A Proposal and Experiment on Traffic Modeling for Use in Traffic Signal Optimization

By

Michael Haertling (mrh4886)

Case Studies in Machine Learning

December 2, 2024

Table of Contents

Background and Motivation

Commuters encounter traffic signals on a daily basis as they are a primary means of managing intersections in the United States. While they are critical for coordinating traffic, they can also be a source of great inefficiency. Traffic signals can be classified into one of four groups:

- Fixed-Time Control: Each signal transition has a fixed delay before activation.
- Pretimed Control: Similar to fixed-time control, but multiple timing schedules can be applied, of which one is active at any given time of day.
- Actuated Control: A simple sensor is employed to detect where vehicles are present. This information then provides active feedback to the signal controller. There are multiple algorithms for incorporating this feedback into signal timings.
- Coordinated Control: Timing schedules between adjacent signals are coordinated to allow a more continuous movement of traffic and minimize the number of stops in aggregate.
- Adaptive Control: Timings are dynamically adjusted based on real-time conditions. This method requires more advanced sensors and advanced algorithms to make real-time decisions.

(Son & Jin, 2024, 1)

The adaptive control provides the most flexibility and opportunity for optimization, but it is the least adopted control system because it requires specialized, licensed controller software and more advanced sensing hardware that carries a significant maintenance cost. As early as 1997 there was research in real-time signal transition controllers using neural networks. (Spall & Chin, 1997) One such system actively in use is the Sydney Coordinated Adaptive Traffic System (SCATS), which was developed by the New South Wales government and has been made available commercially, world-wide.

The product is deployed in over 200 cities where varying success has been achieved. Some studies have been published on the effectiveness of this solution. The SCATS marketing homepage advertizes a 28% reduction in travel time and a 12% reduction in fuel consumption. (*Scats*, n.d.) An Australian study boasts the same 28% reduction in travel times. (*SCATS Study Shows Significant Savings*, 2013) A case study performed in Portland clocked idle time reductions at more mild levels of less than 20% even in the best case street during peak traffic hours. (Slavin et al., 2013, 123) Still, improvements were found to be positive in aggregate. When deployed in highly congested areas, there is evidence of significant improvements using SCATS. It is clear that traffic signals in many current cases operate at far from optimal efficiency. But regardless of the benefit, the cost of implementing an adaptive signal system is often prohibitive. The University of Detroit Mercy reported that SCATS costs roughly $20,000 (20%) more to install and $5,000 (125%) more to maintain per mile per year compared to pre-timed signal intersections. (Dutta et al., 2013) Over time, more competition will likely enter the space, driving costs down. Further optimization may be achieved by real-time integration with semi and fully autonomous vehicles. It is also possible that traditional traffic signals become obsolete as transportation options continue to evolve, eventually rendering current solutions irrelevant. (Yaqoob et al., 2019)

Problem Statement

Rather than a high-cost, transformative solution, as is provided by SCATS and equivalents, I propose the use of a general purpose system for optimizing existing infrastructure. Traffic patterns are usually repetitive and therefore semi-predictable in aggregate. The largest benefit of an adaptive control system lies more in the ability to tailor a policy for each specific intersection rather than the autonomous adjustments of that policy over long periods. There may be drift over time as the surroundings change and throughput requirements wax and wane. It is standard practice to periodically tune signal policies (Koonce & Kittelson & Associates, 2008). Still, a well-considered, customized signal policy can remain reasonably optimal for years. Significant, traffic altering changes such as new roads, large corporate installations, or residential projects can be flagged and surrounding signals scheduled for adjustment proactively.

Buch et al. (2011) highlights emerging methods of object categorization and 3-D modeling for use in the analysis and optimization of urban traffic. Since 2011, significant advancements in machine vision have been realized. The fundamental difficulty in optimizing signals is the accurate modeling of the traffic situation. Challenges in measuring traffic improvements are even highlighted in the Australian SCATS effectiveness report. (*SCATS Study Shows Significant Savings*, 2013) Hardware will be required in most cases to collect the appropriate data, but it does not need to be a permanent fixture with perpetual maintenance and licensing costs. Additionally, the hardware does not need to be installed on all signals simultaneously. A policy of cycling the installation across patches of signals will keep costs down, while still providing the needed data. Once all targeted signals have been processed, the hardware can be stored indoors until a periodic refresh is required, improving the longevity of the equipment and further amortizing costs.

Signal Policy Evaluation Metrics

Leitner et al. (2022) highlights the need for a systematic review framework in evaluating traffic signal policies. Current approaches vary significantly depending on the data available and region of analysis. The above SCATS studies rely on travel time, number of stops, fuel consumption, idling time, and speed differentials. A recent reinforcement learning based traffic optimization paper uses waiting times, time loss, emergency stops, and departure delays in its feedback loop. (Son & Jin, 2024) Despite differences in presentation, these metrics are ultimately aggregate measures of individual vehicle experiences interacting with the signal system. There are two fundamental dimensions of measure:

- How quickly can a vehicle traverse the intersection? This is the most obvious vector of optimization as it contributes to an individual's overall commute time and traffic congestion levels. It is important to be careful with this metric as blind optimization along this dimension in intersections with much higher throughput in one direction can result in untenably long wait times for the few. It is common to use a quadratic or exponential cost function to avoid such scenarios when dealing with wait times.
- How smoothly can a vehicle traverse the intersection? This dimension impacts the comfort of individual commuters and the fuel/emission efficiency effected by a policy. Highly granular and accurate data is required to appropriately measure along this axis.

