Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
16/1 (2023), 1-13. DOI: http://dx.doi.org/10.21609/jiki.v16i1.1073

Improved Mask R-CNN And Cosine Similarity Using RGBD Segmentation For Occlusion Handling In Multi Object Tracking

Siti Hadiyan Pratiwi, Putri Shaniya, Grafika Jati, Wisnu Jatmiko

Faculty of Computer Science Universitas Indonesia, Depok, Indonesia

siti.hadiyan91@ui.ac.id, putri.ratriyani91@ui.ac.id, grafika.jati51@ui.ac.id, wisnuj@cs.ui.ac.id

Abstract

In this study, additional depth images were used to enrich the information in each image pixel. Segmentation, by its nature capable to process image up to pixel level. So, it can detect up to the smallest part of the object, even when it's overlapped with another object. By using segmentation, the main goal is to be able to maintain the tracking process longer when the object starts to be occluded until it is severely occluded right before it is completely disappeared. Object tracking based on object detection was developed by modifying the Mask R-CNN architecture to process RGBD images. The detection results feature extracted using HOG, and each of them got compared to the target objects. The comparison was using cosine similarity calculation, and the maximum value of the detected object would update the target object for the next frame. The evaluation of the model was using mAP calculation. Mask R-CNN RGBD late fusion had a higher value by 5% than Mask R-CNN RGB. It was 68,234% and 63,668%, respectively. Meanwhile, the tracking evaluation uses the traditional method of calculating the id switching during the tracking process. Out of 295 frames, the original Mask R-CNN method had ten switching ID times. On the other hand, the proposed method Mask R-CNN RGBD had much better tracking results with switching ids close to 0.

Keywords—Occlusion, RGBD, Mask R-CNN, Late fusion, Cosine similarity

1. Introduction

Object tracking was still one of the most exciting areas of computer vision. Three main steps in tracking objects were [1,2]: detecting objects, tracking objects from one frame to another, and analyzing the behavior of these objects. Object tracking has challenges that need to be faced [2], including missing information, image noise, complex object shapes, occlusion, illumination, and others. One of which was discussed in this study was occlusion. Occlusion was a condition where the main object was partially or completely blocked, for example, in Fig.1. The main goal of object tracking was to identify the object as being the same object as the initial target when only part of the object was visible or when the object disappears and reappears [1].

The occlusion that occurs can reduce the quality of the object tracking method [34], so many methods have been developed to overcome this problem. In 2007, Pan et al. proposed CAPOA (content-adaptive advanced occlusion analysis) occlusion handling algorithm. This algorithm initialized the target object by manually selecting the region of interest by utilizing the greyscale feature. The selection results were used as a template used for the template matching process in the next frame. Template matching was done using coordinate transformation to estimate the template and find areas similar to the estimated template. Then the template was updated using the new results, and this process was repeated for the following frames. This algorithm has good results to solve occlusion problems but fails for specific events, such as when complete occlusions occur or if the occluder has the same appearance as the target object.

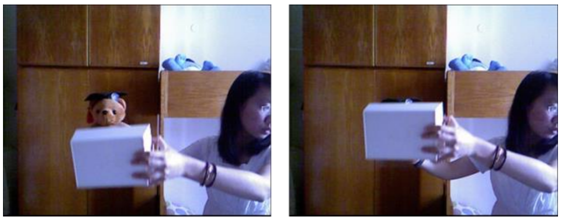

Fig. 1. Example of occlusion state of object teddy bear. Partial occlusion (left) and complete occlusion (right)

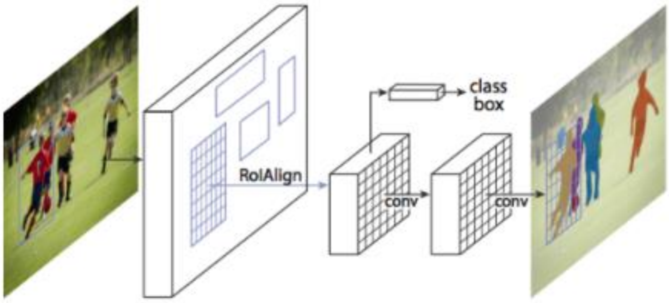

Fig. 2. Mask R-CNN arcitechture [13]

