

```

%=====
=====
%=====
=====
%VIDEO CLASSIFICATION ALGORITHM
%=====
=====
%=====
=====
%COPYRIGHT CHARLES DAVI 2021

%=====
=====
%LOADS DATASET
%=====
=====
clear
clc

pkg load image

directory = "/Users/charlesdavi/Desktop/Datasets/Gesture_Dataset/Training_Set/"

num_observations = 24;
num_frames = 10;
counter = 1;

%loads first image to size dataset-----
i = 1;
j = 1;
file_name = [directory int2str(i) "/" int2str(j) ".jpg"];
I = imread(file_name);
I = rgb2gray(I);
N = size(I(:), 1);
num_rows = size(I,1);
num_cols = size(I,2);

%loads full dataset-----
tic;
for i = 1 : num_observations

    for j = 1 : num_frames

        file_name = [directory int2str(i) "/" int2str(j) ".jpg"];

        I = imread(file_name);
        I = rgb2gray(I);

```

```

    img_matrix(:, :, j) = I;

endfor

avg_img = mean(img_matrix, 3); %time-average of the iamge sequence

dataset(counter, 1 : N) = avg_img(:);
dataset(counter, N + 1) = floor(i/13); %hidden classifier, half are class 0;
counter = counter + 1;

endfor
toc

avg_img = mat2gray(mean(img_matrix, 3));

figure, imshow(avg_img); %displays an average over the sequence of images

%=====
%PARTITIONS EACH FRAME OF THE DATASET
%=====

%loads the first frame of the dataset for sizing-----
I = dataset(1, 1 : N);
I = reshape(I, [num_rows num_cols]); %reshapes back into image
[final_avg_matrix final_indexes] = partition_image_vectorized_gs(I);
N_grid = size(final_avg_matrix, 1); %this the number of columns and rows in the partition

tic;
for i = 1 : num_observations

    I = dataset(i, 1 : N);
    I = reshape(I, [num_rows num_cols]); %reshapes back into image
    [avg_matrix final_indexes] = calc_avg_colors_fixed_N(I, N_grid);
    space_compressed_dataset(i, :) = avg_matrix(:);

endfor
toc

%=====
%CLASSIFIES THE DATATSET
%=====

final_N = size(space_compressed_dataset, 2);

```

```
tic;
nearest_neighbors = NN_fully_vectorized(space_compressed_dataset, final_N);
toc

predicted_class_vector = dataset(nearest_neighbors, N + 1);
actual_class_vector = dataset(:,N + 1);

num_errors = sum(predicted_class_vector != actual_class_vector)

accuracy = 1 - num_errors/num_rows
```