International Journal of Internet, Broadcasting and Communication Vol.11 No.4 76-85 (2019)
http://dx.doi.org/10.7236/IJIBC.2019.11.4.76

IJIBC 19-4-12

Sub-Frame Analysis-based Object Detection for Real-Time Video Surveillance

Bum-Suk Jang[1], Sang-Hyun Lee[2]

[1]BS SOFT Co., LTD,
bsjang@bs-soft.co.kr
[2]Department of Computer Engineering, Honam University, Korea
leesang64@honam.ac.kr

Abstract

We introduce a vision-based object detection method for real-time video surveillance system in low-end edge computing environments. Recently, the accuracy of object detection has been improved due to the performance of approaches based on deep learning algorithm such as Region Convolutional Neural Network(R-CNN) which has two stage for inferencing. On the other hand, one stage detection algorithms such as single-shot detection (SSD) and you only look once (YOLO) have been developed at the expense of some accuracy and can be used for real-time systems. However, high-performance hardware such as General-Purpose computing on Graphics Processing Unit(GPGPU) is required to still achieve excellent object detection performance and speed. To address hardware requirement that is burdensome to low-end edge computing environments, We propose sub-frame analysis method for the object detection. In specific, We divide a whole image frame into smaller ones then inference them on Convolutional Neural Network (CNN) based image detection network, which is much faster than conventional network designed for full frame image. We reduced its computational requirement significantly without losing throughput and object detection accuracy with the proposed method.

Keywords: *object detection, object tracking, convolutional neural network, real time video surveillance, sub-frame analysis.*

1. INTRODUCTION

Object detection has become a very popular field due to the recent development of deep learning based inference network such as region convolutional neural network (R-CNN), single-shot detection (SSD), and you only look once (YOLO). Many researchers have investigated object detection to use on real-time video surveillance system. However, deep learning based networks usually demands general purpose graphic processing unit (GPGPU) to ensure its performance regarding frame per second (FPS) and detection accuracy. However, many edge devices still do not include the GPGPU and/or software framework to support the deep learning inference. For this reason, computation efficient approach for the object detection is highly required for intelligent video surveillance system with mainstream IoT devices. Following section introduces the proposed sub-frame analysis method for computationally efficient object detection.

Manuscript Received: October. 15, 2019 / Revised: October. 21, 2019 / Accepted: October. 30, 2019
Corresponding Author: leesang64@honam.ac.kr
Tel: +82-62-940-5285, Fax: +82-62-940-5285
Department of Computer Engineering, Honam University, Korea

2. RELATED WORKS

2.1 Object Detection Algorithm

Object detection pipelines generally start extracting a set of robust features from input images. Then, classifiers or localizers are used to identify objects in the feature space. These classifiers or localizers are run either in the sliding window fashion or the entire image.

On the other hand, one-stage methods such as YOLO and SSD use networking features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This means one-stage methods network reasons globally about the full image and all the objects in the image.

Recently, one-stage methods are upgraded which called YOLO [1]. It changes to use a more complex backbone for feature extraction [2]. To better detect small objects, it adds Feature Pyramid. Since it scans the images in one round, its time complexity is much smaller than heavier detectors such as R-CNN [3].

However, one-stage methods imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can only have one class.

Therefore, if there are several objects in the grid, there is a limit to the detection performance. Also, if we increase the number of grid by reducing the size to improve the performance of object detection, the number of bounding boxes for learning and reasoning increases exponentially, so scalability for performance improvement is restricted.

In addition, other papers show that in addition to Grid size, there is a classification algorithm suitable for a specific situation such as Surveillance [4].

2.2 Correlation based Object Tracking

Recently, studies for multi object tracking have been combined with detection algorithm to achieve high performance. The correlation-based object tracking algorithm to be used in this paper is based on the article of danelljan [5].

This algorithm works by learning discriminative correlation filters based on a scale pyramid representation. This learn separate filters for translation and scale estimation, and show that this improves the performance compared to an exhaustive scale search. Scale estimation approach of this algorithm is generic as it can be incorporated into any tracking method with no inherent scale estimation.

