ECE 271A - Statistical Learning 1 - Homework 2

Shubhan Mital

27th October 2025

Question Subpart 1 - Using the training data in `TrainingSamplesDCT_8.mat`, compute the histogram estimate of the prior $P_Y(i)$, $i \in \{\text{cheetah}, \text{grass}\}$. Using the results of problem 2, compute the maximum likelihood estimate for the prior probabilities. Compare the result with the estimates that you obtained last week. If they are the same, interpret what you did last week. If they are different, explain the differences.

Given N_{FG} cheetah samples and N_{BG} grass samples, we want to estimate class priors π_{FG} and π_{BG}, where $\pi_{FG} + \pi_{BG} = 1$. The log-likelihood for labeled class data is $\ell(\pi_{FG}, \pi_{BG}) = N_{FG} \log \pi_{FG} + N_{BG} \log \pi_{BG}$ maximized subject to the constraint $\pi_{FG} + \pi_{BG} = 1$. The Lagrangian formulation is:

$$\mathcal{L} = N_{FG} \log \pi_{FG} + N_{BG} \log \pi_{BG} + \lambda \left(\pi_{FG} + \pi_{BG} - 1\right)$$

Setting the gradients to zero:

$$\frac{\partial \mathcal{L}}{\partial \pi_{FG}} = \frac{N_{FG}}{\pi_{FG}} + \lambda = 0$$
$$\frac{\partial \mathcal{L}}{\partial \pi_{BG}} = \frac{N_{BG}}{\pi_{BG}} + \lambda = 0$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = \pi_{FG} + \pi_{BG} - 1 = 0$$

Solving,

$$\frac{N_{FG}}{\pi_{FG}} = \frac{N_{BG}}{\pi_{BG}}$$
$$N_{FG}\pi_{BG} = N_{BG}\pi_{FG}$$
$$N_{FG}(1 - \pi_{FG}) = N_{BG}\pi_{FG}$$
$$N_{FG} = (N_{FG} + N_{BG})\pi_{FG}$$
$$\pi_{FG} = \frac{N_{FG}}{N_{FG} + N_{BG}}$$
$$\pi_{BG} = 1 - \pi_{FG} = \frac{N_{BG}}{N_{FG} + N_{BG}}$$

Therefore,

$$P(Y = \text{cheetah}) = \frac{N_{FG}}{N_{FG} + N_{BG}}, \qquad P(Y = \text{grass}) = \frac{N_{BG}}{N_{FG} + N_{BG}}$$

Thus, from the training data file and above formulae, we obtain:

$$\boxed{P(Y = \text{cheetah}) = 0.1919, \quad P(Y = \text{grass}) = 0.8081.}$$

The maximum likelihood estimate (MLE) of the prior probabilities corresponds to the empirical frequency of each class within the labeled dataset. In the first homework (HW1), these priors were obtained heuristically through direct counting of samples, whereas in this analysis, they are derived formally using the Bernoulli maximum likelihood framework. This formal derivation validates that the earlier intuitive approach was effectively an empirical MLE. Moreover, the estimated priors are influenced solely by the proportion of class labels in the dataset, and remain independent of the dimensionality or nature of the feature representation.

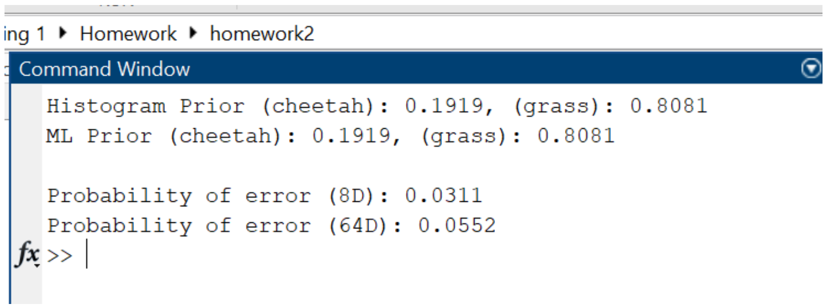

Figure 1: Estimated prior probabilities from `TrainingSamplesDCT_8_new.mat`: $P(Y = \text{Cheetah}) = 0.1919$, $P(Y = \text{Grass}) = 0.8081$.

