clear all;

clc

for k=1:10

filename=['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTrain{k,:}=coeffs;

end

for k=11:17

filename=['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTrain{k,:}=coeffs;

end

for k = 18:27

filename = ['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs] = audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs = coeffs';

xTrain{k,:} = coeffs;

end

for k=28:37

filename=['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTrain{k,:}=coeffs;

end

for k=38:47

filename=['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTrain{k,:}=coeffs;

end

yTrain ={'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'2';'2';'2';'2';'2';'2';'2';'3';'3';'3';'3';'3';'3';'3';'3';'3';'3';'4';'4';'4';'4';'4';'4';'4';'4';'4';'4';'5';'5';'5';'5';'5';'5';'5';'5';'5';'5'};

yTrain=categorical(yTrain);

numFeatures = 14;

numHiddenUnits = 100;

numClasses = 5;

layers = [ ...

sequenceInputLayer(numFeatures)

lstmLayer(numHiddenUnits,'OutputMode','last')

fullyConnectedLayer(numClasses)

softmaxLayer

classificationLayer];

options = trainingOptions('adam', ...

'ExecutionEnvironment','cpu', ...

'GradientThreshold',1, ...

'SequenceLength','longest', ...

'Shuffle','never', ...

'Verbose',0, ...

'Plots','training-progress');

net=trainNetwork(xTrain,yTrain,layers,options)

% define Xtest, yTest

for k=1:10

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTest{k,:}=coeffs;

end

for k=11:17

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

% y=zeros(4,m\*n)

xTest{k,:}=coeffs;

end

for k=18:27

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTest{k,:}=coeffs;

end

for k=28:37

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTest{k,:}=coeffs;

end

for k=38:47

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTest{k,:}=coeffs;

end

yTest ={'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'2';'2';'2';'2';'2';'2';'2';'3';'3';'3';'3';'3';'3';'3';'3';'3';'3';'4';'4';'4';'4';'4';'4';'4';'4';'4';'4';'5';'5';'5';'5';'5';'5';'5';'5';'5';'5'};

yTest=categorical(yTest);

maxEpochs = 100;

miniBatchSize = 27;

gfh

%Classify the test data. Specify the same mini-batch size used for training.

YPred = classify(net,xTest,'MiniBatchSize',miniBatchSize);

%Calculate the classification accuracy of the predictions.

acc = sum(YPred == yTest)./numel(yTest)

clear all;

clc

for k=1:10

filename=['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTrain{k,:}=coeffs;

end

for k=11:17

filename=['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTrain{k,:}=coeffs;

end

for k = 18:27

filename = ['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs] = audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs = coeffs';

xTrain{k,:} = coeffs;

end

for k=28:37

filename=['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTrain{k,:}=coeffs;

end

for k=38:47

filename=['D:\poojarbl\samplepooja\1\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTrain{k,:}=coeffs;

end

yTrain ={'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'2';'2';'2';'2';'2';'2';'2';'3';'3';'3';'3';'3';'3';'3';'3';'3';'3';'4';'4';'4';'4';'4';'4';'4';'4';'4';'4';'5';'5';'5';'5';'5';'5';'5';'5';'5';'5'};

yTrain=categorical(yTrain);

numFeatures = 14;

numHiddenUnits = 100;

numClasses = 5;

layers = [ ...

sequenceInputLayer(numFeatures)

lstmLayer(numHiddenUnits,'OutputMode','last')

fullyConnectedLayer(numClasses)

softmaxLayer

classificationLayer];

options = trainingOptions('adam', ...

'ExecutionEnvironment','cpu', ...

'GradientThreshold',1, ...

'SequenceLength','longest', ...

'Shuffle','never', ...

'Verbose',0, ...

'Plots','training-progress');

net=trainNetwork(xTrain,yTrain,layers,options)

% define Xtest, yTest

for k=1:10

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTest{k,:}=coeffs;

end

for k=11:17

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

% y=zeros(4,m\*n)

xTest{k,:}=coeffs;

end

for k=18:27

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTest{k,:}=coeffs;

end

for k=28:37

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTest{k,:}=coeffs;

end

for k=38:47

filename=['D:\poojarbl\samplepooja\2\_',int2str(k),'.wav'];

[x,fs]= audioread(filename);

coeffs=mfcc(x,fs);

[m,n]=size(coeffs);

coeffs =coeffs';

xTest{k,:}=coeffs;

end

yTest ={'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'2';'2';'2';'2';'2';'2';'2';'3';'3';'3';'3';'3';'3';'3';'3';'3';'3';'4';'4';'4';'4';'4';'4';'4';'4';'4';'4';'5';'5';'5';'5';'5';'5';'5';'5';'5';'5'};

yTest=categorical(yTest);

maxEpochs = 100;

miniBatchSize = 27;

gfh

%Classify the test data. Specify the same mini-batch size used for training.

YPred = classify(net,xTest,'MiniBatchSize',miniBatchSize);

%Calculate the classification accuracy of the predictions.

acc = sum(YPred == yTest)./numel(yTest)