

# SMART TRAFFIC MANAGEMENT SYSTEM

# USING MACHINE LEARNING

A project report in partial fulfillment for the award of the Degree of

**BACHELOR OF TECHNOLOGY IN**

**INFORMATION TECHNOLOGY**

Submitted by

|  |  |
| --- | --- |
| **PUDI UDAY KIRAN** | **320106411037** |
| **REEMALI AKSHAY NAGA KUMAR** | **320106411038** |
|  |  |

Under the esteemed guidance of

**Dr. PRAKASH BETHAPUDI**

Professor

**DEPARTMENT OF INFORMATION TECHNOLOGY AND COMPUTER APPLICATIONS**

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING VISAKHAPATNAM-530003**

**2020-2024**

# SMART TRAFFIC MANAGEMENT SYSTEM

A project submitted by

|  |  |
| --- | --- |
| PUDI UDAY KIRAN | 320106411037 |
| REEMALI AKSHAY NAGA KUMAR | 320106411038 |
|  |  |

In partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

in

**INFORMATION TECHNOLOGY**

Under the guidance of

**Dr. PRAKASH BETHAPUDI**

**Professor**



**DEPARTMENT OF INFORMATION TECHNOLOGY AND COMPUTER APPLICATIONS**

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING VISAKHAPATNAM-530003**

**(2020-2024)**

**DEPARTMENT OF INFORMATION TECHNOLOGY AND COMPUTER APPLICATIONS**

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING VISAKHAPATNAM-530003**



**BONAFIDE CERTIFICATE**

This is to certify that the project title entitled “SMART TRAFFIC MANAGEMENT SYSTEM USING MACHINE LEARNING.”, is the bonafide work done by **Pudi Uday Kiran (320106411037), Reemali Akshay Naga Kumar (320106411038),** submitted in partial fulfillment requirement for the award of Degree of Bachelor of Technology in Information Technology and Computer Applications during the year 2020-2024.

**Dr. PRAKASH BETHAPUDI Prof. KUNJAM NAGESWARA RAO**

**Project guide Head of Department**

# DECLARATION

We declare that the project report entitled “**SMART TRAFFIC MANAGEMENT SYSTEM USING MACHINE LEARNING**” has been done by me in partial fulfillment of the requirement for the award of the degree of “**Bachelor of Technology**”, during the academic year 2020- 2024 under the guidance of “**Dr. PRAKASH BETHAPUDI**”, department of Information Technology and Computer Applications, AU College of Engineering, Andhra University, Visakhapatnam. I, hereby declare that this project work has not been submitted to any other universities/institutions for the award of any degree.

|  |  |  |
| --- | --- | --- |
| NAME | REGD. NO | SIGNATURE |
| Pudi Uday Kiran | 320106411037 |  |
|  |  |  |
| Reemali Akshay Naga Kumar | 320106411038 |  |
|  |  |  |

Visakhapatnam Date:

# ACKNOWLEDGEMENT

We take this opportunity to express deep gratitude to the people who have been instrumental in the successful completion of the project. I would like to show my greatest appreciation to our project guide **Dr. Prakash Bethapudi,** Professor of Department, Department of Information Technology and computer applications, enough for his tremendous support and help without his encouragement and guidance this project Would not have materialized.

We express sincere thanks to **Prof. K. Nageshwar Rao**, Head of the Department Infromation Technology and Computer Applications, Andhra University College of Engineering for his keen interest and providing necessary facilities for this project study.

We express sincere gratitude to **Prof. D. Lalitha Bhaskari,** Chairman, Board of studies, Computer Science and Systems Engineering, Andhra University College Of Engineering for her keen interest and providing necessary facilities for this project study.

We extend our sincere thanks to our academic teaching staff and non-teaching staff for their help throughout our study.

We would also like to extend our gratitude to Lab Technicians and our sincere gratitude to all the faculty members and Non-Teaching Staff in the Department of Computer Science and Systems Engineering for supporting us. We would also like to thank our parents and friends, who have willingly helped us out with their abilities in completing the project work.

|  |  |
| --- | --- |
| With Gratitude, |  |
| **PUDI UDAY KIRAN** | **320106411037** |
|  |  |
| **REEMALI AKSHAY NAGA KUMAR** | **320106411038** |
|  |  |
| B.Tech (INFORMATION TECHNOLOGY) |  |

# ABSTRACT

Congestion in traffic is a serious issue. In existing system signal timings are fixed and they are independent of traffic density, Large red light delays leads to traffic congestion. In this project, we collect the dataset by using IOT tool(photo radar system) to get Video from based traffic estimation system is implemented in which signal timings are updated based on the vehicle counting. This system consists of Deep Learning module it detect the vehicle count of the current system and sends to the traffic signal.

Based on traffic density of Vehicle System will Predict the Traffic Congestion. Eliminate of frequent traffic disconnect of Road network. Act as Road side Unit as server. Implementation of Ambulance Priority system. By implementing an STMS, cities can achieve several benefits, including reduced traffic congestion, shorter travel times, lower fuel consumption and emissions, improved road safety, and enhanced overall quality of life for residents. Furthermore, the scalability and adaptability of the system allow for continuous improvements and expansion to meet evolving transportation needs in urban areas.

The “SMART TRAFFIC MSNAGEMENT SYSTEM” is developed using a programming language machine learning, python with the help of anaconda, Jupyter which is implemented under the Windows operating system.

