Traffic Queue Monitoring with Mask region-based Convolutional Neural Network

Vishal Mandal, Lan P Uong, Peng Jin, Yaw Adu-Gyamfi

Traffic Queue Monitoring with Mask region-based Convolutional Neural Network

Vishal Mandal, Corresponding Author
Department of Civil and Environmental Engineering
University of Missouri
E 1511 Lafferre Hall, Columbia MO 65211
Tel: 573-639-3342; Email: vmghv@mail.missouri.edu

Lan P. Uong
Department of Civil and Environmental Engineering
University of Missouri
E 1511 Lafferre Hall, Columbia MO 65211
Tel: 573-777-2485; Email: lpu5xc@mail.missouri.edu

Peng Jin
Department of Civil and Environmental Engineering
University of Missouri
E 1511 Lafferre Hall, Columbia MO 65211
Tel: 573-825-3128; Email: peng.jin@mail.missouri.edu

Yaw Okyere Adu-Gyamfi
Department of Civil and Environmental Engineering
University of Missouri
C 2647 Lafferre Hall, Columbia MO 65211
Tel: 573-882-7546; Email: adugyamfiy@missouri.edu

Word count: 4,758 words text + 2 tables x 250 words (each) = 5,258 words

Submission Date: 29 July 2018

Abstract

This study implements a video-based traffic queue monitoring system using Mask RCNN: A convolutional neural network (CNN) approach for predicting pixel-level segmentation masks on classified regions of interest. Taking advantage of a large database of annotated video surveillance data and recent advances in machine learning and high-performance computing, we train a deep-learning based model that is able to accurately extract traffic queue-related information from infrastructure mounted video cameras. Several experiments are conducted to fine-tune the system's robustness in different traffic and environmental conditions. Overall, the system achieves 92.8% accuracy in daylight, night, and rainy conditions. Although extremely poor but rare conditions affects the system's accuracy, it is able to learn and correct for false detections when re-trained with data captured under such conditions. A comparative analysis with YOLO (You Look Only Once), a classical single stage CNN method is also conducted. Although Mask RCNN underperformed YOLO by approximately 3% error margin in all categories, its ability to provide pixel level segmentation makes it superior for extracting traffic queue parameters. The outcome of this study could be seamlessly integrated into traffic system such as smart work zone management systems, signal control systems, etc.

Keywords: Convolutional Neural Network, Deep Learning, Traffic Monitoring, Pixel-wise Segmentation.

INTRODUCTION

As the world's population continues to rise, roadways are utilized far and beyond the limits for which they are designed for. This destabilizes traffic on major highways, resulting in the occurrence of relatively unpredictable vehicle queues which produces dangerous driving conditions. With the rise in urbanization and rapid industrialization, the need to have an unclogged and safe traffic system is of utmost importance. It is therefore not surprising that traffic congestion mitigation has been one of the major mandates of the U.S. Department of Transportation (USDOT). Ullman et al. (*1*) postulates that a robust congestion management system which is able to detect and quickly alert response teams about the onset of congestion events has the potential to save lives, reduce queue lengths and wait times and overall improve mobility through congestion bottlenecks. Intelligent queue monitoring systems are therefore an integral component of systems needed to palliate the effects of traffic congestion. The goal of this paper is to develop a real-time queue monitoring system that leverages recent advances in machine learning and computer vision to detect, track and report traffic queue characteristics relevant for congestion management.

Popular queue detection systems receive real-time traffic data from loop detectors or microwave radar sensors. These streams of data are subsequently passed through algorithms that detects queue formation based on speed-volume-occupancy relationships and continuously tracks them as they burgeon and dissipate. Skabardonis et al. (*2*), Bezuidenhout et al. (*3*) and Hourdos' (*4*) work for instance, proposed methods which estimate queue length using aggregated loop detector data in 30-second intervals. With the numerous shortcomings of detector data, probe-based queue detection alternatives are recently being considered. Dinh et al. (*5*), Wang et al. (*6*), Adu-Gyamfi et al. (*7*) and Cheng et al. (*8*) developed end-of-queue and congestion platoon detection systems for roadways by feeding high resolution vehicle probe data through shockwave theory-based algorithms. They report very high detection accuracies on highways with good probe penetration rates.

The current generation of queue detection systems require a very dense deployment of vehicle detectors or sensors which have proven to be very expensive. Also, their effectiveness can be diminished if the prevailing traffic conditions are not accurately captured by the sensors deployed within the area of analysis (*9*). This may lead to false alarms and missed detection which tend to confuse drivers and consequently, reduce a system's credibility. The queue monitoring system developed in this paper is based on live video feeds streaming from infrastructure mounted CCTV cameras. It is inspired by He et al.'s (*10*) recent work on Mask RCNN which used convolutional neural networks to detect objects while generating high quality, pixel level segmentation masks for each instance. We use the backbone of this development to extract low - high level features from traffic surveillance video databases. After a series of mapping and sampling, the features are passed through classifiers which predicts queuing regions in a traffic scene. Finally, a pixel segmentation branch takes the classified regions and generates masks for each of them. Queuing parameters are extracted from the mask's shape as queues form and dissipate over time. The developed system can be used as standalone or preferably integrated into already existing QDS to improve its robustness.

