Journal of Real-Time Image Processing
https://doi.org/10.1007/s11554-021-01165-0

ORIGINAL RESEARCH PAPER

Dynamic error-bounded lossy compression to reduce the bandwidth requirement for real-time vision-based pedestrian safety applications

Mizanur Rahman[1] · Mhafuzul Islam[2] · Cavender Holt[3] · Jon Calhoun[3] · Mashrur Chowdhury[2]

Received: 31 January 2021 / Accepted: 17 August 2021

Abstract
As camera quality improves and their deployment moves to areas with limited bandwidth, communication bottlenecks can impair real-time constraints of an intelligent transportation systems application, such as video-based real-time pedestrian detection. Video compression reduces the bandwidth requirement to transmit the video which degrades the video quality. As the quality level of the video decreases, it results in the corresponding decreases in the accuracy of the vision-based pedestrian detection model. Furthermore, environmental conditions, such as rain and night-time darkness impact the ability to leverage compression by making it more difficult to maintain high pedestrian detection accuracy. The objective of this study is to develop a real-time error-bounded lossy compression (EBLC) strategy to dynamically change the video compression level depending on different environmental conditions to maintain a high pedestrian detection accuracy. We conduct a case study to show the efficacy of our dynamic EBLC strategy for real-time vision-based pedestrian detection under adverse environmental conditions. Our strategy dynamically selects the lossy compression error tolerances that maintain a high detection accuracy across a representative set of environmental conditions. Analyses reveal that for adverse environmental conditions, our dynamic EBLC strategy increases pedestrian detection accuracy up to 14% and reduces the communication bandwidth up to 14 × compared to the state-of-the-practice. Moreover, we show our dynamic EBLC strategy is independent of pedestrian detection models and environmental conditions allowing other detection models and environmental conditions to be easily incorporated.

Keywords Error-bounded lossy compression (EBLC) · Efficient bandwidth usage · Real-time processing · Vision-based object detection · Pedestrian detection

1 Introduction

The number of pedestrian fatalities has risen each year with over 6000 reported deaths in 2018 alone, an increase of over 30% compared to 2009 [1]. The presence of a pre-crash warning system, which tracks both vehicles and pedestrian movements, could have prevented most of these pedestrian-related crashes. Addressing the number of traffic fatalities is a matter of national importance [2]. As transportation begins to shift toward autonomous and self-driving vehicles, roadways and intersections are being outfitted with safety devices, such as cameras and sensors to improve pedestrian safety [3, 4]. Even modern vehicles include an in-vehicle vision-based pedestrian warning system to assist drivers in avoiding pedestrian-related crashes [5, 6]. However, in-vehicle pedestrian warning systems do not provide any pre-crash warning to pedestrians.

✉ Mizanur Rahman
mizan.rahman@ua.edu

Mhafuzul Islam
mdmhafi@clemson.edu

Cavender Holt
cavendh@g.clemson.edu

Jon Calhoun
jonccal@clemson.edu

Mashrur Chowdhury
mac@clemson.edu

[1] Department of Civil, Construction and Environmental Engineering, The University of Alabama, Tuscaloosa, AL, USA

[2] Glenn Department of Civil Engineering, Clemson University, Clemson, SC, USA

[3] Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, USA

Published online: 07 September 2021

Vehicle-to-pedestrian (V2P) communication can provide a 360° view, where a human driver in a connected vehicle as well as a pedestrian at an intersection can receive a safety warning notification if there is a potential pedestrian-vehicle collision risk. However, a pedestrian must carry a hand-held device, which must have a low latency wireless communication technology, and a pedestrian must turn on the pedestrian safety application in his/her phone. The C-V2X (cellular vehicle-to-everything) direct or sideline communication is an example of a low latency communication technology. It is unlikely that such communication technology will be available to all pedestrians' hand-held devices and the pedestrian safety application will be activated in their devices while they are crossing an intersection. Thus, cameras on poles covering the intersection area can be used to monitor pedestrians at an intersection and transmit the video to a roadside transportation infrastructure with wireless communication capabilities. A vision-based safety alert system uses an object detection algorithm to detect pedestrians, generates safety warnings and broadcasts these warnings to surrounding connected vehicles (i.e., a vision-based pedestrian safety alert system) as presented in [6]. For a non-connected vehicle, generated safety warnings from the system can be carried out through dynamic message signs for drivers, or audible warnings or warning signs for pedestrians at an intersection to warn approaching drivers and pedestrians, correspondingly, of an impending collision risk. With this strategy, there is no requirement for pedestrians to carry a low latency communication technology enabled handheld device.

Figure 1 presents such a pedestrian safety alert system, where the cameras are on a light pole at an intersection equipped with vision-based safety alert systems in Clemson, South Carolina, USA [6]. As vision-based pedestrian detection relies on image processing of frames taken from roadside cameras (as shown in Fig. 1) at signalized intersections, the video data must be sent to a roadside video image processing unit (i.e., a part of a roadside transportation infrastructure, as shown in Fig. 1) or to the cloud for video processing. As the size of the video increases, so too does the latency to transfer video from a video camera to a roadside video image processing unit. Increasing the latency

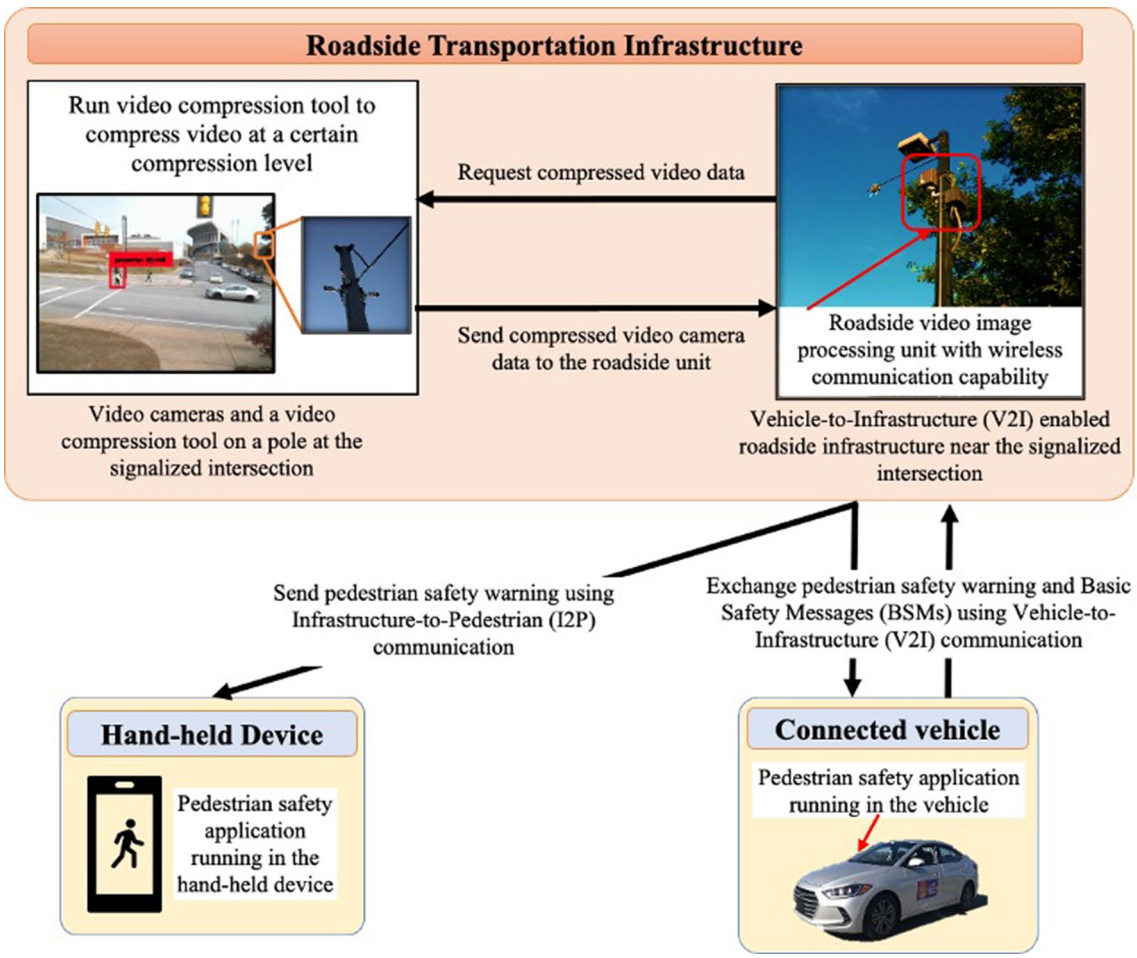

Fig. 1 Utilization of pedestrian detection for a vision-based safety alert system at a signalized intersection

decreases the likelihood that pedestrians are detected reliably, where the reliability will depend on satisfying the real-time latency requirement as needed by the corresponding application, and without detection of pedestrians within the low latency threshold for any safety critical applications, improvements to safety on roadways will be infeasible. Moreover, as high-resolution cameras and an increase in the number of connected devices compete for the available communication bandwidth, the bandwidth available may prohibit any safety–critical applications, such as the vision-based safety alert system. Thus, an efficient communication of video data, from video cameras to processing units, for pedestrian detection is required to ensure the safety of pedestrians on roadways in the vision-based safety alert system presented here. The lossy data compression strategy presented in this paper can significantly decrease the data transmission latency in a communication network between a camera and a video image processing unit of a roadside transportation infrastructure, as well as reduce video data storage requirements.