In conclusion, the Smart Traffic Management System represents a proactive approach to address urban traffic challenges by harnessing the power of technology to create smarter, more efficient transportation networks.

**TABLE OF CONTENTS**

**TITLE PAGE NO:**

1. INTRODUCTION
   1. Overview ………………………………………………………...11
   2. Aim of the project ……………………………………………….12
   3. Scope of the project………………………………………………12
   4. Computational approach………………………………………...12
   5. Python packages…………………………………………………13
2. LITERATURE SURVEY…………………………………………………14
3. PROJECT DESCRIPTION
   1. Existing System…………………………………………………. 16
   2. Proposed System…………………………………………………16
   3. Feasibility Study…………………………………………………17
4. REQUIREMENTS ANALYSIS
   1. System Requirements…………………………………………….18
   2. Python…………………………………………………………….19
   3. Machine Learning……………………………………………….20-23
5. METHODOLOGY
   1. General Architecture………………………………………………23
   2. Workflow Diagram………………………………………………...24
   3. Design……………………………………………………………...25-27
6. RESULTS AND DISCUSSION
   1. Sample code ……………………………………………………….28-32
   2. Test Cases………………………………………………… ………33
   3. Results……………………………………………………………....34-35
7. CONCLUSION AND FUTURE SCOPE………………………………... ….36-37
8. REFERENCES…………………………………………………………………38

**LIST OF FIGURES**

**Fig. No Name Page No.**

5.1 General Architecture 19

5.2 Workflow Diagram 20

5.3.1 Use Case Diagram 25

5.3.2 State Diagram of Traffic Lights 25

5.3.3 Activity diagram 26

**LIST OF ABBREVIATIONS**

**ABBREVIATION EXPANSION**

ANN Artificial Neural Networks

CNN Convolutional Neural Network

YOLO You Only Look Once

IOT Internet of Things

**CHAPTER 1**

**INTRODUCTION**

* 1. **Overview**

The total number of registered automobiles in India was over 295 million, in the year 2019. With an upward increase in the number of vehicles on the road, with that the traffic difficulties have increased as well. Accidents involving roads and traffic frequently go unnoticed and unreported. Illegal traffic operations, such as the transit of overloaded vehicles carrying commodities, the jumping of traffic signals, vehicle over speeding, and so on, must be curtailed. Calculating vehicle density in traffic images is a difficult study topic because it involves dealing directly with hostile yet realistic road conditions such as uncontrolled illuminations, cast shadows, and optical occlusion. Nonetheless, being able to count and resolve vehicles effectively in such settings has a significant impact on traffic surveillance. The extraction of crucial traffic information such as congestion level and lane occupancy is made possible by an accurate vehicle count. In our present day and age, the usage of information to give insights and meaning to the data has increased significantly. One of these ways is machine learning, where the data is generalized from numerous sets of instances. Artificial Neural Networks (ANNs), which attempt to replicate their biological counterparts, are one method employed. The CNN (Convolutional Neural Network) is one of the methods used in it, and it mimics computer vision to recognize, detect, and categorize the objects or entities in the given field of view. By applying N layer processing to the data, the techniques land with the generalization of samples in this case. CNN also offers a variety of ways to consider, one of which is YOLO. This strategy is ideal for a variety of different applications, including real-time vehicle recognition, because of its high accuracy and capacity to outperform other approaches. The most underrated issue in the modern world is traffic.

Road, flyover, bridge, and underpass construction as well as the development of alternative mass public transportation systems are insufficient to ease traffic congestion, which is getting harder to control every day. Therefore, it is imperative to implement a cutting-edge and intelligent system to address these problems. A tool for managing traffic is provided by the Intelligent Traffic Management System, an AI-based application combined

with cameras mounted at traffic intersections is used to detect: Accidents, illegal activities related to traffic law violation, vehicle density to manage green light timing to save wasted time, and all of this with good accuracy under different weather conditions

* 1. **Aim of the project**

The aim of the Smart Traffic Management System project is to mitigate traffic congestion by dynamically adjusting signal timings based on real-time vehicle density data. Utilizing IoT tools for data collection and a deep learning module for vehicle detection, the system predicts congestion levels and optimizes traffic flow. By eliminating frequent traffic disconnects and serving as a roadside unit, it enhances road network efficiency. Additionally, the project implements an ambulance priority system to ensure swift passage during emergencies.

* 1. **Scope of the project**

The scope of the Smart Traffic Management System project encompasses the development of a comprehensive solution for optimizing traffic flow. This includes the design and implementation of IOT-based data collection tools, integration of a deep learning module for vehicle detection, and the creation of algorithms for dynamic signal timing adjustments. The project also involves the deployment of roadside units to facilitate communication and ensure seamless operation. Furthermore, the scope extends to the incorporation of features such as congestion prediction and ambulance priority, aiming to enhance overall road network efficiency and alleviate traffic congestion

* 1. **Computational Approach**

**1.4.1 Data Collection and Preprocessing**

Utilize IOT tools such as photo radar systems to collect real-time traffic data. This involves capturing video footage and extracting relevant information such as vehicle counts and traffic flow patterns. Preprocessing steps may include data cleaning, normalization, and filtering to prepare the data for further analysis.

**1.4.2 Feature Engineering and Selection**

Identify and extract meaningful features from the collected data to inform the traffic management model. This involves analyzing variables such as vehicle speed, density, and direction to understand traffic dynamics. Feature selection techniques may be employed to prioritize important variables while discarding irrelevant ones, optimizing model performance.