Occlusion is well known to be an unavoidable issue. In a survey conducted in 2014 by Lee et al, it is stated that problems with occlusion can occur when the occlusion state is still going on and when the occlusion has been completed. When an occlusion case occurs where there are two objects that overlap each other it will be difficult to determine the position of each object accurately, besides that there is also difficulty in determining the exact position of the object because some or all of the information from the object is lost. Also, when the occlusions state has been completed, it is also a challenging task to determine whether the object that occurs is the same object before it is missing or not. It would get more complicated if the object surrounded by other object with similar or even same characteristics. To overcome all of this problem, there are many methods specifically developed to handle occlusion. The survey explained one of the methods that can be used is depth analysis [36]. It is a method that will be able to determine the distance of the tracked object. So, when an object being occluded by another object, the depth information will be able to separate them based on their distance.

In 2015 Chen et.al. proposed a multi-instance segmentation method specifically to handle occlusion. The method was intended to find occluding area between two or more objects by parsing and categorizing them. Using categorized segmentation hypotheses by SDS [37], occluded area obtained by compare the value of two top scored categorized segmentation. If the proposal area is overlapped, it will be recorded as occluding region. Also, the classification used to get shape prediction of the object. So, by using occluded area, shape prediction, and the class were combined to construct the segmentation candidates.

Developments in terms of software and hardware continue to be carried out to overcome tracking problems. One of the most famous examples was the Kinect camera developed by Microsoft [1]. This camera has a depth sensor to produce images representing object distances by giving each pixel a different color. This image was known as a depth image. The depth image was less complex than RGB images, making them more resistant to image noise such as lighting, camouflage, etc. Despite their simplicity, depth images can differentiate objects at different distances. Generally, depth images were used with RGB images, namely RGBD images. The information obtained from the depth image can enrich the RGB image so that the object difference in each pixel can be more significant [17].

The presence of a depth image triggered the establishment of benchmarks for RGBD data, one of them was Princeton Tracking Benchmark (PTB). With the formation of the dataset made in PTB, Song et al. [1] made a detection-based tracking method using a discriminative model. Feature extraction from RGBD images was carried out in 2 ways: using an RGBD HOG and a 3D point cloud. The two extraction results were then trained using SVM to get the classification results. This method produces a reasonably good tracking but still has a relatively high computational performance.

In 2014, Benou et al. also made object tracking using RGBD, which was focused on solving occlusion problems. This method was divided into three stages. First, detect occlusion using depth image information using the Gaussian Mixture Model (GMM). Second, tracking when occlusion occurs in 2 ways: (1) maintaining the target position using segmentation during partial occlusion, and (2) performing estimates to predict the future location of the target in complete occlusion. The final stage was recovery from occlusion. At this stage, the template matching process was carried out to find the target position again when it was freed from occlusion. This method could prove that the addition of depth images improved the tracker to handle occlusions. However, this method has the limitation of only operating properly for rigid objects, and if the occlude was an object much different from the target object.

This study tried to overcome the occlusion problem by applying depth images to perform detection-based object tracking using a recent detection method that was more efficient than Mask R-CNN [13]. Extending Faster R-CNN [12] does not only detect objects but also masks them and be able to separate each object as a single entity. Mask R-CNN was known as state-of-the-art in terms of image segmentation and instance segmentation. The segmentation can extract visible parts of the object target when partial occlusion happens [33].