Therefore, tracking performance depends on the bounding box of the object to be tracked in order to obtain excellent performance.

2.3 Object Detection and Tracking as an Edge Service

In order to migrate object detection and tracking services to edge network devices, decisions must be made immediately on-site.

In addition, for architectural scalability, it is an indispensable approach to reduce the computational burden on the server or cloud, to perform operations on the edge device, and then to transmit the results to the cloud in terms of efficient use of the network.

Nevertheless, video processing proves to be a huge computational burden for the edge device and the decision-making task should be outsourced to the fog or even cloud nodes for execution. When the fog node is responsible for decision-making, near real-time execution is achieved.

Edge devices cannot afford the storage space for parameters and weight values. Even after the time consuming and computing intensive training is outsourced to the cloud or other powerful nodes, they still need

a huge volume of memory. For example, the most well-known ResNet has three architectures with 51, 101, or 201 layers respectively, which implies a huge number of parameters to be loaded into the small device's memory.

Lookup-based Convolutional Neural Network(L-CNN) which proposed from Seyed Yahya Nikouei [6], reduces the computational cost of the CNN itself, without much sacrificing the accuracy of the whole network. Also, the network is specialized for human detection to reduce unnecessary huge numbers in each layer. Moreover,

In addition, a study was conducted to combine object detection and tracking algorithms to solve the inverse relationship between camera coverage and object detection performance [7].

3. PROPOSED OBJECT DETECTION USING SUB-FRAME ANALYSIS

3.1 Object Detection and Tracking

Bounding Box to perform object localization and classification in the city, Yolo divides the image into a grid of uniform size and extracts two candidate bounding boxes per grid. In the extracted candidate bounding box, the bounding box is finally extracted through Non Maximum Suppression(NMS) and the object can be recognized through the class identifier of the bounding box.

To keep the size of the grid constant, Yolo forces the input image to a size that the inference engine can process. The change size can be arbitrarily set within a multiple of 32. The quality of the bounding box differs depending on the resize the image.

In addition, since the number of objects detected in the grid cannot exceed the number of candidate bounding boxes, there are objects that cannot be detected when there are many objects in the grid.

Also, when the network model for object reasoning in each grid becomes smaller like Tiny version, it can be seen that the bounding box quality of object drops significantly.

The following figure depicts object performance according to resize size. When resizing the same image to 640 × 320, 12 objects were detected. When resized to 320 × 160, 10 objects were detected. In addition, when the same image was inferred as YoloV3 Tiny (learning and inference network minimized version), six objects were detected in 640 × 320 size and the object could not be detected in 320 × 160.

Also, for the same object, 640 × 320 showed higher Intersection of Union (IoU) than Ground Truth. As shown in Fig 1.

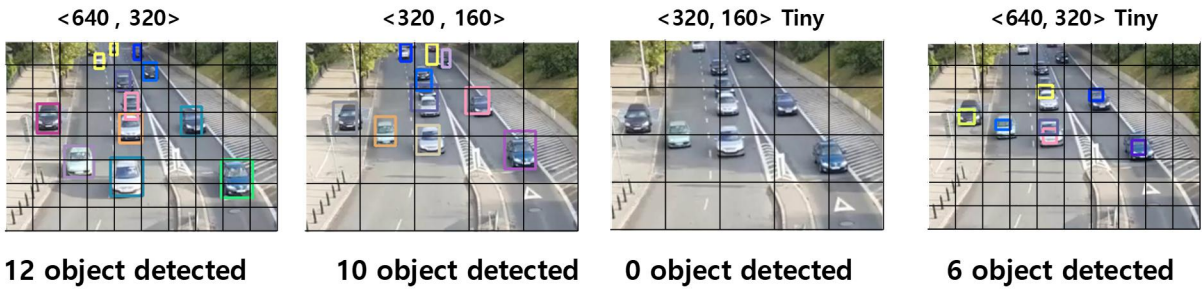

Figure 1. The number of detection depends on grid size.

