ELECRICAL ENGINEERING PROJECT

COLLISION WARNING SYSTEM USING 2D-LIDAR

SUBMITTED BY,

GROUP-B

TEAM-5

TEAM MEMBERS:

Poorani Ayswariya PS_21134

Pranav Raj S_21135

P Yagnitha_21138

Sarrvesh N _21146

Sidesh Sundar S_21150

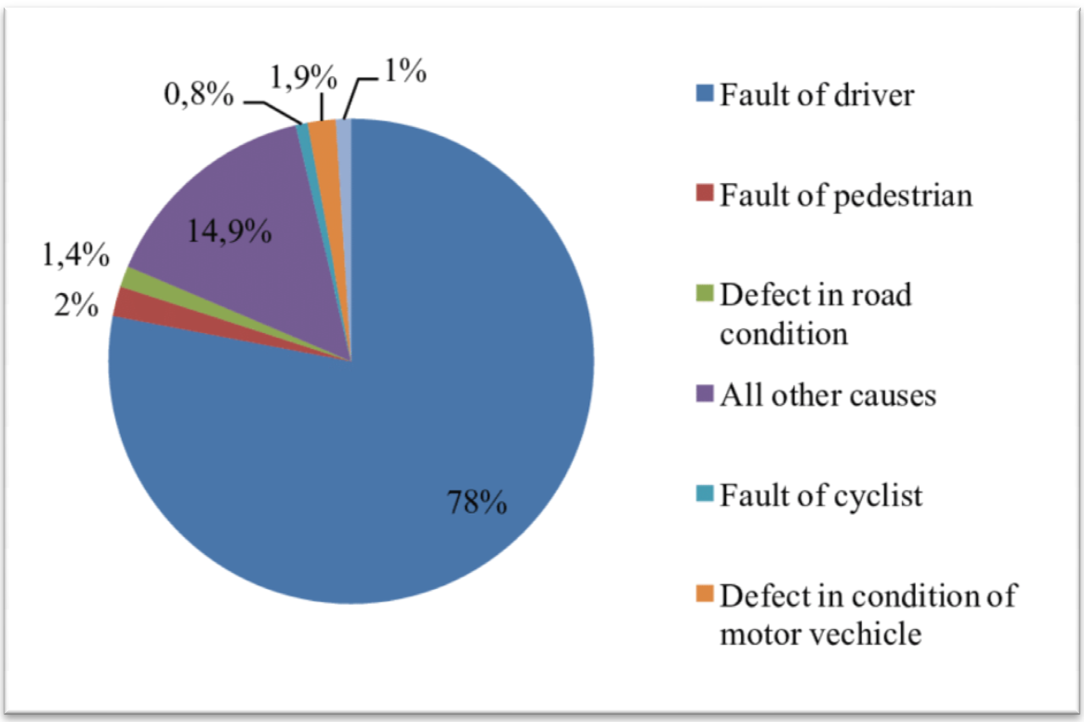
ACKNOWLEDGEMENT:

We would like to express our gratitude to our professors Dr. Utsav Bhowmik sir and Dr. Sitadevi Bharatula mam who gave us this golden opportunity to do the project on the topic "**Collision Warning System using 2D-LIDAR**".

We would like to extend our gratitude to the AIE department of Amrita Vishwa Vidhyapeetham, Chennai who assigned this project to us and helped us to improve our knowledge.

ABSTRACT:

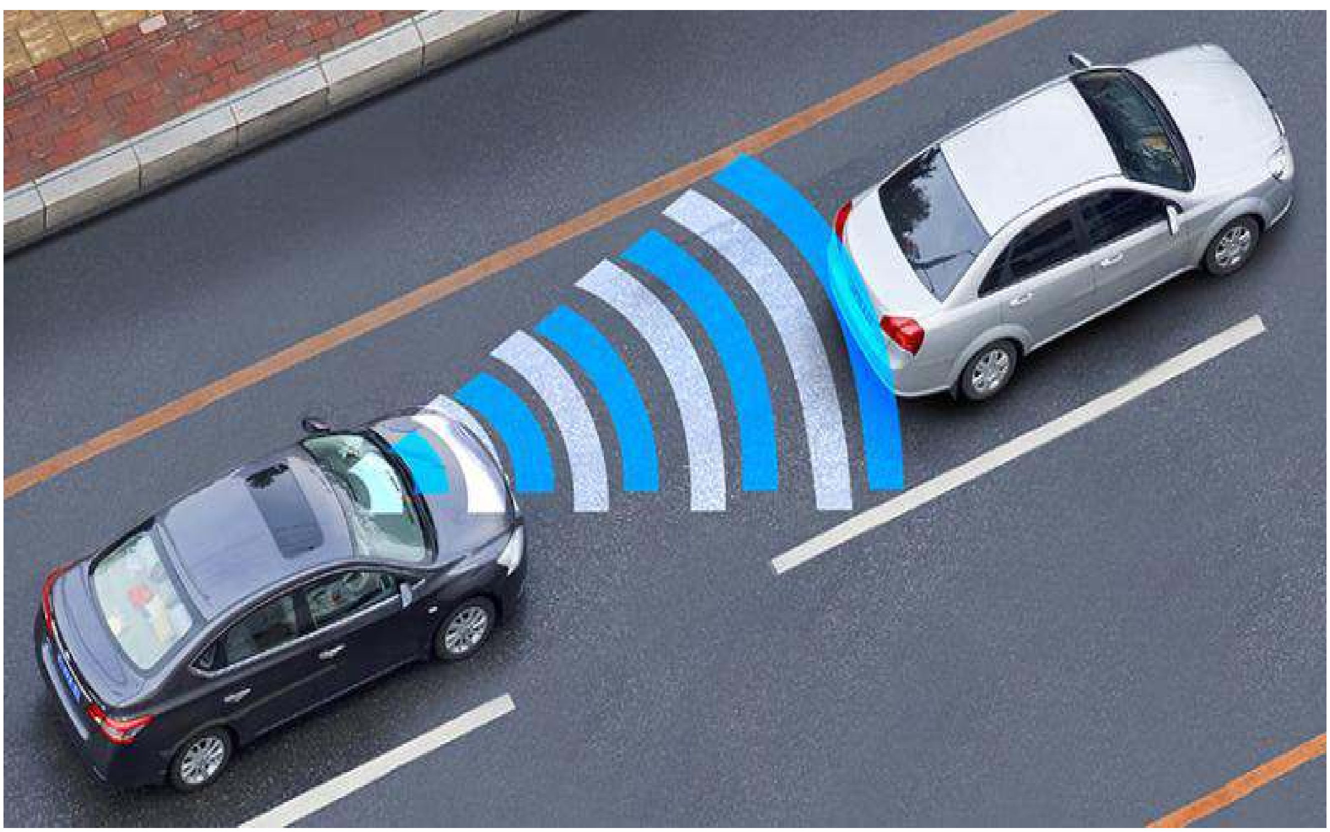
Road safety is one of the major concerns of the whole community at the national level as well as global level. As the number of vehicles increases day by day, accidents are also increasing rapidly. Several studies have been made to improve road safety. Based on the results of research on causes of accidents, the human error comes into play in almost 78% of accidents in India.



These accidents can be due to false estimation of nearby vehicles or less focus of driver .The main goal of this project is designing a collision warning system using 2D-LIDAR in Matlab simulator. This system was briefly explained in the following pages.

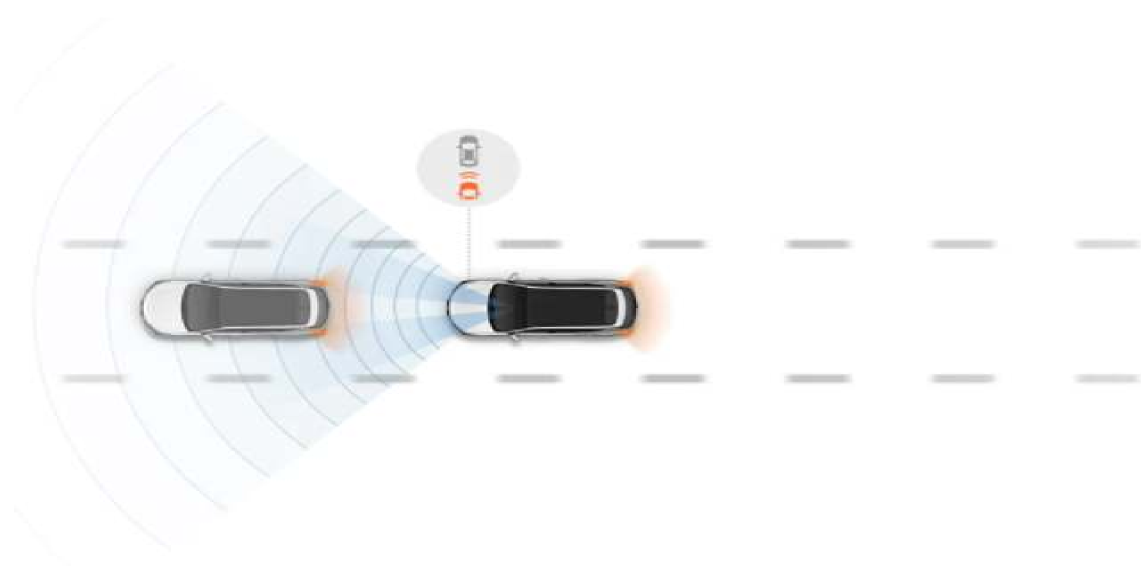
INTRODUCTION:

The collision warning system (CWS) is a safety system which is designed to alert drivers to avoid imminent collisions and reduce the risk of accidents. This system was made using variety of technologies such as lasers, radar, cameras, GPS, lidar and so on. This system was designed in different types such as Forward collision warning system, Blind spot warning, Cross traffic warning etc…



WHAT IS CWS?

CWS stands for Collision Warning System. It is a safety system which finds about the vehicle or obstacle nearby and alerts driver accordingly to reduce the chance of collision and to avoid it.



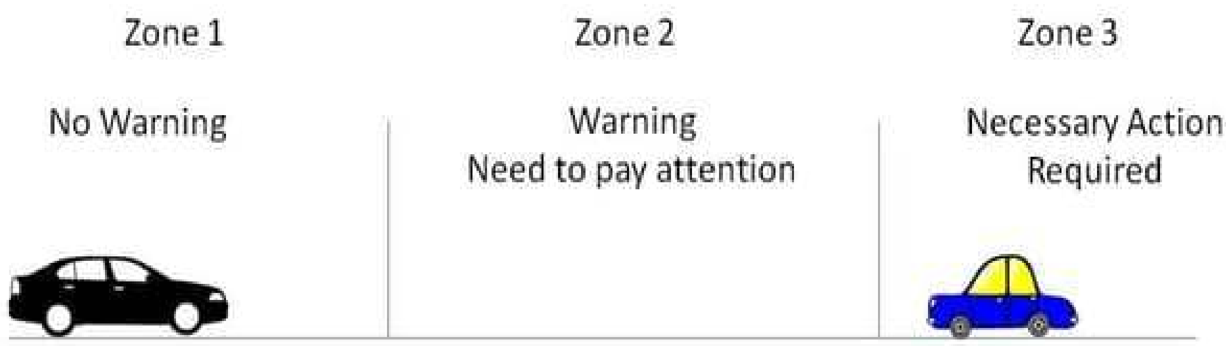
WHY CWS IS REQUIRED?

This system can measure the distance between the driving car and obstacle.

It can calculate the safety distance of the car in three stages such as

- ✓ Collision is imminent
- ✓ High chance of collision
- ✓ Collision can be avoided

If driver does not keep certain minimum distance, it will alert the driver so that collision can be avoided.



WHAT IS 2D-LIDAR?

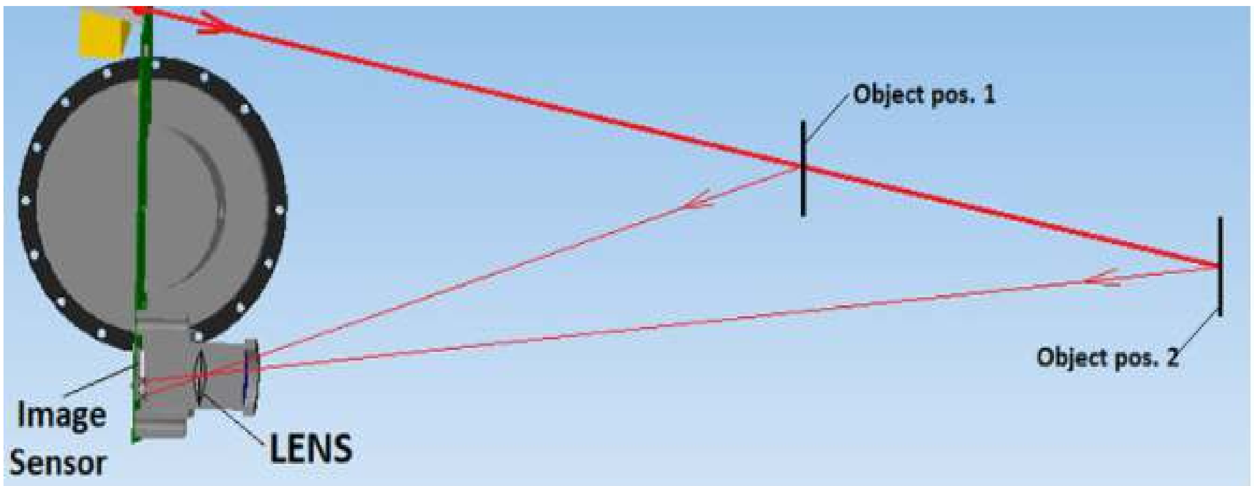
The LIDAR represents **Light Detection and Ranging sensor**. This is a calculation method that allows determining how far obstacles are from the sensor. A LIDAR uses a laser beam for detection, analyzing, and tracking purposes.

