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

WINTER SEMESTER 2021-2022

DENSITY BASED TRAFFIC CONTROL SYSTEM

Course Code – ECE2010

Course – CONTROL SYSTEMS

Slot – G1 + TG1

Faculty Name – Dr. BAGUBALI A

Student Name – SRISHTI CHOPRA

Registration No. – 20BML0046

TEAM

20BML0014 – KEERTHANA B

20BML0046 – SRISHTI CHOPRA

ABSTRACT

Traffic congestion is a serious problem in India. Due to the existing inefficient traffic control system, citizens spend long hours waiting in roadways. It also takes up a lot of manual labor, i.e., the traffic policemen, to clear and monitor the traffic every now and then. The added congestion to this issue is that even the emergency vehicles like an ambulance or a fire engine must have to wait in these lanes for the traffic to clear. This situation clearly needs to be handled in a more professional and sophisticated manner. Managing the traffic by using a density-based control system is a lot more efficient than the existing open loop control system that works just based upon the traditional timer method. The project is finalized and implemented using an ATMEGA8 microcontroller which supports both density-based traffic control, as well as emergency detection and clearance. The microcontroller stores the vehicles' count in its memory and based on different vehicles count, the microcontroller takes the decision and updates the traffic light. So, the light stays green for that particular lane with high vehicle count, until the traffic turns back normal. The system is also experimented using deep learning via YOLO algorithm. YOLO can be implemented on embedded controllers using Transfer Learning technique.

INTRODUCTION

Due to rapid increase of automobiles and large time delays between traffic lights. Therefore, we will use a density-based traffic control system. It would be imperative that a new design be developed so as to enhance the flow and control of traffic using the density of vehicles on a particular lane by automatically assigning the left-over time in an idle lane to the lane with more traffic density, hence, reducing traffic congestion in those lanes. The proposed project is economically efficient as it reduces the traffic delays and enhances petrol and diesel usage. This also helps in improving the sustainable development by reducing the air pollution caused by long-time vehicle usage. This plan can be implemented all across the globe to maximise its benefits. The emergency detection and override can indeed save lives and improve human lifeline. The modernization and further technological enhancements are just around the corner and this traffic control system would surely be a major contributing factor to it. The reduction in waiting time also improves the travellers' mood and enhances their mental health. Thus, this closed loop system can turn tables and make a very efficient replacement for the existing open loop traffic control system. This system is implemented using 2 ways – microcontroller-based and Deep learning-based methods.

- ***State of art***

  Microcontroller based project: The technology used in this system is a microcontroller based closed loop control system. This is one among the latest and finest techs in field and can be easily implemented nationwide without much difficulty. The Proposed project has an embedded C code that works as the heart of the microcontroller. The IR and Acoustic sensors work remarkably well in detection the vehicles and ambulance respectively.

  Machine learning based project: The Camera captures and sends images after regular short intervals to our system. The system determines further the number of cars in the line and hence computes its relative density with respect to other lanes. Time allotment module takes input (as traffic density) from this system and determines an optimized and efficient time slot. This value is then triggered by the microprocessor to the respective Traffic Lights.

RELATED WORKS

➢ ***Literature Review***

SL. No.	Authors	Title	Year of publication	Inference
1.	1. Sai Surya Prakash Moka 2. Sai Manikanta Pilla 3. S. Radhika	**Real Time Density Based Traffic Surveillance System Integrated with Acoustic Based Emergency Vehicle Detection.**	2020	- Digital image processing technique is used to analyse traffic density. - Signal Processing is used to detect Emergency vehicles. - sound acquisition with noise removal using LMS (least mean square) filter. [1]
2.	1. Dr. P. Amudhavalli 2. M. Monica 3. R. Sandhya Shree	**Effective Traffic Management System for Emergency Vehicles.**	2017	- Using Radio signals with a buzzer to indicate presence of ambulance in the Lane. - Provides speed control to vehicles while in accident prone zones. [2]
3.	1. Liang-Tay Lin 2. Hung-Jen Huang 3. Jim-Min Lin 4. Fongray Frank Young	**A New Intelligent Traffic Control System for Taiwan.**	2009	- Based on MEMS technology. - A non-intrusive vehicle detection using image processing technique. - Uses AI to adapt traffic dynamics and renew traffic control strategies. [3]
4.	1. S. Neelakandan 2. M. A. Berlin 3. Sandesh Tripathi 4. V. Brindha Devi 5. Indu Bhardwaj 6. N. Arulkumar	**IoT-based traffic prediction and traffic signal control system for smart city.**	2021	- Traffic prediction done using OWENN algorithm. - Traffic, weather and direction are the inputs to predict the upcoming traffic. - The Plant in this system is an Intel 80,286 microprocessor. [4]
5.	1. Uthara E. Prakash 2. Athira Thankappan 3. Vishnupriya K. T 4. Arun A. Balakrishnan	**Density Based Traffic Control System Using Image Processing.**	2018	- Image acquired using webcam in each Traffic prone zone. The vehicle count is done based on these images. - Matlab is used for image processing and signal timing allotment. [5]
6.	1. Pavel Pokorný	**Determining Traffic Levels in Cities Using Google Maps.**	2017	- Adds a transport layer to the google maps configuration.

