COMPRESSED VIDEO STREAM BASED OBJECT DETECTION

PYEONG KANG KIM[1] and HYUNG HEON KIM[1]
TAE WOO KIM[1] and Young Kyun Cha[2]

[1]Planning & Strategic Team, INNODEP.inc, Seoul, Korea
[2]School of Information Security, Korea University, Seoul, Korea

ABSTRACT

Nowadays, the need for research on an intelligent video monitoring system is increasing worldwide. Among the object detection methods, the core technology of the intelligent video monitoring system, or object detection using a deep learning-based convolutional neural network, is used widely due to its proven performance. Nonetheless, deep learning-based object detection requires many hardware resources because it decodes the videos to analyze. Therefore, this article suggests an advanced object recognition technique by conducting compressed video stream-based object detection in order to reduce consumption of resources for object detection as well as improve performance and confirms via the performance evaluation that speed and recognition rate improved compared to existing algorithms such as YOLO, SSD, and Faster R-CNN.

KEYWORDS

object detection, convolutional neural network, Inception V4, Motion Vector

1. INTRODUCTION

Nowadays, with acts of terrorism, crimes, and safety accidents occurring in public spaces all over the world, the technology for detecting and classifying objects in video is gaining more attention, and the need to research it is increasing. Object recognition is a technology for detecting and classifying objects in video to recognize their types. In particular, among object recognition technologies, the technology for detecting objects is regarded as a difficult one because it should be able to detect the type and location of objects in video. The existing technologies for detecting objects such as low-level SIFT (Scale-Inveriant Feature Transform)[1], SURF (Speeded-Up Robust Feature)[2], Haar Wavelet[3], and HOG (Histogram of Oriented Gradient)[4] have limited performance. Since 2012, the deep learning-based convolutional neural network has gained much attention because it showed greater performance than the existing ones in ImageNet 2012[5]. As a result, many researches have recently been conducted to utilize the convolutional neural network for object detection. Deep learning-based object detection technologies are composed of R-CNN[6], technology for detecting the object candidate region (region proposal) in the input image based on the Selective Search algorithm, SPPNet[7], Fast R-CNN[8], and Faster R-CNN[9]. On the other hand, YOLO (You Only Look Once)[10] and SSD (Single Shot multi-box Detector)[11] (more accurate version of YOLO) divide the input image into grids in a fixed size for real-time application and show the odds of the bounding box in each grid turning into the bounding box of the target object, including the class of each object through posterior probability.

Natarajan Meghanathan et al. (Eds) : ACITY, AIAA, DPPR, CNDC, WIMNET, WEST, ICSS - 2019
pp. 81-88, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91708

Note, however, that deep learning-based object detection technology relies heavily on hardware performance, and it is difficult to process in real time or to process various channels simultaneously because it decodes the input video and analyzes the decoded video. Therefore, to resolve the problems of the existing deep learning-based object detection methods, this study detected moving objects using the motion vector information of compressed video stream to reduce the computation workload and classify using a modified Inception V4 structure in order to suggest a more efficient object recognition algorithm.

2. COMPRESSED VIDEO STREAM-BASED OBJECT DETECTION MOETHOD

2.1. Object Detection Using Motion Vector

Generally, video is decoded to reduce its size (volume). Therefore, decoding is a process to compress video. Video compression performs prediction with two methods: intra-prediction and inter-prediction. The role of prediction is to predict the original signal and similar signals to send the difference value. The better the prediction is, the more efficient the decoding.

Prediction is divided into intra-prediction and inter-prediction. Intra-prediction creates a prediction block using the current block and adjacent pixels and sends residual data (difference). In this case, the residual signal and prediction mode index indicating the direction of prediction should be sent. Inter-prediction supports the assumption of motion. Motion assumption finds the block with the minimum difference value compared to the current block within the search scope to discover the prediction block and sends the residual signal (difference value between the current and prediction blocks). In this case, the movement parameter indicating the location of prediction block is also sent. The motion parameter contains 3 elements: motion vector indicating the location of prediction block; reference index indicating the image containing the motion vector; and prediction direction flag indicating whether the prediction is conducted from the past or future image. Among them, the motion vector may have a very large value; thus requiring many bits to be sent. As such, in the case of motion vector, the prediction motion vector is extracted through prediction, and only the residual vector (difference between the current motion vector and the prediction motion vector) is sent. It is a method that conducts motion vector prediction to obtain the prediction motion vector and sends only the difference vector with the motion vector of the current block. This motion vector prediction is called AMVP (Advanced Motion Vector Prediction).

