A Hybrid RNN-CNN Approach to Automate Traffic Pattern Analysis in Roundabouts

Fayol Ateufack Zeudom[1]
fayol.ateufackzeudom@valpo.edu

Jay Grsossman[2]
jay.grossman@valpo.edu

Sami Khorbotly[1]
sami.khorbotly@valpo.edu

[1]Department of Electrical and Computer Engineering, [2]Department of Civil and Environmental Engineering
Valparaiso University, Valparaiso, IN, USA

Abstract – **Roundabouts have become a fundamental component in our urban transportation networks. Traffic pattern analysis is a critical step in the optimization of the design and utilization of those roundabouts. The traditional analysis performed by humans is labor intensive and cost prohibitive. In this work, we propose an alternative system that can automatically process a recorded aerial video of a roundabout and report the traffic patterns in it. The proposed system uses a pre-trained YOLO model to identify vehicles in the video frames. A hybrid RNN-CNN system tracks each vehicle from one frame to another as they travel through roundabout. Finally, statistical analysis is used to automate the counting of vehicles entering and exiting different sides of the roundabout. The output of the proposed system is a spreadsheet with the number of vehicles that entered/exited at all entry/exit points of the roundabout. The results show that the proposed system has an accuracy of 98.7%.**

I. Introduction

Traffic Intersections are essential components in our transportation networks. Roundabouts, or traffic circles, such as the one shown in Figure 1, are gaining increasing popularity as an alternative to the traditional light-operated intersections especially in dense urban areas. Research shows that installing roundabouts within municipalities positively impacts both traffic flows and business [1]. In that regard, it is very important to design and optimize those roundabouts for larger capacity, higher throughput, better flow, and minimized delay.

In order to achieve such an optimized design, transportation engineers extensively study the roundabouts and analyze their traffic patterns. The traditional method of conducting these studies consists of humans, equipped with stop watches and clipboards, counting the vehicles driving through a roundabout and analyzing their driving patterns. Human analysts can be conducting this study in-person, in real-time or by watching a previously recorded video. In both cases, it is extremely time-consuming and can be prohibitively expensive to hire humans to manually perform this job. For example, it is not uncommon to have 5-10 vehicles simultaneously drive through a medium sized roundabout. Accurately tracking those vehicles, in person, would require the presence of multiple humans at the site. In the case of a recorded video, accurate tracking can be achieved with only one human analyst but would require rewinding the video several times, which requires more time to complete the job.

Due to the recent advancements in machine learning and computer vision technology, it is now possible to automatically extract valuable information from images or video streams. In other words, it is possible for a computer-vision based system to identify the numbers and the locations of vehicles entering and exiting at the different locations of a roundabout. This offers a scalable, efficient, and cost-effective alternative to the traditional counting methods. In this paper, we present a system that is developed to take in an aerial drone-captured video of a roundabout and automatically analyze the video to generate a report of the traffic patterns in it.

Figure 1. An aerial picture of a traffic roundabout.

The rest of the paper is organized as follows: section II is a literature review of similar works. Section III discusses the video analysis process and explains why independently detecting vehicles in every frame is not enough. Section IV presents our proposed end-to-end solution to the problem. This section consists of 4 subsections detailing the 4 components of the proposed system. Section V discusses the obtained results, and the paper is concluded with a summary in section VI.

II. Literature Review

Image Processing and computer vision have been extensively used in the analysis and planning of urban traffic applications [2]. These applications include analyzing both parked vehicles in parking utilization studies [3] and in-motion vehicles when analyzing traffic patterns and vehicular trajectories [4]. Advanced Machine Learning (ML) techniques like Convolutional Neural Networks (CNNs) have shown noticeable effectiveness with the detection and tracking of vehicles in real-time, enabling accurate traffic volume counts and intersection turning pattern recognition [4]. However, challenges persist in real-world applications, such as irregular lighting, occlusions, and complex urban environments, impacting the performance of computer vision systems in tasks like traffic flow estimation and autonomous driving perception [5]. Addressing these challenges is crucial for practical deployment of computer vision solutions in traffic management and intelligent transportation systems. The main problem tackled in this paper is to identify that a vehicle is the same one through whole time it is visible in the video.

III. Video Analysis

Color videos are time-domain series of image frames displayed in succession. Every frame (or image) is an array of numbers representing the color intensities of every pixel in that image. The clarity or the amount of details displayed in the video is mainly defined by two parameters; the frame rate, or the number of frames per second (fps), and the frame resolution, defined by the number of pixels in every frame. In this work, we are using a 60 fps video at a resolution of 1920 x 1080 pixels in each frame.