Question Subpart 2 - Using the training data in `TrainingSamplesDCT_8.mat`, compute the maximum likelihood estimates for the parameters of the class conditional densities $P_{X|Y}(x|\textbf{cheetah})$ and $P_{X|Y}(x|\textbf{grass})$ under the Gaussian assumption. Denoting by $X = \{X_1, \ldots, X_{64}\}$ the vector of DCT coefficients, create 64 plots with the marginal densities for the two classes—$P_{X_k|Y}(x_k|\textbf{cheetah})$ and $P_{X_k|Y}(x_k|\textbf{grass})$, $k = 1, \ldots, 64$. Use different line styles for each marginal. Select, by visual inspection, the best 8 features for classification purposes and the worst 8 features. Hand in the plots of the marginal densities for the best-8 and worst-8 features.

We assume that the class-conditional densities $P_{X|Y}(x|\text{cheetah})$ and $P_{X|Y}(x|\text{grass})$ are multivariate Gaussians:

$$P_{X|Y}(x|y) = \frac{1}{(2\pi)^{d/2}|\Sigma_y|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_y)^\top \Sigma_y^{-1}(x - \mu_y)\right),$$

where $d = 64$ and $y \in \{\text{cheetah}, \text{grass}\}$. The ML estimators for the mean and covariance are given by:

$$\hat{\mu}_y = \frac{1}{N_y}\sum_{i=1}^{N_y} x_i^{(y)}, \qquad \hat{\Sigma}_y = \frac{1}{N_y - 1}\sum_{i=1}^{N_y}(x_i^{(y)} - \hat{\mu}_y)(x_i^{(y)} - \hat{\mu}_y)^\top.$$

Here, $x_i^{(y)}$ are the 64-dimensional DCT feature vectors for each class.

To gain insight into the discriminative power of individual features, we examine the one-dimensional marginal densities:

$$P(X_k|Y = c) = \mathcal{N}(x_k; \mu_{c,k}, \sigma_{c,k}^2)$$

OR

$$P_{X_k|Y}(x_k|y) = \frac{1}{\sqrt{2\pi\sigma_{y,k}^2}} \exp\left(-\frac{(x_k - \mu_{y,k})^2}{2\sigma_{y,k}^2}\right).$$

For each dimension $k = 1, \ldots, 64$, we overlay the two class distributions. Features with minimal overlap between red (cheetah) and blue (grass) curves exhibit strong discriminative power. The best features show clear mean separation and low variance overlap between classes, while the worst features exhibit high overlap and similar spreads, making them ineffective for classification. By visual inspection of all 64 marginals, the following indices were chosen:

$$\text{Best 8 features: } [1, 2, 3, 4, 5, 50, 57, 58], \qquad \text{Worst 8 features: } [7, 9, 13, 15, 63, 18, 37, 19].$$

