IEICE Communications Express, Vol.13, No.3, 76–79

ComEX

LETTER

Streaming quality control based on object-detection accuracy

Nobuaki Akutsu[1, a)], **Takuya Shindo**[1, b)], **Takefumi Hiraguri**[1, c)], **Hideaki Yoshino**[1, d)], **and Nobuhiko Itoh**[1, e)]

Abstract This study aims to establish an adaptive video quality control method that satisfies object-detection accuracy and reduces bandwidth consumption simultaneously for remote real-time video analysis systems. Existing video quality control methods determine video quality by considering human perceptual characteristics; this reduces bandwidth consumption while providing a high quality of experience. However, it cannot reduce bandwidth consumption in systems that use object-detection engines to detect people and vehicles. Thus, this study proposes a video quality control method to reduce bandwidth consumption. To this end, the bandwidth consumption and ratio of the number of frames satisfying the mean average precision requirements to the total number of frames (herein referred to as the success rate) are evaluated. The results confirm that the proposed method can reduce bandwidth consumption to 49% of that of the existing video quality control method at the same success rate.

Keywords: object detection and streaming quality
Classification: Network

1. Introduction

As image recognition technology and machine learning have advanced along with mobile networks, new services for the Internet of Things (IoT) are being developed with real-time recognition for remote locations. For example, using mobile networks, a connected car service can collect real-time information, such as images around an intersection and locations of vehicles from connected devices, such as roadside cameras and vehicles. Real-time information helps drivers avoid road-traffic collisions. A road-traffic collision avoidance scheme using roadside cameras was proposed by [1]. In the future, cameras located in public spaces can be used for vehicle traffic control. By reducing the video bitrate per camera while satisfying desired detection accuracy, various cameras can be implemented in the social infrastructure, contributing to the stable operation of remote-control services with video streaming.

Although research on upgrading the object-detection engine is being conducted worldwide to improve object-detection accuracy in applications involving Yolov3 [2], video quality control methods that optimize the video bitrate input to detection engines have not been sufficiently investigated.

Recently, H.265 constant rate factor (CRF) mode, which is one of the conventional video quality control methods, has been proposed [3] to address the abovementioned gap. However, the H.265 CRF mode causes fluctuations in the mean average precision (mAP) when Yolov3 is used as the object-detection engine. Therefore, the H.265 CRF mode has to continuously send high-quality videos at all times, considering the periods of decreasing mAP, but this increases the network load.

This study aims to overcome this challenge and proposes a video quality control method that reduces the network load caused by video transmission when Yolov3 is used to detect objects with high accuracy. The proposed video quality control method is favorable to object-detection engines and networks, and this study demonstrates its effectiveness.

The remainder of this manuscript is organized as follows. Section 2 presents H.265 CRF mode and its challenges. Section 3 presents proposed dynamic CRF control. Section 4 presents performance evaluation. Finally, Section 5 presents the conclusions drawn from the study findings.

2. H.265 CRF mode and its challenges

The H.265 CRF mode controls the video bitrate such that the quality of experience (QoE) [4] is constant. Therefore, the H.265 CRF mode is a superior video quality control method for humans. The lower the CRF parameter, the higher the video quality and bitrate.

The detection accuracy of the object-detection engine is represented by mAP, which is a measure of match between object-detection results and the correct data. To investigate the basic characteristics of the H.265 CRF mode, we evaluated mAP using the H.265 CRF mode as a video quality control method and Yolov3 as the object-detection engine. A 10-min streaming video of a real intersection shown in Fig. 1, was used as the evaluation video. The frame rate of the video was 10 frames per second.

Figure 2 shows the mAP characteristics of the H.265 CRF mode. The horizontal and vertical axes represent time and mAP, respectively. Green indicates a high-quality video (CRF = 2), and blue indicates a low-quality video (CRF = 30). mAP is commonly used as a measure of object-detection accuracy [5]. In this study, mAP is calculated using the common objects in context (COCO) dataset [6, 7]. As this study aims to support safe driving assistance, mAP is only calculated for objects that include a person, bicycle, motorcycle, car, truck, and bus in the COCO dataset. Therefore, the mAP on the vertical axis in Fig. 2 shows the

[1] Nippon Institute of Technology, 4–1 Gakuendai, Miyashiro-machi, Minamisaitama-gun, Saitama, 345–8501 Japan
[a)] 2228001@stu.nit.ac.jp
[b)] t-shindo@nit.ac.jp
[c)] hira@nit.ac.jp
[d)] yoshino@nit.ac.jp
[e)] itoh@nit.ac.jp

DOI: 10.23919/comex.2023XBL0158
Received November 20, 2023
Accepted December 15, 2023
Publicized January 15, 2024
Copyedited March 1, 2024

Fig. 1 Snapshot of evaluation video.

