Faculty of Engineering & Applied Science

ENGR 4940U
Capstone System Design I

Design of Smart Traffic Signals for Autonomous Transportation and Connected Communities in Smart Cities

Team Retrospective Report

Group#: 8
Section CRN#: 44900
Advisor: Dr. Hossam Gaber, P.Eng.
Coordinator: Dr. Vijay Sood, P.Eng.
Due Date: December 6th, 2022

First Name	Last Name	Student Number
Tirth	Patel	100751761
Vatsal	Patel	100728903
Saro	Karimi	100755079

Acknowledgement

We want to appreciate and sincerely thank our advisor Dr. Hossam Gaber for making this work possible. His guidance accompanies us through every phase of this project. We also want to express our gratitude to our coordinator Dr. Vijay Sood for providing us with inputs regarding the reports and the overall project. This project would not have been possible without your guidance and encouragement.

Finally, we would like to take this opportunity to express our profound gratitude to our parents for all the time and the money they have spent on us, but more importantly for their moral support throughout our studies.

Table of Contents

Abbreviations

NHTSA	National Highway Traffic System Administration
ITS	Intelligent Transportation Systems
V2X	Vehicle-to-everything
V2I	Vehicle-to-infrastructure
STS	Smart Traffic Signals

List of Figures

1. Introduction

Smart traffic signals are a solution to replace old and outdated traffic signals. Smart traffic combines conventional traffic signals with a variety of sensors and artificial intelligence to effectively route automobile and pedestrian traffic. Classis traffic signals have many disadvantages. For example, according to studies conducted in the United States by the National Highway Traffic System Administration (NHTSA), from 1997 to 2004, an average of 51% of fatal traffic collisions were caused by drivers who disregarded red signals at intersections with traffic lights. Also, rear-end collisions are another type of defect. Studies show that when a new classic traffic signal is installed, a sudden decrease in angle collision happens but more rear-end collisions start.

Smart traffic is a technology that municipalities may now quickly and affordably improve safety and traffic flow on their local streets into their traffic cabinets and intersections. Additionally, implementing these systems now, or modernizing the city's current Intelligent Transportation Systems (ITS) infrastructure, can result in significant cost savings, increased system reliability, and great for the drivers as well. Vehicle-to-everything communication (V2X) is used with Intelligent vehicle-to-infrastructure (V2I). It improves traffic situational awareness throughout the whole road network.

These signals can learn and update themselves by using the data they gather and process it which decreases human interactions. Also, the new system is much smarter, and by implementing decision-making methods, it would have more flexibility to change and adapt.

The scope of this project is to create and design an intelligent traffic signal management system for replacing the original systems. This system will help to increase the traffic flow, decrease accidents caused at intersections, have faster travel time, lower wait time, less consumption, etc. The system will be following a procedure that has three stages for now. Each stage has a different plan to implement the device and when we get to the final stage, the system will be fully operational. The STS includes many components and software. The sensors will be used, improved, or removed based on their use of them, but mainly the scoop of the project will be creating artificial intelligence to control the traffic flow by analyzing the data it captures. Also, this AI will be used on a blockchain which will be installed on the cars. The expectation to create a demo of this system will be in December 2022 and another demo in April 2023.

2. What went well

The most important thing that went well among us was the communication that we had with each other. Each member engaged in an open conversation with one another, exchanging ideas, opinions, and thoughts among fellow teammates, while also listening to what others had to say.

Instead of focusing only on the overall quantity of tasks getting completed, we reached an agreement that creates collaborative goals focused on outcomes and results. Then, a detailed strategy for how they will accomplish these goals collectively and with the assistance of each individual may be established. They now have a defined course to follow and a common goal to work toward.

Everyone in our group accomplishes their fair part of the job and knows exactly what they were responsible for and how they integrate into the operation of the project getting done. Whenever someone else needs support doing their task, the whole team is constantly ready to lend a hand. Teams frequently perform better when given organizational guidance and access to the necessary materials. So, these are the things that went well in our group from the beginning.