				- Then converts these maps to HTML5 canvas to create individual canvases. - An algorithm is created to read the pixels at any coordinates and evaluate their colours. - The numeric output of the system gives the traffic levels. - Programming language used is JavaScript. [6]
7.	1. Hiroaki Nakanishi 2. Toru Namerikawa	**Optimal Traffic Signal Control for Alleviation of Congestion based on Traffic Density Prediction by Model Predictive Control**	2016	- Achievement of more effective alleviation of traffic jam by applying MPC to traffic flow model calculating optimal green time considering future traffic flow in real time. - Controls traffic flow in each link without exceeding over x max z of link 13.
8.	1. Anurag Kanungo 2. Ayush Sharma 3. Chetan Singla	**Smart Traffic Lights Switching and Traffic Density Calculation using Video Processing**	2014	- This system consists of video cameras on the traffic junctions for each side as if it is a four-way junction - Hardcoded versus intelligent traffic control - Junction Traffic Simulation - Switching Algorithm; Traffic Density Calculation - Video and Image Processing
9.	1. Kazuhito Komada 2. Takashi Nagatani	**Modeling and Simulation for Vehicular Traffic in City Network Controlled by Signals**	2009	- The vehicle moves through the series of signals on a path selected by the driver. - The selected path is one of the straight, zigzag, and random paths in city traffic network - Non-linear map model

10.	1. Shyam Shankaran R 2. Logesh Rajendran	**Real-Time Adaptive Traffic Control System For Smart Cities**	2021	- Intelligent transport system architecture - Using image processing, the proposed system adapts the traffic signal timer according to the random traffic density - ATCS frameworks proves beneficial.
11.	1. Sk Riyazhussain, 2. Riyazhussain, 3. C.R.S. Lokesh 4. P.Vamsikrishna 5. Goli Rohan	**Raspberry Pi Controlled Traffic Density Monitoring System**	2016	- Use of Rasperry Pi - Computer Vision is used for system designing - System Design Open CV used rather than MATLAB. - TDM could be used in public places.
12.	1. Dhruv Patel 2. Yogesh Rohilla	**Infrared sensor based Self-Adaptive Traffic Signal System using Arduino Board**	2020	- Measures density of vehicles using infrared sensors at each of the four roads at the intersection. - Self-adaptive traffic light system - Core of project is Arduino Mega Development Board

PROBLEM STATEMENT

Waiting for long hours, in a never-ending line of vehicles, while we have to reach our destination right on time is surely one of the most frustrating situations. The existing traffic control system works on a static basis. Meaning, no matter how many vehicles are present in each lane, it still sets the same timer for all lanes. This kind of a system is not just tiring for the travellers but it is also highly inefficient in terms of petrol and diesel consumption. Moreover, whenever there is any emergency vehicle that must pass through, it requires physical clearance of the route by traffic police. The long hours of running automobiles on roads increases pollution rate and degrades the country's economy. Statistics show that the developed countries loose around billions of dollars annually due to such traffic congestions.

- ***Existing system***

Open loop control system – This is a traditional timer-based system. The signals work with the same amount of delay, irrespective of the density / emergency vehicle in each lane. This results in long hours of delays and having to wait under traffic. Even an ambulance has to wait the same duration in the signal. This is highly inefficient and causes huge economic and environmental expense every day.

PROPOSED SOLUTION

METHOD 1:

In this system we will use IR (infrared) sensors to measure the traffic density. We will arrange one IR sensor for each road. We interface these sensors with the ATmega8 controller.

The circuit will consist of 4 IR sensors, ATmega8 controller and 4 traffic lights.

The IR transmitter always emits IR rays from it. These IR sensors are placed in such a way that when we place an obstacle in front of this IR pair, the IR receiver should be able to receive the IR rays. When we power the IR sensors the transmitter rays hit the object and reflect. We chose the sensors based on accuracy, range, calibration, and affordability. Although IR sensors are usually disturbed by noise in surroundings, we have used this as they are readily available, cheap, easy to interface and are highly reliable.

***Emergency/priority-based vehicles*: **For vehicles which need priority like ambulances, fire fighters we need to turn that lane open immediately. We do this using an ultrasonic sensor by which these vehicles can be recognized based on their siren.

We will be using Proteus for simulation:

- For normal traffic system, we are using a time interval for 3 seconds for each second.
- For simulation purposes we will use a switch which will act as an IR sensor to show if there is traffic in any lane and hence the microcontroller can change the signal light accordingly.
- 4 other switches will be used for priority-based vehicles. Each switch for each lane.

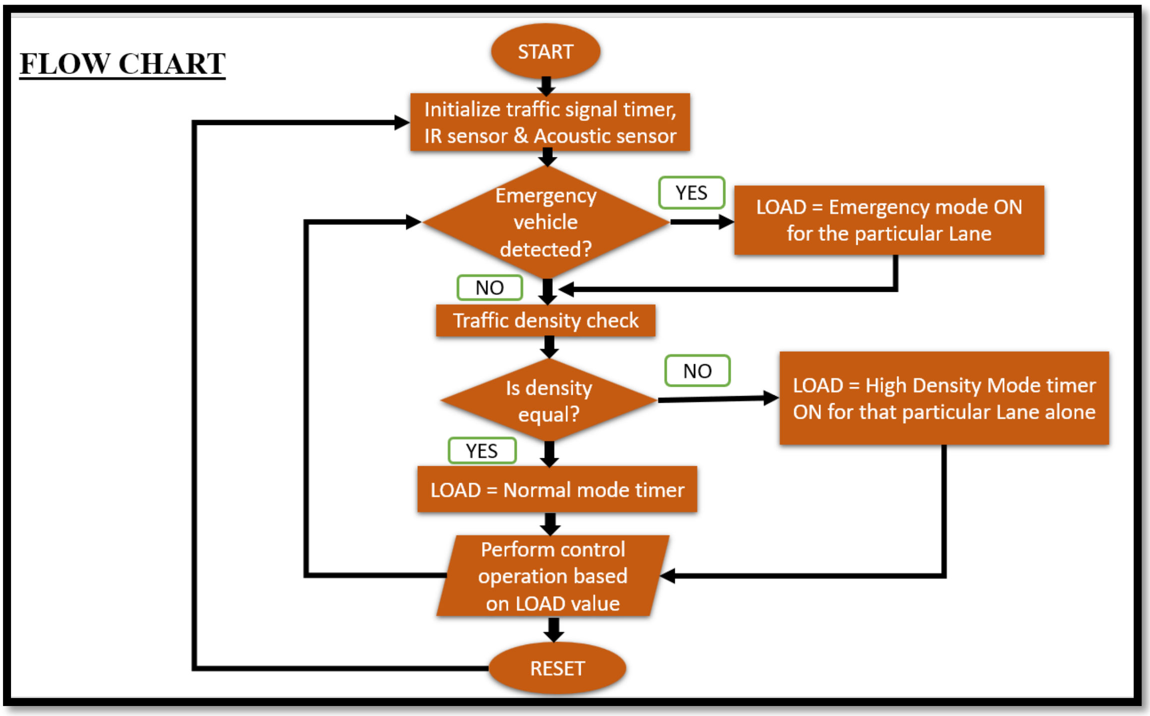
METHOD 2:

The YOLO models are end-to-end deep learning models and are preferred due to their detection speed and accuracy. A YOLO network has a structure similar to that of a normal CNN, consisting of several convolutional and max-pooling layers, ending with two fully connected layers, YOLO follows a very unique approach where it employs a single neural network to look at the entire image just once, because of this it has various advantages over classifier-based systems. To predict each bounding box, it uses features from the entire image.