Data compression trades computational time for a reduction in data size. Video compression algorithms employ lossy data compression, which trades inaccuracies in the video's frames for larger reductions in video size [7]. However, as the level of loss increases, the quality of the video decreases. For image and video compression algorithms, this typically results in "blocking," where pixel blocks are approximated by a single value [8]. Quantifying the level of acceptable loss defines the video compression limit for a given algorithm. Common metrics to evaluate the level of loss in video data include peak signal-to-noise ratio (PSNR), root-mean-squared error (RMSE), and structural similarity index (SSIM) [9]. Lossy compression algorithms that ensure a fixed level of loss in the compressed data are referred to as error-bounded lossy compression (EBLC) algorithms.

Through the judicious use of EBLC, video streaming companies, such as Netflix and YouTube, optimize the video quality given the amount of available bandwidth [10]. Our prior work shows the utility of using EBLC for real-time pedestrian safety applications to reduce the bandwidth requirements to transmit video data by up to 30× without deterioration in the pedestrian detection accuracy [5]. Although showing the potential of EBLC for pedestrian detection, our prior work has several limitations. First, in the prior work, we use a single static error tolerance for the deployment. Thus, the static error tolerance of the system does not adapt to different situations, hurting utility and safety. Next, in our prior work, compressed data is fed into a detection model trained for uncompressed data. This degrades the detection accuracy and decreases pedestrian safety. Finally, our prior work evaluates the EBLC system with a limited number of environmental conditions. However, as we show in Sect. 5 of this paper, environmental conditions (e.g., rain, darkness) impact the ability to leverage EBLC by making it more difficult to maintain high pedestrian detection accuracy. Thus, adapting the error tolerance based upon environmental conditions ensures pedestrian detection accuracy does not deteriorate in adverse environmental conditions.

The objective of this paper is to reduce bandwidth requirements for pedestrian detection in adverse environmental conditions by developing a real-time EBLC strategy to dynamically change the video compression threshold depending on the current environmental conditions while maintaining a high pedestrian detection accuracy. Moreover, to further improve pedestrian detection accuracy, we calibrate the detection model based on the compression level to improve detection accuracy on highly compressed data. Using this strategy, we maintain an appropriate pedestrian accuracy across a representative selection of environmental conditions.

2 Contribution of the paper

The primary contribution of our paper is the development of a dynamic EBLC strategy for video feeds from a roadside camera to edge devices, such a roadside computer, used for real-time pedestrian detection and potential crash alert. The dynamic EBLC strategy accounts for environmental factors and ensures a defined pedestrian detection accuracy is maintained while effectively reducing the communication bandwidth requirements for a wireless video streaming application. We demonstrate that our strategy is independent of any specific pedestrian detection model such that any pedestrian detection model can be used within the strategy presented in this paper. In addition, any other environmental factor, such as snow and rain, can be incorporated in our dynamic EBLC strategy by following the steps presented in the Sect. 4 of the paper for incorporating any new environmental condition, such as snow. Moreover, our strategy is dynamic, which is applicable to image recognition applications beyond pedestrian detection, where environmental conditions or the visual quality of video feeds vary overtime. The dynamic EBLC strategy reduces the communication bandwidth usage of a video feed, which allows more videos to be transmitted concurrently through a fixed bandwidth. Furthermore, dynamic EBLC significantly reduces the storage requirements for video archiving for later offline analysis. Thus, the dynamic EBLC strategy presented in this paper allows storage of videos of longer duration without the need of modifying the underlying hardware.

3 Related work

This section describes existing work related to EBLC, pedestrian detection, and image classification methods. Examining the limitations of the existing methods, we identify an appropriate lossy video compression technique, pedestrian detection, and image classification method for our dynamic EBLC strategy.

3.1 Error-bounded lossy compression

Lossless data compression, such as the Lempel-Ziv algorithm (LZ77) [11], allows for the reduction in the data size with no loss in the data's accuracy. Lossy compression (LC) significantly reduces data sizes and offers better compression ratios than lossless compression, but at the expense of inaccuracies in the decompressed data [12]. In the context of video compression, LC compresses by introducing noise into each frame by representing the frame with fewer bits [7]. Typically, the larger the loss in data accuracy, the larger the compression ratio [10, 13]. Current state-of-the-art LC algorithms known as EBLC algorithms offer the ability to control the level of loss introduced when compressing the data [14]. Modern video compression algorithms, such as H.264 [15] and high-efficiency video coding (HEVC) [16], are optimized for high-resolution videos by encoding more information into each compressed bit. H.264 and HEVC compress videos by identifying regions of inter- and intra-frame similarity and then applying transforms, such as the discrete cosine transform [17] and encoding the coefficients or using delta encoding to encode the differences between two frames.

Previous work in the area of lossy compression and object detection have considered approaches to improve both the bitrate of communication and the accuracy of object detectors run on the video frames. In one approach [18], object saliency maps are used as a preprocessing step to improve the compression of the video frames. This video encoding method enables performance benefits in the communication bitrate and accuracy of the object detection model. Another approach [19] finds that temporal fluctuations in irrelevant background portions of the frames caused degradation of object detection performance. To remedy this performance deficit, the authors in [19] propose an encoding method to stabilize the temporal fluctuations in the frames. As a result of this encoding scheme, the bitrate and accuracy of detection improve. The methods proposed in this paper tackle generalizing lossy compression by focusing on error-bounded lossy compressors such that we determine the quality level of compression and its impact on pedestrian detection. Moreover, we consider pedestrian detection in dynamically changing environments using compression, which is not considered in prior research.

Due to the need to understand the impact of inaccuracies on the quality-of-service, EBLC has not received much attention in the intelligent transportation systems (ITS) domain. In the context of pedestrian detection, quality-of-service is determined by maintaining fixed detection accuracy. Any deterioration in the detection accuracy can lead to unsafe situations for pedestrians. Our prior work [5] shows that using EBLC and a static error tolerance reduces bandwidth requirements for pedestrian detection by over 30× with no deterioration in detection accuracy. Furthermore, this prior work shows that a single static lossy compression tolerance does not work as well on cloudy or rainy weather conditions as it works in sunny weather conditions. Throughout the day and year environmental conditions change, degrading the utility of a static lossy compression approach. By dynamically adapting the error tolerance and the performence of the detection model, we maintain a high pedestrian detection accuracy in adverse environmental situations.

3.2 Machine learning methods for pedestrian detection and environment classification

The advent of deep learning significantly improved the accuracy and computational time of object detection and classification. The state-of-the-art deep learning-based object detection models operate in real time and provide a high detection accuracy. Object detection models are classified into two categories: (i) region-based object detection and (ii) single-shot object detection. Region-based object detection models include: Region-Convolutional Neural Network (R-CNN) [20]; Fast R-CNN [21]; and Faster R-CNN [22]. The single-shot object detection models include: Single Shot MultiBox Detector (SSD) [23] and You Only Look Once—Version 3 (YOLOv3) [24]. Single Shot Multibox Object Detectors encapsulate all computation within a single network. This allows for easy training and easy integration into systems that need object detection. SSD is a comparable method to YOLOv3 as they tend to have similar mean Average Precision (mAP) scores. Their primary difference is inference speed in which YOLOv3 tends to beat SSD. By generalizing our results across, these two models we can establish the baseline validity of our results across all single SSD models. All these deep learning models run in real time. However, in terms of pedestrian detection accuracy, YOLOv3 shows a better detection accuracy (81% at 20 fps) [5]. Deep learning excels in the domain of object and image classification [23]. In the area

of deep learning, Convolution Neural Networks (CNNs) excel in image classification tasks [24]. The state-of-the-art CNN-based classification models include: Visual Geometry Group (VGG) [25] and InceptionV3 [26]. Visual Geometry Group (VGG-16) is a 16-layer convolutional neural network that is known for its high classification accuracy on a small number of classes and its real-time performance. Inception V3 is also known for high detection accuracy and is built with convolution, average pooling, max pooling, concatenation, and fully connected layers.

