Lab Assignment: HOG Feature Extraction and Analysis

Objective: The objective of this lab is to understand the process of extracting Histogram of Oriented Gradients (HOG) features from an image. You will experiment with different parameters such as cell size and block size to observe their effect on HOG features. Additionally, you will apply HOG feature extraction on different types of images and reflect on the results.

Instructions:

Part 1: HOG Feature Extraction on Your Own Image

1. Use your own image as input for the HOG feature extraction. Replace test.jpg in the given code with the path to your personal image.
2. Run the provided code to calculate the HOG features.
3. Observe the output and save the visualization of the following:
   - Gamma-normalized image
   - Gradient Magnitude of the image
   - Gradient orientation visualization using quiver
   - HOG features visualization for the dominant orientation per cell
4. Save the results in a PDF format for submission.

Part 2: Experiment with HOG Parameters

1. Modify the cell_size and block_size parameters in the provided code and observe how the changes affect the resulting HOG features.
2. Run the code using the following parameter values:
   - **Cell size:** 8x8 pixels, **Block size:** 2x2 cells
   - **Cell size:** 16x16 pixels, **Block size:** 2x2 cells
3. Answer the following questions in your reflection:
   - How does changing the cell size and block size affect the HOG visualization?
   - Which cell size and block size provide the most visually distinguishable features?
4. Include the HOG visualization results from each parameter setting in your PDF submission.

PTO

Part 3: HOG Feature Extraction on a Car or Truck Image

1. Obtain an image of a car or a truck and use it as input for the HOG feature extraction.
2. Use the original cell_size of 4x4 and block_size of 2x2 to calculate the HOG features for the car/truck image.
3. Save the following visualizations for your report:
4.
   - Gamma-normalized image of the car/truck
   - Gradient magnitude and gradient orientation visualization
   - HOG features visualization for the dominant orientation per cell
5. Write a reflection discussing the differences in the HOG feature representation of the car/truck image compared to your own face image.

Submission Requirements:

- A single PDF document containing:
  - Visualizations from each task.
  - Reflections on changing parameters and comparing different types of images.
- Answer the following questions in your reflection:
  1. What effect did changing the cell_size and block_size parameters have on the HOG features?
  2. How did the HOG features of your face compare to those of a car/truck?
  3. Which parameter settings do you think would work best for distinguishing between different types of images?

Optional Challenge:

- Try using a different gamma value for normalization (e.g., gamma = 0.5 or gamma = 1.5). How does this impact the visualization of gradient magnitude and HOG features? Include these observations in your PDF.

Evaluation Criteria:

- Correct implementation of HOG feature extraction.
- Clarity and completeness of your PDF submission.

Make sure to experiment and reflect on your results! The more insights you gain, the better your understanding of HOG feature extraction will be.

