TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Transportation Research Record
2019, Vol. 2673(6) 129–139

DOI: 10.1177/0361198119843255
journals.sagepub.com/home/trr
SAGE

Real-Time Pedestrian Detection Approach with an Efficient Data Communication Bandwidth Strategy

Mizanur Rahman[1], Mhafuzul Islam[1], Jon Calhoun[2], and Mashrur Chowdhury[1]

Abstract

Vehicle-to-pedestrian communication could significantly improve pedestrian safety at signalized intersections. However, it is unlikely that pedestrians will typically be carrying a low latency communication-enabled device with an activated pedestrian safety application in their hand-held device all the time. Because of this, multiple traffic cameras at a signalized intersection could be used to accurately detect and locate pedestrians using deep learning, and broadcast safety alerts related to pedestrians to warn connected and automated vehicles around signalized intersections. However, the unavailability of high-performance roadside computing infrastructure and the limited network bandwidth between traffic cameras and the computing infrastructure limits the ability of real-time data streaming and processing for pedestrian detection. In this paper, we describe an edge computing-based real-time pedestrian detection strategy that combines a pedestrian detection algorithm using deep learning and an efficient data communication approach to reduce bandwidth requirements while maintaining high pedestrian detection accuracy. We utilize a lossy compression technique on traffic camera data to determine the tradeoff between the reduction of the communication bandwidth requirements and a defined pedestrian detection accuracy. The performance of the pedestrian detection strategy is measured in relation to pedestrian classification accuracy with varying peak signal-to-noise ratios. The analyses reveal that we detect pedestrians by maintaining a defined detection accuracy with a peak signal-to-noise ratio 43 dB while reducing the communication bandwidth from 9.82 Mbits/sec to 0.31 Mbits/sec, a 31 $\times$ reduction.

According to National Highway Traffic Safety Administration, pedestrian fatalities are increasing in the United States every year. Pedestrians account for 14% of U.S. road fatalities with over 5,376 annual fatalities in 2015 (based on the latest statistics related to pedestrian fatalities) (*1*). Moreover, on average, 69,000 pedestrians are injured annually on U.S. roadways. However, recent research shows that enabling dedicated short-range communication (DSRC), which is a low latency data communication medium for safety applications, in a pedestrian hand-held device can increase pedestrian safety significantly through vehicle-to-pedestrian (V2P) communications (*2*). The DSRC-enabled V2P system gives a 360° view for which both the driver and the pedestrian are warned of a possible collision using DSRC-based safety alerts. However, it is very unlikely that all pedestrians will always be carrying a DSRC-enabled device with an activated pedestrian safety application. In addition, the current cellular communication network is not applicable for pedestrian safety applications because of its high data exchange latency (*3*).

Videos and images are commonly used in traffic monitoring of roadways and object detection applications in intelligent transportation systems (ITS) (*4*, *5*). Multiple traffic cameras at a signalized intersection could be used to accurately detect and locate pedestrians instead of a DSRC-enabled pedestrian hand-held device. Computing infrastructure can be used to process video data and detect pedestrians from the video data using the camera feed and then broadcast safety alerts about pedestrians to warn vehicles around a signalized intersection. In a typical ITS deployment, one or more transportation sensors (e.g., traffic cameras, roadway sensors) transmits sensing data (e.g., image, numerical, text sensor data)

[1]Center for Connected Multimodal Mobility (C^2M^2), Glenn Department of Civil Engineering, Clemson University, Clemson, SC
[2]Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC

Corresponding Author:
Address correspondence to Mizanur Rahman: mdr@clemson.edu

over a network to an ITS processing location (ITS center, such as a traffic management center) with substantially more processing capability. However, a centralized computing service cannot support real-time connected vehicle (CV) applications, such as a pedestrian safety application, owing to the often unpredictable network latency, high data loss rate, and expensive bandwidth, especially in a mobile environment like the CV environment (*4*, *6*).

Edge computing is a new computing concept that enables data analysis near the source of the data for real-time safety applications (*7*, *8*). Edge computing pushes the frontier of computing applications, data, and services away from centralized computing infrastructures to the edges. For example, a roadside data infrastructure located in the next immediate edge layer (e.g., roadside transportation infrastructure) from the associated CVs can offer computational ability to support CV safety applications.

However, network bandwidth between bandwidth-hungry roadside surveillance devices, especially from traffic cameras, and computing infrastructure limits the ability of real-time data streaming and processing for ITS applications. More efficient use of the bandwidth would allow for the deployment of real-time vision-based object detection using deep learning algorithms. Understanding this constraint will enable us to build systems that can be widely deployed. Lossy compression (LC) of data can significantly reduce data storage requirements for massive data volumes and decrease data transmission time in the communication network of an ITS system. Lossy compressed video data degrades the image quality and eventually reduces the chances of object detection, however, it does allow for a more efficient use of the bandwidth. Though the computation ability of some devices is currently limited, as is communication, it is possible they will be substantially improved in the near future. As the quality of communication and computation increases, LC will still be an effective technique to reduce bandwidth requirements (*9*). Reducing bandwidth requirements would allow use of less expensive components, lowering the price of deployment (*10*).

