Table of Contents

PROJECT 2

Matt Ruffner EE699 Speech Processing March 7th, 2019

```
% generate/load test data
[testData,trainData] = generateFeatures('audio/');

% generate covariance matrix over entire dataset
allData=[];
for i=1:5
    for j=1:length(testData{i})
        allData=[allData; testData{i}{j}.audio];
    end
end
dataCov=cov(allData);

% create empty confusion matricies for the different analyses
Cf=zeros(5,5);
CfG1=zeros(5,5);
CfG2=zeros(5,5);
CfG3=zeros(5,5);
CfNorm=zeros(5,5);
CfSlow=zeros(5,5);
CfFast=zeros(5,5);
CfSoft=zeros(5,5);
```

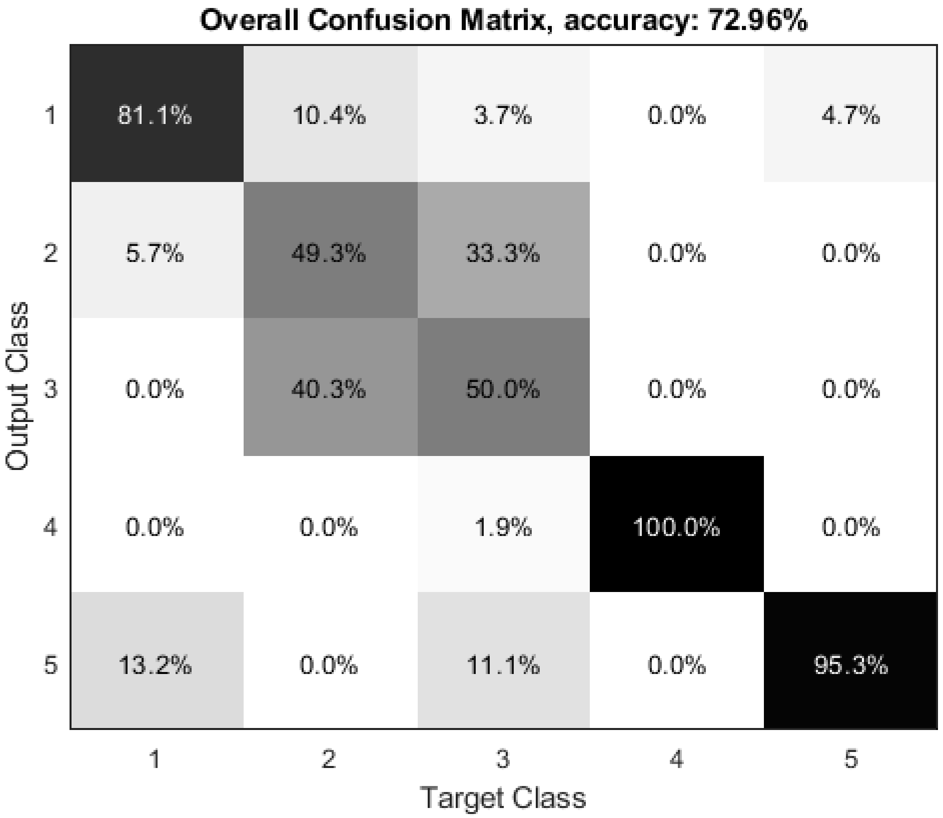
COMPARE BASED ON WORD, SPEAKER AND CONDITION. CREATE CONFUSION MATRICIES

iterate over 'training' words

```
for i=1:5
    % compare with the 5 kinds of 'test' words
    for k=1:length(testData{i})
        Cc=zeros(1,5);
        for j=1:5
            % USE MY DTW FUNCTION
            [Cc(j),path]=ruffdtw(trainData{i}.audio, testData{j}
{k}.audio, dataCov);
```

```
            % THIS LINE USES THE BUILT IN FUNCTION FOR REFERENCE
            %Cc(j)=dtw(trainData{i}.audio', testData{j}{k}.audio');
        end
        [val,pos]=min(Cc);
        % CATEGORIZE BASED ON SPEAKER
        switch testData{1}{k}.cat
            case 'GENERAL1'
                CfG1(i,pos)=CfG1(i,pos)+1;
            case 'GENERAL2'
                CfG2(i,pos)=CfG2(i,pos)+1;
            case 'GENERAL3'
                CfG3(i,pos)=CfG3(i,pos)+1;
        end
        % CATEGORIZE BASED ON ENUNCIATION
        switch testData{1}{k}.type
            case 'FAST'
                CfFast(i,pos)=CfFast(i,pos)+1;
            case 'SLOW'