4 Dynamic error-bounded lossy video compression strategy

Compressing a video with a low-quality level greatly improves the compression ratio and reduces the bandwidth requirement to transfer the video but causes visual artifacts in the video. However, as the quality level of the video decreases, its ability to be used for video analytics decreases as well as features become less pronounced. For pedestrian detection, this results in lower detection accuracy. Furthermore, environmental conditions (e.g., rain, night-time darkness, fog) alter the compression ratio and makes pedestrian detection more difficult by obscuring pedestrians. Dynamically adapting the video compression quality level based on the current environmental condition ensures that we always detect pedestrians with high accuracy throughout the day and the year.

Figure 2 presents our framework for our dynamic EBLC strategy that uses machine learning to detect pedestrians. This paper develops a dynamic feedback control system that adapts the compression level to maintain the same detection accuracy of a system communicating the raw lossless video data. Figure 1 (see Sect. 1) presents a real-world deployment of our dynamic EBLC strategy. In our system, a roadside video monitoring camera collects video data and transfers it to an attached video compression unit [27]. The video compression unit compresses the raw video stream using a set tolerance level. In our experiments, we set the tolerance based on the PSNR ratio between the raw video and the resulting compressed video. The exact PSNR value depends on the environmental conditions (e.g., rain and night-time darkness). We use H.264 for video compression but note that other video compression algorithms work with our dynamic EBLC strategy. After compression, the compressed video streams are sent wirelessly to the roadside edge computing infrastructure. This edge computing infrastructure contains three main components: (i) a set of pre-trained and calibrated pedestrian detection models for different environmental conditions; (ii) the active pedestrian detection model to process video image; and (iii) an environmental condition detection model to identify the current environment for a given video.

This paper focuses on the development of a dynamic EBLC strategy that is independent of the vision-based pedestrian detection method. Given an environmental condition, the edge computing infrastructure selects an appropriate model from the set of pre-trained and calibrated models. In addition, it determines the corresponding PSNR for the model that yields the largest reductions in bandwidth while still maintaining the same detection accuracy. The selected PSNR value is periodically sent to the roadside video monitoring camera for use when compressing the video stream.

Each time the video compression unit located near the video camera receives a new PSNR value from the roadside edge computing device, the compression unit dynamically adapts its compression level. At the same time, the edge computing device selects the calibrated machine learning model for the current environmental condition and PSNR

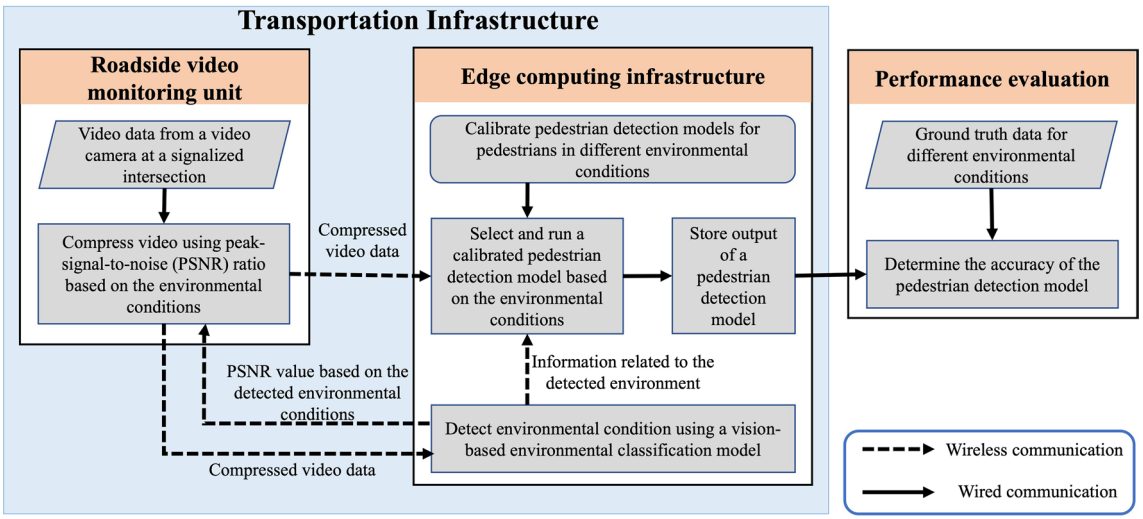

Fig. 2 Dynamic error-bounded lossy video compression strategy

value. In our design, a pedestrian detection model is trained with video images with different PSNR values for a specific environmental condition. For example, we select three levels of rain: (1) light rain; (2) moderate rain; and (3) heavy rain. For each level of rain, we initially compress the video to six different PSNR levels measured in decibels (dB): (10) 56 dB; (20) 49 dB; (30) 43 dB; (40) 37 dB; (50) 31 dB; and (51) 30 dB. The value in the parentheses shows the Constant Rate Factor (CRF) corresponding to each PSNR value. A smaller CRF results in less error during compression. The CRF is the error control knob we tune for the compressors inside FFmpeg [28], a software tool used to process audio and video files.

To determine the optimum CRF and corresponding PSNR that maintains a high pedestrian detection accuracy, a reference lookup table is constructed offline. The reference table contains only the models that have a pedestrian detection accuracy equal to that of the baseline model. To construct the reference lookup table and the catalog of corresponding models, we train and evaluate a model on data compressed with a CRF of 10 (highly accurate) along with computing the PSNR. Next, we increase the CRF by 10 (degrading video quality and improving compression) until the new model's detection accuracy drops below the minimum threshold. At this point, we vary the CRF by 1 to fully explore the range between the last valid CRF and the first invalid CRF. Again, we evaluate each model to determine if it meets our quality-of-service standards; rejecting any models that do not. After exploring each CRF in the interval, we have a lookup table that allows us to select a trained model given a requested CRF or PSNR value.

We calculate the accuracy of the pedestrian detection model by comparing it with manually annotated ground truth data. To establish a baseline accuracy, we perform pedestrian detection on the uncompressed video feed coming from traffic cameras for all scenarios and calculate the accuracy based on a manually annotated ground truth. For a compression baseline, we compress the video stream to a fixed quality level using standard image difference metric, PSNR, and use a pedestrian detection model with weights calibrated for the compressed data.

In this compression framework, there are three steps: (i) lossy video compression; (ii) calibration of the pedestrian detection model; and (iii) environmental condition detection using an environment classification model. The following subsections describe, in detail, our approach for each step in our dynamic EBLC strategy.

4.1 Error-bounded Lossy compression (EBLC)

Using field-collected data, we compress each video using different CRF values using the FFmpeg video compression tool [28]. The video compression level is controlled by the CRF value, and the CRF range is from 0 to 51, where 0 indicates no compression (no loss in data accuracy), and 51 indicates the maximum compression level (high degree of data inaccuracies). After that, we calculate the PSNR by comparing the original video's frames and the compressed video file. Thus, we use the CRF of FFmpeg to compress videos yielding different compression ratios (i.e., small for CRFs near 0 and large for CRFs near 51). However, to make our results independent from the FFmpeg tool, we determine the PSNR value corresponding to each CRF value. Figure 3 presents the feedback-based EBLC algorithm, which compresses the video feed based on the environmental condition. The compression tool compresses the video to a compression level, which maintains a high pedestrian detection accuracy. After resizing the image of the compressed video, a detection model is selected from a library of calibrated pedestrian detection models that account for various environmental conditions.