In this study, we developed an edge computing-based real-time pedestrian detection strategy that combined a deep learning-based pedestrian detection application and an efficient data communication approach to reduce bandwidth requirements while maintaining high object detection accuracy. We used LC techniques on video data to maximize bandwidth and increase its resiliency for data transmission between video surveillance and computing infrastructure. We then determined the trade-off between the reduction in the communication bandwidth requirements and the defined object detection accuracy with traffic camera data collected from signalized intersections in Clemson, SC.

The remainder of this paper is structured as follows. The Related Studies section describes related work on vision-based object detection models and LC for traffic camera data. The Research Method section presents the strategy for determining the compressibility of traffic camera data using a vision-based pedestrian detection algorithm. An analysis of an LC-based pedestrian detection strategy is then presented. The final section provides a concluding discussion.

Related Studies

There is limited work evaluating the performance of LC to problems in the ITS domain. The use of LC for video data requires a definition of correctness, and in the context of real-time object detection using vision-based techniques, the correctness metric is the classification precision and recall of the object detection model. Although lossless compression reduces the size of data and incurs no information loss, it results in low compression ratios and compression/decompression bandwidths (*11*). LC, however, significantly reduces the size of videos by introducing noise when representing each frame with fewer bits (*12*). Error-bounded LC allows for limiting the amount of loss via a user-defined error bound. The performance of LC is typically faster than lossless compression and results in large compression ratios (i.e., smaller files) (*13*). Because error-bounded LC allows user control of the accuracy level, the tolerance can be dynamically modified to ensure quality of service. The achievable compression bandwidth is therefore dependent on the magnitude of the LC's induced noise/error (*14*). LC techniques for video data are commonly used in online streaming platforms such as Netflix and YouTube (*15*) but have yet to be investigated in the ITS domain. Because of its novelty in this domain, there are several open questions about what is required from LC with respect to the compression ratio, the compression bandwidth, what constitutes an acceptable amount of noise, and where it is deployed in the ITS infrastructure. If LC is carried out, error accumulation that further degrades video quality is possible. To date, the impact of error accumulation resulting from multiple LC operations remains unstudied.

There are several video compression formats in common use, for example, H.262, H.264, high efficiency video coding (HEVC) (*16*). H.262 compresses each frame of a video by applying a discrete cosine transform to sub-blocks of the image and encoding the coefficients or using the previous and next frames and compressing the difference between the common sub-blocks (*17*). H.264 compresses similarly to H.262 but uses a lower bit rate to achieve the same level of quality (*18*). HEVC, like H.262 and H.264, identifies inter- and intra-frame regions of

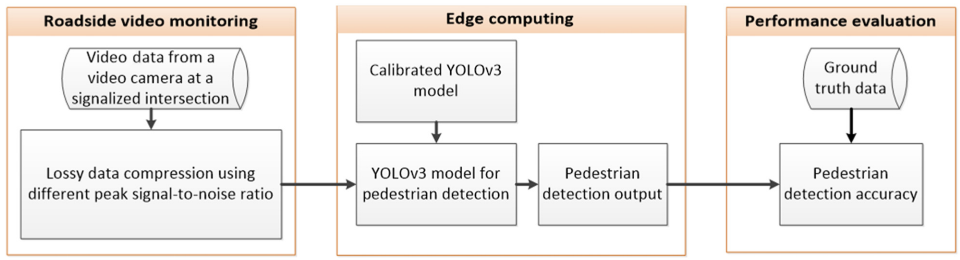

Figure 1. Edge computing-based pedestrian detection strategy using a lossy data compression technique.

similarity, but is further optimized to lower the bit rate while keeping the same quality level. HEVC is intended for use with high-resolution videos: 1080p, 4K, and 8K resolution (*19*). To create a video file, a video compression format is combined with an audio compression format into container format (video file) such as AVI, MP4, or FLV. It is often useful to convert video files between various formats, or for further improving the compression ratio for transmission or long-term storage. FFmpeg is a powerful cross-platform multimedia framework capable of decoding, encoding, transcoding, and filtering most audio and video formats (*20*). In this study, we used FFmpeg to investigate the tradeoff between LC of video data and pedestrian detection accuracy using a deep learning model.

For traffic operational analysis, different types of algorithms, such as embedded algorithms for loop detector systems, computer vision-based algorithms, and machine learning-based algorithms have been used for solving traffic-related problems (*21–24*). Machine learning-based algorithms improve the accuracy of traffic operational analyses compared with statistical methods, as they can learn from previous experiences of similar roadway conditions. To detect an object, machine learning systems take a classifier for that object and evaluate it at various locations and scales in a test image. Systems such as deformable parts models use a sliding window approach in which the classifier is run at evenly spaced locations over the entire image (*25*). More recent approaches like the Region-Convolutional Neural Network (R-CNN) use region proposal methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene (*26*). These complex pipelines are slow and hard to optimize because each individual component must be trained separately. Recently, Redmon et al. developed an object detection model, You Only Look Once (YOLO), to detect objects in real-time (*27*). This model can detect an object with a single network evaluation, unlike systems like R-CNN, which require thousands of evaluations for a single image. Because of this mechanism, YOLO is 1,000 times faster than R-CNN and 100 times faster than Fast R-CNN. Thus, it is applicable for real-time pedestrian detection in a connected transportation system.