In our application, the goal of tracking the vehicles throughout the video requires the accomplishment of two tasks: Identifying the vehicles (with their locations) in every frame and tracking the vehicles' locations from one frame to another. In other words, it is not enough to independently identify the vehicles in every frame. It is necessary, when identifying a vehicle in a frame, to also correlate that vehicle to another vehicle that was identified in a previous frame to achieve the time-domain tracking.

Fortunately, a good solution to achieve the first task of identifying vehicles in a frame already exists. This solution, a pre-trained ML YOLO model, is discussed in section IV. The second task is the more challenging one and is the main contribution of this paper. It may naively appear that the time-domain tracking can be achieved by simply connecting every vehicle in a frame *n* to the pixelwise nearest vehicle detected in the previous frame *n-1*. This solution, at least theoretically, makes sense. Because, in a 60 fps video, the time delay between two consecutive frames is less than 17*ms*. Afterall, how fast can a vehicle travel in such a short time? In reality, there are non-obvious issues that come into play. First of all, the YOLO model detector, while very effective, is not 100% accurate and may not detect all the vehicles in every single frame of the video. Similarly, a vehicle may be occluded for multiple frames by an object such as a tree and may, at no fault of the detector, go undetected for multiple frames. Finally, the video-capturing drone, while ideally stationary, may slightly change location and/or orientation throughout the video capturing process because of the wind or GPS imperfections. All these issues make tracking a vehicle by simply matching it to the pixelwise nearest vehicle from one frame to another an inaccurate and non-practical approach.

Researchers have developed solutions to deal with this problem. State-of-the-art algorithms such as DeepSORT [6] are used for object tracking (mostly human tracking). They use Kalman Filters (linear quadratic estimation) to predict the positions of humans in motion. It is based on convolutional neural networks trained to identify persons and re-identify them in subsequent frames. Unfortunately, when tested on aerial drone videos of roundabouts, DeepSORT performed poorly because of the non-linear nature of the vehicular motion at roundabouts. In other words, DeepSORT does a decent job predicting the position and tracking a vehicle driving mostly on a highway or a straight-line street. However, when a vehicle enters the roundabout, the non-linear path is much harder to predict causing a significant amount of vehicular ID switching. Moreover, DeepSORT was very slow and was unable to run in real-time on our 60 fps video. This is not surprising considering the computational cost of computing large covariance matrices it needs to compute for every frame.

IV. The YOLO Vehicle Detection Model

Analyzing traffic patterns in a roundabout can be broken down into two parts: Identifying the vehicles in a frame and tracking those vehicles from one frame to another. To identify the vehicles in an image, we used the pre-trained "You Only Look Once" (YOLO) computer vision model. YOLO is a state-of-the-art neural network-based real-time object detection and image segmentation model. It was developed by Joseph Redmon and Ali Farhadi at the University of Washington [7]. It was launched in 2015 and quickly gained popularity for its high speed and accuracy. For our system, we are using the YOLO version 8 (YOLOv8) roboflow version [8] that was fine-tuned on 5,317 aerial drone images of intersections with 43,310 annotations. The model has a precision score of 98.0% and a recall score of 97.9%.

The YOLOv8 model takes as input an aerial image of the roundabout and returns the same image with bounding boxes around the vehicles. The YOLOv8 model returns a percentage figure that indicates the model's confidence that the box contains a vehicle. This confidence level can be used as a design parameter to allow the system designer to set the threshold beyond which a vehicle is detected. Figure 2 shows the same picture of the roundabout shown in Figure 1 processed by the YOLOv8 model. You can see that every vehicle in the image is successfully recognized. Also, you can see in the figure that each box has a unique label to distinguish various vehicles from one another. For example, vehicles 6 and 10 are parked in the northeast side of the picture while vehicle 162 is entering the roundabout from the east entry point, and so on.

Figure 3 also shows an aerial picture of the same traffic roundabout, taken only one second (60 frames) after the one shown in Figure 1. If YOLOv8 model is applied to this image, the expectation is that it will be able to identify the vehicles in it. The only problem is that it will arbitrarily assign unique numbers to every box. This will result in a loss of continuity from one frame to another, which will make it impossible to track the time-domain motion of a specific vehicle.

