Internship Report on

Smart City Traffic Patterns

Prepared by

PRATEEK DUBEY.

Executive Summary
This report provides details of the Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).   This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time.   Overall, the project provides a practical and hands-on experience in solving real-world problems in the realm of machine learning.   This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

NEED OF SMART CITY TRAFFIC PATTERNS

- **The congestion of urban traffic is becoming one of the critical issues with incresing population and automobile in cities.**

- **Traffic jams not only cause extra delay and stress for the drivers, but also increase fuel consumption, add transportation cost , and increases co2 pollution.**
- **There can be different causes of congestionin traffic like insufficient capacity, unrestrained demand, large red light delays, etc.**
- **The traffic lights are one the of the critical factors affecting the traffic flow**

Context

Mission

You are working with the government to transform your city into a smart city. The vision is to convert it into a digital and intelligent city to improve the efficiency of services for the citizens. One of the problems faced by the government is traffic. You are a data scientist working to manage the traffic of the city better and to provide input on infrastructure planning for the future.

The government wants to implement a robust traffic system for the city by being prepared for traffic peaks. They want to understand the traffic patterns of the four junctions of the city. Traffic patterns on holidays, as well as on various other occasions during the year, differ from normal working days. This is important to take into account for your forecasting.

Proposed System Model

- **Our proposed system will pass a snapshot from the CCTV camera at traffic junction for the real time traffic density calculation using image processing and computer vision.**

- **We are using YOLO(YOU ONLY LOOK ONCE) model for object dectection in order to detect the vehicle.**

- **The scheduling algorithm will use the traffic density and approximately set the optimal green signal for each signal, and update the red signal times of the other signals.**

Advantages of proposed system

1. **Real time traffic light switching according to current traffic density.**
2. **Virtually no new hardware to be installed.**
3. **Less expensive than sensors.**
4. **Autonomous NO need of manpower.**

DRAWBACK OF CONENTIONAL SYSTEMS

1. **The manual controlling system require a large numbers of manpower.**
2. **Conventional traffic lights uses a timer for every day phase , which is fixed and does not adapt according to real time on the road.**
3. **Electronic sensors i.e. proximity sensors or loop detectors, the accuracy and converage are often in conflict because the collection of high quality information is usually based on sophisticated and expensive techologies and thus limited budget will reduce the number of facilities.**

The key features are listed below depending on the city's size and the scope of the governmental policies. It can be integrated into an intelligent traffic management system. They include:

1. **Traffic Jam Detection: With cloud connectivity, sensors, and CCTV cameras tracking intersections 24×7, technicians can remotely monitor all the streets in real-time from the city's traffic control room.**

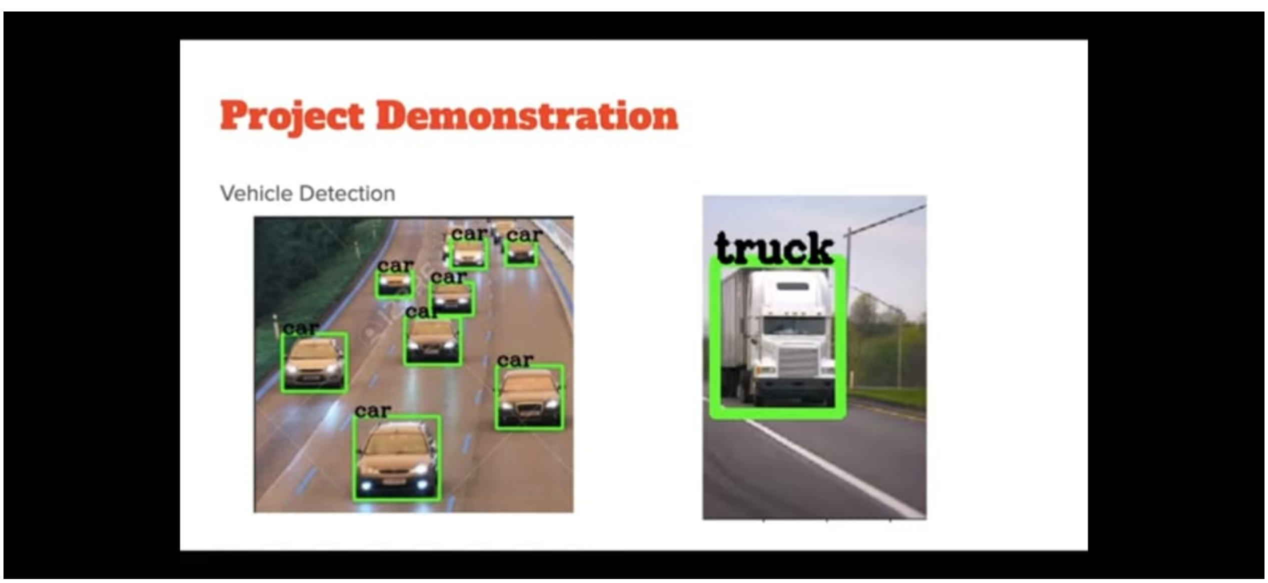
2. **Connected Vehicles: A smart traffic system using IoT technology can connect with roadside tracking devices to enable direct communication between intelligent vehicles & intersections.**

3. **Modular Control: Real-time detection of congestion triggers dynamic adjustments in the systems meant for controlling traffic lights, express lanes, and entry alarms.**

4. **Emergency Navigation: A system with edge data processing & programmatic alerting capabilities can alert response units (police, ambulance & tow trucks) in case of a car crash or collision. It reduces the crucial time an injured driver or passenger remains unattended.**

5. **Road Safety Analytics: Systems with pattern detection capabilities can immediately flag high cruising speeds and reckless driver or inappropriate pedestrian behavior.**

6. **Digital Payments: Commercial traffic management systems enable quick and convenient electronic transactions in real time while ensuring financial data safety**



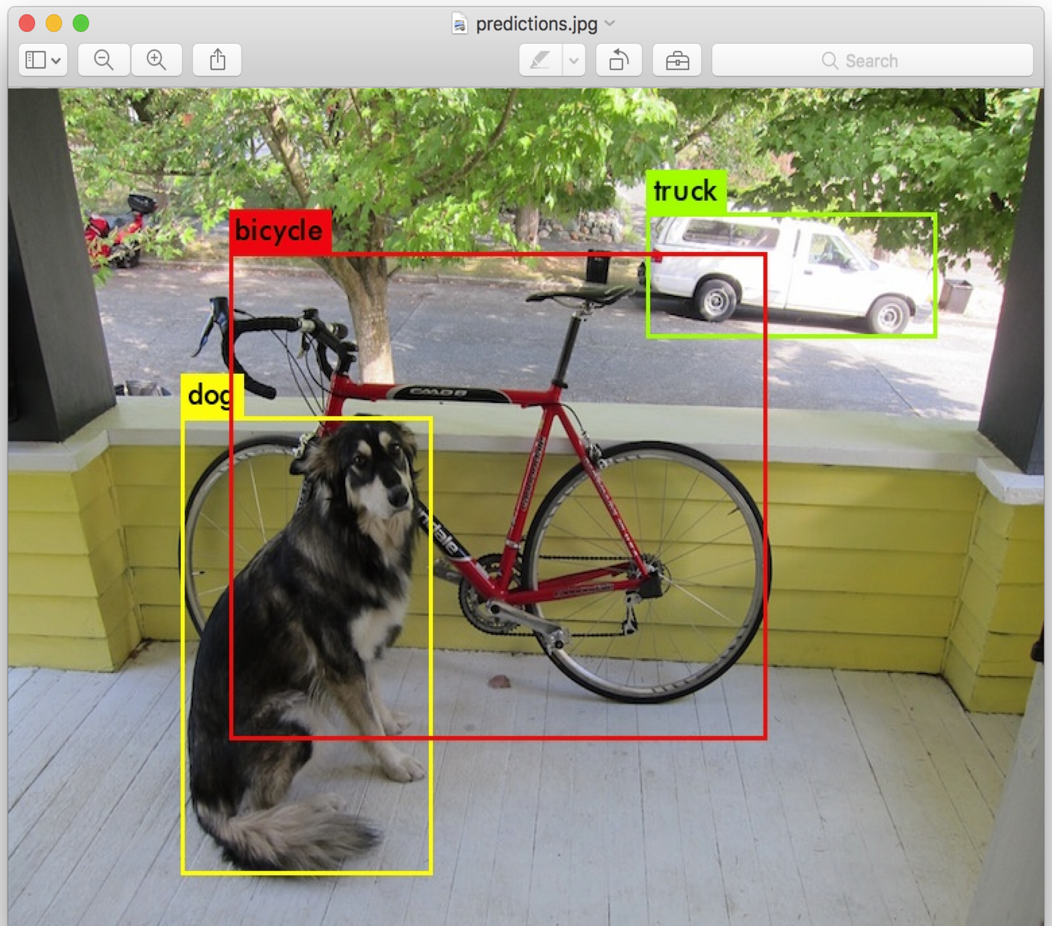
⇒ **Traffic prediction has always been a challenge for transportation planners and city managers. With the increasing growth of cities and the number of vehicles on the roads, the need for accurate and reliable traffic predictions has become more pressing. In recent years, machine learning has shown great promise in solving this problem.**