The quality of object tracking is determined by how precisely the bounding box boxed the object in the detection phase. As shown in Fig 2, the quality of the bounding box depends on the resizing size and the detection algorithm. In general, the higher the resolution and the more sophisticated the detection algorithm, the higher the bounding box quality.

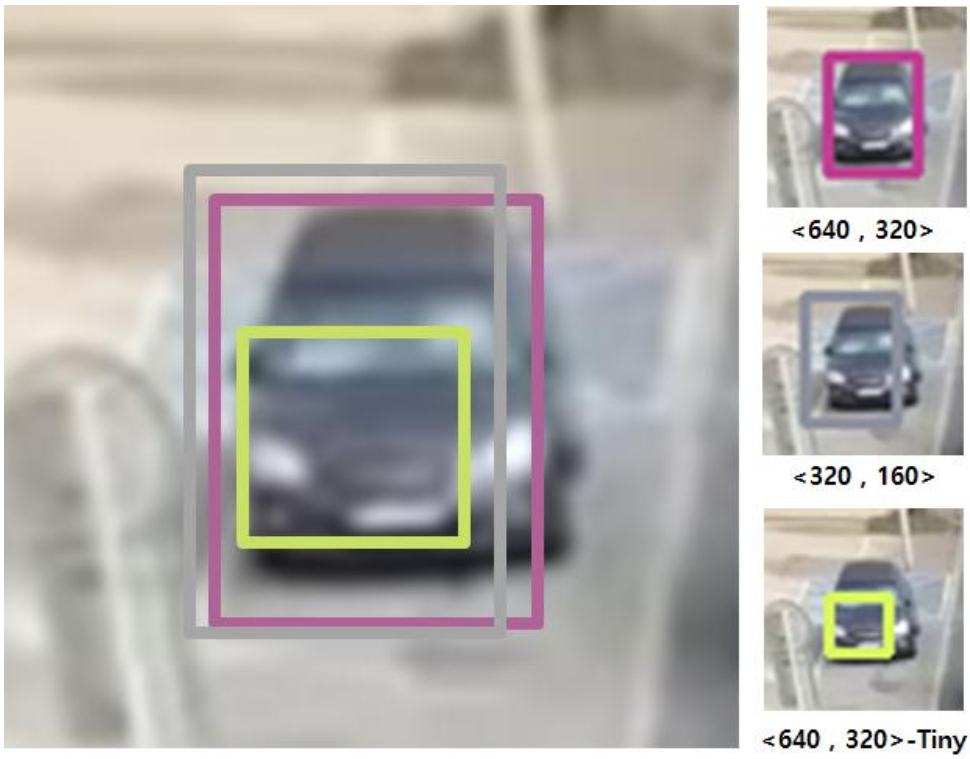

Figure 2. Object detection performance depends on grid size

This is because in the one-stage method, the image is divided into a grid size of a constant size. If the size of the object to be detected is too large for the grid size, it can be subdivided and ignored if it is too small.

tracking interval In this paper, we propose a new approach to video analysis, which is based on video analysis. In video analysis, besides analyzing images, it is significant to explore association [8, 9, 10, 11] among image frames. Most recent studies have focused on tracking-by-detection [12].

Since the amount of object tracking is smaller than object detection, we combine object detection algorithm and object tracking algorithm to achieve real time object detection. To increase the proposed method's throughput, set the object detection period and track the object for video surveillance between intervals. When the tracking interval is N, the computation per second TH can be expressed as follows.

$$TH = DETECTION(FPS - N) + TRACKING(N)$$

Where, DETECIONT(N) is time to detect object of N frames and TRACKING(N) is time to track object of N frames. Therefore, the value of TH converges close to TRACKING (N) when the tracking interval is very high. However, since object tracking is not perfect and there may be objects newly found or disappearing in the middle of the frame, an appropriate level of interval should be set to improve detection performance.

