Appendix A: Matlab® Code

```
% 1) Import Data

files = dir('*.cep');
for i=1:length(files)
eval(['load ' files(i).name ' -ascii']);
end


% 2) Create Input, Test and Target Matrices
% 2.a) Input Matrix
input = [];
for i = 1:240;
    eval (['input = [ input a' num2str(i) ' ]; ' ]);
    eval (['input = [ input ae' num2str(i) ' ]; ' ]);
    eval (['input = [ input e' num2str(i) ' ]; ' ]);
    eval (['input = [ input i' num2str(i) ' ]; ' ]);
    eval (['input = [ input o' num2str(i) ' ]; ' ]);
    eval (['input = [ input u' num2str(i) ' ]; ' ]);
end


% 2.b) Test Matrix

test = [];
for i = 1:120;
    eval (['test = [test a' num2str(i) ' ]; ' ]);
    eval (['test = [test ae' num2str(i) ' ]; ' ]);
    eval (['test = [test e' num2str(i) ' ]; ' ]);
    eval (['test = [test i' num2str(i) ' ]; ' ]);
    eval (['test = [test o' num2str(i) ' ]; ' ]);
    eval (['test = [test u' num2str(i) ' ]; ' ]);
end

% 2.c) Target Matrix

target1= [.9 .1 .1 .1 .1 .1]';
target2= [.1 .9 .1 .1 .1 .1]';
target3= [.1 .1 .9 .1 .1 .1]';
target4= [.1 .1 .1 .9 .1 .1]';
target5= [.1 .1 .1 .1 .9 .1]';
target6= [.1 .1 .1 .1 .1 .9]';
t= [target1 target2 target3 target4 target5 target6];
target=[];
for i=1:240
target=[target t];
end
```

```
% 3) Create Network
% 3.a) FFBP network trained using Levenberg-Marquardt
algorithm

numHiddenNeurons = 20; % Adjust from 20 to 200 with a step of
20
net =
newff(input,target,numHiddenNeurons,{'tansig','purelin'},'tra
inlm','learngdm','mse');
net.divideParam.trainRatio = 60/100;
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 20/100;
net.trainparam.show = 25;
net.trainparam.epoches = 1000;
net.trainparam.time = inf;
net.trainparam.goal = 0;
net.trainparam.max_fail = 6;
net.trainparam.mem_reduc = 1;
net.trainparam.min_grad = 1e-010;
net.trainparam.mu = 0.001;
net.trainparam.mu_dec = 0.1;
net.trainparam.mu_inc = 10;
net.trainparam.mu_max = 10000000000;

% 3.b) FFBP network trained using Resilient Backpropagation
algorithm

% numHiddenNeurons = 20; % Adjust from 20 to 200 with a step
of 20
% net =
newff(input,target,numHiddenNeurons,{'tansig','purelin'}c,'tr
ainrp','learngdm','mse');
% net.divideParam.trainRatio = 60/100;
% net.divideParam.valRatio = 20/100;
% net.divideParam.testRatio = 20/100;
% net.trainparam.show = 25;
% net.trainparam.epoches = 1000;
% net.trainparam.time = inf;
% net.trainparam.goal = 0;
% net.trainparam.max_fail = 6;
% net.trainparam.min_grad = 1e-010;
% net.trainparam .delt_inc = 1.2;
% net.trainparam .delt_dec = 0.5;
% net.trainparam .delta0 = 0.07;
% net.trainparam .deltamax = 50;
```

```
% 3.c) Elman network trained using Variable Learning Rate
Gradient Descent algorithm

% numHiddenNeurons=20; % Adjust from 20 to 200 with a step of
20
% net =
newelm(input,target,numHiddenNeurons,{'tansig','purelin'},'tr
aingdx','learngdm','mse');
% net.divideParam.trainRatio = 60/100;
% net.divideParam.valRatio = 20/100;
% net.divideParam.testRatio = 20/100;
% net.trainparam.show = 25;
% net.trainparam.epoches = 1000;
% net.trainparam.time = inf;
% net.trainparam.goal = 0;
% net.trainparam.max_fail = 6;
% net.trainparam.lr = 0.01;
% net.trainparam.lr_inc = 1.05;
% net.trainparam.lr_dec = 0.7;
% net.trainparam.max_perf_in = 1.04;
% net.trainparam.mc = 0.9;
% net.trainparam.min_grad = 1e-010;



% 4) Train and Apply Network

[net,tr] = train(net,input,target);
output = sim(net,test);



% 5) Process the Output Data
% 5.a) Getting the maximum value of each column

y=[];
for i=1:720
y(i)= max(output(:,i));
end
```

```
% 5.b) Equaling the maximum value of each column to 1 and
setting other values to zeros

for i=1:720
for j=1:6
if output(j,i)==y(i);
output(j,i)= 1;
x=0;
else
output(j,i)=0;
end
end
end


% 6) Master key the target and the output matrices to 1D
matrices
% 6.a) Creating 1D matrix of the output

o=[];
for i=1:720
if output(:,i)==[1;0;0;0;0;0];
o=[o 1];
else if output(:,i)==[0;1;0;0;0;0];
o=[o 2];
else if output(:,i)==[0;0;1;0;0;0];
o=[o 3];
else if output(:,i)==[0;0;0;1;0;0];
o=[o 4];
else if output(:,i)==[0;0;0;0;1;0];
o=[o 5];
else if output(:,i)==[0;0;0;0;0;1];
o=[o 6];
else
o=[o 7];
end
end
end
end
end
end
end
```

```
% 6.b) Creating 1D matrix of the target

w=[1 2 3 4 5 6];
t=[];
for i=1:120
t=[t w];
end



% 7) Getting the confusion matrix and recognition rates

cm= confusionmat(t,o);
Recognition_Rate_a= (cm(1,1)/120)*100
Recognition_Rate_ae= (cm(2,2)/120)*100
Recognition_Rate_e= (cm(3,3)/120)*100
Recognition_Rate_i= (cm(4,4)/120)*100
Recognition_Rate_o= (cm(5,5)/120)*100
Recognition_Rate_u= (cm(6,6)/120)*100
Total_Recognition_Rate=
((cm(1,1)+cm(2,2)+cm(3,3)+cm(4,4)+cm(5,5)+cm(6,6))/720)*100
```

