Lab

clear all;

clc;

% Load training and test data using |imageDatastore|.

syntheticDir =fullfile(toolboxdir('vision'),'visiondata','digits','synthetic');

handwrittenDir = fullfile(toolboxdir('vision'),'visiondata','digits','handwritten');

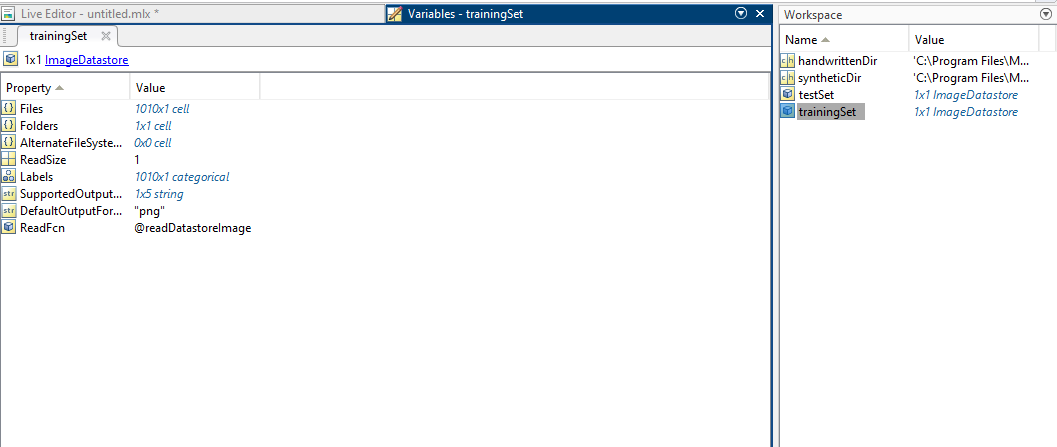
% |imageDatastore| recursively scans the directory tree containing the

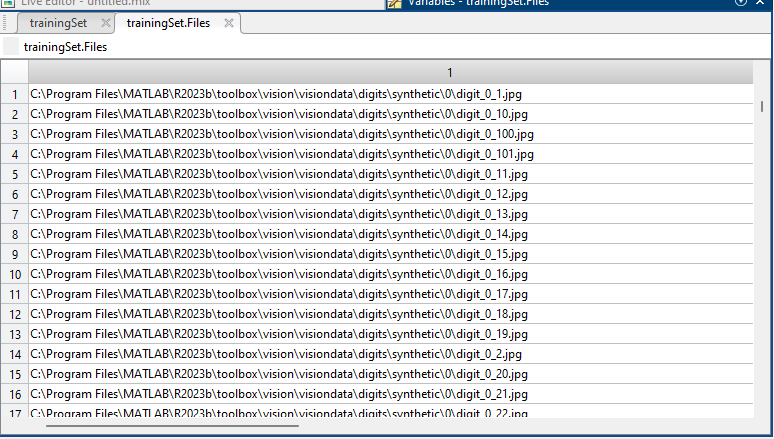
% images. Folder names are automatically used as labels for each image.

trainingSet = imageDatastore(syntheticDir,'IncludeSubfolders',true,'LabelSource','foldernames');

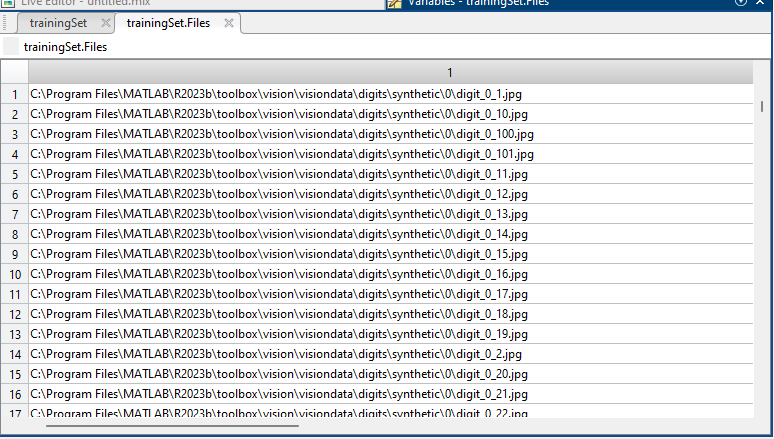
testSet = imageDatastore(handwrittenDir,'IncludeSubfolders',true,'LabelSource','foldernames');

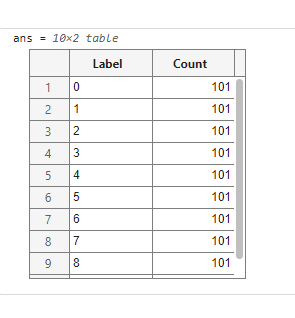
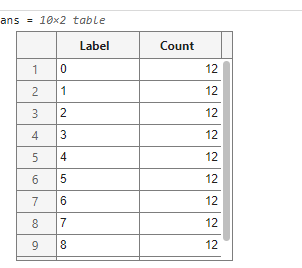
image data store- was used for big data files and images in size of tb,gb.

  
 by clicking the training set we see all this .



In that training set,go to training set files .you will able to see this much.



countEachLabel(trainingSet)  
  
  
countEachLabel(testSet)  
  
  
it is having same count for every sample in both training and test dataset which is indicating that it is a well defined dataset.  
  
  
figure;#I need to start with ‘figure’ otherwise the image may be overwrite

subplot(2,3,1);#(SUBPLOT IS USE FOR MULTIPLE IMAGES)(2 rows,3 columnsand in that first one)

imshow(trainingSet.Files{102});#102 is the s.no of the training set and it will display the particular file.

subplot(2,3,2);

imshow(trainingSet.Files{304});

subplot(2,3,3);

imshow(trainingSet.Files{809});

subplot(2,3,4);

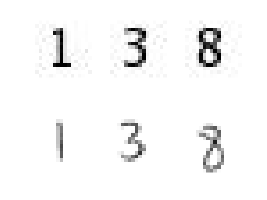
imshow(testSet.Files{13});

subplot(2,3,5);

imshow(testSet.Files{37});

subplot(2,3,6);

imshow(testSet.Files{97});



% Loop over the trainingSet and extract pixel values as features from each image. A

% similar procedure will be used to extract features from the testSet.

numTrainImages = numel(trainingSet.Files);#number of values or length of the values (here it is 1010)

temp = readimage(trainingSet,3); % randomly reading an image in the training set (here it is the 1st image in the training set, hence 1)

imgSize=size(temp);

featureLength=prod(imgSize);#product of 16\*16\*3

trainingFeatures = zeros(numTrainImages,featureLength);#starting from zeros array

for i = 1:numTrainImages

img = readimage(trainingSet,i);

pixelValuesSingleRowTrain=reshape(img,1,featureLength);

trainingFeatures(i, :) = pixelValuesSingleRowTrain;

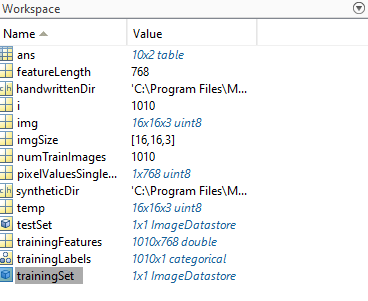
end

% Get labels for each image.

trainingLabels = trainingSet.Labels;



image value is multiplied 16\*16\*3 so you get 768

  
by this I can view training set and labels.

kNNclassifier = fitcknn(trainingFeatures, trainingLabels,'NumNeighbors',7)