Research Method

Figure 1 presents the general framework for the real-time pedestrian detection using the YOLOv3 (YOLO: version 3) deep learning model combined with LC of data. We extracted video data from roadside traffic monitoring cameras and used it as the input for the LC data algorithm. Following compression, we transferred the data to the edge-computing infrastructure and used a pre-trained and calibrated YOLOv3 model to detect pedestrians. From the YOLOv3 model, we found pedestrians detected every tenth of a second. Using the field-collected video data, we prepared the ground truth data by manually labeling pedestrians in the images to evaluate the pedestrian detection accuracy. Then, using the YOLOv3 model output and ground truth data, we evaluated the pedestrian detection accuracy.

Edge Computing

Edge computing is a new computing concept that enables data analytics at source for real-time safety applications (*7*, *8*). Edge computing pushes the frontiers of computing applications, data, and services away from centralized computing infrastructures to the edges. For example, a roadside data infrastructure located in the next immediate edge layer (e.g., roadside transportation infrastructure) from the associated CVs can offer computational ability to support CV safety applications. In general, an edge-centric CV systems consist of three edge layers (at least): i) mobile edge (e.g., CVs); ii) fixed edge (e.g., roadside infrastructures); and iii) system edge (e.g., backend server at a traffic management center) (*28*).

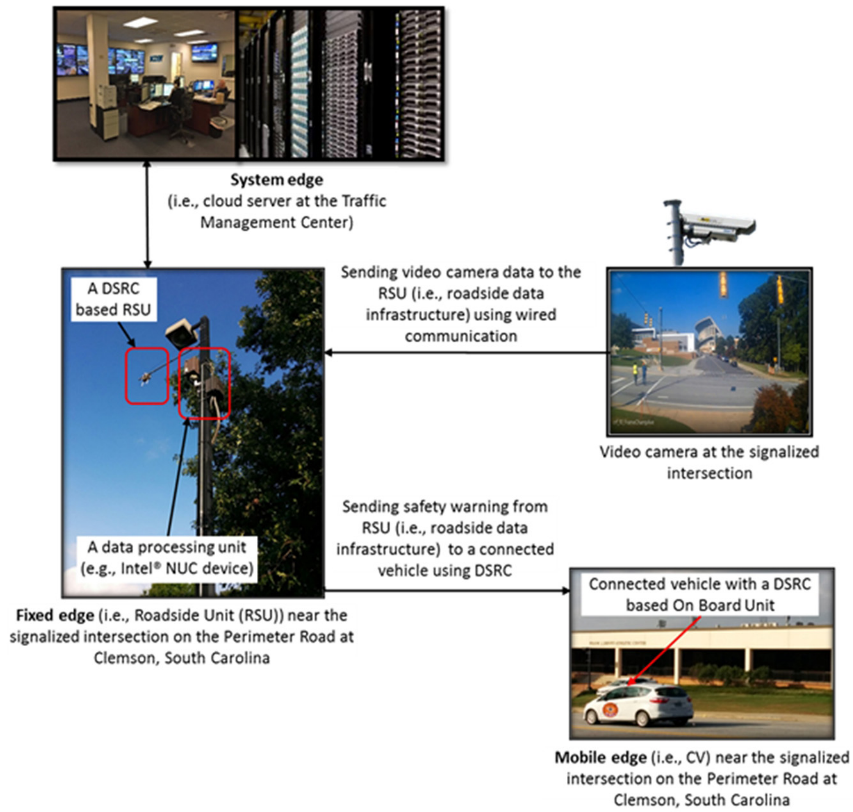

Figure 2. Edge-computing infrastructure for pedestrian detection.

The CVs participating in our system act as mobile edges, and were equipped with a DSRC-based on-board unit. A fixed edge includes a data processing unit, such as an Intel® NUC device, which has a similar processing capability to that used in our experiments, and a DSRC-based roadside unit (RSU) that communicates with CVs. A fixed edge can communicate with the mobile edges using DSRC and with a system edge using optical fiber/Wi-Fi. A system edge is a single end-point for a cluster of fixed edges. A fixed edge can be extended to support a video camera and other sensing devices, such as weather sensors and GPS. Fixed edges are connected to a system edge that can effectively serve as a backend resource. In our CV environment, the video camera installed at the intersection sent video data to the fixed edge (i.e., roadside infrastructure) and the pedestrian detection model was implemented in an RSU that included a data processing unit that ran the pedestrian detection model and a DSRC-based RSU that communicated with the CVs (as shown in Figure 2).

Lossy Video Compression Strategy

We utilized a commonly used LC format, H.264, on the videos originating from a signalized intersection in Clemson, SC, to quantify the impact of LC in an ITS environment. To compress video data, we used the FFmpeg multimedia tool, specifically, the Windows 64-bit binary release (N-82324-g872b358) (*29*). FFmpeg is a versatile tool supporting operations on a wide variety of video formats and containers. FFmpeg provides an extensive command line interface for video transcoding, filtering, and streaming of video, images, and audio.