Appendix B: Recognition Rates' Results

Table B.1: Recognition rate (in %) of different number of hidden neurons for each frame data set using FFBP network. (Set 1)

Neurons / Frame	20	40	60	80	100	120	140	160	180	200
SF10ms	76	75	78	78	73	78	74	78	66	67
SF15ms	78	74	79	78	78	82	17	60	66	67
SF20ms	79	68	67	79	72	73	82	81	78	73
SF25ms	80	73	63	81	83	81	18	67	78	80
SF30ms	79	72	63	72	71	71	84	54	83	81
SF35ms	80	70	68	71	68	83	83	60	52	81
SF40ms	78	70	69	83	83	83	71	80	80	81
SF45ms	62	76	76	81	67	83	19	58	63	81
SF50ms	79	75	67	83	67	83	83	40	70	82
SF55ms	68	70	62	82	66	84	25	77	55	81
SF60ms	79	71	70	81	36	83	30	76	28	82
SF65ms	83	81	78	82	69	82	76	59	80	81
SF70ms	79	71	56	80	71	82	73	75	78	82
MF30ms	77	80	78	78	78	78	78	78	77	80
MF40ms	78	79	77	66	78	79	77	63	79	79
MF50ms	77	78	75	77	76	76	78	79	65	66
MF60ms	76	78	79	80	79	65	77	73	71	69
MF70ms	78	77	73	79	79	73	75	59	47	69
MF80ms	75	77	80	79	61	66	59	59	47	71
MF90ms	79	77	79	77	75	66	60	79	68	67
MF100ms	76	76	67	79	71	79	67	59	55	47