4.2 Pedestrian detection model calibration

The YOLOv3 model [24] divides an image into multiple regions and assigns probabilities to the bounding boxes for each region, where a feature is detected. This model can capture the global context of the image as it looks at the whole image simultaneously. The YOLOv3 model consists of 53 convolutional layers followed by 2 fully connected layers and 1×1 reduction layer followed by 3×3 convolutional layers [24]. The YOLOv3 model can have different input image sizes, such as 320×320×3, 416×416×3 and 608×608×3. Based on our experiments, we found that the input image size of 416×416×3 provides the highest pedestrian detection accuracy with a low computational

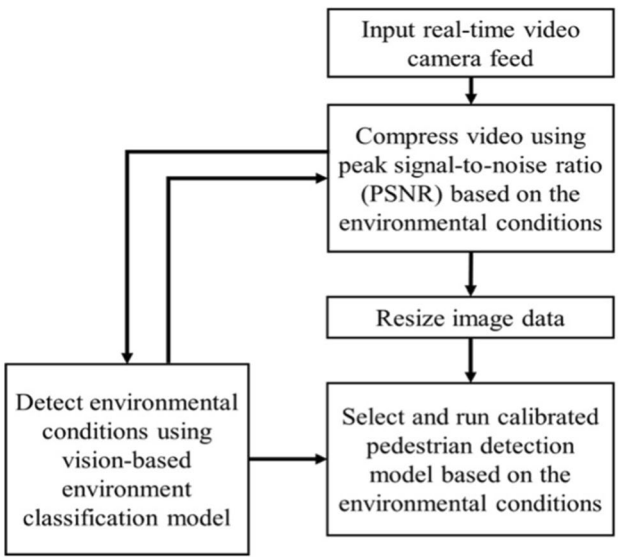

Fig. 3 Feedback-based real-time EBLC algorithm

cost. In this study, we use the input image size of 416×416×3 and then normalized the image at the preprocessing layer of the YOLOv3 model. We also use a SSD model for pedestrian detection and compare its performence in terms of accuracy with YOLOv3 model. The backbone of our SSD model ishethe ResNet-50 classification model, whichis a convolutional neural network with 50 layers. An image size of 416×416×3 was also inputted to the SSD model to achieve comparable experimental conditions to the YOLOv3 model.

To achieve a much higher pedestrian detection accuracy for different environmental conditions (e.g., rain and lighting), we train the YOLOv3 and SSD ResNet-50 model on augmented data. We perform data augmentation for different rain and lighting conditions to produce more realistic images for night-time darkness and rain. To generate augmented data for the model calibration, we alter the night-time darkness of the images by changing the pixel values of the first channel in the HSL (hue, saturation, lightness) color space of an image. Based on the rain intensity, different types of rainy environments are created by adding random small lines on the image and making the image a little blurry to replicate a realistic rainy environment [29].

To train the models, we down sample the video at 10 frames per second (fps) to extract frames for pedestrian safety applications [30]. After that, we have used the standard Pascal Visual Object Class (VOC) format to annotate each extracted frame from the video file. Each pedestrian detection model splits an image into multiple regions and calculates the probabilities for each region of being a pedestrian. Based on the calculated probabilities, a detection model generates bounding boxes for pedestrians. The YOLOv3 and SSD ResNet-50 models can generate multiple bounding boxes for a single pedestrian, which reduces pedestrian detection accuracy significantly. We have used a non-max suppression method [31] to improve the pedestrian detection accuracy by keeping one bounding box and excluding other unnecessary bounding boxes detecting each pedestrian. This algorithm takes the bounding boxes for a pedestrian and selects the one with the highest confidence score. The intersection over union (IOU) of this box is calculated with each of the other bounding boxes for the pedestrian. If this score is higher than the threshold IOU set, then it is thrown out as there is a substantial overlap of the predictions.

The primary hyperparameters to tune for these models are the learning rate, image input size, batch size, and epochs of the network will train. The learning rate hyperparameter is essential to tune such that you obtain an optimal set of weights in a sufficient amount of time. A larger learning rate will usually result in faster learning but at the cost of a group of suboptimal weights. When tuned too large, the performance of a model may oscillate over the training period, which is caused by a set of diverging weights. If the learning rate is too low, the model may never converge to a set of weights. The image input size parameter can be tuned to improve the performance of the model. In general, larger images perform better as it is easier for models to detect larger objects. The batch size parameter adjusts how many samples the train on before the model updates its internal parameters. The epoch parameter is the number of passes the model will make through the training data set, while the model is learning. It is essential to balance the learning rate with batch size and the number of epochs such that the model doesn't overfit to its training data set. By tuning these parameters in our models, we were able to see the performance increases in our models.

4.3 Environmental condition detection

In this paper, to detect and classify different environmental conditions, we use a vision-based Convolution Neural Network (CNN) deep learning model. The classifier takes an image as input and classifies it among seven different environmental conditions: normal weather, light dark, medium dark, high dark, light rain, moderate rain, and heavy rain (as shown in Table 1). The model's input image is 416×416×3 pixels, and the output is a 7×1 matrix, W. For example, as shown in Fig. 4, an output, $W=[0, 0, 1, 0, 0, 0, 0]^T$ indicates

Table 1 Selected environmental conditions and corresponding video compression scenarios

Evaluation scenarios	Environmental condition	Category of environmental condition	Constant rate factor (CRF) range	Minimum average PSNR value corresponding to CRF value in column 4
1	Normal	Sunny weather	0–10, 11–20, 21–30, and 31–33	56 dB (corresponding to CRF 10), 49 dB (corresponding to CRF 20), 43 dB (corresponding to CRF 30), and 41 dB (corresponding to CRF 33)
2	Lighting condition	Light dark		
3		Medium dark		
4		High dark		
5	Rainy condition	Light rain		
6		Moderate rain		
7		Heavy rain		

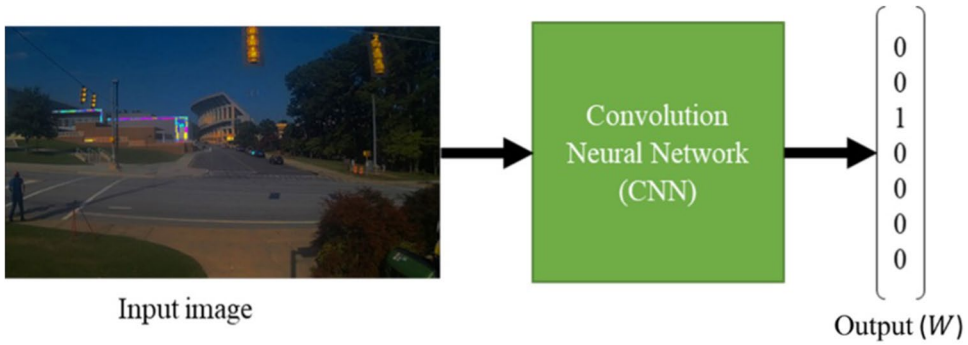

Fig. 4 Environmental condition classifier using convolution neural network (CNN)

a medium dark weather condition. Being a simple CNN-based classifier, the model is able to run on a roadside video image processing unit with less capable computation resources.

5 Analysis and results

In this section, we describe the environmental and lossy video compression scenarios, data generation and deep learning model calibration for different environmental conditions (see Table 1). In addition, we report the pedestrian detection accuracy for each condition.

5.1 Environmental and lossy video compression Scenarios

In this study, we consider three different environmental conditions: (i) normal (sunny weather) condition; (ii) nighttime darkness; and (iii) rain. For the lighting and rainy conditions, we further break these down into three additional categories. The categories for the lighting condition are light, medium and high, and the categories for the rainy condition are light, moderate and heavy. Prior work finds that the pedestrian detection accuracy in sunny weather decreases from the no compression baseline condition if the CRF value is greater than 30 (PSNR 43 dB) [5]. Thus, for each category of environmental condition, we present four compression scenarios: (a) CRF = 10 (PSNR = 56 dB); (b) CRF = 20 (PSNR = 49 dB); (c) CRF = 30 (PSNR = 43 dB); (d) CRF = 33 (PSNR = 41 dB). However, in a real-world deployment, more compression scenarios would be used. After collecting video data for the normal weather condition, we generate data for the different environmental conditions and compression scenarios to evaluate pedestrian detection accuracy. For each scenario, we calculate the pedestrian detection accuracy to determine the maximum compression ratio at which we maintain the baseline pedestrian detection accuracy.

