

```

%=====
%=====

%=====
%VECTORIZED IMAGE CLASSIFICATION - MNIST FASHION STAT CLUSTER PREDICTION
%=====

%=====
%=====

%COPYRIGHT CHARLES DAVI, 2021

%=====
%LOADS THE DATASET
%=====

clear
clc
pkg load image

directory = '/Users/charlesdavi/Desktop/Datasets/fashion/img/';

num_images = 5000

for i = 0 : num_images - 1

    I = imread([directory 'fashion' int2str(i) '.png']);

    IMG_array{i+1} = I;

endfor

%loads classifiers-----
file_name = '/Users/charlesdavi/Desktop/Datasets/fashion/class_full_dataset.txt';

A = csvread(file_name);

IMG_category = A(1:num_images);

%=====
%EXTRACTS SHAPE INFORMATION
%=====

tic;
[final_avg_matrix final_indexes] = partition_image_vectorized_gs(IMG_array{1}); %this is to size the partitions for the
entire dataset
toc

N = size(final_avg_matrix,1);

tic;
%iterates through entire dataset
for i = 1 : num_images

    I = IMG_array{i};

    [avg_matrix] = calc_avg_color_vect(final_indexes, I, N); %this extracts shape information

    input_vector = reshape(avg_matrix, [1 N^2]);

```

```

input_vector(N^2+1) = IMG_category(i); %this is the hidden classifier

dataset(i,:) = input_vector;

endfor
toc

%=====
%GENERATES CLUSTERS
%=====
tic;
N = N^2;
s = std(dataset(:,1:N)); %calculates the standard deviation of the dataset in each dimension
s = mean(s); %takes the average standard deviation
s = s*N;

num_rows = size(dataset,1);

cluster_matrix = zeros(num_rows,num_rows);

delta = s/24; %this is the value of delta based upon experimentation

for i = 1 : num_rows

    input_vector = dataset(i,:);
    [cluster_vector diff_vector] = find_delta_cluster(input_vector, dataset, delta, N);
    cluster_matrix(i,:) = cluster_vector;

endfor
toc

%=====
%GENERATES TRAINING / TESTING DATASET
%=====

num_iterations = 150;

accuracy_p = [];
accuracy_c = [];
accuracy_cp = [];

num_rows = size(dataset,1);

for i = 1 : num_iterations

    %permutes the dataset

    %Generates a training and testing dataset-----
    num_training_rows = floor(.85*num_rows); %selects a portion of the dataset
    training_rows = randperm(num_rows,num_training_rows);
    testing_dataset = dataset;
    testing_dataset(training_rows,:) = [];
    training_dataset = dataset(training_rows,:);
    num_testing_rows = size(testing_dataset,1);

    %=====
    %CLUSTER PREDICTION STEP
    %=====
    [predicted_class confidence_vector probability_vector CP_accuracy error_vector] = cluster_prediction(dataset,
    training_dataset, testing_dataset, training_rows, cluster_matrix, N);

```

```

CP_accuracy_vector(i) = CP_accuracy;

%=====
%PROBABILITY; CONFIDENCE
%=====

%probability-----
counter = 1;
increment = .01;
num_levels = size(0 : increment : 1,2);

for j = 0 : increment : 1

    x = find(probability_vector >= j);

    num_errors = sum(error_vector(x));

    num_predictions = size(x,2);

    if(num_predictions > 0)

        accuracy_p(i,counter) = 1 - num_errors/num_predictions;

    endif

    counter = counter + 1;

endfor

%confidence-----
counter = 1;
increment = .01;
num_levels = size(0 : increment : 1,2);

confidence_vector = confidence_vector/max(confidence_vector);

for j = 0 : increment : 1

    x = find(confidence_vector >= j);

    num_errors = sum(error_vector(x));

    num_predictions = size(x,2);

    if(num_predictions > 0)

        accuracy_c(i,counter) = 1 - num_errors/num_predictions;

    endif

    counter = counter + 1;

endfor

%confidence and probability-----
counter = 1;
increment = .001;
num_levels = size(0 : increment : 1,2);

```

```

confidence_vector = confidence_vector/max(confidence_vector);

for j = 0 : increment : 1

    x = find(probability_vector >= j);
    y = find(confidence_vector >= j);

    temp1 = zeros(num_testing_rows,1);
    temp2 = zeros(num_testing_rows,1);

    temp1(x) = 1;
    temp2(y) = 1;

    z = temp1.*temp2;

    z = find(z == 1);

    z = z';

    num_errors = sum(error_vector(z));

    num_predictions = size(z,2);

    if(num_predictions > 0)

        accuracy_cp(i,counter) = 1 - num_errors/num_predictions;

    endif

    counter = counter + 1;

endfor

endfor %end of outer loop

plot_data_p = mean(accuracy_p);
plot_data_c = mean(accuracy_c);
plot_data_cp = mean(accuracy_cp);

figure, plot(plot_data_p)
figure, plot(plot_data_c)
figure, plot(plot_data_cp)

```