**1.4.3 Model Development**

Develop a deep learning model capable of accurately detecting and counting vehicles from the preprocessed data. This involves designing neural network architectures tailored to the task of vehicle detection, training the model using labeled data, and fine-tuning its parameters to improve accuracy and efficiency.

**1.4.4 Model Evaluation and Validation**

Assess the performance of the developed model using various evaluation metrics such as precision, recall, and F1-score. Validation techniques such as cross-validation or holdout validation may be employed to ensure the model's generalization ability and robustness to unseen data. Iterative refinement of the model may be conducted based on validation results to enhance performance.

**1.4.5 Deployment and Monitoring**

Implement the trained model within the Smart Traffic Management System framework for real-world deployment. This involves integrating the model with the system's infrastructure, such as roadside units and traffic signal controllers. Continuous monitoring of the system's performance is crucial to identify any issues or inefficiencies, allowing for timely adjustments and improvements.

* 1. **Python Packages**

**NumPy:**

NumPy is a Python library used for working with multidimensional arrays.

NumPy can be used to perform a wide variety of mathematical operations on arrays. NumPy arrays are faster and more compact than Python lists. An array consumes less memory and is convenient to use and it provides a mechanism of specifying the data types.

**CHAPTER 2**

# LITERATURE SURVEY

Literature survey is mainly carried out in order to analyse the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated them to propose solutions and work on this project.

Reviewing the literature on the topic area at this time helps the researcher to focus further interviews more meaningfully on certain aspects found to be important in the published studies even if these had not surfaced during the earlier questioning. The literature survey can be conducted for several reasons. The literature review can be in any area of the business.

**Papers:**

# Title: "A Survey of Smart Parking Solutions"

# Authors: Azzam Sleit, Ahmad M. El-Desouky, and Mutasem Jarrah

# Published in: IEEE Access, 2020

# Summary: This survey paper provides an overview of various smart parking solutions including sensor-based systems, IoT-based solutions, and mobile applications. It discusses challenges, technologies, and future directions in smart parking systems.

# 2. Title: "An Intelligent Traffic Management System Based on Internet of Things"

# Authors: Shangguang Wang, Xufei Mao, Hui Gao, and Shuang Chen

# Published in: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)

# Summary: This paper proposes an intelligent traffic management system based on IoT technologies. It presents a framework that integrates various components such as data collection, analysis, and decision-making to optimize traffic flow and reduce congestion.

# 3. Title: "Smart Traffic Management System for Congestion Control Using VANET"

# Authors: A. V. Suresh, S. R. Biradar, and G. M. Ahirrao

# Published in: 2019 International Conference on Communication and Signal Processing (ICCSP)

# Summary: This paper presents a smart traffic management system utilizing Vehicular Ad-Hoc Networks (VANETs) for congestion control. It proposes a distributed architecture where vehicles communicate with each other to share traffic information and make decisions to alleviate congestion.

**CHAPTER 3**

# PROJECT DESCRIPTION

# 3.1 Existing System

* Almost all urban cities in the world use traffic light signals to control the traffic on the roads.
* Different types of traffic light control systems are developed which are vehicle actuated lights and static traffic lights.
* But their traffic lights timing are fixed and switching patterns are also predefined in the system it is independent of traffic conditions for the different lanes and they are not changing with real time data.
* In existing system human and Automatic based traffic light Control system system only mostly in use.
* Also some of the sensor based Traffic monitoring also used These techniques are having huge drawbacks.

**Disadvantages**

* That human based system need huge manpower in the form of traffic Police.
* Less accuracy of processing sensor signals
* Prone to Sensor and Man Made errors which leads to system efficiency drop

# 3.2 Proposed System

* This system proposes a new system for predicting the traffic Density by image processing using AI module.
* A camera will be installed alongside the traffic light.
* It will capture image sequences.
* The image sequence will then be analyzed using digital image processing for vehicle detection, and according to traffic conditions on the road, traffic Density can be Estimated.
* This system employs YOLOv5 and AlexnetNet V3 Convolutional neural network pre-trained model to accurately detect the number of vehicles present on the road, Average Vehicle Area and identify emergency vehicles in real-time.
* Using this information, this system can dynamically adjust traffic signals and reroute vehicles to minimize congestion and ensure priority access for emergency vehicles.

**Advantages**

* The process is in real time so in can able to detect from the video stream from live Camera.
* The implementation of this approach runs at 30-40 frames per second, so that it can detect the vehicle very quickly.
* It uses low power processor with 2.4 GHz by using that we can able to achieve low power operation.

# 3.3 Feasibility Study

# 3.3.1 Economic Feasibility

A comprehensive analysis indicates that the implementation of a Smart Traffic Management System (STMS) presents a favorable economic outlook. Initial investment costs are offset by long-term savings in fuel consumption, reduced travel times, and decreased infrastructure maintenance expenses, leading to a positive return on investment.

**3.3.2 Technical Feasibility**

# Evaluation of available technologies demonstrates the feasibility of implementing an STMS, leveraging advancements in IOT, machine learning, and data analytics. Existing infrastructure can be integrated with sensor networks and intelligent algorithms to efficiently manage traffic flow, optimize signal timings, and provide real-time updates to commuters.