Table B.2: Recognition rate (in %) of different number of hidden neurons for each frame data set using FFBP network. (Set 2)

Neurons / Frame	20	40	60	80	100	120	140	160	180	200
SF10ms	76	74	63	78	45	76	71	32	80	51
SF15ms	75	79	79	80	74	80	80	79	81	73
SF20ms	78	73	81	82	81	80	82	72	82	80
SF25ms	62	73	83	84	80	78	82	82	83	80
SF30ms	78	80	82	81	75	81	82	22	83	79
SF35ms	80	81	76	82	81	83	69	80	33	43
SF40ms	56	79	81	84	81	81	83	83	83	80
SF45ms	78	58	71	60	73	83	84	31	80	80
SF50ms	78	57	78	84	71	83	83	82	83	81
SF55ms	82	71	73	84	81	82	83	29	81	81
SF60ms	78	83	14	83	66	84	72	82	84	82
SF65ms	81	80	81	84	82	71	82	83	82	82
SF70ms	82	63	58	83	73	80	82	82	83	82
MF30ms	81	78	82	81	81	79	79	79	80	80
MF40ms	80	81	81	79	80	79	81	72	80	79
MF50ms	79	82	79	82	78	80	79	80	72	81
MF60ms	81	80	79	73	79	81	79	76	69	72
MF70ms	81	78	81	74	80	78	79	73	70	68
MF80ms	78	78	78	82	80	72	80	79	72	78
MF90ms	80	81	78	66	64	80	78	78	60	67
MF100ms	77	80	62	79	80	80	80	60	56	80

Table B.3: Recognition rate (in %) of different number of hidden neurons for each frame data set using FFBP network. (Set 3)

Neurons / Frame	20	40	60	80	100	120	140	160	180	200
SF10ms	75	78	77	75	61	73	78	78	75	73
SF15ms	78	65	78	46	79	82	80	71	77	74
SF20ms	73	71	74	36	76	81	73	80	33	76
SF25ms	82	79	81	73	80	83	76	63	78	77
SF30ms	81	82	81	78	75	79	51	78	82	80
SF35ms	61	72	83	81	74	85	78	82	79	78
SF40ms	83	81	83	75	77	68	26	19	84	78
SF45ms	81	79	83	78	78	72	77	84	25	81
SF50ms	80	84	82	66	82	83	78	65	80	79
SF55ms	82	82	80	72	78	84	80	28	82	79
SF60ms	78	82	83	79	82	83	79	78	83	78
SF65ms	82	84	83	78	74	52	75	84	75	82
SF70ms	82	81	85	75	81	83	79	82	81	82
MF30ms	81	81	81	79	67	82	82	82	82	37
MF40ms	83	81	84	83	82	82	74	82	75	70
MF50ms	82	83	83	82	79	82	83	69	75	62
MF60ms	81	83	82	81	81	69	69	69	73	48
MF70ms	80	82	82	82	82	83	74	74	62	59
MF80ms	80	80	82	80	82	80	80	73	74	63
MF90ms	80	83	83	82	81	80	81	80	72	82
MF100ms	79	81	81	75	82	81	81	74	81	70

Table B.4: Recognition rate (in %) of different number of hidden neurons for each frame data set using Elman network. (Set 1)

