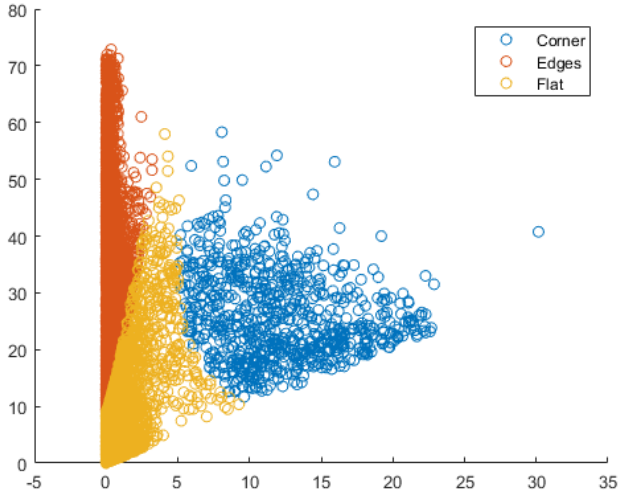
Fall 2019

Computer Vision Programming Documentation

Robson Adem

Question 1:

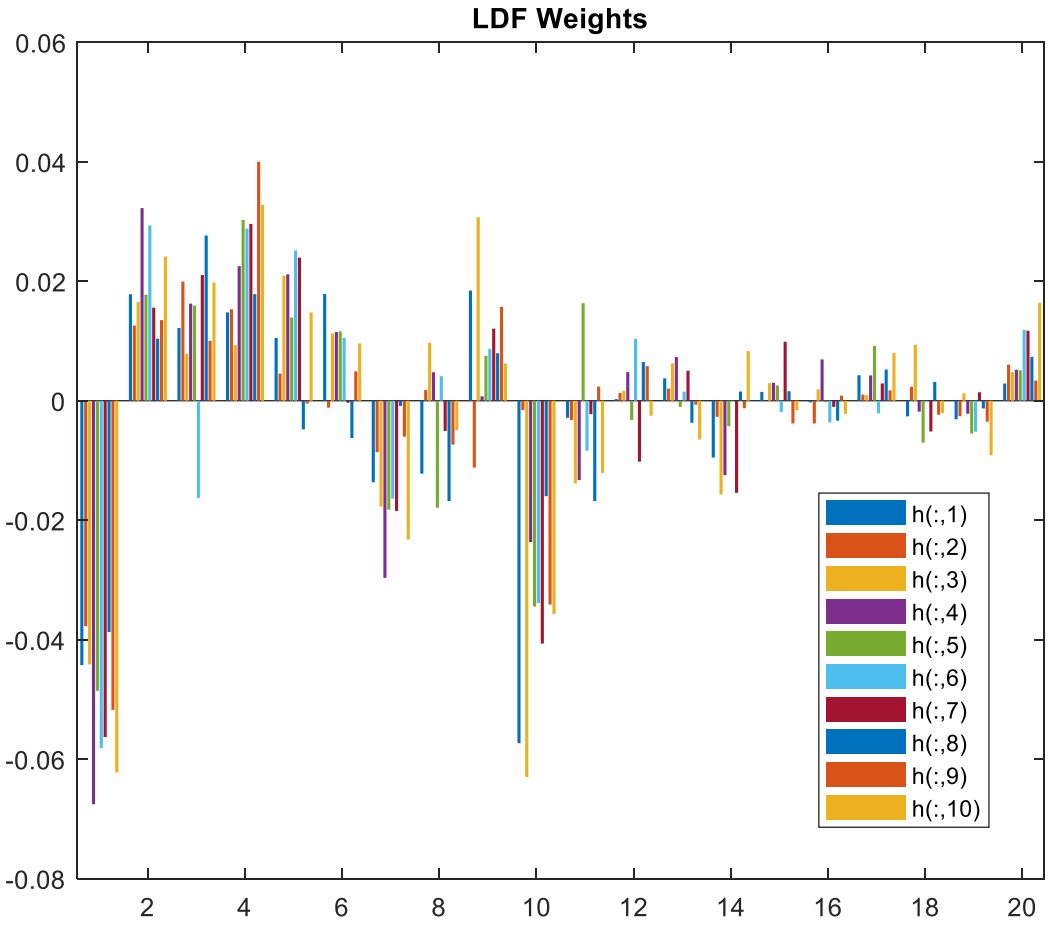
As shown in programming assignment two, we were able to implement the Harris corner detector algorithm to extract points of interests using the cornerness measure. Following that, we further implemented a method to classify interest points as corners, edges, and flat using normal distribution parameters and Mahalanobis distance measure. First, we defined a 2 x 1 vector using the eigenvalues for each interest point. Then, we computed mean and covariance as the parameters of normal distributions that best fit the edges, corners, and flat regions. Following that, we used Mahalanobis distance to classify each point using the estimated distributions and obtained a confusion matrix for each class as shown below.