                CfSlow(i,pos)=CfSlow(i,pos)+1;
            case 'SOFT'
                CfSoft(i,pos)=CfSoft(i,pos)+1;
            case 'TRAIN'
                CfNorm(i,pos)=CfNorm(i,pos)+1;
        end
        Cf(i,pos)=Cf(i,pos)+1;
    end
    % NORMALIZE CONFUSION MATRICIES
    Cf(i,:)=Cf(i,:)./sum(Cf(i,:));
    CfG1(i,:)=CfG1(i,:)./sum(CfG1(i,:));
    CfG2(i,:)=CfG2(i,:)./sum(CfG2(i,:));
    CfG3(i,:)=CfG3(i,:)./sum(CfG3(i,:));
    CfFast(i,:)=CfFast(i,:)./sum(CfFast(i,:));
    CfSlow(i,:)=CfSlow(i,:)./sum(CfSlow(i,:));
    CfSoft(i,:)=CfSoft(i,:)./sum(CfSoft(i,:));
    CfNorm(i,:)=CfNorm(i,:)./sum(CfNorm(i,:));
end

% PLOT CONFUSION MATRICIES
% SCRIPT THANKS TO Vahe Tshitoyan
% https://github.com/vtshitoyan/plotConfMat

figure(1)
acc=plotConfMat(Cf);
title(sprintf('Overall Confusion Matrix, accuracy: %.2f%%', acc))

figure(2)
acc=plotConfMat(CfG1);
title(sprintf('GENERAL1 Confusion Matrix, accuracy: %.2f%%', acc))

figure(3)
acc=plotConfMat(CfG2);
title(sprintf('GENERAL2 Confusion Matrix, accuracy: %.2f%%', acc))

figure(4)
```

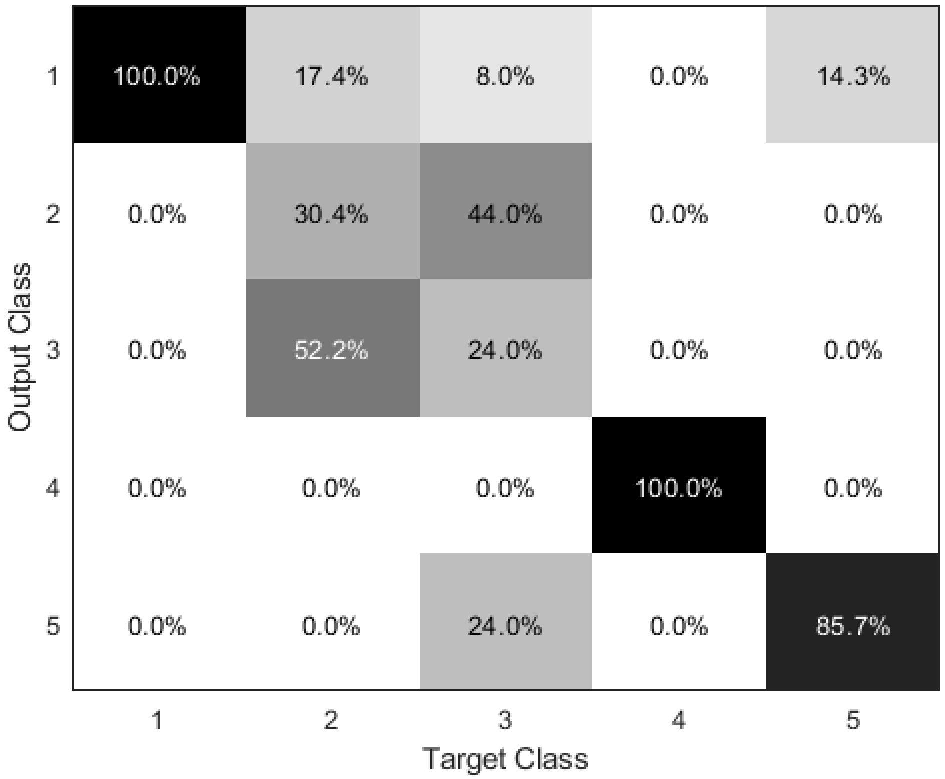
```
acc=plotConfMat(CfG3);
title(sprintf('GENERAL3 Confusion Matrix, accuracy: %.2f%%', acc))

figure(5)
acc=plotConfMat(CfNorm);
title(sprintf('Normal Voice Confusion Matrix, accuracy: %.2f%%', acc))

figure(6)
acc=plotConfMat(CfSlow);
title(sprintf('Slow Voice Confusion Matrix,  accuracy: %.2f%%', acc))

figure(7)
acc=plotConfMat(CfFast);
title(sprintf('Fast Voice Confusion Matrix,  accuracy: %.2f%%', acc))

figure(8)
acc=plotConfMat(CfSoft);
title(sprintf('Soft Voice Confusion Matrix,  accuracy: %.2f%%', acc))
```