Neurons / Frame	20	40	60	80	100	120	140	160	180	200
SF10ms	76	75	76	17	38	29	22	73	17	73
SF15ms	76	77	76	17	39	28	23	76	17	75
SF20ms	79	25	76	17	37	31	26	76	17	75
SF25ms	78	78	78	77	78	37	22	77	76	76
SF30ms	78	78	77	17	40	78	24	63	76	75
SF35ms	80	24	77	17	37	32	25	77	17	76
SF40ms	78	78	17	53	78	38	22	77	78	77
SF45ms	78	79	17	50	78	78	20	77	77	77
SF50ms	77	77	78	17	38	32	24	77	17	78
SF55ms	79	24	27	27	78	38	21	78	78	77
SF60ms	79	79	17	52	76	78	34	40	27	77
SF65ms	79	80	17	52	77	38	22	78	78	78
SF70ms	78	80	17	52	77	38	22	78	78	78
MF30ms	78	78	79	77	77	75	77	79	78	76
MF40ms	80	79	79	78	76	38	78	78	78	78
MF50ms	79	80	79	80	78	80	78	79	78	76
MF60ms	77	80	80	80	78	44	79	78	78	79
MF70ms	79	78	80	81	80	78	45	74	78	77
MF80ms	79	79	80	80	79	78	78	78	78	79
MF90ms	79	79	80	79	79	79	79	78	79	80
MF100ms	79	79	80	78	80	78	78	78	79	79

Table B.5: Recognition rate (in %) of different number of hidden neurons for each frame data set using Elman network. (Set 2)

Neurons / Frame	20	40	60	80	100	120	140	160	180	200
SF10ms	69	33	77	78	77	76	76	24	37	68
SF15ms	80	79	80	79	80	30	21	78	30	79
SF20ms	81	80	81	81	81	73	18	79	33	80
SF25ms	81	81	83	82	81	62	18	79	31	81
SF30ms	82	81	81	82	82	73	18	79	34	81
SF35ms	82	81	81	81	82	73	18	79	32	80
SF40ms	81	82	81	80	81	72	18	80	35	80
SF45ms	81	82	82	80	81	73	17	80	34	80
SF50ms	82	81	82	82	80	71	17	80	32	80
SF55ms	80	81	81	81	81	72	18	81	31	79
SF60ms	81	81	83	81	81	71	18	80	33	79
SF65ms	80	81	82	80	82	73	17	79	34	80
SF70ms	81	82	82	80	81	73	18	79	34	80
MF30ms	81	82	82	81	81	17	31	81	16	80
MF40ms	81	80	82	83	82	42	81	81	80	81
MF50ms	80	83	81	82	82	81	82	82	81	81
MF60ms	81	81	82	79	81	81	81	53	72	79
MF70ms	81	83	81	81	82	80	81	80	33	81
MF80ms	81	82	81	82	81	82	82	80	81	80
MF90ms	82	83	80	81	82	82	82	81	81	29
MF100ms	81	82	82	80	82	82	83	80	82	82

Table B.6: Recognition rate (in %) of different number of hidden neurons for each frame data set using Elman network. (Set 3)

Neurons / Frame	20	40	60	80	100	120	140	160	180	200
SF10ms	77	77	75	74	74	69	18	73	34	17
SF15ms	80	78	80	78	77	70	19	77	32	74
SF20ms	82	78	81	80	77	72	20	77	30	76
SF25ms	82	80	82	81	78	72	21	78	34	77
SF30ms	83	81	83	81	80	73	20	80	34	79
SF35ms	85	82	83	82	81	59	21	80	35	79
SF40ms	86	83	84	82	81	73	21	80	80	81
SF45ms	83	83	83	83	81	74	20	81	34	81
SF50ms	86	83	84	82	82	34	21	80	31	79
SF55ms	83	84	85	83	82	73	21	81	35	80
SF60ms	83	84	81	82	23	81	79	39	80	38
SF65ms	84	83	84	83	82	73	22	81	35	81
SF70ms	85	83	84	83	82	73	22	81	35	81
MF30ms	83	81	81	81	81	78	28	81	26	15
MF40ms	81	83	81	44	82	83	83	82	83	81
MF50ms	82	83	83	83	83	84	81	35	83	80
MF60ms	82	81	82	83	83	82	84	56	76	81
MF70ms	75	83	82	82	84	83	82	81	33	82
MF80ms	83	82	83	84	85	82	84	82	81	82
MF90ms	82	83	83	82	82	82	82	81	80	24
MF100ms	83	84	83	85	84	84	84	83	83	82