However, the current Mask R-CNN algorithm is limited to only using an RGB image as an input. The proposed method will try to utilize this method on RGBD image. With modification in the layer input of Mask R-CNN, depth images were combined using early fusion [11] and late fusion. The results of object detection from the frame-n were used as the target object in the frame-(n+1). This process continued until the last frame of the available data. Then tracking results were determined by calculating the similarity of the detection results between two frames. With some improvements made to the Mask R-CNN, this method worked with RGBD images.

2. Literature Overview

Object Detection

In 2017, a novel model architecture Mask R-CNN was introduced by He et al. [13]. This method was a modification of Faster R-CNN[12]. Previously, Faster R-CNN only had two results, bounding box and class. It was not designed for the pixel-to-pixel result. Then it was modified by adding a third network called RoIAlign to perform instances segmentation. In the Mask R-CNN, segmentation was obtained by forming a mask against the detection results. RoIAlign could improve mask accuracy by 10% to 50%. Mask R-CNN can also beat the previous state-of-the-art instance segmentation method when performed on the COCO dataset. It also takes a short training time by running at about 200ms per Frame on GPU. The structure of the object can be shown in Fig.2.

The simple rectangular matrix represents a mask in each pixel. We were at a point 1 means that the corresponding pixel belongs to an object of a particular class, and 0 means that the pixel does not belong to an object. Mask prediction was made using FCN, which maintains spatial object layout. RoIAlign then does essential work in this pixel-to-pixel process.

Object Tracking

Moujahid et al. [21] develop visual object tracking via the local soft cosine similarity (Fig.3). This method was developed to perform object tracking. Developed based on the calculation of cosine similarity, the author then proposes a new local soft cosine similarity method. The object tracking process begins with initializing and forming a sample of candidate objects in the current frame. Furthermore, the sample formed was calculated for the similarity to the target sample, and the best result was selected. Then, the selected candidate was updated to be the target sample for the next frame. This process was repeated until it reached the last frame.

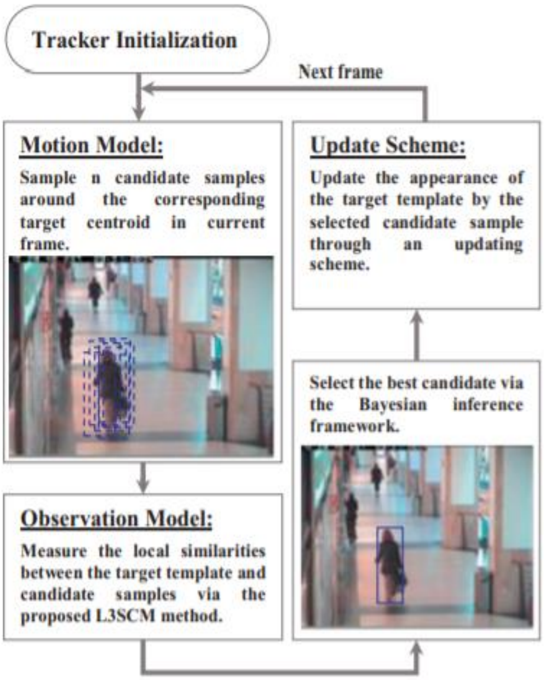
Tracker Initialization
Next frame
Motion Model:
Sample n candidate samples around the corresponding target centroid in current frame.
Update Scheme:
Update the appearance of the target template by the selected candidate sample through an updating scheme.
Select the best candidate via the Bayesian inference framework.
Observation Model:
Measure the local similarities between the target template and candidate samples via the proposed L3SCM method.

Fig. 3. Object tracking method [6]

RGBD Image Fusion

A late fusion process for the Retina-Net method using the Resnet-101 backbone was carried out. It worked by duplicating the architecture where each network was used to process RGB and depth images in parallel [22]. The output of the FPN RGB and FPN depth were then combined to obtain the features of the combination of the two. The architectural form of the fusion result in this study can be seen in Fig.4.

3. Methodology

The proposed method (Fig.5) consists of 2 parts: the detection process and the tracking process. In the detection process, RGB and depth images were combined by adding the feature extraction results of the two images to obtain the detection results on the frame.