YOLO model divides the input image into SS grids. Each grid cell predicts k hounding boxes and confidence values of bounding boxes, as well as C conditional class probabilities. The confidence score means how confident the model is about the object in the box and also the accuracy of the box it's predicting. Each hounding box makes 5 predictions: x. y, w, h, and confidence. The (x, y) coordinates represent the centre offset of the box w.r.t boundaries of the grid cell. The wand hare width and height, predicted relative to the whole image. Confidence (cfd) prediction is defined as P When a grid cell contains a portion of a ground truth box the value of Pr (Object) is 1 else it is 0. IOU stands for the intersection over the union between the predicted bounding box and the ground truth box. With the help of these predictions, we derive the class specific confidence score of individual bounding boxes and finally select the bounding boxes with high confidence scores in each grid cell in order to make global predictions of a traffic sign in the image.

METHODOLOGY

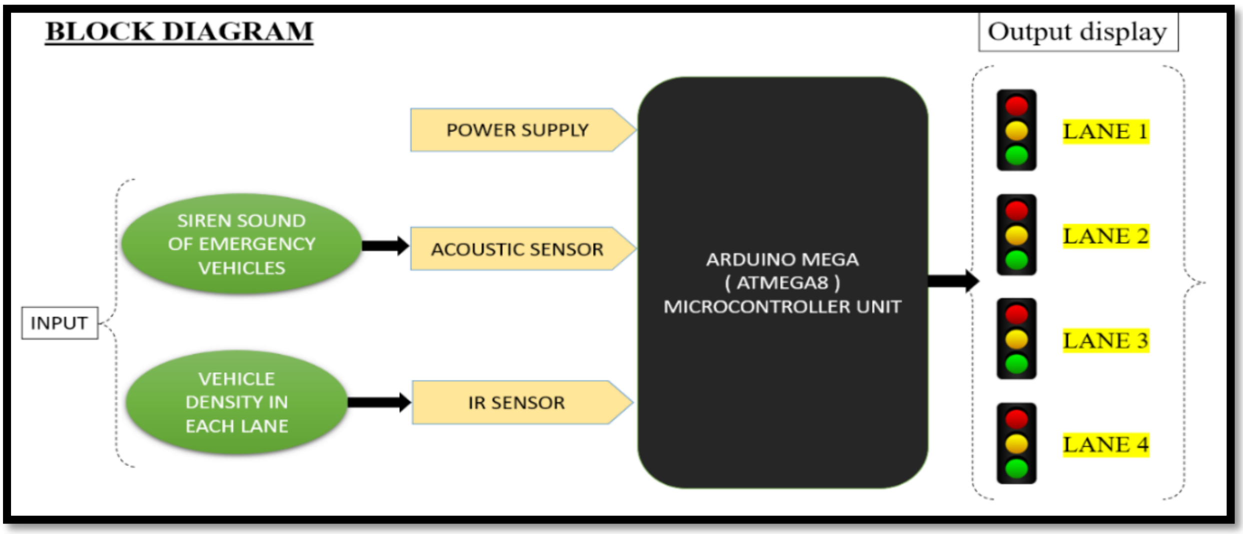
➢ ***Algorithm & Mathematical Model***



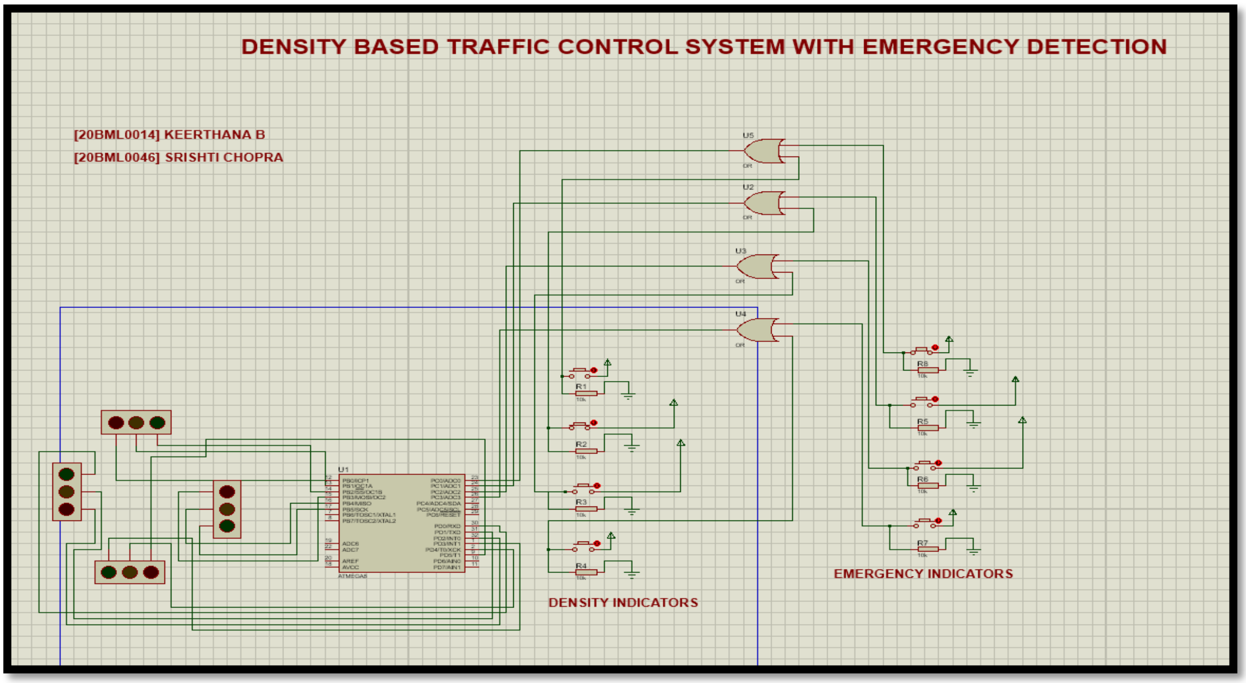
- Firstly, an acoustic sensor is used to detect the siren sounds of incoming emergency vehicles like an ambulance or a fire engine. The siren sounds have a minimum frequency of 725 Hz. So, the acoustic sensor produces the output whenever it senses any sound above 800Hz.
- The second service is adaptive signal timer. An IR sensor in placed in all 4 lanes of an interconnection. It senses the density and no. of vehicles coming into each lane. If it senses that the density of vehicles in any one of the lanes is comparatively higher than the others, the green signal timer for that lane is delayed a bit longer until the traffic is been cleared out.
- In the cases where the traffic density seems to be fairly equal among all the lanes, then the normal timer system is activated to work.
- This system works as a closed loop control system as the system operations depends on density of roads and there are no manual input settings involved.