⇒

⇒ **Traffic prediction involves estimating the future behavior of traffic in a particular area. This information is useful for a variety of purposes, including reducing congestion, optimizing transportation systems, and improving road safety. In the past, traffic prediction has been based on traditional methods such as rule-based models and time-series analysis. However, these methods are often limited in their ability to capture the complexity and variability of traffic patterns.**

⇒

⇒ **Machine learning, on the other hand, is well-suited to handle large and complex datasets, making it an ideal tool for traffic prediction. Machine learning algorithms can automatically identify patterns and relationships in traffic data and use these to make predictions about future traffic conditions**


predictions.jpg
Search
truck
bicycle
dog

Introduction to YOLO Algorithm for Object Detection

YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals.

This article introduces readers to the YOLO algorithm for object detection and explains how it works. It also highlights some of its real-life applications.

Introduction to object detection

Object detection is a phenomenon in computer vision that involves the detection of various objects in digital images or videos. Some of the objects detected include people, cars, chairs, stones, buildings, and animals.

This phenomenon seeks to answer two basic questions:

1st *What is the object?* This question seeks to identify the object in a specific image.

2nd *Where is it?* This question seeks to establish the exact location of the object within the image.

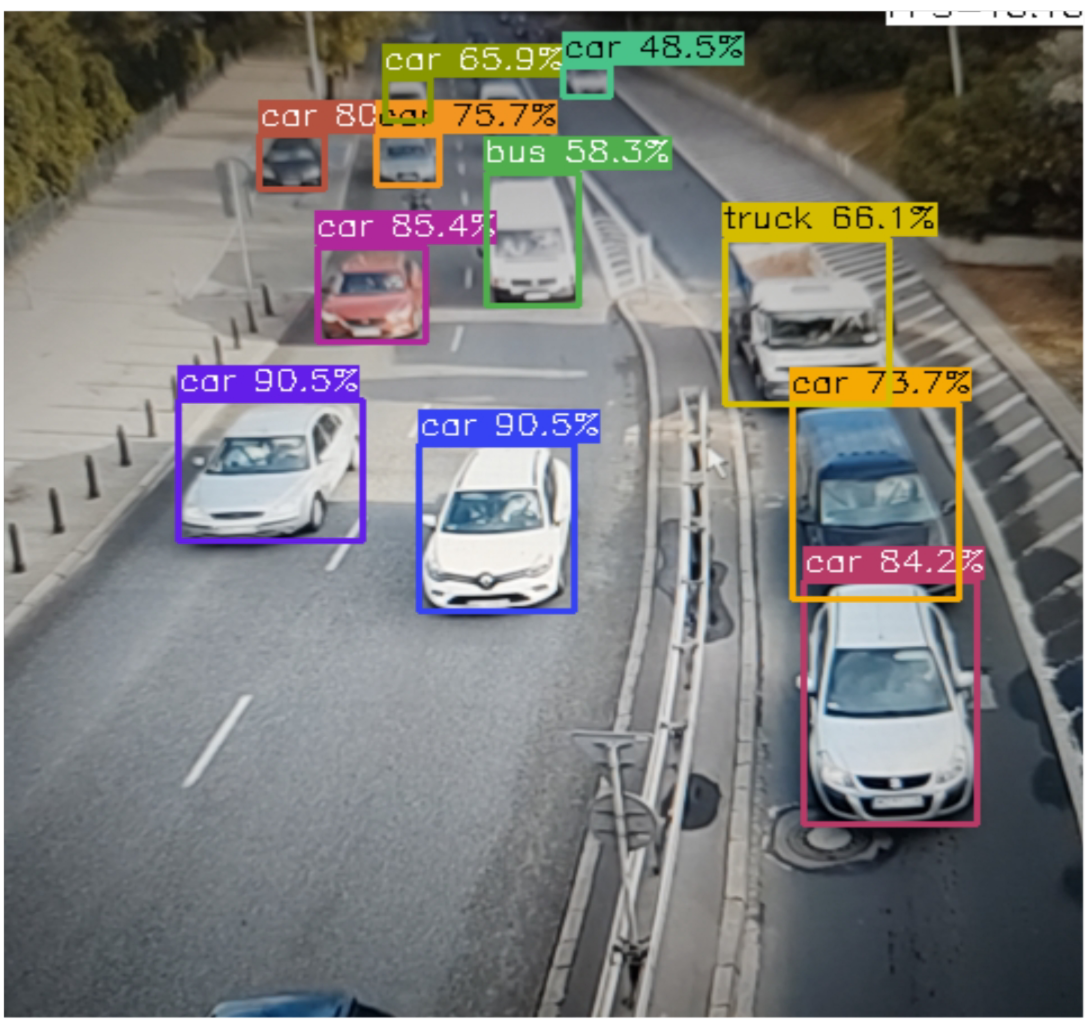
Object detection consists of various approaches such as fast R-CNN, Retina-Net, and Single-Shot MultiBox Detector (SSD). Although these approaches have solved the challenges of data limitation and modeling in object detection, they are not able to detect objects in a single algorithm run. **YOLO algorithm** has gained popularity because of its superior performance over the aforementioned object detection techniques.

YOLO is an abbreviation for the term 'You Only Look Once'. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.

YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.

This means that prediction in the entire image is done in a single algorithm run. The CNN is used to predict various class probabilities and bounding boxes simultaneously.

The YOLO algorithm consists of various variants. Some of the common ones include tiny YOLO and YOLOv3.



Why the YOLO algorithm is important

YOLO algorithm is important because of the following reasons:

- **Speed:** This algorithm improves the speed of detection because it can predict objects in real-time.
- **High accuracy:** YOLO is a predictive technique that provides accurate results with minimal background errors.
- **Learning capabilities:** The algorithm has excellent learning capabilities that enable it to learn the representations of objects and apply them in object detection.

Applications of YOLO

YOLO algorithm can be applied in the following fields:

- **Autonomous driving:** YOLO algorithm can be used in autonomous cars to detect objects around cars such as vehicles, people, and parking signals. Object detection in autonomous cars is done to avoid collision since no human driver is controlling the car.

- **Wildlife:** This algorithm is used to detect various types of animals in forests. This type of detection is used by wildlife rangers and journalists to identify animals in videos (both recorded and real-time) and images. Some of the animals that can be detected include giraffes, elephants, and bears.

- **Security:** YOLO can also be used in security systems to enforce security in an area. Let's assume that people have been restricted from passing through a certain area for security reasons. If someone passes through the restricted area, the YOLO algorithm will detect him/her, which will require the security personnel to take further action.

Smart Traffic Management Systems are technology solutions that municipalities can integrate into their traffic cabinets and intersections today for fast, cost-effective improvements in safety and traffic flow on their city streets. What's more, deploying these systems today, or upgrading your city's existing Intelligent Transportation Systems (ITS) infrastructure can create huge efficiencies and cost savings, while massively improving system reliability, all of which have excellent ROI.

These systems utilize sensors, cameras, cellular routers and automation to monitor and automatically direct traffic and reduce congestion. The right technology solution can be scaled to any size and painlessly upgraded at any time. Simultaneously, these technology solutions prepare Smart Cities for coming technology evolutions, including Connected Vehicle and the full deployment of 5G networks.

Why Evaluate Smart Traffic Technology?

Budgets for public infrastructure are always tight, and constructing roads and bridges is always expensive. Smart Traffic Management Systems help municipal and regional transportation departments to cope with the situation — quickly and cost-effectively. Integrating smart traffic technology helps them affordably get better performance from their existing infrastructure.

Let's explore how augmenting and retrofitting infrastructure can dramatically improve the efficiency and safety of existing traffic networks.

