An extractive streaming system
to reduce monitoring traffic

Kimihiro Mizutani1, a)

1 Department of Informatics, Kindai University, 3–4–1 Kowakae, Higashi Osaka, Osaka 577–0818, Japan
a) mizutani@info.kindai.ac.jp

Abstract: Generally, a monitoring system has tradeoff between network fee and monitoring precision. If the system user wants to monitor a something target from a remote office precisely, the network fee becomes higher because of taking lots of monitoring packets (i.e., video frame). Here, I propose an extractive streaming system to address this problem. The proposed system is installed for both the video sender and receiver. The sender system extracts the pre-determined target area (e.g., license plate) with high quality from a video frame, and sends both roughly compressed entire video frame (e.g., encoded H.264) and the extract area to the receiver in parallel. Subsequently, the receiver system overlays the extracted area on the entire video frame. Through these process, the user can monitor a target area (i.e., focused area) precisely with low network fee. I evaluated the proposed system and investigated the extent to which additional latency and communication traffic was incurred by the proposed system and conducted that the system results in reduced network traffic with decreased latency.

Keywords: streaming application, traffic reduction, monitoring object

References

[1] Y. Romano, J. Isidoro, and P. Milanfar, “RAISR: rapid and accurate image super-resolution,” IEEE Trans. Comput. Imag., vol. 3, no. 1, pp. 110–125, 2016. DOI: 10.1109/tci.2016.2629284

[2] K. Zhang, W. Zuo, and L. Zhang, “Learning a single convolutional super-resolution network for multiple degradations,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 3262–3271, 2018. DOI: 10.1109/cvpr.2018.00344

[3] A. Lucas, A.K. Katsaggelos, S. Lopez-Tapua, and R. Molina, “Generative adversarial networks and perceptual losses for video super-resolution,” IEEE Trans. Image Process., vol. 28, no. 7, pp. 3312–3327, 2019. DOI: 10.1109/icip.2018.8451714

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788, 2016. DOI: 10.1109/cvpr.2016.91

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, “Ssd: Single shot multibox detector,” Proc. European Conference on Computer Vision, pp. 21–37, 2016. DOI: 10.1007/978-3-319-46448-0_2

[6] R. Girshick, “Fast r-cnn,” Proc. IEEE International Conference on Computer Vision, pp. 1440–1448, 2015. DOI: 10.1109/iccv.2015.169
1 Introduction

Internet of things (IoT) devices are widely distributed and running for monitoring some places (e.g., dashboard camera and security camera). The devices price is cheap so that the setup cost of the IoT devices is low. However, the network communication cost among IoT devices and control centers becomes higher than that of setup cost because the network cost bases pay-per-use. To reduce the network cost, the super-resolution system is useful, which transforms the compressed monitored data (i.e., roughly encoded data such as h.264 100 kbps) to larger one with high quality. In concrete terms, the sender system compresses/encodes the sending data to small data, and sends to receiver. The receiver decodes the received data to previous/pre-encoded data. Through these processes, the sender and receiver can communicate any data with low network cost. The conventional super-resolution systems focused on encoding/decoding the entire video frame such that [1, 2, 3]. However, the entire video frame transportation may generate waste traffic in case that the user wants to monitor only a specific part of a video frame, correctly.

To address this problem, I propose a extractive streaming system. In the proposed system, the sender extracts a specific part with high quality using deep learning technique before sending the video frame, and sends both the extracted high quality data and the encoded entire video frame to the receiver in parallel (Fig. 1). The receiver synchronizes the both data and overlays the extract data on the decoded entire video frame. Based on these processes, the proposed system realizes super-resolution of specific part on compressed entire video frame. Therefore, the proposed system can reduce network costs incurred by IoT monitoring devices in cases where the user only wants to monitor a specific part.

Fig. 1. Abstraction of the proposed system.
2 Extractive streaming system

The proposed system must be installed for both video sender and receiver. First, a sender captures a high quality video (i.e., HD and Full-HD), and then inputs each video frame into deep learning algorithm specialized in data extracting noting that it must be trained on extracting targets pre-determined by a user [4, 5, 6]. For example, if a user wants to monitor license plates on high quality, the user must train the deep learning algorithm for detecting/extracting license plate. Second, the sender transmits high quality extracted data, their positions in a frame (e.g., a pair of x and y positions of the license plates), and the roughly compressed/encoded entire video frame to the receiver by Real-time Transport Protocol (RTP) [7]. In general RTP implementation, a timestamp and a sequence number are included as packet header information to maintain the order of the video frames. Therefore, the sender implementation based on RTP can associate the encoded video frame and the extracted data with a RTP sequence when the sender transmitted the all video frames to the receiver. Finally, a receiver stores the all received frames and overlays the extracted data and the frame having same RTP sequence number in real-time. It is noting that the overlay process uses both the extracted data and their positions in a frame.