**3.3.3 Social Feasibility**

# Stakeholder engagement and public acceptance play crucial roles in the success of an STMS. Initiatives for community involvement, transparent communication, and user-friendly interfaces ensure that the system addresses the needs and concerns of residents, fostering trust and support for its implementation.

**CHAPTER 4**

# REQUIREMENT ANALYSIS

# 4.1 System Requirements

# 4.1.1 Hardware Specification

* Hard Disk: Greater than 500 GB
* RAM: Greater than 4 GB
* Processor: I3 and Above

# 4.1.2 Software Specification

# Python programming language

# Jupyter

# Anaconda

# Python Libraries used:

# Numpy

# 

**4.2 Python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

• Python is Object-Oriented Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

• History of Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

• Python’s source code is fairly easy-to-maintain.

It has several features like:

• Easy-to-learn Python has few keywords, simple structure, and a clearly defined syntax.

• Easy-to-read Python code is more clearly defined and visible to the eyes.

• Python provides interfaces to all major commercial databases.

• Interactive Mode Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

• It can be used as a scripting language or can be compiled to byte-code for building large applications.

• It supports functional and structured programming methods as well as OOP.

• It can be easily integrated with C, C++, COM, CORBA, and Java.

# 4.3 Machine Learning

# Machine Learning is a category of algorithms that allow software applications to predict much better results without being specifically programmed. The basic premise of machine learning is to build algorithms that receive input data and use statistical analysis to predict output data while output data is updated like many input data become valid. The processes involved in machine learning are similar to the processes of data mining and predictive modelling. Both require searching for certain patterns by date, and adjusting program actions accordingly. Many people are also familiar with machine learning from internet shopping and the advertisements that are shown to them depending on what they are buying. This is because referral engines use machine learning to customize ads that are delivered online in near real time. In addition to personalized marketing, other well-known cases in which machine learning is used are fraud detection, spam filtering, threat detection of countries in the network, maintenance, predictability, and building the flow of news.

# 4.3.1 How machine learning works

# There are 3 types of algorithms in machine learning. They are:

# 1. Supervised Algorithms

# 2. Unsupervised Algorithms

# Supervised Algorithms

# Supervised learning algorithms are used when the dataset is labeled, meaning each data instance is associated with a corresponding label or outcome. In the context of a smart traffic management system, supervised learning can be applied in various ways:

# ****Traffic Flow Prediction****:

# By collecting historical data on traffic flow (e.g., vehicle counts, speed, congestion levels) along with corresponding timestamps and weather conditions, supervised learning algorithms such as regression or time series analysis can be used to predict future traffic flow patterns. These predictions can help optimize traffic light timings, plan road maintenance, or suggest alternative routes to drivers.

**Anomaly Detection**:

Supervised algorithms can be trained to identify anomalies in traffic patterns, such as accidents, road closures, or unusual congestion. This can be achieved by labeling instances of normal traffic behavior and then using classification algorithms to detect deviations from the norm in real-time.

**Unsupervised Algorithms**:

Unsupervised learning algorithms are used when the dataset is unlabeled, meaning there are no predefined categories or outcomes. In the context of a smart traffic management system, unsupervised learning can be valuable for:

**Traffic Clustering**:

Unsupervised algorithms like k-means clustering can group together similar traffic patterns or clusters, such as peak traffic hours, different types of road users (e.g., cars, buses, bicycles), or areas with similar congestion levels. This information can inform decisions about resource allocation, urban planning, and infrastructure improvements.

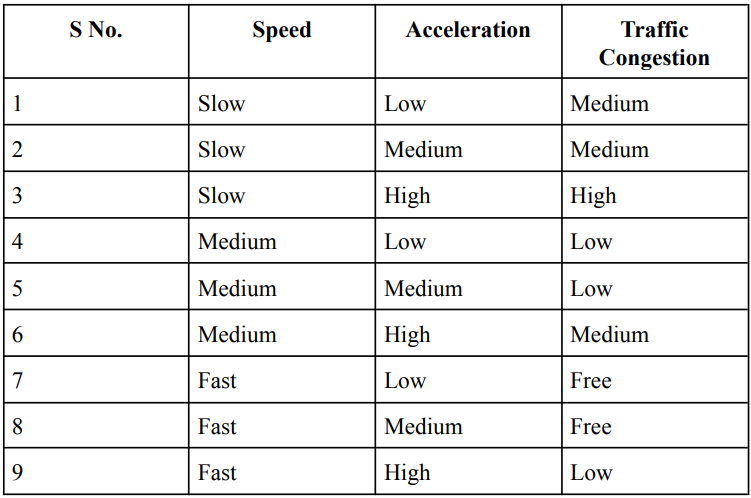
**Route Optimization**:

Unsupervised algorithms can analyze historical traffic data to identify common routes taken by drivers and suggest optimized routes to minimize travel time or avoid congestion. This can be particularly useful for navigation systems and ride-sharing platforms

In summary, both supervised and unsupervised learning algorithms play crucial roles in a smart traffic management system, from predicting traffic flow and detecting anomalies to clustering traffic patterns and optimizing routes. By leveraging machine learning techniques, such systems can improve traffic efficiency, reduce congestion, and enhance overall urban mobility

**Fuzzy Rule:**

The context providers such as Radar Camera or CORDON multi-target Photo Radar System, Automatic Incident Detection (AID), Timer provides real-time data such as the average speed of the vehicles, average acceleration of the vehicles, if an accident took place, current light status and timer value, etc at every intersection. The data we obtained from these providers is primary data. We further classify this and refine this data into a proper dataset. We classify average speed into three types as “Slow”, “Medium” and “Fast”. Similarly, we classify average acceleration into three types as “Low”, “Medium”, and “High”. Now based on the fuzzy rule, we further derive traffic congestion levels at each intersection. This fuzzy rule depends on average speed and average acceleration at the intersection



Based on the above fuzzy rule, we will derive the traffic congestion level at each intersection in real-time. So now our approach is a hybrid approach that makes the traffic signal decision based on the timer and the local traffic density.

