HOG Feature Extraction and Classification in MATLAB

October 16, 2025

```
% Create image datastore
imds = imageDatastore(’/MATLAB Drive/kent-mango’, ...
    ’IncludeSubfolders’, true, ...
    ’LabelSource’, ’foldernames’);
% Loads all images from the given folder and its
   subfolders.
% Labels are automatically assigned based on folder names
   .

% Split into training and testing sets
[trainingSet, testSet] = splitEachLabel(imds, 0.8, ’
   randomized’);
% Divides the dataset into 80% training and 20% testing
   with random selection.

countEachLabel(trainingSet);
% Displays the number of images in each label of the
   training set.

% Define HOG parameters
cellSize = [4 4];
% Specifies the cell size for the HOG feature extraction.

img = imread(’/MATLAB Drive/kent-mango/defect/Mango_12_A.
   JPG’);
% Reads one sample image from the dataset.

img = imresize(img, [224 224]);
% Resizes the image to 224x224 pixels to maintain uniform
    size.
```

```
img = im2gray(img); % Always convert to grayscale before HOG
% Converts the RGB image to grayscale since HOG uses intensity gradients.

[hog_4x4, vis4x4] = extractHOGFeatures(img, 'CellSize', cellSize);
% Extracts HOG features and visualization data from the grayscale image.

hogFeatureSize = length(hog_4x4);
% Stores the number of HOG features extracted from the image.

numImages = numel(trainingSet.Files);
% Counts the total number of training images.

% Preallocate feature matrix
trainingFeatures = zeros(numImages, hogFeatureSize, 'single');
% Initializes a matrix to store HOG features of all training images.

% Extract HOG features for each training image
for i = 1:numImages
    img1 = readimage(trainingSet, i);
    img1 = imresize(img1, [224 224]);
    img1 = im2gray(img1);  % Ensure grayscale conversion
    trainingFeatures(i, :) = extractHOGFeatures(img1, 'CellSize', cellSize);
end
% Loops through all training images, resizes, converts to grayscale,
% and extracts HOG features for classifier training.

% Get labels
trainingLabels = trainingSet.Labels;
% Retrieves the labels of training images.

% Train classifier (simple neural net)
classifier = fitcnet(trainingFeatures, trainingLabels);
% Trains a neural network classifier using the HOG features and labels.

% Extract HOG features for test set
```

```
numTest = numel(testSet.Files);
testFeatures = zeros(numTest, hogFeatureSize, 'single');
% Initializes a feature matrix for the test dataset.

for j = 1:numTest
    imgTest = readimage(testSet, j);
    imgTest = imresize(imgTest, [224 224]);
    imgTest = im2gray(imgTest);
    testFeatures(j, :) = extractHOGFeatures(imgTest, '
       CellSize', cellSize);
end
% Extracts HOG features for all test images in the
   dataset.

testLabels = testSet.Labels;
% Retrieves actual labels for test images.

% Predict
predictedLabels = predict(classifier, testFeatures);
% Uses the trained model to predict labels for test
   images.

% Confusion matrix
confMat = confusionmat(testLabels, predictedLabels);
confusionchart(confMat);
% Generates and displays a confusion chart to visualize
   classifier accuracy.
```

Listing 1: MATLAB code with inline explanations