Environment Model Structure

A model of traffic behavior at an intersection can take many forms depending on the raw data available and the processing techniques employed. The model itself directly impacts which optimization techniques can be used and what evaluation metrics can be calculated. One ideal is a top-down, two dimensional representation including the road pathing, signal positioning, and bounding boxes for each vehicle. A detailed model can be initialized into a simulation software like the open source Simulation of Urban MObility (SUMO). SUMO supports modeling of both vehicles and pedestrians with tools for route finding, visualization, and emission calculation. There has also been recent research into a more robust transportation simulation called LimSim, which aims to overcome the limitations of SUMO. (Wen et al., 2024)

A simulation enables multiple objectives:

- Bulk testing of policy: Policies can be virtually applied to modeled signals and the vehicle pathing algorithm will simulate how human drivers would interact with the system.
- Scenario interpolation: It is impractical to perfectly capture the movement of all objects in the vicinity of a traffic signal. The built-in pathing algorithms can be leveraged to fill in the gaps with realistic, predicted behavior.
- Scenario joining: The policies of adjacent traffic signals can affect each other. This is the motivation behind coordinated control systems. Data captured separately at signals can be joined together to model multiple signals simultaneously, enabling joint optimization of signal timing schedules.
- Scenario rendering: A scenario can be rendered into a video, making it much easier to visually inspect the impact of adjustments to signal policies.
- Measurement of evaluation criteria: Every vehicle is simulated, making it possible to track all critical metrics such as number and duration of stops and speed. These metrics can then be aggregated as appropriate to achieve the meaningful, empirical evaluation metrics.

Environment Sensing

The following are desirable attributes of a traffic signal environment sensing system for use in modeling traffic patterns at an intersection:

- Portable: The system should be easily installed and uninstalled. This minimizes the cost to reuse hardware across large regions and makes more regular deployments, to capture gradual pattern updates, more economical.
- Anonymous: This system should have inbuilt safeguards to guarantee the privacy of individuals included in the data capture process. This prevents misuse of this data for the purpose of locating and tracking persons. Protected personal information includes license plates, facial images, decals, and stickers.
- Scalable: This system should be widely deployable with consideration to hardware, network, software, and cloud compute costs.
- Robust: This system should provide adequate data to produce a simulated model of the intersection's traffic patterns with enough accuracy to meaningfully capture its unique aspects. This system should support both vehicular and pedestrian traffic.
- Accessible: Data collection and aggregation should require little to no manual effort and be available within days of collection. The system should produce a consolidated dataset suitable for use in a variety of optimization algorithms.

The many complexities of city interactions can be divided into layers that can be modeled independently but interact with each other. ("Understanding Cities With Machine Eyes: A Review of Deep Computer Vision in Urban Analytics," 2020, 6) This use case focuses on the human and transportation/traffic interactions and therefore must be capable of sensing both vehicles and pedestrians to be generally applicable in high-traffic areas.

Input Hardware

Hardware selection is the first and most limiting decision. Many traffic signals already have actuation mechanisms that detect if a vehicle is present and may even track the number of vehicles that pass the threshold. If the pedestrian crossing has a button based activation, data could be collected on frequency of use. Unfortunately, relying on these existing mechanisms presents some challenges:

- Existence - Most signals do not utilize an actuation mechanism.
- Collection - Often, this data is not persisted beyond the feedback to the signal system. The system would need to be retrofitted with additional hardware to facilitate collection.
- Retention - The data, if stored, is often on a short retention policy.

- Reliability - These mechanisms can be old and poorly maintained.

Similar, specialized hardware combinations could be devised that may be capable of providing sufficient data to support the full requirement list, but collection inevitably becomes more complex as wider and more intricate intersections are considered. Simplicity and cost of installation are likely to suffer.

To avoid these challenges, I propose reliance on the optical analysis of camera feeds. Cameras are reasonably cheap, accessible, and often already deployed at and around traffic signals. However, this requires far more consideration in how the data is processed compared to a system reliant on simpler inputs. This approach does offer the potential for higher quality and granularity of derived data. Machine vision is a heavily researched field with a wide variety of well-tested, productionized solutions available. Additionally, the system can be updated as the industry progresses without the need to replace the hardware.