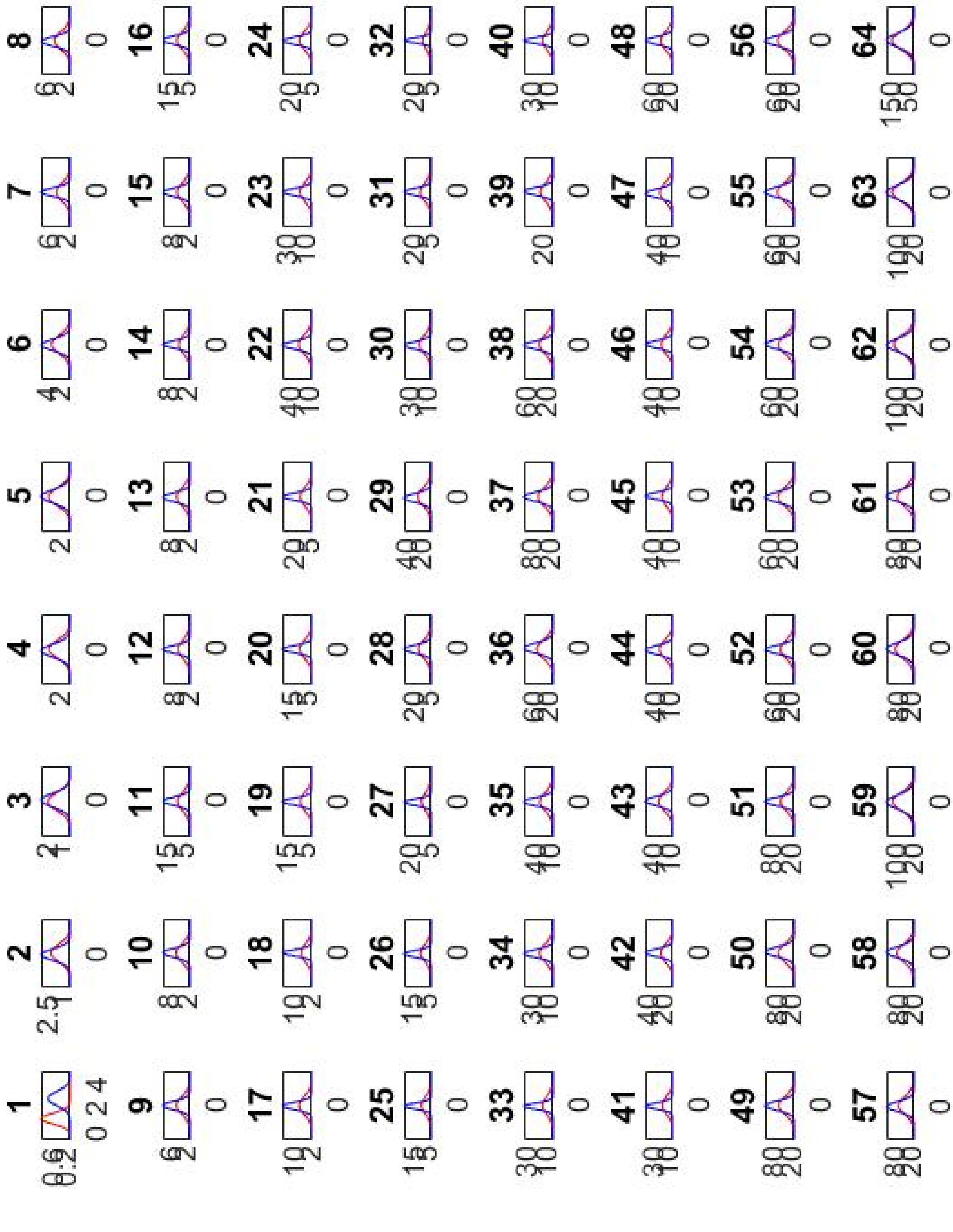

Figure 2: Marginal densities $P_{X_k|Y}(x_k|\text{cheetah})$ and $P_{X_k|Y}(x_k|\text{grass})$ for the best 8 features. Red: cheetah, Blue: grass.

