Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRONICS ENGINEERING (SENSE)

(PROJECT REPORT)

VEHICLE DETECTION BY USING VEHICLE ENGINE SOUND

NEURAL NETWORKS AND FUZZY CONTROL

Course Code: - ECE3009
SLOT-G1
WIN SEM 2019-2020

SUBMITTED BY:

- **VISHAKHA KUMAR** **17BEC0673**
- **MISHU DAVE** **17BEC0835**
- **DIKSHA RAGHUNANTHAN** **17BMD0068**
- **JEVETHA VIJAYADASAN** **17BMD0073**

PROJECT GUIDE

PROF. SARANYA K C

Winter Semester 2019 -20
(MAY)

CERTIFICATE

Certified that this project report “VEHICLE DETECTION BY USING VEHICLE ENGINE SOUND” is the bona-fide work of-

VISHAKHA KUMAR	17BEC0673
MISHU DAVE	17BEC0835
DIKSHA RAGHUNANTHAN	17BMD0068
JEVETHA VIJAYADASAN	17BMD0073

who carried out the project under my supervision of G1 slot and SENSE school during year 2019-2020 at V.I.T. University, Vellore, Pin Code- 632014

Faculty Signature:

Date: 05/06/2020

ACKNOWLEDGEMENTS

We would like to express our gratitude to our faculty Prof. SARANYA K C.*, who gave us the golden opportunity to do this wonderful project on the topic "*VEHICLE DETECTION BY USING VEHICLE ENGINE SOUND*" which helped us in doing a lot of Research work and come across many new things. Secondly, we would also like to thank our university, and other batch mates who helped us a lot in finalizing this project within the limited time frame.*

CONTENTS

AIM

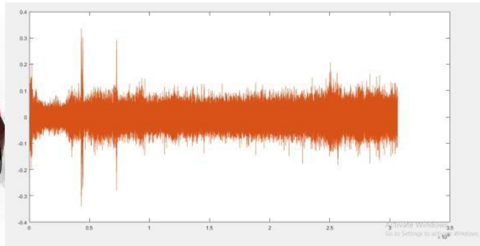
This project aims for detection/classification of cars of different companies/models using 2 layered neural network with back propagation algorithm. Different cars engine sounds will be having different frequencies, this property is utilized for car classification.

INTRODUCTION

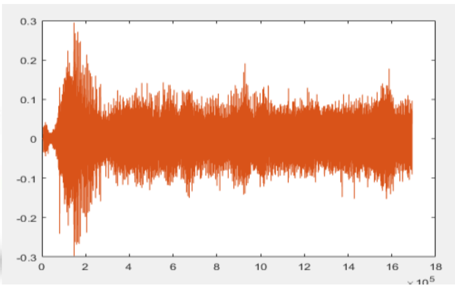
a. Data collection:

- To collect 7 different car engine sounds using mobile microphone.
- To scan for car engine sound

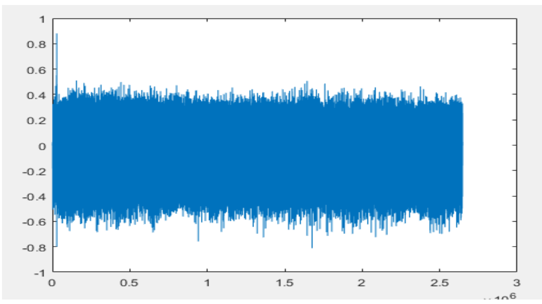
1. ALTO LXI



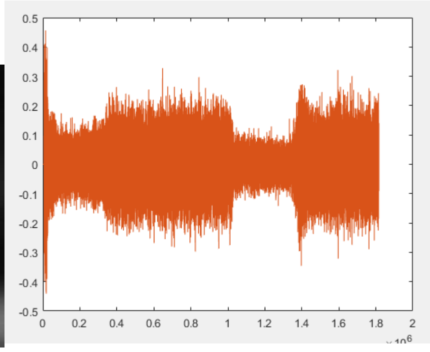
2. ALTO 800



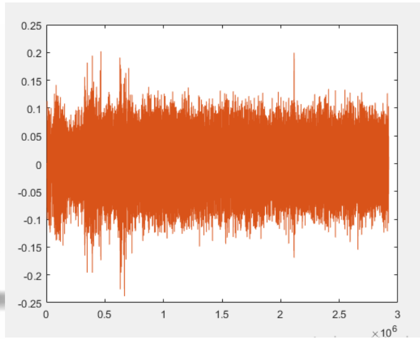
3. FORD



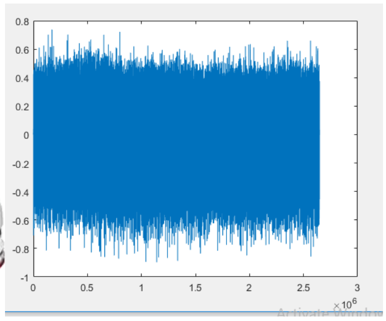
4. CORLA ELTIS



5. TITANIUM



6. SWIFT



b. Data pre-processing:

- The car sounds .mp3 file are read into MATLAB using audio-read command.

  [yalto]=audioread('C:\Users\User\Desktop\Mishu\Sounds\alto.mp3');

  [yalto800]=audioread('C:\Users\User\Desktop\Mishu\Sounds\alto 800.mp3');

  [yford]=audioread('C:\Users\User\Desktop\Mishu\Sounds\ford.mp3');