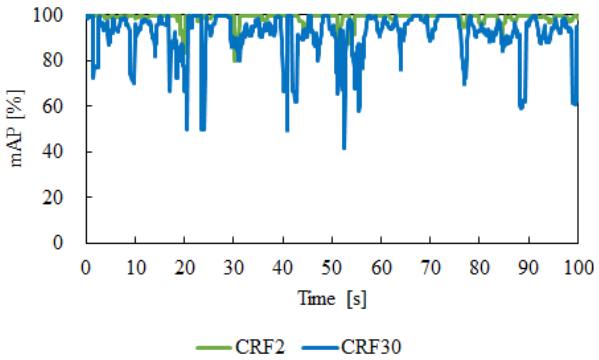

Fig. 2 mAP of the H.265 CRF mode.

detection-accuracy results for the objects per frame.

Figure 2 shows that the temporal variation in mAP can be suppressed when a high-quality video is used, whereas the temporal variation in mAP is large when a low-quality video is used. Therefore, to maintain a high mAP at all times, a high-quality CRF should be set, which requires a large video-transmission bandwidth and places a huge load on the network. However, there are several segments in which mAP was equivalent to that of a high-quality video, even for low-quality videos. Examples include the intervals from 26 to 29 s and from 71 to 76 s. Therefore, it is not necessary to transmit high-quality videos at such intervals, and transmitting low-quality videos reduces network load while maintaining mAP.

3. Proposed dynamic CRF control

As indicated in Section 2, the H.265 CRF mode requires transmitting high-quality streaming video to maintain a high mAP. At certain intervals, mAP equivalent to that of high-quality video can be obtained, even if a low-quality video is transmitted. This section proposes a method to reduce data required for streaming video transmission by dynamically changing video quality, particularly the CRF parameter, while maintaining mAP equivalent to that of the conventional H.265 CRF mode.

3.1 Architecture

The architecture of the proposed system, as illustrated in Fig. 3, consists of a surveillance camera, IoT gateway(IoT-GW), base station, and server. The link between the surveillance camera and the IoT-GW is wired, the link between

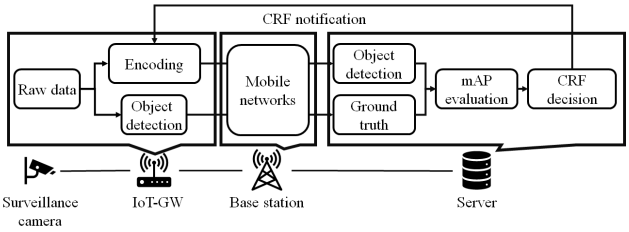

Fig. 3 Architecture of the proposed method.

the IoT-GW and the base station is wireless, and the link between the base station and the server is wired. The video from the surveillance camera reaches the server via the IoT-GW, base station, and mobile networks. The server detects objects from the streaming video, for example, in the case of a vehicle-traffic safety system, and sends alert messages to high-risk vehicles based on the detection results. The robustness of the vehicle-traffic safety system depends on the object-detection accuracy. Therefore, a higher mAP is required.

The sequence of the proposed system is as follows. First, the surveillance camera sends a raw high-quality streaming video (high quality) to the IoT-GW, and then, the IoT-GW performs two processes upon receiving the raw streaming video. In the first process, the IoT-GW inputs the raw streaming video into the object-detection engine to obtain ground truth (GT), including the type of detected objects and their bounding boxes that indicate the location of the objects in the high-quality video. Here, GT obtained from as the raw data was used to calculate mAP on the server. This is because the raw data are the highest-quality video and the detection results based on that high-quality video are the most accurate. The second process involves encoding the video received from the surveillance camera based on the CRF parameters specified by the server. Following the two processes, the IoT-GW sends the GT and encodes the video to the server. When the server receives this data, it calculates mAP using the two received datasets and determines the next streaming video quality. This method for determining the quality of the next streaming video is explained in detail in Section 3.2. When the server determines the quality of the next video, it notifies the IoT-GW of the CRF parameters for the subsequent streaming video.

