blue));
end
```

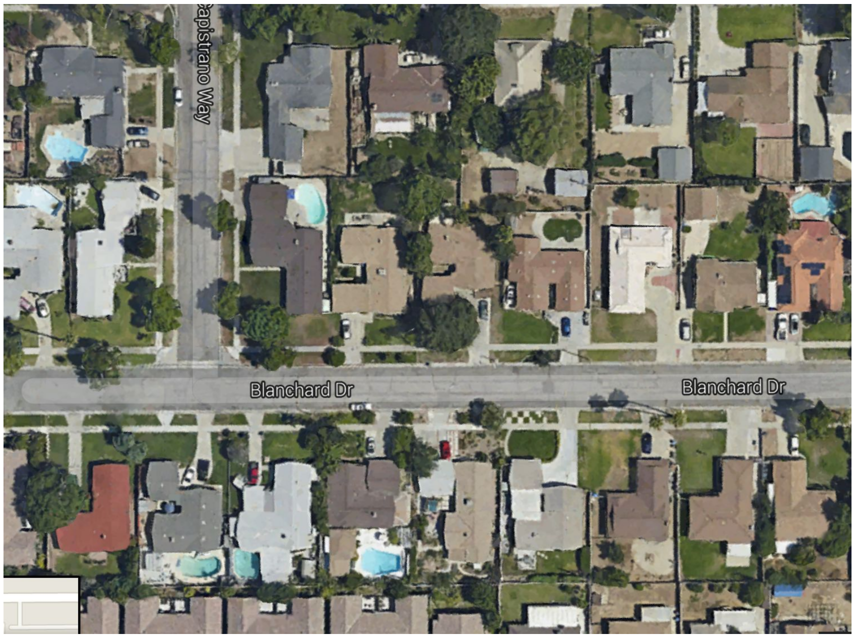

Figure 5.1. “test0.png”

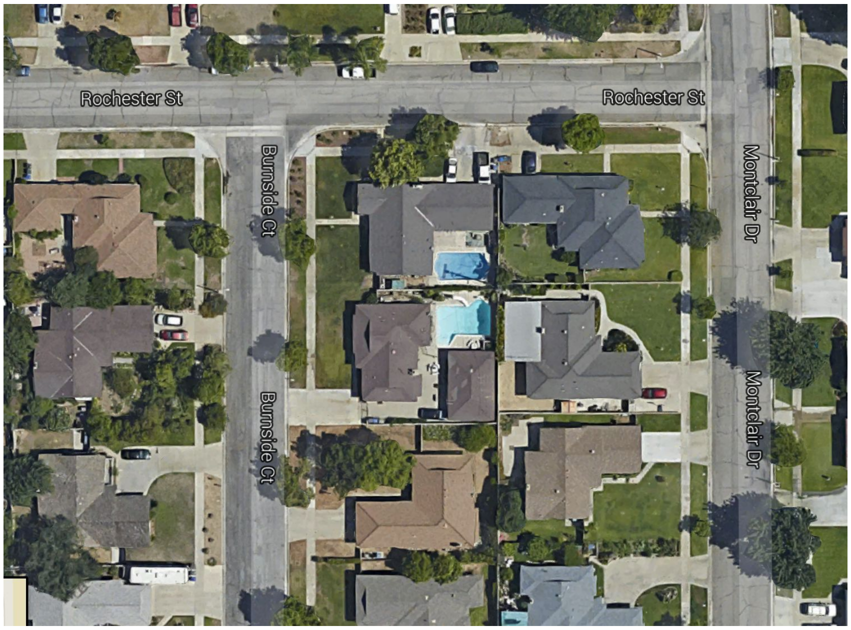

Figure 5.2. “test1.png”

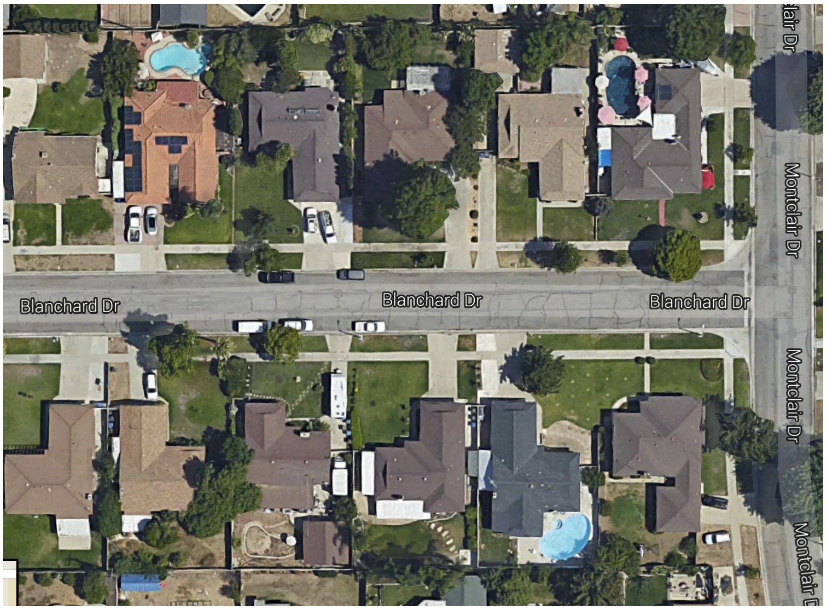

Figure 5.3. “test2.png”

Threshold	Total	TN+FN	FP+TP	FP	FN	TN	TP	FP/(FP+TN)	TP/(TP+FN)
1	4125	4017	108	37	10	4007	71	0.00914935707220574	0.876543209876543
2	4125	4044	81	18	18	4026	63	0.00445103857566766	0.777777777777778
3	4125	4048	77	14	18	4030	63	0.00346191889218595	0.777777777777778
4	4125	4059	66	10	25	4034	56	0.00247279920870425	0.691358024691358
5	4125	4063	62	9	28	4035	53	0.00222551928783383	0.654320987654321
6	4125	4066	59	8	30	4036	51	0.0019782393669634	0.62962962962963
7	4125	4069	56	8	33	4036	48	0.0019782393669634	0.592592592592593
10	4125	4072	53	6	34	4038	47	0.00148367952522255	0.580246913580247
13	4125	4076	49	5	37	4039	44	0.00123639960435213	0.54320987654321
17	4125	4080	45	3	39	4041	42	0.000741839762611276	0.518518518518518
23	4125	4084	41	3	43	4041	38	0.000741839762611276	0.469135802469136
33	4125	4088	37	2	46	4042	35	0.000494559841740851	0.432098765432099
46	4125	4092	33	2	50	4042	31	0.000494559841740851	0.382716049382716
59	4125	4096	29	2	54	4042	27	0.000494559841740851	0.333333333333333
70	4125	4100	25	2	58	4042	23	0.000494559841740851	0.283950617283951
111	4125	4105	20	1	62	4043	19	0.000247279920870425	0.234567901234568
147	4125	4110	15	0	66	4044	15	0	0.185185185185185
156	4125	4113	12	0	69	4044	12	0	0.148148148148148
194	4125	4115	10	0	71	4044	10	0	0.123456790123457
260	4125	4117	8	0	73	4044	8	0	0.0987654320987654
337	4125	4120	5	0	76	4044	5	0	0.0617283950617284
372	4125	4123	2	0	79	4044	2	0	0.0246913580246914
386	4125	4124	1	0	80	4044	1	0	0.0123456790123457