Data Processing Architecture

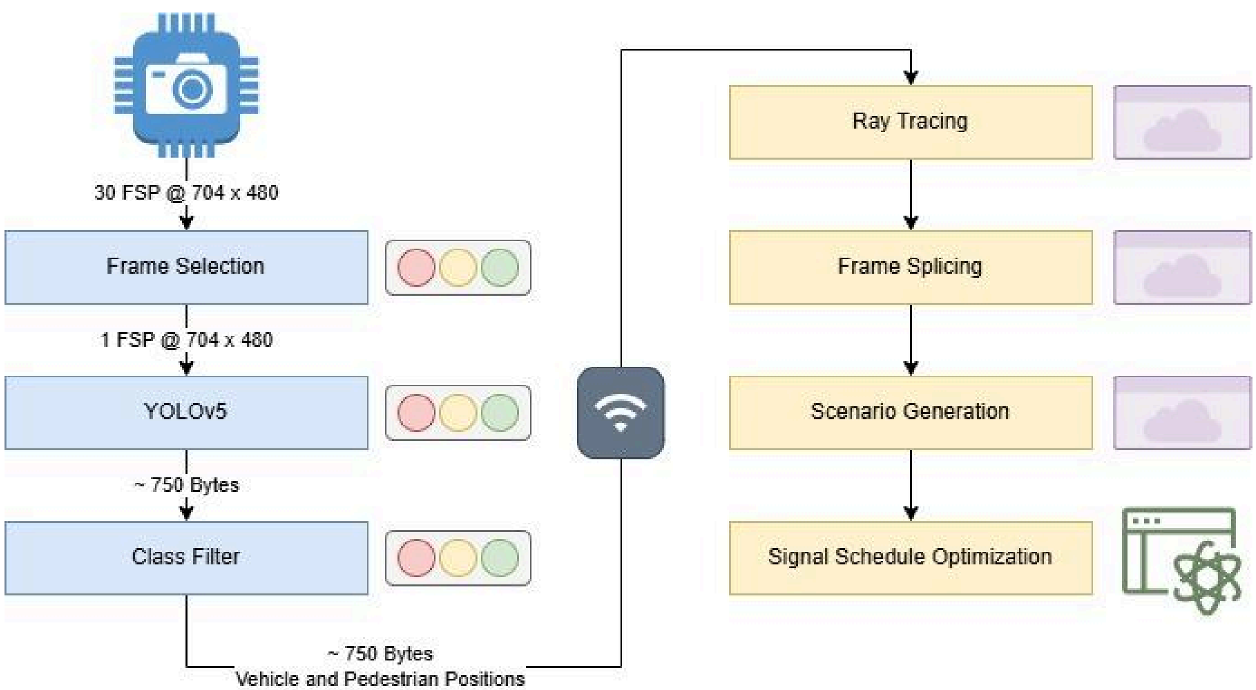

Figure 1. Data Flow Architecture Diagram

Given the non-real-time nature of this problem, a centralized, cloud approach has a few benefits including ease of software updates and minimized hardware on site. Privacy, however, is a common driver for EdgeAI solutions. (Pandit, 2024) The proposed design prevents the need to send potentially sensitive information to the cloud. The modularity of this design makes it straightforward to migrate to a centralized compute approach or even offer both modalities to be selected based on user need and preference.

As depicted in *Figure 1*, the light-weight YOLOv5 model is deployed locally at the intersection. A single processing unit can be deployed, which will receive video streams from all intersection cameras. This edge AI node works in favor of multiple key objectives:

- Scalability: The throughput of data to be streamed to the cloud is massively reduced. Even at the low resolution of 704 by 480, each frame is roughly 48 KB. A full 30 FPS stream would be nearly 1.5 MB per second per camera. This can be reduced by applying frame selection locally, but we still see a multi-order-of-magnitude decrease in network requirements. This makes large scale deployments much more tenable.
- Anonymity: The raw camera feed, which may contain personally identifiable information, is not persisted anywhere except short term memory, locally at the intersection. The bounding box and metadata information sent to the cloud has effectively been completely anonymized, preventing the misuse of this data by design.

1. Camera Input (Local)

One or more raw camera feeds are expected to stream to this data pipeline. Resolution does not need to be particularly high to accurately identify and classify objects. A standard HD input is more than sufficient and will actually be downsampled to keep processing requirements minimal. Older, low-resolution inputs are also acceptable.

2. Frame Sampler (Local)

Frame sampling settings can be hard-set or automatically adjusted within a range depending on processing capabilities and the number of cameras being streamed. It is critical that the traffic speed limit be considered in this configuration. High-speed intersections may require a higher sampling frequency to ensure that all vehicles are captured in at least two frames.

3. Object Detection and Classification (Local)

The objective of this module is to locate and classify all vehicles and pedestrians in a given video frame. This is a well-studied machine vision problem but the constraints requiring a light-weight solution limit the processing resources that can be committed to this task. Many traditional, mathematical methods of performing object detection are too slow for use in an edge AI solution.

Fortunately, advances in convolutional network solutions have resulted in viable alternatives. Sufficiently deep networks are able to outperform traditional methods on challenging datasets. (Krizhevsky et al., 2012, 8) Neural network based models typically fall into one of two categories: two-stage and single-stage detectors. Two stage detectors generally take longer to generate proposals, but the more complex architecture enables better granularity in identifying an arbitrary number of objects. (Zaidi et al., 2021, 7)

I narrowed down to two potential options that can accommodate the speed requirements, both of which are deep learning based model architectures capable of bounding and classifying many objects in each image.

- Faster RCNN: This model utilizes convolutional neural networks to perform region proposals (RPN) and object classification. Unlike its predecessors, it uses image pyramids and anchor boxes to fit bounds to object proposals. (Zaidi et al., 2021, 8) This FCNN approach is a gateway to achieving near-real-time throughput capabilities by re-using convolutional outputs across region proposals, minimizing duplicative processing efforts. (Ren et al., 2016, 3)
- YOLOv5: The original You Only Look Once (YOLO) model takes the unification seen in the Faster RCNN even further by merging all modules into a single, contiguous CNN. This produces a much simpler model that can be trained as a single module. (Redmon et al., 2016, 2) Further improvements have been made to the model over subsequent years from the initial release in 2015. Ultralytics has released a version 5 of the YOLO model, which is used for testing in this paper. (*Ultralytics/yolov5: V7.0 - YOLOv5 SOTA Realtime Instance Segmentation*, 2022)

Faster RCNN was tested using the PyTorch fasterrcnn_resnet50_fpn_v2 model from the torchvision package. While it does seem to do better at identifying clusters of objects, significant hardware would be needed to run this model across multiple video streams locally. For that reason, detailed implementation, testing, and analysis has been focused on the YOLOv5 model. There is also documented success in performing vehicle detection using the YOMOv5 model. (Zhang et al., 2022)

4. Class Filter (Local)

Only relevant classes need to be forwarded to the cloud, namely all vehicle and pedestrian variants.