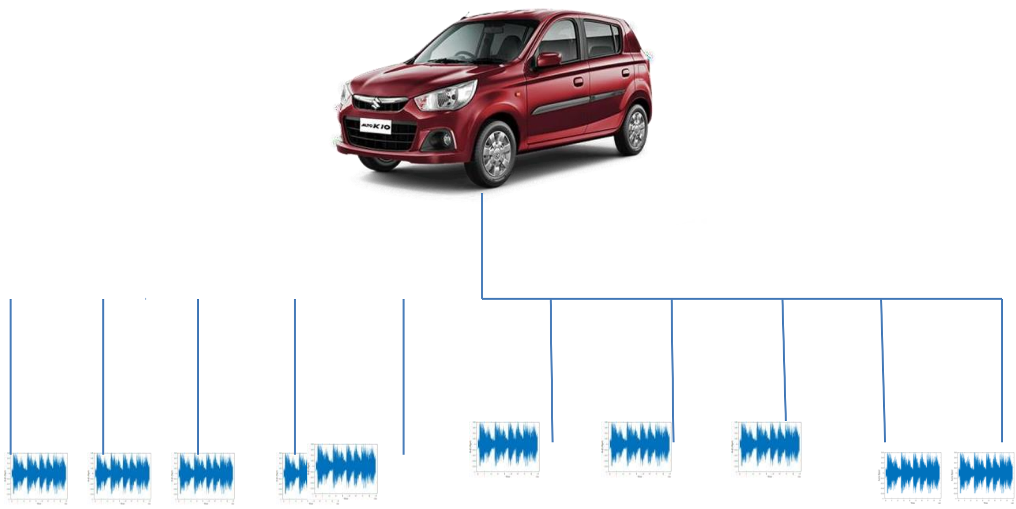
  [ycor]=audioread('C:\Users\User\Desktop\Mishu\Sounds\Corola eltis .mp3');

  [yswift]=audioread('C:\Users\User\Desktop\Mishu\Sounds\swift.mp3');

  [ytoyo]=audioread('C:\Users\User\Desktop\Mishu\Sounds\toyoto.mp3');

  [ytita]=audioread('C:\Users\User\Desktop\Mishu\Sounds\Titanium 4.mp3');

- Then we have to divide each type of car engine sound into 5s, so that we will get 10 training car sound for each type.

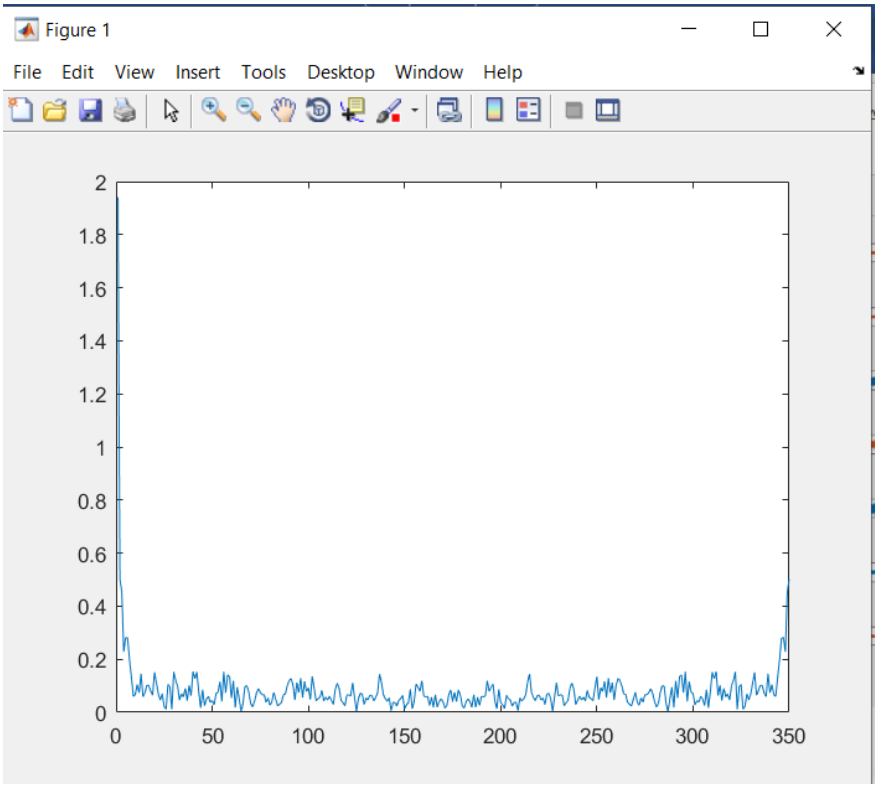


MATLAB CODE:

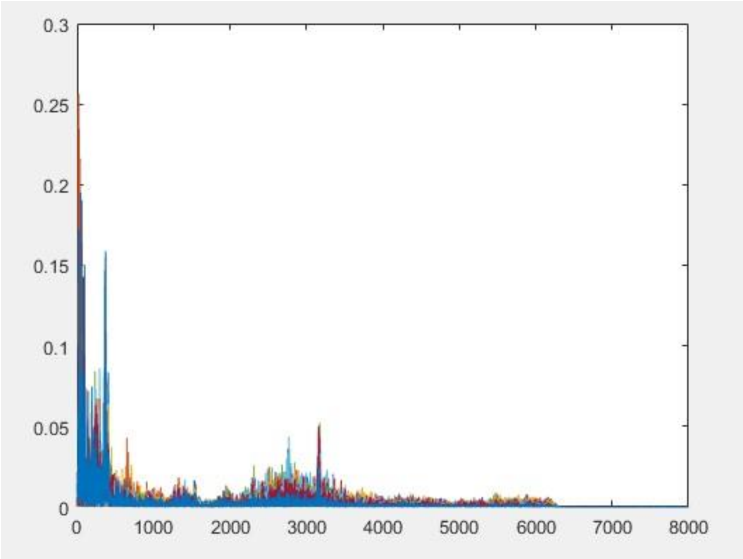
```
function D=dataa(y)
    d=zeros(8000,50);
    a=1;
    q=8000;
    for k=1:50
        d(:,k)=y(a:q);
        a=a+8000;
        q=q+8000;
    end
    df=dct(d);
```

```
dn=df/norm(df);



da=abs(dn);
    D=da([1:10],:);

return
```

- Then we have to apply fast Fourier transform to all trainings engine sounds to get the frequencies of engine sound
- Then we normalize the signal after FFT.
- And then we have to take absolute value.



The above figure shows the FFT of the first signal.



