

```

clear all
clc
close all
cd C:\Users\sangw\Desktop\Xin_Final

control_sPTD = readtable('control_T2_sPTD.csv');
control_PPRM = readtable('control_vs PPRM_T2_top100(1).csv');

sPTD = readtable('sPTD_T2_top00(1).csv');
PPRM = readtable('PPRM_T2_top100.csv');

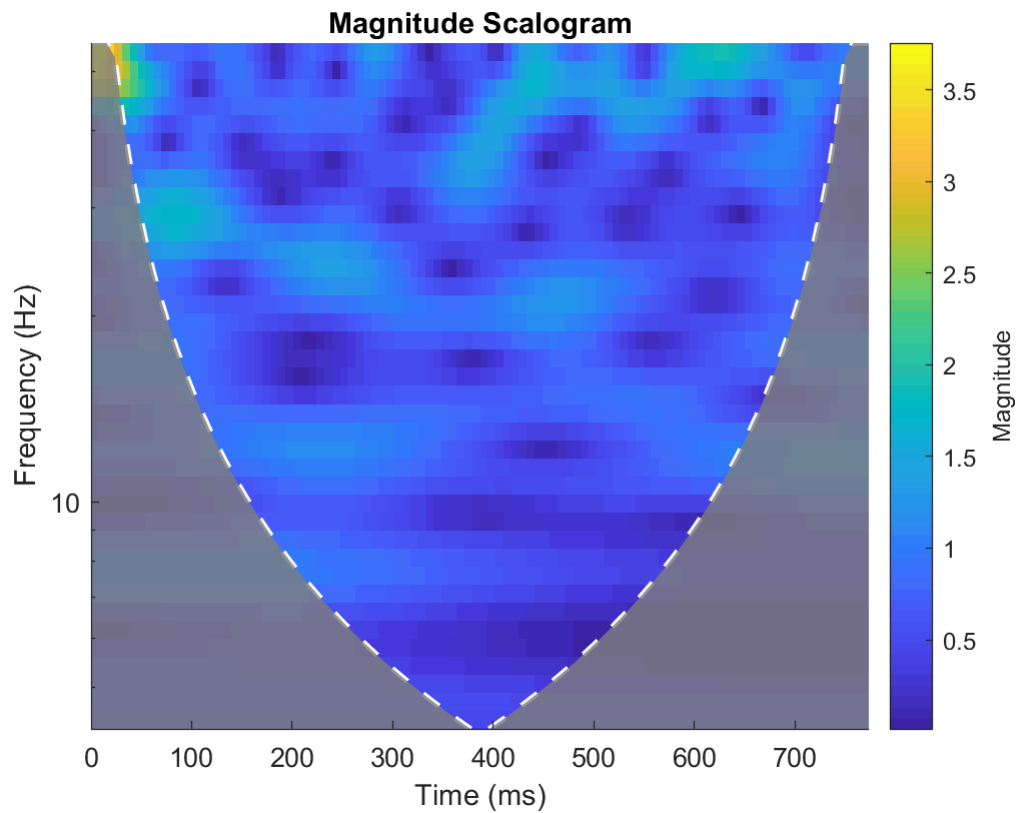
```

```

[m,n] = size(sPTD);
Fs=128;

for c=2:n
    cd C:\Users\sangw\Desktop\Xin_Final\myImages\sPTD
    sPTD_individual=sPTD(:,c);
    sPTD_individual_array=table2array(sPTD_individual);
    sPTD_individual_array=sPTD_individual_array.';
    cwt(sPTD_individual_array,Fs);
    %plot(sPTD_individual_array)
    saveas(gcf,sprintf('%d.png',c));
    cd C:\Users\sangw\Desktop\Xin_Final
end

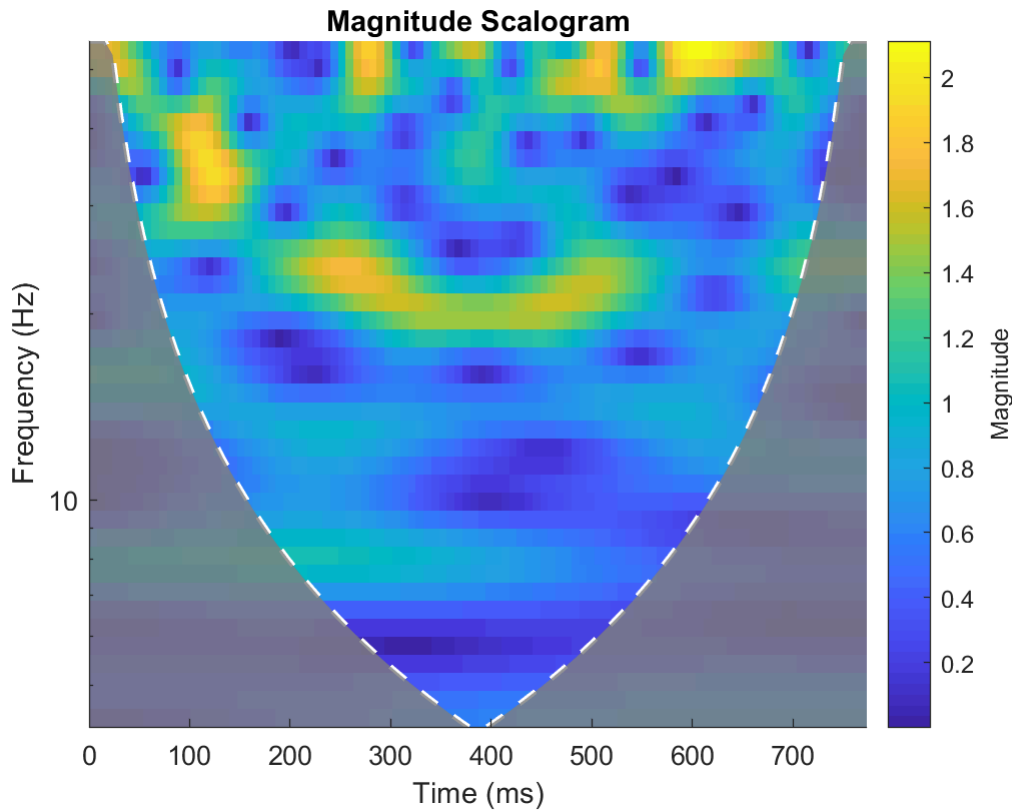
```



```

for c=2:n
    cd C:\Users\sangw\Desktop\Xin_Final\myImages\control_sPTD
    sPTD_individual=control_sPTD(:,c);
    sPTD_individual_array=table2array(sPTD_individual);
    sPTD_individual_array=sPTD_individual_array.';
    cwt(sPTD_individual_array,Fs);
    %plot(sPTD_individual_array)
    saveas(gcf,sprintf('%d.png',c));
    cd C:\Users\sangw\Desktop\Xin_Final
end