5.2 Data generation and description

To obtain data for our baseline normal sunny weather condition (no data compression), we collect field data from the Perimeter Road and Avenue of Champions intersection at Clemson, South Carolina on January 4th, 2019 at 12 PM. We use a camera (i.e., Logitech C920 WEBCAM HD) on a data collection pole and record video of the intersection including pedestrians on the crosswalk. This data set contains a total of 427 images, where pedestrians are moving in four directions, such as north–south, south–north, east–west and west–east. After collection of the field data, we perform data augmentation to generate images for different environmental conditions. Using data augmentation as described in the Sect. 4.2 of Sect. 4, we create seven environmental conditions, as shown in Fig. 5. Thus, for each environmental condition, we generate 427 images based on the data collected from the field. This data set is publicly available at https://drive.google.com/open?id=1XA0hOfjvIb1l29rvkbUnwjN6kMj12KaD.

5.3 Pedestrian detection model training and evaluation

To improve the pedestrian detection accuracy using the YOLOv3 and ResNet-50 models, we use a pre-trained version of each model and retrained the model on our collected and generated data set for different environmental conditions and different compression levels. For the normal sunny weather's 427 images, we split our data set further into train, test, and validation sets with the following percentages 63%, 20%, and 17%, respectively. In total, we evaluate 28 unique configurations (7 environmental configuration and 4 compression levels). After data augmentation, we generate a total of 11,956 images, which includes 7,532 images for training, 2391 images for testing, and 2033 images for validation. Using these data sets, we retrain the models. After training, to

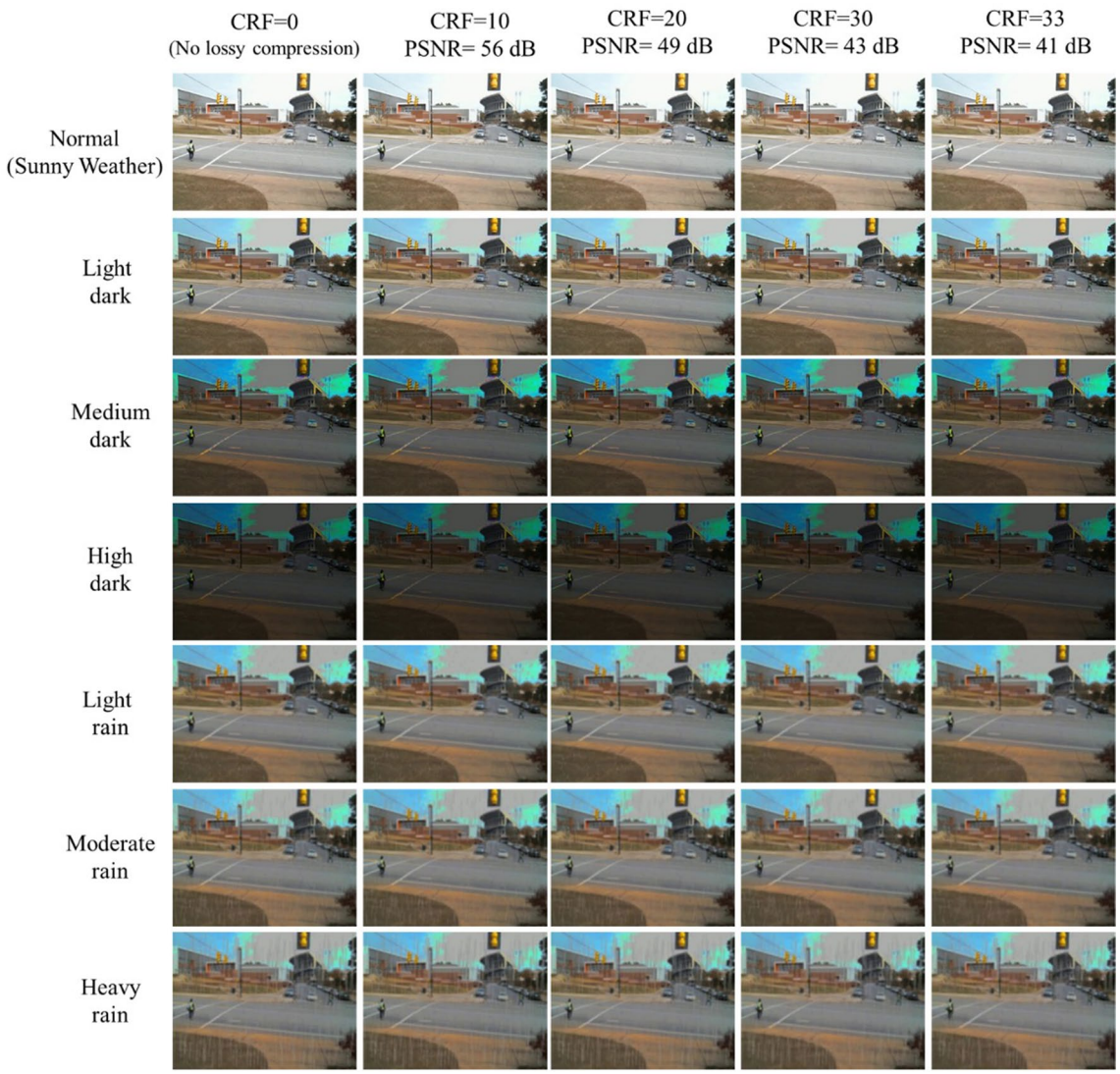

Fig. 5 Video compression with different environmental weather conditions

further improve the pedestrian detection accuracy, we use a non-max suppression method with an IOU value of 0.6 [5] to suppress false positives in the pedestrian detection output.

5.4 Environmental condition detection

To detect different environmental conditions, we use a CNN-based deep learning model. In particular, we use the VGG-16 model [27] as the base network, and we train on our own data set. We normalize and resize each input image from $416 \times 416 \times 3$ to a size of $224 \times 224 \times 3$ to match with the VGG-16 model input layer size. The convolution network with linear rectified units (ReLU), max pooling, and a fully connected layer with ReLU acts as an image encoder to extract the image features of various weather conditions. As shown in Fig. 6, this classification model classifies the image into one of the seven classes: normal-sunny weather, light dark, medium dark, high dark, light rain, moderate rain, and heavy rain. The model is trained on the augmented data sets for these seven environmental conditions. Similarly, we split the data sets into 63% for training, 20% for testing, and 17% for validation for each weather condition. Based on the testing data set, our CNN-based model is able to classify the weather condition with 97% accuracy.

5.5 Evaluation of dynamic EBLC framework

Our EBLC communication scheme is able to leverage multiple machine learning models to detect pedestrians. Figure 7 shows the pedestrian detection accuracy for the two models we consider (YOLOv3 and ResNet-50) in sunny weather with various

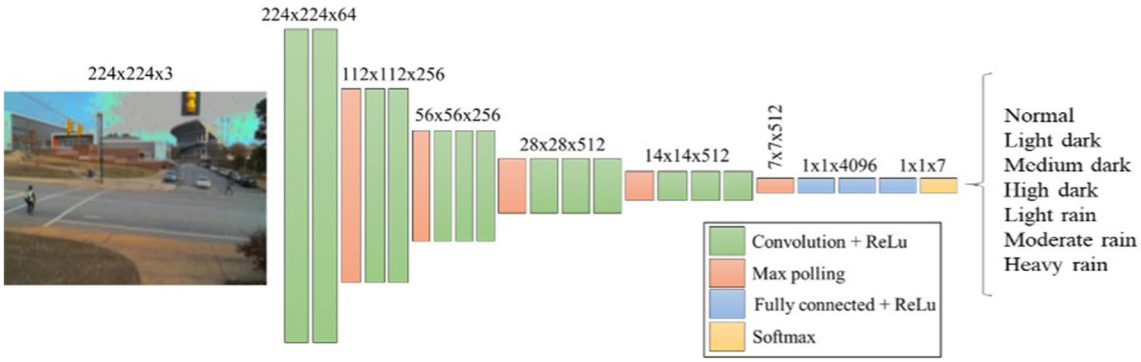

Fig. 6 CNN-based environmental condition classifier

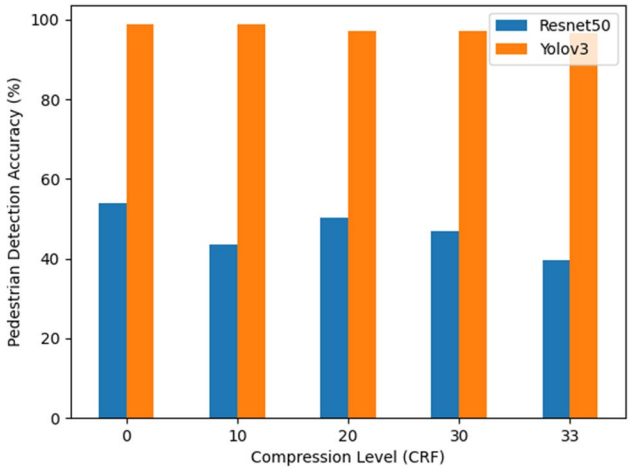

Fig. 7 Pedestrian detection accuracy for different compression levels with two pedestrian detection models