- For our neural network first ten absolute values of each car engine sound is the input features.
- This is the function that takes audiodata and process it to gives out data that is suitable for neural network input.

```
function data=carsound(y)
    a=y(:,1);
    af=fft(a);
    z=abs(af);
    figure;
    plot(z)
    an=af/norm(af);
    aa=abs(an);
    figure;
    plot(aa);
    data=zeros(10,10);
    q=1;
    a=10;
    for k=1:10
        data(:,k)=aa(q:a);
        q=q+10;
        a=a+10;
    end
return
```

```
%% calling the function named dataa by the variables
assigned at the beginning of the program.

y1=dataa(yalto);
y2=dataa(yalto800);
y3=dataa(yford);
y4=dataa(ycor);
y5=dataa(yswift);
y6=dataa(ytoyo);
y7=dataa(ytita);


%% calling the function carsound.

x=[y1';y2';y3';y4';y5';y6';y7'];
data=carsound(x);


%% calling the function cardetect.

x=[y1';y2';y3';y4';y5';y6';y7'];
[w1,w2,b2,b3]=cardetect(x);
```

X is a matrix of size 70*10, Each row represents a car.
First 10 row represents altovxi, second 10 row represents alto800 And so on….

c. Neural Network for classification

- The car data X is fed into the neural network,
- Our neural network is made up of two layers,
- 1 hidden and 1 output layer.
- Hidden layer consists of 2 neuron. Output layer consists of 7 neurons.
- Tanh activation function is used.

Step1. Input X to the neural network.

Step2. Randomly initializing weights.

Step3. Finding z1.
Step4. Applying tanh activation function for z1 to get a1.
Step5.Finding z2.
Step6. Applying tanh activation function for z2 to get a2.
Step7. Calculating error.(Output-target)
<u>Back propagation:</u>
Step8. Computing del2

Step9. Computing del1.
Step10.Updating weights and bias.

For each car there will be a target –
alto car target -Y=[1 0 0 0 0 0 0]^T
Alto 800- Y=[0 1 0 0 0 0 0]^T

Our aim is to get output equal to target.
Using back propagation algorithm all weights and bias are updated.

```
function [w1,w2,b2,b3]=cardetect(X)
    u=eye(7);
    y = repelem(u,1,10);
    l2=5; %no. of neuron in layer 2 hidden layer
    n=size(X,2); % no of features of the image=no of pixels of
the image.
    nt=7;
    w1=rand(l2,n); % no of input * output neurons
    b2=zeros(l2,1);
    b3=zeros(nt,1); % for output layer
    w2=rand(nt,l2); % no of input * output neurons
    size(w2)
    for p=1:10000000
        q=ceil(rand(1)*70);
        a0=X(q,:);
        Y=y(:,q);
        z1=w1*a0'+b2;
        z1=z1/norm(z1);
        a1=tanh(z1);


        z2=(w2*a1)+b3;
        %z2=z2/norm(z2);
        a2=tanh(z2);
        del2=(a2-Y).*(1-a2.^2);
        del1=(1-(a1).^2).*(sum((w2.*del2)))';
        w1(1,:)=w1(1,:)-0.1*del1(1)*a0;
        w1(2,:)=w1(2,:)-0.1*del1(2)*a0;
        for k=1:7
            w2(k,:)=w2(k,:)-(0.1*del2(k)*a1');
        end
        b2=b2-0.1*del1;
        b3=b3-0.1*del2; %trained one image
    end
return
```

LITERATURE REVIEW

Paper 1

When a vehicle passes the micro-phone, the recorded acoustic signal shows a peak in energy. The energy contour is smoothed and peaks are automatically located for detection of vehicle sound signal. Mel frequency cepstral coefficients are extracted for detection the regions around detected peaks. The feature vectors are used for training ANN/KNN classifiers.

Paper 2

Sound emitted by vehicles are captured for a two lane undivided road carrying moderate traffic. Simultaneous arrival of different types vehicles, overtaking at the study location, sound of horns, continuous high energy noises on the back ground are the different challenges encountered in the data collection. Different features were explored out of which smoothed log energy was found to be useful for automatic vehicle detection by locating peaks. Mel-frequency ceptral-coefficients extracted from fixed regions around the detected peaks along with the manual vehicle labels are utilised to train an Artificial Neural Network (ANN).

Paper 3

An investigation of acoustic features relating to vehicular traffic on roadways is reported. Computable features that relate to the type of vehicle and state of motion can be useful in monitoring traffic congestion. In the present work, different vehicles, broadly classified into two, three wheelers and heavy vehicle, are studied for their acoustic signatures. A source filter model of engine sound is used to derive suitable features. The performance of formant based features is compared with that of Mel-frequency cepstral coefficients (MFCC) via a k-NN classifier on a manually labelled database of traffic sounds.

Paper 4

This paper attempts to explore the possibility of using sound signatures for vehicle detection and classification purposes. Sound emitted by vehicles are captured for a two-lane undivided road carrying moderate traffic. Simultaneous arrival of different types vehicles, overtaking at the study location, sound of horns, random but identifiable back ground noises, continuous high energy noises on the back ground are the different challenges encountered in the data collection. Different features were explored out of which smoothed log energy was found to be useful for automatic vehicle detection by locating peaks. Mel-frequency ceptral-coefficients extracted from fixed regions around the detected peaks along with the manual vehicle labels are utilized to train an Artificial Neural Network (ANN). The classifier for four broad classes heavy, medium, light and horns were trained. The ANN classifier developed was able to predict categories well.

Paper 5

An investigation of acoustic features relating to vehicular traffic on roadways is reported. Computable features that relate to the type of vehicle and state of motion can be useful in monitoring traffic congestion. In the present work, different vehicles, broadly classified into two, three wheelers and heavy vehicle, are studied for their acoustic signatures. A source filter model of engine sound is used to derive suitable features. The performance of formant-based features is compared with that of Mel-frequency cepstral coefficients (MFCC) via a k-NN classifier on a manually labelled database of traffic sounds

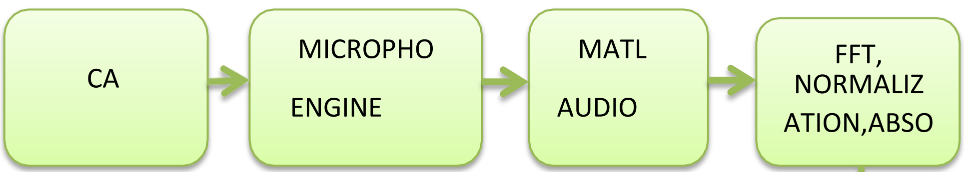
METHODOLOGY

In this section, we would like to describe the methodology we used:

- Neural Network for classification
- The car data X is fed into the neural network,
- Our neural network is made up of two layers,
- 1 hidden and 1 output layer.
- Hidden layer consists of 2 neurons. Output layer consists of 7 neurons.
- Tanh activation function is used.
- Our aim is to get output equal to target.
- Using back propagation algorithm all weights and bias are updated.