3 Evaluation

I evaluated the proposed system which used the latest data extraction algorithm based on deep learning (YOLOv3) [8] for extracting a specific part of a video frame, and used H.264 (100 bps) for video frame compression/transmission in real-time. The reason using YOLOv3 is that it can be executed on multi-platform such as RaspberryPi and general purpose computer. I set the target of YOLOv3 as license plates, and aimed to overlay the extracted HD quality license plate to the H.264 (100 kbps) video frame. To train the YOLOv3 for extracting license plates, I collected 1,000 images of vehicles that include clear license plates from a bus company and Kaggle dataset [9]. I used images to train YOLOv3 at 2,000 episodes. I set up YOLOv3 with Adam (learning rate: 0.01), and accepted YOLOv3’s default neural network architecture. The reason why using the Adam optimizer (learning rate: 0.01) is that it was also default setting of YOLOv3 and the extraction accuracy among other optimizer (e.g., Momentum SGD) showed no difference in the results.

With these settings, I evaluated the speed of extracting license plate against different machines. The results are shown in and showed the result into the left of Fig. 2. The number of (a), (b), and (c) indicated Frame Per Second (FPS) of a server equipped with a Xeon E5-2620 and a GTX1080 Ti, a compact mobile PC equipped with a Core i7-8750H and a GTX 1050Ti Mobile, and a RaspberryPi3 model B+ equipped with a Broadcom BCM2837B0 and a Neuro Stick 2, respectively. Note that I only provided CPUs and GPUs specifications because the performance primarily depended on these devices. The results indicate the extracting speed improved as devices’ specifications became better. In case of monitoring with IoT devices, the speed is not sensitive and it is enough to realize over 5 fps in general. Even if IoT devices such (b) and (c) are used, I could conclude the extracting speed is satisfied with general monitoring requirement. Next, I evaluated the extracting accuracy
Fig. 2. Extracting accuracy and the speed.

Fig. 3. The amount of data transfer for the video taken by bus camera around the city.

against the ratio of training data and test data, and showed the result in center of Fig. 2. The result indicated the extracting accuracy had no significant difference even if the ratio changed. Next, I evaluated the extracting accuracy against the learning parameters and showed the result in right of Fig. 2. The result indicated the accuracy changed by the learning rate, and the highest value 72.5% when learning rate was set by 0.01.

Finally, I evaluated the amount of traffic required to transmit the all video frames of three different dashboard cameras against H.264 HD transmission(HD), Extractive streaming (Proposed system), and H264 100 kbps encoded transmission (100 kbps), which are shown in Fig. 3. In the figure, (a), (b) and (c) represented different dashboard camera videos. The video (a) was taken by a bus running around a suburb, and the second video (b) was taken by a bus running around a city road. The third video (c) was taken by a bus running around a mountain road. The length of each videos were 30 minutes and the maximum numbers of license plates per a frame are different in the videos; for examples, a frame in the city road video contained up to four license plates, on the other hands, a frame in the mountain roads video contained up to two license plates.

The result of HD took about 600 MB for transmitting all frames in all videos:
(a), (b) and (c). On the other hands, the results of the proposed system were different
in the videos. In (b), the result was about 40 MB, which is the worst result against
those of (a) and (c). This is because (b) was the video taken in a city so that it
contained a lot of cars’ license plates and their HD data transmissions become larger
than the other videos. The results of (a) and (c) were similar and they were about
32 MB, therefore, the amount of transmission data is quite lower than that of HD
and it is similar with that of 100 kbps. I could conclude that if the user (e.g., a
bus company) wants to monitor the license plate precisely by using cost effective
streaming, the proposed was suitable for its requirement.

4 Conclusion and future perspective

General super-resolution system is suitable for an entire video frame super resolution
with low network fee, however, it is not suitable for viewing a specific parts of a
frame accurately. To address the problem, I proposed extractive streaming system,
which realizes both taking low network fee and the accurately confirmation of a
specific part of a frame. In the evaluation, I confirmed that my proposed system
reduced the amount of transmission data against three videos containing different
situations. Next, I will apply this scheme into loiterer and vermin tracking. In both
tracking, almost all of the monitoring traffic is waste because the suspicious targets
do not appear at all. Therefore, my scheme’s function of real-time target monitoring
with lower traffic is effective solution for them and it is worth applying it into them.

Acknowledgments

The author thanks the member of Minato Kanko bus company for collecting research
data (e.g., license plates and dashboard camera’s videos) used in this paper.