Overall Confusion Matrix, accuracy: 72.96%

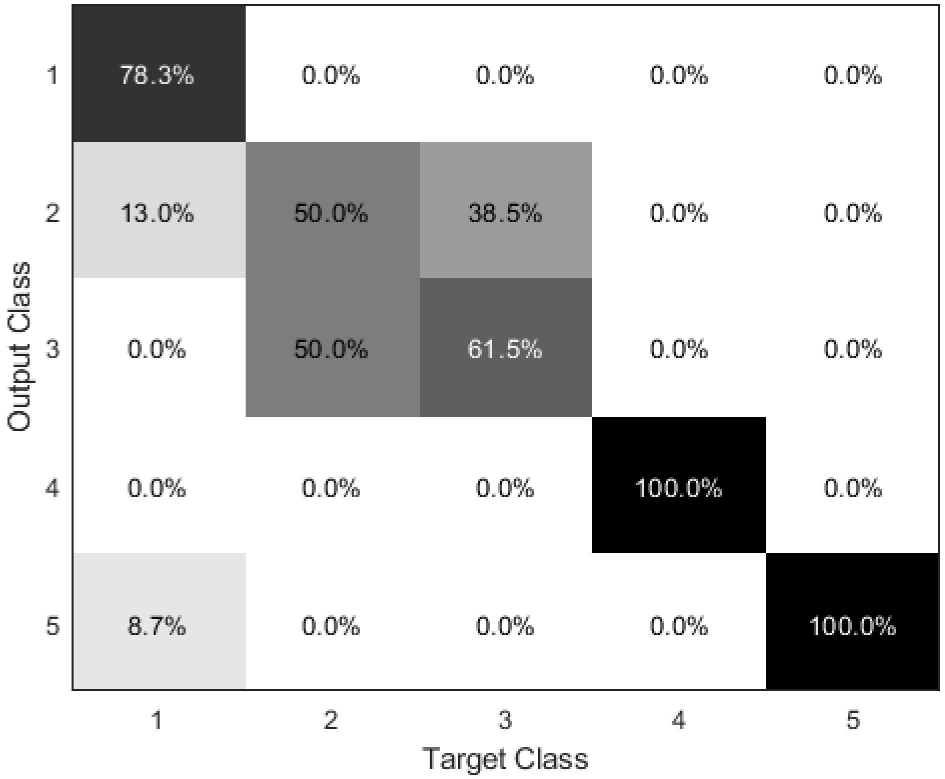
Output Class \ Target Class	1	2	3	4	5
1	81.1%	10.4%	3.7%	0.0%	4.7%
2	5.7%	49.3%	33.3%	0.0%	0.0%
3	0.0%	40.3%	50.0%	0.0%	0.0%
4	0.0%	0.0%	1.9%	100.0%	0.0%
5	13.2%	0.0%	11.1%	0.0%	95.3%