YOLO Model for Pedestrian Detection

YOLOv3 Model. The YOLOv3 model (*27*) divides an image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. The model looks at the whole image at the test time so its predictions are informed by the global context in the image. The network architecture of the YOLOv3 model was inspired by the GoogLeNet model for image classification (*24*). The YOLOv3 network has 24 convolutional layers followed by two fully connected layers. The YOLOv3 model uses 1×1 reduction layers followed by 3×3 convolutional layers (*30*). Figure 3 presents the YOLOv3 model for

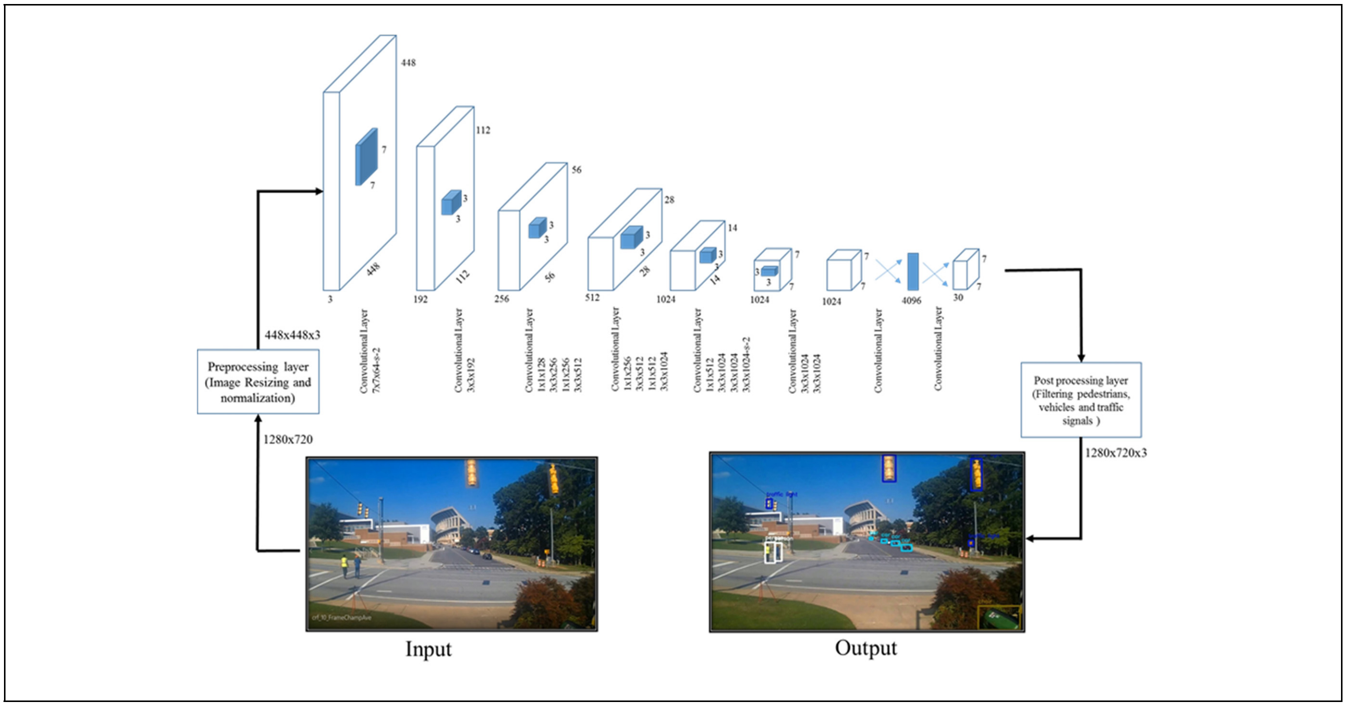

Figure 3. Integration of YOLOv3 model with compressed video data for pedestrian detection.

pedestrian detection using field-collected video input. To process video images, this model resizes the input image to 448 × 448p and normalizes the image at the preprocessing layer. In this study, the size of our input image was 1280 × 720p.

YOLOv3 Model Calibration. Initially, we use pre-trained YOLOv3 weights for pedestrian detection providing 81% accuracy, which is not sufficient for a pedestrian safety application. To improve the pedestrian detection accuracy of the YOLOv3, we calibrated the model using the video data collected from traffic cameras. To train the YOLOv3 model, we extracted 10 frames per second (fps) from the video. Next, we annotated each extracted frame to generate an annotated image file in the standard Pascal Visual Object Class format (*31*). In total, we collected 1,300 images from two different intersections: (i) data collected for a signalized intersection with a small number of pedestrians and during sunny weather; and (ii) data collected for a signalized intersection with a large number of pedestrians and during cloudy weather, both located in Clemson, SC. We used 900 images for training and 400 images for validation of the YOLOv3 model. However, after training the model, we found that it predicted multiple overlapping bounding boxes for a single pedestrian, which significantly reduced pedestrian detection accuracy. To remove the overlapping bounding boxes and to improve the pedestrian detection accuracy, we implemented a non-max suppression method (*32*). The non-max suppression method takes a bounding box with a high confidence score and removes the overlapping regions using an intersection of union (IoU) value larger than 0.6. The IoU is calculated using the following formulation:

$$IoU_{i,j} = \frac{O_{i,j}}{U_{i,j}}$$

where

$IoU_{i,j}$ = the IoU value of bounding boxes i and j,

$O_{i,j}$ = the overlapped area of bounding boxes i and j, and

$U_{i,j}$ = the area of the union of bounding boxes i and j.