The remainder of this paper is organized as follows: First we discuss closely related research work that has been done in the area of video-based queue monitoring systems. This is followed by a summary of the key contributions of the current study. Next, the key points of the

methodology adopted is described. This section will also include a description of models developed and the data used to evaluate the effectiveness of the queue detection system. Afterwards, results of the study will be analyzed, highlighting key advantages, bottlenecks and challenges. The last section will include a conclusion of the study and make final recommendations of how to maximize the use of video-based queue detection systems.

RELATED WORK

There are two main groups of video-based traffic queue monitoring systems: three-step-inference-based and one-step-classification-based approaches. Inference-based approaches first learn to detect and recognize cars on the road, then it tracks each detected vehicle to estimate the average speed and finally classifies the traffic-scene as either congested or uncongested, based on a predefined speed threshold. Classification-based approaches on the other hand are trained to recognize congested scenes directly without the intermediate steps of vehicle detection and tracking. Several variations of both techniques have been studied in literature. Willis et al. (*11*) analyzed traffic congestion classification using deep neural network on traffic imagery by training a two-phase network using GoogLeNet and bespoke deep subnet for both image processing and congestion detection. In their study, a deep-learned classifier was able to detect traffic congestion with an accuracy of about 95%. Chakraborty et al. (*12*) used camera images and upon applying DCNN and YOLO algorithms in different environmental conditions, concluded with YOLO model achieving the highest accuracy of 91.2% followed by DCNN with 90.2%. Overall, one-step classification-based approaches tend to excel at interpretation of the congestion state of traffic scenes and are robust to different camera configurations or environmental conditions. The main limitation of this approach is its inability to accurately predict and track the location, extent and severity of traffic queues.

Regarding inference-based approaches, several methods have been presented that estimate queue lengths by extracting speed, occupancy and other features from videos. Morris et al. (*9*) developed a portable system for extracting traffic queue parameters at signalized intersections from video. The authors used straightforward image processing tasks like background subtraction, clustering and segmentation to isolate vehicles and estimated queue lengths for different calibrated cameras at different intersections. A similar research methodology was applied by Hao et al. (*13*) to study traffic queue detection capability from videos. They used maximum similarity matching standard method and focused on a fixed background area window to rectify the deviations in real-time images and the lane region in setting. This was followed by region partitioning approach, for mapping the distances from video-image to the actual distance. In order to map between video-distances and actual distances, AOI regionalism approach was adopted wherein the roads were divided into equal lane segments. Although the authors report significantly high accuracy rates (> 85%), inference-based approaches are impractical for network level evaluation since traffic speed estimation from videos requires calibration for each camera.

Advanced vehicle detection studies (*14*) which focuses on machine learning frameworks for vehicle detection, and multi-object tracking algorithms developed by Bewley et al. (*15*) could be used to improve significant portions of Morris et al. (*9*) and Hao et al.'s (*13*) work. Recent studies on Convolutional Neural Networks (CNN) for vehicle detection and classification has yielded superior performance over other algorithms. Bautista et al. in (*16*) employed CNN to investigate its performance in accurately detecting and classifying vehicles using low quality traffic cameras. Most automatic queue monitoring system builds on the fact of how well vehicles are detected. An unsupervised learning algorithm used to classify congested scenes using feature

learning and density estimation is proposed in (*17*). Similarly, a real-time computer vision system based on feature learning to track vehicles and monitor traffic is proposed in (*18*). In order to measure traffic parameters, a real-time computer vision system is introduced in (*19*), that is capable of tracking vehicles under congested conditions using a feature-based tracking approach.

RESEARCH CONTRIBUTION

In this study, 1,509 traffic images under varying conditions from Iowa, Virginia and New York were obtained to train deep learning models for traffic queue detection and monitoring. We aim to open-source all the image resources we used and the corresponding annotated training data sets so involved during our study (*20*). As far as our knowledge is concerned, we believe this is the first time a Mask region based convolutional neural network has been used to detect traffic queues in real-time. Likewise, our model is also capable of extracting queue related parameters from traffic videos. The most significant aspect of using a Mask R-CNN model for predicting traffic queues is that there is a pixel-wise segmentation of congested regions which makes the detections more precise. Chakraborty et al. (*12*) uses a bounding box approach to detect congestion where in the size of box covers exceedingly larger areas and the overall detection method may not seem as precise as the one performed with Mask-RCNN.