2.2. Detection of Motion Vector

As described above, motion vector is the distance of movement (displacement) in motion assumption, displayed as vector. In short, it is the location coordinates difference vector in prediction encoding based on motion assumption and motion consumption. Figure 1. is a diagram of the motion vector.

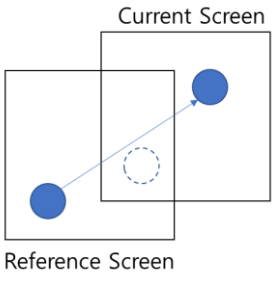

Figure 1. Motion Vector

In this article, among AMVPs, mean absolute difference is used to detect motion vector. Mean absolute difference is a measure to evaluate the similarity between two blocks. Formula (1) is the formula for mean absolute difference.

$$MAD_{(k,l)}(x,y) = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{i=0}^{N-1}|I_t(k+i,l+j) - I_{t-1}(k+x+i, L+y+j)| \quad (1)$$

The value of displacement (x,y) that minimizes the mean absolute difference is regarded as motion vector of the relevant block, as shown in Formula (2)

$$v(k,l) = \underset{(x,y)}{\arg min} MAD_{(k,l)}(x,y) \quad (2)$$

2.3. Detection of Region Proposal

Once the calculation of motion vector is finished, as first action, the accumulated value of the motion vector in each block of video is obtained. In this case, the block of video contains macro block and sub block, and the accumulated time is set as 500ms by identifying the pattern information composed of average data of meaningful motion object and noise. In the 2nd step, to identify whether the accumulated motion vector area is the assumed area of the actual motion object, after comparing with the pre-established threshold value, the video block with accumulated motion vector value exceeding such threshold value is set as motion object area. Even if a motion vector is generated, if the accumulated value for a period of time is less than the threshold value, it should be regarded as noise and ignored in the detection stage. In the 3rd step, the boundary of the motion object is detected. The region proposal detected in the 2nd step is smaller than the actual object size, or multiple detection areas will be set in one object.

Figure 2. describes the result of the 1st detection for region proposal.

Figure 2. 1st detection for region proposal

To resolve this problem, various video blocks adjacent to the 1st candidate area block should be identified again. The motion vector value for the various adjacent video blocks was compared to the 2nd threshold value, and the block with motion vector value exceeding the 2nd threshold value is set as the region proposal block again.

In addition, adjacent blocks whose coding type is intra picture were set as object area again. Intra-picture does not have a motion vector, so it is impossible to determine whether an object is in the

adjacent block based on the motion vector. In this case, the adjacent intra-picture should be regarded as region proposal to include the object area. Figure 3. shows the result of detection of region proposal.

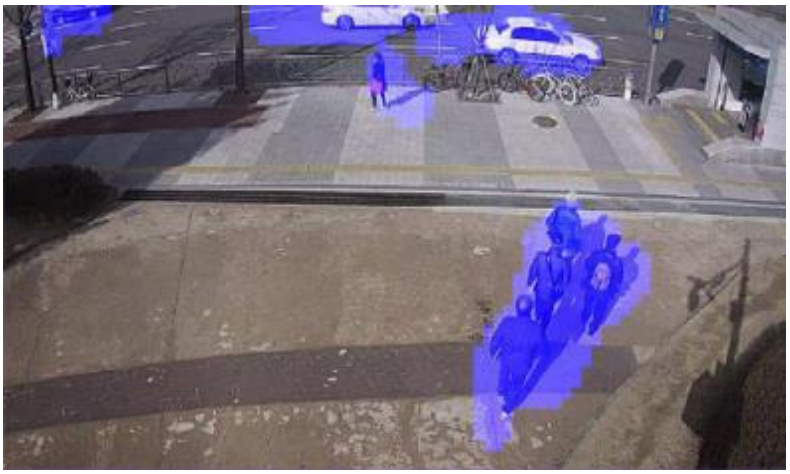

Figure 3. Result of detection for region proposal

3. OBJECT CLASSIFICATION BASED ON INCEPTION V4

The most intuitive way to improve the performance of CNN is to deepen its structure. Note, however, that a deep CNN structure is likely to cause problems such as overfitting and vanishing gradient, and the required calculation workload is increasing drastically. This problem can be resolved by reducing CNN's density (sparse structure), but dense structure is better to resolve the problem of calculation workload.