Furthermore, with the results obtained, object tracking in the next frame was carried out by calculating the similarity value using the cosine similarity calculation. The object with the highest similarity value determined the tracking result. The experiment was conducted to see how effective the depth image could improve the tracking, especially if occlusion involved in the process.

A. Dataset

The dataset used in this study comes from the Princeton Tracking Benchmark (PTB) [1], which explicitly provides a dataset for RGBD video. The dataset was provided in the form of paired image slices between RGB images and depth images (Fig.4). An experiment was carried out for one of the available sequences: the teddy bear.

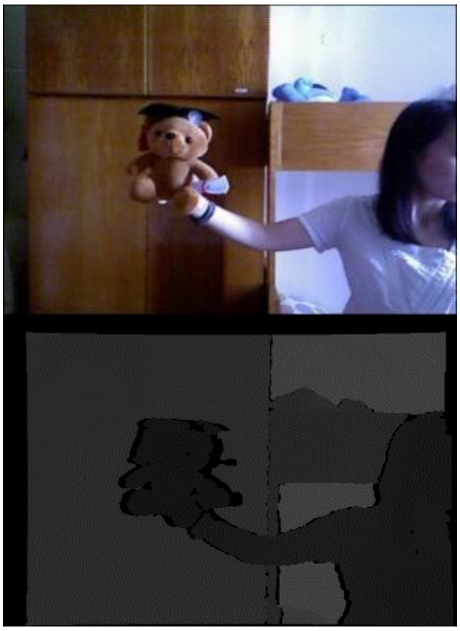

Fig. 4. Paired RGBD image: RGB(Above) and depth (Below)

The sequence teddy bear has the following specifications: (1) has 597 images consisting of 297 pairs of RGBD images, and (2) has a speed of 30 fps. annotations were done manually so that three classes were formed in this sequence, namely: teddy bear, box, and person.

B. RGBD Late Fusion

The depth image only has one channel, which cannot be input directly to Mask R-CNN because its first layer was designed to input RGB Images with three channels. For this reason, it was necessary to modify the input layer [20] so that depth image information can be included. First, the input layer for RGB images were duplicated. Then, the configuration was set to the input layer by changing the image channel from 3 to 1. The architecture can be seen in Fig. 8.

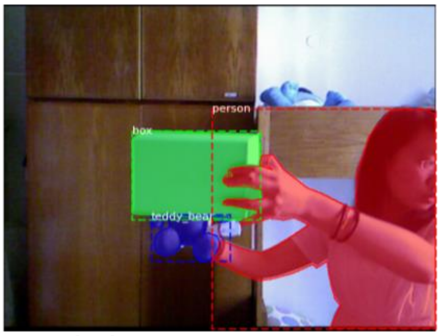

Fig. 5. Groundtruth preview

The late fusion method works by dividing the network in parallel for each RGB and Depth input into an identical architecture. Furthermore, the results from each network were combined to form a feature map [17]. The merging process carried out in this study uses the addition method where the results of the RGB image process were added to the results of the Depth image process.

By referring to the late fusion process that has been carried out by Brown et al. [22], a similar implementation was carried out using Resnet-50. The merging process was carried out using the addition method, where the results from each FPN that was at the same level would be added to each other. As explained in the previous sub-chapter, architecture was formed for each RGB (3 channel) and depth (1 channel) input before the fusion process was carried out. Then the results of each FPN in the two architectures were added before the feature map was formed by ROI pooling. The architecture formed can be seen in Fig.9.

C. Cosine Similarity

Cosine similarity was used to calculate the degree of similarity between two objects. In calculating cosine similarity, the first thing to do was do a scalar multiplication between the query and the document, then add it up. After that, do the multiplication between the length of the document and the query length squared, then calculate the square root. Furthermore, the result of the scalar multiplication was divided by the long multiplication in the document and query.

$$Similarity = cos(\theta) = \frac{A.B}{||A|| \cdot ||B||} \quad (1)$$

The search for the maximum value from the results of the cosine similarity calculation was done by comparing the values horizontally and vertically. An illustration of finding the maximum value can be seen in table 1.