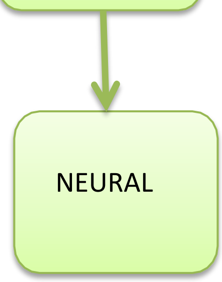
FLOWCHART

.



ALGORITHM

- The car data X is fed into the neural network,
- Our neural network is made up of two layers,



- 1 hidden and 1 output layer.
- Hidden layer consists of 2 neurons.
- Output layer consists of 7 neurons.
- Tanh activation function is used.
    1) Step1. Input X to the neural network
    2) Step2. Randomly initializing weights.
    3) Step3. Finding z1.
    4) Step4. Applying tanh activation function for z1 to get a1.
    5) Step5.Finding z2.
    6) Step6. Applying tanh activation function for z2 to get a2
    7) Step7. Calculating error (Output-target)
    8) Step8. Computing del2 (back propagation)
    9) Step9. Computing del1(back propagation)
    10) Step10.Updating weights and bias.

- For each car there will be a target – alto car target -Y= $[1\ 0\ 0\ 0\ 0\ 0\ 0]^T$
- Alto 800- Y= $[0\ 1\ 0\ 0\ 0\ 0\ 0]^T$

- Our aim is to get output equal to target.
- Using back propagation algorithm all weights and bias are updated.

MATLAB CODE:

The below code is the main MATLAB code from where we will run the program and read the audio signals and call the functions to convert the audio signals into compiler-readable data and another function to find the Fast Fourier Transform the audio signal and the last function to classify the neural network.

```
clc
clear all

%% Reading the audio signals and storing them into different
variables

[yalto]=audioread('C:\Users\User\Desktop\Mishu\Sounds\alto.mp3
');
[yalto800]=audioread('C:\Users\User\Desktop\Mishu\Sounds\alto
800.mp3');
[yford]=audioread('C:\Users\User\Desktop\Mishu\Sounds\ford.mp3
');
[ycor]=audioread('C:\Users\User\Desktop\Mishu\Sounds\Corola
eltis .mp3');
```

```
[yswift]=audioread('C:\Users\User\Desktop\Mishu\Sounds\swift.m
p3');
[ytoyo]=audioread('C:\Users\User\Desktop\Mishu\Sounds\toyoto.m
p3');
[ytita]=audioread('C:\Users\User\Desktop\Mishu\Sounds\Titanium
4.mp3');


%% Plotting the graphs of the input audio signals

[y,Fs]=audioread('C:\Users\User\Desktop\Mishu\Sounds\alto.wav'
);
info=audioinfo('C:\Users\User\Desktop\Mishu\Sounds\alto.wav');
t=0:seconds(1/Fs):seconds(info.Duration);
t=t(1:end-1);
subplot(7,1,1)
plot(t,y);
title('alto');
[y1,Fs]=audioread('C:\Users\User\Desktop\Mishu\Sounds\alto-
800.wav');
info=audioinfo('C:\Users\User\Desktop\Mishu\Sounds\alto-
800.wav');
t=0:seconds(1/Fs):seconds(info.Duration);
t=t(1:end-1);
subplot(7,1,2)
plot(t,y1);
title('alto 800');
[y2,Fs]=audioread('C:\Users\User\Desktop\Mishu\Sounds\ford.wav
');
info=audioinfo('C:\Users\User\Desktop\Mishu\Sounds\ford.wav');
t=0:seconds(1/Fs):seconds(info.Duration);
t=t(1:end-1);
subplot(7,1,3)
plot(t,y2);
title('ford');
[y3,Fs]=audioread('C:\Users\User\Desktop\Mishu\Sounds\Corola-
eltis-.wav');
info=audioinfo('C:\Users\User\Desktop\Mishu\Sounds\Corola-
eltis-.wav');
t=0:seconds(1/Fs):seconds(info.Duration);


t=t(1:end-1);
subplot(7,1,4)
plot(t,y3);
title('Corola eltis');
[y4,Fs]=audioread('C:\Users\User\Desktop\Mishu\Sounds\swift.wa
v');
info=audioinfo('C:\Users\User\Desktop\Mishu\Sounds\swift.wav')
;
t=0:seconds(1/Fs):seconds(info.Duration);
```

```
t=t(1:end-1);
subplot(7,1,5)
plot(t,y4);
title('swift');
[y5,Fs]=audioread('C:\Users\User\Desktop\Mishu\Sounds\toyoto.w
av');
info=audioinfo('C:\Users\User\Desktop\Mishu\Sounds\toyoto.wav'
);
t=0:seconds(1/Fs):seconds(info.Duration);
t=t(1:end-1);
subplot(7,1,6)
plot(t,y5);
title('toyoto');
[y6,Fs]=audioread('C:\Users\User\Desktop\Mishu\Sounds\Titanium
-4.wav');
info=audioinfo('C:\Users\User\Desktop\Mishu\Sounds\Titanium-
4.wav');




t=0:seconds(1/Fs):seconds(info.Duration);
t=t(1:end-1);
subplot(7,1,7)
plot(t,y6);
title('Titanium 4');


%% calling the function named dataa by the variables assigned
at the beginning of the program.

y1=dataa(yalto);
y2=dataa(yalto800);
y3=dataa(yford);
y4=dataa(ycor);
y5=dataa(yswift);
y6=dataa(ytoyo);
y7=dataa(ytita);



%% calling the function carsound.

x=[y1';y2';y3';y4';y5';y6';y7'];
data=carsound(x);


%% calling the function cardetect.

x=[y1';y2';y3';y4';y5';y6';y7'];
%[w1,w2,b2,b3]=cardetect(x);
```