Inception is designed as a structure with this advantage. In case of the Inception network, various features can be extracted by performing layers of convolution in various sizes and pooling layers in parallel and merging its result. This structure requires heavy calculation workload to perform convolution in various sizes. To resolve this problem, a bottleneck structure that reduces dimension using convolution with size of 1 is established. Basically, the Inception network has a deeper structure than the VGG16 model, so it prevents Gradient vanishing/Gradient Exploding indirectly using the modification of activation functions (ReLU, etc.) and Initialization; note, however, that it is also possible to stabilize overall learning by applying Batch Normalization methods to Inception V3 to conduct Batch normalization prior to activation functions.

Recently, Inception V4 -- an upgraded version of V3 -- was introduced, and it showed excellent performance.

In this article, the Inception V4 network is used to classify region proposal based on stabilized learning and high performance.

3.1. Object Classification

Figure 4. is a diagram of the Inception V4 model used in this article to classify objects. To classify objects, the region proposal of the video inputted through the aforesaid detection of compressed stream-based region proposal was set as the input video. Input size was set as 128x128x3 considering speed and performance. The input video will be used to generate the feature map in 35x35x384 size from the area with 9 layers (Stem). It will then pass Inception-A

and Inception-C in various layers. In particular, the structure of the existing Inception V4 passes the 3 Inception structures; in this article, however, Inception-B was skipped to improve speed.

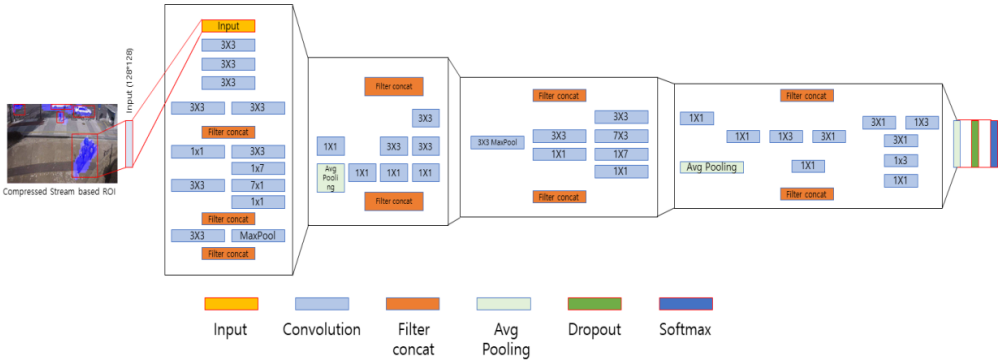

Figure 4. Inception V4

Then, to reduce the size of the feature map, classification was performed via Softmax in the final layer after passing Reduction-B

4. REALIZATION AND TEST

For realization and performance test, Windows 10 OS and Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz, RAM 64.0GB, GPU NVidia Quadro P5000 were used. As Video Dataset, Caltech Video Dataset was used.

Caltech Video Dataset is composed of around 250,000-frame videos containing 350,000 bounding boxes and 2,300 pedestrians taken from a vehicle driving in the urban environment. Each video is 30Hz with size of 640 × 480 pixel, and the total play time is 10 hours. Figure 5. shows the Caltech Video

Figure 5. Caltech Video Dataset

If the detection result is overlapped with Ground truth by more than 50%, it is regarded as true positive. Otherwise, it is regarded as misdetection. In short, it should be regarded as success if. (Note, however, that R is the boundary quadrangle of the detection result, and G is the boundary quadrangle of Ground truth.)

4.1. Implementation

The compressed stream-based object detection and classification suggested in this article is realized as shown in Fig. 5.2. Detailed parameters for realization referred to the design, and classification classes were divided into pedestrian, vehicle, and others. Fig. 5.2(a) shows the correct recognition, and (b) shows the wrong recognition for the actual screen

(a) correc recognition (b) Wrong recognition

Figure 5. Implementation Results

4.2. Performance Test for the Object Detection Method

To test the performance of the compressed stream-based object detection method suggested in this article, the processing speed of the existing method that analyzes decoded video to detect objects was compared with that of the suggested method. In case of the existing object detection method, the processing time was defined as a period from decoding of video to object detection. In case of the suggested algorithm, it was defined as the period from parsing of compressed stream to detection of region proposal.