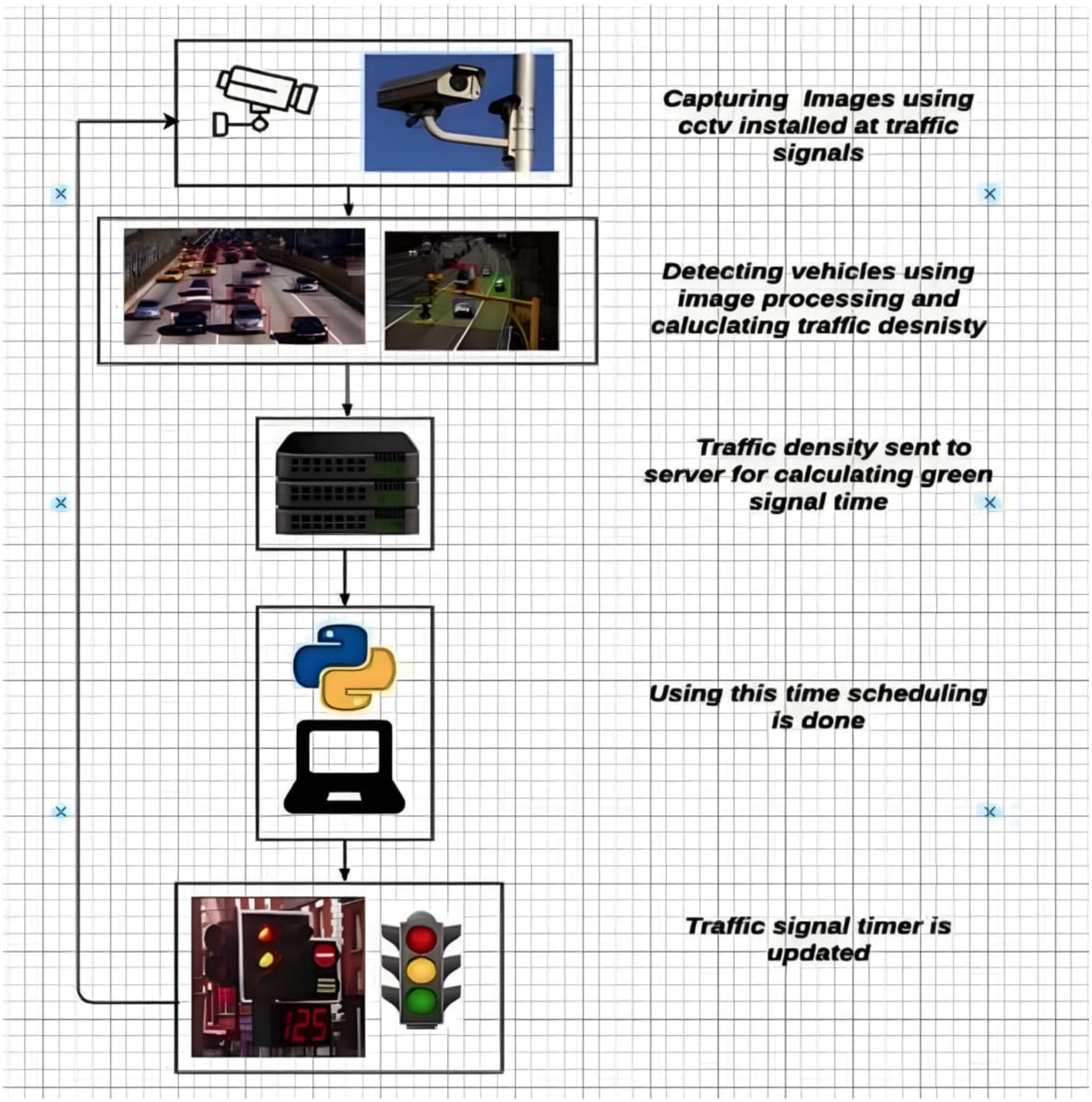
Smart Traffic Management: Smart Cities Do More With Less

The problems plaguing our streets and highways are well known. Traffic slow-downs can cause debilitating congestion and add to urban air pollution. Businesses suffer from delivery delays and lost productivity. Emergency vehicles are slowed down by bottlenecks, potentially putting lives at risk. And all of it diminishes the city's overall quality of life.

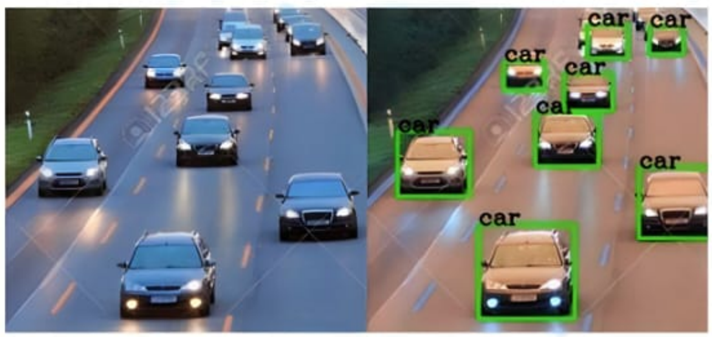
Meanwhile, cities and regional governments continually ask their traffic management teams, civil engineers and highway maintenance crews to do more with less. In the face of these challenges, innovative cities — "Smart Cities" — are using a coordinated array of hardware, software and cloud solutions to increase traffic flow and improve safety. Smart Traffic Management Systems, which are included in the umbrella of “intelligent transportation systems” and sometimes called “intelligent traffic management,” are automated systems that incorporate the latest advances in Internet of Things (IoT) technology.

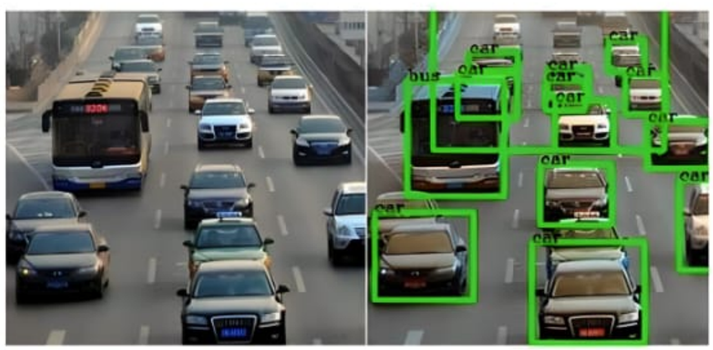
These systems can optimize traffic flow and enhance safety by using sensors, cameras, routers and cellular technology to dynamically adjust control mechanisms such as traffic lights, freeway on-ramp meters, bus rapid transit lanes, highway message boards and even speed limits.

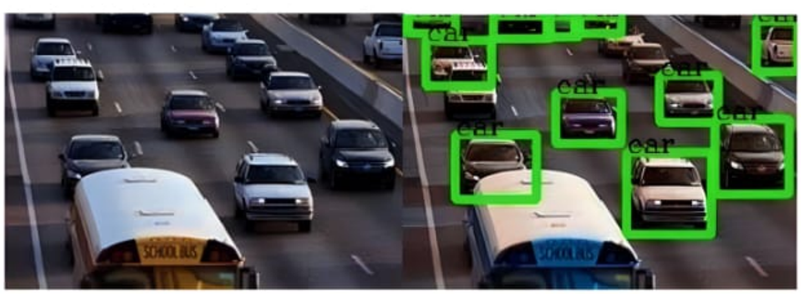
Today, Smart Traffic Management Systems make it possible to increase the capacity of city streets without actually adding new roads. With the advent of connected vehicle technology, these systems will also be able to directly control vehicles when needed — braking them in intersections, for example, to prevent accidents with pedestrians or other vehicles. Smart Cities are deploying these systems now to be prepared when the vehicle technology is fully tested and deployed.


Capturing Images using cctv installed at traffic signals
Detecting vehicles using image processing and caluclating traffic desnisty
Traffic density sent to server for calculating green signal time
Using this time scheduling is done
Traffic signal timer is updated

Following are some images of the output of the Vehicle Detection Module:







Signal Switching Algorithm

The Signal Switching Algorithm sets the green signal timer according to traffic density returned by the vehicle detection module, and updates the red signal timers of other signals accordingly. It also switches between the signals cyclically according to the timers.

The algorithm takes the information about the vehicles that were detected from the detection module, as explained in the previous section, as input. This is in JSON format, with the label of the object detected as the key and the confidence and coordinates as the values. This input is then parsed to calculate the total number of vehicles of each class. After this, the green signal time for the signal is calculated and assigned to it, and the red signal times of other signals are adjusted accordingly. The algorithm can be scaled up or down to any number of signals at an intersection.