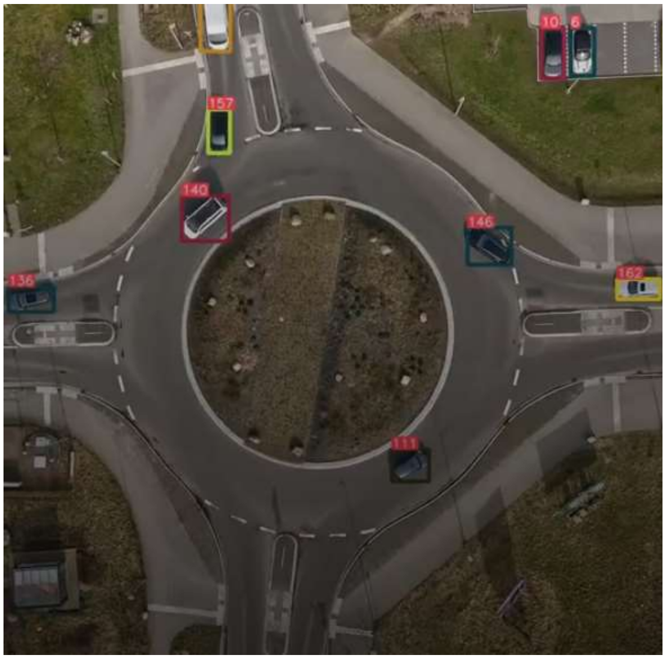

Figure 2. The roundabout picture shown in Figure 1 processed by the YOLOv8 model.

Figure 3. An aerial picture of the same traffic roundabout shown in Figure 1 after 1 second.

V. The Proposed System

As explained earlier, successfully identifying vehicles using the YOLOv8 model is necessary but not sufficient to track a vehicle and perform the traffic analysis. Additional work is needed to connect the different frames to each other and make sure that each vehicle is assigned the same label throughout its travel time in the roundabout. To achieve this goal, four subsystems were created.

A. The RNN-Based Prediction Subsystem

This subsystem uses the recent history of the locations of a vehicle to predict its future location. A Recurrent Neural Network (RNN) using the Long-Short-Term-Memory [9] architecture is trained with up to 20 previous locations, obtained from the positions in the most recent 20 frames. The RNN uses the existing data to extract information about how fast is the vehicle traveling and whether or not it is turning, and if it is turning, in what direction and at what angle. All this information is used to predict the 8 future positions of every vehicle in the frame.

The effectiveness of this subsystem was tested by providing the RNN with information about the last 20 bounding boxes and recording the predictions of the next 8 bounding boxes. The predicted bounding boxes were compared to the actual bounding boxes (unknown to the RNN). The results showed that the centroids of the bounding boxes were predicted with a mean square error of 2 pixels over the 8 bounding boxes.

B. The CNN Matching Subsystem

The Convolutional Neural Network (CNN) is another deep learning subsystem trained to compare the contents of sub-images. It takes in the contents of two bounding boxes, *i.e.* two vehicle images that may be at different angles and positions and determines whether they correspond to the same vehicle. The training of the model is achieved by, first employing DeepSort on sample videos, then manually labeling the IDs of vehicles in a “training” video. The trained CNN matching subsystem was tested on a “test” video that is different from the training video. The performance of the subsystem was measured by comparing its predictions to the actual outcomes determined by a human evaluator. The CNN subsystem delivered an accuracy of 96%.

C. The Decision Algorithm

The decision algorithm ties the two deep learning subsystems discussed above to achieve the best tracking performance. A flow chart of the algorithm is shown in Figure 4. The algorithm starts in an arbitrary frame *n-1.* For each of the identified vehicles in that frame, the RNN subsystem uses the 20 most recent available bounding boxes to predict the position of its future boxes (FB) in 8 frames {*n*, *n*+1, …, *n*+7}. When frame *n* becomes available, the YOLOv8 system detects a new set of bounding boxes, the Current Boxes (CB). At this point, the system compares the positions of the FBs predicted from the previous frames and the CBs obtained in the current frame. If the position of a CB has an overlap of 75% or more with the predicted position of a FB, it is decided that the CB corresponds to the same vehicle that was predicted to be at FB, and the same identity is assigned to it in the current frame. If the position of a CB has an overlap of more than 30% but less than 75% with the predicted position of a FB, this is a close call that requires using the CNN matching subsystem. If the system returns that the contents of the two boxes match, the CB corresponds to the same vehicle that was predicted to be at FB, and the same identity is assigned to it in the current frame. Otherwise, the CB is not identified as a box corresponding to a vehicle in the previous frame and is assigned a new identity. Finally, if a CB identified in frame *n* has less than 30% overlap with all the predicted FBs, it is also assigned a new identity.