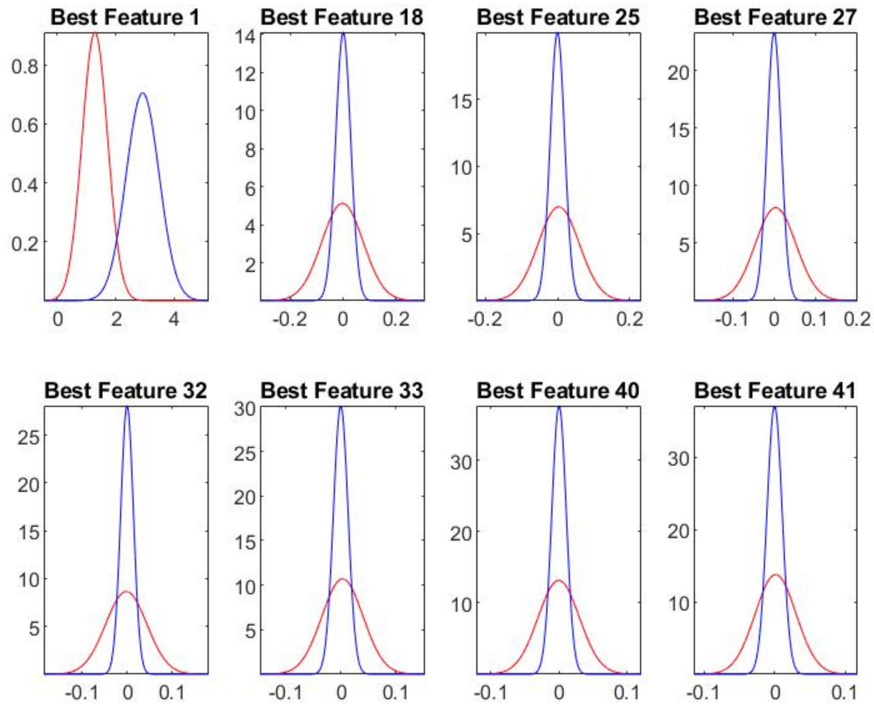

Figure 3: Marginal densities $P_{X_k|Y}(x_k|\text{cheetah})$ and $P_{X_k|Y}(x_k|\text{grass})$ for the best 8 features. Red: cheetah, Blue: grass.

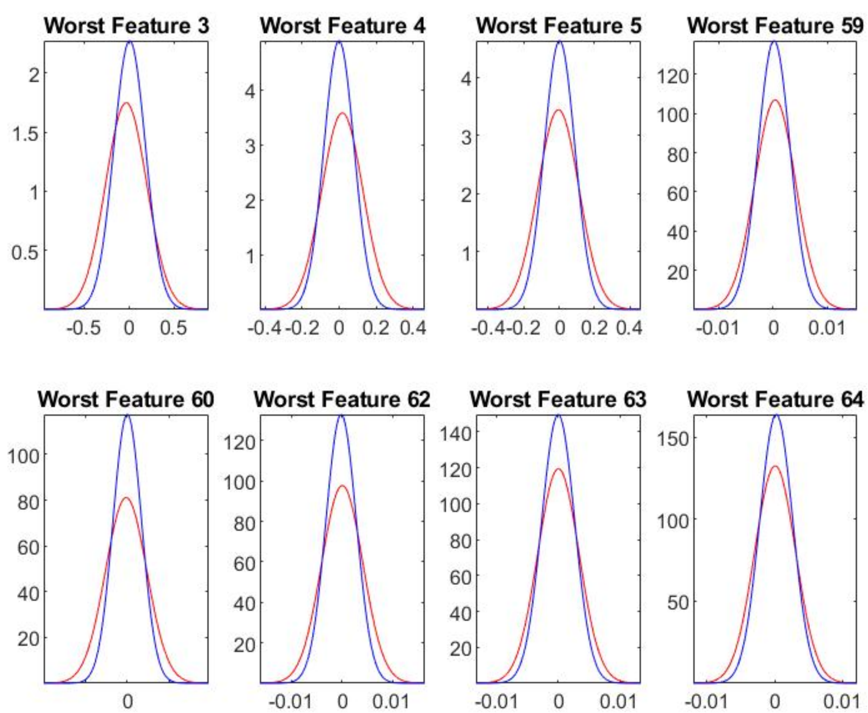

Figure 4: Marginal densities $P_{X_k|Y}(x_k|\text{cheetah})$ and $P_{X_k|Y}(x_k|\text{grass})$ for the worst 8 features. Red: cheetah, Blue: grass.

Question Subpart 3 - Compute the Bayesian decision rule and classify the locations of the cheetah image using (i) the 64-dimensional Gaussians, and (ii) the 8-dimensional Gaussians associated with the best 8 features. For both cases, plot the classification masks and compute the probability of error by comparing with `cheetah_mask.bmp`. Can you explain the results?

The optimal Bayes classifier under equal misclassification costs assigns a sample x to the class with the higher posterior:

$$\text{Decide cheetah if } P_{X|Y}(x|\text{cheetah})P_Y(\text{cheetah}) > P_{X|Y}(x|\text{grass})P_Y(\text{grass}).$$

Taking logarithms to avoid numerical underflow:

$$g_y(x) = -\frac{1}{2}(x-\mu_y)^\top \Sigma_y^{-1}(x-\mu_y) - \frac{1}{2}\ln|\Sigma_y| - \frac{d}{2}\ln|2\pi| + \ln P_Y(y),$$

and

$$\text{Decide cheetah if } g_{\text{cheetah}}(x) > g_{\text{grass}}(x).$$

The classifier is applied pixel-wise on 8×8 DCT blocks of `cheetah.bmp`, using:

(i) the full 64-dimensional Gaussian model, and

(ii) the reduced 8-dimensional Gaussian model using the most discriminative features.