Table 5.1. The table of total positive and negative values for “test0.png”

Threshold	Total	TN+FN	FP+TP	FP	FN	TN	TP	FP/(FP+TN)	TP/(TP+FN)
1	4125	4017	108	71	10	4007	37	0.0174104953408534	0.787234042553192
2	4125	4036	89	52	10	4026	37	0.0127513487003433	0.787234042553192
3	4125	4040	85	50	12	4028	35	0.0122609122118686	0.74468085106383
4	4125	4045	80	45	12	4033	35	0.0110348209906817	0.74468085106383
5	4125	4049	76	41	12	4037	35	0.0100539480137322	0.74468085106383
6	4125	4052	73	38	12	4040	35	0.00931829328102011	0.74468085106383
7	4125	4053	72	37	12	4041	35	0.00907307503678274	0.74468085106383
10	4125	4059	66	32	13	4046	34	0.00784698381559588	0.723404255319149
13	4125	4062	63	30	14	4048	33	0.00735654732712114	0.702127659574468
17	4125	4069	56	23	14	4055	33	0.00564001961745954	0.702127659574468
23	4125	4075	50	18	15	4060	32	0.00441392839627268	0.680851063829787
33	4125	4081	44	14	17	4064	30	0.0034330554193232	0.638297872340426
46	4125	4092	33	6	20	4072	27	0.00147130946542423	0.574468085106383
59	4125	4100	25	5	27	4073	20	0.00122609122118686	0.425531914893617
70	4125	4105	20	4	31	4074	16	0.000980872976949485	0.340425531914894
111	4125	4111	14	2	35	4076	12	0.000490436488474743	0.25531914893617
147	4125	4114	11	2	38	4076	9	0.000490436488474743	0.191489361702128
156	4125	4114	11	2	38	4076	9	0.000490436488474743	0.191489361702128
194	4125	4115	10	2	39	4076	8	0.000490436488474743	0.170212765957447
260	4125	4118	7	2	42	4076	5	0.000490436488474743	0.106382978723404
337	4125	4121	4	1	44	4077	3	0.000245218244237371	0.0638297872340425
372	4125	4122	3	0	44	4078	3	0	0.0638297872340425
386	4125	4123	2	0	45	4078	2	0	0.0425531914893617