Functions used are:-

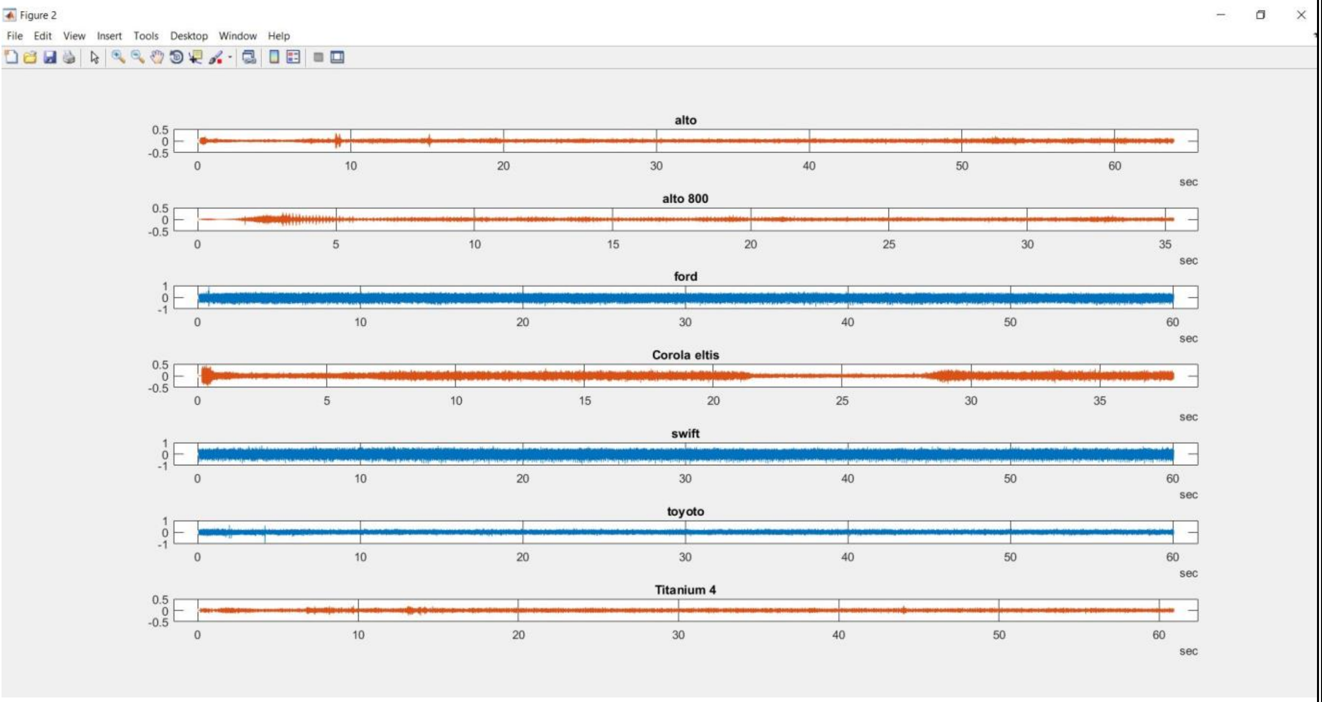
```
function D=dataa(y)
    d=zeros(8000,50);
    a=1;
    q=8000;
    for k=1:50
        d(:,k)=y(a:q);
        a=a+8000;
        q=q+8000;
    end
    df=dct(d);
    dn=df/norm(df);
    da=abs(dn);
    D=da([1:10],:);
return
```

```
function data=carsound(y)
    a=y(:,1);
    af=fft(a);
    z=abs(af);
    figure;
    plot(z)
    an=af/norm(af);
    aa=abs(an);
    figure;
    plot(aa);
    data=zeros(10,10);
    q=1;
    a=10;
    for k=1:10
        data(:,k)=aa(q:a);
        q=q+10;
        a=a+10;
    end
return
```

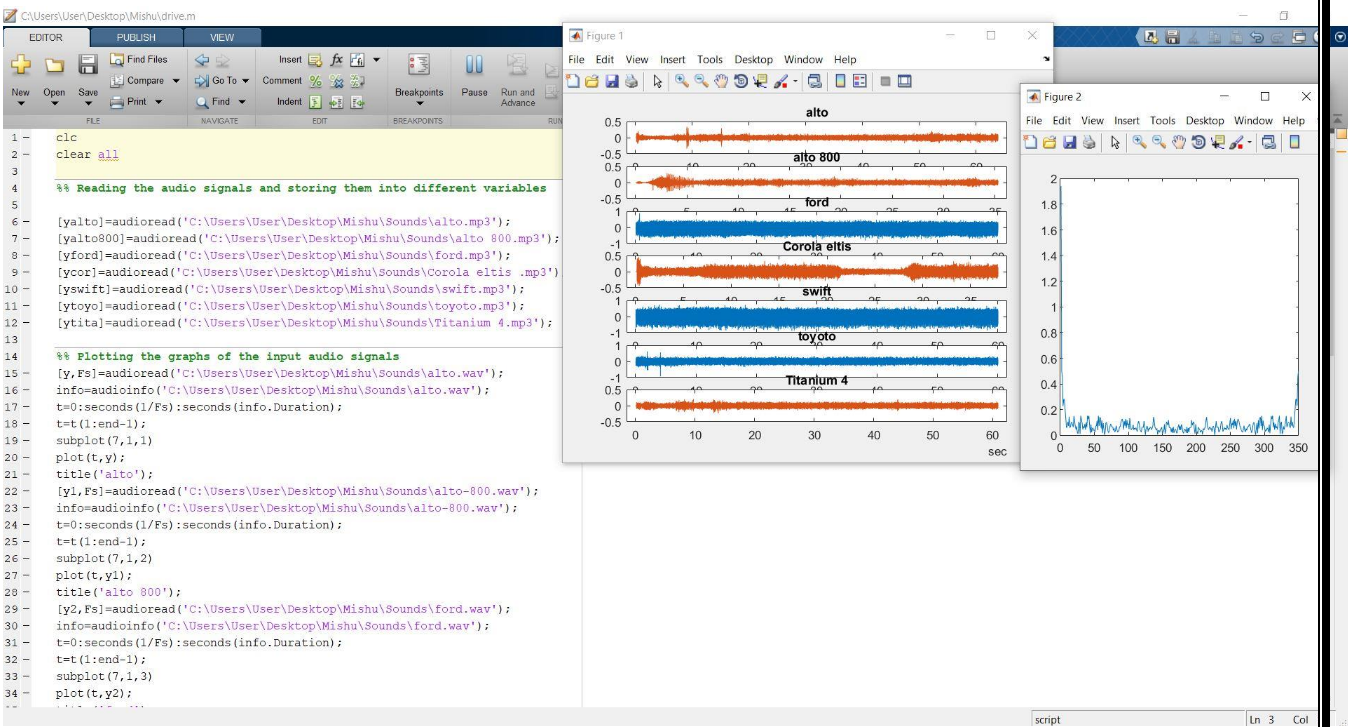
```
function [w1,w2,b2,b3]=cardetect(X)
    u=eye(7);
    y = repelem(u,1,10);
```

```
    l2=5; %no. of neuron in layer 2 hidden layer
    n=size(X,2); % no of features of the image=no of
pixels of the image.
    nt=7;
    w1=rand(l2,n); % no of input * output neurons
    b2=zeros(l2,1);
    b3=zeros(nt,1); % for output layer
    w2=rand(nt,l2); % no of input * output neurons
    size(w2)
    for p=1:10000000
        q=ceil(rand(1)*70);
        a0=X(q,:);
        Y=y(:,q);
        z1=w1*a0'+b2;
        z1=z1/norm(z1);
        a1=tanh(z1);
        z2=(w2*a1)+b3;
        %z2=z2/norm(z2);
        a2=tanh(z2);
        del2=(a2-Y).*(1-a2.^2);
        del1=(1-(a1).^2).*(sum((w2.*del2)))';
        w1(1,:)=w1(1,:)-0.1*del1(1)*a0;
        w1(2,:)=w1(2,:)-0.1*del1(2)*a0;
        for k=1:7
            w2(k,:)=w2(k,:)-(0.1*del2(k)*a1');
        end

       b2=b2-0.1*del1;
        b3=b3-0.1*del2; %trained one image
    end
return
```