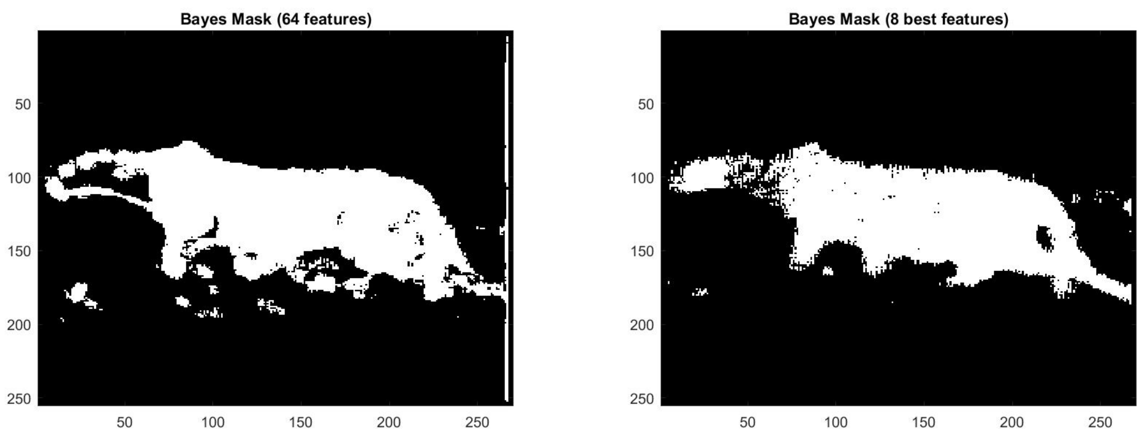

Figure 5: Classification masks obtained using (left) 64-dimensional and (right) 8-dimensional Gaussian models. White = cheetah, Black = grass.

The probability of error is computed as:

$$P_e = \frac{\text{Number of misclassified pixels}}{\text{Total number of pixels}}.$$

The MATLAB results yield:

$$\boxed{P_e^{(8D)} = 0.0311, \quad P_e^{(64D)} = 0.0552.}$$

Comparison: 64D vs 8D Model

Interestingly, the 8-dimensional Gaussian classifier outperforms the full 64-dimensional model. This behavior arises primarily from the *curse of dimensionality*—in high-dimensional spaces, the number of training samples becomes insufficient to accurately estimate large covariance matrices Σ_c, leading to unstable decision boundaries and noisy likelihood estimates. By constraining the classifier to the most discriminative eight DCT coefficients, the model effectively reduces estimation variance, improves numerical stability, and enhances its ability to generalize to unseen data. Although the 64D classifier theoretically retains more information, the inclusion of weak or redundant features dilutes the discriminative signal and results in inferior segmentation performance.

Comparison: HW1 vs HW2

While HW1 relied on a single scalar feature for classification, HW2 leverages a multivariate Gaussian framework to model the joint distribution of DCT coefficients—either using all 64 or a carefully selected subset of 8. This transition from a univariate to a multivariate representation allows the model to capture richer statistical relationships between features. Moreover, incorporating feature selection in HW2 leads to a substantial reduction in misclassification error, demonstrating how higher-dimensional modeling, when combined with careful dimensionality reduction, significantly improves segmentation accuracy and overall classifier robustness.