```



```
alex = alexnet;
layers = alex.Layers
```

```
layers =
    25x1 Layer array with layers:
```

1	'data'	Image Input	227x227x3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Grouped Convolution	2 groups of 128 5x5x48 convolutions with stride [1 1] and padding [0 0 0 0]
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	'pool2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv3'	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	'relu3'	ReLU	ReLU
12	'conv4'	Grouped Convolution	2 groups of 192 3x3x192 convolutions with stride [1 1] and padding [0 0 0 0]
13	'relu4'	ReLU	ReLU
14	'conv5'	Grouped Convolution	2 groups of 128 3x3x192 convolutions with stride [1 1] and padding [0 0 0 0]
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout	50% dropout
20	'fc7'	Fully Connected	4096 fully connected layer
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout	50% dropout
23	'fc8'	Fully Connected	1000 fully connected layer
24	'prob'	Softmax	softmax
25	'output'	Classification Output	crossentropyex with 'tench' and 999 other classes

```
layers(23) = fullyConnectedLayer(2);
layers(25) = classificationLayer
```

```
layers =
```

```
25x1 Layer array with layers:
```

1	'data'	Image Input	227x227x3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Grouped Convolution	2 groups of 128 5x5x48 convolutions with stride [1 1] and padding
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	'pool2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv3'	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	'relu3'	ReLU	ReLU
12	'conv4'	Grouped Convolution	2 groups of 192 3x3x192 convolutions with stride [1 1] and padding
13	'relu4'	ReLU	ReLU
14	'conv5'	Grouped Convolution	2 groups of 128 3x3x192 convolutions with stride [1 1] and padding
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout	50% dropout
20	'fc7'	Fully Connected	4096 fully connected layer
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout	50% dropout
23	''	Fully Connected	2 fully connected layer
24	'prob'	Softmax	softmax
25	''	Classification Output	crossentropyex

```
allImages = imageDatastore('myImages', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
```

```
[trainingImages, testImages] = splitEachLabel(allImages, 0.8, 'randomize');
testImages2=testImages;
```

```
inputSize = alex.Layers(1).InputSize
```

