

```

clc; clear;
dsFolder = "p_dataset_26";
subFolder = ["0", "4", "7", "8", "A", "D", "H"];
categories = ["0", "4", "7", "8", "A", "D", "H"];

```

```

allData = table('Size', [0, 2], 'VariableTypes', {'cell', 'cell'}, 'VariableNames',
{'Image', 'Label'});

for i = 1:length(subFolder)
    folderPath = fullfile(dsFolder, subFolder(i));
    pngFiles = dir(fullfile(folderPath, "*.png"));

    for j = 1:length(pngFiles)
        pngFilePath = fullfile(folderPath, pngFiles(j).name);
        image = imread(pngFilePath);
        allData = [allData; {image, categories(i)}];
    end
end

```

```

% define images & labels
numImages = size(allData, 1);
imageSize = size(allData.Image{1});
images = zeros(imageSize(1), imageSize(2), 1, numImages);

for i = 1:numImages
    images(:, :, 1, i) = allData.Image{i};
end

labels = categorical(allData.Label);

```

```

% split data into train and test set
cv = cvpartition(labels, 'HoldOut', 0.2);
trainIdx = training(cv);
testIdx = test(cv);
trainImages = images(:, :, :, trainIdx);
trainLabels = labels(trainIdx);
testImages = images(:, :, :, testIdx);
testLabels = labels(testIdx);

```

```

% Define the layers of the CNN
layers = [
    imageInputLayer([imageSize 1])

    convolution2dLayer(3, 8, 'Padding', 'same')
    batchNormalizationLayer
    reluLayer

```

```

maxPooling2dLayer(2, 'Stride', 2)

convolution2dLayer(3, 16, 'Padding', 'same')
batchNormalizationLayer
reluLayer

maxPooling2dLayer(2, 'Stride', 2)

fullyConnectedLayer(numel(categories))
softmaxLayer
classificationLayer
];

```

% Training options

```

options = trainingOptions( ...
    'sgdm', ... % Stochastic Gradient Descent with Momentum
    'InitialLearnRate', 0.01, ...
    'MaxEpochs', 10, ...
    'Shuffle', 'every-epoch', ...
    'ValidationData', {testImages, testLabels}, ...
    'ValidationFrequency', 30, ...
    'Verbose', true, ...
    'Plots', 'training-progress' ...
);

```

% Train the CNN

```

net = trainNetwork(trainImages, trainLabels, layers, options);

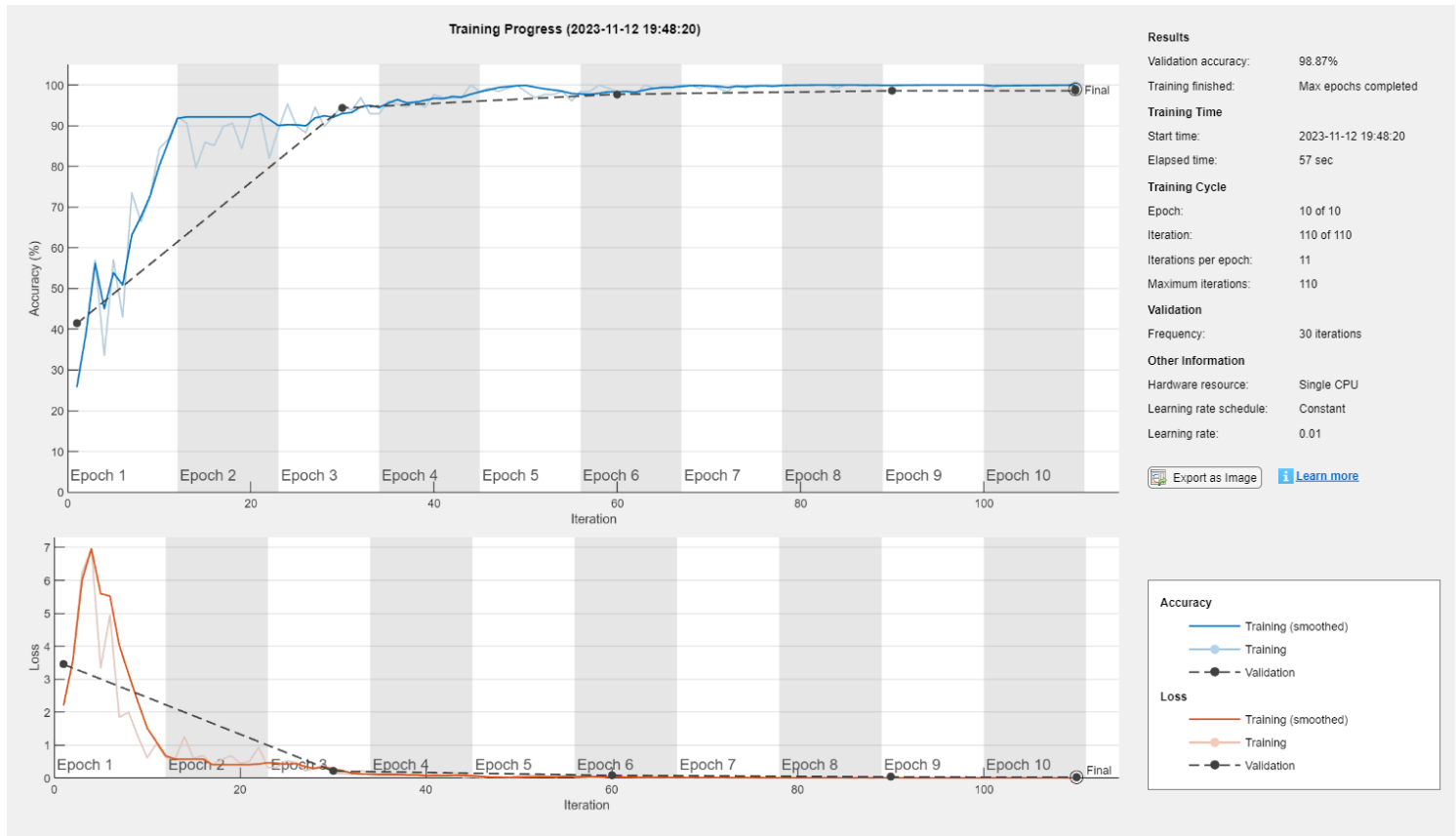
```

Training on single CPU.

Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:03	25.78%	41.41%	2.2098	3.4553	0.01
3	30	00:00:18	95.31%	94.37%	0.1470	0.2141	0.01
5	50	00:00:27	98.44%		0.0765		0.01
6	60	00:00:32	98.44%	97.75%	0.0568	0.0850	0.01
9	90	00:00:47	100.00%	98.59%	0.0095	0.0389	0.01
10	100	00:00:52	100.00%		0.0055		0.01
10	110	00:00:57	100.00%	98.59%	0.0072	0.0303	0.01

Training finished: Max epochs completed.



```
% Classify Test Images
```

```
predictedLabels = classify(net, testImages);
```

```
% Calculate the Accuracy
```

```
accuracy = sum(predictedLabels == testLabels) / numel(testLabels);
```

```
fprintf('Test Accuracy: %.2f%%\n', accuracy * 100);
```

```
Test Accuracy: 98.87%
```