In this approach, as per the existing model, at a particular crossing, if the traffic light status of one lane is green, then the corresponding remaining traffic light status should be red. And when the traffic light timer expires, change the current traffic light status, i.e. change one light status to Green ⇒ Yellow ⇒ Red and another to Red ⇒ Yellow ⇒ Green anticlockwise. In this approach, if traffic density in the lane is LOW and current light status is GREEN for at least 10 seconds and the respective timer is not expired, while there is a lane with HIGH traffic density having current light status RED for at least 10 seconds and the respective timer is not expired; then change the light status with HIGH traffic density to GREEN and LOW traffic density to RED and reset the timer

Also, if traffic density in all the lanes is approximately similar or in the moderate limit, then traffic lights status changes based on the traffic light timer, not on the congestion level. And if an accident takes place on one particular road, then people are informed that an accident took place in that particular road by blinking the yellow traffic light until the current timer expires indicating people to go slow until the issue is resolved and inform nearby traffic control stations.

**CHAPTER 5**

# METHODOLOGY

# 5.1 General Architecture

# Figure 5.1: System Architecture Flow Chart

# ● Traffic Light:

# ○ L1 → Red

# ○ L2 → Yellow

# ○ L3 → Green

# ● Average Speed:

# ○ avg\_speed <= 20.0 → Low

# ○ 20.0 < avg\_speed <= 50.0 → Medium

# ○ avg\_speed > 50.0 → high

# ● Average acceleration:

# ○ avg\_acc <= 10.0 → Low

# ○ 10.0 < avg\_acc <= 35.0 → Medium 10

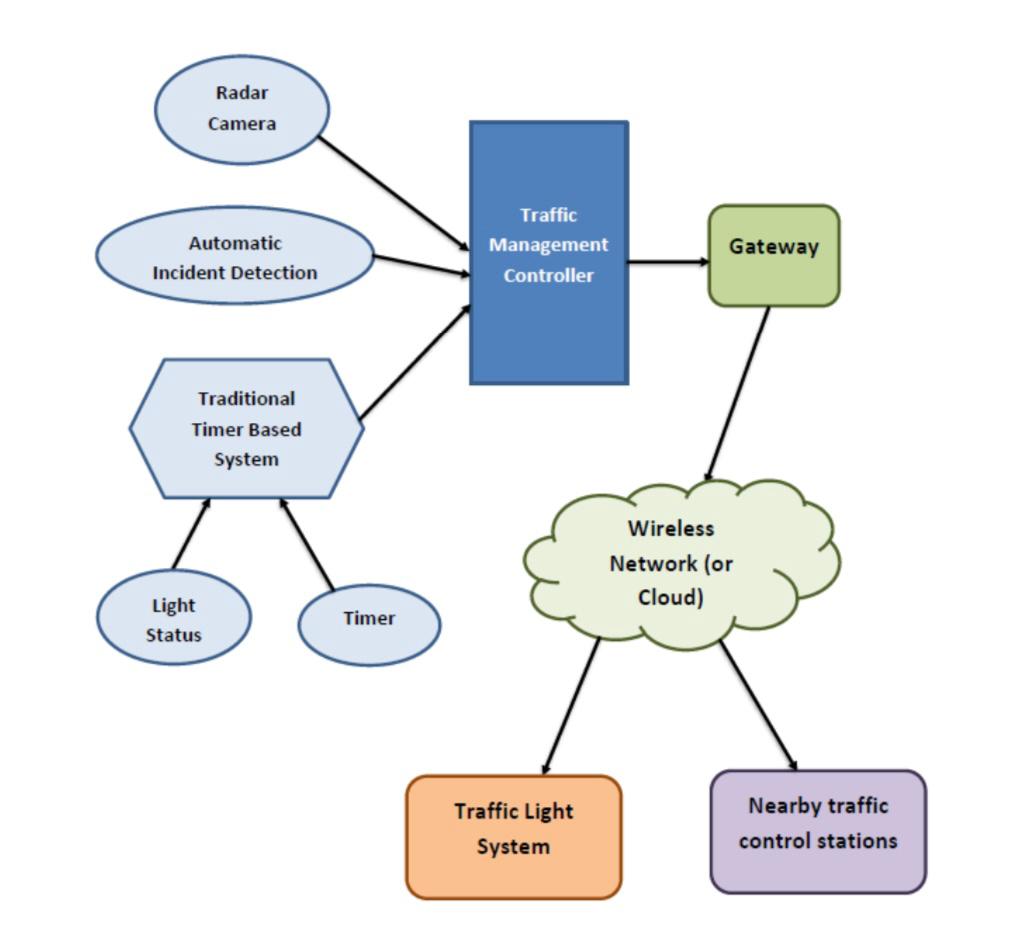
# ○ avg\_acc <= 35.0 → High

# 

# ● Automatic Incident Detection:

# ○ 0 → No accident

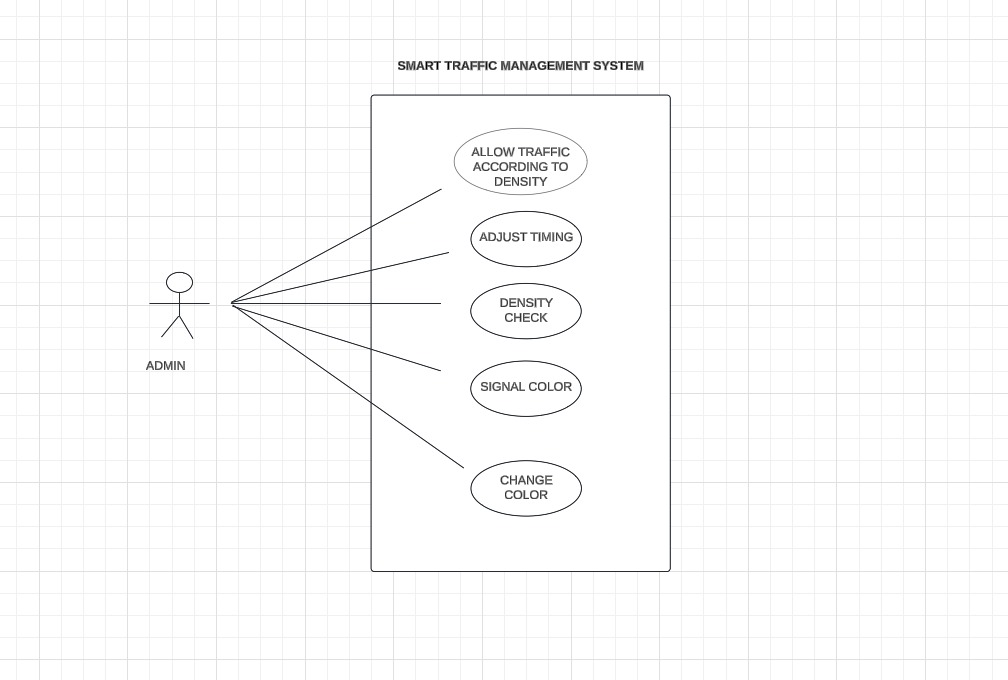
# ○ 1 → accident occurred.

**5.2 Workflow Diagram:**

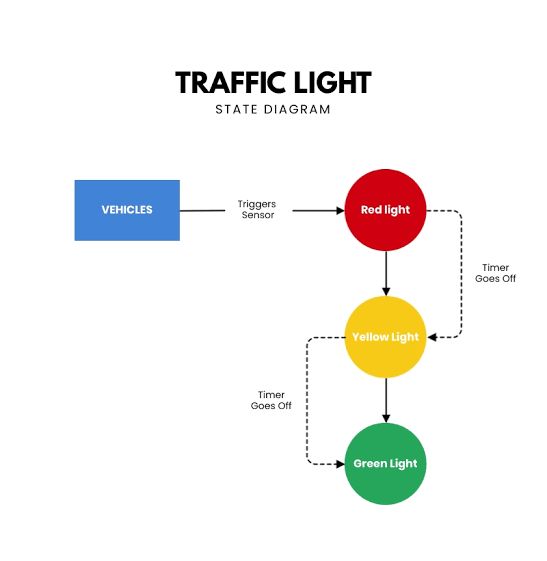
The architecture of the context-aware management system for the Smart Traffic Management System is shown above. Various context providers feed information to the Traffic Management Controller. Here the context providers are the Radar Camera which provides the average speed and average acceleration of the vehicles, the Automatic Incident Detection (AID) which informs if an accident took place or not, and the Traditional Timer Based System which provides light status and timer status. This controller then interacts with this data and finds the set of actions to take place in real-time according to the defined context behaviours. Now the action to be taken is finalised. This information is then uploaded to the cloud with the help of gateways. The finalized information is then updated in the Traffic Light System and also notified to the nearest traffic control station with the help of a wireless network (or through Cloud).