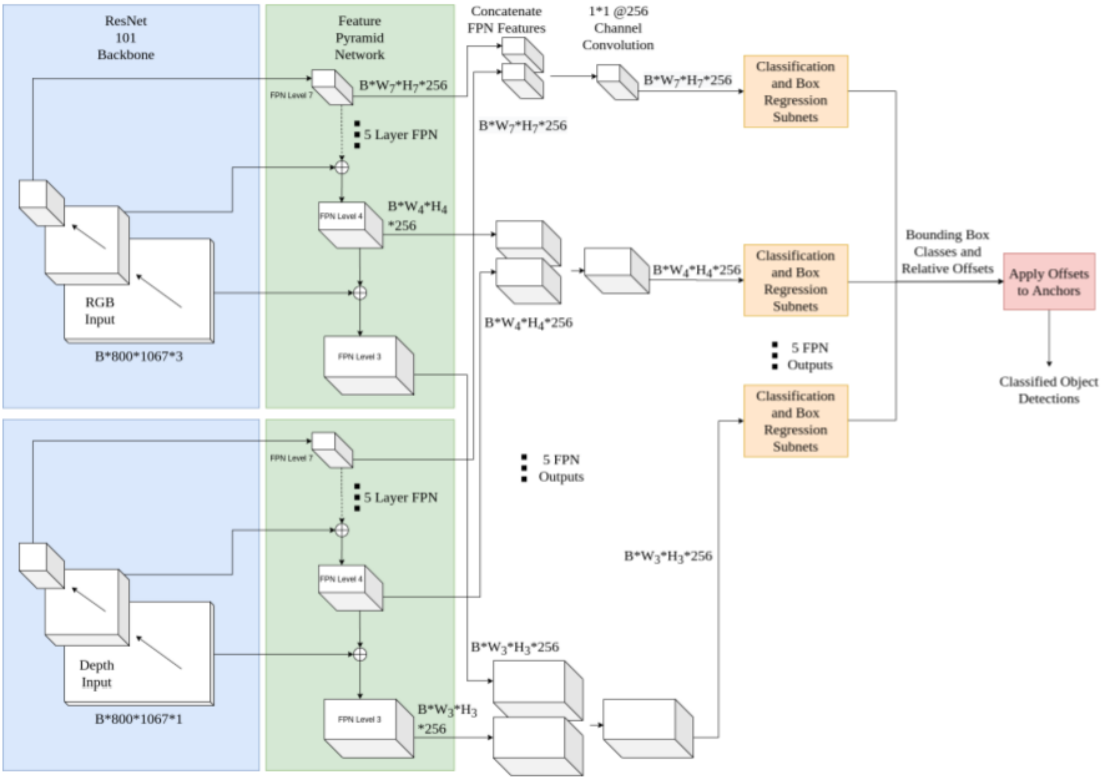

Fig. 6. RGBD image late fusion using Resnet-101

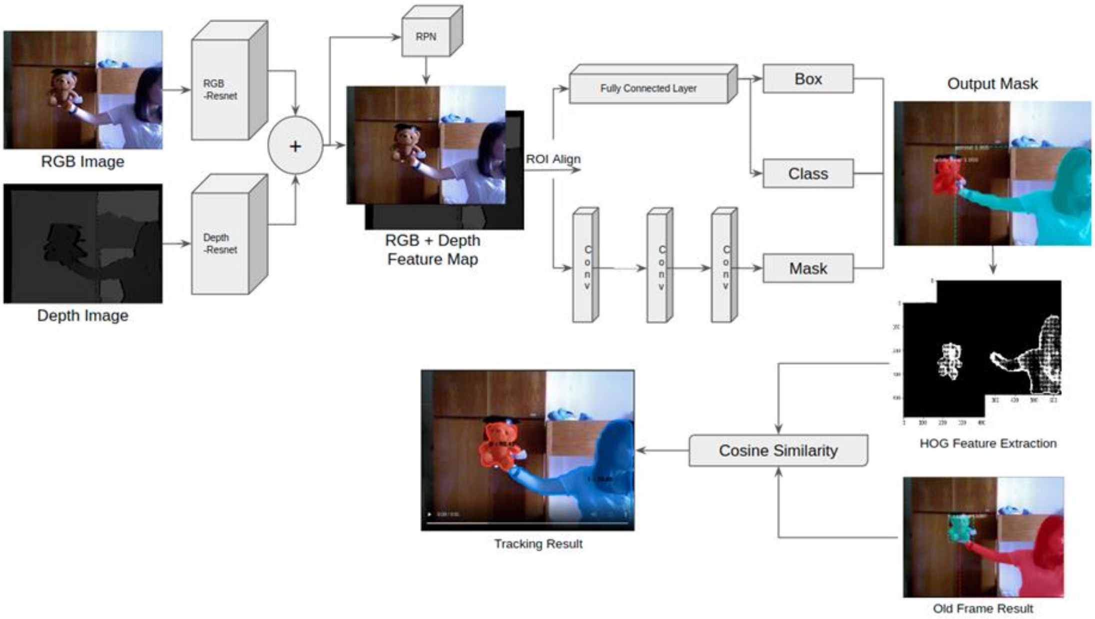

Fig. 7. Proposed method

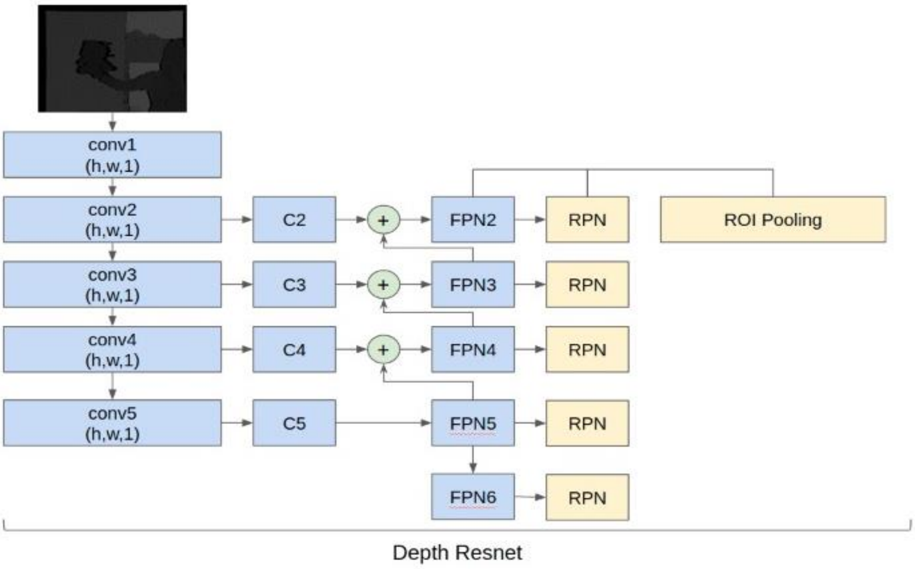

Fig. 7. Resnet-50 with 1 channel input for depth image

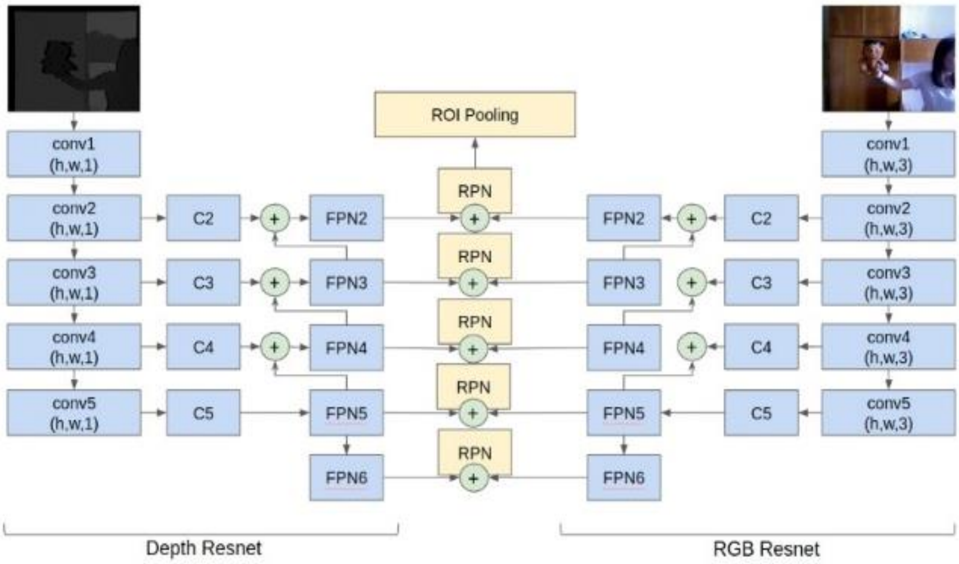

Fig. 8. RGBD image late fusion Resnet-50

Table 1. Cosine similarity value comparison

Frame	Id	a	b	c
1	0	**0.679636**	0.335477	
	1	0.322393	0.791559	
Frame	Id	a	b	c
51	0	0.394730	**0.722472**	0.392875
	1	0.587503	0.399975	0.489091
	2	0.495183	0.413560	0.671416

In frame 1, there are 2 objects that have been successfully detected, then the two objects are compared with each other. In frame 1, the largest cosine similarity value is 0.679636 which is found in object a. Therefore, object a will be given id 0. Then on frame 51 there are 3 objects detected. In the same way, a comparison of similarity values is carried out. Although the number of objects in the frame is more, the same comparison process will be carried out one by one. Both vertically and horizontally for id tracking 0 obtained object b. Therefore, object b will be assigned tracking id 0.