The following figure depicts the operation of the algorithm according to the tracking interval cycle.

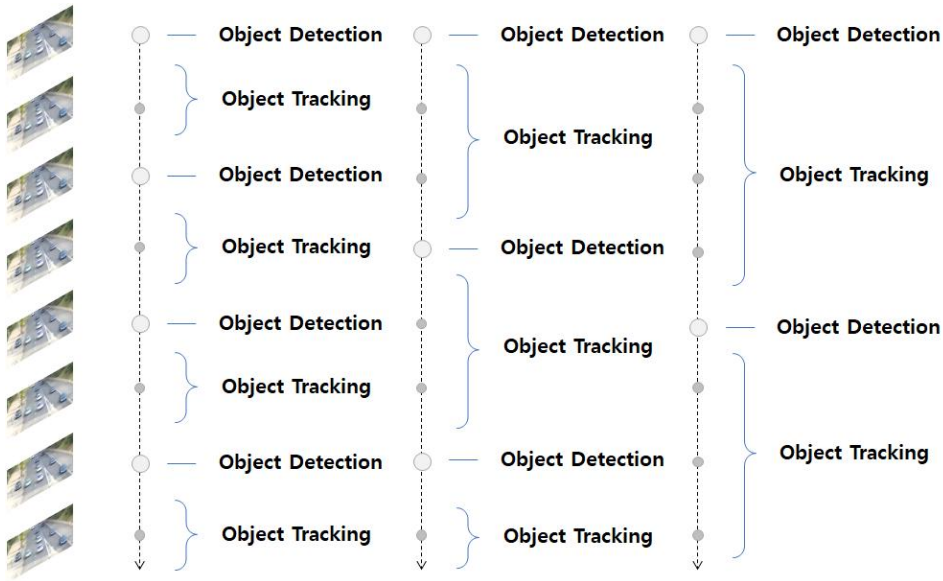

Figure 3. Object Detection with Tracker overview

3.2 Sub-Frame Analysis

YOLO is one of the fastest architectures and has been applied in context with very limited computational resources [13]. In this paper, we developed sub-frame analysis based method to improve detection performance in real video surveillance system. Since one-stage method detects an object by resizing the input image, resize size can be determined according to hardware performance and characteristics of environment to be detected. It is advantageous to reduce the size for real time object detection. But, excessive resizing of target image can cause missing when detecting objects. Instead of resizing the image, sub-frame analysis based method divides it into a certain size that can be deduced from Yolo and infer it sequentially. The Fig 4. depicts the behavior of sub-frame analysis based method. It can solve the object detection missing of the image and not increase the amount of computation.

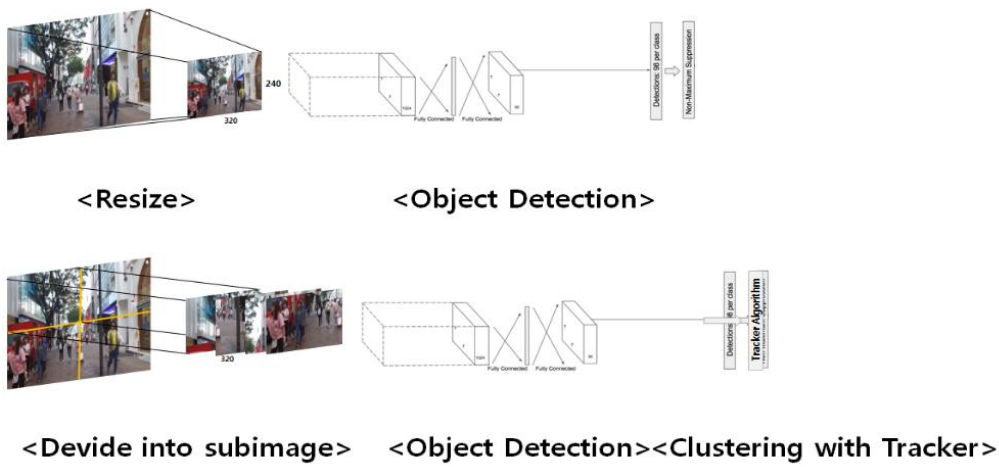

Figure 4. The concept of sub-frame analysis based method

3.3 Sub-Frame Analysis based Object detection and tracking

Since sub-frame analysis based method detects only a part of the image after dividing the image into a certain size, accurate detection is difficult when the object moves between sub-images or exists between them. As shown in the figure below, there exist objects that are not detected or not detected correctly compared with the actual image Fig 5. (B).