5. Model Building (Cloud)

The collected data can now be processed into scenarios. First, baseline data will need to be collected to recreate the scene itself:

- Positioning and field of view of each camera (location and view vector)
- Positioning of each traffic signal
- Timing schedule of each traffic signal
- Speed limits on visible roads
- Road curvature

Depending on the complexity of the intersection, this modeling can be done in any 3D modeling software to make the tuning process quick and intuitive. With a little rendering work, the road bounds can be overlaid onto a sample frame image from each camera to validate the final configuration. This helps to ensure that the camera perspective is modeled as closely as possible which can have a significant impact on the resulting scenario accuracy. This baseline scene model only needs to be generated once because it will be shared across every frame rendered because the camera perspective is static. Vanhoey et al. (2017) produced a full 3D reconstruction of a whole city in Switzerland using a wide variety of machine vision inputs, proving feasibility in semi-autonomous, large scale city modeling.

Next, the objects identified in each sampled frame must be mapped onto a 2D plane representing the ground. Given the above scene information, the vector extending from the camera's perspective toward a given pixel can be traced to the intersection with the ground. This structure is depicted well in Figure 2 of (Pérez et al., 2016, 4), where the object is the ground. The transformation matrices representing the pinhole model should be configured using the contextual information available in the scene model. This context with simplifying assumptions preclude the need for stereo vision photogrammetry. (Pérez et al., 2016, 5)

A standard pixel offset from the true bounding box center should be manually configured for each camera to improve center mappings with little additional system overhead. The class of each object, car or pedestrian, should dictate what type of object is instantiated. Vehicles should be further classified as small, medium, large, or semi-truck based on the area of the bounding boxes. This can be done by using a clustering algorithm (like k-means) with four possible classes, after normalizing the vehicle sizes with respect to their distance from the camera. This sub-classing system can easily be expanded to support more or less vehicle types with minimal effort.

6. Frame Splicing (Cloud)

Objects must be correlated between adjacent frames to enable final model rendering. Each object should receive a unique identifier that remains static between frames. Traditionally, techniques such as mean shift, Scale-invariant feature Transform (SIFT), Kalman filters, and other types of optical flow analysis are used to make correlations between contiguous frames of a video. The Lucas-kanade Optical Flow method, for example, assumes that adjacent frames are approximately constant within a small area and uses linear algebra to correlate objects between frames. (Patel & Upadhya, 2013, 8) This assumption enables superior processing to many alternatives. These techniques are not directly applicable to our use case because the video input is no longer available at this processing step, but inspiration from these approaches can be garnered.

Despite the lack of visual context, various aspects specific to this use case can be leveraged to make up that deficit.

- Bounding box sizes: Given the model of the read, the expected change in bounding box size between two frames for a specific object can be estimated with reliable accuracy. The primary caveat of this measure is vehicles coming in and out of occlusion may experience dramatic changes in bounding box sizing.

- Color samples: One or more color samples collected within the bounds of each object are included in each object metadata that is sent to the cloud. While the colors will not be identical between frames, it can be expected that the proportions of red, green, and blue will be similar between adjacent frames. This supplies a strong correlation indicator that requires very little processing to utilize.
- Traffic direction expectations: A detected vehicle's mapping to the road model yields an expectation of what direction the vehicle is likely moving in. This can be used to strongly discourage any correlation proposals counter to this rule.
- Traffic speed expectations: Similar to traffic direction, the speed limit is known and can be used as a contributor to the overall confidence score of each correlation proposal. Additionally, if the speed of an object in previous frames is known, it can be expected that the speed in subsequent frames is unlikely to change dramatically.

Environment Sensing Experiment

Dataset

The data used for evaluating models is provided by the Government and Municipalities of Quebec. (Government of Canada, 2020) The dataset includes just under 10,000 images taken from traffic cameras. Each image has a corresponding annotation file (in XML format) detailing the bounds and class of relevant objects in the frame. Labeled classes include: bus, construction, cyclist, pedestrian, and vehicle. For this exercise, all classes except construction are considered relevant. Image sizes vary throughout the dataset, but all are less than 720p resolution.

YOLOv5 Execution Results

Graphic renderings of YOLOv5 predictions depict the following components:

- Green bounding boxes: The correct/true object annotations for vehicles and pedestrians, as are included in the dataset.
- Yellow bounding boxes: Vehicle and pedestrian bounding box predictions made by the YOLOv5 model.
- Blue bounding boxes: Other object predictions made by the YOLOv5 model that are not relevant to this use-case. These are most commonly traffic signals.

- Red floating-point numbers: These are customized loss scores determined by calculating the amount of bounding box overlap and normalized to remove bias toward nearby vehicles. A loss value of 1.0 indicates a full miss, while a loss value of 0 indicates a perfect match.