**5.3 Design:**

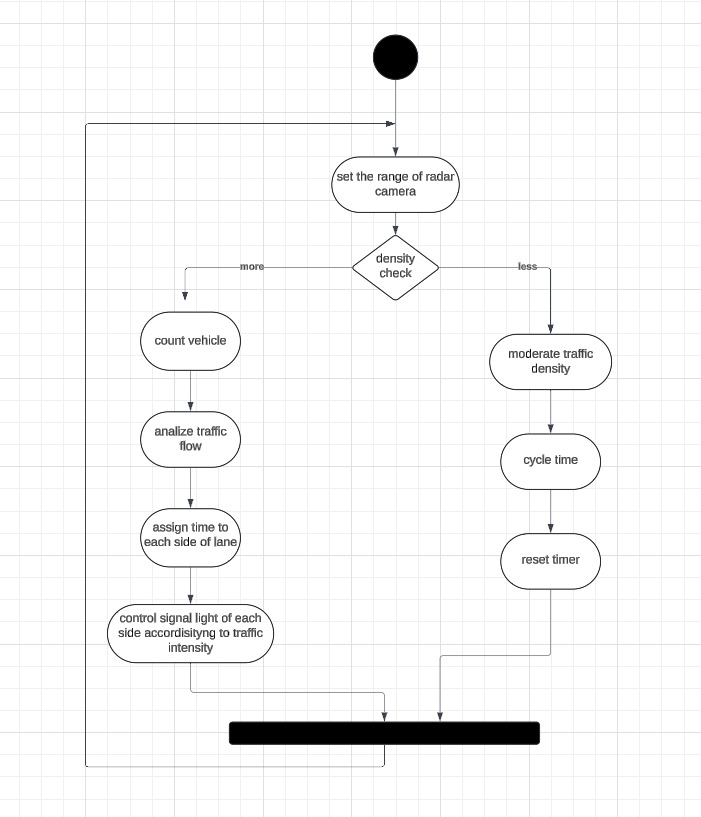
**5.3.1 Use case diagram**

****

**5.3.2 State Diagram of Traffic Lights**

****

**5.3.3 Activity Diagram**

****

**CHAPTER 6**

# RESULTS AND DISCUSSION

**6.1 Sample code**

**Importing the necessary libraries and loading the Dataset**

import pandas as pd

df=pd.read\_csv("CAC Smart Traffic Management - Sheet1.csv")

df.shape

df.head()

class Light:

def \_\_init\_\_(self,light\_color, timer):

self.light\_color=light\_color

self.timer=timer

class Crossing:

def \_\_init\_\_(self, crossingID, laneID, time, avg\_speed, avg\_acc, textSpeed, textAcc, density, accident):

self.crossingID=crossingID

self.laneID=laneID

self.time=time

self.avg\_speed=avg\_speed

self.avg\_acc=avg\_acc

self.textSpeed=textSpeed

self.textAcc=textAcc

self.density=density

self.accident=accident

Entities = []

density="FREE"

light\_color="RED"

accident="NO"

for index, row in df.iterrows():

if(row["Current\_Light\_Status"]=="L1"):

light\_color = "RED"

elif(row["Current\_Light\_Status"]=="L2"):

light\_color = "YELLOW"

elif(row["Current\_Light\_Status"]=="L3"):

light\_color = "GREEN"

lightItem=Light(light\_color, row["Current\_Timer"])

if(row["Average\_Speed"]<=20.0 and row["Average\_Acceleration"]<=10.0):

avg\_speed="LOW"

avg\_acc="LOW"

density="MEDIUM"

elif(row["Average\_Speed"]<=20.0 and row["Average\_Acceleration"]>=10.0 and row["Average\_Acceleration"]<=35.0):

avg\_speed="LOW"

avg\_acc="MEDIUM"

density="MEDIUM"

elif(row["Average\_Speed"]<=20.0 and row["Average\_Acceleration"]>35.0):

avg\_speed="LOW"

avg\_acc="HIGH"

density="HIGH"

elif(row["Average\_Speed"]>=20.0 and row["Average\_Speed"]<=50.0 and row["Average\_Acceleration"]<=10.0):

avg\_speed="MEDIUM"

avg\_acc="LOW"

density="LOW"

elif(row["Average\_Speed"]>=20.0 and row["Average\_Speed"]<=50.0 and row["Average\_Acceleration"]>=10.0 and row["Average\_Acceleration"]<=35.0):

avg\_speed="MEDIUM"

avg\_acc="MEDIUM"

density="LOW"

elif(row["Average\_Speed"]>=20.0 and row["Average\_Speed"]<=50.0 and row["Average\_Acceleration"]>35.0):

avg\_speed="MEDIUM"

avg\_acc="HIGH"

density="MEDIUM"

elif(row["Average\_Speed"]>50.0 and row["Average\_Acceleration"]<=10.0):

avg\_speed="HIGH"

avg\_acc="LOW"

density="FREE"

elif(row["Average\_Speed"]>50.0 and row["Average\_Acceleration"]>=10.0 and row["Average\_Acceleration"]<=35.0):

avg\_speed="HIGH"

avg\_acc="MEDIUM"

density="FREE"

elif(row["Average\_Speed"]>50.0 and row["Average\_Acceleration"]>35.0):

avg\_speed="HIGH"

avg\_acc="HIGH"

density="LOW"

else:

density="Error!"

if(row["Automatic Incident Detection (AID)"]==0):

accident="NO"

else:

accident="YES"

crossingItem=Crossing(row["Crossing\_Id"], row["Lane\_Id"], row["Time"], row["Average\_Speed"], row["Average\_Acceleration"], avg\_speed, avg\_acc, density, accident)

Entities.append({

"Light":lightItem,

"Crossing":crossingItem

})

Entities

start = "\033[1m"

end = "\033[0;0m"

i=0

for entity in Entities:

i+=1

if i<=80:

print("\n\n" + start + "Context Scenario for ==>" + end + "\nCrossing: ",entity["Crossing"].crossingID,"\nLane ID: ", entity["Crossing"].laneID,"\nCurrent Time: ",entity["Crossing"].time,"\nAverage Speed: ",entity["Crossing"].avg\_speed,"\nAverage Acceleration: ", entity["Crossing"].avg\_acc, "\nSignal Timer: ",entity["Light"].timer,"\nLight Status: " + str(df["Current\_Light\_Status"][i-1]) + "\nAccident Detected: " + str(df["Automatic Incident Detection (AID)"][i-1]) + "." +end + " ")

start = "\033[1m"

end = "\033[0;0m"

greenCID = ""

greenLID = ""

greenDENSITY = ""