PTO

```
close all;
clear all;
clc;

% Load necessary package
pkg load image;

% Load and preprocess the image
img = imread('test.jpg'); % Replace with the path to your image
if size(img, 3) == 3
    img = rgb2gray(img); % Convert to grayscale if the image is in color
end

% Resize the image to 128x64 for standardization
img = imresize(img, [128, 64]);

% Gamma normalization
gamma = 0.9; % Example gamma value
img = im2double(img) .^ gamma; % Convert to double and apply gamma normalization

figure;
imshow(img);
title('Gamma-normalized Image');

% Parameters for HOG
cell_size = 4;         % Size of each cell (4x4 pixels)
block_size = 2;        % Number of cells per block (2x2 cells)
num_bins = 9;          % Number of orientation bins (0° to 180°)
bin_size = 180 / num_bins;

% Compute gradients using Sobel operator on the entire image
Gx = imfilter(double(img), [-1 0 1; -2 0 2; -1 0 1], 'conv'); % Horizontal gradient
Gy = imfilter(double(img), [-1 -2 -1; 0 0 0; 1 2 1], 'conv'); % Vertical gradient

% Compute gradient magnitude and orientation
magnitude = sqrt(Gx.^2 + Gy.^2);
orientation = atan2d(Gy, Gx);
orientation(orientation < 0) += 180; % Convert orientation to [0, 180] range

% Display gradient magnitude and orientation
figure;
imshow(magnitude, []);
title('Gradient Magnitude of the Image');

% Display gradient orientation using quiver
figure;
imshow(img); % Show original image as background
hold on;

% Downsample the quiver plot for better visualization, e.g., every 4th pixel
step = 4; % Adjust the step to make the plot less dense if needed

% Plot gradient orientations using Gx and Gy as the vector components
[x, y] = meshgrid(1:step:size(Gx, 2), 1:step:size(Gy, 1));
quiver(x, y, Gx(1:step:end, 1:step:end), Gy(1:step:end, 1:step:end), 'color', 'r',
'LineWidth', 0.5);

title('Gradient Orientation Visualization Using Quiver');
hold off;

% Calculate number of cells in each direction
```

```
num_cells_row = floor(size(img, 1) / cell_size);
num_cells_col = floor(size(img, 2) / cell_size);

% Preallocate the HOG feature array
hog_features = zeros(1, num_cells_row * num_cells_col * num_bins);

% Calculate HOG features for each cell
counter = 1;
for row = 1:cell_size:(size(img, 1) - cell_size + 1)
    for col = 1:cell_size:(size(img, 2) - cell_size + 1)
        % Extract cell gradient magnitudes and orientations
        cell_magnitude = magnitude(row:row+cell_size-1, col:col+cell_size-1);
        cell_orientation = orientation(row:row+cell_size-1, col:col+cell_size-1);

        % Calculate histogram for the cell manually
        cell_histogram = zeros(1, num_bins);
        for bin = 1:num_bins
            % Define the orientation range for the bin
            angle_min = (bin - 1) * bin_size;
            angle_max = bin * bin_size;

            % Find pixels within the bin range
            mask = (cell_orientation >= angle_min) & (cell_orientation < angle_max);

            % Sum the magnitudes of pixels within the current orientation bin
            cell_histogram(bin) = sum(cell_magnitude(mask));
        end

        % Store cell histogram in hog_features
        hog_features(counter:counter+num_bins-1) = cell_histogram;
        counter = counter + num_bins;
    end
end

% Block normalization: Divide the HOG features into blocks of 2x2 cells and normalize
block_histograms = [];
for row = 1:num_cells_row - block_size + 1
    for col = 1:num_cells_col - block_size + 1
        % Compute the index in hog_features for the current block
        start_index = ((row - 1) * num_cells_col + col - 1) * num_bins + 1;
        end_index = start_index + block_size * block_size * num_bins - 1;

        % Extract the block of HOG features
        block = hog_features(start_index:end_index);

        % Normalize the block histogram
        norm_block = block / sqrt(sum(block .^ 2) + 1e-6);

        % Append the normalized block to block_histograms
        block_histograms = [block_histograms, norm_block];
    end
end

% Final HOG descriptor for the entire image
hog_descriptor = block_histograms;

% Display results
disp("HOG Descriptor for the entire image:");
disp(size(hog_descriptor));
disp("Length of HOG descriptor vector:");
disp(length(hog_descriptor));
```

```matlab

% Visualization of HOG Features on a Blank Canvas
figure;
imshow(ones(size(img)), []); % Blank figure with same size as image
hold on;

% Visualize dominant gradient direction per cell
for row = 1:cell_size:(size(img, 1) - cell_size + 1)
    for col = 1:cell_size:(size(img, 2) - cell_size + 1)
        % Get cell gradient data
        cell_magnitude = magnitude(row:row+cell_size-1, col:col+cell_size-1);
        cell_orientation = orientation(row:row+cell_size-1, col:col+cell_size-1);

        % Calculate histogram for the cell manually
        cell_histogram = zeros(1, num_bins);
        for bin = 1:num_bins
            % Define the orientation range for the bin
            angle_min = (bin - 1) * bin_size;
            angle_max = bin * bin_size;

            % Find pixels within the bin range
            mask = (cell_orientation >= angle_min) & (cell_orientation < angle_max);

            % Sum the magnitudes of pixels within the current orientation bin
            cell_histogram(bin) = sum(cell_magnitude(mask));
        end

        % Find the dominant orientation
        [max_value, dominant_bin] = max(cell_histogram);
        dominant_orientation = (dominant_bin - 1) * bin_size + bin_size / 2; % Center of the bin

        % Convert the orientation to radians for plotting
        angle_rad = deg2rad(dominant_orientation);

        % Plot the dominant gradient direction using quiver
        scale_factor = 0.3;
        quiver(col + cell_size / 2, row + cell_size / 2, ...
               scale_factor * max_value * cos(angle_rad), ...
               scale_factor * max_value * sin(angle_rad), ...
               'color', 'w', 'LineWidth', 1);
    end
end

title('HOG Features Visualization for Dominant Orientation per Cell');
hold off;

```