Note that a vehicle that was given a new identity as a new vehicle has 7 more chances to be reclaimed if it matches a FB. However, a vehicle with no match in 8 previous frames is either a new vehicle that just appeared in the video, or it was already present, but its trajectory wasn't predicted well enough. This will lead to an error in the final results. The outcome of the system can be seen in Figure 5 showing the roundabout picture shown in Figure 3 processed by the proposed system. You can see that all vehicles in this figure are assigned the same label they were assigned in Figure 2. For example, in Figure 2, the white vehicle that was entering the frame from the east side (#162) is still assigned the same label (#162) in Figure 5. The same is true for all the vehicles in these pictures.

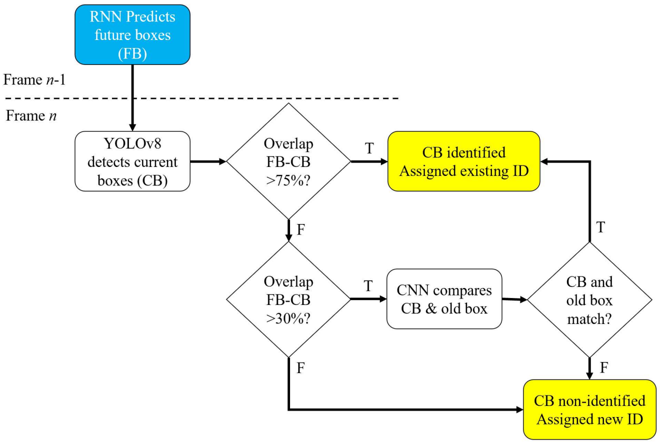
RNN Predicts future boxes (FB)
Frame *n*-1
Frame *n*
YOLOv8 detects current boxes (CB)
Overlap FB-CB >75%?
T
CB identified Assigned existing ID
F
Overlap FB-CB >30%?
T
CNN compares CB & old box
CB and old box match?
T
F
F
CB non-identified Assigned new ID

Figure 4. A flowchart of the decision algorithm.

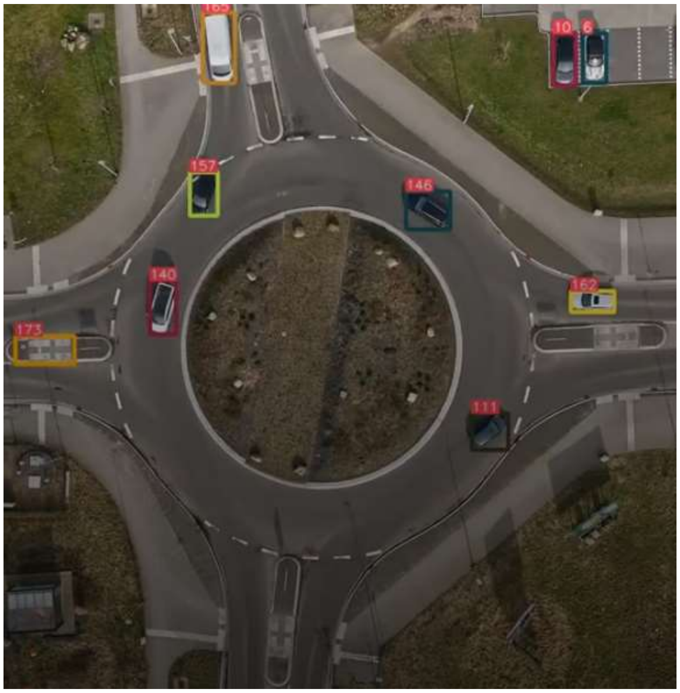

Figure 5. The roundabout picture shown in Figure 3 processed by the YOLOv8 model and the proposed system.

D. The Counting Subsystem

At this point, the three subsystems described above allow the tracking of the vehicles from the time they enter the roundabout till the time they exit it. The only missing piece is the ability to automatically count the number of vehicles that entered/exited at each entrance/exit of the roundabout. Since the drone is, ideally, stationary throughout the capture of the video, it is fair to expect that the pixel coordinates of the four entrances/exits of the roundabout will not change during the time of the video. These locations are labeled: north entrance, north exit, east entrance, east exit, south entrance, south exit, west entrance, and west exit.