GENERAL1 Confusion Matrix, accuracy: 58.89%

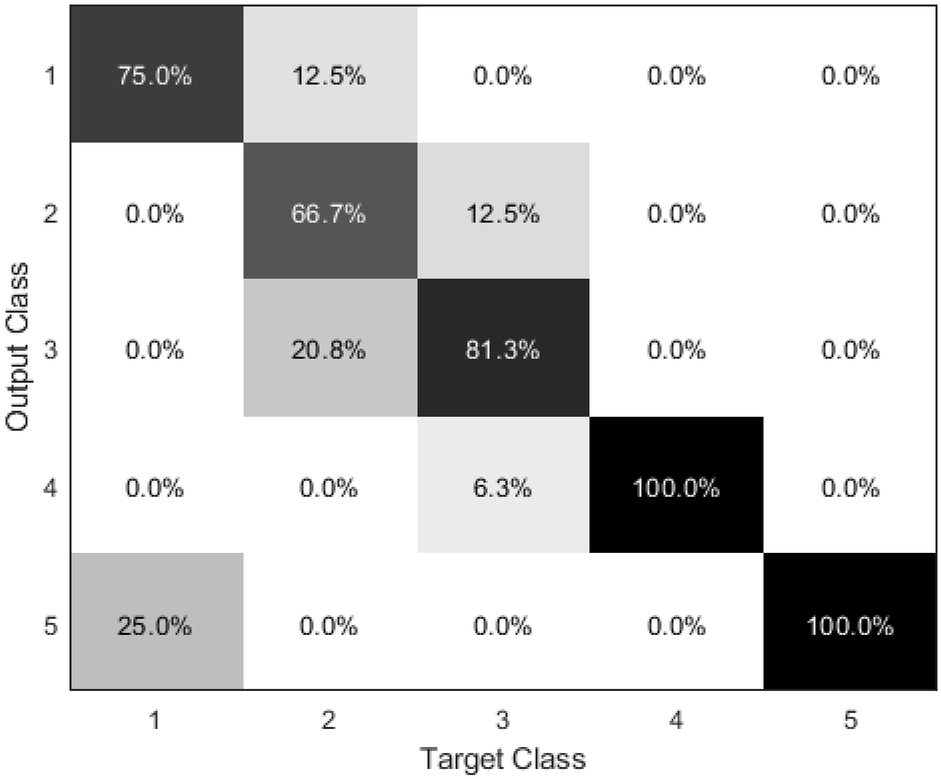


Output Class \ Target Class	1	2	3	4	5
1	100.0%	17.4%	8.0%	0.0%	14.3%
2	0.0%	30.4%	44.0%	0.0%	0.0%
3	0.0%	52.2%	24.0%	0.0%	0.0%
4	0.0%	0.0%	0.0%	100.0%	0.0%
5	0.0%	0.0%	24.0%	0.0%	85.7%

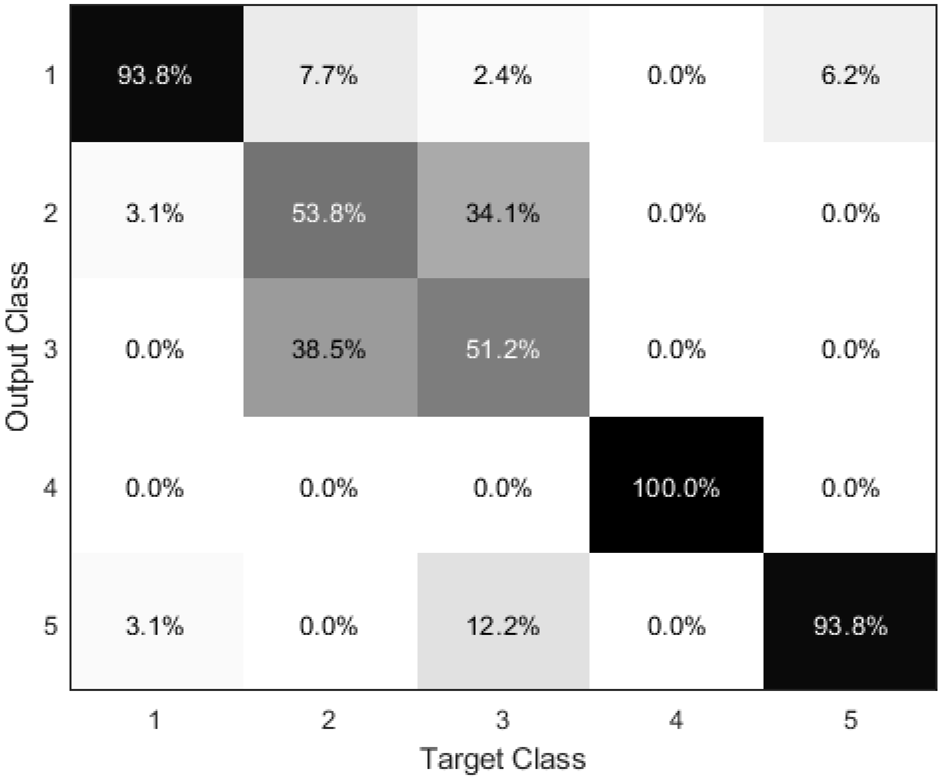
GENERAL2 Confusion Matrix, accuracy: 77.78%