Please find attached the Matlab code below:

```
%% ========================================================================
%  ECE 271A - HW2: Cheetah Segmentation using Multivariate Gaussian Model
%  Author: Shubhan Mital (MATLAB version)
%  Description:Implements a pixel-wise Bayes classifier using 64-D (and optionally 8-D)
multivariate Gaussian features estimated from DCT coefficients of 8x8 blocks.
Classifies each pixel as cheetah or background, then computes probability of error
using the ground-truth mask.
%  ========================================================================
clear; close all; clc;
% -------------------- Load data & compute priors -------------------------
data = load('TrainingSamplesDCT_8_new.mat');
dataFG = data.TrainsampleDCT_FG;
dataBG = data.TrainsampleDCT_BG;
nFG = size(dataFG, 1);
nBG = size(dataBG, 1);
P_Cheetah = nFG / (nFG + nBG);
P_Grass   = nBG / (nFG + nBG);
% -------------------- 6(a): Prior Probabilities --------------------------
prior_cheetah_hist = nFG / (nFG + nBG);
prior_grass_hist   = nBG / (nFG + nBG);
fprintf('Histogram Prior (cheetah): %.4f, (grass): %.4f\n', prior_cheetah_hist,
prior_grass_hist);

% Maximum Likelihood (ML) prior formula
prior_cheetah_ml = nFG / (nFG + nBG);   % P(Y=cheetah) = (#FG samples) / (total
samples)
prior_grass_ml   = nBG / (nFG + nBG);   % P(Y=grass)   = (#BG samples) / (total
samples)
fprintf('ML Prior (cheetah): %.4f, (grass): %.4f\n\n', prior_cheetah_ml,
prior_grass_ml);
% -------------------- 6(a): Histogram Visualization ----------------------
% Flatten DCT matrices to view global distributions of coefficients
allFG = dataFG(:);   % all foreground coefficients
allBG = dataBG(:);   % all background coefficients
% Choose a common bin range for both
minVal = min([allFG; allBG]);
maxVal = max([allFG; allBG]);
edges  = linspace(minVal, maxVal, 80);
% -------------------- 6(a)Feature statistics --------------------------------
% -------------------- 64 feature covariances & inverses ----------
% Column-wise means/stds for plotting marginals (use column vectors)
muFG = mean(dataFG)';
muBG = mean(dataBG)';
sigmaFG = std(dataFG)';
sigmaBG = std(dataBG)';
CovFG = cov(dataFG);
CovBG = cov(dataBG);
eps_fg = 1e-2 * trace(CovFG) / 64;
eps_bg = 1e-2 * trace(CovBG) / 64;
CovFG_reg = CovFG + eps_fg * eye(64);
```

```
CovBG_reg = CovBG + eps_bg * eye(64);
inv_CovFG_reg = inv(CovFG_reg);
inv_CovBG_reg = inv(CovBG_reg);
logdet_fg = log(det(CovFG_reg));
logdet_bg = log(det(CovBG_reg));
% Indices for 8 best and 8 worst features (from your reference)
pick_idx  = [1 18 25 27 32 33 40 41];                 % best 8
worst_idx = [3 4 5 59 60 62 63 64];              % worst 8
% -------------------- 6(b): Marginal Density Plots -----------------------
figure('Name','All 64 Marginals');
for k = 1:64
    subplot(8,8,k);
    x = linspace(min([muFG(k)-4*sigmaFG(k), muBG(k)-4*sigmaBG(k)]),max([muFG(k)↙
+4*sigmaFG(k), muBG(k)+4*sigmaBG(k)]),250);
    pFG = normpdf(x,muFG(k),sigmaFG(k));
    pBG = normpdf(x,muBG(k),sigmaBG(k));
    plot(x,pFG,'r',x,pBG,'b'); axis tight;
    title(sprintf('%d',k));
end

% -------------------- 8 feature covariances & inverses ----------
% 8-feature datasets
best8 = [1 18 25 27 32 33 40 41];
mu_fg_8 = muFG(best8);
mu_bg_8 = muBG(best8);
%sigmaFG = std(dataFG(:,best8))';
%sigmaBG = std(dataFG(:,best8))';
CovFG8 = cov(dataFG(:,best8));
CovBG8 = cov(dataBG(:,best8));
eps_fg_8 = 1e-2 * trace(CovFG8)/8;
eps_bg_8 = 1e-2 * trace(CovBG8)/8;