Table 1. Summarizes the evaluation of object detection

Method	**VA with Decompressed (Calm background+Point generation method:SIFT)**	**VA without Decompressed**
Test time per image	0.17	0.01

As shown in Table Ⅰ, the processing speed of the suggested algorithm is around 17 times faster than that of the existing algorithm..

4.3. Performance Test for Object Classification

To test the performance of object classification methods through the compressed stream-based region proposal suggested in this article, it was compared with the deep learning-based object detection and classification algorithms. The existing algorithms used for comparison were R-CNN, Fast R-CNN, Faster R-CNN, and YOLO, and the evaluation items were processing speed, classification performance, etc. Table Ⅱ presents the test results of the object classification methods.

Table 2. Summarizes the evaluation of object detection

Method	R-CNN	Fast R-CNN	Faster R-CNN	YOLO	Suggested method
Test time per image	50	2	0.2	0.022	0.02
Speedup	1x	25x	250x	2,272x	2,500x
mAP(Caltech)	78	78.9	78.9	72	80.1

The processing speed of the suggested method is 1.1~2,500 times faster than that of the existing algorithms. Classification performance is also relatively higher than that of the existing algorithms.

5. CONCLUSION

The existing methods that detect region proposal by analyzing decoded video are not versatile due to slow processing speed and high dependency on hardware, since decoding requires many computing resources and the entire video should be analyzed. To resolve this problem, this article suggested an object recognition algorithm (VADNET) that detects motion region proposal by detecting motion vector based on the compressed stream of decoded video and classifies objects through the Inception[24] network.

To design the algorithm suggested in this article, the motion vector of compressed stream was analyzed to detect objects in the region proposal; a deep learning-based network that uses this as input video was then formed to design and realize a model for classifying object class.

To test the performance of the suggested algorithm, Caltech Video Dataset was used. The test result shows that processing speed increased by 17 times compared to that of the existing object detection methods. The test result of object classification methods shows that classification performance improved slightly, but processing speed improved by up to 2,500 times.

ACKNOWLEDGEMENTS

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2014-3-00123, Development of High Performance Visual Discovery Platform for Realtime and Large-Scale Data Analysis and Prediction)

REFERENCES

[1] D.G. Lowe, "Distinctive Image Features from Scale Invariant Keypoints," Inte. J. Comput. Vision, vol. 60, no. 2, 2004, pp.91-110.

[2] H. Bay et al., “Speeded-Up Robust Features(SURF),” Comput. Vision Image Understanding, vol. 110, no. 3, 2008, pp.346-359.

[3] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” Proc.IEEE Conput. Soc. Conf. Comput. Vision Pattern Recogn., Kauai, HI,USA, Dec. 8-14, 2001, pp.I:511-I:518

[4] N.Dalal and B.Triggs, “Histograms of Oriented Gradients for Human Detection,” IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn., San Diego, CA, USA, June 20-25, 2015, pp. 886-893.

[5] A. Krizhevsky et al, “ImageNet Classification with Deep Convolutional Neural Networks,” Conf. Neural Inform. Process. Syst., Lake Tahoe, NV, USA, Dec. 3-6, 2012,pp.1097-1105.

[6] R. Girshick et al., “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, ” IEEE Conf. Comput. Vision Pattern Recogn., Columbus, OH, USA, June 23-28, 2014, pp.580-587.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” In ECCV, 2014.

[8] R. Girshick, “Fast R-CNN,” IEEE int. Conf. Comput. Vision, Santiago, Chile, Dec. 7-13, 2015, pp. 1440-1448.

[9] S.Ren et al., “Faster R-CNN:Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. Pattren Anal. Mach. Intell., vol. 39, no. 6, 2017,pp.1137-1149.

[10] J.Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” IEEE, Conf. Comput. Vision Pattern Recogn., Las Vegas, NV, USA, June 27-30,pp.779-788.

[11] W.Liu et al., “SSD: Single Shot MultiBox Detector,”Eur.Conf.Comp.Vision,Amsterdam, Netherlands, Oct. 8-16,2016,pp.21-37.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna,“Rethinking the inception architecture for computer vision,” In Proc. CVPR, 2016.