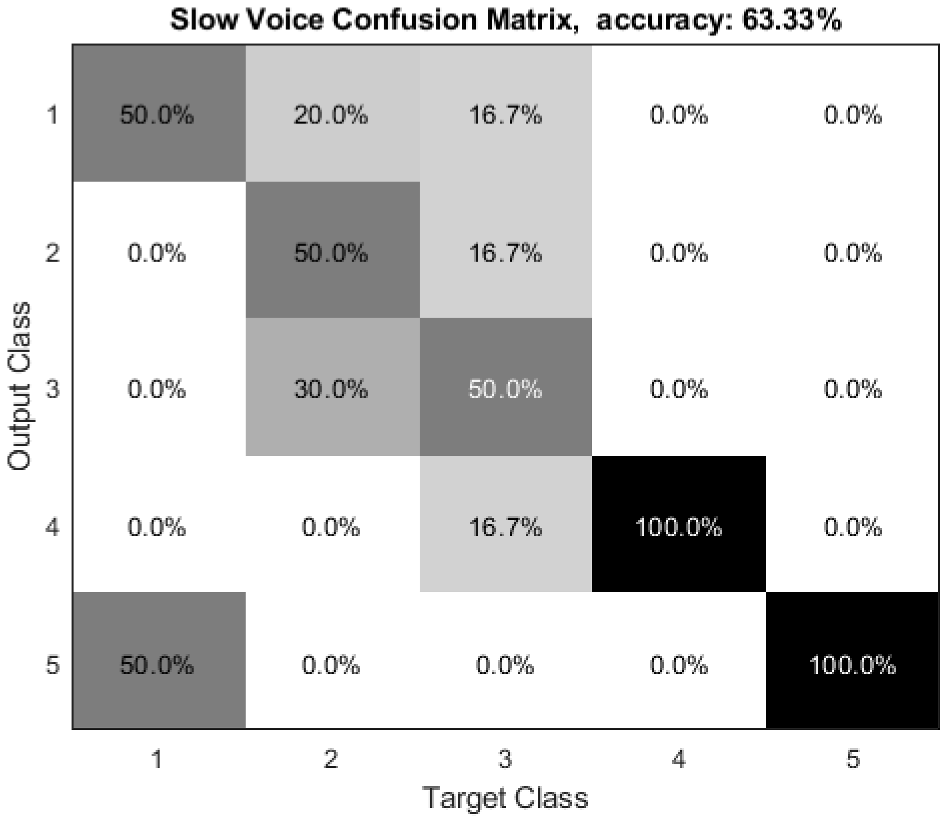
Output Class \ Target Class	1	2	3	4	5
1	78.3%	0.0%	0.0%	0.0%	0.0%
2	13.0%	50.0%	38.5%	0.0%	0.0%
3	0.0%	50.0%	61.5%	0.0%	0.0%
4	0.0%	0.0%	0.0%	100.0%	0.0%
5	8.7%	0.0%	0.0%	0.0%	100.0%



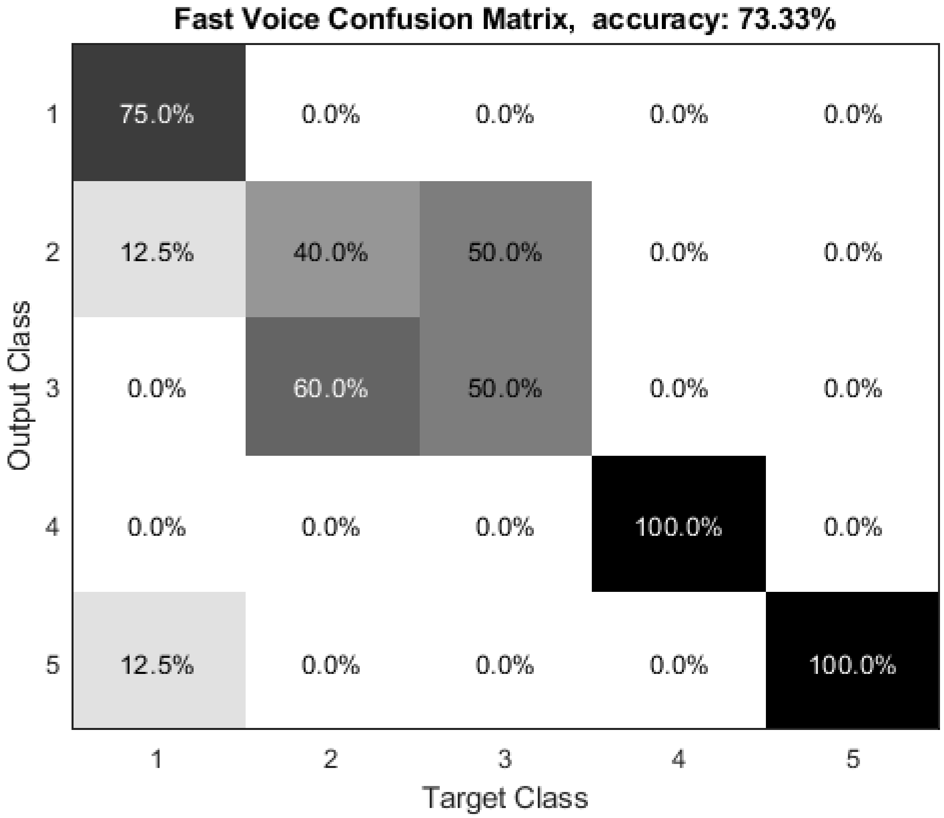
GENERAL3 Confusion Matrix, accuracy: 82.22%



