

**School of Computer Science**

**Masters in Applied Computing (M.A.C)**

**COMP8157**

**Advanced Database Topics**

**Professor Dr Shafaq khan**

**Project Proposal**

**by**

**Hamza Baig (110089314)**

**Arjun Kheni (110094003)**

**Meet Kevadiya (110092731)**

**Himani Rabari (110091002)**

Table of Contents

[Introduction: 3](#_Toc126087486)

[Problem statement: 3](#_Toc126087487)

[Motivation for solution: 3](#_Toc126087488)

[Methodology: 3](#_Toc126087489)

[Methodology for the project: 3](#_Toc126087490)

[SCRUM: 3](#_Toc126087491)

[Milestones for the project: 4](#_Toc126087492)

[Milestone 1 (February 28th): 4](#_Toc126087493)

[Milestone 2 (March 28th): 4](#_Toc126087494)

[Technology: 4](#_Toc126087495)

[References: 5](#_Toc126087496)

# Introduction:

The smart traffic system aims to improve road safety and reduce the number of road accidents by detecting traffic violations using computer vision. The system is designed to detect violations such as jumping a red light and record the plate number of the vehicles involved. The recorded information will be used to send an e-ticket to the violator.

# Problem statement:

As per Ontario Ministry of Road Transportation, there have been a significant reduction in road accidents in recent years. However, despite of global pandemic, 72,917 road accident occurred in Canada in 2020 which caused death of 1745 victims. Among which 33.2% of drivers were not wearing seatbelt, 25.3% were driving over the speed limit and 17.6% were impaired at the time of incident. Furthermore, distracted driving such as use of mobile phones or dashboard screen also caused death of 45 people in Ontario itself.

Running red lights, speeding, and illegal parking have become major issues that cause traffic jams and accidents on the road. An automated system is required to correctly identify and implement these infractions. The aim is to create a system that is capable of:

* Determine the type of crime (e.g., running red light, speeding)
* Find the offending car by scanning licence plate
* Give proof of the infraction and fine accordingly (such a photo or video of the incident)

# Motivation for solution:

The use of computer vision technology in the smart traffic system offers several advantages over traditional enforcement methods. The system is quick and efficient in detecting violations and recording the information of the violator. It reduces the need for manual enforcement and minimizes human error. Moreover, the e-ticketing system makes it convenient for violators to pay their fines without having to visit a physical location.

# Methodology:

The advanced traffic management system will employ computer vision algorithms and deep learning to identify traffic infractions. The cameras situated at crossroads will take pictures which will then be inputted into the system for evaluation. The system will utilize image processing methods to recognize the existence of a car at the intersection and determine whether it ran a red light. The license plate number of the vehicle will be extracted and documented. The collected information will be utilized to create an electronic ticket and send it to the offending party.

## Methodology for the project:

### SCRUM:

1. Form a cross-functional team: Scrum requires a cross-functional team with all the necessary skills to complete the project. This includes developers, testers, project managers, and stakeholders.
2. Define product backlog: A product backlog is a prioritized list of features and requirements that your team will work on. Create a product backlog for your smart traffic system project that includes all the necessary features and requirements.
3. Plan sprints: Sprints are short, time-boxed iterations of work that last 1-4 weeks. Plan sprints for your project, setting goals for each sprint and prioritizing the items in your product backlog.
4. Hold daily stand-up meetings: Scrum requires daily stand-up meetings to ensure that the team stays on track and that everyone is aware of what others are working on.
5. Review and retro: At the end of each sprint, hold a sprint review meeting to demonstrate what was completed during the sprint. Also, hold a sprint retro meeting to discuss what went well and what could be improved in the next sprint.
6. Repeat the process: Repeat the process of planning, developing, and reviewing for each sprint until all the items in the product backlog have been completed.

## Milestones for the project:

### Milestone 1 (February 28th):

1. Develop the computer vision algorithms: Collect dataset or prepare a new dataset for training the computer vision model.
2. Develop the computer vision algorithms: Choose appropriate computer vision algorithms to detect traffic violations. Create model for extracting license plate from images.
3. Implement the system components: Start implementing the system components, including the camera, computer vision algorithms, and data storage/database and retrieval system.
4. Test the system: Test the system components to ensure they work as expected.
5. Evaluate the performance: Evaluate the performance of the computer vision algorithms and make any necessary improvements.

### Milestone 2 (March 28th):

1. Integrate the system components: Integrate the camera, computer vision algorithms, and data storage and retrieval system into a single system.
2. Test the integrated system: Test the integrated system to ensure it functions as intended.
3. Deploy the system: Deploy the system in a real-world environment and monitor its performance.
4. Evaluate and refine the system: Evaluate the performance of the system and refine it based on the results.
5. Prepare documentation: Prepare comprehensive documentation on the system, including its design, implementation, and performance.

# Technology:

* Computer Vision: OpenCV, TensorFlow
* License Plate Recognition: ALPR API (Automatic License Plate Recognition)
* Stop sign and Speed detection of road: Google Roads API
* Object Detection: YOLOv3, Faster R-CNN
* E-ticketing System: Node.js, Laravel
* Database: Firebase

# References:

1. OpenCV Library, "OpenCV - Open Source Computer Vision Library," [Online]. Available: https://opencv.org/. [Accessed: 31-Jan-2023].
2. TensorFlow, "TensorFlow - An Open Source Machine Learning Framework," [Online]. Available: https://www.tensorflow.org/. [Accessed: 31-Jan-2023].
3. PyTorch, "PyTorch - An Open Source Machine Learning Library," [Online]. Available: https://pytorch.org/. [Accessed: 31-Jan-2023].
4. Node.js, "Node.js - A JavaScript runtime built on Chrome's V8 JavaScript engine," [Online]. Available: https://nodejs.org/. [Accessed: 31-Jan-2023].
5. Laravel, "Laravel - The PHP Framework For Web Artisans," [Online]. Available: https://laravel.com/. [Accessed: 31-Jan-2023].
6. Django, "Django - The High-Level Python Web Framework," [Online]. Available: https://www.djangoproject.com/. [Accessed: 31-Jan-2023].