D. Evaluation

Object Detection

The model formed was evaluated by calculating the mean average precision (mAP). mAP is the average of the average precision values. This formula was used if the number of classes in the object detection process was more than 1. The mAP calculation formula can be seen in equation (1), where the value of N was the number of classes contained in the data.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \qquad (2)$$

The map calculation was carried out using the Intersection over Union (IOU) value. It is a value to see how close the predicted result is to ground truth, IOU value will be determined at the beginning of the evaluation, usually it was set to 0.5. then it will be increased gradually (0.7; 0.8; 0.9). The larger IoU value indicates that the prediction results were closer to the ground truth.

$$\text{IoU} = \frac{\text{Prediction} \cap \text{Groundtruth}}{\text{Prediction} \cup \text{Groundtruth}} \qquad (3)$$

Object tracking

A calculation was carried out on the id switching that occurs. Id switching calculation was used in the traditional method [30] by comparing generated id with the ground truth (Fig.10). Because the dataset used does not have a tracking ground truth, the id switching that occurs refers to the object id in the previous frame. If the id generated from the tracking process does not match the id in the previous frame, then it would be counted as a switched id. The less id switching occurs, the tracking results can be said to be better. A comparison was made with the novel Mask R-CNN that detects objects for RGB images only.

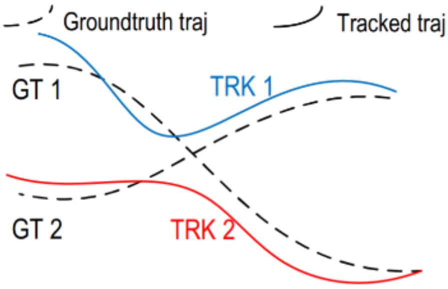

Fig. 9. Traditional id switching evaluation [30]

4. Result and Analysis

A. Detection Result

Testing of the model was done by training the RGBD teddy bear dataset, which consists of 295 pairs of images with 160 epochs. The evaluation was carried out using several IoU values. The purpose was to see which method was better when the area of overlap between the prediction results and the ground truth was getting bigger. The comparison of the training results was shown in Table 2.