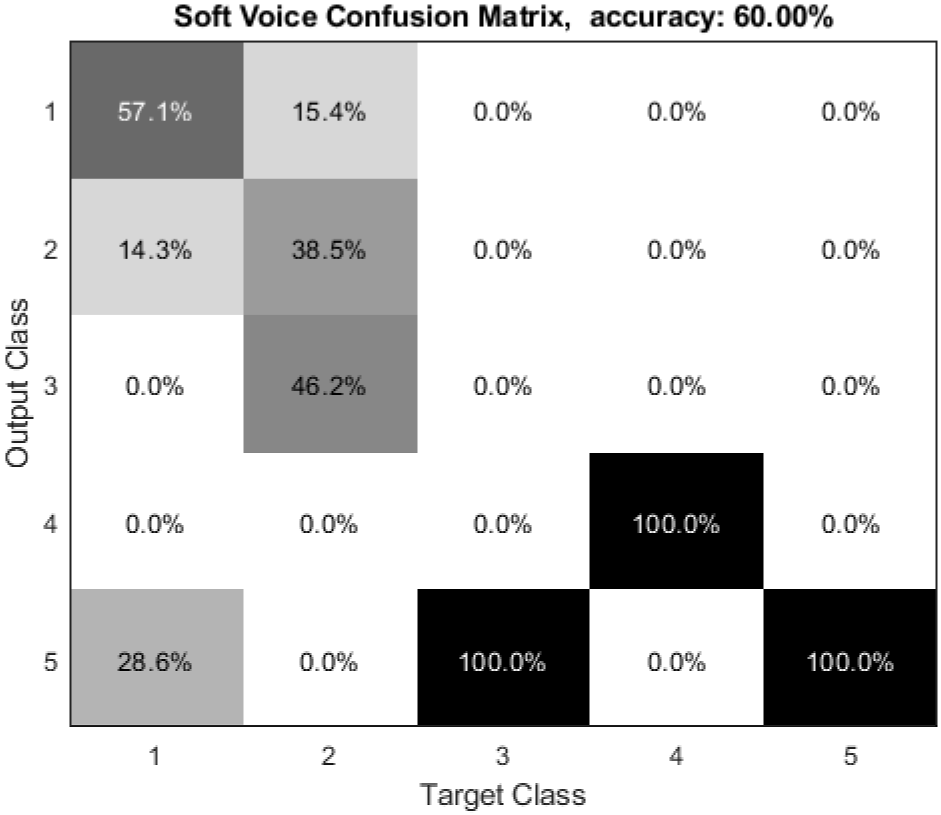
Normal Voice Confusion Matrix, accuracy: 76.67%



Slow Voice Confusion Matrix, accuracy: 63.33%



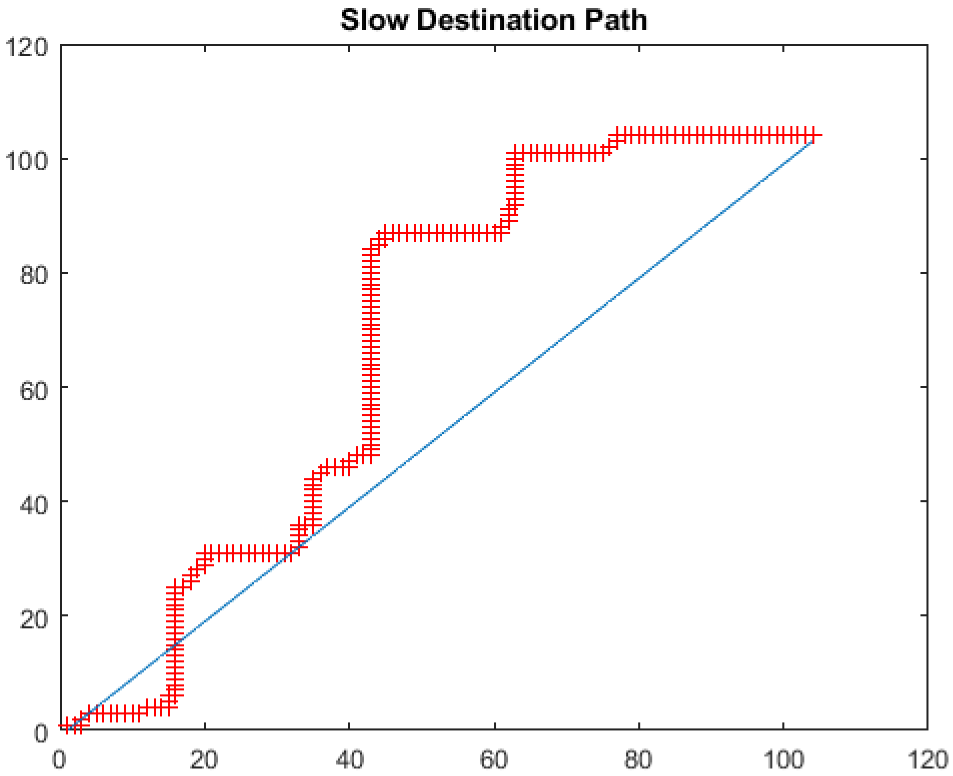
Fast Voice Confusion Matrix, accuracy: 73.33%