Table 5.2. The table of total positive and negative values for “test1.png”

Threshold	Total	TN+FN	FP+TP	FP	FN	TN	TP	FP/(FP+TN)	TP/(TP+FN)
1	4125	4019	106	65	16	4003	41	0.0159783677482793	0.719298245614035
2	4125	4037	88	50	19	4018	38	0.012291052114061	0.666666666666667
3	4125	4045	80	42	19	4026	38	0.0103244837758112	0.666666666666667
4	4125	4053	72	35	20	4033	37	0.00860373647984267	0.649122807017544
5	4125	4059	66	33	24	4035	33	0.00811209439528024	0.578947368421053
6	4125	4062	63	30	24	4038	33	0.00737463126843658	0.578947368421053
7	4125	4067	58	26	25	4042	32	0.0063913470993117	0.56140350877193
10	4125	4074	51	20	26	4048	31	0.00491642084562439	0.543859649122807
13	4125	4080	45	16	28	4052	29	0.00393313667649951	0.508771929824561
17	4125	4083	42	14	29	4054	28	0.00344149459193707	0.491228070175439
23	4125	4086	39	13	31	4055	26	0.00319567354965585	0.456140350877193
33	4125	4092	33	8	32	4060	25	0.00196656833824975	0.43859649122807
46	4125	4094	31	7	33	4061	24	0.00172074729596853	0.421052631578947
59	4125	4099	26	5	36	4063	21	0.0012291052114061	0.368421052631579
70	4125	4103	22	4	39	4064	18	0.000983284169124877	0.315789473684211
111	4125	4110	15	4	46	4064	11	0.000983284169124877	0.192982456140351
147	4125	4111	14	3	46	4065	11	0.000737463126843658	0.192982456140351
156	4125	4112	13	2	46	4066	11	0.000491642084562439	0.192982456140351
194	4125	4115	10	1	46	4069	9	0.000245700245700246	0.163636363636364
260	4125	4116	9	0	46	4070	9	0	0.163636363636364
337	4125	4121	4	0	51	4070	4	0	0.0727272727272727
372	4125	4123	2	0	53	4070	2	0	0.0363636363636364
386	4125	4124	1	0	54	4070	1	0	0.0181818181818182

Table 5.3. The table of total positive and negative values for “test2.png”