3.2 CRF selection algorithm

The proposed algorithm controls the quality of the next video frame based on the detection accuracy of the previous frame. As frame rates are increasing and the time interval between frames is becoming smaller, we assumed that the quality of the video in the previous frame is effective in determining the quality of the video in the next frame. Specifically, it determines the CRF for the next video frame CRF_{t+1} based on the current mAP mAP_t of the video encoded by the current CRF CRF_t.

In the algorithm for mAP_{req}, which indicates the required value of detection accuracy, the upper threshold of mAP, mAP_{upper} and the lower threshold of mAP, mAP_{lower} were preset and divided into four states. The next CRF can be determined using the following equation:

$$CRF_{t+1} = \begin{cases} CRF_t + \alpha & (mAP_t \geq mAP_{upper}), \\ CRF_t & (mAP_{req} \leq mAP_t < mAP_{upper}), \\ CRF_t - \beta & (mAP_{lower} \leq mAP_t < mAP_{req}), \\ \gamma & (mAP_t < mAP_{lower}), \end{cases} \quad (1)$$

where α, β, and γ are constants. In the case of $mAP_t \geq mAP_{upper}$, we considered that CRF_t was excessive and CRF_{t+1} was set as $CRF_t + \alpha$ to downgrade the video quality. This degradation decreased the network load. In the case of $mAP_{req} \leq mAP_t < mAP_{upper}$, CRF_{t+1} was set as CRF_t because the current video quality was appropriate. When $mAP_{lower} \leq mAP_t < mAP_{req}$, the algorithm marginally upgraded the video quality. Specifically, CRF_{t+1} was set as $CRF_t - \beta$. When $mAP_t < mAP_{lower}$, the algorithm drastically upgraded the video quality to recover mAP degradation. Specifically, CRF_{t+1} was set to γ, which presents high quality.

When developing a state-of-the-art object-detection engine using the proposed method, the object-detection engines of the IoT-GW and the server, as shown in Figure 3, can be replaced with the latest object-detection engine. Moreover, the video quality can be determined based on the detection accuracy of the object-detection engine. Therefore, if the latest object-detection engine can maintain the desired detection accuracy even when a low-quality video is input, a low-quality CRF can be selected using the proposed algorithm. Thus, the proposed method can be easily integrated into the latest object-detection engine, because the quality of video that can satisfy the desired detection accuracy is dynamically selected by the proposed CRF selection algorithm.

4. Performance evaluation

4.1 Simulation setup

The proposed algorithm was compared with the H.265 CRF mode by simulation. The content shown in Fig. 1 was used for evaluation. The desired detection accuracy mAP_{req} was considered to be 90%. The CRF value that satisfied the desired detection accuracy of 90% in the H.265 CRF mode was 10, as confirmed by a preliminary evaluation. Therefore, the performance of the H.265 CRF mode was measured by setting CRF to 10.

The novelty of the proposed algorithm lies in the fact that it can be used to classify four states using mAP_{upper}, mAP_{req}, and mAP_{lower} and to dynamically control CRF according to these states. Performance evaluation was conducted to confirm the effectiveness of the dynamic control of CRF. Therefore, to satisfy the desired mAP_{req}, the values that corroborated well with the experimental results were used. Specifically, mAP_{upper}, mAP_{lower}, α, β, and γ were set to 98, 88, 2, 2, and 10, respectively.

The dynamic CRF selection calculates CRF_{t+1} every 100 ms at a frame rate of 10 frames per second. This study compared the success rates and bandwidth consumption, and the success rate $SuccessRate_{mAP}$ can be formulated as follows:

$$SuccessRate_{mAP} = \frac{SuccessFrame}{TotalFrame} \cdot 100, \quad (2)$$

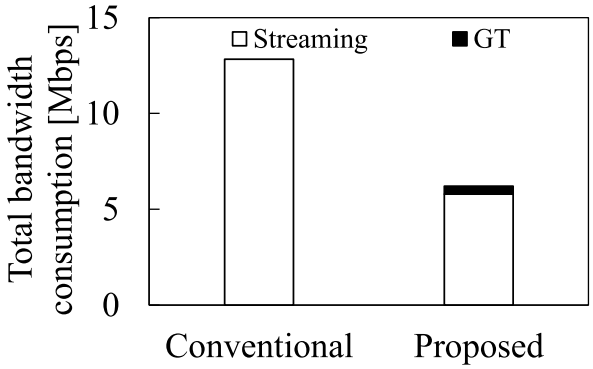

Fig. 4 Total bandwidth consumption.