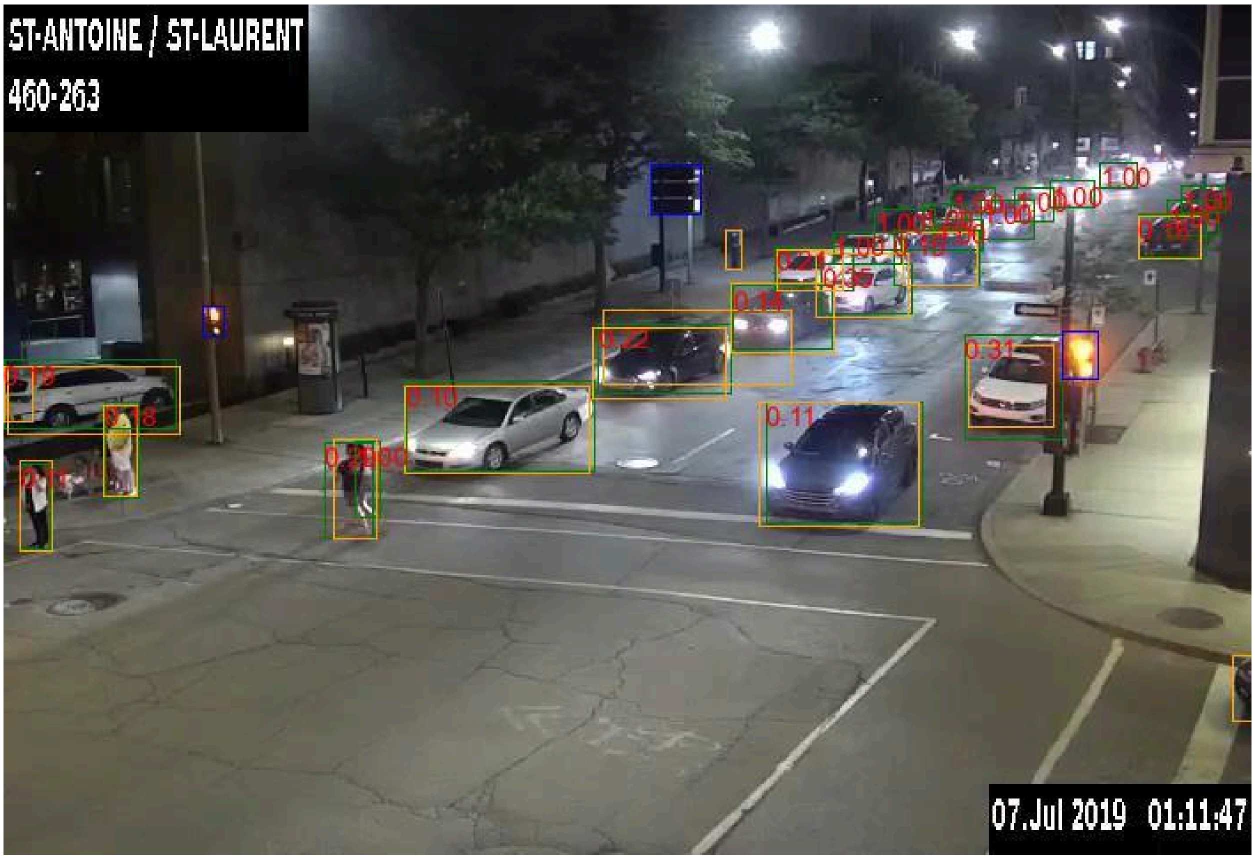

Figure 2. YOLOv5 (small) prediction sample frame

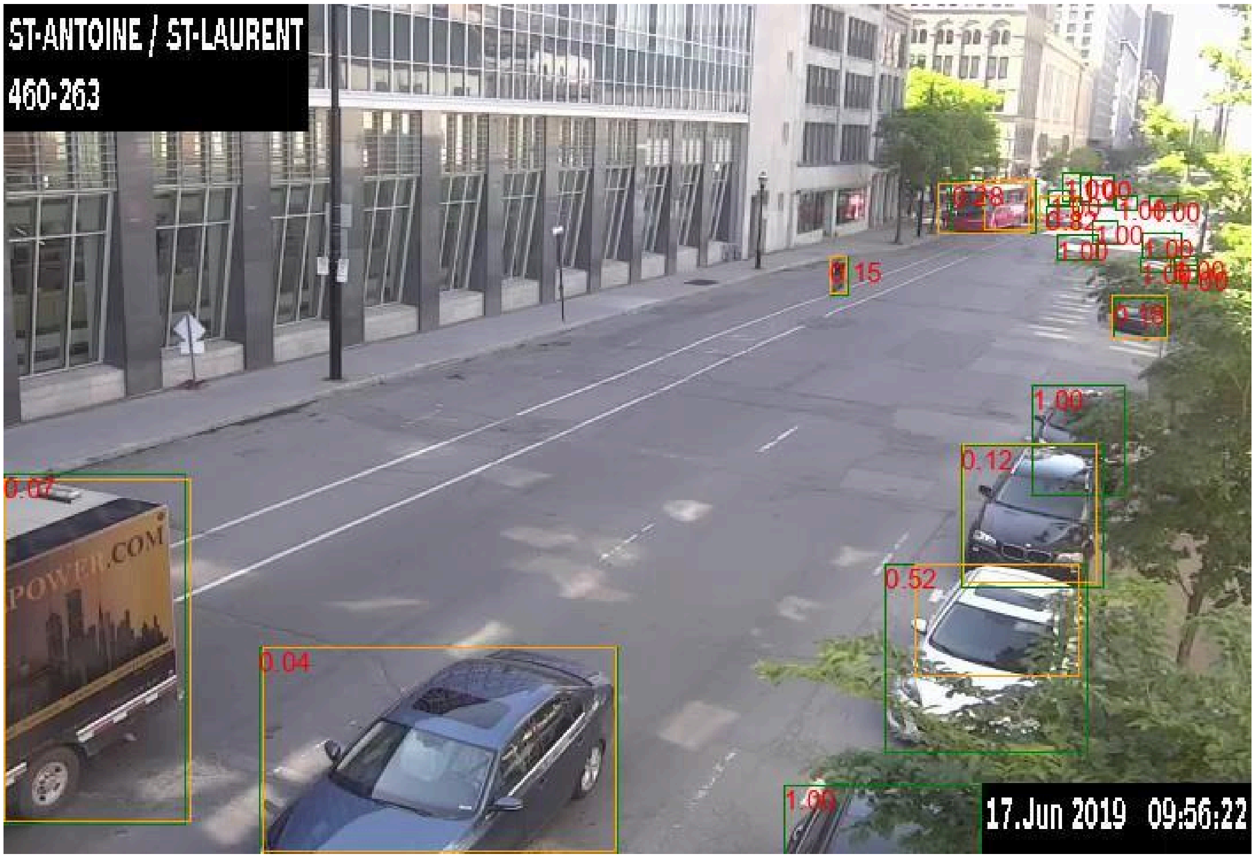

Figure 3. YOLOv5 (small) prediction sample frame

Figure 4. YOLOv5 (small) prediction sample frame

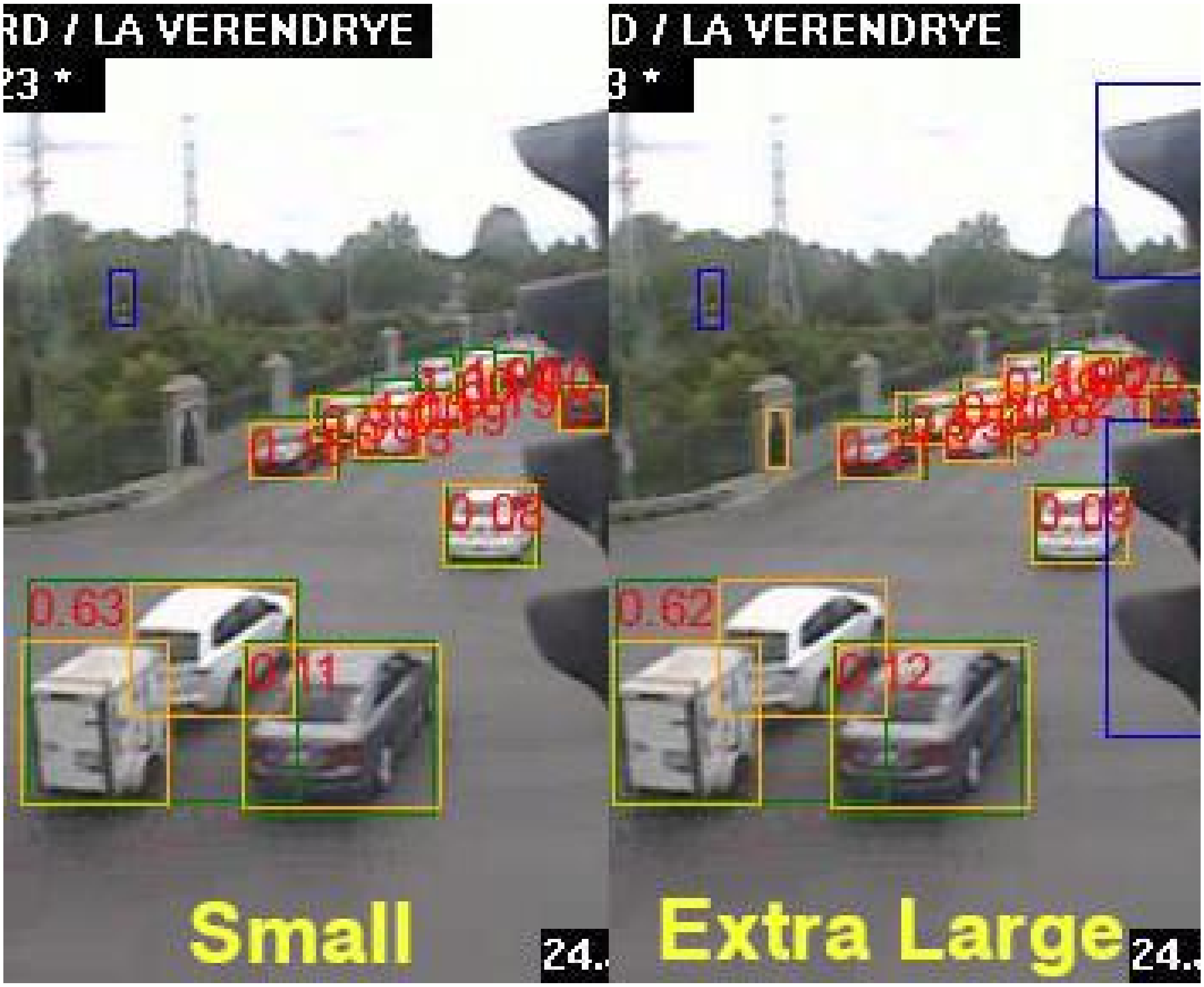

Figure 5. YOLOv5 (small vs extra large) prediction differences

The YOLO model does produce impressive results considering its processing speed. The v5 architecture provides designs for multiple size variants of the network, offering a trade-off between processing speed and detection accuracy. Each variant has been benchmarked over the traffic signal dataset to quantify the trade-off for this specific use case.