OUTPUT:

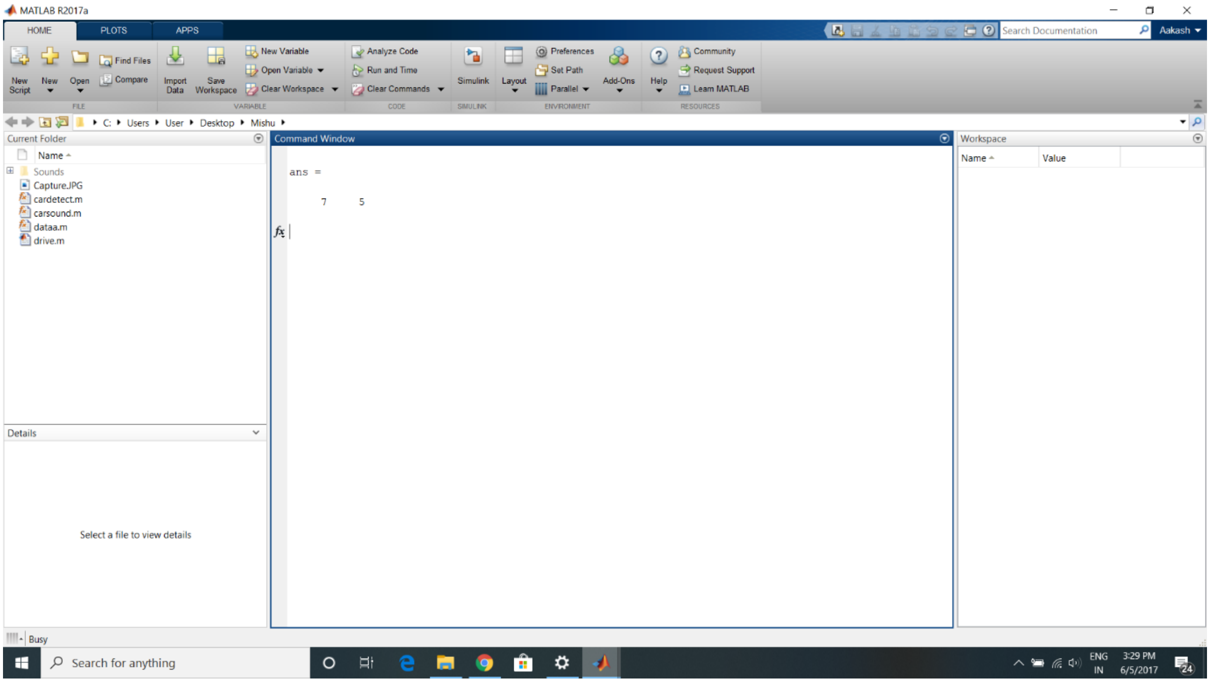


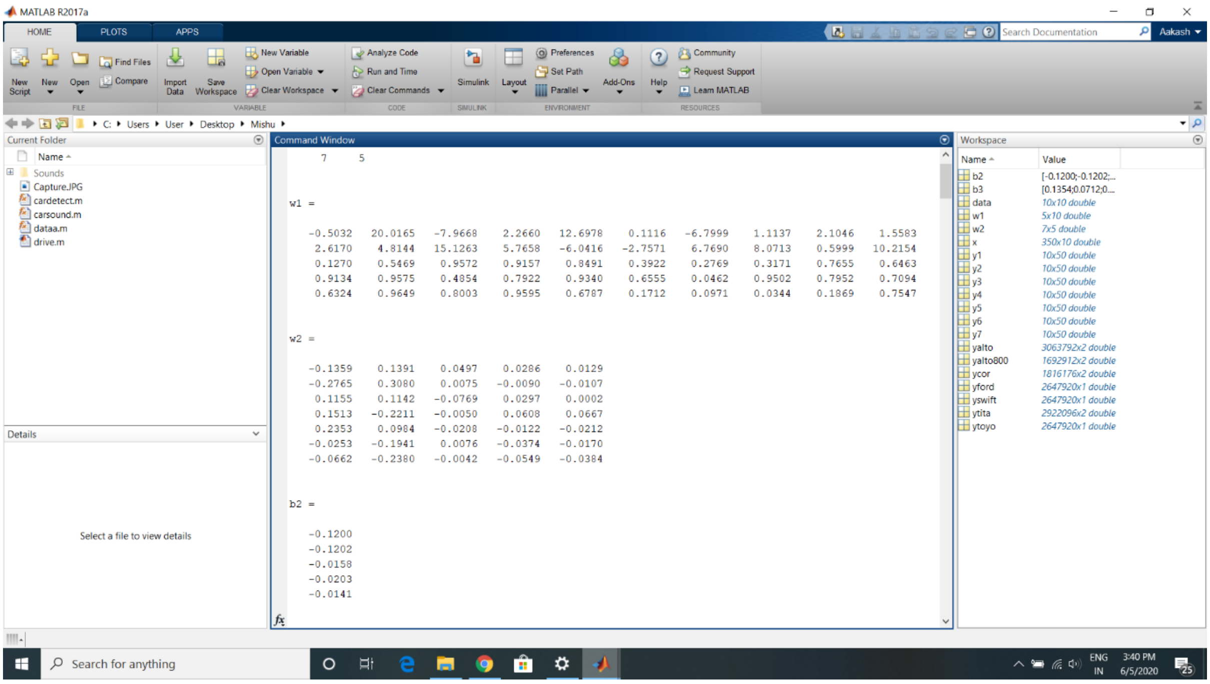
The above figure shows the waveforms of the input audio signals

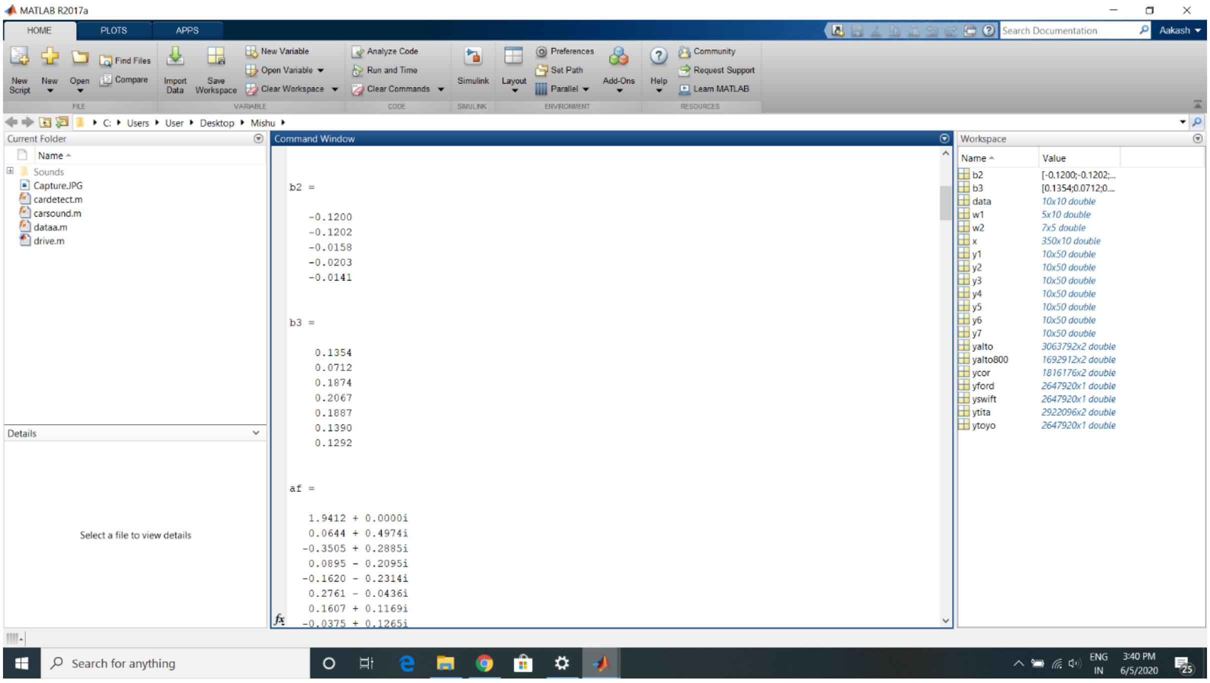


The above figure shows the combined image of our code, input signal and the output signal.

RESULT:


MATLAB R2017a
Command Window
ans =
7 5
Current Folder
Sounds
Capture.JPG
cardetect.m
carsound.m
dataa.m
drive.m
Select a file to view details
Workspace
Busy


MATLAB R2017a
Command Window
7 5
w1 =
-0.5032 20.0165 -7.9668 2.2660 12.6978 0.1116 -6.7999 1.1137 2.1046 1.5583
2.6170 4.8144 15.1263 5.7658 -6.0416 -2.7571 6.7690 8.0713 0.5999 10.2154
0.1270 0.5469 0.9572 0.9157 0.8491 0.3922 0.2769 0.3171 0.7655 0.6463
0.9134 0.9575 0.4854 0.7922 0.9340 0.6555 0.0462 0.9502 0.7952 0.7094
0.6324 0.9649 0.8003 0.9595 0.6787 0.1712 0.0971 0.0344 0.1869 0.7547
w2 =
-0.1359 0.1391 0.0497 0.0286 0.0129
-0.2765 0.3080 0.0075 -0.0090 -0.0107
0.1155 0.1142 -0.0769 0.0297 0.0002
0.1513 -0.2211 -0.0050 0.0608 0.0667
0.2353 0.0984 -0.0208 -0.0122 -0.0212
-0.0253 -0.1941 0.0076 -0.0374 -0.0170
-0.0662 -0.2380 -0.0042 -0.0549 -0.0384
b2 =
-0.1200
-0.1202
-0.0158
-0.0203
-0.0141
Workspace
b2 [-0.1200;-0.1202;...
b3 [0.1354;0.0712;0...
data 10x10 double
w1 5x10 double
w2 7x5 double
x 350x10 double
y1 10x50 double
y2 10x50 double
y3 10x50 double
y4 10x50 double
y5 10x50 double
y6 10x50 double
y7 10x50 double
yalto 3063792x2 double
yalto800 1692912x2 double
ycor 1816176x2 double
yford 2647920x1 double
yswift 2647920x1 double
ytita 2922096x2 double
ytoyo 2647920x1 double



With the help of simulation we find that out of 7 cars, 5 cars have accuracies greater than 55% And that's why it detects only 5 of the car engines.