For a 2D-LIDAR only **one laser beam** is necessary. Indeed, it will pulse based on a spin movement and collect horizontal distance to the targets to get data on the X and Y axes.

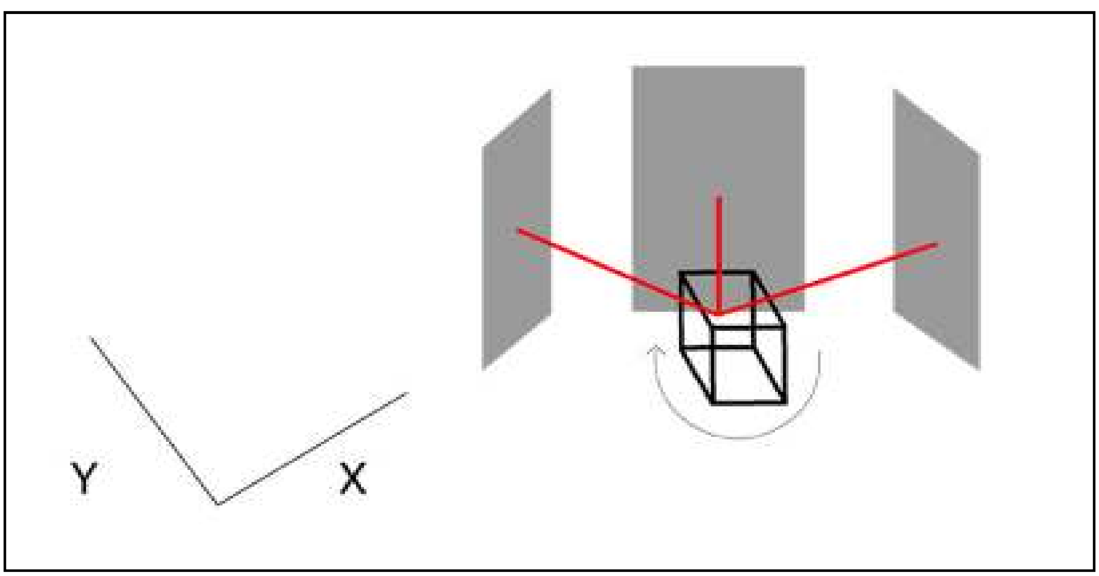
WORKING OF 2D-LIDAR:

Light is emitted from the LIDAR and travels to a target. It will reflect off of its surface and comes back to its source. As the speed of light is a constant value, the LIDAR is able to calculate the distance to the target.