Using the re-trained YOLOv3 model and non-max suppression method, the performance of pedestrian detection rose to 98% without any video compression.

Performance Measurements

To evaluate the accuracy of our vision-based pedestrian detection strategy, we measured the pedestrian classification accuracy at different video compression ratios. Accuracy refers to the percentage of video frames in which pedestrians are detected correctly. Manually labeled images were considered the ground truth values. For this binary classification problem, the classification accuracy was measured using the following formula for each test data set:

$$A = \frac{x}{n} * 100$$

where

A = the percentage of pedestrian detection accuracy,

x = the number of frames where pedestrians are correctly classified, and

n = the total number of frames for pedestrian detection events.

Analysis and Results

Data Description

We collected video data for two different scenarios: (1) Scenario 1: data collected for a signalized intersection with a small number of pedestrians and during sunny weather; and (2) Scenario 2: data collected for a signalized intersection with a large number of pedestrians and during cloudy weather. For Scenario 1, we collected video data from the Perimeter Road and Avenue of Champions intersection, which is a three-legged signalized intersection. We collected video data from three directions. For each direction, we collected video data for 30 min from 5:30 to 6:00 p.m. The size of each video data was 2 GB. For Scenario 2, we collected video data from College Avenue and Highway 93. We collected data from one roadway direction that had a high number of pedestrians; the data was collected between 4:30 and 5:00 p.m. For both scenarios, the raw video from the cameras pass through a post-processing step that transforms it from 30 fps to 10 fps. In a typical CV environment, we collect basic safety messages (BSMs) every tenth of a second to develop real-time safety-related applications. To mimic the standard for BSMs, we used 10 fps for detecting pedestrians and broadcasting safety alerts to the surrounding vehicles. To evaluate the performance of our detection model, we manually labeled each frame of the resulting videos to produce our ground truth data. The same 10 fps input video from the labeling step formed the input to the YOLOv3 model. We used calibrated pre-trained weights for this model of pedestrian detection as described in the previous section.

Lossy Video Compression Strategy

Using field-collected data, we compressed video for different constant rate factor (CRF) values to generate video data with various compression levels. The range of the CRF values was 0 to 51, where 0 indicates no compression, and 51 is the highest compression level. After that we calculated peak signal-to-noise ratio (PSNR) by comparing the original and compressed video files. PSNR is a well-known and widely used metric in digital signal processing. Thus, CRF of FFmpeg was used to compress the video. However, it was important to know that the video quality was measured by the PSNR and not CRF to ensure our results' independence from the FFmpeg tool. Thus, determining what PSNR values we were using for a specific CRF value was necessary. Table 1 provides a summary of average PSNR and CRF values for several evaluation scenarios.

Table 1. Video Constant Rate Factor and Peak Signal-to-Noise Ratio

Evaluation scenarios	Constant rate factor value for video compression	Average peak signal-to-noise ratio
1	10	56 dB
2	20	49 dB
3	30	43 dB
4	33	41 dB
5	35	40 dB
6	37	39 dB
7	40	37 dB
8	50	31 dB
9	51	30 dB

Figure 4 presents compressed images from the field-collected video data after FFmpeg, yielding different PSNR values. We observed that the quality of images deteriorated with decreasing PSNR values. Thus, as the video becomes more compressed, more noise is introduced that deteriorates quality. As the range of the CRF value was 0 to 51, we varied the CRF value within that range and calculated a range of PSNR values from 30 to 56 dB. A decreasing PSNR makes identification of pedestrians more challenging and results in a lower probability of detection.

Pedestrian Detection using YOLOv3 Model

Figure 5 presents a pedestrian detection output frame from the YOLOv3 model using compressed field-collected video data for Scenario 1 with varying PSNR values. As Figure 5 shows, we found that the YOLOv3 model detects pedestrians accurately with PSNR values of 39 dB or higher. However, a comprehensive evaluation of the pedestrian detection model is required compared with the ground truth data. For the comprehensive evaluation, we evaluated the pedestrian detection accuracy for different values of PSNR. Figure 6 shows that the maximum accuracy of pedestrian detection is 98% with the calibrated pre-trained weights and no compression for both scenarios 1 and 2. For Scenario 1, for LC video with PSNRs of 56 dB, 49 dB, and 43 dB, the detection accuracy remains constant at 98%. However, as the video becomes more distorted as a result of high levels of compression, the accuracy of the YOLOv3 model starts to decrease. At the highest level of compression (PSNR 30 dB) the prediction accuracy is 60% for Scenario 1 and

Figure 4. Compressed images from the field-collected data using increasing levels of video compression and different PSNR values.