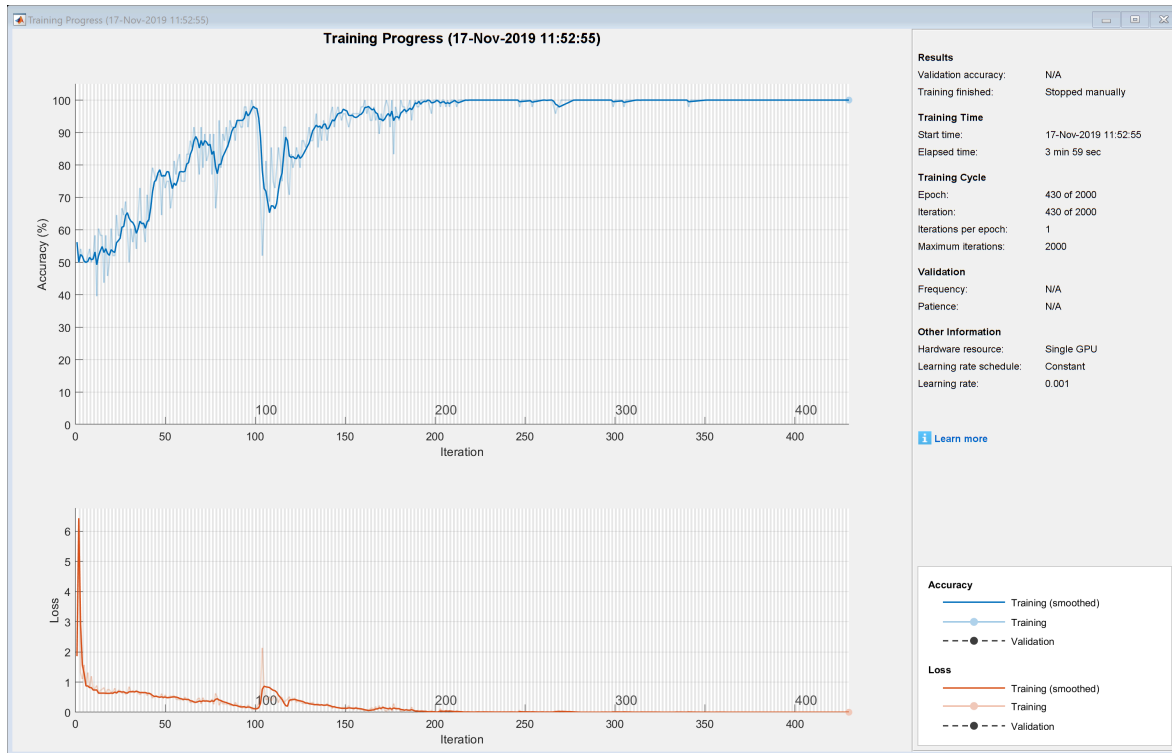
```
inputSize = 1x3
227 227 3
```

```
pixelRange = [-30 30];
imageAugmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandXTranslation',pixelRange, ...
    'RandYTranslation',pixelRange);
```

```
trainingImages = augmentedImageDatastore(inputSize(1:2),trainingImages, ...
    'DataAugmentation',imageAugmenter);
```

```
testImages = augmentedImageDatastore(inputSize(1:2),testImages);
```

```
opts = trainingOptions('sgdm', 'InitialLearnRate', 0.001, 'MaxEpochs', 2000, 'MiniBatchSize', 64);
myNet = trainNetwork(trainingImages, layers, opts);
```



Training on single GPU.

Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:01	56.25%	1.8599	0.0010
50	50	00:00:29	72.92%	0.5724	0.0010
100	100	00:00:56	95.83%	0.1028	0.0010
150	150	00:01:24	95.83%	0.1509	0.0010
200	200	00:01:51	100.00%	0.0257	0.0010
250	250	00:02:19	100.00%	0.0028	0.0010
300	300	00:02:46	100.00%	0.0005	0.0010
350	350	00:03:14	100.00%	0.0023	0.0010
400	400	00:03:42	100.00%	0.0003	0.0010
430	430	00:03:59	100.00%	0.0001	0.0010

```
predictedLabels = classify(myNet, testImages);
accuracy = mean(predictedLabels == testImages2.Labels)
```

```
accuracy = 0.9167
```

```
save myNet.mat
```

```
predictedLabels = classify(myNet, testImages);
accuracy = mean(predictedLabels == testImages2.Labels)
```

```
accuracy = 0.9167
```