**LAB - 9**

**CBIR**

**HOG FEATURE EXTRACTION**

**PRIYANSHU SHARMA**

**15BCE1282**

**QUESTION - 1**

**EXTRACT HOG FEATURE FOR THE DIGIT DATASET**

**CODE -**

srcFiles = dir('C:\Users\PRIYANSHU SHARMA\Desktop\PRIYANSHU\6 STUDY\CBIR\MATLAB\LAB 9\\*.png');

for i=1:10

%k = i-1;

img = imread(strcat('C:\Users\PRIYANSHU SHARMA\Desktop\PRIYANSHU\6 STUDY\CBIR\MATLAB\LAB 9\',srcFiles(i).name));

[hog\_2x2, vis2x2] = extractHOGFeatures(img,'CellSize',[2 2]);

[hog\_4x4, vis4x4] = extractHOGFeatures(img,'CellSize',[4 4]);

[hog\_8x8, vis8x8] = extractHOGFeatures(img,'CellSize',[8 8]);

% Show the original image

figure;

subplot(2,3,1:3); imshow(img);

% Visualize the HOG features

subplot(2,3,4);

plot(vis2x2);

title({'CellSize = [2 2]'; ['Length = ' num2str(length(hog\_2x2))]});

subplot(2,3,5);

plot(vis4x4);

title({'CellSize = [4 4]'; ['Length = ' num2str(length(hog\_4x4))]});

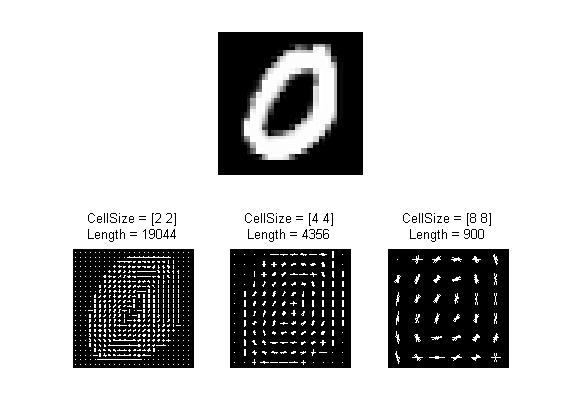
subplot(2,3,6);