Hardware Used in Benchmark:

Component	**Model**
CPU	AMD Ryzen 7 2700X, 3700 Mhz, 8 Core(s), 16 Logical Processor(s)
GPU	Nvidia Geforce RTX 2070

Model Performance Benchmark:

Model Variant	Layers	Params	GFLOPs	Weight Size (MB)	Sec/Frame (CPU)	Sec/Frame (GPU)
Small	213	7225885	16.4	14.1	0.1184	0.0166
Medium	290	21172173	48.9	40.8	0.2569	0.0195
Large	367	46533693	109.0	89.2	0.4850	0.0278
Extra Large	444	86705005	205.5	166	0.9067	0.0399

As is common, the model scales better on GPU, while approaching a full order of magnitude difference when running on CPU. All sizes are able to process in less than one second on either set of hardware.

Constraints on model size selection will come from the hardware selected for computation in the field and the number of cameras that need to be supported. All models should fit on a modern edge AI chip like Nvidia's Jetson Modules. The Jetson Nano model (the lightest of the series) should support small deployments with up to 4 camera inputs. Its 4GB of memory will hold model weights and necessary frame samples. The 470 GFLOPs should support a medium-large YOLO model. (Nvidia, 2024)

Small Model Accuracy Metrics:

	Object Misses	Total Objects	Ratio Found	Average Loss
Pedestrian	8931	15728	0.4322	0.6910
Vehicle	42465	91748	0.5371	0.6005
Bus	494	1445	0.6581	0.4760
Cyclist	501	1184	0.5769	0.6311
Total	52391	110105	0.5242	0.6121

Extra Large Model Accuracy Metrics:

	Object Misses	Total Objects	Ratio Found	Average Loss
Pedestrian	6797	15728	0.5678	0.5722
Vehicle	32978	91748	0.6406	0.4950
Bus	325	1445	0.7751	0.3537
Cyclist	343	1184	0.7103	0.5488
Total	40443	110105	0.6327	0.5048

It should be noted that YOLO is expected to perform poorly on small, clustered objects due to its simplified object proposal system. (Redmon et al., 2016, 4) This is acknowledged and accepted in this design decision because it should not make a significant difference in the system's ability to model the intersection. Distant cars will eventually travel closer into frame and be detected. The approach detail is not mandatory information in most cases. Pedestrians, while often clustered, do not need to be individually detected. As long as one person is detected, the group is effectively represented.

Figure 5 depicts the same annotated image with the small model results on the left and extra large on the right. The larger model shows improvement in isolating and identifying tightly clustered objects. Much of the aggregate improvement in detections fall into this category.

Conclusion

These results indicate that it is feasible to achieve acceptable detection accuracy with a model light-weight enough to perform image processing on site, which achieves both the privacy and scalability objectives. Further study can be performed to tune the model for traffic detection if more traffic signal images can be procured and labeled. Initial labeling can be performed by slower but more accurate object detection routines and validated/tuned by humans as needed.

Further research is needed in processing this data into a set of models that can be used for traffic signal optimization. Contextual and supplementary data, such as Google Maps traffic data, can be used to fill in the inaccuracies and imperfections to produce more realistic reconstructions of reality. There have also been continued improvements in non-optical vehicle detection mechanisms capable of detecting the class of passing vehicles. (Mills et al., 2007) Combined with the detection capabilities demonstrated in this paper, a more robust model can be created.

Further Research Opportunities

- Training of a YOLOv5 model using a much larger dataset, tailored for traffic signal object detection to achieve improved detection accuracy.
- Evaluation of other object detection algorithms/models.
- Execution testing on edge compute infrastructure such as NVIDIA's JetsonAI chips and Google's TPUs.
- Traffic signal optimization techniques (given a model of traffic throughput)
  - Reinforcement Learning: RL shows promise where the traffic signals are the agents and its actions are signal transitions, where SUMO is used to simulate vehicle responses. (Wang et al., 2022, 9)
  - Genetic algorithms: GA has been used with some past success to produce optimal traffic signal policies. (Foy et al., 1993, 115)
- Scenario generation from the raw traffic signals collection through video feeds. This should include interpolation of vehicle behavior with tolerance for inaccuracies in the raw input feed.
- Incorporation of supplementary data such as Google Maps road maps and traffic data, available for a fee, (*Google Maps Traffic API*, 2024) into the scenario rendering process.

References

Buch, N., Velastin, S., & Orwell, J. (2011, 3 17). A Review of Computer Vision Techniques for the Analysis of Urban Traffic. *IEEE*, 920 - 939. IEEE. 10.1109/TITS.2011.2119372

Dutta, U., McAvoy, D., Lynch, J., & Vandeputte, L. (2013, 03 05). *A SCATS adaptive signal control system costs approximately $28,800 per mile per year. | ITS Deployment*

Evaluation. | ITS Deployment Evaluation. Retrieved November 22, 2024, from https://www.itskrs.its.dot.gov/2013-sc00277

Foy, M. D., Benekohal, R. F., & Goldberg, D. E. (1993, 3 5). Signal Timing Determination Using Genetic Algorithms. *TRANSPORTATION RESEARCH RECORD*, (1365), 108-115. https://onlinepubs.trb.org/Onlinepubs/trr/1992/1365/1365.pdf#page=112

Gill, S. S., Golec, M., Hu, J., Xu, M., Du, J., Wu, H., Walia, G. K., Murugesan, S. S., Ali, B., Kumar, M., Ye, K., Verma, P., Kumar, S., Cuadrado, F., & Uhlig, S. (2024, 10 20). Edge AI: A Taxonomy, Systematic Review and Future Directions. *Springer Cluster Computing*, *28*(18), 11953 - 11981. arxiv. 10.1007/s10586-024-04686-y

