% 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

% ...

% ...

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

dataDir = 'directory';

categoriesTrain = {'synthetic and raw baseline', 'synthetic and raw waves'}; % List of class names for training data

categoriesTest = {'synthetic and raw baseline', 'synthetic and raw waves'}; % List of class names for testing data

% Training Data

imdsTrain = imageDatastore(fullfile(dataDir, categoriesTrain), 'LabelSource', 'foldernames');

% Split the training data into training and validation sets

[trainData, validationData] = splitEachLabel(imdsTrain, 0.7, 'randomize');

% Testing Data

imdsTest = imageDatastore(fullfile(dataDir, categoriesTest), 'LabelSource', 'foldernames');

% Filter out images from specified classes in the testing data

classesToExclude = {'SyntheticBaselineImages','SyntheticWavesImages'};

testDataWithoutExcludedClasses = subset(imdsTest, ~ismember(imdsTest.Labels, classesToExclude));

% 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(categoriesTrain))

softmaxLayer()

classificationLayer()

];

% Set training options

options = trainingOptions('adam', ...

'MiniBatchSize', 32, ...

'MaxEpochs', 10, ...

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

'ValidationData', validationData, ... % Use validation data during training

'ValidationFrequency', 10, ...

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

'Verbose', true);

% Train the CNN

net = trainNetwork(trainData, layers, options);

% Classify the filtered test data

YPred = classify(net, testDataWithoutExcludedClasses);

YValidation = testDataWithoutExcludedClasses.Labels;

accuracy = mean(YPred == YValidation);