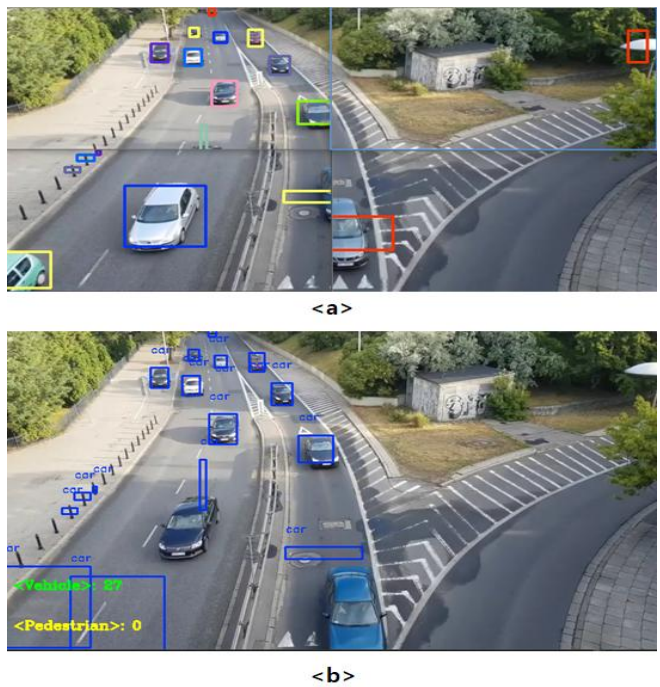

Figure 5. Problem of sub-frame analysis based method

In order to solve the above problem, we solved the problem according to the inference time of the sub-image by driving the tracking algorithm that tracks the detected objects in each sub-image. The figure below is a diagram depicting the outline of sub-frame analysis based object detection tracker.

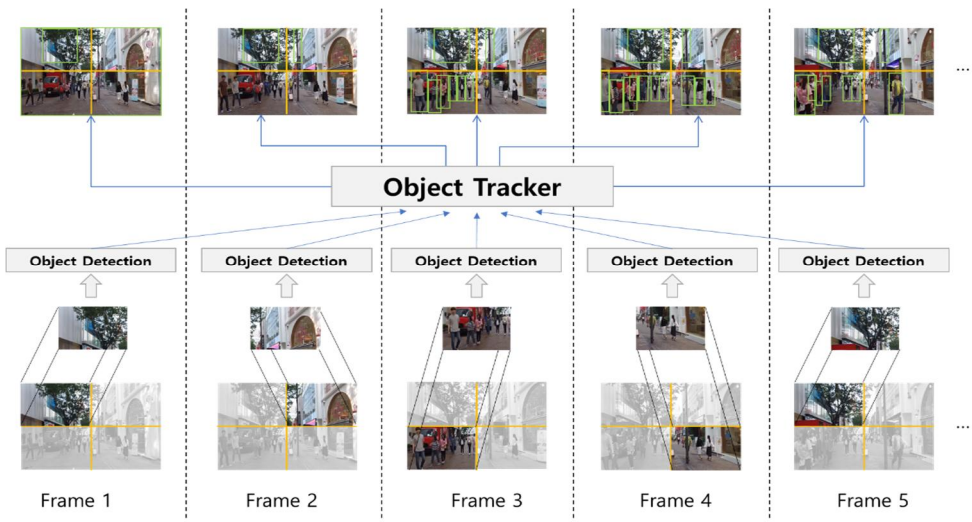

Figure 6. Sub-frame analysis based object detection tracker overview

The object tracker manages the detected objects after performing object detection on sub-image of frame 1 to 1 quadrant, and object tracker further manages the object tracking for each sub-image in the frame afterwards. The object tracker updates and tracks the list of detected objects by performing object detection again on the sub-image detected in frame 1 in frame 5.