The following factors were considered while developing the algorithm:

- The processing time of the algorithm to calculate traffic density and then the green light duration – this decides at what time the image needs to be acquired
- Number of lanes
- Total count of vehicles of each class like cars, trucks, motorcycles, etc.
- Traffic density calculated using the above factors
- Time added due to lag each vehicle suffers during start-up and the non-linear increase in lag suffered by the vehicles which are at the back [13]
- The average speed of each class of vehicle when the green light starts i.e. the average time required to cross the signal by each class of vehicle [14]The minimum and maximum time limit for the green light duration - to prevent starvation

$$GST = \frac{\sum_{vehicleClass} (NoOfVehicles_{vehicleClass} * AverageTime_{vehicleClass})}{(NoOfLanes + 1)}$$

Working of the algorithm

When the algorithm is first run, the default time is set for the first signal of the first cycle and the times for all other signals of the first cycle and all signals of the subsequent cycles are set by the algorithm. A separate thread is started which handles the detection of vehicles for each direction and the main thread handles the timer of the current signal. When the green light timer of the current signal (or the red light timer of the next green signal) reaches 0 seconds, the detection threads take the snapshot of the next direction. The result is then parsed and the timer of the next green signal is set. All this happens in the background while the main thread is counting down the timer of the current green signal. This allows the assignment of the timer to be seamless and hence prevents any lag. Once the green timer of the current signal becomes zero, the next signal becomes green for the amount of time set by the algorithm.

The image is captured when the time of the signal that is to turn green next is 0 seconds. This gives the system a total of 5 seconds (equal to value of yellow signal timer) to process the image, to detect the number of vehicles of each class present in the image, calculate the green signal time, and accordingly set the times of this signal as well as the red signal time of the next signal. To find the optimum green signal time based on the number of vehicles of each class at a signal, the average speeds of vehicles at startup and their acceleration times were used, from which an estimate of the average time each class of vehicle takes to cross an intersection was found. The green signal time is then calculated using the formula below.

$$GST = \frac{\sum_{vehicleClass} (NoOfVehicles_{vehicleClass} * AverageTime_{vehicleClass})}{(NoOfLanes + 1)}$$

where:

- GST is green signal time
- noOfVehiclesOfClass is the number of vehicles of each class of vehicle at the signal as detected by the vehicle detection module,
- averageTimeOfClass is the average time the vehicles of that class take to cross an intersection, and
- noOfLanes is the number of lanes at the intersection

The average time each class of vehicle takes to cross an intersection can be set according to the location, i.e., region-wise, city-wise, locality-wise, or even intersection-wise based on the characteristics of the intersection, to make traffic management more effective. Data from the respective transport authorities can be analyzed for this.

The signals switch in a cyclic fashion and **not** according to the densest direction first. This is in accordance with the current system where the signals turn green one after the other in a fixed pattern and does not need the people to alter their ways or cause any confusion. The order of signals is also the same as the current system, and the yellow signals have been accounted for as well.

Order of signals: Red → Green → Yellow → Red

Following are some images of the output of the Signal Switching Algorithm:

i. *: Initially, all signals are loaded with default values, only the red signal time of the second signal is set according to green time and yellow time of first signal.*

```
  GREEN TS 1 -> r: 0  y: 5  g: 1
    RED TS 2 -> r: 6  y: 5  g: 20
    RED TS 3 -> r: 131  y: 5  g: 20
    RED TS 4 -> r: 131  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 5  g: 0
    RED TS 2 -> r: 5  y: 5  g: 20
    RED TS 3 -> r: 130  y: 5  g: 20
    RED TS 4 -> r: 130  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 4  g: 0
    RED TS 2 -> r: 4  y: 5  g: 20
    RED TS 3 -> r: 129  y: 5  g: 20
    RED TS 4 -> r: 129  y: 5  g: 20

 Green Time:  9
 YELLOW TS 1 -> r: 0  y: 3  g: 0
    RED TS 2 -> r: 3  y: 5  g: 10
    RED TS 3 -> r: 128  y: 5  g: 20
    RED TS 4 -> r: 128  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 2  g: 0
    RED TS 2 -> r: 2  y: 5  g: 10
    RED TS 3 -> r: 127  y: 5  g: 20
    RED TS 4 -> r: 127  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 1  g: 0
    RED TS 2 -> r: 1  y: 5  g: 10
    RED TS 3 -> r: 126  y: 5  g: 20
    RED TS 4 -> r: 126  y: 5  g: 20

    RED TS 1 -> r: 150  y: 5  g: 20
  GREEN TS 2 -> r: 0  y: 5  g: 10
    RED TS 3 -> r: 15  y: 5  g: 20
    RED TS 4 -> r: 125  y: 5  g: 20

    RED TS 1 -> r: 149  y: 5  g: 20
  GREEN TS 2 -> r: 0  y: 5  g: 9
    RED TS 3 -> r: 14  y: 5  g: 20
    RED TS 4 -> r: 124  y: 5  g: 20
```

ii. *: The leftmost column shows the status of the signal i.e. red, yellow, or green, followed by the traffic signal number, and the current red, yellow, and green timers of the signal. Here, traffic signal 1 i.e. TS 1 changes from green to yellow. As the yellow timer counts down, the results of the vehicle detection algorithm are calculated and a green time of 9 seconds is returned for TS 2. As this value is less than the minimum green time of 10, the green signal time of TS 2 is set to 10 seconds. When the yellow time of TS 1 reaches 0, TS 1 turns red and TS 2 turns green, and the countdown continues. The red signal time of TS 3 is also updated as the sum of yellow and green times of TS 2 which is 5+10=15.*

```
 GREEN TS 1 -> r: 0  y: 5  g: 1
   RED TS 2 -> r: 6  y: 5  g: 20
   RED TS 3 -> r: 119  y: 5  g: 20
   RED TS 4 -> r: 134  y: 5  g: 20

YELLOW TS 1 -> r: 0  y: 5  g: 0
   RED TS 2 -> r: 5  y: 5  g: 20
   RED TS 3 -> r: 118  y: 5  g: 20
   RED TS 4 -> r: 133  y: 5  g: 20

YELLOW TS 1 -> r: 0  y: 4  g: 0
   RED TS 2 -> r: 4  y: 5  g: 20
   RED TS 3 -> r: 117  y: 5  g: 20
   RED TS 4 -> r: 132  y: 5  g: 20

Green Time:  25
YELLOW TS 1 -> r: 0  y: 3  g: 0
   RED TS 2 -> r: 3  y: 5  g: 25
   RED TS 3 -> r: 116  y: 5  g: 20
   RED TS 4 -> r: 131  y: 5  g: 20

YELLOW TS 1 -> r: 0  y: 2  g: 0
   RED TS 2 -> r: 2  y: 5  g: 25
   RED TS 3 -> r: 115  y: 5  g: 20
   RED TS 4 -> r: 130  y: 5  g: 20

YELLOW TS 1 -> r: 0  y: 1  g: 0
   RED TS 2 -> r: 1  y: 5  g: 25
   RED TS 3 -> r: 114  y: 5  g: 20
   RED TS 4 -> r: 129  y: 5  g: 20

   RED TS 1 -> r: 150  y: 5  g: 20
 GREEN TS 2 -> r: 0  y: 5  g: 25
   RED TS 3 -> r: 30  y: 5  g: 20
   RED TS 4 -> r: 128  y: 5  g: 20

   RED TS 1 -> r: 149  y: 5  g: 20
 GREEN TS 2 -> r: 0  y: 5  g: 24
   RED TS 3 -> r: 29  y: 5  g: 20
   RED TS 4 -> r: 127  y: 5  g: 20
```

iii. *: After a complete cycle, again, TS 1 changes from green to yellow. As the yellow timer counts down, the results of the vehicle detection algorithm are processed and a green time of 25 seconds is calculated for TS 2. As this value is more than the minimum green time and less than maximum green time, the green signal time of TS 2 is set to 25 seconds. When the yellow time of TS 1 reaches 0, TS 1 turns red and TS 2 turns green, and the countdown continues. The red signal time of TS 3 is also updated as the sum of yellow and green times of TS 2 which is 5+25=30.*