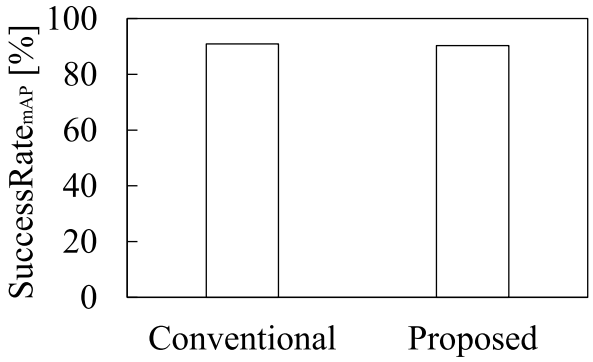

Fig. 5 Success rate.

where $SuccessFrame$ is the number of frames satisfying the desired detection accuracy mAP_{req} during the simulation and $TotalFrame$ is the total number of frames sent by surveillance during the simulation.

The proposed method is considered to be effective if the $SuccessRate_{mAP}$ of the proposed method is equal to or greater than the $SuccessRate_{mAP}$ of the conventional method, and if the bandwidth consumption of the proposed method is less than or equal to that of the conventional method. For the simulation evaluation, mAP_{req} was set to 0.9, and the percentage of frames satisfying the condition mAP_t is equal to or greater than mAP_{req} was calculated.

4.2 Simulation results

The total bandwidth consumption and success rate are presented in Figs. 4 and 5, respectively. The total bandwidth consumption of our proposed method was computed as the sum of the bandwidths of GT and the streaming video. As the H.265 CRF mode does not send GT information, we set GT to zero in the H.265 CRF mode.

As shown in Fig. 4, the total bandwidth consumption of the proposed method is 49% of that of the H.265 CRF mode. Although the proposed method has an extra overhead for GT transmission of slightly 415 kbps, the overhead enables the proposed method to constantly monitor the object-detection accuracy and dynamically select the optimal CRF. Consequently, the proposed method can reduce the total bandwidth consumption.

As shown in Fig. 5, the success rate of the proposed method is approximately equal to that of the H.265 CRF mode. Therefore, the proposed method achieves the same

object-detection accuracy as the H.265 CRF mode while reducing the total bandwidth consumption to 49% of that of the H.265 CRF mode.

5. Conclusion

This study proposed a dynamic CRF control method based on the current mAP to reduce the total bandwidth consumption and achieve the same object-detection accuracy as the H.265 CRF mode. The simulation results reveal that the total bandwidth consumption of our proposed method is 49% of that of the H.265 CRF mode while achieving the same success rates as the H.265 CRF mode. Therefore, the proposed method enables the setting of a CRF that satisfies the desired detection accuracy and adapts to the characteristics of object-detection engines. We will work on the optimization method of the threshold used in the proposed method as a future study.

Acknowledgments

This work was partially supported by JSPS KAKENHI (Grant Numbers 21K21293 and 20K04471).

References

[1] S. Vishnu, U. Ramanadhan, N. Vasudevan, and A. Ramachandran, "Vehicular collision avoidance using video processing and vehicle-to-infrastructure communication," 2015 International Conference on Connected Vehicles and Expo (ICCVE), pp. 387–388, 2015. DOI: 10.1109/iccve.2015.36

[2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[3] W. Robitza, "Crf guide (constant rate factor in x264 and x265)," http://slhck.info/video/2017/02/24/crf-guide.html.

[4] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, vol. 45, no. 4, pp. 325–338, 2015. DOI: 10.1145/2785956.2787486

[5] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, pp. 261–318, 2020. DOI: 10.1007/s11263-019-01247-4

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, "Microsoft coco: Common objects in context," Proc. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, Sept. 6-12, 2014, Part V 13, pp. 740–755, 2014. DOI: 10.1007/978-3-319-10602-1_48

[7] "Coco-common objects in context," https://cocodataset.org.