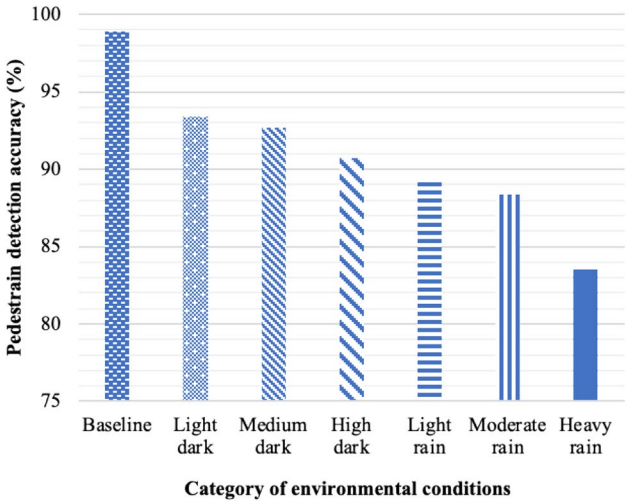

Fig. 8 Pedestrian detection accuracy for different environmental conditions with no data compression and using a model trained for baseline data

levels of compression. From the figure, we observe that as the compression level increases (higher CRF) the images become more distorted (lower PSNR) and the ability to accurately detect pedestrians decreases. YOLOv3 sees a 2% drop in accuracy over the compression levels, and ResNet-50 sees a 14.36% drop in accuracy over the compression levels. Although, the two models exhibit similar trends in accuracy reduction as compression level increases, the poor baseline accuracy of ResNet-50 (54.01%) is unacceptable for safety–critical applications. Thus, e focus on YOLOv3 model for the reminder of our evaluation. Note that the SSD ResNet-50 model demonstrates similar trends in terms of accuracy for different compression scenarios; however, the pedestrian detection accuracy of SSD ResNet-50 is always much less compared to the YOLOv3 model accuracy found in this research .

To investigate the impact of different environmental conditions on pedestrian detection accuracy, we evaluate the accuracy of the YOLOv3 model for pedestrian detection in different weather conditions by training only on sunny weather data. Figure 8 presents pedestrian detection accuracy for different environmental conditions with no data compression. We find that the pedestrian detection accuracy continues to reduce as the weather condition continues to deteriorate. Pedestrian detection accuracy reduces even more if we compress the image before pedestrian detection during adverse weather conditions. Thus, the accuracy of the pedestrian detection model varies based on the environmental condition and the level of lossy compression. Therefore, it is important to train the pedestrian detection model with data for different environmental conditions and on the level of lossy compression.

We evaluate the pedestrian detection accuracy for different environmental conditions with different CRF values ranging from 0 to 33. We limit our CRF value to 33, as CRF values above 33 (41 dB) yields unacceptable deterioration in the pedestrian detection accuracy. Figure 9 shows that using our dynamic EBLC framework that leverages models trained

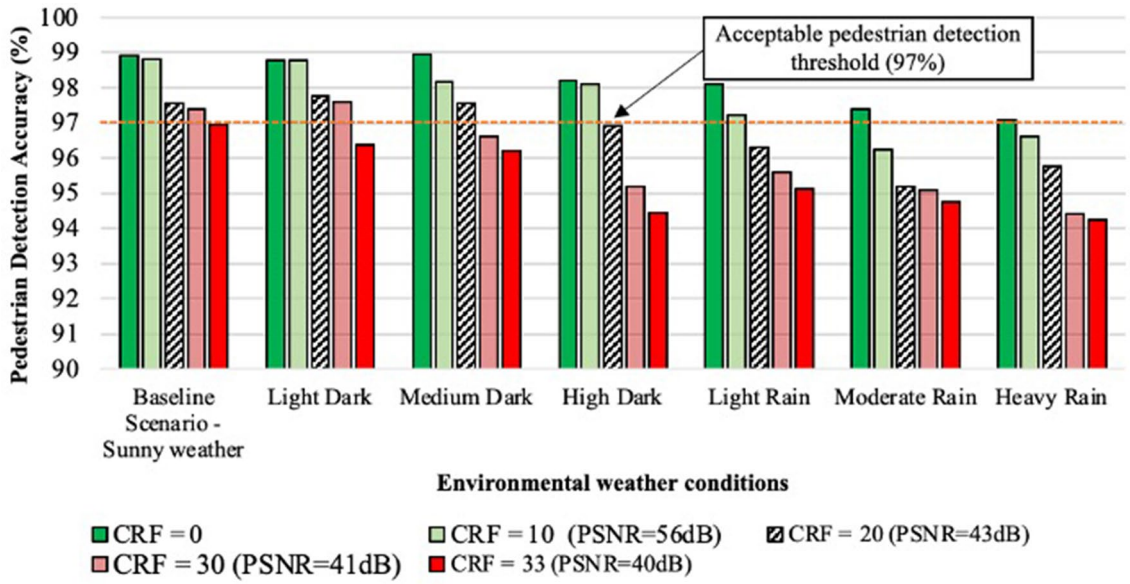

Fig. 9 Pedestrian detection accuracy for different weather conditions with different compression levels

for each environmental and compression level, we improve pedestrian detection accuracy for all adverse environmental conditions. In a heavy rain condition, we find a 14% improvement compared to the baseline condition, i.e., no lossy compression (see Table 2). As the weather condition becomes more adverse, the pedestrian detection accuracy goes down. Similarly, the detection accuracy decreases as the CRF values increase, meaning that the pedestrian detection accuracy decreases as the compression ratio increases. From Fig. 9, we determine the minimum video quality level that still maintains a fixed pedestrian detection threshold. In our case, we consider the pedestrian detection baseline accuracy as 97%, which is the lowest accuracy we found for all environmental scenarios without any data compression (CRF = 0). Therefore, we are able to apply compression in all environmental conditions except in moderate and heavy rain. Future work will explore improving the detection accuracy when using high levels of compression.

Table 2 shows the maximum CRF or minimum PSNR based on the different environmental conditions. The original communication bandwidth without any compression is 9.82 MBits/sec. The maximum CRF ranges from 0 to 30, while the minimum PSNR ranges from 41 to 56 dB. Depending on the PSNR value, we reduce bandwidth by 1.5× to 18× in our case study. However, our previous study [5] showed that using EBLC and a static error tolerance reduces bandwidth requirements for pedestrian detection by over 30× with no deterioration in detection accuracy. As stated in the literature review section of the paper, the static tolerance does not work well on adverse weather conditions, such as in cloudy or rainy conditions. Because we use a different data set and we compress data to a fixed-accuracy (i.e., target PSNR) in

Table 2 Maximum compression ratio achieved for different weather conditions

Weather condition	Baseline Pedestrian detection accuracy (no compression) (%)	Dynamic EBLC framework Pedestrian detection accuracy (%)	Improvement in Pedestrian detection using dynamic EBLC framework (%)	Maximum constant rate factor (CRF) (or minimum PSNR)	Required bandwidth (MBits/sec)	Bandwidth reduction
Normal	97	97	−1	30 (41 dB)	0.53	18×
Light dark	93	97	4	30 (41 dB)	0.68	14×
Medium dark	92	97	5	20 (43 dB)	1.01	9.5×
High dark	90	97	7	10 (56 dB)	4.05	2.5×
Light rain	89	97	8	10 (56 dB)	5.15	1.5×
Medium rain	88	97	9	0	9.82	0×
Heavy rain	83	97	14	0	9.82	0×

the current study, it allows for variations in the compression ratio to meet a fixed-accuracy requirement. Thus, we do not achieve the same bandwidth reductions.