Simulation Module

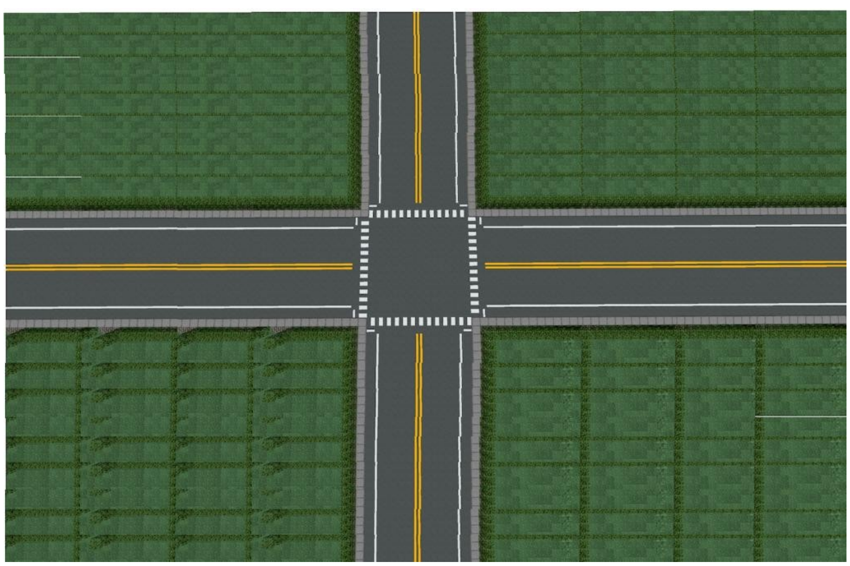
A simulation was developed from scratch using Pygame to simulate real-life traffic. It assists in visualizing the system and comparing it with the existing static system. It contains a 4-way intersection with 4 traffic signals. Each signal has a timer on top of it, which shows the time remaining for the signal to switch from green to yellow, yellow to red, or red to green. Each signal also has the number of vehicles that have crossed the intersection displayed beside it. Vehicles such as cars, bikes, buses, trucks, and rickshaws come in from all directions. In order to make the simulation more realistic, some of the vehicles in the rightmost lane turn to cross the intersection. Whether a vehicle will turn or not is also set using random numbers when the vehicle is generated. It also contains a timer that displays the time elapsed since the start of the simulation

Key steps in development of simulation

a. Took an image of a 4-way intersection as background.

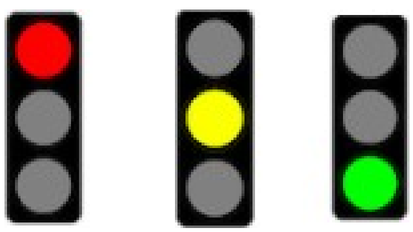
b. Gathered top-view images of car, bike, bus, truck, and rickshaw.

c. Resized them.



d. Rotated them for display along different directions.

e. Gathered images of traffic signals - red, yellow, and green.



f. Code: For rendering the appropriate image of the signal depending on whether it is red, green, or yellow.

g. Code: For displaying the current signal time i.e. the time left for a green signal to turn yellow or a red signal to turn green or a yellow signal to turn red. The green time of the signals is set according to the algorithm, by taking into consideration the number of vehicles at the signal. The red signal times of the other signals are updated accordingly.

h. Generation of vehicles according to direction, lane, vehicle class, and whether it will turn or not all set by random variables. Distribution of vehicles among the 4 directions can be controlled. A new vehicle is generated and added to the simulation after every 0.75 seconds.

i. Code: For how the vehicles move, each class of vehicle has different speed, there is a gap between 2 vehicles, if a car is following a bus, then its speed is reduced so that id does not crash into the bus.

j. Code: For how they react to traffic signals i.e. stop for yellow and red, move for green. If they have passed the stop line, then continue to move if the signal turns yellow.

k. Code: For displaying the number of vehicles that have crossed the signal.

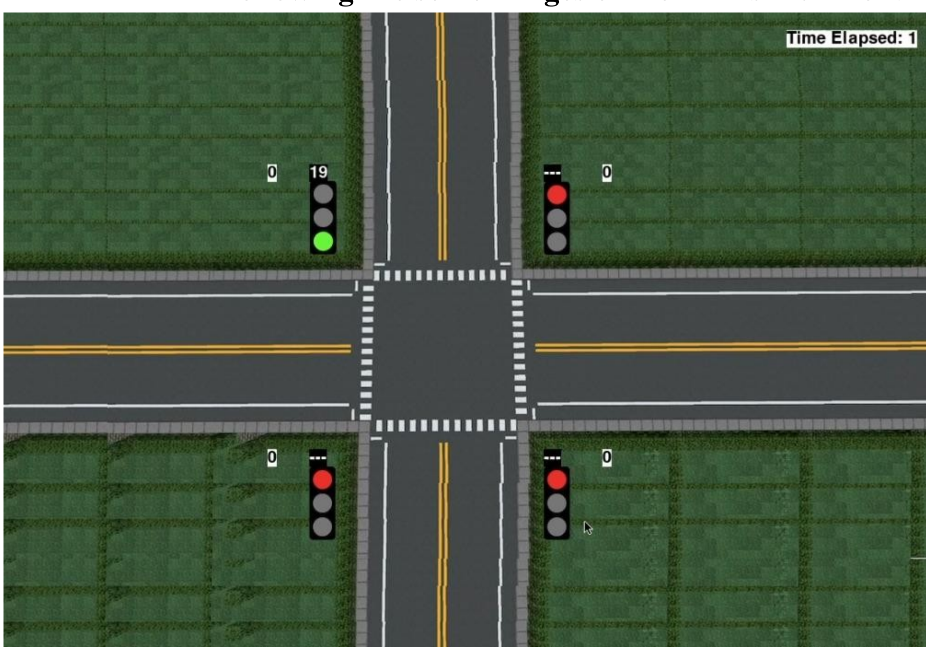
l. Code: For displaying the time elapsed since the start of the simulation.

m. Code: For updating the time elapsed as simulation progresses and exiting when the time elapsed equals the desired simulation time, then printing the data that will be used for comparison and analysis.

n. To make the simulation closer to reality, even though there are just 2 lanes in the image, we add another lane to the left of this which has only bikes, which is generally the case in many cities.

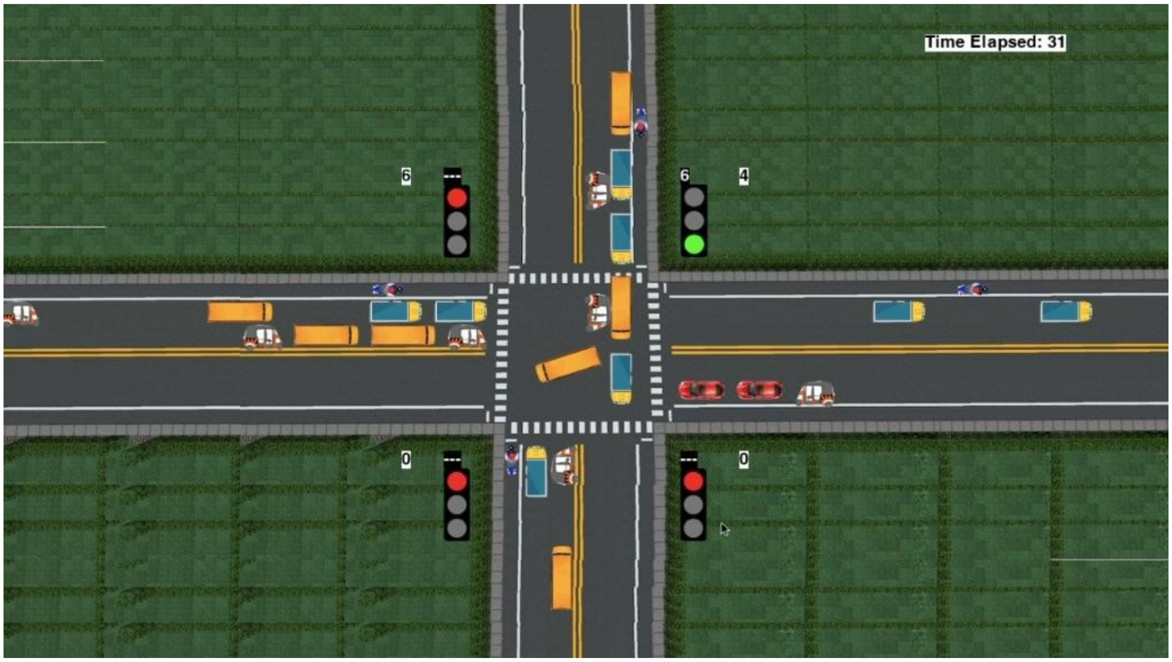
o. Vehicles turning and crossing the intersection in the simulation to make it more realistic.