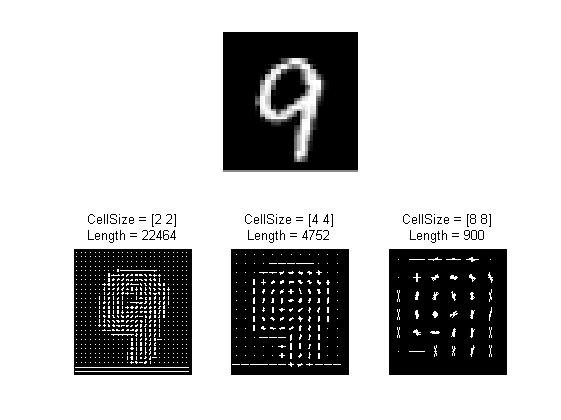
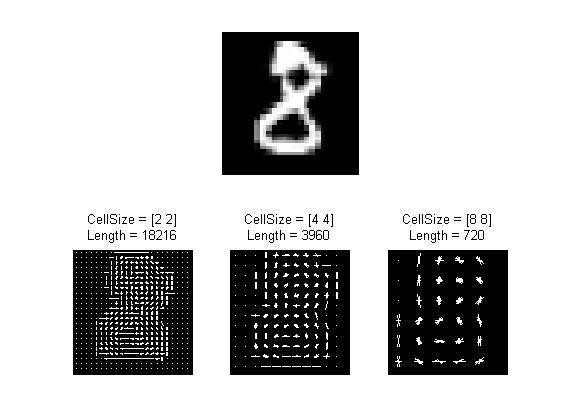
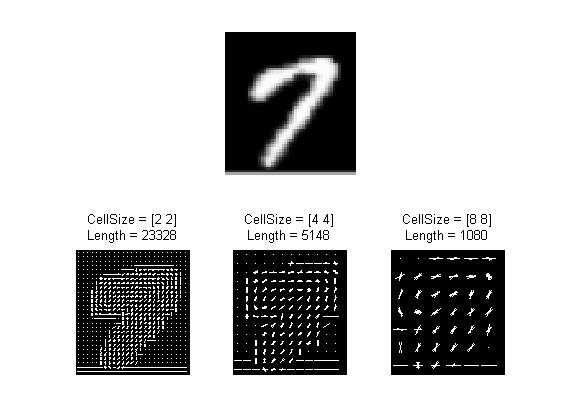
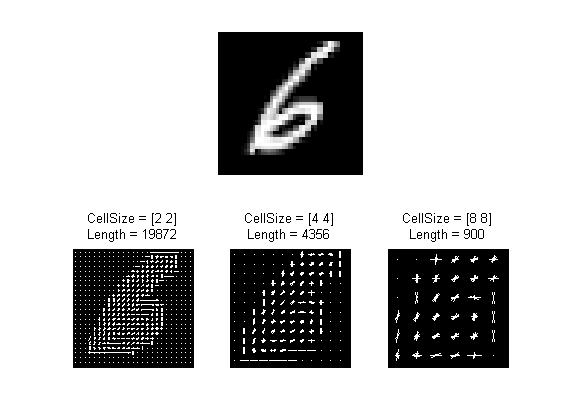
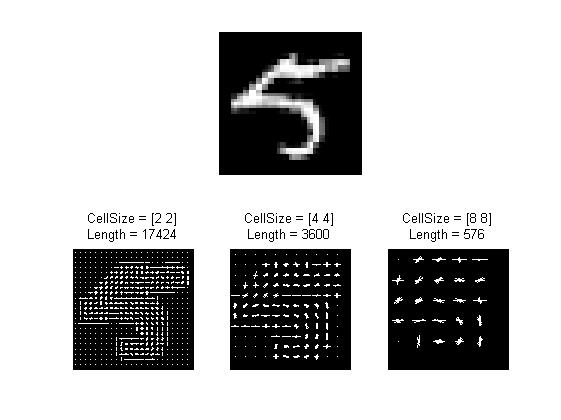
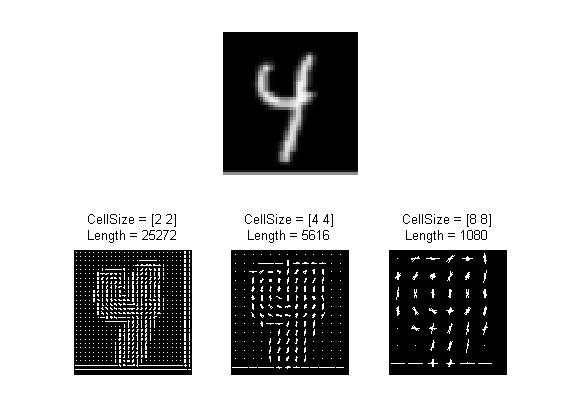
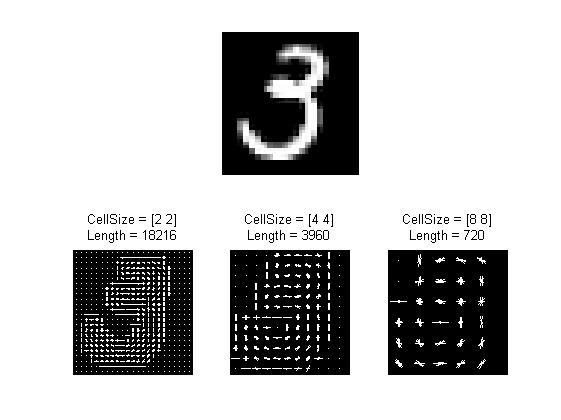
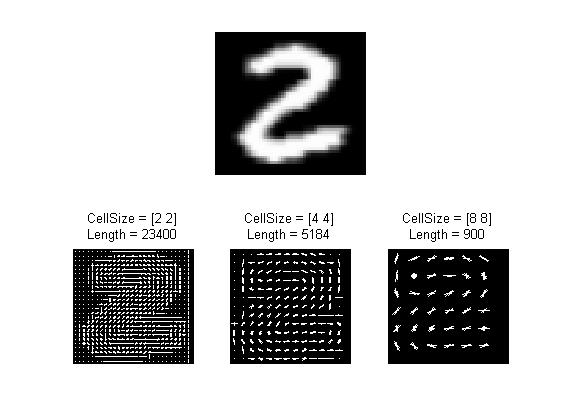
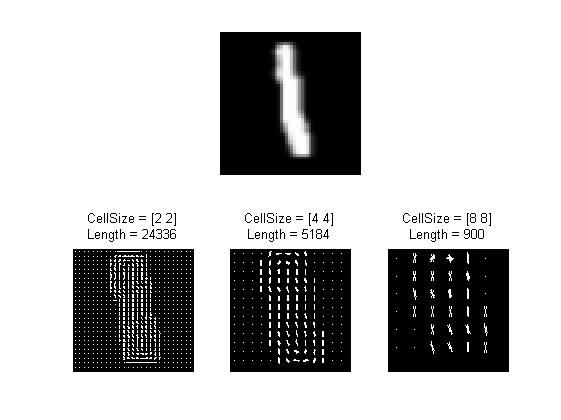
plot(vis8x8);

title({'CellSize = [8 8]'; ['Length = ' num2str(length(hog\_8x8))]});

end

**OUTPUT**





**HUMAN DETECTION AND CLASSIFICATION USING HOG FEATURES**

**CODES**

**runExampleSearch.m**

addpath('./common/');

addpath('./graphics/');

addpath('./search/');

load('hog\_model.mat');

hog.threshold = 0.4;