METHOD 1 - MICROCONTROLLER

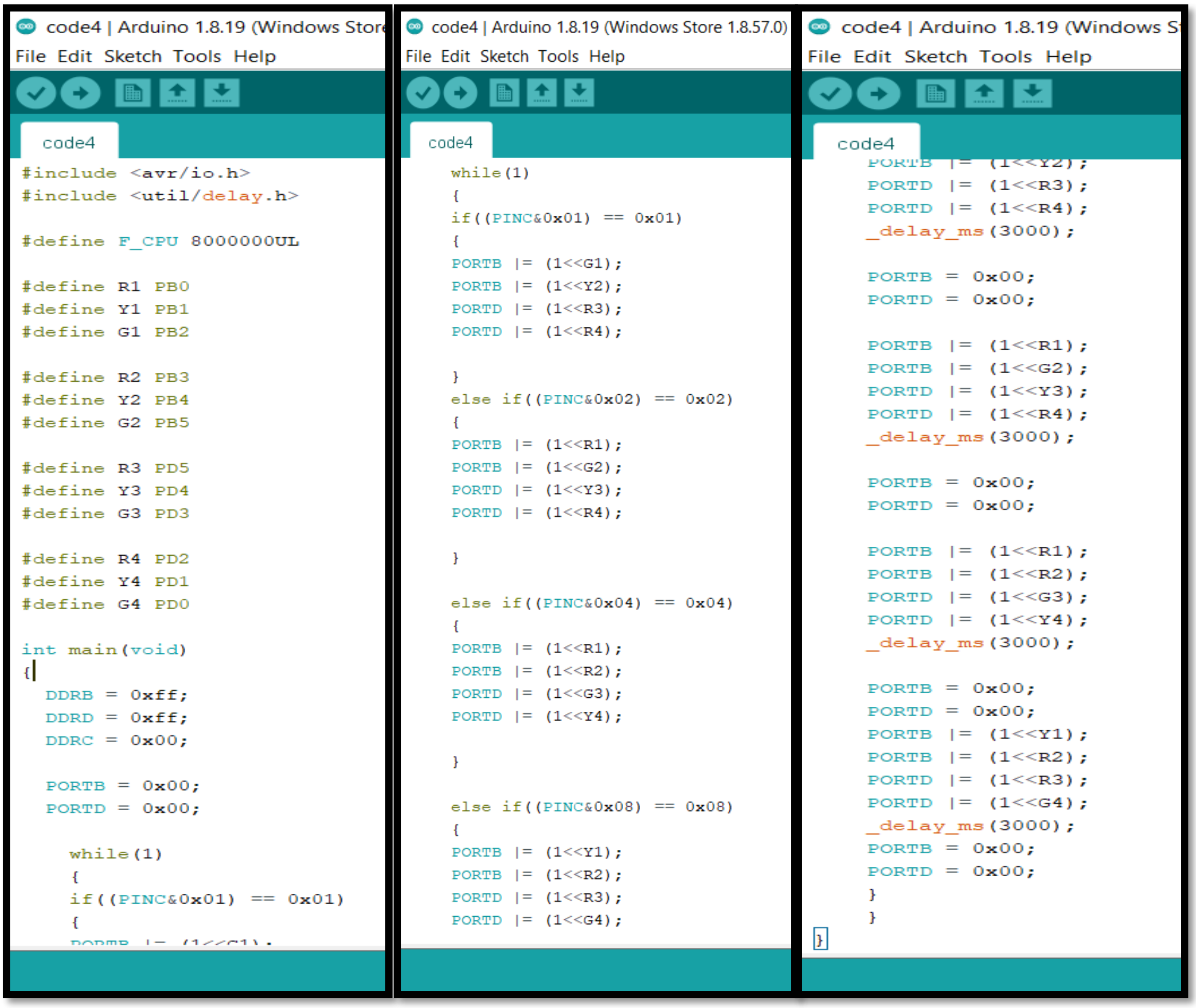
- ***Architecture***



- ***Circuit Block:***



➢ *Microcontroller Code:*




```
#include <avr/io.h>
#include <util/delay.h>

#define F_CPU 8000000UL

#define R1 PB0
#define Y1 PB1
#define G1 PB2

#define R2 PB3
#define Y2 PB4
#define G2 PB5

#define R3 PD5
#define Y3 PD4
#define G3 PD3

#define R4 PD2
#define Y4 PD1
#define G4 PD0

int main(void)
{
  DDRB = 0xff;
  DDRD = 0xff;
  DDRC = 0x00;

  PORTB = 0x00;
  PORTD = 0x00;

    while(1)
    {
    if((PINC&0x01) == 0x01)
    {
    PORTB |= (1<<G1);
```

```
    while(1)
    {
    if((PINC&0x01) == 0x01)
    {
    PORTB |= (1<<G1);
    PORTB |= (1<<Y2);
    PORTD |= (1<<R3);
    PORTD |= (1<<R4);

    }
    else if((PINC&0x02) == 0x02)
    {
    PORTB |= (1<<R1);
    PORTB |= (1<<G2);
    PORTD |= (1<<Y3);
    PORTD |= (1<<R4);

    }

    else if((PINC&0x04) == 0x04)
    {
    PORTB |= (1<<R1);
    PORTB |= (1<<R2);
    PORTD |= (1<<G3);
    PORTD |= (1<<Y4);

    }

    else if((PINC&0x08) == 0x08)
    {
    PORTB |= (1<<Y1);
    PORTB |= (1<<R2);
    PORTD |= (1<<R3);
    PORTD |= (1<<G4);
```

```
    PORTB |= (1<<Y2);
    PORTD |= (1<<R3);
    PORTD |= (1<<R4);
    _delay_ms(3000);

    PORTB = 0x00;
    PORTD = 0x00;

    PORTB |= (1<<R1);
    PORTB |= (1<<G2);
    PORTD |= (1<<Y3);
    PORTD |= (1<<R4);
    _delay_ms(3000);

    PORTB = 0x00;
    PORTD = 0x00;

    PORTB |= (1<<R1);
    PORTB |= (1<<R2);
    PORTD |= (1<<G3);
    PORTD |= (1<<Y4);
    _delay_ms(3000);

    PORTB = 0x00;
    PORTD = 0x00;
    PORTB |= (1<<Y1);
    PORTB |= (1<<R2);
    PORTD |= (1<<R3);