In this way, the accuracy of object detection and tracking according to the parallax between sub-images can be increased, and the increase in the computation amount can be suppressed. In conclusion, in contrast to pipelining object detection and tracking when analyzing video, this method plays a role of detected object manager which merges detection result of sub-image according to time difference.

Therefore, when analyzing a video with a larger size resolution, the existing method has a complexity of O(N2) when the size of the image is increased to N when the size of the image is doubled besides the complexity of O (N) of our proposed method.

The propose sub-frame analysis based object detection consists of two steps, which are sub-frame-wise object detection and the object tracking across the sub-frames. At the first step, a full image frame is segmented into smaller sub-frames. Next, object detection network with reduced input resolution detects object of each sub-frame.

Detection result of each sub-frame is temporarily stored in the object buffer and then used by object tracker, which is going to conduct at the second step of the proposed method. That is, the object tracking conducts on the full frame to track objects move across the sub-frames.

4. PERFORMANCE EVALUATION

In this section, we will demonstrate how sub-frame Analysis based Object Detection Tracker helps object detection and tracking system improve the performance at a certain image size. Lastly, we compared the Yolo V3 as a fastest object detection algorithm and sub-frame Analysis based Object Detection Tracker with several tracking interval environment.

4.1 Setup and evaluation

We developed the object detection and tracking system followed the framework described in section 3. We choose YoloV3 which trained with CoCO dataset [14]. And result of YoloV3 is assumed to be Oracle and the performance of the sub-fame analysis based object detection tracker is evaluated through Accuracy, Precision, Recall & F1 Score. Four key variables are used to calculate the above items. Each variable is shown below.

True positive and true negatives are the observations that are correctly predicted and therefore shown in green. We want to minimize false positives and false negatives so they are shown in red color. These terms are a bit confusing. So let's take each term one by one and understand it fully.

	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Figure 7. Description of TP, FN, FP, TN

Once you understand these four parameters then we can calculate Accuracy, Precision, Recall and F1 score.

Accuracy: Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

$$\text{Accuracy} = TP + TN/TP + FP + FN + TN$$

Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = TP / (TP+FP)$$

Recall: Recall is the ratio of correctly predicted positive observations to the all observations in actual class

$$\text{Recall} = TP / (TP+FN)$$

F1 score: F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

$$\text{F1 Score} = 2 \times (\text{Recall} \times \text{Precision}) / (\text{Recall} + \text{Precision})$$

4.2 Speed Analysis

The major computational cost of the proposed method is object detection tracking for image of video stream and object tracking. When using 5 seconds video clip of road traffic, the speed of the proposed method is 5 frames per second with Python and without GPU environment at 0 tracking interval. We compare the speed of our method with original YoloV3 and YoloV3-tiny algorithm with same trained model and environment.
As shown in Fig 8, sub-frame analysis based object detection tracker showed higher throughput performance in both YoloV3 and the version of Tiny. In the 320 size, the average performance was 55% higher. In the case of the 640 size image, the performance was 103%, and in the case of 960 size image, the performance was 315%.