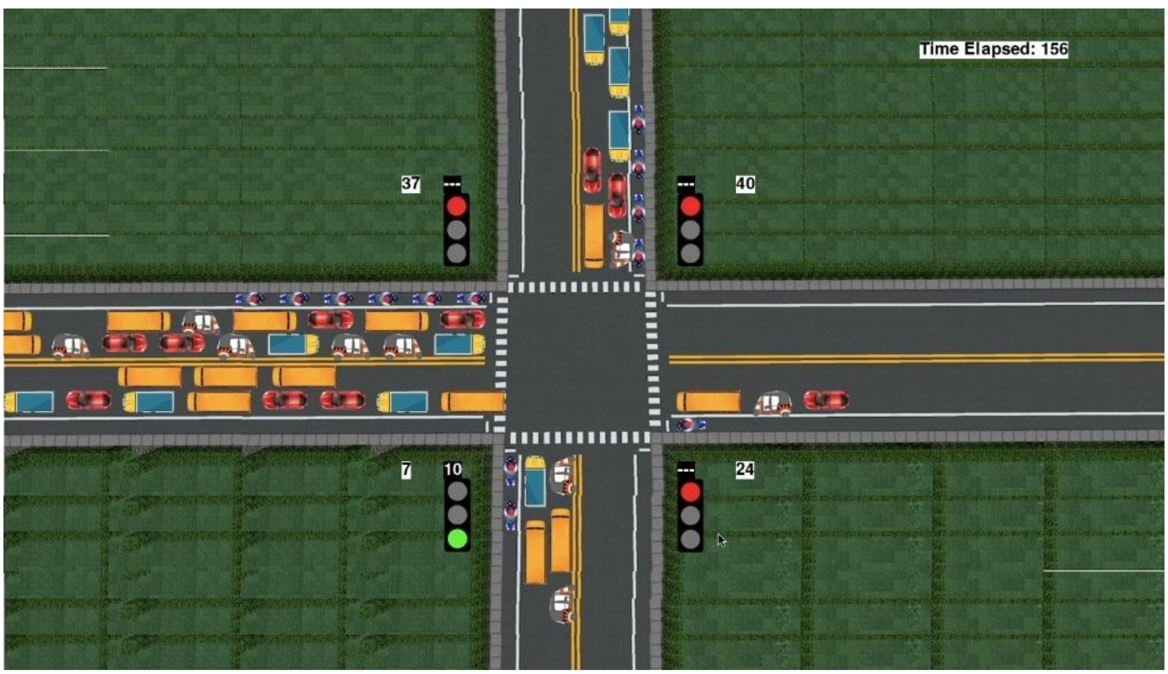
Following are some images of the final simulation:



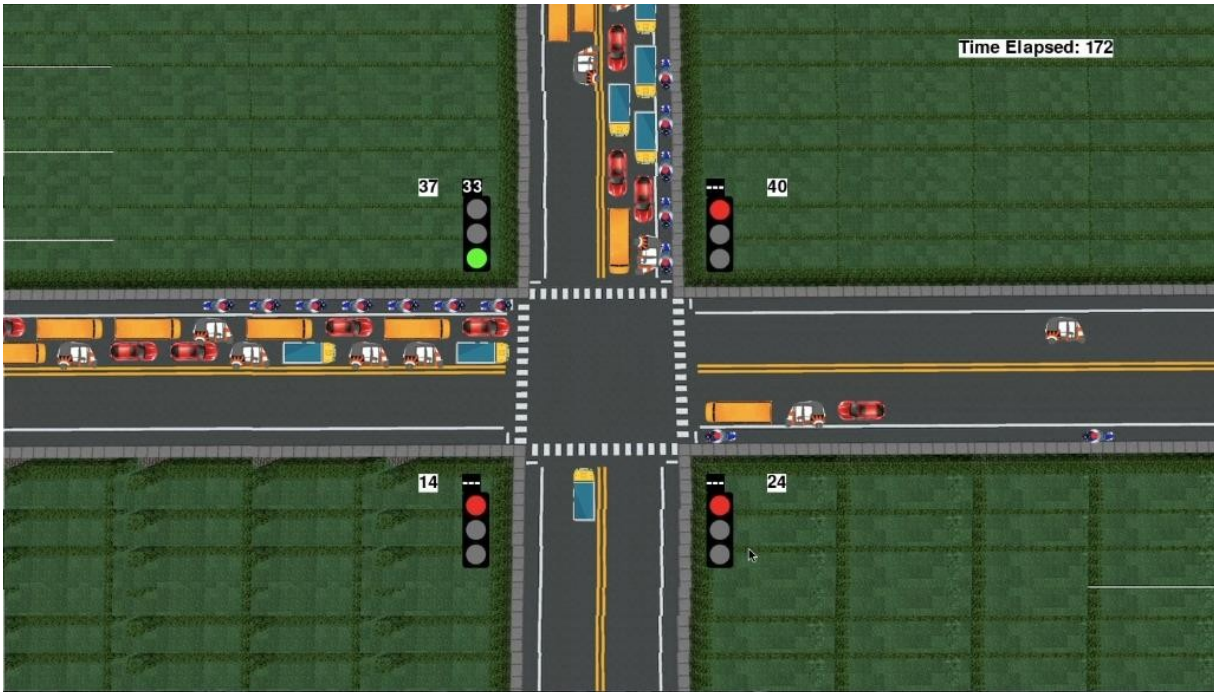
: *Simulation just after start showing red and green lights, green signal time counting down from a default of 20 and red time of next signal blank. When the signal is red, we display a blank value till it reaches 10 seconds. The number of vehicles that have crossed can be seen beside the signal, which are all 0 initially. The time elapsed since the start of simulation can be seen on top rightt.*

: Simulation showing yellow light and red time for next signal. When red signal time is less than 10 seconds, we show the countdown timer so that vehicles can start up and be prepared to move once the signal turns green.

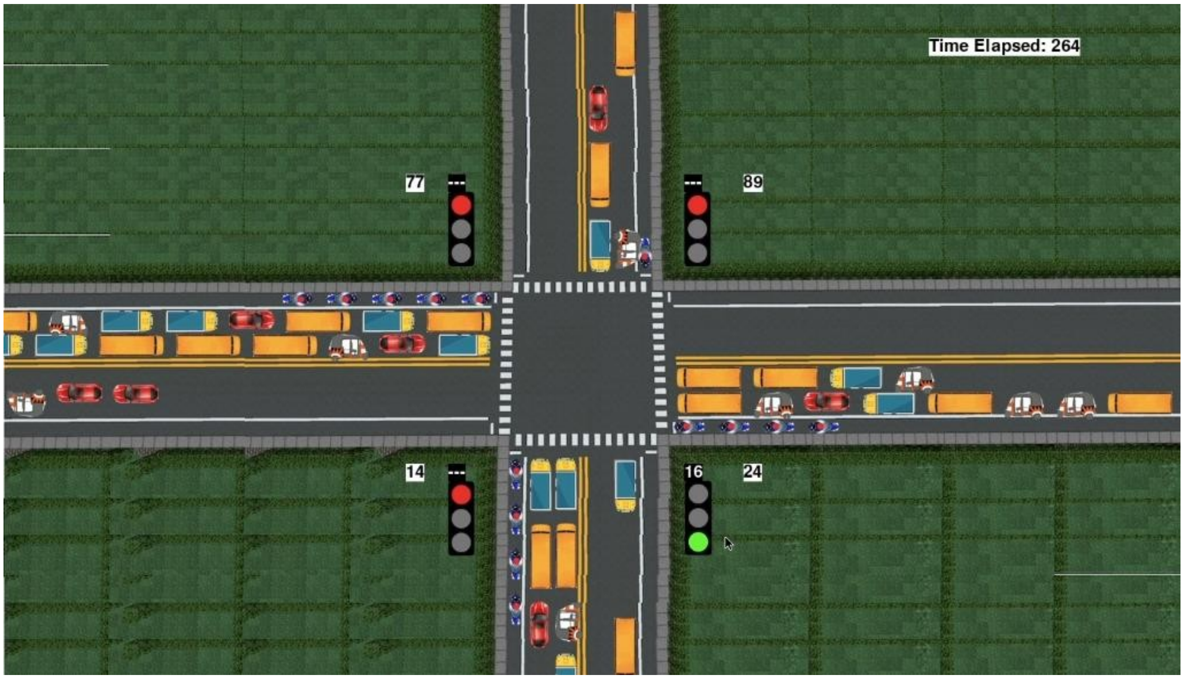




- *Simulation showing green time of signal for vehicles moving up set to 10 seconds according to the vehicles in that direction. As we can see, the number of vehicles is quite less here as compared to the other lanes. With the current static system, the green signal time would have been the same for all signals, like 30 seconds. But in this situation, most of this time would have been wasted. But our* *adaptive system detects that there are only a few vehicles, and sets the green time accordingly, which is 10 seconds in this case.*



- *Simulation showing green time of signal for vehicles moving right set to 33 seconds according to the vehicles in that direction.*



- : *Simulation showing green time of signal for vehicles moving left set to 24 seconds according to the vehicles in that direction.*

About need of relevant Internship in career development:

Relevant internships play a crucial role in career development for several reasons:

1. Practical Experience: Internships provide an opportunity to gain practical, hands-on experience in a specific field or industry. They allow you to apply the theoretical knowledge you've acquired in a real-world setting, helping you develop a deeper understanding of the work involved and the skills required.