    PORTD |= (1<<G4);
    _delay_ms(3000);
    PORTB = 0x00;
    PORTD = 0x00;
    }
    }
}
```

```
// KEERTHANA B
// SOHIL DHIREN SHETH
#include <avr/io.h>
#include <util/delay.h>
#define F_CPU 8000000UL
#define R1 PB0
#define Y1 PB1
#define G1 PB2
#define R2 PB3
#define Y2 PB4
#define G2 PB5
```

```
#define R3 PD5
#define Y3 PD4
#define G3 PD3
#define R4 PD2
#define Y4 PD1
#define G4 PD0
int main(void) {
  DDRB = 0xff;
  DDRD = 0xff;
  DDRC = 0x00;
  PORTB = 0x00;
  PORTD = 0x00;
    while(1)    {
    if((PINC&0x01) == 0x01)    {
    PORTB |= (1<<G1);
    PORTB |= (1<<Y2);
    PORTD |= (1<<R3);
    PORTD |= (1<<R4);    }
    else if((PINC&0x02) == 0x02)    {
    PORTB |= (1<<R1);
    PORTB |= (1<<G2);
    PORTD |= (1<<Y3);
    PORTD |= (1<<R4);    }
    else if((PINC&0x04) == 0x04)    {
    PORTB |= (1<<R1);
    PORTB |= (1<<R2);
    PORTD |= (1<<G3);
    PORTD |= (1<<Y4);    }
    else if((PINC&0x08) == 0x08)    {
    PORTB |= (1<<Y1);
    PORTB |= (1<<R2);
    PORTD |= (1<<R3);
    PORTD |= (1<<G4);    }
    else    {
```

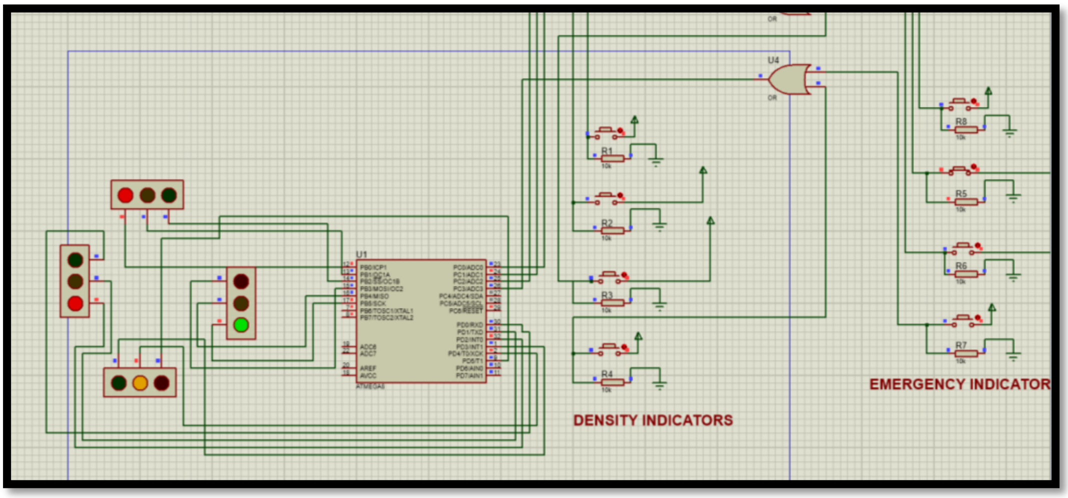
```
PORTB = 0x00;
PORTD = 0x00;
PORTB |= (1<<G1);
PORTB |= (1<<Y2);
PORTD |= (1<<R3);
PORTD |= (1<<R4);
_delay_ms(7000);
 PORTB = 0x00;
PORTD = 0x00;
PORTB |= (1<<R1);
PORTB |= (1<<G2);
PORTD |= (1<<Y3);
PORTD |= (1<<R4);
_delay_ms(7000);
PORTB = 0x00;
PORTD = 0x00;
PORTB |= (1<<R1);
PORTB |= (1<<R2);
PORTD |= (1<<G3);
PORTD |= (1<<Y4);
_delay_ms(7000);
PORTB = 0x00;
PORTD = 0x00;
PORTB |= (1<<Y1);
PORTB |= (1<<R2);
PORTD |= (1<<R3);
PORTD |= (1<<G4);
_delay_ms(7000);
PORTB = 0x00;
PORTD = 0x00;    }  }  }
```