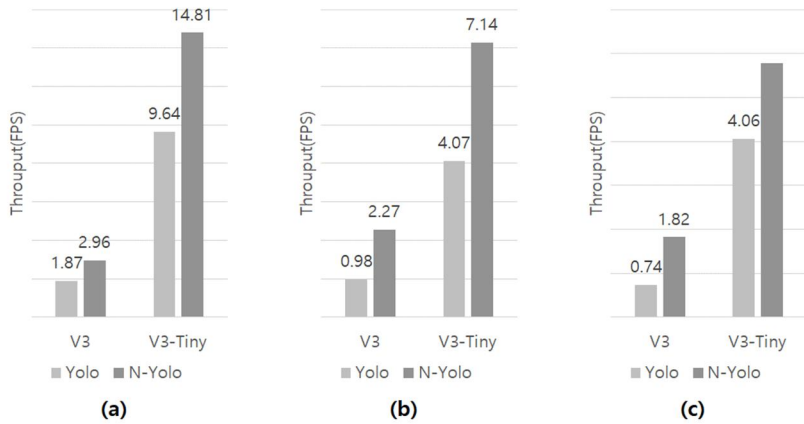

Figure 8. Graph of comparsion sub-frame analysis based object detection tracker with Yolo, YoloV3 and YoloV3-tiny. (a) is experiment with video has 320 × 320 resolution, (b) 640 × 320 and (c) 960 × 320.

4.3 Qualitative evaluation

It also plots the results of accuracy, precision, recall, f1 score, and IoU for qualitative comparison.

The accuracy performance decreases as the tracking interval becomes longer. Throughput performance improves, but it converges to a specific value when the interval becomes longer. Therefore, it is advisable to set the appropriate tracking interval according to the characteristics of the input video and the environmental requirements, considering the trade-off between accuracy and throughput.

As shown in Fig 9 (a), the minimum tracking interval to achieve 6 FPS is 36 frames, with an F1 score of 0.66 and an average IoU value of the experimental results and Ground Truth is 0.62. On the other hand, as described in (b), in N-YoloV3, the tracking interval to achieve 6 FPS is 5 frames, the F1 score is 0.73, and the average IoU value with the ground truth is 0.75. Thus, the sub-frame analysis based object detection tracker with YoloV3 has about 10.6% better F1 score and 20.9% better average IoU than YoloV3-Tracker.

Consequentially, as shown in Fig 9 (a), Efficiency goes up as the resolution increased and As shown in Fig 9 (b), Throughput goes up as the object detection perform rarely, but another metric gradually decreased. Considering both the throughput and the accuracy, we chose the object detection interval to 11.