PRACTICAL APPROACH

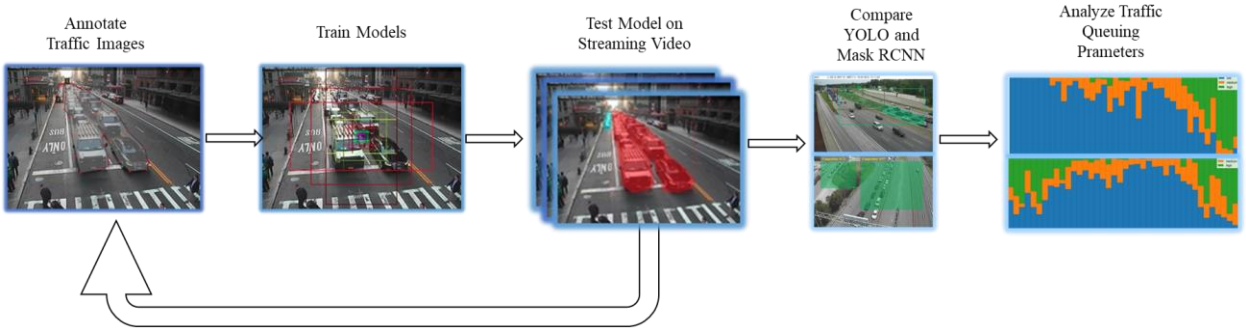


FIGURE 1 Flowchart of Mask-RCNN step-wise operations

The methodology adopted for implementing Mask-RCNN based traffic queue monitoring is shown in Figure 1 above. We first annotated queuing scenes from hundreds of traffic surveillance images. The annotated images are then used to train both Mask-RCNN and YOLO models. To effectively handle memory and speed requirement, NVIDIA GTX 1080Ti GPU was used. The training time for Mask-RCNN and YOLO were 3 and 22 hours respectively. The trained models were tested on live traffic video-feeds to evaluate their performance. If a particular congested scene is missed by the model, images are sampled from the scene for annotation which is subsequently used for building a new model. After the model's accuracy reached appreciable levels, a comparative analysis between Mask RCNN and YOLO was performed. Finally, Mask RCNN model was used to extract traffic queuing parameters from work-zones, freeways and intersections using RITIS data. The algorithms used in our study are described in detail as follows:

Mask-RCNN

Mask R-CNN, is an extension to Faster R-CNN. In addition to performing tasks analogous to Faster RCNN, Mask R-CNN augments it by adding high-quality masks and segments the region of interest pixel-by-pixel. Our model is based on Feature Pyramid Network (FPN) and is implemented with resnet101 backbone. Here, ResNet101 serves as our feature extractor. It is worth mentioning that the early layers detected lower level features such as edges and corners and the later layers could effectively detect higher-level features such as vehicles, traffic queues, etc. from the images. Likewise, with FPN, we observed that it improved the standard feature extraction pyramid by introducing a second pyramid that took higher level features from the first pyramid and consequently passed them down to lower layers. That, actually allowed features at every level to have access to both high and low-level features. We set the minimum detection confidence at 90% and ran it at 50 validation steps. Our model was run at 30th epoch with each epoch having 100 iterative steps. We followed an image centric training wherein the images are resized to the shape of a square.

On passing through the backbone network, images were converted from 1024×1024px× 3 (RGB) to a feature map of shape 32×32×2048. Each of our batch had 1 image per GPU and each image had 200 trained Region of Interests (ROIs). The model was trained on NVIDIA GTX 1080Ti GPU with a batch size equals to 1 and a learning rate of 0.001. We used the constant learning rate throughout the iteration. Similarly, we used a weight decay of 0.0001 and a learning momentum of 0.9. For training the model using a sample dataset of 1,509 images, took us nearly 3 hours. The total training time was relatively shorter because the number of images were moderate in number. Larger the number of images in the sample dataset, better the results for masked detections. However, the problem with instance-level segmentation is that if the number of training images are way too many, there might not be as expected improvements in the nature of detections. To effectively remedy this issue, it is advisable to have the right number of images for the training sample.

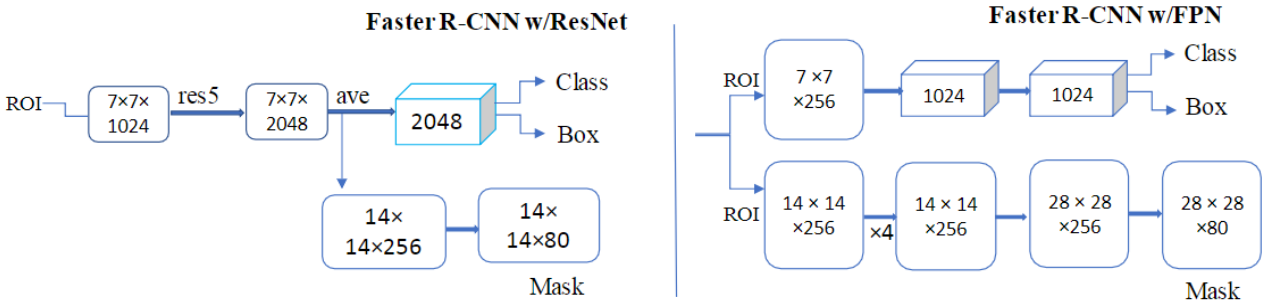


FIGURE 2 Mask-RCNN Framework

YOLO

You look only once (YOLO) is the state of the art object detection algorithm. It is a real-time object detection system which unlike traditional classifier systems looks into the image only once and can detect the objects in it. In our study, we used YOLO to compare accuracy of test results for queue detection with Mask-RCNN. Current object detection algorithms repurpose CNN classifiers in order to conduct detections. For example, to perform any object detection, these algorithms use a classifier for that object and test it at varied locations and scales in the test image. The good thing about YOLO is that it reframes object detection that is, instead of looking

at a single image 1,000 times to perform detection, it just looks at the image once and performs accurate object predictions. A singe CNN concurrently predicts multiple bounding boxes and class probabilities for those generated boxes. It is because of this feature that makes YOLO extremely fast and easy to implement to different scenes. The CNN architecture used by YOLO is presented in Table 1. The model uses standard layer types: convolutional with a 3 × 3 kernel and max pooling with a 2 × 2 kernel. The last convolutional layer has a 1 × 1 kernel, which helps minimize data to the shape 13 × 13 × 125. This 13 × 13 structure is the size of grid where the image gets apportioned. For every grid cell, we have 35 channels that represent data for the bounding boxes as well as class predictions. Each of these grid cells predict 5 bounding boxes and those boxes are described by seven data elements: the values of x, y, width, and height for the bounding box's rectangle; the confidence score; congested and non-congested probability distribution.