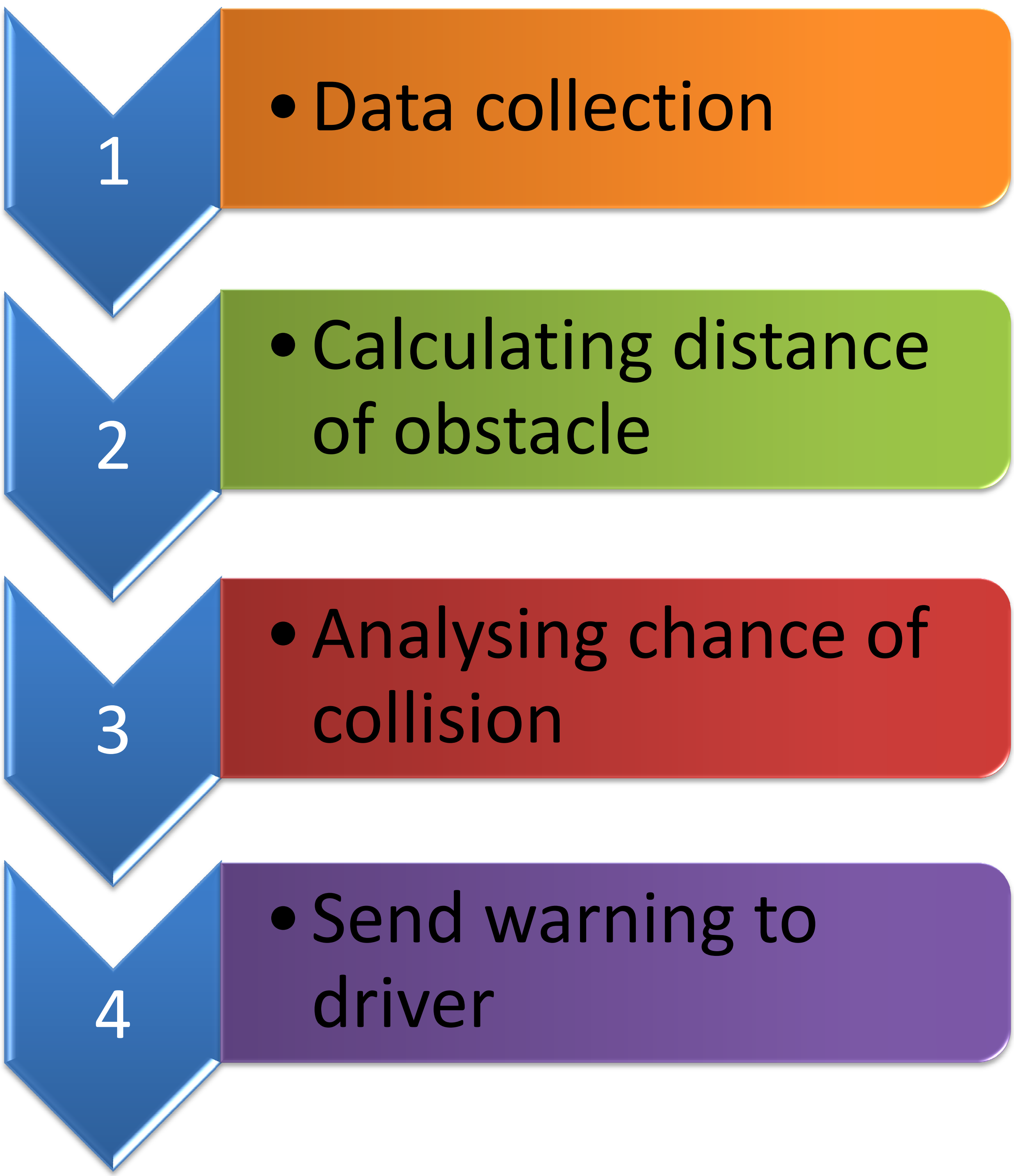
Distance = (Speed of Light x Time of Flight) / 2



SCHEMATIC IMAGE OF 2D-LIDAR:

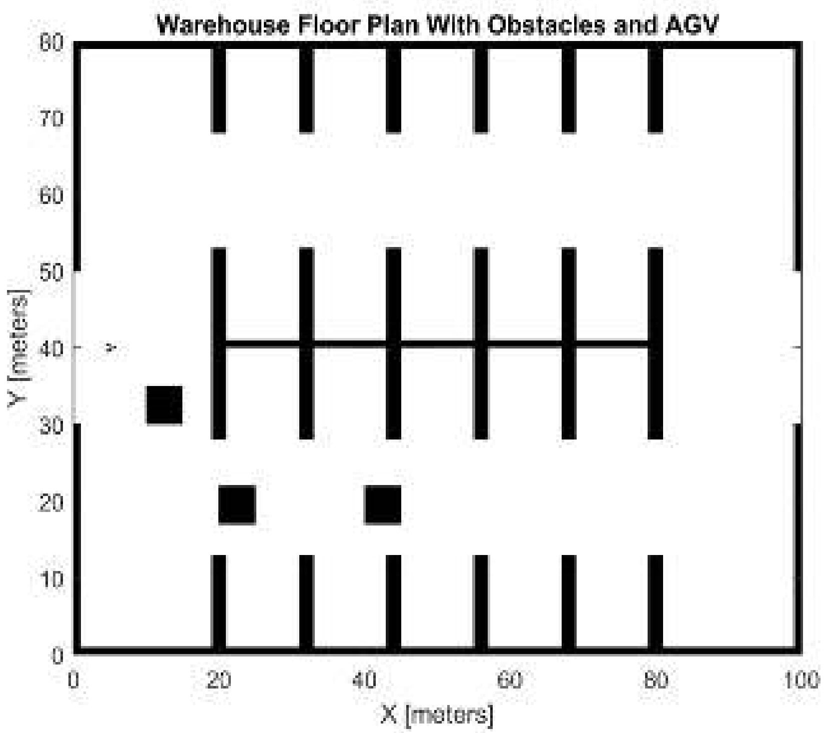


WORK FLOW OF CWS:

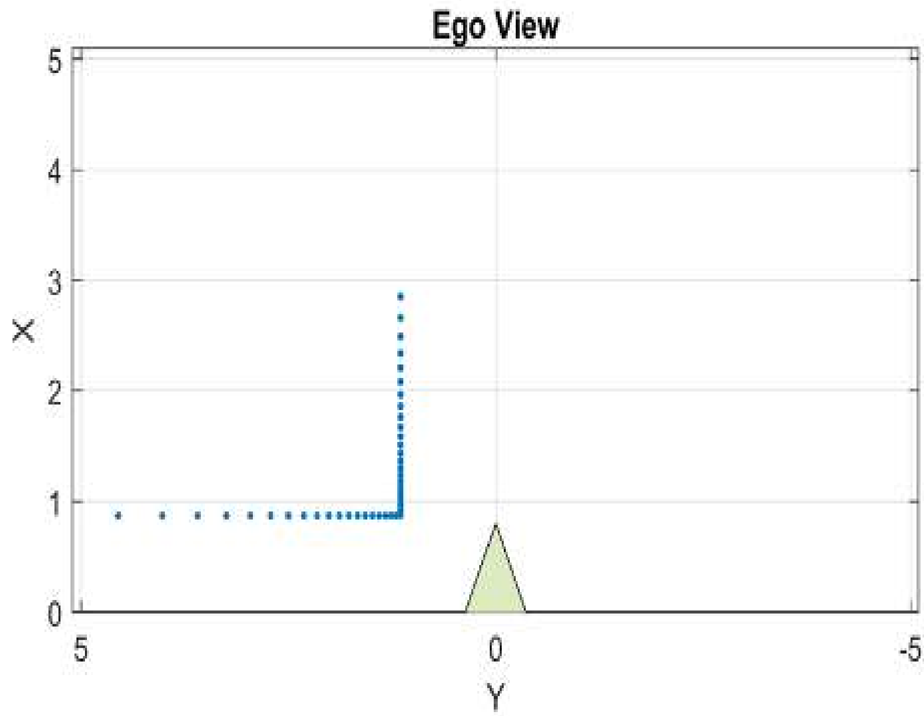


CODING PART OF CWS WITH OUTPUT:

```
map = helperCreateBinaryOccupancyMap;
figure
show(map)
title('Warehouse Floor Plan With Obstacles and AGV')
pose = [5 40 0];
helperPlotRobot(gca, pose);
```

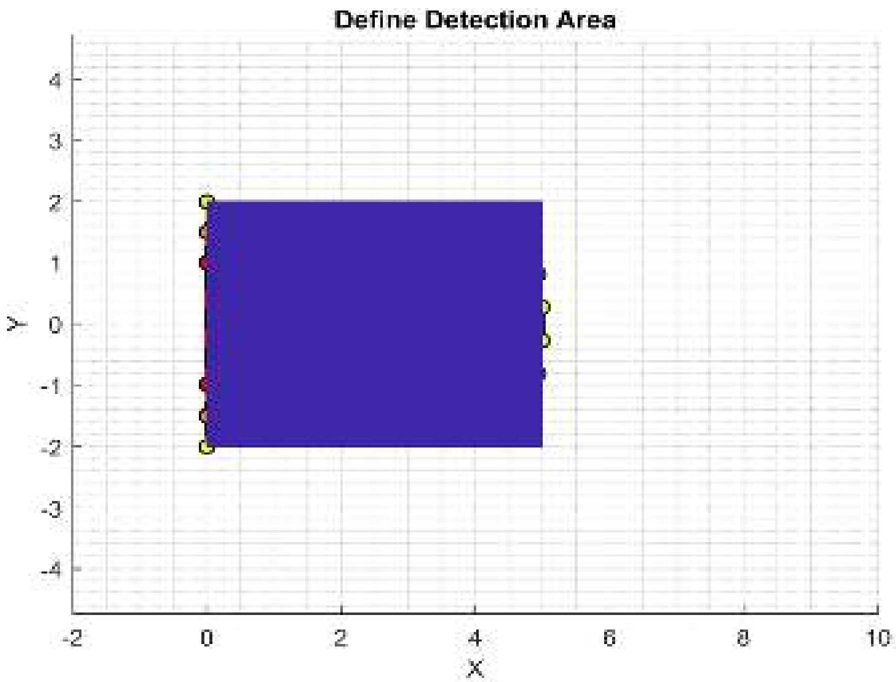



```
lidar = rangeSensor;
lidar.HorizontalAngle = [-pi/2 pi/2];
lidar.Range = [0 5];
load waypoints.mat
traj = waypointsMap;
Vehiclepose = traj(350, :);
[ranges, angles] = lidar(Vehiclepose, map);
scan = lidarScan(ranges, angles);
plot(scan)
title('Ego View')
helperPlotRobot(gca, [0 0 Vehiclepose(3)]);
```




```
display = helperVisualizer;
hRobot = plotBinaryMap(display, map, pose);
figure
detAxes = gca;
title(detAxes,'Define Detection Area')
axis(detAxes, [-2 10 -2 4])
xlabel(detAxes, 'X')
ylabel(detAxes, 'Y')
axis(detAxes, 'equal')
grid(detAxes, 'minor')
t = linspace(-pi/2, pi/2, 30)';
colors = [1 1 1; 1 1 0; 1 0.5 0; 1 0 0];
radius = [5 2 1];
detAreaHandles = repmat(images.roi.Polygon, [3 1]);
pos = [cos(t) sin(t)] * radius(1);
pos = [0 -2; pos(14:17, :); 0 2];
detAreaHandles(1) = drawpolygon(...
    'Parent', detAxes, ...
    'InteractionsAllowed', 'reshape', ...
    'Position', pos, ...
    'StripeColor', 'black', ...
    'Color', colors(2, :));
pos = [cos(t) sin(t)] * radius(2);
pos = [0 -1.5; pos(12:19, :); 0 1.5];
detAreaHandles(2) = drawpolygon(...
    'Parent', detAxes, ...
    'InteractionsAllowed', 'reshape', ...
    'Position', pos, ...
```

