ECE 30200 Project 2

Name: Vaibhav Ramachandran

The task in the project was to extract as much of the cat as possible from a 500 x 375-pixel picture containing a cat with grass in the background. We had to classify the image into its foreground and background and highlight them as such.

**Problem 1:**

The main purpose of this problem was to write a MATLAB function that would compute the mean and covariance matrix of the train\_cat and train\_grass column matrices that were extracted from the data file called training\_data.mat. The train\_cat matrix was 64 x 1976 and the train\_grass matrix was 64 x 9556. The goal was to compute the mean and covariance matrix of the rows such that the mean vectors are 64 x 1 and the covariance matrices are 64 x 64. I used the mean() and cov() MATLAB functions to accomplish this.

My MATLAB function file, my\_training.m looks like:

Y = im2double(imread('cat\_grass.jpg'));

load('training\_data.mat');

K\_cat = 1976;

K\_grass = 9556;

function [mu\_cat, mu\_grass, sigma\_cat, sigma\_grass] = my\_training( train\_cat, train\_grass )

mu\_cat = mean(train\_cat,2);

mu\_grass = mean(train\_grass,2);

sigma\_cat = cov(transpose(train\_cat));

sigma\_grass = cov(transpose(train\_grass));

end

**Problem 2:**

The main purpose of this problem was to highlight all the 8 x 8 patches which belong to the cat such that we can display the portion where the cat is present. In order to perform this a Maximum-A-Posteriori decision with a multivariate Gaussian distribution was used. Z was assumed to be the random variable vector that would represent an 8 x 8 patch of the image. Using the multivariate Gaussian PDF formula, the conditional probability of Z given the class label was calculated. The equations used were:

Here d = 64, since the dimensionality is 64. Next using Baye’s theorem, the posterior probabilities are computed which are the conditional probability of the class label given the patch Z. If the posterior probability of the cat is greater than that of the grass, then that patch is highlighted. The formulas for the posterior probabilities were:

The patch would be highlighted if the following condition were true:

A double for loop was used to achieve this. The MATLAB function, my\_testing.m looked as follows:

function [ Output ] = my\_testing( Y, mu\_cat, mu\_grass, sigma\_cat, sigma\_grass, K\_cat, K\_grass )

M = 375;

N = 500;

d = 64;

Output = zeros(M-8,N-8);

inverse\_cat = pinv(sigma\_cat);

inverse\_grass = pinv(sigma\_grass);

determinant\_cat = det(sigma\_cat);

determinant\_grass = det(sigma\_grass);

f\_cat = K\_cat/(K\_cat + K\_grass);

f\_grass = K\_grass/(K\_cat + K\_grass);

for i=1:M-8

for j=1:N-8

z = Y(i+[0:7],j+[0:7]);

z\_vector = z(:);

f\_z\_given\_cat = exp(-0.5 \* transpose((z\_vector - mu\_cat)) \* inverse\_cat \* (z\_vector - mu\_cat)) / ((2\*pi)^(d/2) \* sqrt(determinant\_cat));

f\_z\_given\_grass = exp(-0.5 \* transpose((z\_vector - mu\_grass)) \* inverse\_grass \* (z\_vector - mu\_grass)) / ((2\*pi)^(d/2) \* sqrt(determinant\_grass));

if((f\_z\_given\_cat \* f\_cat) > (f\_z\_given\_grass \* f\_grass))

Output(i,j) = 1;

end

end

end

end

**Problem 3:**

The main purpose of this problem was to display the result of Output using the imshow() command in MATLAB. The final image obtained after running the code is shown below. Also the output calculation run time was also obtained using the tic-toc command.

imshow(Output);

tic

[ Output ] = my\_testing( Y, mu\_cat, mu\_grass, sigma\_cat, sigma\_grass, K\_cat, K\_grass );

toc

Elapsed time is 4.432460 seconds.

The run-time was found to be 4.432460 seconds. The final image using imshow(Output) is:



Fig 1: Result of Problem 2. Image classification of the Cat and Grass image obtained using imshow(). Shape of cat is highlighted in white.

**Problem 4:**

The main purpose of this problem was to speed up the run time of the previous program so that not a lot of time is wasted computing. To minimize the computing load inside the double for-loops, I calculated the inverse of the covariance matrices prior to the start of the loops. Also I calculated the determinants of the sigma matrices outside the double for-loop. Finally I took the log of the posterior probabilities and compared them thereby eliminating the exponential term in the double for-loop. It got rid of multiplication as well. The updated MATLAB function, my\_testing\_updated.m is shown below:

function [ Output ] = my\_testing\_updated( Y, mu\_cat, mu\_grass, sigma\_cat, sigma\_grass, K\_cat, K\_grass )

M = 375;

N = 500;

d = 64;

Output = zeros(M-8,N-8);

inverse\_cat = pinv(sigma\_cat);

inverse\_grass = pinv(sigma\_grass);

determinant\_cat = det(sigma\_cat);

determinant\_grass = det(sigma\_grass);

f\_cat = K\_cat/(K\_cat + K\_grass);

f\_grass = K\_grass/(K\_cat + K\_grass);

coeff\_cat = log(1 / ((2\*pi)^(d/2) \* sqrt(determinant\_cat)));

coeff\_grass = log(1 / ((2\*pi)^(d/2) \* sqrt(determinant\_grass)));

for i=1:M-8

for j=1:N-8

z = Y(i+[0:7],j+[0:7]);

z\_vector = z(:);

f\_z\_given\_cat = (-0.5 \* transpose((z\_vector - mu\_cat)) \* inverse\_cat \* (z\_vector - mu\_cat)) + coeff\_cat;

f\_z\_given\_grass = (-0.5 \* transpose((z\_vector - mu\_grass)) \* inverse\_grass \* (z\_vector - mu\_grass)) + coeff\_grass;

if((f\_z\_given\_cat + log(f\_cat)) > (f\_z\_given\_grass + log(f\_grass)))

Output(i,j) = 1;

end

end

end

end

imshow(Output);

tic

[ Output ] = my\_testing\_updated( Y, mu\_cat, mu\_grass, sigma\_cat, sigma\_grass, K\_cat, K\_grass );

toc

Elapsed time is 4.063950 seconds.

The run-time was found to be 4.063950 seconds. The final image using imshow(Output) is:



Fig 2: Result of Problem 4, updating the my\_testing.m function. Image classification of the Cat and Grass image obtained using imshow(). Shape of cat is highlighted in white.