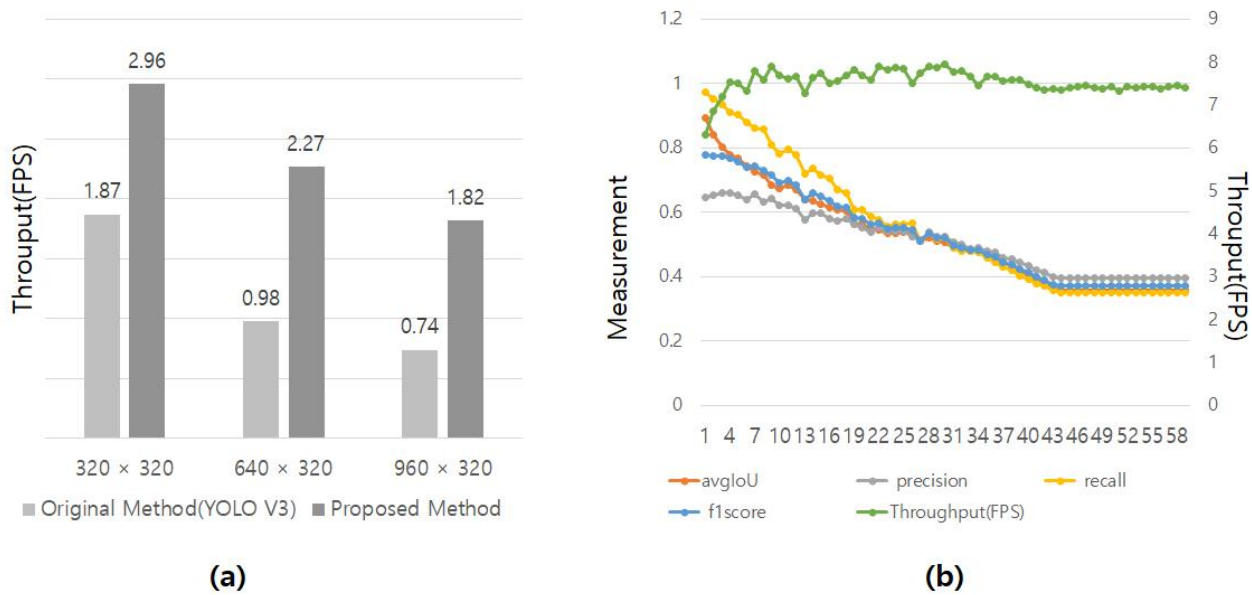

Figure 9. (a) Comparison between original method and proposed method and (b) Performance of the proposed method for different object detection interval

5. CONCLUSION

In this paper, the sub-frame analysis based object detection method was proposed for real-time video surveillance application. Experiment revealed that the proposed method could operate the object detection with F1 score of 0.74 and 5 FPS on GPU-less low-cost IoT.

The sub-frame analysis based object detection tracker with YoloV3 and correlation-based tracker, we developed a real-time object detection and tracker.

By using image segmentation and image merging with tracking algorithm method. Extensive experiments have been done to verify the reliability of our proposed method. This method has scalability for real-time video surveillance in edge computing environments with limited computing power.

In order to accomplish this achievement, we concluded that quality of bounding box is important for efficient operation of correlation-based object tracking and combined with YoloV3 which produces the most efficient bounding box among object detection algorithms. In addition, the object tracker is used as a merging manager of detected objects to improve linearity scalability.

REFERENCES

[1] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR)," 2017 IEEE Conference on, pp. 6517–6525. IEEE, 2017.

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

[3] R. Girshick, "Fast r-cnn," In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[4] S.B. Lee, H.G. Kim, H.K.Seok, J.H. Nang, "Comparison of Fine-Tuned Convolutional Neural Networks for Clipart Style Classification" International Journal of Internet, Broadcasting and Communication(IJIBC), Vol.10 No.4, pp.50-64, 2018
DOI: https://doi.org/10.7236/IJIIBC.2018.10.4.50

[5] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan and Michael Felsberg, "Accurate Scale Estimation for Robust Visual Tracking," In Proceedings British Machine Vision Conference, 2014.
DOI: https://doi.org/10.5244/C.28.65

[6] Seyed Yahya Nikouei, Yu Li Chen, Sejun Song, Ronghua Xu, Baek-Young Choi, Timothy R. Faughnan, "Real-Time Human Detection as an Edge Service Enabled by a Lightweight CNN," IEEE International Conference on Edge Computing (EDGE), 2018.

[7] H.M. Kwon, V. Kumaran, and S. Gupta, "Real-time Tracking and Identification for Multi-Camera Surveillance System," The Journal of the Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 10, No. 1, pp. 16-22, Feb. 2018.
DOI: https://doi.org/10.7236/IJIBC.2018.10.1.3.

[8] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," 2016 IEEE International Conference on Image Processing (ICIP), August 2016.
DOI: https://doi.org/10.1109/ICIP.2016.7533003

[9] L. Leal-Taixe, C. Canton-Ferrer, and K. Schindler, "Learning by Tracking: Siamese CNN for Robust Target Association," In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 33 - 40, 2016.

[10] N. Wojke, A. Bewley and D. Paulus, "Simple online and realtime tracking with a deep association metric," Proceedings International Conference on Image Processing, ICIP, 2017 Sep 17 - 20, 2018.
DOI: https://doi.org/10.1109/ICIP.2017.8296962

[11] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," In Proceedings of the IEEE International Conference on Computer Vision, pp. 4705 - 4713, 2015.

[12] N. Le, A. Heili, and J. M. Odobez, "Long-termtime-sensitive costs for CRF-based tracking by detection," In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016.

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," In CVPR, pp. 779 - 788, 2016.

[14] 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," In ECCV, pp. 740 - 755, 2014.
DOI: https://doi.org/10.1007/978-3-319-10602-1_48