5.6 Performance modeling of our dynamic EBLC framework

The goal of our dynamic EBLC scheme is to reduce the bandwidth requirements and, therefore, overall time to transfer video data needed for pedestrian detection. Data compression trades computational time for reductions in data size, and in turn, reduces the bandwidth requirements when transmitting the data. However, if too much time is taken to compress the data, then the total time of compression and transmission can exceed the time to send the original uncompressed data. To quantify this tradeoff, we construct communication performance models. Let N be the number of bytes in the original uncompressed message (i.e., a segment of video data) and B be the communication bandwidth capacity (bytes/second), then we define the time to transmit the message as

$$T_{\text{send\_orig}} = \frac{N}{B}$$

When using data compression, we must add an additional term, C, to account for the speed at which we compress the data (also known as the compression bandwidth) in units of bytes/seconds and the compressed data size in bytes N'. Thus, we define the time to send compress data as the sum of the time to compress and transmit the compressed data:

$$T_{\text{send\_EBLC}} = \frac{N}{C} + \frac{N'}{B}$$

Communication of the video data to a roadside edge computing transportation infrastructure, which includes edge computing unit and safety alert broadcasting unit, only accounts for part of the workflow of the pedestrian safety alert system. Upon receiving the data at the roadside edge computing transportation infrastructure from the roadside video monitoring unit, it is fed into the edge computing unit, which runs a machine learning model. The model determines if pedestrians are present and generate metadata, such as location and speed of pedestrians at a cost of T_{process}. After that, the edge computing unit transmits the acknowledgment of pedestrian detection to the safety alert broadcasting unit of the roadside edge computing transportation infrastructure at a cost of T_{ack}. Once the safety alert broadcasting unit receives the acknowledgment of pedestrian detection, it broadcasts a safety alert to the connected devices (e.g., connected vehicles or roadside changeable message sign) at a cost of T_{bcast}. Thus, the total time for the original workflow is

$$T_{\text{orig}} = T_{\text{send\_orig}} + T_{\text{process}} + T_{\text{ack}} + T_{\text{bcast}}$$

Similarly, the total time for the workflow that uses EBLC is

$$T_{\text{EBLC}} = T_{\text{send\_EBLC}} + T_{\text{process}} + T_{\text{ack}} + T_{\text{bcast}}$$

As we modify the compression level, the compression bandwidth changes. In general, allowing more distortions into the data results in larger compression ratios and larger compression bandwidths as it takes more time (lower compression rate, i.e., lower compression bandwidth) to compress data with little to no loss. Moreover, the selection of computational hardware impacts compression bandwidth. Therefore, to abstract our results for future faster hardware and other compression algorithms, with differing compression bandwidths (both higher and lower than the achieved 14.3 MB/s in our experiments), we evaluate our models on a range of different compression bandwidth values. For the communication bandwidth capacity, we measure the "eduroam" network on Clemson's campus when transmitting a 1 GB data file wirelessly from the roadside video monitoring unit to the roadside edge computing transportation infrastructure and obtain a communication bandwidth of 5.1 MB/s. To determine under what conditions our EBLC improves performance, we compute the speedup in communication time for sending the video data to the data processing infrastructure as

$$\text{Speedup} = \frac{T_{\text{send\_orig}}}{T_{\text{send\_EBLC}}}$$

Thus, the speedup represents the factor by which we improve the communication time if the speedup is greater than 1 or the factor by which we degrade performance if the speedup is less than 1. Figure 10 shows the speedup of the communication time when using EBLC for 3 scenarios. Scenario 1 (Fig. 10a) uses a communication bandwidth value of 0.51 MB/s and represents rural locations with a low bandwidth capacity or high traffic volume locations, where available bandwidth to a single user is limited. Scenario 2 (Fig. 10b) uses the measured communication bandwidth of 5.1 MB/s and represents a regular traffic volume scenario. Scenario 3 (Fig. 10c) uses a communication of 51 MB/s and represents a non-bandwidth constrained environment. We observe in each scenario that speedup (shades of red color) is possible but depends on the compression and communication bandwidth along with the compression ratio. When the communication bandwidth is low, less is required from the compressor in terms of speed and data reduction to see benefits (i.e., more configurations are colored in shades of red indicated EBLC communication is faster than the original communication). However, as the achieved communication bandwidth improves

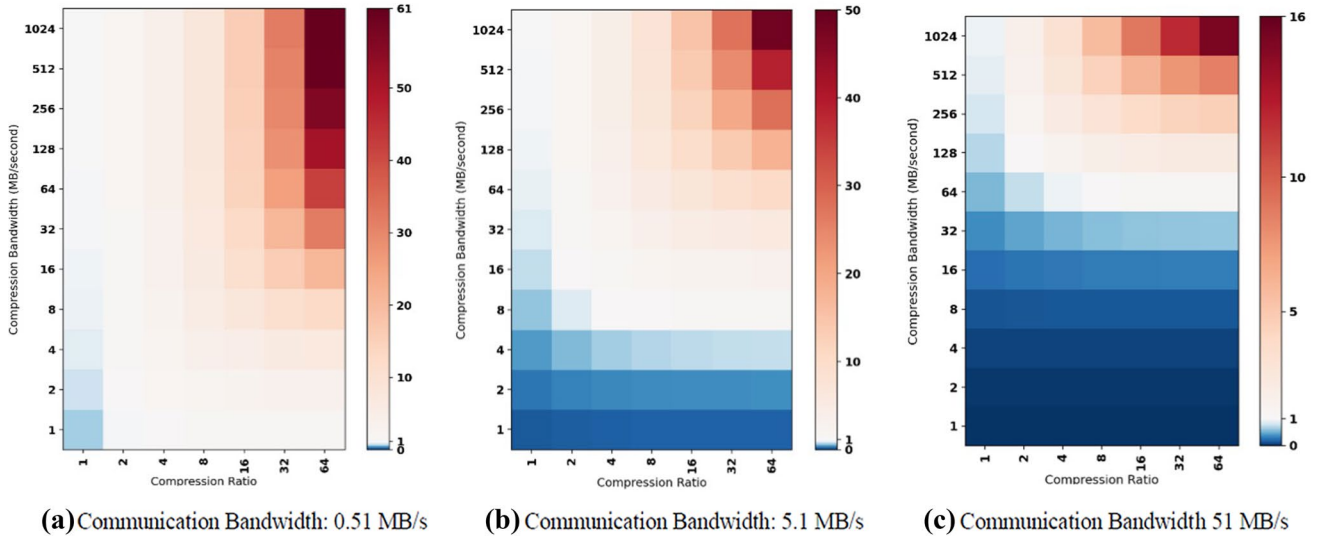

Fig. 10 Speedup for time to communicate video using our EBLC scheme for various communication bandwidth values. Note: The white color indicates no speedup. Shades of red color indicate EBLC communication is faster. Shades of blue color indicate EBLC communication is slower

as shown in Fig. 10b, c, the compression algorithm needs either larger compression bandwidths or larger compression ratios to achieve the speedup (i.e., most of the configurations are shades of blue indicated EBLC communication is slower than the original method of communication). If paired with an appropriate compressor, then large speedups are possible in all three scenarios. During our experiments, we find a maximum compression ratio of 18× and a compression bandwidth of 14.3 MB/s leading to a speedup of 2.4×.

6 Conclusions

Dynamically adapting the video compression quality level based on environmental conditions ensures the reduction of the communication bandwidth requirement for transferring a video wirelessly while detecting pedestrians with high accuracy. The contribution of this study is developing a feedback-based real-time dynamic EBLC strategy considering different environmental conditions by reducing the communication bandwidth while maintaining a baseline (i.e., no compression and sunny weather) pedestrian detection accuracy. Depending on different environmental factors, our strategy dynamically selects the error tolerance for error-bounded lossy compression that yields the best performance. Through our dynamic EBLC strategy, we maintain a high pedestrian detection accuracy using the YOLOv3 detection model across a selection of the different environmental levels of rain and night-time darkness. Our EBLC strategy is independent of the pedestrian detection model, and any type of pedestrian detection model can be used in our framework. Our analysis reveals that in adverse environmental conditions, the dynamic EBCL strategy can reduce the bandwidth requirements for transmitting video over prior approaches up to 14× while maintaining the baseline accuracy that transmits lossless videos. Results show that if the weather condition is adverse, the bandwidth reduction is lower. Even for moderate and heavy rainy conditions, we could not compress video at all if we are required to maintain a 97% pedestrian detection accuracy. In our future study, we will consider unexplored trade-offs, such as the energy efficiency of our ELBC strategy and how to utilize multiple intra-frame compression tolerances to further improve the compression ratio to maximize the bandwidth usage.