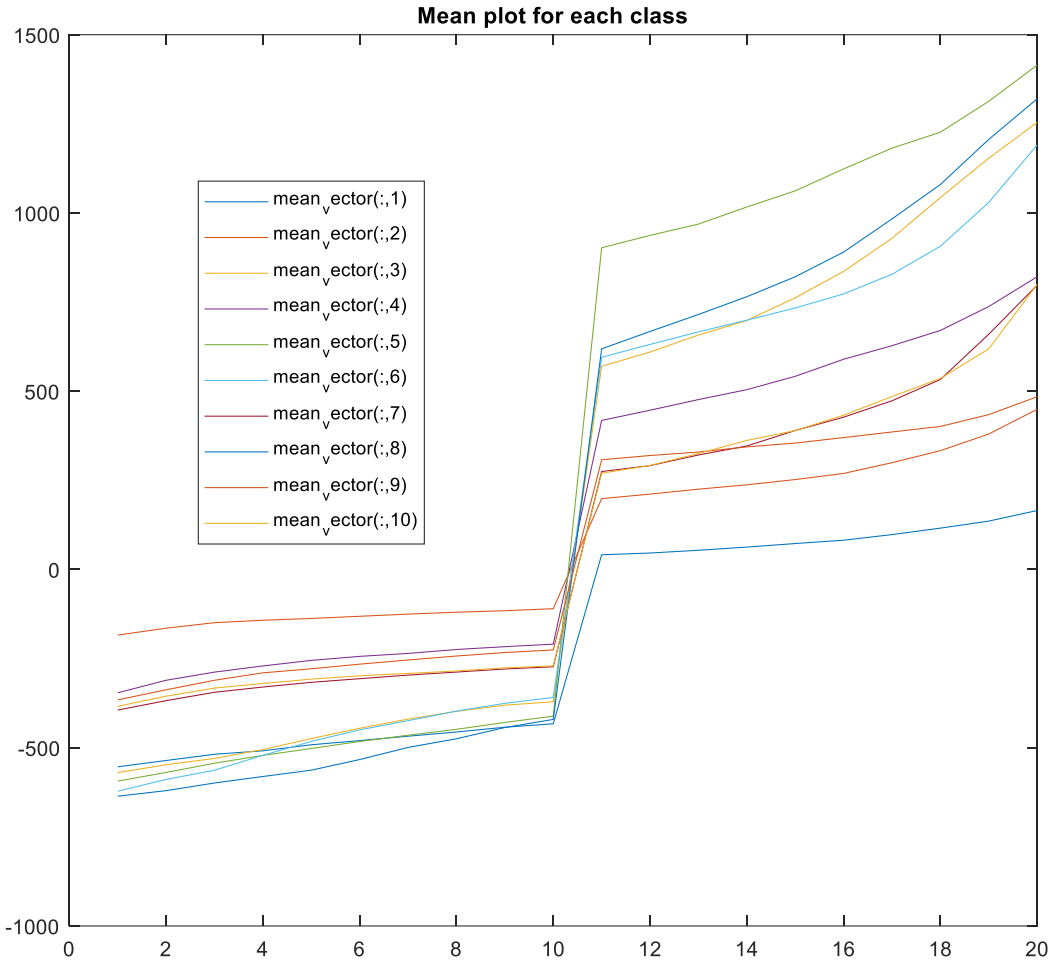


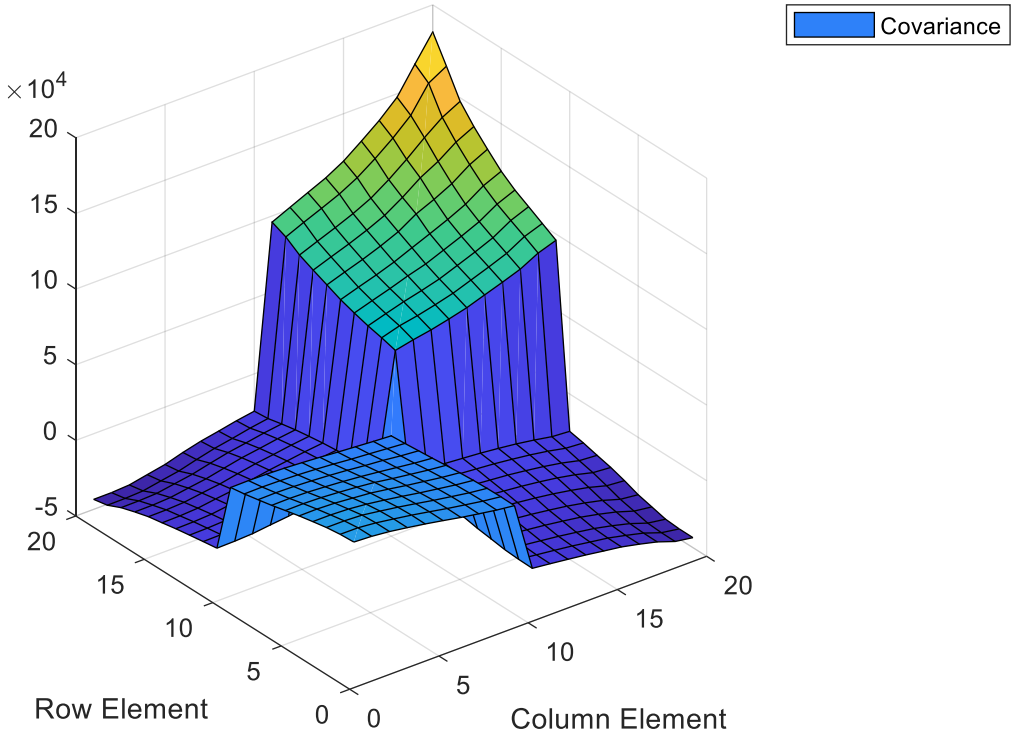
	Corner	Edges	Flat		
Corner	412	210	166	52%	48%
Edges	3885	7273	5756	43%	57%
Flat	218	129	104422	100%	0%

Question 2:

Provided with a dataset of digits with 10 classes (0 – 9), we extracted image features using the Harris Corner Detection algorithm. For each class, we used 50 images for training and then used different 50 sets of images to test. For each image, we created a 20 x 1 feature vector using the 10 most positive and 10 most negative R scores. Then, with the training feature vector for each class, we set up a Bayes linear discriminant function for recognizing each class. Following that, we obtained confusion matrix to evaluate probability of correct recognition and probability of error for each class. The results are shown as follows.




Mean plot for each class
1500
1000
500
0
-500
-1000
0
2
4
6
8
10
12
14
16
18
20
mean_vector(:,1)
mean_vector(:,2)