RESULTS AND DISCUSSION

Simulation scenario

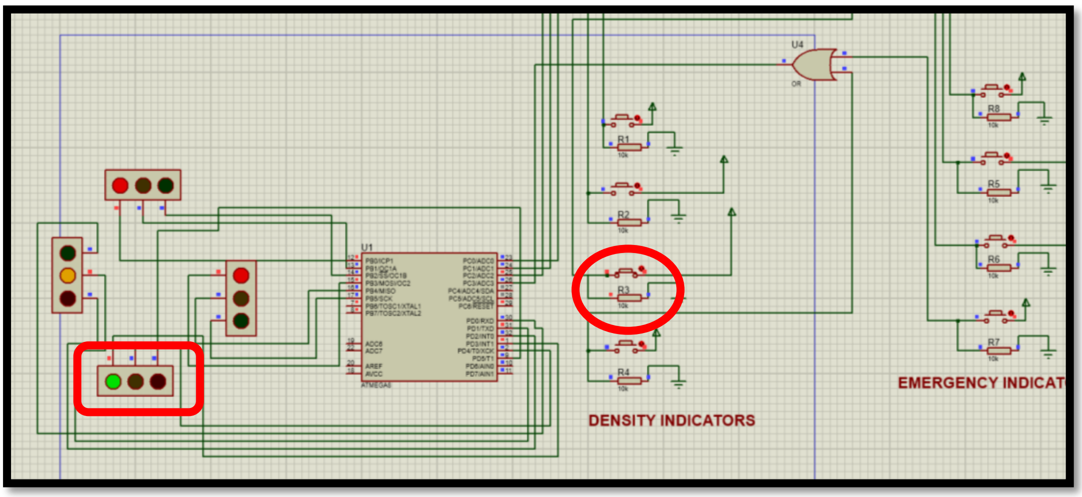
The simulation block is shown in the fig (a). the system is implemented using an ATMEGA8 microcontroller which takes the control input from IR sensors (high density indicators) and acoustic sensors (emergency vehicle indicators). The output can be seen as per the working of the TCS. This system works as per 3 case scenarios.

<u>Case 1:</u> Normal timer at equal No. of vehicles and zero emergency vehicles in all lanes (ALL 8 SWITCHES / SENSORS ARE OFF) – This indicates that all the IR and acoustic sensors are off and hence all roads have equal traffic with zero emergency vehicles. So the traffic signal control system works normally with a 7 second green light delay for each signal.

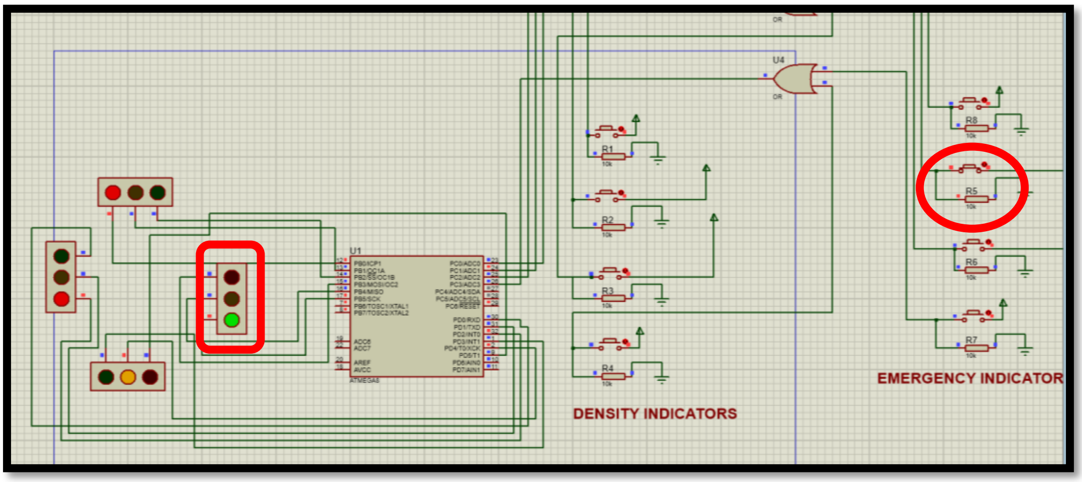


fig(a)

<u>Case 2:</u> High Density at Lane 3 – This is indicated by the 3rd lane's IR sensor turned ON. So, the no. of vehicles in this lane is the highest and hence its traffic signal gets turned to green, while keeping the others in Red light until the sensor goes back to the off state once the traffic has decreased.



Case 3: EMERGENCY Vehicle at Lane 2 – This will make the acoustic sensor in lane 2 to trigger the corresponding traffic signal to turn green, while all other signals remain in Red until the acoustic sensor is turned off, indicating that the emergency vehicle has passed the signal.