55% for Scenario 2. This analysis shows that at a PSNR compression threshold of 43 dB, acceptable pedestrian detection accuracy is maintained. The Scenario 2 evaluation yielded a similar PSNR pattern for pedestrian detection accuracy. Therefore, the number of pedestrians does not affect detection accuracy until a PSNR value of 43 (as shown in Figure 6). However, it can be seen in Scenario 2 that after a PSNR value of 43, accuracy decreases more rapidly. Scenario 2's environmental conditions were not as ideal as Scenario 1's. Bad weather and dark environments have been shown to reduce video quality (*33*, *34*). Since we used error-bounded LC, we were able to dynamically adapt the loss in the video based on environmental conditions. In the worst case, our scheme reverted to using raw video footage from the camera. Another alternative to improve detection accuracy is to apply video processing techniques to improve quality by removing noise, or to change the brightness/contrast. We will address these issues in our future work. For calculating bandwidth requirements for each PSNR values, we used the following equation:

Figure 5. Pedestrian detection output from the YOLOv3 model using increasing levels of video compression and different CRF and PSNR values.

$$Required\ bandwidth = \frac{S}{T}$$

where S is the size of the video (Mbits) and T is the duration of the video (s). Table 2 presents the bandwidth requirements for different PSNR values. We calculated the bandwidth requirements for PSNR values that did not cause deterioration in the detection accuracy (56 dB, 49 dB, and 43 dB). For these configurations, the corresponding bandwidth requirements were 4.98 Mbits/s, 1.63 Mbits/s, and 0.31 Mbits/s, respectively. Compared with the original bandwidth requirements for the uncompressed video (9.82 Mbits/s), we were able to achieve the

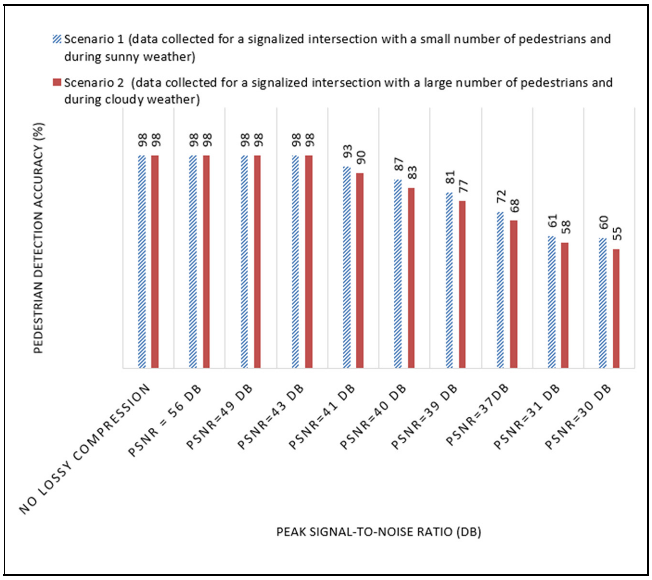

Figure 6. Pedestrian detection accuracy for different PSNR values and scenarios.

Table 2. Communication Bandwidth Requirement for Different PSNR Values

CRF value	PSNR value	Required bandwidth (Mbits/s)
0	NA	9.82
10	56 dB	4.98
20	49 dB	1.63
30	43 dB	0.31

Note: CRF = constant rate factor; PSNR = peak signal-to-noise ratio.

same pedestrian accuracy at only 0.31 Mbits/s. This resulted in a 31 × reduction in the bandwidth required to send the video feed to a processing facility.

LC significantly reduces the storage and bandwidth requirements to archive and transmit video for ITS applications (*35*). Reducing storage requirements allows for more minutes of video to be stored or archived without modification to the underlying hardware (*36*). Reducing the bandwidth requirements for communicating video allows for more concurrent video feeds to be sent given a fixed bandwidth; therefore, more cameras can transmit their feeds without needing upgrades to the network. Provided the system can compress more frames per second than the video feed creates, the video can be compressed in real-time.

In our case study, false positive were found for PSNR values less than 43 dB. However, as we are not recommending PSNR values less than this for compressing video data, this had no impact on our approach in the operational environment. However, such false positives can be eliminated by using a method such as the object symmetry approach (*37*).

Contributions of the Paper

The primary contribution of our paper is the combination a vision-based pedestrian detection model, the YOLOv3, with an LC for reducing data communication bandwidth while maintaining a defined pedestrian detection accuracy. Our approach highlights the feasibility of using LC to lower the bandwidth requirement for pedestrian detection. Error-bounded LC significantly reduces the storage and bandwidth requirements to archive and transmit the video. Reducing storage requirements allows for more minutes of video to be stored or archived with no modification to the underlying hardware. Reducing the bandwidth requirement of the video feed allows for more concurrent video feeds to be sent given a fixed bandwidth; therefore, more cameras can transmit their feeds with no upgrades to the network. Thus, our approach directly contributes to a real-world implementation of a pedestrian detection technique with limited bandwidth and storage capacity, as computation power and storage capacity will be limited by the edge computing-based roadside infrastructure. In addition, using PSNR as our compressed video quality metric will allow other researchers to use PSNR values that yield an acceptable combination of detection accuracy and a reduction in the communication bandwidth requirements for evaluating other video compression formats. In addition, acceptable PSNR values can form a starting point for determining the optimal configuration for integration into real-world deployment. Future ITS deployments in CV environments will collect, analyze, and transmit massive amounts of data wirelessly from CVs, roadside sensors, cell phones, and cameras to roadside edge-computing devices. As the number of connected devices increases, the need for reducing bandwidth requirements will grow. This work provides an initial step in the exploration and integration of techniques that reduce bandwidth requirements without sacrificing system accuracy.

