% Load your dataset and organize it into training and testing sets

% For this example, let's assume you have a dataset in a folder structure like:

% - dataset

% - class1

% - image1.jpg

% - image2.jpg

% ...

% - class2

% - image1.jpg

% - image2.jpg

% ...

% ...

dataDir = 'directory';

categories = {'\_noise\_baseline', 'noise\_waves'}; % List of class names

imds = imageDatastore(fullfile(dataDir, categories), 'LabelSource','foldernames');

% Split the data into training and testing sets

[trainData, testData] = splitEachLabel(imds, 0.7, 'randomize');

% Create a CNN architecture

layers = [

imageInputLayer([128 128 3])

convolution2dLayer(3, 128, 'Padding', 'same')

reluLayer()

maxPooling2dLayer(2, 'Stride', 2)

convolution2dLayer(3, 64, 'Padding', 'same')

reluLayer()

maxPooling2dLayer(2, 'Stride', 2)

fullyConnectedLayer(length(categories))

softmaxLayer()

classificationLayer()

];

% Set training options

options = trainingOptions('adam', ...

'MiniBatchSize', 32, ...

'MaxEpochs', 10, ...

'Shuffle', 'every-epoch', ...

'ValidationData', testData, ...

'ValidationFrequency', 10, ...

'Plots', 'training-progress', ...

'Verbose', true);

% Train the CNN

net = trainNetwork(trainData, layers, options);

% Classify the test data

YPred = classify(net, testData);

YValidation = testData.Labels;

accuracy = mean(YPred == YValidation);