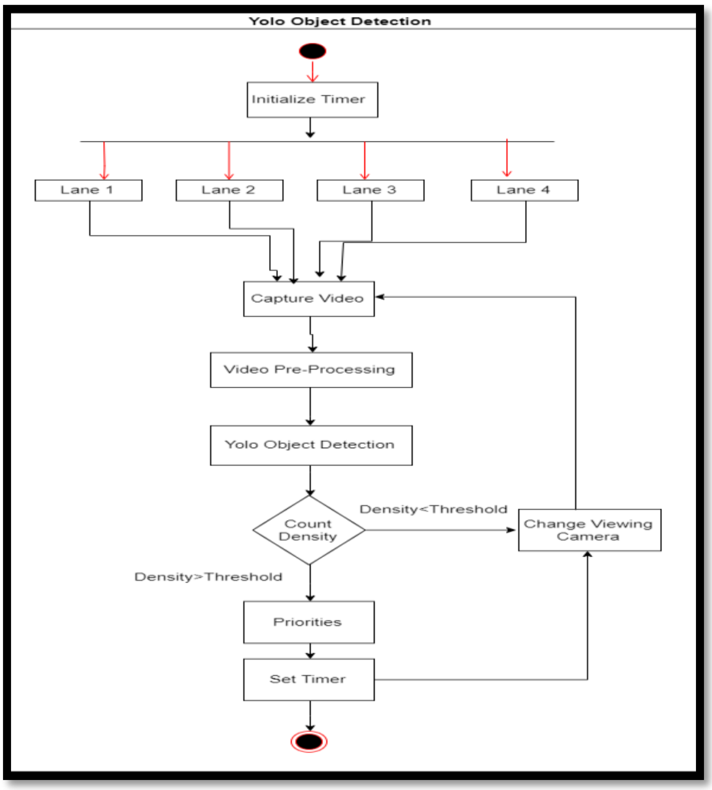


- ***Simulation parameters***

PARAMETERS	VALUES
Number of signals	4
Delay per signal	7
Density sensor	IR sensor
Emergency vehicle sensor	Acoustic sensor
Emergency vehicle sensing range	800Hz and above
Simulation platform	Proteus 8 Professional
Code compilation platform	Arduino IDE
Code language	C with microcontroller coding

METHOD 2 – MACHINE LEARNING

- BLOCK DIAGRAM



- CODE

multithreading.py > countVehicles

```python
from tracking.centroidtracker import CentroidTracker
from tracking.trackableobject import TrackableObject
import tensornets as nets
import cv2
import numpy as np
import time
import dlib
import tensorflow.compat.v1 as tf
import os
import threading
def countVehicles(param):
    # param -> path of the video
    # list -> number of vehicles will be written in the list
    # index ->Index at which data has to be written

    tf.disable_v2_behavior()

    # Image size must be '416x416' as YoloV3 network expects that specific image size as input
    img_size = 416
    inputs = tf.placeholder(tf.float32, [None, img_size, img_size, 3])
    model = nets.YOLOv3COCO(inputs, nets.Darknet19)

    ct = CentroidTracker(maxDisappeared=5, maxDistance=50) # Look into 'CentroidTracker' for further info al
    trackers = [] # List of all dlib trackers
    trackableObjects = {} # Dictionary of trackable objects containing object's ID and its' corresponding ce
    skip_frames = 10 # Numbers of frames to skip from detecting
    confidence_level = 0.40 # The confidence level of a detection
    total = 0 # Total number of detected objects from classes of interest
    use_original_video_size_as_output_size = True # Shows original video as output and not the 416x416 imag

    video_path = os.getcwd() + param # "/videos/4.mp4"
    video_name = os.path.basename(video_path)
```

```python
    # print("Loading video {video_path}...".format(video_path=video_path))
    if not os.path.exists(video_path):
        print("File does not exist. Exited.")
        exit()

    # YoloV3 detects 80 classes represented below
    all_classes = ["person", "bicycle", "car", "motorbike", "aeroplane", "bus", "train", "truck", \
                "boat", "traffic light", "fire hydrant", "stop sign", "parking meter", "bench", \
                "bird", "cat", "dog", "horse", "sheep", "cow", "elephant", "bear", "zebra", "giraffe", \
                "backpack", "umbrella", "handbag", "tie", "suitcase", "frisbee", "skis", "snowboard", \
                "sports ball", "kite", "baseball bat", "baseball glove", "skateboard", "surfboard", \
                "tennis racket", "bottle", "wine glass", "cup", "fork", "knife", "spoon", "bowl", "banana'
                "apple", "sandwich", "orange", "broccoli", "carrot", "hot dog", "pizza", "donut", "cake",
                "chair", "sofa", "pottedplant", "bed", "diningtable", "toilet", "tvmonitor", "laptop", "m
                "remote", "keyboard", "cell phone", "microwave", "oven", "toaster", "sink", "refrigerator'
                "book", "clock", "vase", "scissors", "teddy bear", "hair drier", "toothbrush"]

    # Classes of interest (with their corresponding indexes for easier looping)
    classes = { 1 : 'bicycle', 2 : 'car', 3 : 'motorbike', 5 : 'bus', 7 : 'truck' }

    with tf.Session() as sess:
        sess.run(model.pretrained())
        cap = cv2.VideoCapture(video_path)

        # Get video size (just for log purposes)
        width =  int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
        height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

        # Scale used for output window size and net size
        width_scale = 1
        height_scale = 1
```

RESULTS AND DISCUSSION

The system processed the image and video only once and gives the following output of the traffic

PROJECT INNOVATIONS

CASE 1: Innovation apart from based paper.

BASE PAPER	OUR PROJECT
• This project is done on OpenCV platform that uses C and C++ language. This is a very old way of programming that is difficult to be supported by many AI and modern interfaces.	• The YOLO based project is constructed using python language. This is a latest and easily compatible form of coding in the 21st century. It can easily be approved by the government and implemented in all states. The microcontroller coding is also a latest implementation of the TCS.
• This system captures the image of the traffic and does image processing and noise filtering multiple times before feeding it back into the system for detection of vehicles.	• The YOLO system captures the image and reviews it only once before sending it into the system for detection of vehicles. This reduces the processing time and increases system efficiency.
• This is a high budget project. It requires HD cameras and high-end video processing filters with other additional equipment.	• This system mainly relies on the code and just a few low range hardware or a microcontroller for the MC based idea. So, this is a budget friendly system for a large-scale implementation.
• This system does not support emergency alert.	• The proposed project enables emergency vehicle detection and override of the TCS.

CASE 2: Innovations that will improve the existing open loop traffic control system.