TABLE 1 YOLO Model Architecture Used

Layer	Kernel	Stride	Output Shape
Input			[416, 416, 3]
Convolution	3×3	1	[416, 416, 16]
Max Pooling	2×2	2	[208, 208, 16]
Convolution	3×3	1	[208, 208, 32]
Max Pooling	2×2	2	[104, 104, 32]
Convolution	3×3	1	[104, 104, 64]
Max Pooling	2×2	2	[52, 52, 64]
Convolution	3×3	1	[52, 52, 128]
Max Pooling	2×2	2	[26, 26, 128]
Convolution	3×3	1	[26, 26, 256]
Max Pooling	2×2	2	[13, 13, 256]
Convolution	3×3	1	[13, 13, 512]
Max Pooling	2×2	1	[13, 13, 512]
Convolution	3×3	1	[13, 13, 1024]
Convolution	3×3	1	[13, 13, 1024]
Convolution	1×1	1	[13, 13, 35]

The implementation steps for YOLO are discussed as follows:

(i) The input image is resized to 416 × 416 pixels.
(ii) The image is passed through a CNN in a single pass.
(iii) The output of a CNN is a 13 × 13 × k tensor that describes bounding boxes for the grid cells. The value of k is related to the number of classes. For example: k = (number of classes + 5) * 5.
(iv) The confidence scores for all the bounding boxes is computed and the boxes that fall below a certain predefined threshold is rejected.

For our model, we have 13 × 13 = 169 grid cells and each cell predicts 5 bounding boxes. It is important to mention that we have altogether 845 bounding boxes. In ideal terms, majority of these boxes have very low confidence scores and therefore, to have a better congestion detection capability we used a confidence threshold of 45%.

Data Description

Traffic images of Iowa, New York and Virginia were obtained from Iowa 511, New York State Department of Transportation and RITIS respectively. Upon visual inspection of traffic images, the ones with highly congested regions were stored into a database whereas the rest were discarded. The total image count with visible traffic congestion was 1,509. The acquired data was sub-divided into 1,184 training and 325 validation image sets. The datasets consisted of images taken at different times of the day in different environmental conditions and contained congestion of all sort, from multiple regions of heavily congested areas to the regions low on traffic. In order to test accuracy of Mask-RCNN and YOLO models, a set of 1,000 traffic surveillance images (500 congested and 500 uncongested) was used. Finally, for studying traffic queue related parameters, video feeds from congestion at work-zone, freeway and intersection was used.

Some of the traffic images obtained from Iowa 511, RITIS and New York State DOT under different environmental conditions and camera orientations are shown in Figure 3.

FIGURE 3 Traffic Queue Images: 1st Row - Intersections during day, 2nd Row - Freeways at night, 3rd row - Freeways during snow, 4th Row- Work Zones

RESULTS

In this section we first evaluate the performance of Mask RCNN on a set of 1000 traffic surveillance images (500 congested and 500 uncongested) and compare its performance with the classical YOLO framework. Next, the results of a real-time implementation of Mask RCNN for queue monitoring at work zones, freeways and intersections is discussed.

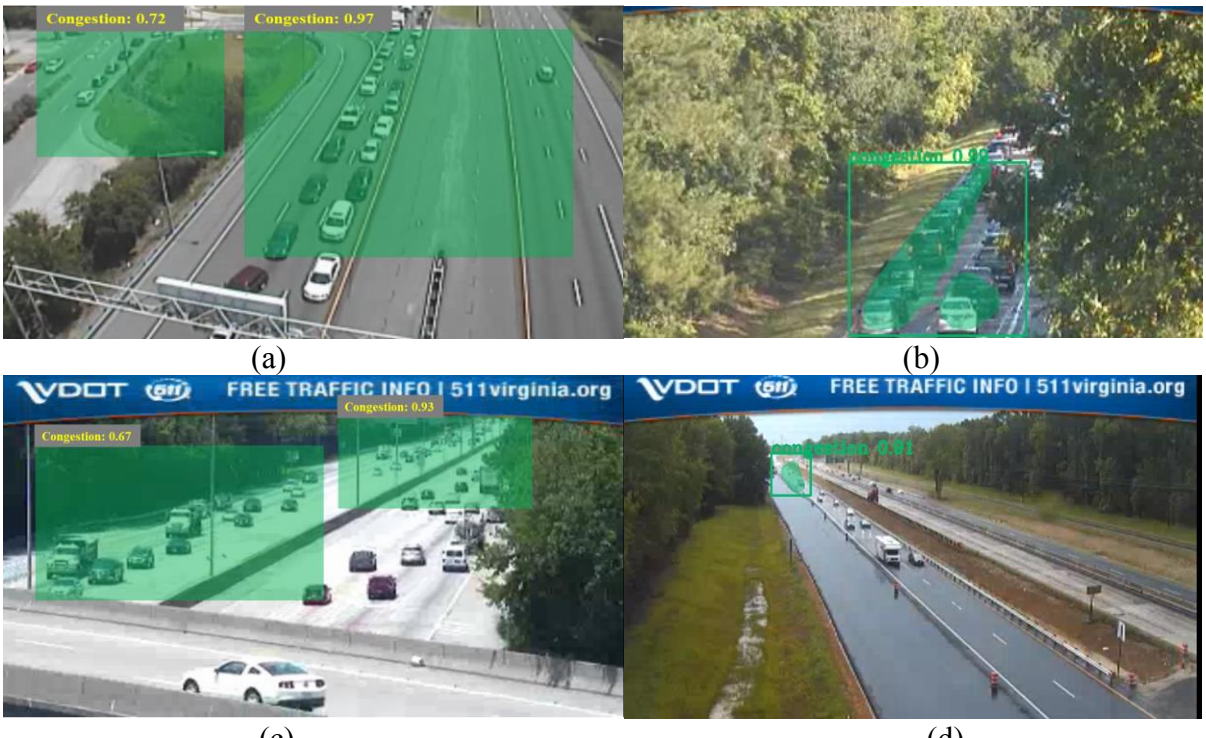
Standard performance metrics of precision, recall and accuracy, shown in equations (i), (ii) and (iii) respectively were used.