Acknowledgements This material is based on a study partially supported by the Center for Connected Multimodal Mobility (C^2M^2) (USDOT Tier 1 University Transportation Center) Grant headquartered at Clemson University, Clemson, South Carolina, USA. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Center for Connected Multimodal Mobility (C^2M^2), and the U.S. Government assumes no liability for the contents or use thereof. This material is also based upon work supported by the National Science Foundation under Grant No. SHF-1910197.

Author contributions The authors confirm contribution to the paper as follows: MR: conceptualization; methodology; data curation; formal analysis; and roles/writing—original draft. MI: data curation; formal analysis; and writing—original draft preparation. CH: formal analysis and writing—original draft preparation. JC: conceptualization, funding acquisition; writing—review and editing. MC: conceptualization, methodology, funding acquisition; writing—review and editing.

Funding This material is based on a study partially supported by the Center for Connected Multimodal Mobility (C^2M^2) (USDOT Tier 1 University Transportation Center) Grant headquartered at Clemson University, Clemson, South Carolina, USA. This material is also based

upon work supported by the National Science Foundation under Grant No. SHF-1910197.

Availability of data and materials Not applicable.

Code availability Not applicable.

Declarations

Conflicts of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Pedestrian Safety. https://www.nhtsa.gov/road-safety/pedestrian-safety. Accessed 9 Sep 2020.
2. Sewalkar, P., Seitz, J.: Vehicle-to-pedestrian communication for vulnerable road users: survey, design considerations, and challenges. Sensors. **19**(2), 358 (2019)
3. Gerónimo, D., López, A.M.: Vision-based pedestrian protection systems for intelligent vehicles. Springer, New York (2014)
4. Rosenbaum, D., Gurman, A., and Stein, G.: Forward collision warning trap and pedestrian advanced warning system. US Patent 9,251,708. Mobileye Vision Technologies Ltd (2016)
5. Rahman, M., Islam, M., Calhoun, J., Chowdhury, M.: Real-time pedestrian detection approach with an efficient data communication bandwidth strategy. Transp. Res. Rec. (2019). https://doi.org/10.1177/0361198119843255
6. Islam, M., Rahman, M., Chowdhury, M., Comert, G., Sood, E.D., Apon, A.: Vision-based personal safety messages (PSMs) generation for connected vehicles. IEEE Trans. Veh. Technol. (2020). https://doi.org/10.1109/TVT.2020.2982189,2020
7. Ohm, J.R., Sullivan, G.J., Schwarz, H., Tan, T.K., Wiegand, T.: Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC). IEEE Trans. Circuits Syst. Video Technol. **22**, 1669–1684 (2012)
8. Shizhong, L., Bovik, A.C.: Efficient DCT-domain blind measurement and reduction of blocking artifacts. IEEE Trans. Circuits Syst. Video Technol. (2002). https://doi.org/10.1109/TCSVT.2002.806819
9. Wang, Z., Bovik, A.C.: A universal image quality index. IEEE Signal Process. Lett. (2002). https://doi.org/10.1109/97.995823
10. De Cock, J., Li, Z., Manohara, M., Aaron, A.: Complexity-based consistent-quality encoding in the cloud. 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, (2016). https://doi.org/10.1109/ICIP.2016.7532605
11. Ziv, J., Lempel, A.: A universal algorithm for sequential data compression. IEEE Trans. Inf. Theory 23(3), 337–343 (1977). https://doi.org/10.1109/TIT.1977.1055714
12. Zemliachenko, A., Lukin, V., Ponomarenko, N., Egiazarian, K., Astola, J.: Still image/video frame lossy compression providing a desired visual quality. Multidimens. Syst. Signal Process. (2016). https://doi.org/10.1007/s11045-015-0333-8
13. Sayood, K.: Introduction to data compression. Morgan Kaufmann (2017). ISBN 978-0128094747.
14. Di, S., Cappello, F.: Fast error-bounded lossy HPC data compression with SZ. 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Chicago, IL, (2016). https://doi.org/10.1109/IPDPS.2016
15. 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), (2017)
16. ITU-T and ISO/IEC JTC 1. High efficiency video coding. ITU-T Rec. H.265 and ISO/IEC 23008–2, (2018)
17. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. IEEE Trans. Comput. (1974). https://doi.org/10.1109/T-C.1974.223784
18. Galteri, L., Bertini, M., Seidenari, L., Del Bimbo, A.: Video Compression for Object Detection Algorithms. 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, (2018). https://doi.org/10.1109/ICPR.2018.8546064.
19. Kong, L., Dai, R.: Object-detection-based video compression for wireless surveillance systems. IEEE Multimed. (2017). https://doi.org/10.1109/MMUL.2017.29
20. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2014). https://doi.org/10.1109/CVPR.2014.81
21. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision, (2015). https://doi.org/10.1109/ICCV.2015.169
22. Hanna, E., Cardillo, M.: Faster R-CNN: towards real-time object detection with region proposal networks. Biol. Cons. (2013). https://doi.org/10.1016/j.biocon.2012.08.014
23. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C.: SSD: single shot multibox detector. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), (2016). https://doi.org/10.1007/978-3-319-46448-0_2
24. Redmon, J., Farhadi, A., Ap, C.: YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767, (2018).
25. Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput. **29**(9), 2352–2449 (2017)
26. Bengio, Y.: Learning deep architectures for AI. Found Trends Mach Learn. **2**(1), 1–127 (2009)
27. Husemann, R., Susin, A.A., Roesler, V.: Optimized solution to accelerate in hardware an intra H. 264/SVC video encoder. IEEE Micro **38**(6), 8–17 (2018)
28. FFmepg Developers. ffmpeg tool (Version N-82324-g872b358) (2018).http://ffmpeg.org
29. Automold--Road-Augmentation-Library. (2019) https://mail.google.com/mail/u/0/#inbox/FMfcgxwDqThrFzXlSbjbfZjckWqNjLbZ
30. ARC-IT. Service Packages (2019). https://local.iteris.com/arc-it/html/servicepackages/servicepackages-areaspsort.html
31. Rothe, R., Guillaumin, M., Van Gool, L.: Non-maximum suppression for object detection by passing messages between windows. In: Asian Conference on Computer Vision. Springer, Cham, pp. 290–306 (2014)

Mizanur Rahman is an assistant professor in the Department of Civil, Construction and Environmental Engineering at the University of Alabama, Tuscaloosa, Alabama. He received his M.Sc. and Ph.D. degrees in Civil Engineering (Transportation systems), from Clemson University, in 2013 and 2018, respectively. His research focuses on traffic flow theory, and transportation cyber-physical systems for connected and automated vehicles and smart cities.

Mhafuzul Islam received his Ph.D. in 2021 from Clemson University. He received the BS degree in Computer Science and Engineering from the Bangladesh University of Engineering and Technology (BUET) in 2013 and the MS degree in Civil Engineering from Clemson University in 2018. His research interest includes transportation cyber-physical systems with an emphasis on data-driven connected autonomous vehicles.

Cavender Holt is a B.S. Student in the Holcombe Department of Electrical and Computer Engineering at Clemson University. His research interests lie in lossy and lossless data compression algorithms and their use in high-performance computing (HPC) and intelligent transportation systems.

Jon Calhoun is an assistant professor in the Holcombe Department of Electrical and Computer Engineering at Clemson University. He received his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 2017. His research interests lie in fault tolerance and resilience for high-performance computing (HPC) systems and applications, lossy and lossless data compression algorithms and their use in HPC and intelligent transportation systems, and power-aware computing.

Mashrur Chowdhury is the Eugene Douglas Mays Chair in Transportation in the Glenn Department of Civil Engineering at Clemson University. He is the director of the USDOT UTC Center for Connected Multimodal Mobility (C^2M^2) (http://cecas.clemson.edu/c2m2). He is the co-director of the Complex Systems, Analytics and Visualization Institute (CSAVI) (http://clemson-csavi.org) at Clemson University. He is a senior member of IEEE. He is a Fellow of the American Society of Civil Engineering (ASCE) and an alumnus of the National Academy of Engineering (NAE) Frontiers of Engineering program.