Existing Timer based Open loop Traffic system	Proposed closed loop traffic control system
• No priority is given to lanes with more traffic.	• Priority is given more to lanes with high traffic density. This will turn on the high traffic lanes first.
• No priority to lanes with emergency vehicles in it.	• Higher priority given to lanes with ambulance and other emergency vehicles.
• Less efficient and more time consuming in clearing the traffic.	• More efficient in reducing time required for clearing the traffic lanes. This enables not only modernized

• Requires manual inspection of the roads and traffic by traffic policemen.	road systems but also uplifts the driver's mood in terms of reduced time for travel. • Since this is a completely computerized system, it does not require much of manual inspection on the road. It works fine all by itself.

CASE 3: The difference between the two proposed solution.

Machine Learning project (solution - 1)	**Microcontroller project (solution - 2)**
• It is based on YOLO (You Only Look Once) algorithm that does sensing from a video or image by just processing it once. • It is a more futuristic aspect of traffic control system since it is made using Deep learning with python. • It takes just a camera to detect the vehicles and ambulance in lanes	• It uses IR and Acoustic sensors for sensing no. of vehicles and ambulance present on the road lane to give commands for the Traffic light system. • This is a more conservative way of coding. It is simple yet meets all the needs of the proposed project. • It takes a few sensors and filtering units to capture the input and make a dynamic TCS.

CONCLUSION

In this design work, a density-based traffic control system was developed for traffic control at '+' road intersection to reduce unnecessary time wastage and minimize road traffic casualties which the existing traffic control system has failed to achieve. we have succeeded in minimizing the traffic congestions created by the fixed time-based traffic light system. The system is effective, and the cost of production is very low.

FUTURE SCOPE

- Incorporation of renewable energy sources for 24 hours performance of the system.
- To transmit captured vehicle plate numbers of defaulters in real time to relevant agencies.
- To modify the system to control more than 4 lanes of traffic.
- Instead of using ultrasonic sensors for priority vehicles we can use RFID tag (which can be verified from a authorised agency database) as it will act as a fool proof system and no other vehicles can put the siren and get away.

APPENDIX

Sl. No.	Title	Page
1.	ABSTRACT	1
2.	INTRODUCTION - State of art - Problems faced	1
3.	RELATED WORKS - Literature review	2
4.	PROBLEM STATEMENT - Existing system	4
5.	PROPOSED SOLUTION	5
6.	METHODOLOGY - Architecture - Algorithm & mathematical model - Circuit block - Microcontroller code	6
7.	RESULTS AND DISCUSSION - Simulation scenario - Simulation parameters	11
8.	PROJECT INNOVATIONS	15
9.	CONCLUSIONS	16
10	FUTURE SCOPE	16
11.	REFERENCES	18

REFERENCES

[1] Moka, S. S. P., Pilla, S. M. and Radhika, S. (2020) 'Real Time Density Based Traffic Surveillance System Integrated with Acoustic Based Emergency Vehicle Detection', 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), Computer, Communication and Signal Processing (ICCCSP), 2020 4th International Conference on, pp. 1–7. doi: 10.1109/ICCCSP49186.2020.9315209.

[2] Dr. P. Amudhavalli, M. Monica, R. Sandhya Shree (2) (no date) 'Effective Traffic Management System for Emergency Vehicles' , International Journal of Innovative Technology and Exploring Engineering, 9(1), pp. 453–457. doi: 10.35940/ijitee.A4186.119119.

[3] Liang-Tay Lin et al. (2009) 'A new intelligent traffic control system for Taiwan', 2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST), Intelligent Transport Systems Telecommunications,(ITST),2009 9th International Conference on, pp. 138–142. doi: 10.1109/ITST.2009.5399369.

[4] Neelakandan, S. et al. (2021) 'IoT-based traffic prediction and traffic signal control system for smart city', Soft Computing - A Fusion of Foundations, Methodologies & Applications, 25(18), pp. 12241–12248. doi: 10.1007/s00500-021-05896-x.

[5] Prakash, U. E. et al. (no date) 'Density based traffic control system using image processing', 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research, ICETIETR 2018. doi: 10.1109/ICETIETR.2018.8529111.

[6] Pokorny, P. (2017) 'Determining Traffic Levels in Cities Using Google Maps', 2017 Fourth International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), Mathematics and Computers in Sciences and in Industry (MCSI), 2017 Fourth International Conference on, MCSI, pp. 144–147. doi: 10.1109/MCSI.2017.33.

[7] Nakanishi, H. and Namerikawa, T. (2016) 'Optimal traffic signal control for alleviation of congestion based on traffic density prediction by model predictive control', 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Society of Instrument and Control Engineers of Japan (SICE), 2016 55th Annual Conference of the, pp. 1273–1278. doi: 10.1109/SICE.2016.7749173.

[8] Kanungo, Anurag; Sharma, Ayush; Singla, Chetan (2014). [IEEE 2014 Recent Advances in Engineering and Computational Sciences (RAECS) - Chandigarh, India (2014.03.6-2014.03.8)] 2014 Recent Advances in Engineering and Computational Sciences (RAECS) - Smart traffic lights switching and traffic density calculation using video processing. , (), 1–6. doi:10.1109/raecs.2014.6799542

[9] Komada, Kazuhito; Nagatani, Takashi (2009). [IEEE 2009 IEEE Toronto International Conference - Science and Technology for Humanity (TIC-STH 2009) - Toronto, ON, Canada (2009.09.26-2009.09.27)] 2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH) - Modeling and simulation for vehicular traffic in city network controlled by signals. , (), 66–71. doi:10.1109/TIC-STH.2009.5444416

[10] Shyam Shankaran R;Logesh Rajendran; (2021). Real-Time Adaptive Traffic Control System For Smart Cities . 2021 International Conference on Computer Communication and Informatics (ICCCI), (), –. doi:10.1109/ICCCI50826.2021.9402597

[11] Riyazhussain, S. K. et al. (no date) 'Raspberry Pi controlled Traffic Density monitoring system', Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016, pp. 1178–1181. doi: 10.1109/WiSPNET.2016.7566322.

[12] D. Patel and Y. Rohilla, "Infrared Sensor based Self–Adaptive Traffic Signal System using Arduino Board," 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, pp. 175-181, doi: 10.1109/CICN49253.2020.9242560.