CONCLUSION:

Our neural network is made up of 2 layers, with 2 hidden units, with this we are getting accuracy (For training data) as shown below.
Accuracy can be increased by having more training data, proper activation function and no. of hidden units.

CAR	ACCURACY
1. Alto=========================>>	93.654 %
2. Alto 800======================>>	95.871 %
3. Ford=========================>>	56.45 %
4. Corolla-Altis===================>>	35.690 %
5. Swift========================>>	87.872 %
6. Toyota=======================>>	55.45 %
7. Titanium=====================>>	93.652 %

FUTURE ENHANCEMENT

1) A Deep-Learning-Based Vehicle Detection Approach for Insufficient and Nighttime Illumination Conditions:

    - Most object detection models cannot achieve satisfactory performance under nighttime and other insufficient illumination conditions, which may be due to the collection of data sets and typical labeling conventions.

    - Study proposing a specifically optimized system based on the Faster region-based CNN model. The system has a processing speed of 16 frames per second for 500 × 375-pixel images, and it achieved a mean average precision (mAP) of 0.8497 in our validation segment involving urban nighttime and extremely inadequate lighting conditions.

    - Proposed methods can achieve high detection performance in various nighttime environments, such as urban nighttime conditions with insufficient illumination, and extremely dark conditions with nearly no lighting. The proposed system outperforms original methods that have an mAP value of approximately 0.2.

2) **Night-Time Vehicle Sensing in Far Infrared Image with Deep Learning**

    - The use of night vision systems in vehicles is becoming increasingly common. Several approaches using infrared sensors have been proposed in the literature to detect vehicles in far infrared (FIR) images. Firstly, vehicle candidates are generated using a constant threshold from the infrared frame. Contours are then generated by using a local adaptive threshold based on maximum distance, which decreases the number of processing regions for classification and reduces the false positive rate.
    - Finally, vehicle candidates are verified using a deep belief network (DBN) based classifier. The detection rate is 93.9% which is achieved on a database of 5000 images and video streams.

3) Use Radar Sensors for Vehicle Detection

    - Radar sensors use Frequency Modulated Continuous Wave (FMCW) radar to reliably detect moving or stationary targets, including cars, trains, trucks and cargo in extreme weather conditions.

    - For example, trains create difficult environments for mechanical and technical equipment to operate properly. Passing trains create excess wind, dirt, debris, and potential for impact. A radar sensor can reliably detect the presence of trains even in harsh conditions. Similarly, a radar sensor can also be used to reliably detect the presence of large trucks at an outdoor loading dock even in extreme weather conditions.

4) Pretraining Convolutional Neural Networks for Image-Based Vehicle Classification

- Vehicle detection and classification are very important for analysis of vehicle behavior in intelligent transportation system, urban computing, etc. Convolutional neural networks (CNNs) has been applied for vehicle classification. In order to achieve a more accurate classification, unrelated background noise can be removed as much as possible based on a trained object detection model.
- In addition, an unsupervised pretraining approach has been introduced to better initialize CNNs parameters to enhance the classification performance. Through the data enhancement on manual labeled images many labeled images can be obtained in each category of motorcycle, transporter, passenger, and others, with samples for training and samples for testing.

REFERENCES

1) George, Jobin & Mary, Leena & S., Riyas. (2013). Vehicle Detection and Classification From Acoustic Signal Using ANN and KNN. 2013 International Conference on Control Communication and Computing, ICCC 2013. 10.1109/ICCC.2013.6731694.
2) George, Jobin & Cyril, Anila & Koshy, Bino & Mary, Leena. (2013). Exploring Sound Signature for Vehicle Detection and Classification Using ANN. International Journal on Soft Computing. 4. 10.5121/ijsc.2013.4203.
3) Nikhil Bhave and Preeti Rao.(2011) Vehicle engine sound analysis applied to traffic congestion estimation. CMMR/FRSM.
4) G. Hochmuth. “On the genetic evolution of a perfect Tic-tac-toe strategy”, pages 75–82. Stanford University Book Store 2003.
5) E. Soedarmadji. Decentralized decision making in the game of tictac-toe. In Proceedings of the 2006 IEEE Symposium on Computational Intelligence and Games, pages 34–38,2005.
6) An article take from the resource provided by the North Western University, USA www.math.northwestern.edu/~mlerma/courses/cs310-04w/notes/dm-dectrees.pdf
7) Thomas Hauk, Michael Buro and Jonathan Schaeffer, “*-Minimax Performance in Backgammon”,
8) Computer and Games Conference, Ramat-Gan 2004, pg 61-66
9) Russell Stuart J., Norvig Peter. “Artificial Intelligence. A modern approach.”. Prentice Hall, 1995.ISBN 0-13-103805-2.

10) An Article from the resource provided by the University of California,Berkeley,USA,http://www.ocf.berkeley.edu/~yosenl/extras/alpha beta/alphabeta.htm

11) An article from the lecture notes provided by the University of California, Irvine, USA, www.ics.uci.edu/~dechter/courses/ics-271/fall-08/lecture-notes/6.games.ppt

12) Sivaraman Sriram, Rajkumar Vijayarangan, Saaisree Raghuraman, Xiaobu Yuan*, Senior Member, IEEE, 2009. Implementing a No-Loss State in the Game of Tic-Tac-Toe using a Customized Decision Tree Algorithm, International Conference on Information and Automation June 22 -25, 2009, Zhuhai/Macau, China

13) Deva Prasad Nayak UIET, Panjab University, Chandigarh, India, Volume 8, No. 7, July – August 2017. GAME OF TIC-TAC-TOE: SIMULATION USING MIN-MAX ALGORITHM, International Journal of Advanced Research in Computer Science.