3. Things that surprised us

What surprised us was the ability to detect cars easily with python and a library called OpenCV. As you can see in the below pictures, by writing a few lines of code, we were able to create a simple machine-learning algorithm to detect the cars passing the camera and the output accuracy was off by one to two cars.

```python
def main():
    fgbg = cv2.bgsegm.createBackgroundSubtractorMOG()
    count = 0
    SKIP_FRAMES = 10
    trackers = []
    cars_in = 0
    # cars_out = 0
    Font = cv2.FONT_HERSHEY_COMPLEX_SMALL
    while True:
        ret, frame = cap.read()
        if not ret:
            break

        frame_r = cv2.resize(frame, (1280, 960))
        frame_r = frame_r[300:950, 200:1200]
        frame_cropped = frame_r
        # frame_cropped=frame_r[200:640,0:320]

        fgmask_r = fgbg.apply(frame_cropped)

        fgmask_r = cv2.dilate(fgmask_r, (9, 9), 2)
        cv2.imshow('fgmask', fgmask_r)
        contours, h = cv2.findContours(fgmask_r, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

        trackers_to_del = []
        for tid, trackersid in enumerate(trackers):
            trackingQuality = trackersid[0].update(frame_cropped)
            if trackingQuality < 5:
                trackers_to_del.append(trackersid[0])
```

```python
        try:
            for trackersid in trackers_to_del:
                trackers.pop(tid)
        except IndexError:
            pass

        if (count % SKIP_FRAMES) == 0:
            for num, cnt in enumerate(contours):
                area = cv2.contourArea(cnt)
                if area in range(400, 8000):
                    x, y, w, h = cv2.boundingRect(cnt)
                    rect = dlib.rectangle(x, y, x + w, y + h)
                    tracking = False

                    for trackersid in trackers:
                        pos = trackersid[0].get_position()
                        startX = int(pos.left())
                        startY = int(pos.top())
                        endX = int(pos.right())
                        endY = int(pos.bottom())
                        tx, ty = findCenter(startX, startY, endX, endY)
                        t_location_chk = pointInRect(x, y, w, h, tx, ty)
                        if t_location_chk:
                            tracking = True

                    if not tracking:
                        tracker = dlib.correlation_tracker()
                        tracker.start_track(frame_cropped, rect)
                        trackers.append([tracker, frame_cropped])
```

```python
        for num, trackersid in enumerate(trackers):
            pos = trackersid[0].get_position()
            startX = int(pos.left())
            startY = int(pos.top())
            endX = int(pos.right())
            endY = int(pos.bottom())
            offset = 0
            #cv2.rectangle(frame_cropped, (startX - offset, startY - offset),
                          #(endX + offset, endY + offset), (0, 255, 250), 1)

            if endX < 320 and endY >= 280:
                cars_in += 1
                trackers.pop(num)
```

```python
        count += 1
        cv2.putText(frame_r, f"IN:{cars_in}",
                    (20, 40), Font, 1, (255, 255, 255), 1)
        #cv2.putText(frame_r, f"OUT:{cars_out}",
                    #(550, 40), Font, 1, (255, 0, 0), 1)

        cv2.namedWindow("frame")
        cv2.moveWindow("frame", 0, 0)
        cv2.imshow('frame', frame_r)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
```

Figure 3.1: Screenshot of a simple machine learning algorithm

In the picture below you can see the output of how we see the cars and how the computer sees them. It helps to figure out this method to work on in the future to create a better system for our project.