mean_vector(:,3)
mean_vector(:,4)
mean_vector(:,5)
mean_vector(:,6)
mean_vector(:,7)
mean_vector(:,8)
mean_vector(:,9)
mean_vector(:,10)



	0	1	2	3	4	5	6	7	8	9		
0	49	0	0	0	0	0	0	0	1	0	98%	2%
1	2	41	0	2	1	0	3	1	0	0	82%	18%
2	0	0	30	0	2	6	4	6	0	2	60%	40%
3	0	3	1	29	0	5	4	0	3	5	58%	42%
4	0	0	0	0	45	4	0	1	0	0	90%	10%
5	0	0	5	3	2	30	1	6	0	3	60%	40%
6	0	2	1	3	0	1	29	0	2	12	58%	42%
7	0	0	8	1	5	1	3	29	0	3	58%	42%
8	0	1	0	1	0	0	0	0	48	0	96%	4%
9	0	0	0	4	0	0	10	0	0	36	72%	28%
										Pc	*73%*	27%

Confusion Matrix for Training Data

	0	1	2	3	4	5	6	7	8	9		
0	1	28	0	5	0	0	2	0	13	1	2%	98%
1	0	8	8	0	1	0	11	2	0	20	16%	84%
2	0	29	0	13	0	0	1	0	5	2	0%	100%
3	0	3	3	20	23	0	0	0	1	0	40%	60%
4	0	32	0	9	0	1	3	0	1	4	0%	100%
5	0	18	0	9	0	1	10	0	3	9	2%	98%
6	0	26	0	11	0	0	4	0	1	8	8%	92%
7	0	19	0	7	0	0	4	0	17	3	0%	100%
8	0	49	0	1	0	0	0	0	0	0	0%	100%
9	0	8	0	23	0	1	0	0	16	2	4%	96%
											7%	93%

Confusion Matrix for Test Data

References: Lecture notes and slides