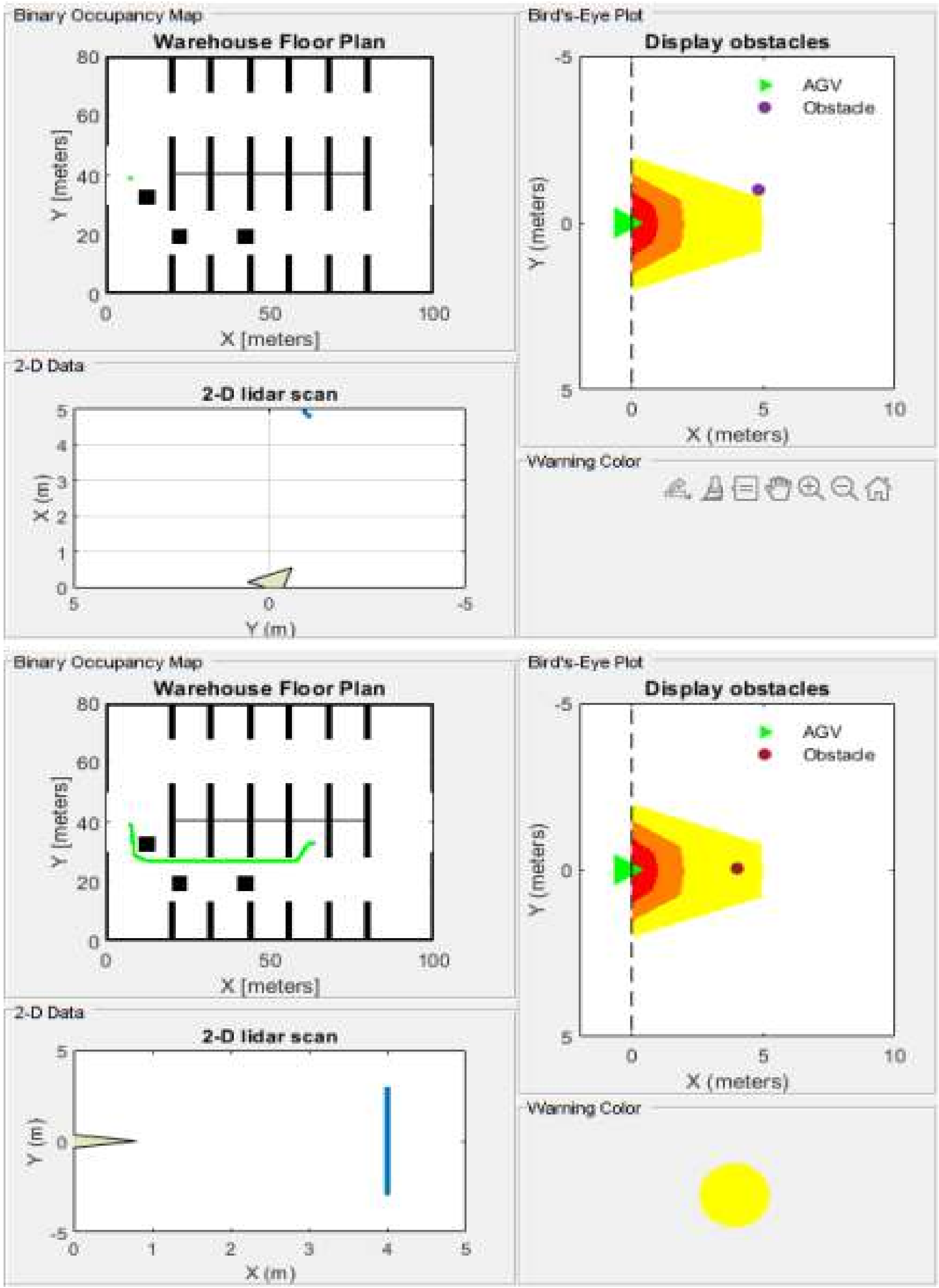
```
    'StripeColor', 'black', ...
    'Color', colors(3, :));
pos = [cos(t) sin(t)] * radius(3);
pos = [0 -1; pos(10:21, :); 0 1];
detAreaHandles(3) = drawpolygon(...
    'Parent', detAxes, ...
    'InteractionsAllowed', 'reshape', ...
    'Position', pos, ...
    'StripeColor', 'black', ...
    'Color', colors(4, :));
axesDet = gca;
[detArea,bbox] = helperSaveDetectionArea(axesDet, detAreaHandles);
```




```
ax3 = getDetectionAreaAxes(display);
h = imagesc(ax3, [bbox(1) (bbox(1) + bbox(3))], ...
    -[bbox(2) (bbox(2) + bbox(4))], ...
    detArea);
colormap(ax3, colors);
plotObstacleDisplay(display)
%Red – Collision is imminent
%Orange – High chance of collision
%Yellow – Apply caution measures
for ij = 27:size(traj, 1)
    currentPose = traj(ij, :);
    [ranges, angles] = lidar(currentPose, map);
    scan = lidarScan(ranges, angles);
    cart = scan.Cartesian;
    cart(:, 3) = 0;
    pc = pointCloud(cart);
```

```
    minDistance = 0.9;
    [labels, numClusters] = pcsegdist(pc, minDistance);
    updateMapDisplay(display, hRobot, currentPose);
    plotLidarScan(display, scan, currentPose(3));
    if exist('sc', 'var')
        delete(sc)
        clear sc
    end
    nearxy = zeros(numClusters, 2);
    maxlevel = -inf;
    for i = 1:numClusters
        c = find(labels == i);
        xy = pc.Location(c, 1:2);
        a = [xy(:, 1) xy(:, 2)] - repmat(bbox([1 2]), [size(xy, 1) 1]);
        b = repmat(bbox([3 4]), [size(xy, 1) 1]);
        xy_org = a./b;
        idx = floor(xy_org.*repmat([size(detArea, 2) size(detArea, 1)],[size(xy_org, 1), 1]));
        validIdx = 1 <= idx(:, 1) & 1 <= idx(:, 2) & ...
            idx(:, 1) <= size(detArea, 2) & idx(:, 2) <= size(detArea, 1);