$$Precision = \frac{TP}{TP + FP} \quad \text{(i)}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{(ii)}$$

$$Accuracy = \frac{TP}{TP + FP + TN + FN} \quad \text{(iii)}$$

TP, FN, FP, TN are abbreviated as True Positive, False Negative, False Positive and True Negative respectively. While testing, if the congested image is correctly labeled such that the predicted label is also 'congested', then that particular image is classified as true positive (TP). Likewise, if any uncongested image is correctly labelled as 'uncongested', then it is classified as true negative (TN). In cases, where the actual label is 'congested' but the predicted label is 'uncongested', the image is classified as false negative (FN). Similarly, if the actual label is 'uncongested' and the predicted label is congested, then the classifications are made in the false positive category (FP). Figure 4 shows some of the true classifications and misclassifications obtained from Mask-RCNN and YOLO models:



(a) (b)

(c) (d)

(e) (f)

(g) (h)

FIGURE 4 Classification of predicted queues examples: True Positive-(a, b), False Positive-(c, d), False Negative-(e, f), True Negative-(g, h) obtained from YOLO and Mask R-CNN respectively

Figure 4*a* and 4*b* were accurately predicted as congested and classified as true positives. Figure 4*a* and 4*b* were the detections made by YOLO and Mask-RCNN respectively. YOLO predicts congestion using bounding box whereas Mask-RCNN creates a color-masked region around the congested area. Likewise, Figure 4*c-d* shows misclassification of non-congested images as congested and are classified as false positives. YOLO incorrectly predicted an uncongested image to a congested one due to the presence of an overhead bridge which is uncongested (Figure 4*c*). On the other hand, Mask-RCNN couldn't correctly interpret the image as the group of vehicles appeared far away from the camera (Figure 4*d*). Example of false negatives are shown in Figure 4*e-f*, where YOLO and Mask-RCNN failed to detect congestion. Traffic-queues quite distant from camera image was responsible for misclassification by YOLO (Figure 4*e*). Glaring effect as well as distant queues resulted in Mask-RCNN's incorrect classification (Figure 4*f*). Finally, Figure 4*g-h* were correctly classified as true negatives as per the initial uncongested labeling.

The precision, recall and accuracy values obtained from both models are shown in Table 2. YOLO achieved the highest precision, accuracy and a lower recall value compared to Mask-RCNN. From Table 2, it is evident that the overall performance of Mask-RCNN is quite comparable to that of YOLO. Since, Mask-RCNN supports pixel-wise segmentation compared to a bounding box approach followed by YOLO, queues detection is much more precise. Therefore, in context of traffic queues detection and study of queue related parameters, Mask-RCNN outperforms YOLO as it selects only the regions occupied by queues, thereby facilitating an accurate congestion measure.

TABLE 2 Precision, Recall and Accuracy Values Obtained from Mask-RCNN and YOLO

Model	Precision (%)	Recall (%)	Accuracy (%)
Mask-RCNN	92.8	95.6	90.5
YOLO	95.5	94.8	93.7

Case Study

In this section, we undertake a case study where the Mask RCNN model developed is implemented in real time for queue monitoring at an intersection, on a freeway and construction work zone.

Extracting Queue Parameters

Video camera perspective distortions make it challenging to extract queuing parameters from a traffic scene. A typical approach around this is to calibrate the camera to a specific height, viewing angle, zoom level, etc. Although this is effective, it is not scalable. A second alternative directly uses image pixel values to represent queue parameters. With this approach, queue information from one location cannot be compared to another location because camera geometric configurations may differ. In the following steps, we develop a simple, calibration free method for extracting queue length parameters from video surveillance feeds. The approach is scalable and can be used to compare queuing levels at different locations.

Step 1: Extract queue regions in video with Mask RCNN.
Step 2: Calculate the pixel length of each detected queue mask.
Step 3: Accumulate length over time (minimum duration is 1 week).
Step 4: Use adaptive thresholding (Figure 5) to bin queue lengths into different severity levels: low, medium and high.
Step 5: Generate heat map of queuing levels and compare.

