```
clc; clear;
dsFolder = "p_dataset_26";
subFolder = ["Sample1", "Sample2", "Sample3", "SampleA", "SampleB", "SampleC"];
categories = ["1", "2", "3", "A", "B", "C"];
```

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

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

    for j = 1:length(matFiles)
        matFilePath = fullfile(folderPath, matFiles(j).name);
        image = load(matFilePath).imageArray;

        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
```

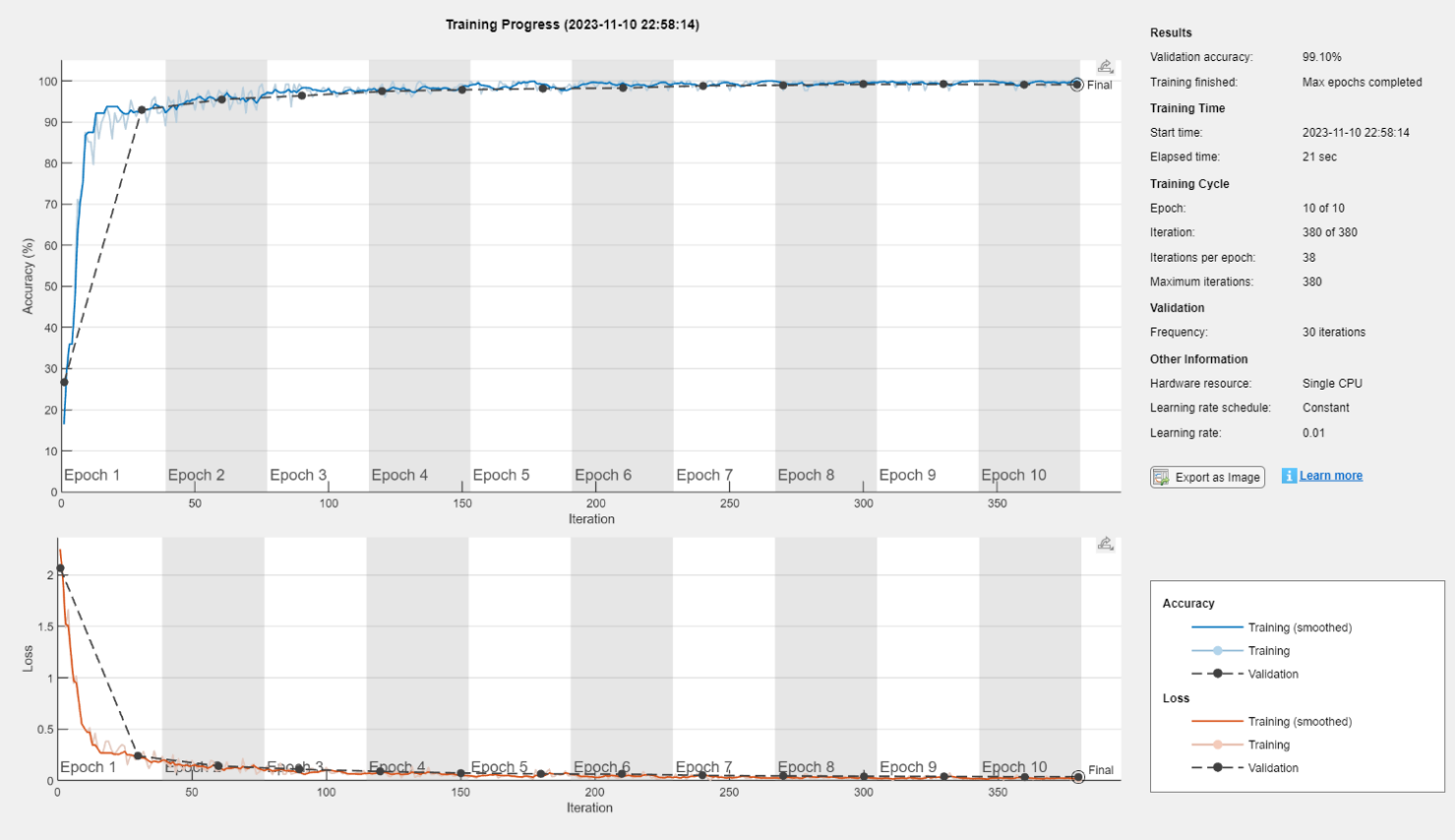
```
    maxPooling2dLayer(2, 'Stride', 2)

    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  |  Mini-batch  |  Validation  |  Mini-batch  |  Validation  |  Base Learnin
|         |             |   (hh:mm:ss)   |   Accuracy   |   Accuracy   |     Loss     |     Loss     |      Rate
|=========================================================================================================================
|       1 |           1 |       00:00:04 |       16.41% |       26.66% |       2.2535 |       2.0679 |          0.0
|       1 |          30 |       00:00:06 |       92.97% |       92.95% |       0.2532 |       0.2397 |          0.0
|       2 |          50 |       00:00:07 |       92.97% |              |       0.1932 |              |          0.0
|       2 |          60 |       00:00:07 |       96.88% |       95.49% |       0.0968 |       0.1421 |          0.0
|       3 |          90 |       00:00:09 |       98.44% |       96.39% |       0.0848 |       0.1103 |          0.0
|       3 |         100 |       00:00:09 |       98.44% |              |       0.1279 |              |          0.0
|       4 |         120 |       00:00:10 |       99.22% |       97.46% |       0.0568 |       0.0892 |          0.0
|       4 |         150 |       00:00:11 |       99.22% |       97.87% |       0.0510 |       0.0737 |          0.0
|       5 |         180 |       00:00:13 |       98.44% |       98.11% |       0.0462 |       0.0666 |          0.0
|       6 |         200 |       00:00:13 |       98.44% |              |       0.0348 |              |          0.0
|       6 |         210 |       00:00:14 |      100.00% |       98.28% |       0.0274 |       0.0633 |          0.0
|       7 |         240 |       00:00:15 |       98.44% |       98.85% |       0.0515 |       0.0507 |          0.0
|       7 |         250 |       00:00:15 |       97.66% |              |       0.0622 |              |          0.0
|       8 |         270 |       00:00:16 |       97.66% |       98.93% |       0.0630 |       0.0468 |          0.0
|       8 |         300 |       00:00:18 |       99.22% |       99.18% |       0.0225 |       0.0424 |          0.0
|       9 |         330 |       00:00:19 |       99.22% |       99.18% |       0.0296 |       0.0396 |          0.0
|      10 |         350 |       00:00:20 |       99.22% |              |       0.0244 |              |          0.0
|      10 |         360 |       00:00:20 |      100.00% |       99.10% |       0.0203 |       0.0376 |          0.0
|      10 |         380 |       00:00:21 |      100.00% |       99.10% |       0.0192 |       0.0365 |          0.0
```

```
|=========================================================================================================================================
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: 99.10%
```