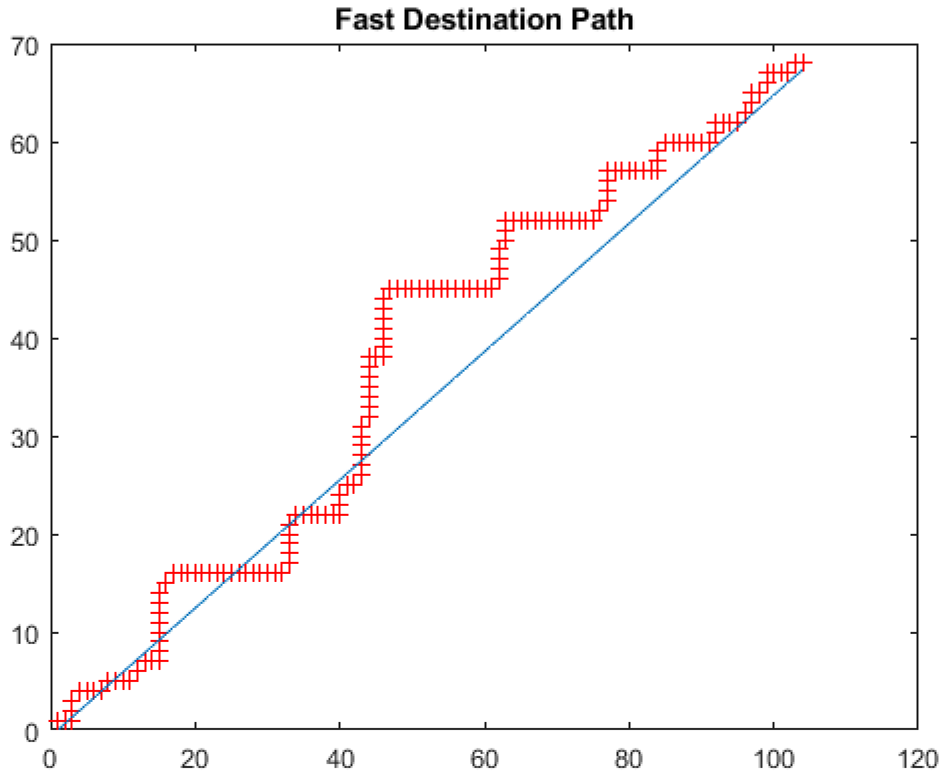


EXAMPLE ALIGNMENTS FOR 'FAST' VS 'SLOW' DESTINATION

```
% test a slow destination
[sDist,sPath]=ruffdtw(trainData{4}.audio,testData{4}{11}.audio,
 dataCov);

% test a fast destination
[fDist,fPath]=ruffdtw(trainData{4}.audio,testData{4}{22}.audio,
 dataCov);

sSize=max(sPath);
fSize=max(fPath);
sMidLine=sSize(2)*(0:1/sSize(1):1-1/sSize(1));
fMidLine=fSize(2)*(0:1/fSize(1):1-1/fSize(1));

figure(9)
plot(sPath(:,1),sPath(:,2), 'r+')
hold on
plot(1:sSize(1),sMidLine)
title('Slow Destination Path')
hold off

figure(10)
```

```
plot(fPath(:,1),fPath(:,2), 'r+')
hold on
plot(1:fSize(1),fMidLine)
title('Fast Destination Path')
hold off
```