FIGURE 5 Adaptive Thresholding Steps

Steps shown for Adaptive Thresholding

Initialize: L, M, H
Input: PL – pixel lengths
for each location **do**
  for each [day, hour, minute] in [30 days, 24 hours, 60 minutes] **do**
    % extract first, second, third quartile pixel lengths
    $\mathrm{Q} = \mathrm{percentile}[\mathrm{PL}, \{\mathrm{Q1}, \mathrm{Q2}, \mathrm{Q3}\}]$
  end
  $\mathrm{L} = \mathrm{Q}[\{\mathrm{Q1}, \mathrm{Q2}, \mathrm{Q3}\}].\mathrm{mean}.\max + k * \mathrm{Q}[\{\mathrm{Q1}\}].\mathrm{std}$
  $\mathrm{M} = \mathrm{Q}[\{\mathrm{Q1}, \mathrm{Q2}, \mathrm{Q3}\}].\mathrm{mean}.\max + k * \mathrm{Q}[\{\mathrm{Q2}\}].\mathrm{std}$
  $\mathrm{H} = \mathrm{Q}[\{\mathrm{Q1}, \mathrm{Q2}, \mathrm{Q3}\}].\mathrm{mean}.\max + k * \mathrm{Q}[\{\mathrm{Q3}\}].\mathrm{std}$
end
Output: L, M, H

The Mask RCNN framework was used to quantify queuing levels at a work zone, freeway and intersection locations. The heat map plots in Figure 6 through 8 are used to illustrate the results.

In general, the model is able to clearly capture the onset and dissipation of queues. The heat map for the freeway and intersection were able to detect AM and PM peak hour periods. At the work zone site, only a PM peak hour was detected. After further investigation, it was realized that work zone activities started after the AM peak, hence the low levels of queueing.