Conclusions

Edge computing enables analytics at the data source for real-time safety applications. In this study, we developed and evaluated an edge computing-based real-time pedestrian detection strategy combining a pedestrian detection algorithm and an efficient data communication approach to reduce bandwidth while maintaining high object detection accuracy. We utilize LC video data at different quality levels to determine the tradeoff between the reduction of communication bandwidth requirements and a defined object detection accuracy. The performance of the pedestrian detection strategy was measured in relation to pedestrian classification accuracy with varying PSNRs. The analyses revealed that we detected pedestrians by maintaining a defined detection accuracy (98%) with a

PSNR of 43 dB, while reducing the communication bandwidth from 9.82 Mbits/s to 0.31 Mbits/s, 31× reduction in bandwidth. This strategy facilitates intelligent uses of LC that will allow engineers to effectively increase both bandwidth and storage capacity enabling them to work with larger quantities of data. Our method is applicable to detecting any external objects like vehicles, bicycles, motorcycles, and so forth, which are not equipped with CV devices. However, pedestrians are the most vulnerable road users. Thus, our research focused on improving intersection pedestrian safety to show the applicability of our strategy. Future work will evaluate our technique at other busier intersections under various weather conditions and times of day to show the efficacy of our approach for pedestrian detection. Future work will also consider unexplored trade-offs important to embedded transportation cyber-physical systems such as energy efficiency, compression/decompression time variation, and effective bandwidth using error bound LC.

Acknowledgments

This material is based on a study supported by the Center for Connected Multimodal Mobility (C^2M^2) grant headquartered at Clemson University, SC.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: MR, MI, MC, JC; data collection: MR, MI; interpretation of results: MR, MC, JC; draft manuscript preparation: MR, MC, JC, MI. All authors reviewed the results and approved the final version of the manuscript.

References

1. Pedestrian Safety. https://www.nhtsa.gov/road-safety/pedestrian-safety. Accessed August 1, 2018.
2. Wu, X., R. Miucic, S. Yang, S. Al-Stouhi, J. Misener, S. Bai, and W. H. Chan. Cars Talk to Phones: A DSRC Based Vehicle–Pedestrian Safety System. *2014 IEEE 80th Vehicular Technology Conference (VTC Fall)*, Vancouver, BC, Canada, IEEE, 2014, pp. 1–7.
3. Dey, K. C., A. Rayamajhi, M. Chowdhury, P. Bhavsar, and J. Martin. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Communication in a Heterogeneous Wireless Network–Performance Evaluation. *Transportation Research Part C: Emerging Technologies*, Vol. 68, 2016, pp. 168–184.
4. Chowdhury, M., A. Apon, and K. Dey. *Data Analytics for Intelligent Transportation Systems*. Elsevier, Amsterdam, the Netherlands, 2017.
5. Zhang, G., R. Avery, and Y. Wang. Video-Based Vehicle Detection and Classification System for Real-Time Traffic Data Collection using Uncalibrated Video Cameras. *Transportation Research Record: Journal of the Transportation Research Board*, 1993. 2007: 138–147.
6. Deka, L., and M. Chowdhury. *Transportation Cyber-Physical Systems*. Elsevier, Amsterdam, the Netherlands, 2018.
7. Shi, W., J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, Vol. 3, No. 5, 2016, pp. 637–646.
8. Rayamajhi, A., M. Rahman, M. Kaur, J. Liu, M. Chowdhury, H. Hu, J. McClendon, K. C. Wang, A. Gosain, and J. Martin. ThinGs In a Fog: System Illustration with Connected Vehicles. *2017 IEEE 85th Vehicular Technology Conference* (VTC Spring), Sydney, Australia, IEEE, 2017, pp. 1–6.
9. Sathish, V., M. J. Schulte, and N. S. Kim. Lossless and Lossy Memory I/O Link Compression for Improving Performance of GPGPU Workloads. *2012 21st International Conference on Parallel Architectures and Compilation Techniques (PACT)*, Minneapolis, MN, IEEE, pp. 325–334.
10. Fritz, M. H. Y., R. Leinonen, G. Cochrane, and E. Birney Efficient Storage of High Throughput DNA Sequencing Data using Reference-Based Compression. *Genome Research*, Vol. 21, No. 5, 2011, pp. 734–740.
11. Son, S. W., Z. Chen, W. Hendrix, A. Agrawal, W. K. Liao, , and A. Choudhary. Data Compression for the Exascale Computing Era-Survey. *Supercomputing Frontiers and Innovations*, Vol. 1, No. 2, 2014, pp. 76–88.
12. Ohm, J. R., G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand. Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC). In *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 12, 2012, pp. 1669–1684.
13. Zemliachenko, A., V. Lukin, N. Ponomarenko, K. Egiazarian, and J. Astola. Still Image/Video Frame Lossy Compression Providing a Desired Visual Quality. *Multidimensional Systems and Signal Processing*, Vol. 27, No. 3, 2016, pp. 697–718.
14. Di, S., and F. Cappello. Fast Error-Bounded Lossy HPC Data Compression with SZ. *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Chicago, IL, 2016, pp. 730–739.
15. Cock, J. D., A. Mavlankar, A. Moorthy, and A. Aaron. A Large-Scale Video Codec Comparison of x264, x265 and libvpx for Practical VOD Applications. *Proc. SPIE 9971, Applications of Digital Image Processing XXXIX*, 997116, 27 September 2016. https://doi.org/10.1117/12.2238495.
16. Sayood, K. *Introduction to Data Compression*. Morgan Kaufmann, Burlington, MA, 2017.
17. ITU-T and ISO/IEC JTC 1. Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video. *ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2)*, 1994.
18. ITU-T and ISO/IEC JTC 1. Advanced Video Coding for Generic Audiovisual Services. *ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4)*, April 2017.
19. ITU-T and ISO/IEC JTC 1. High Efficiency Video Coding. *ITU-T Rec. H.265 and ISO/IEC 23008-2*, 2018.
20. FFmepg Developers. ffmpeg tool (Version N-82324-g872b358) [Software]. http://ffmpeg.org. Accessed August 1, 2018.