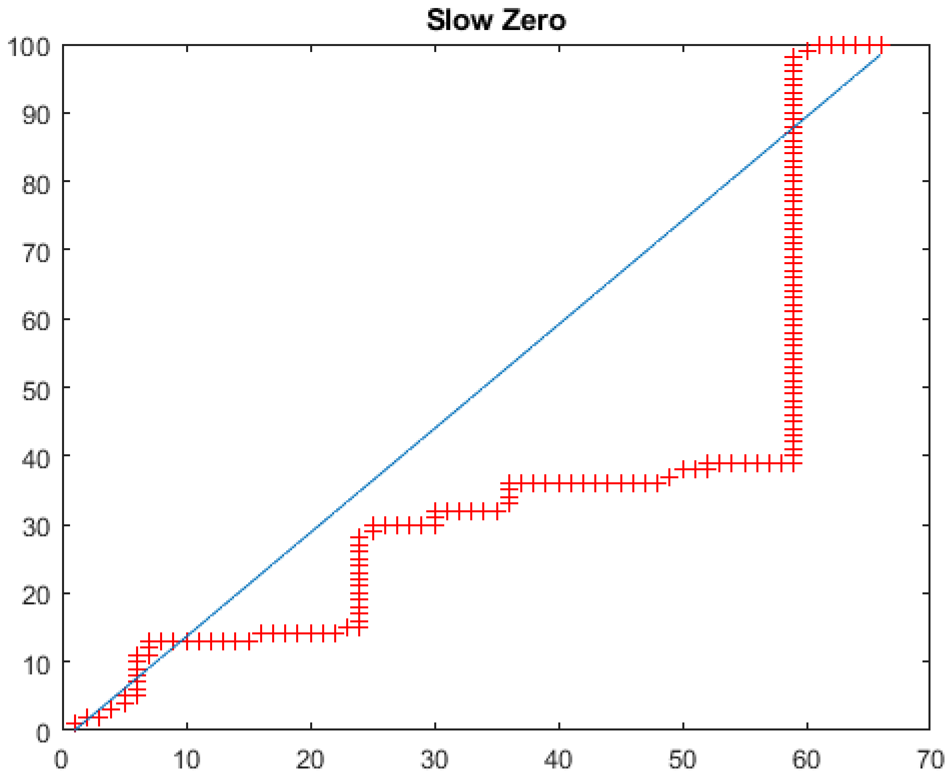


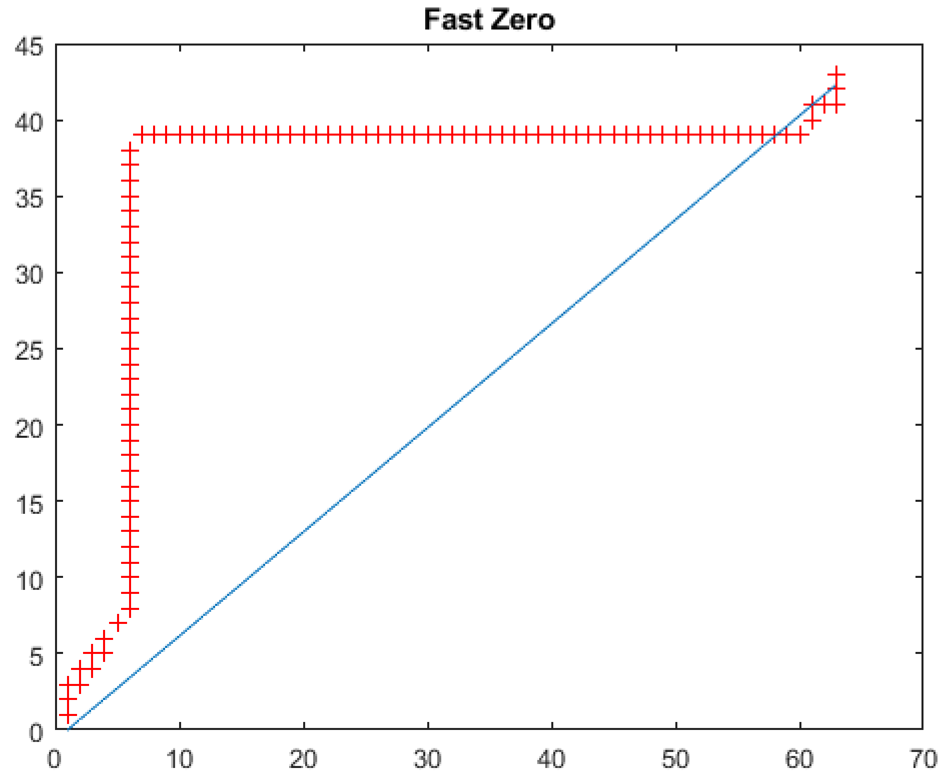


EXAMPLE ALIGNMENTS FOR 'FAST' VS 'SLOW' ZERO

```
% test a slow destination
[sDist,sPath]=ruffdtw(trainData{5}.audio,testData{5}{11}.audio,
 dataCov);

% test a fast destination
[fDist,fPath]=ruffdtw(trainData{5}.audio,testData{5}{22}.audio,
 dataCov);

sSize=max(sPath);
fSize=max(fPath);
sMidLine=sSize(2)*(0:1/sSize(1):1-1/sSize(1));
fMidLine=fSize(2)*(0:1/fSize(1):1-1/fSize(1));

figure(11)
plot(sPath(:,1),sPath(:,2), 'r+')
hold on
plot(1:sSize(1),sMidLine)
title('Slow Zero')
hold off

figure(12)
```

```
plot(fPath(:,1),fPath(:,2), 'r+')
hold on
plot(1:fSize(1),fMidLine)
title('Fast Zero')
hold off
```





Published with MATLAB® R2018b