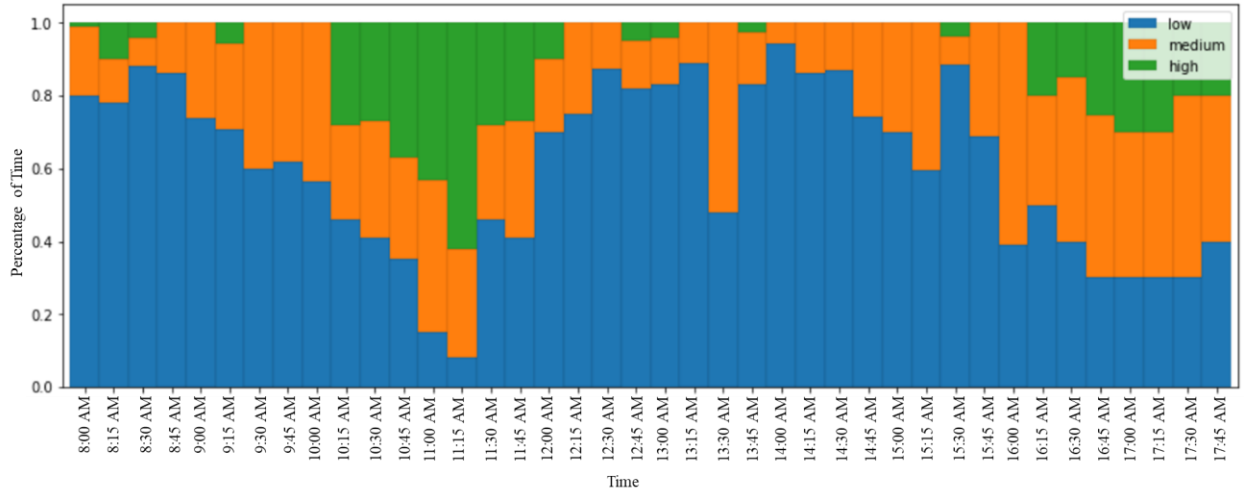


FIGURE 6 Heat map of traffic queue severity at freeway

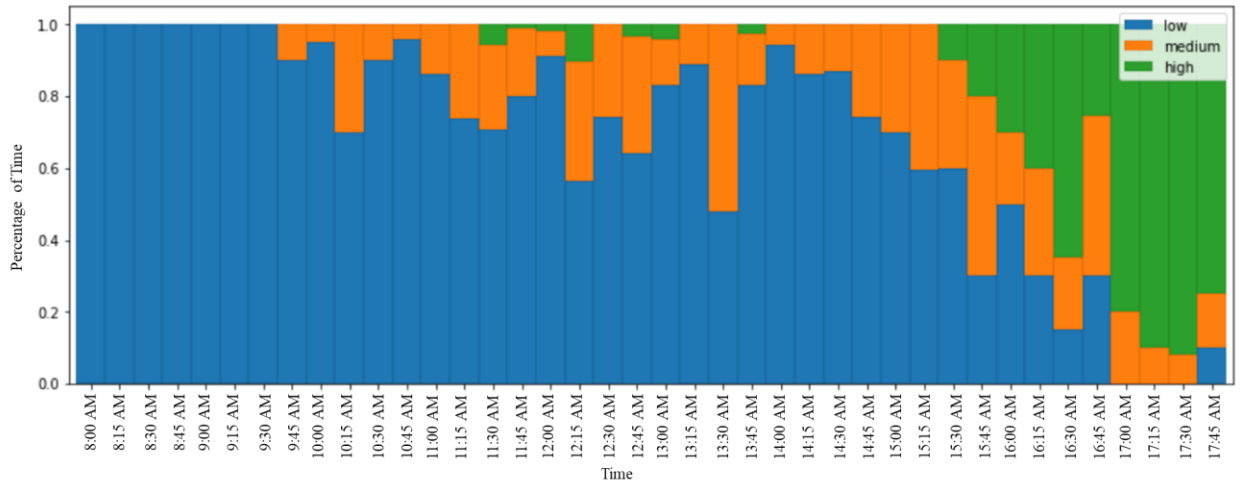


FIGURE 7 Heat map of traffic queue severity at work zone

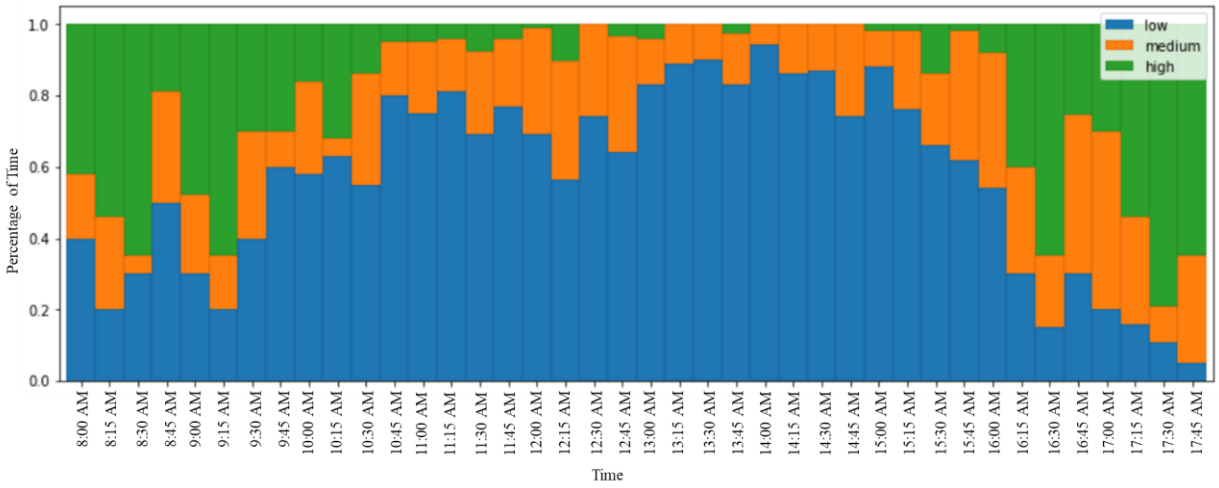


FIGURE 8 Heat map of traffic queue severity at an intersection

Bottlenecks and Challenges

Mask RCNN takes approximately 0.3 seconds to process a traffic scene. A typical frame rate for CCTV cameras is 15 frames per second (fps). At this rate, the methodology developed in this paper cannot be used in real time. One way around this is to use YOLO (which can process 50 frames in one second) to process video feeds initially, if a scene is flagged as congested, Mask RCNN model can be called to extract the queue parameters for that particular scene. This way, the model is not running on every single frame from the traffic scene. Alternatively, feeds from CCTV cameras could be re-sampled at 1 fps instead of 15fps. Another bottleneck encountered was regarding how queues are described. A queue at an intersection could just be a platoon of vehicles on a freeway. Training the Mask RCNN to be able to distinguish between queues at intersections and freeways was a challenge. Eventually, we had to create two different models: one for uninterrupted and the other for interrupted.

CONCLUSION

The rapid advancement in the field of machine learning and high-performance computing have highly augmented the scope of video-based traffic management systems. In the current study, we implemented two deep learning algorithms, Mask-RCNN and YOLO. Mask-RCNN was used to detect traffic queues from real-time video feeds whereas YOLO was used for comparison of test results. To ensure uniformity, same dataset containing 1,509 images was used to train both Mask-RCNN and YOLO. Also, in order to establish accurate comparison between the two models, sample dataset consisting of 1,000 (500 congested and 500 uncongested) images was used.

Mask-RCNN achieved an accuracy of 92.8% while the highest accuracy achieved by YOLO was 95.5%. The discrepancies in correctly detecting congestion was largely due to the poor image quality, traffic queues located far away from the camera, single-lane blockages and glaring effect. All these issues significantly affected the accuracies of the models. Performance in terms of correctly detecting congestion was found to be better during the day-time than at night. Similarly, for images with too many objects, queue detection wasn't very accurate which caused a small dip in the overall performance. However, for all conditions, the models were found to record accuracies greater than 90%. Therefore, it is quite evident that proposed models are capable of detecting queues in challenging conditions as well. In order to extract queue length parameters of video feeds from intersection, freeway and work zone, adaptive thresholding was

used to bin queue lengths into different severity levels (i.e. low, medium and high). By generating heat maps, queueing levels at different locations were analyzed. For intersection and freeway, AM and PM peak hours were detected whereas for work zone, only PM peak hour was detected. Hence, the proposed Mask-RCNN model was able to effectively monitor the onset and dissipation of queues.

Future studies in this area could look into a more robust traffic queue-detection system using a larger image dataset and could use different model architectural designs to enhance congestion detection accuracies. These systems could be further used to automatically calibrate different CCTV cameras, remain resolute to any changes in camera orientation and be able to accurately extract queue-length parameters in feet or meters.