        cols = idx(validIdx, 1);
        rows = idx(validIdx, 2);
        levels = double(detArea(sub2ind(size(detArea), rows, cols)));

        if ~isempty(levels)
            level = max(levels);
            maxlevel = max(maxlevel, level);
            xyInds = find(validIdx);
            xyInds = xyInds(levels == level);
            nearxy(i, :) = helperNearObstacles(xy(xyInds, :));
        else
            nearxy(i, :) = helperNearObstacles(xy);
        end
    end
    switch maxlevel
        case 3
            circleDisplay(display, colors(4, :))
        case 2
            circleDisplay(display, colors(3, :))
        case 1
            circleDisplay(display, colors(2, :))
        otherwise
            circleDisplay(display, [])
    end
    for i = 1:numClusters
        sc(i, :) = displayObstacles(display, nearxy(i, :));
    end
    updateDisplay(display)
    pause(0.01)
end
```


Binary Occupancy Map
Warehouse Floor Plan
Y [meters]
X [meters]
Bird's-Eye Plot
Display obstacles
AGV
Obstacle
Y (meters)
X (meters)
2-D Data
2-D lidar scan
X (m)
Y (m)
Warning Color
Binary Occupancy Map
Warehouse Floor Plan
Y [meters]
X [meters]
Bird's-Eye Plot
Display obstacles
AGV
Obstacle
Y (meters)
X (meters)
2-D Data
2-D lidar scan
Y (m)
X (m)
Warning Color

CONCLUSION:

In India, every year, about 3% to 5% of country's GDP was invested in road accidents. Collision warning systems will make a major change in road safety-related issues once it was implemented in a large number of vehicles. It would increase the situation awareness of drivers by eliminating or decreasing human errors. So, major concern about this road safety issues can be solved to the certain level.