2. Skill Development: Internships offer a platform to enhance and develop valuable skills that are relevant to your desired career path. You can learn industry-specific tools, technologies, and best practices, as well as develop transferable skills such as communication, teamwork, problem-solving, and time management.

3. Industry Exposure: Internships provide an insider's view of the industry or profession you're interested in. You can gain exposure to the day-to-day operations, company culture, and industry trends. This exposure helps you make informed decisions about your career path and can guide your future choices.

4. Networking Opportunities: Internships allow you to connect and network with professionals in your chosen field. Building relationships with mentors, colleagues, and industry experts can open doors to future job opportunities and provide valuable guidance and advice.

5. Resume Building: Having relevant internship experience on your resume demonstrates to potential employers that you have practical experience in the field and are proactive in your professional development. It can make your resume stand out and increase your chances of securing future employment.

6. Clarifying Career Goals: Internships can help you explore different areas within a specific field, giving you a clearer understanding of your interests and strengths. They allow you to test the waters and gain insight into whether a particular career path aligns with your goals and aspirations.

Overall, relevant internships provide a valuable platform for career development by offering practical experience, skill development, industry exposure, networking opportunities, resume building, clarity of career goals, and professional growth. They serve as a bridge between academic learning and professional employment, allowing you to make informed decisions about your future career path.

Brief about Your project/problem statement:

As an AI language model, I don't have personal projects or problem statements. However, I can assist you with understanding and formulating a problem statement for your project.

Participating in an internship or collaboration program with USC/UCT could provide the following opportunities:

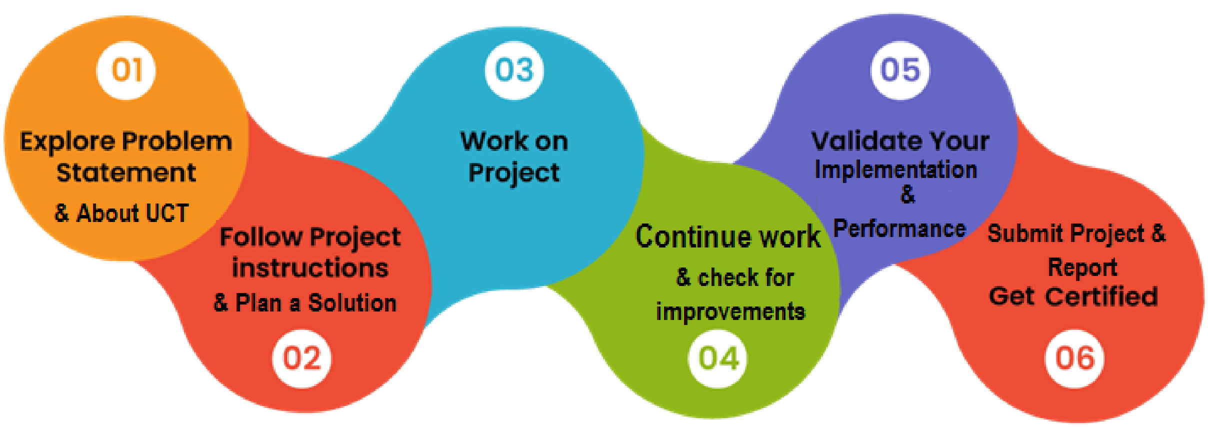
1. Industry Exposure: Working with a renowned institution and industry partner gives you exposure to real-world industrial problems, challenges, and practices. It allows you to gain insights into the operations and dynamics of the industry, which can be invaluable for your career.

2. Practical Experience: Internships and collaborations offer hands-on experience, enabling you to apply your knowledge and skills in a practical setting. Working on projects or problem statements provided by industry partners helps bridge the gap between theory and practice, enhancing your understanding and proficiency in your chosen field.

3. Networking: Collaborating with USC, UCT, and their industry partner gives you the opportunity to build a professional network. You can connect with professionals, mentors, and experts in the industry, which can lead to future job prospects, references, and valuable connections.

4. Skill Development: Engaging in projects and problem-solving within the context of an internship or collaboration provides an avenue for skill development.

How Program was planned:



documentation, and collaborative development workflows. These experiences will help you align your skills and practices with industry expectations.

understanding of industry practices. Remember to seek guidance from your mentors, collaborate with your peers, and make the most of the learning experience throughout the internship.

Thanks to Upskill Campus (USC) or The IoT Academy in collaboration with UniConverge Technologies Pvt Ltd (UCT) for this Internship Oppurtinity.

1 Introduction

1.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

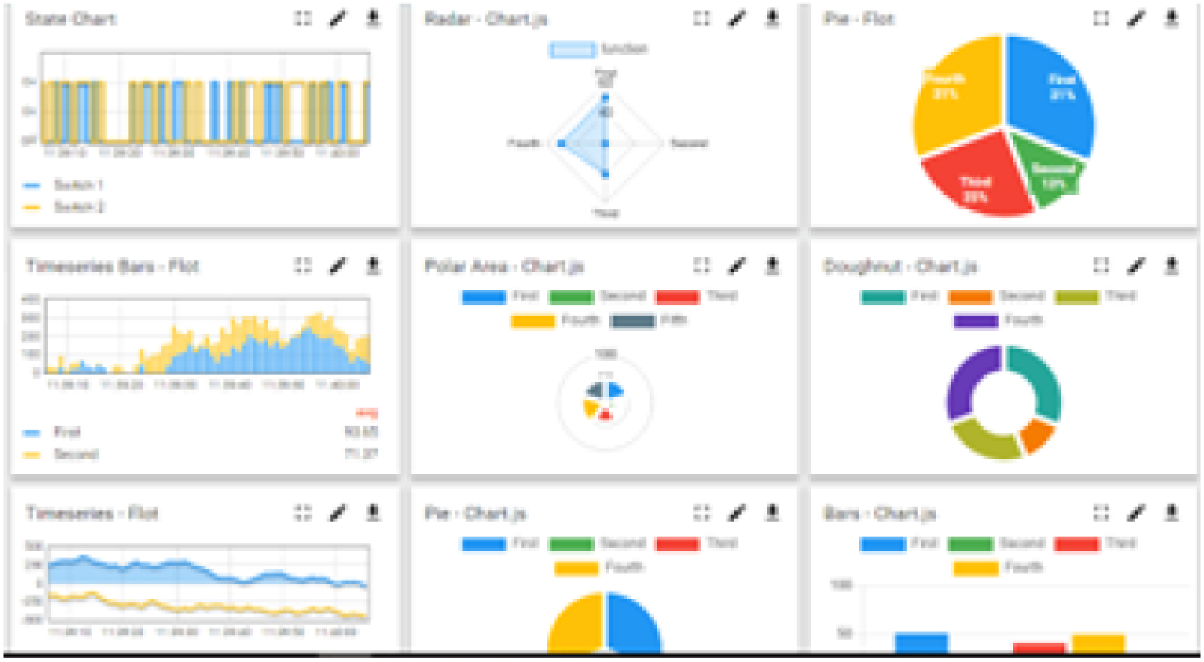
i. UCT IoT Platform ( )