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: study conception and design: Vishal Mandal, Lan P. Uong, Peng Jin, Yaw Okyere Adu-Gyamfi; data collection: Vishal Mandal, Yaw Okyere Adu-Gyamfi; analysis and interpretation of results: Vishal Mandal, Lan P. Uong, Peng Jin, Yaw Okyere Adu-Gyamfi; draft manuscript preparation: Vishal Mandal, Lan P. Uong, Peng Jin, Yaw Okyere Adu-Gyamfi. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. Ullman, G. L., V. Iragavarapu, and R.E. Brydia. Safety Effects of Portable End-of-Queue Warning System Deployments at Texas Work Zones. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2555, 2016, pp. 46-52. https://doi.org/10.3141/2555-06.
2. Skabardonis, A., and N. Geroliminis. Real-time Monitoring and Control on Signalized Arterials. Journal of Intelligent Transportation Systems, 2008, 12 (2), pp. 64-74. https://doi.org/10.1080/15472450802023337.
3. Bezuidenhout, J.J., P. Ranjitkar, and R. Dunn. Estimating Queue at Traffic Signals. The Open Transportation Journal, 2014, 8, pp. 73-82.
4. Hourdos, J. Development of a Queue Warning System Utilizing ATM Infrastructure System Development and Field Testing. *Minnesota Department of Transportation*, 2017, available at: http://mndot.gov/research/reports/2017/201720.pdf.
5. Dinh, T.U.J., R. Billot, E. Pillet, and N.E.E. Faouzi. Real-Time Queue-End Detection on Freeways with Floating-Car Data: Practice-Ready Algorithm. *Transportation Research Record: Journal of Transportation Research Board*, No. 2470, 2014, pp. 46-56. https://doi.org/10.3141/2470-05.
6. Wang, Z., Q. Cai, B. Wu, L. Zheng, and Y. Wang. Shockwave-Based Queue Estimation Approach for Under-saturated and Over-saturated Signalized Intersections Using Multi-Source Detection Data. *Journal of Intelligent Transportation Systems*, Vol. 21, Issue 3, 2017, pp. 167-178. https://doi.org/10.1080/15472450.2016.1254046.
7. Adu-Gyamfi, Y.O., A. Sharma. Comprehensive Data-Driven Evaluation of Wide-Area Probe Data: Opportunities and Challenges. Presented at 95th Annual Meeting of the Transportation Research Board, Washington, D.C., 2016
8. Cheng, Y., X. Qin, J. Jin, and B. Ran. An Exploratory Shockwave Approach to

Estimating Queue Length Using Probe Trajectories. *Journal of intelligent transportation systems*, Volume 16, Issue 1, 2012, pp. 12-23.

9. Morris, T., J. A. Schwach, and P. Michalopoulos. Low-Cost Portable Video-Based Queue Detection for Work-Zone Safety. *University of Minnesota, Center for Transportation Studies*, 2011, Report no. CTS 11-02.
10. He, K., G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. arXiv:1703.06870, 2018.
11. Willis, C., D. Harborne, R. Tomsett, and M. Alzantot. A Deep Convolutional Network for Traffic Congestion Classification. *In Proc NATO IST-158/RSM-010 Specialists' Meeting on Content Based Real-Time Analytics of Multi-Media Streams*, 2017, pp. 1–11.
12. Chakraborty, P., Y.O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar. Traffic Congestion Detection from Camera Images using Deep Convolution Neural Networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2018. https://doi.org/10.1177/0361198118777631.
13. Hao, C., and L. Yongyi. Research on Queue Detection Technology Based on Video for City Road Section. *ICTIS 2011: Multimodal Approach to Sustained Transportation System Development: Information, Technology, Implementation, 2011, pp. 652-661*.
14. Adu-Gyamfi, Y. O., S. K. Asare, A. Sharma, T. Titus. Automated Vehicle Recognition with Deep Convolutional Neural Networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2017, No. 2645, pp. 113–122.
15. Bewley, A., Z. Ge, L. Ott, F. Ramos, B. Upcroft. Simple Online and Realtime Tracking. *Image Processing (ICIP), 2016 IEEE International Conference on. IEEE*, 2016, pp. 3464-3468. arXiv:1602.00763.
16. Bautista, C. M., C. A. Dy, M.I. Mañalac, R.A. Orbe, and M. Cordel. Convolutional neural network for vehicle detection in low resolution traffic videos. In *Region 10 Symposium (TENSYMP), 2016 IEEE* (pp. 277-281). IEEE.
17. Yuan, Y., J. Wan, and Q. Wang. Congested scene classification via efficient unsupervised feature learning and density estimation. Pattern Recognition 56, 2016, pp. 159-169.
18. Coifman, B., D. Beymer, P. McLauchlan, and J. Malik, J. A real-time computer vision system for vehicle tracking and traffic surveillance. Transportation Research Part C: Emerging Technologies, 6(4), 1998, pp. 271-288.
19. McLauchlan, P., D. Beymer, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In cvpr (p. 495). IEEE, 1997.
20. TITAN-Lab. *Using Mask-RCNN to detect traffic queues*, 2018, Available at: https://github.com/TITAN-lab/MaskRCNN_Traffic-Queues