% Read in the image to be searched.

img = imread('C:\Users\TEMP\Documents\MATLAB\hog\_matlab-master\Images\Validation\IMG\_9418.jpg');

tic();

% Search the image for persons.

resultRects = searchImage(hog, img);

elapsed = toc();

fprintf('Image search took %.2f seconds\n', elapsed);

%%

% Validate the search results.

% Load the annotations file.

goodRects = load('C:/Users/TEMP/Documents/MATLAB/hog\_matlab-master/Images/Validation/IMG\_9418\_annotations.csv');

requiredIndeces = (goodRects(:, 5) == 1);

optionalIndeces = (goodRects(:, 5) == 0);

goodRects = [goodRects(requiredIndeces, :); goodRects(optionalIndeces, :)];

numVisiblePeople = sum(requiredIndeces);

rectsFound = zeros(numVisiblePeople, 1);

numFalsePositives = 0;

resultRects = [resultRects, zeros(size(resultRects, 1), 1)];

for k = 1 : size(resultRects, 1)

indeces = checkRectOverlap(resultRects(k, :), goodRects, 0.5);

if (isempty(indeces))

% Indicate it's a bad result.

resultRects(k, end) = 0;

% Increment the number of false positives.

numFalsePositives = numFalsePositives + 1;

% If we found one or more matches...

else

% For each of the matches...

for i = 1 : length(indeces)

% If we found a 'required' match...

if (goodRects(indeces(i), 5) == 1)

% Indicate it's a good result.

resultRects(k, end) = 1;

% Indicate we found this person.

rectsFound(indeces(i)) = 1;

else

% Indicate it's an optional result.

resultRects(k, end) = -1;

end

end

end

end

% The number of unique visible people that we found.

totalVisibleFound = sum(rectsFound);

% Print the results.

fprintf('Found %d / %d people (%.2f%%), with %d false positives.\n', ...

totalVisibleFound, numVisiblePeople, ...

totalVisibleFound / numVisiblePeople \* 100.0, ...

numFalsePositives);

hold off;

imagesc(img);

hold on;

% Draw each of the detection hits.

for i = 1 : size(resultRects, 1)

rect = resultRects(i, :);

if rect(end) == 0

continue;

end

color = 'b';

% If the match is a good one (or an optional one), color it red.

if (rect(end) ~= 0)

color = 'r';

end

% Draw the results.

drawRectangle(resultRects(i, :), color);

end

**GetHOGDescriptor.m**

function H = getHOGDescriptor(hog, img)

H = [];

assert(isequal(size(img), hog.winSize))

hx = [-1,0,1];

hy = hx';

dx = filter2(hx, double(img));

dy = filter2(hy, double(img));

dx = dx(2 : (size(dx, 1) - 1), 2 : (size(dx, 2) - 1));

dy = dy(2 : (size(dy, 1) - 1), 2 : (size(dy, 2) - 1));

angles = atan2(dy, dx);

magnit = ((dy.^2) + (dx.^2)).^.5;

histograms = zeros(hog.numVertCells, hog.numHorizCells, hog.numBins);

for row = 0:(hog.numVertCells - 1)

rowOffset = (row \* hog.cellSize) + 1;

for col = 0:(hog.numHorizCells - 1)

colOffset = (col \* hog.cellSize) + 1;

rowIndeces = rowOffset : (rowOffset + hog.cellSize - 1);

colIndeces = colOffset : (colOffset + hog.cellSize - 1);

cellAngles = angles(rowIndeces, colIndeces);

cellMagnitudes = magnit(rowIndeces, colIndeces);

histograms(row + 1, col + 1, :) = getHistogram(cellMagnitudes(:), cellAngles(:), hog.numBins);

end

end

for row = 1:(hog.numVertCells - 1)

for col = 1:(hog.numHorizCells - 1)

blockHists = histograms(row : row + 1, col : col + 1, :);

magnitude = norm(blockHists(:)) + 0.01;

normalized = blockHists / magnitude;

H = [H; normalized(:)];

end

end

end

**OUTPUT**

Image search took 85.38 seconds

Found 6 / 6 people (100.00%), with 5909 false positives.