21. Coifman, B., and S. Kim. *Assessing the Performance of the Speed Info Sensor*. No. FHWA/OH-2013/21. Ohio Department of Transportation, 2013. https://ntl.bts.gov/lib/51000/51200/51209/134723_FR.pdf. Accessed July 28, 2017.
22. Ma, Y., M. Chowdhury, A. Sadek, and M. Jeihani. Integrated Traffic and Communication Performance Evaluation of an Intelligent Vehicle Infrastructure Integration (VII) System for Online Travel-Time Prediction. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 3, 2012, pp. 1369–1382.
23. Mossi, J. M., A. Albiol, and V. N. Ornedo. Real-Time Traffic Analysis at Night-Time. *Proc., 18th IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 2941–2944.
24. Rong, Y. U., W. A. N. G. Guoxiang, J. Zheng, and W. A. N. G. Haiyan. Urban Road Traffic Condition Pattern Recognition Based on Support Vector Machine. *Journal of Transportation Systems Engineering and Information Technology*, Vol. 13, No. 1, 2013, pp. 130–136.
25. Felzenszwalb, F., R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, 2010, pp. 1627–1645.
26. Girshick, R., J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, IEEE, 2014, pp. 580–587.
27. Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 779–788.
28. Chowdhury, M., M. Rahman, A. Rayamajhi, S. M. Khan, M. Islam, Z. Khan, and J. Martin. Lessons Learned from the Real-World Deployment of a Connected Vehicle Testbed. *Transportation Research Record: Journal of Transportation Research Board*, 2018. 2672(22): 10–23.
29. FFmpeg. https://www.ffmpeg.org/. Accessed August 1, 2018.
30. Lin, M., Q. Chen, and S. Yan. *Network in Network*. 2014. https://arxiv.org/abs/1312.4400.
31. Everingham, M., S. M. A. Eslami, L. Van Gool, C. K. I. Williams, and J. Winn. The PASCAL Visual Object Classes Challenge: A Retrospective. 2015, pp. 98–136. https://doi.org/10.1007/s11263-014-0733-5.
32. Rothe, R., M. Guillaumin, and L. Van Gool 2014, November. Non-Maximum Suppression for Object Detection by Passing Messages between Windows. In *Asian Conference on Computer Vision*. Springer, Cham, Switzerland, 2014, pp. 290–306.
33. Cozzolino, A., F. Flammini, V. Galli, M. Lamberti, G. Poggi, and C. Pragliola Evaluating the Effects of MJPEG Compression on Motion Tracking in Metro Railway Surveillance. *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, Berlin, Heidelberg, Germany, 2012, pp. 142–154.
34. Jia, Z., H. Wang, R. E. Caballero, Z. Xiong, J. Zhao, and A. Finn. A Two-step Approach to See-through Bad Weather for Surveillance Video Quality Enhancement. *Machine Vision and Applications*, Vol. 23, No. 6, 2012, pp.1059–1082.
35. Sathish, V., M. J. Schulte, and N. S. Kim. Lossless and Lossy Memory I/O Link Compression for Improving Performance of GPGPU Workloads. *2012 21st International Conference on Parallel Architectures and Compilation Techniques (PACT)*, Minneapolis, MN, IEEE, 2012, pp. 325–334.
36. Fritz, M. H. Y., R. Leinonen, G. Cochrane, and E. Birney. Efficient Storage of High Throughput DNA Sequencing Data using Reference-Based Compression. *Genome Research*, Vol. 21, No. 5, 2011.
37. Afrakhteh, M., and M. Park. Double Phase Pedestrian Detection with Minimal Number of False Positives per Image. *Proceedings of the 6th International Conference on Informatics, Environment, Energy and Applications*, Jeju Island, Korea, ACM, 2017, pp. 107–111.

The Standing Committee on Intelligent Transportation Systems (AHB15) peer-reviewed this paper (19-04803).