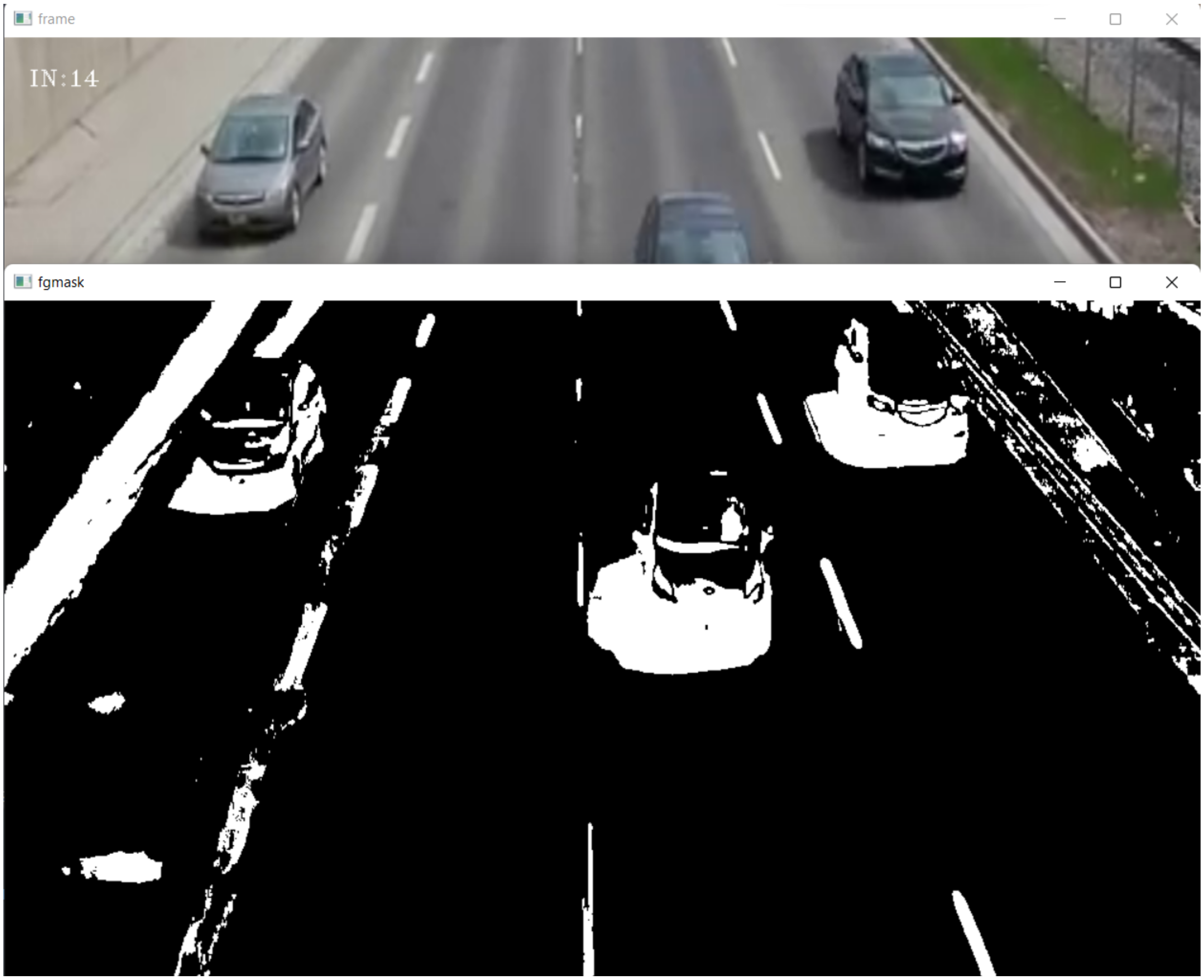

Figure 3.2: Detecting cars

We can combine this technique with the other things we learned and improve our skills in future of this project.

4. Lesson Learned

We learned many lessons while working on the capstone project. Firstly, we have come to understand that working together is more effective than working alone. To effectively complete each deliverable, we distributed tasks according to each other's strengths and weaknesses. We discovered that while not everyone contributes the same kind of work or has the same knowledge, the group's variety can be one of its greatest assets.

Secondly, we learned that time management and making a schedule is a key to completing each project phase on time, and encouraged us to consider what we want to achieve each day, week, and month. Also, we learned that if there is a delay in one part of the project phase, it can result in the whole project being delayed.

Lastly, we learned a technology called LoRaWAN which is a device using the Helium blockchain to connect the devices and have the ability to connect all these connected devices to the internet as well. This system helps us to connect cars since one of the abilities of this technology is that it can connect devices using only radio-based communication.

5. Conclusion

To conclude, our project combines conventional traffic signals with various sensors and artificial intelligence to effectively route vehicles and pedestrians at intersections. The STS can learn and update themselves by using the data they gather and process it which decreases human interactions. Also, the new system is much smarter, and by implementing decision-making methods, it would have more flexibility to change and adapt. We used tools such as the Gantt chart and learned many techniques that will help us complete future projects efficiently and effectively.