Google Maps Traffic API. (2024). Google Maps Platform. https://developers.google.com/maps/documentation/distance-matrix/distance-matrix#distance-matrix-advanced

Government of Canada. (2020, 10 26). *Annotated images taken from the video feed from traffic cameras (archives)*. https://open.canada.ca/data/en/dataset/3c30b818-3cd9-4877-8273-600a2ee80b05

Koonce, P., & Kittelson & Associates. (2008, 6 1). *Traffic signal timing manual*. Repository and Open Science Access Protal. https://rosap.ntl.bts.gov/view/dot/800

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Leitner, D., Meleby, P., & Miao, L. (2022, 8). Recent advances in traffic signal performance evaluation. *Journal of Traffic and Transportation Engineering (English Edition)*, *9*(4), 507-531. ScienceDirect. 10.1016/j.jtte.2022.06.002

Mills, M. K., Gibson, D., & Klein, L. A. (2007, 12). *A New Look at Sensors*. Federal Highway Administration. https://highways.dot.gov/public-roads/novdec-2007/new-look-sensors

Nvidia. (2024). *Jetson Modules*. Nvidia. https://developer.nvidia.com/embedded/jetson-modules

Pandit, B. (2024, Aug 6). *What Is Edge AI? How It Works, Applications, Challenges*. DataCamp. Retrieved 11 22, 2024, from https://www.datacamp.com/blog/edge-ai

Patel, D., & Upadhya, S. (2013, 1). Optical Flow Measurement using Lucas kanade Method. *International Journal of Computer Applications*, *61*(10), 0975 – 8887. https://research.ijcaonline.org/volume61/number10/pxc3884611.pdf

Pérez, L., Rodríguez, I., Rodriguez, N., Usamentiaga, R., & Garcia, D. F. (2016, 3 5). Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. *Sensors*, *16*, 335. 10.3390/s16030335

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016, 5 9). *You Only Look Once: Unified, Real-Time Object Detection*. arxiv. arXiv:1506.02640

Ren, S., He, K., Girshick, R., & Sun, J. (2016, 1 6). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* (3). arxiv. arXiv:1506.01497v3

scats. (n.d.). scats. https://www.scats.nsw.gov.au/home

SCATS study shows significant savings. (2013, 12 16). ITS International. Retrieved November 23, 2024, from https://www.itsinternational.com/feature/scats-study-shows-significant-savings

Slavin, C., Feng, W., Figliozzi, M. A., & Koonce, P. (2013, 12). A Statistical Study of the Impacts of SCATS Adaptive Traffic Signal Control on Traffic and Transit Performance. *Transportation Research Record Journal of the Transportation Research*, (2356), 117-126. ResearchGate. 10.3141/2356-14

Son, S., & Jin, J. (2024, 2 22). *Applying Reinforcement Learning to Optimize Traffic Light Cycles* (1). arXiv:2402.14886v1

Spall, J. C., & Chin, D. C. (1997). Traffic-responsive signal timing for system-wide traffic control. *Transportation Research Part C: Emerging Technologies*, *5*(3-4), 153-163. ScienceDirect. 10.1016/S0968-090X(97)00012-0

ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. (2022, 11 22). ultralytics. 10.5281/zenodo.3908559

Understanding cities with machine eyes: A review of deep computer vision in urban analytics. (2020). *Cities*, *96*(102481). UCL Discovery. 10.1016/j.cities.2019.102481

Vanhoey, K., Oliveira, C., Riemenschneider, H., & B ´ odis-Szomoru, A. (2017, 6 30). VarCity – the Video: the Struggles and Triumphs of Leveraging Fundamental Research Results in a Graphics Video Production. *SIGGRAPH '17 Talks*. https://varcity.ethz.ch/paper/siggraph2017_vanhoey_varcity.pdf

Wang, B., He, Z., Sheng, J., & Chen, Y. (2022, 11 20). Deep Reinforcement Learning for Traffic Light Timing Optimization. *Processes*, *10*(11), 2458. mdpi. https://doi.org/10.3390/pr10112458

Wen, D., Zhou, Y., Li, X., Shi, Y., Huang, K., & Letaief, K. B. (2024, 11 12). *A Survey on Integrated Sensing, Communication, and Computation* (2). arxiv. https://arxiv.org/abs/2408.08074v2

Wen, L., Fu, D., Mao, S., Cai, P., Li, Y., & Qiao, Y. (2023, 7 13). LimSim: A Long-term Interactive Multi-scenario Traffic Simulator. *IEEE International Conference on Intelligent Transportation Systems*. arXiv. arXiv:2307.06648

Yaqoob, I., Khan, L. U., Kazmi, S. M. A., & Imran, M. (2019). Autonomous Driving Cars in Smart Cities: Recent Advances, Requirements, and Challenges. *IEEE*, *34*(1), 174-181. IEEE. 10.1109/MNET.2019.1900120

Zaidi, S., Ansari, M., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2021, 5 12). A Survey of Modern Deep Learning based Object Detection Models. *IET Computer Vision*. arXiv. arXiv:2104.11892

Zhang, Y., Guo, Z., Wu, J., Tian, Y., & Tang, H. (2022, 9 27). Real-Time Vehicle Detection Based on Improved YOLO v5. *Sustainability*, (2022). MDPI. 10.3390/su141912274