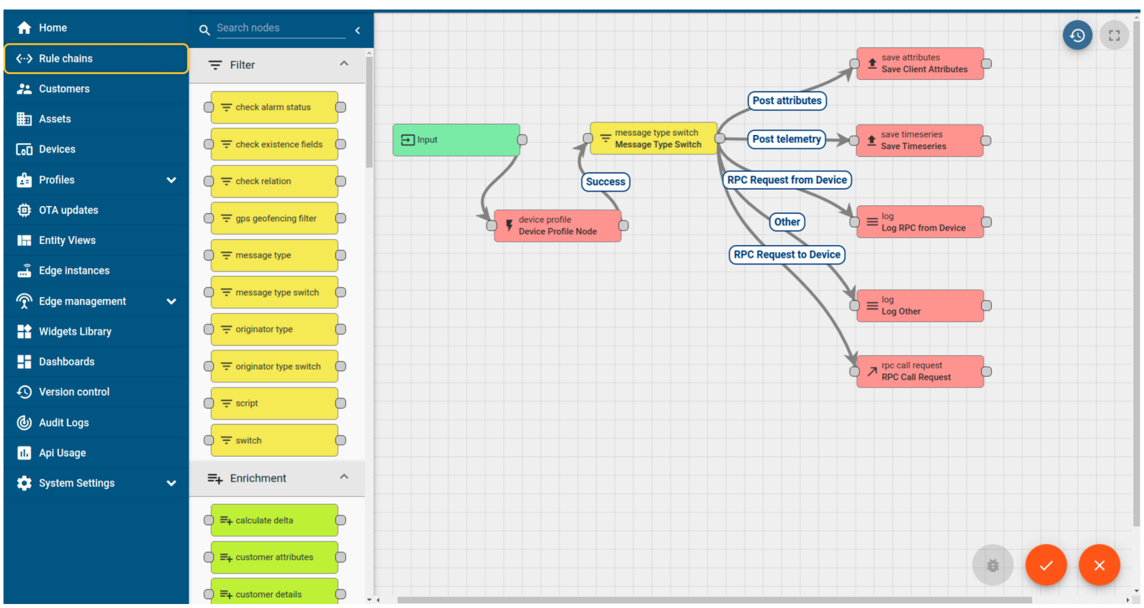
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to
• Build Your own dashboard
• Analytics and Reporting
• Alert and Notification

• Integration with third party application(Power BI, SAP, ERP)
• Rule Engine





ii. **Smart Factory Platform (** FACTORY WATCH **)**

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

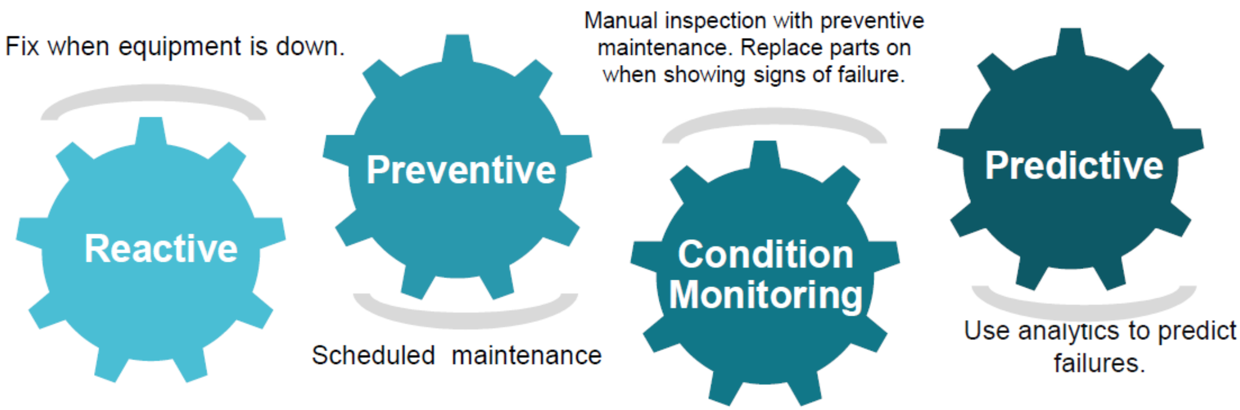
Its unique SaaS model helps users to save time, cost and money.

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

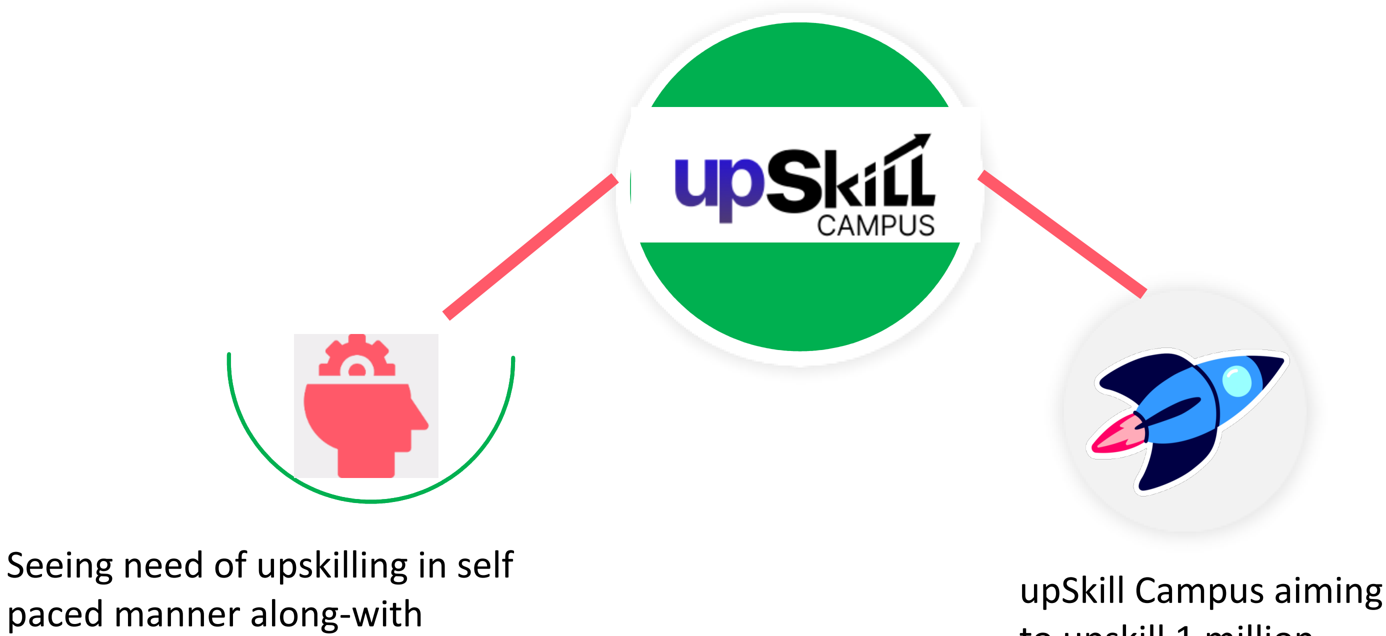
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



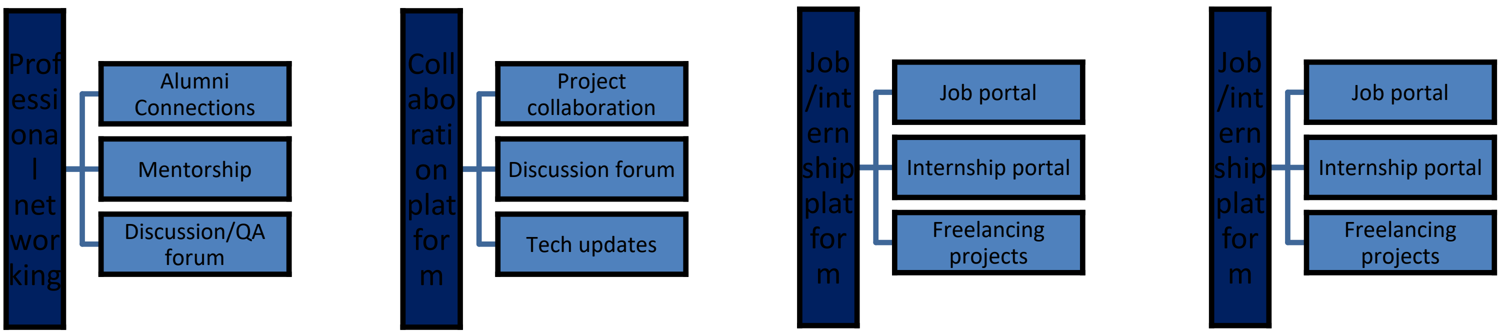
1.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



https://www.upskillcampus.com/



1.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

1.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

Thank You