greenTIME = ""

greenCOLOR = ""

j=0

i = 0

for entity in Entities:

i+=1

j+=1

if(i==1 or i==3 or i==5 or i==7):

if(entity["Light"].light\_color == "GREEN"):

greenCID = entity["Crossing"].crossingID

greenLID = entity["Crossing"].laneID

greenTIMER = 60-entity["Light"].timer

greenTIME = entity["Crossing"].time

greenCOLOR = entity["Light"].light\_color

x = (i-2)%8

if i == 8:

if j<=80:

if(greenCOLOR == "GREEN"):

print("\n\n" + start + "Context Scenario for ==>" + end + "\nCrossing: " + greenCID + "\nTime: " + greenTIME + start + "\nFrom the obtained context information at the Crossing ID " + greenCID + "; the Lane ID " + greenLID + ", where the current light status is " + greenCOLOR + " will \nchange to RED,and at the Lane ID R" + str(x) +", where the current light status is red will change to GREEN in " + str(greenTIMER) + " seconds." + end + " ")

i = 0

greenCID = ""

greenLID = ""

greenTIMER = ""

greenTIME = ""

greenCOLOR = ""

# 6.2 Test Cases:

# 6.2.1 No Two Lanes getting Green Light

# 

# 6.2.2 Accident Detection

# 

# 6.2.3 Light Changes To Green Where The Lane Contains High Density

# 

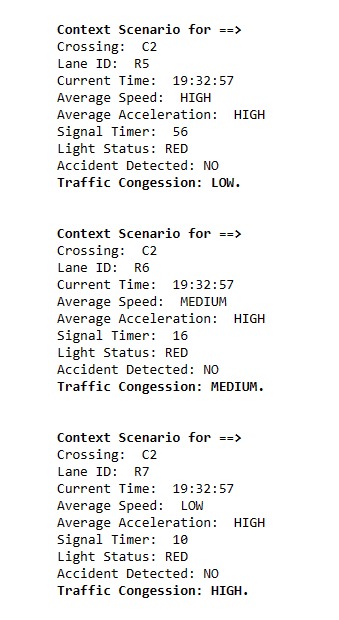
# 6.3 Result:

# A smart traffic management system has been modelled and presented. The system is an event-based context-aware system that takes the context input from listed context providers, processes as per the context behaviours structured in the model and are finally consumed by the traffic light system as the context consumer. The model operates on the fuzzy theory to classify the traffic into certain criteria.

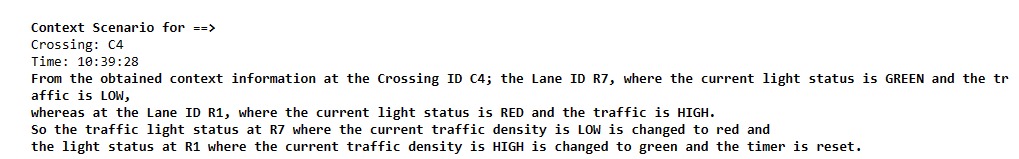
# Here as the traffic density in all the lane are approximately similar or in moderate limit, so traffic lights status changes based on the default traffic light timer.

# 

**Congession Levels**



**Final Output**



**CHAPTER 7**

# CONCLUSION AND

# FUTURE ENHANCEMENTS

# 7.1 Conclusion

A smart traffic management system with dynamic operational features along with the traditional timer-based traffic light system has been modelled and presented. The system is smart enough to handle the case and device failure based uncertainties, any accident occurrence and also the issue of wrong-lane driving.

The real-time implementation of the system will for sure bring a better and smarter traffic management system to come into real-life action, and help to tackle the prevailing issues of the obsolete timer-based traffic light system.

# 7.2 Future Enhancement

# AI-Powered Predictive Analytics: Utilize advanced machine learning algorithms to predict traffic patterns, congestion, and accidents. This allows for proactive traffic management and optimization of traffic flow.

# Integration with Autonomous Vehicles: Incorporate communication systems that enable seamless interaction between autonomous vehicles and traffic infrastructure. This integration can optimize traffic flow, reduce congestion, and enhance safety.

# Dynamic Traffic Signal Control: Implement adaptive traffic signal control systems that adjust signal timing in real-time based on traffic conditions, pedestrian activity, and other factors. This can reduce waiting times and improve traffic flow.

# Smart Intersection Management: Develop intelligent intersection management systems that prioritize different modes of transportation, such as pedestrians, cyclists, and public transit, based on real-time demand and safety considerations.

# Connected Vehicle Technology: Expand the use of connected vehicle technology, allowing vehicles to communicate with each other and with infrastructure. This enables cooperative maneuvers, such as platooning, which can improve traffic flow and fuel efficiency.

**CHAPTER 8**

# REFERENCES

# 1. Title: "A Survey of Smart Parking Solutions"

# Authors: Azzam Sleit, Ahmad M. El-Desouky, and Mutasem Jarrah

# Published in: IEEE Access, 2020

# 2. Title: "An Intelligent Traffic Management System Based on Internet of Things"

# Authors: Shangguang Wang, Xufei Mao, Hui Gao, and Shuang Chen

# Published in: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)

# 3. Title: "Smart Traffic Management System for Congestion Control Using VANET"

# Authors: A. V. Suresh, S. R. Biradar, and G. M. Ahirrao

# Published in: 2019 International Conference on Communication and Signal Processing (ICCSP)