To find the coordinates of all the entrances and exits, the system records the coordinates at which the vehicles make their first and last appearances in the video, respectively. The k-means clustering method [10] is used on those recorded coordinates. Any vehicle that has a horizontal movement of less than 108 pixels (10% of the width of the frame) was deemed to be motionless and was discarded in the counting process. See vehicles #6 and #10 in Figure 5. The Elbow method [11] was used to estimate the best number of clusters for entries and exits. The answer was four entries and four exits. In the counting process, a vehicle is counted to be entering/exiting at an entrance/exit cluster if its coordinates are within 1.5 standard deviations of that cluster.

VI. Results

The proposed system was applied to a drone-captured aerial video of a roundabout. The time duration of the video is 3 minutes and 27 seconds. During this time, a total of 78 vehicles traveled through the roundabout. Table I shows the numbers of vehicles traveling from each of the four entry points (rows) to each of the four exit points (columns). Additionally, each of the four columns shows both the number of vehicles counted by the proposed system and the actual number counted by a human. For example, in the first row, one can see that the system counted 0 vehicles entering from the North entrance and exiting from the North exit. This number agrees with the actual number obtained by a human count. Also in the first row, the system counted 9 vehicles entering from the North entrance and exiting from the East exit. This number also is confirmed by the actual number. The only time an error is observed is in the South entrance to South exit case where the system erroneously detected one vehicle that was not confirmed by the actual count. With a total of 1 erroneous vehicle counted out of 78 traveling through the roundabout, the proposed system demonstrated an accuracy of 98.7%.

VII. Summary

In this paper, we presented an end-to-end system to automatically analyze the traffic patterns in a traffic roundabout. The input to the system is an aerial video of the roundabout to by analyzed. The system uses a YOLOv8 system to identify vehicles in different frames of the video. The main contribution of this paper is a proposed tracking system that tracks vehicles throughout the video frames. This is accomplished by combining a RNN-based subsystem with a CNN-based one using an innovative decision algorithm. Once the vehicles are tracked from entry to exit, a counting subsystem is used to generate a spreadsheet reporting the number of vehicles in the video with their entry/exit points. The results showed the system has an accuracy of 98.7%.

Table I. A comparison of the results generated by the proposed system to the actual numbers.

Entry/Exit	North		East		South		West	
	System	Actual	System	Actual	System	Actual	System	Actual
North	0	0	9	9	5	5	1	1
East	6	6	0	0	12	12	9	9
South	9	9	18	18	**1**	0	3	3
West	0	0	4	4	2	2	0	0

References

[1] E. Russell, D. Landman, and R. Goddavarthy, “A Study of the Impact of Roundabouts on Traffic Flows and Business.” Final Report to K-TRAN, Nov. 2012. https://rosap.ntl.bts.gov/view/dot/25365

[2] Y. Pi, N. Duffieid, A. Behzadan, and T. Lomax, “Visual Recognition for Urban Traffic Data Retrieval and Analysis in Major Events Using Convolutional Neural Networks.” Springer Journal of Computational Urban Science, 2022.

[3] C. Smith, F. Ateufak-Zeudom, J. Grossman, and S. Khorbotly, “Computer Vision-Based System to Study Parking Utilization,” *IEEE Electro-Information Technology Conference*, Eau-Claire, WI, May 2024.

[4] U. Jana, J. Karmakar, P. Chakraborty, T. Huang, D. Ness, D. Ritcher, and A. Sharma, “Automated approach for computer vision based vehicle movement classification at traffic intersections.” Journal of Future Transportation. 2023. https://doi.org/10.3390/futuretransp3020041

[5] A. Talha, L. Jinlong, Y. Hongkai, C. Ruey, L. Yisheng, and K. Ruimin, “Deep learning based computer vision methods for complex traffic environments perception: a review. Journal of *Data Science for Transportation. Vol.* 6. No. 1. 2024. https://doi.org/10.1007/s42421-023-00086-7

[6] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric.” *IEEE International Conference on Image Processing*, Beijing, China, 2017. https://arxiv.org/abs/1703.07402

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection”. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016. https://arxiv.org/abs/1506.02640

[8] M. Tancred, VAID Computer Vision Project Dataset [Open Source Dataset]. Oct 2022. Roboflow Universe. Retrieved from https://universe.roboflow.com/mattias-tancred/vaid-jy5xo

[9] J. Schmidhuber and S. Hochreiter, “Long short-term memory,” *Journal of Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] K. Sinaga and M. Yang, “Unsupervised K-Means Clustering Algorithm,” IEEE Access, vol. 8, pp. 80716-80727, 2020. doi: 10.1109/ACCESS.2020.2988796.

[11] R. Thorndike, “Who Belongs in the Family?,” *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953. doi:10.1007/BF02289263. S2CID 120467216