```
clear
clc
close all
addpath mfcc
warning off

count = 0;
MFCCf=[];
data= ["sen_1NN.m4a", "sen_2NN.m4a", "sen_3NN.m4a", "sen_4NN.m4a", "sen_5NN.m4a", "sen_6SN.m4a"
       "sen_1NO.m4a", "sen_2NO.m4a", "sen_3NO.m4a", "sen_4NO.m4a", "sen_5NO.m4a", "sen_6SO.m4a"
       "sen_1NS.m4a", "sen_2NS.m4a", "sen_3NS.m4a", "sen_4NS.m4a", "sen_5NS.m4a", "sen_6SS.m4a"

for F=1:length(data)
    count=0;

    [x, fs]=audioread(data(F));

    fr_len = 0.8;                       % 80ms frames
    fr_N = ((fr_len)*fs);               % number of samples in 1s frame
    R_shift = 1*fr_N;

    for i = 1:R_shift:(length(x)-fr_N)
        count=count+1;
        n=[i:i+fr_N-1];

        speech = x(n);
        Tw = 25;           % analysis frame duration (ms)
        Ts = 19;           % analysis frame shift (ms)
        alpha = 0.97;      % preemphasis coefficient
        R = [100 1000];    % frequency range to consider
        M = 20;            % number of filterbank channels
        C = 12;            % number of cepstral coefficients
        L = 22;            % cepstral sine lifter parameter

        hamming = @(N)(0.54-0.46*cos(2*pi*[0:N-1].'/(N-1)));

        % Feature extraction (feature vectors as columns)
        [ MFCCs, FBEs, frames,eframes ] = mfcc( speech, fs, Tw, Ts, alpha, hamming, R, M, C, L

        for k = 1:size(MFCCs,2)
            if and(k > 1,k<size(MFCCs,2))
                MFCCd(:,k) = (MFCCs(:,k+1) - MFCCs(:,k-1))/2;
                eframesd(k) = (eframes(k+1) - eframes(k-1))/2;
            else
                MFCCd(:,k) = MFCCs(:,k);
                eframesd(k) = eframes(k);
            end
        end

        for j = 1:size(MFCCs,2)
            if and(j > 1,j<size(MFCCs,2))
                 MFCCdd(:,j) = (MFCCd(:,j+1) - MFCCd(:,j-1))/2;
                 eframesdd(j) = (eframesd(j+1) - eframesd(j-1))/2;
```

```
            else
                MFCCdd(:,j) = MFCCd(:,j);
                eframesdd(j) = eframesd(j);
            end
        end

        MFCCfeat = transpose([MFCCs;MFCCd;MFCCdd;eframes;eframesd;eframesdd]);

        MFCCf=[MFCCf;MFCCfeat];
        X=MFCCf;
        X(isnan(X))=0;

        k=2;
        % fit MFCC features in a Gaussian Mixture Model and extract means
        % Find edges using the given formula in the paper
        GMM_model = fitgmdist(X,k,'RegularizationValue',0.5,'CovarianceType','diagonal');
        for index=1:length(GMM_model.mu(1,:))
            edges(index)=max(GMM_model.mu(:,index))-min(GMM_model.mu(:,index));
        end
        Vol_matrix(count,F) = prod(edges, 'all');
    end
end

for F=1:length(data)
    [Volume] = nonzeros(Vol_matrix(:,F));
    Variance(F)=round(var(Volume),3);
end
Variance = Variance';
%writematrix(Variance, 'MyVAR_z.txt');
```