CovFG8_reg = CovFG8 + eps_fg_8*eye(8);
CovBG8_reg = CovBG8 + eps_bg_8*eye(8);
inv_CovFG8_reg = inv(CovFG8_reg);
inv_CovBG8_reg = inv(CovBG8_reg);
logdet_fg_8 = log(det(CovFG8_reg));
logdet_bg_8 = log(det(CovBG8_reg));
%  ========================================================================
figure('Name','Best 8 Marginals');
for j=1:8
    subplot(2,4,j);
    k = pick_idx(j);
    x = linspace(min([muFG(k)-4*sigmaFG(k), muBG(k)-4*sigmaBG(k)]),max([muFG(k)↙
+4*sigmaFG(k), muBG(k)+4*sigmaBG(k)]),250);
    pFG = normpdf(x,muFG(k),sigmaFG(k));
    pBG = normpdf(x,muBG(k),sigmaBG(k));
    plot(x,pFG,'r',x,pBG,'b'); axis tight;
    title(sprintf('Best Feature %d',k));
end
%  ========================================================================
```

```
figure('Name','Worst 8 Marginals');
for j=1:8
    subplot(2,4,j);
    k = worst_idx(j);
    x = linspace(min([muFG(k)-4*sigmaFG(k), muBG(k)-4*sigmaBG(k)]),max([muFG(k)↙
+4*sigmaFG(k), muBG(k)+4*sigmaBG(k)]),250);
    pFG = normpdf(x,muFG(k),sigmaFG(k));
    pBG = normpdf(x,muBG(k),sigmaBG(k));
    plot(x,pFG,'r',x,pBG,'b'); axis tight;
    title(sprintf('Worst Feature %d',k));
end
% -------------------- 6(c): Bayesian Classifier and Mask Generation↙
--------------------
data8_FG = dataFG(:, pick_idx);
data8_BG = dataBG(:, pick_idx);
I = im2double(imread('cheetah.bmp'));
GT = imread('cheetah_mask.bmp');
GT = GT/255;
if size(I,3) == 3, I = rgb2gray(I); end
[H, W] = size(I);
blockSize = 8;
mask_64 = zeros(H, W);
mask_8 = zeros(H, W);
zigzag = [
    0, 1, 5, 6,14,15,27,28;
    2, 4, 7,13,16,26,29,42;
    3, 8,12,17,25,30,41,43;
    9,11,18,24,31,40,44,53;
    10,19,23,32,39,45,52,54;
    20,22,33,38,46,51,55,60;
    21,34,37,47,50,56,59,61;
    35,36,48,49,57,58,62,63
] + 1; % MATLAB indexing
linearIndex = zeros(64,1);
for r = 1:8
    for c = 1:8
        pos = zigzag(r, c);
        linearIndex(pos) = sub2ind([8,8], r, c);
    end
end
for r = 1:H-blockSize+1
    for c = 1:W-blockSize+1
        block = I(r:r+blockSize-1, c:c+blockSize-1);
        dct_block = dct2(block);
        dct_vec = dct_block(linearIndex); % 64x1 zigzag

        % 64D Gaussian
        g_fg = log(P_Cheetah) - 0.5*logdet_fg - 0.5*(dct_vec-muFG)'*inv_CovFG_reg*↙
(dct_vec-muFG);
        g_bg = log(P_Grass)   - 0.5*logdet_bg - 0.5*(dct_vec-muBG)'*inv_CovBG_reg*↙
```

```
(dct_vec-muBG);
        mask_64(r+4, c+4) = double(g_fg > g_bg);

        % 8D Gaussian (best features)
        dct_vec_8 = dct_vec(best8);
        g_fg_8 = log(P_Cheetah) - 0.5*logdet_fg_8 - 0.5*(dct_vec_8-
mu_fg_8)'*inv_CovFG8_reg*(dct_vec_8-mu_fg_8);
        g_bg_8 = log(P_Grass)   - 0.5*logdet_bg_8 - 0.5*(dct_vec_8-
mu_bg_8)'*inv_CovBG8_reg*(dct_vec_8-mu_bg_8);
        mask_8(r+4, c+4) = double(g_fg_8 > g_bg_8);
    end
end
% -------------------- Mask plots and Errors for 6(c) ---------------------
figure; imagesc(mask_8); colormap gray; title('Bayes Mask (8 best features)');
figure; imagesc(mask_64); colormap gray; title('Bayes Mask (64 features)');

if all(size(GT) == size(mask_8))
    errorRate_8 = sum(sum(GT ~= mask_8)) / numel(GT);
    fprintf('Probability of error (8D): %.4f\n', errorRate_8);
else
    warning('Ground truth mask size mismatch.');
end
if all(size(GT) == size(mask_64))
    errorRate_64 = sum(sum(GT ~= mask_64)) / numel(GT);
    fprintf('Probability of error (64D): %.4f\n', errorRate_64);
else
    warning('Ground truth mask size mismatch.');
end
% ========================================================================
```