Table 2. mAP comparison of Mask R-CNN and Mask R-CNN RGBD with late fusions

	mAP .7	mAP .8	mAP .9
RGB [1]	98,423%	96,209%	63,668%
Late Fusion	95,608%	93,806%	68,234%

The Table 2 shows that the detection results between the Mask R-CNN method using only RGB images have a better mAP value when the IoU value = 0.7. Using only RGB images can have an mAP value that reaches 98,423%, where this value was almost 3% higher than the late fusion method. Furthermore, when the IoU value was increased to 0.8, the method that uses only RGB images was still better and has a more significant difference in value with the late fusion method of about 3%

An interesting thing happened when the IoU value reached 0.9. In this condition, the mAP value of the late fusion method can exceed the previous method. With a value of 68.234%, the late fusion method increased by ±5%. IoU 0.9 can be used as the maximum value of overlapping objects formed because the value sought was more significant than that, and it was not get results. From these results, it can be concluded that the detection results using the late fusion method were closer to the ground truth.

B. Tracking Result

The original Mask R-CNN was the primary method for object tracking. The result of the instance segmentation was extracted, and the similarity with the target object would be calculated. Experiments were also conducted for the modified Mask R-CNN with the early fusion and late fusion methods for RGB-D images.

The results of tracking objects with changed ids were shown in a table according to the object class. The first id used as a reference for each class can be different for the two methods tested. The difference was due to the random assignment of ids according to object detection results. For example, the teddy bear object using the Mask R-CNN RGB method in the first frame had ID = 1 (Fig.12), while the Mask R-CNN RGBD method could had ID = 0 (Fig.14). It applies to all classes in this experiment.

A. Teddy Bear

Object tracking with Mask R-CNN RGB (Fig.12) gives a tracking id of 1 to the teddy bear object in the first frame, but this id can only be maintained until frame 39, which changes to tracking id 3. Changes in tracking id indicate that the tracker failed to identify the current object frame as an object in the previous frame. In addition, the id change with this method also occurs in 10 other

Frames. The list of frames that have an id change can be seen in Table 3.

Table 3. Id switching Mask R-CNN RGB for teddy bear

Frame #	*Tracking*_id rgb	class_id/0	Switch Count
39	3	1	1
41	1	1	2
42	3	1	3
66	1	1	4
68	3	1	5
204	2	1	FAT
	3	1	-
205	2	1	6
217	3	1	7
218	2	1	8
219	3	1	9
288	1	1	10

Furthermore, there was a slight improvement in the results of object tracking with the early fusion method. It can be seen from Table 4 there were a slight reduction in the id switching that occurs only seven times. However, the number of frames with false alarms increased slightly to 3. Detailed Id switching process was shown in Fig.13.

Table 4. Id switching Mask R-CNN RGBD early fusion for teddy bear

Frame #	Tracking_id rgbd early fusion	class_id/0	Switch Count
39	3	1	1
41	1	1	2
68	3	1	3
153	1	1	4
204	3	1	FAT
	1	1	-
205	3	1	5
217	1	1	FAT
	3	1	-
218	1	1	FAT
	3	1	-
219	1	1	6
289	2	1	7

On the other hand, object tracking with late fusion provides tracking id 0 on the teddy bear object in the first frame. In Fig.14 can be seen that it can identify the teddy bear as an object with tracking id 0 until the last frame.

Table 5. Id switching Mask R-CNN RGBD late fusion for teddy bear

Frame #	Tracking_id rgbd late fusion	class_id/0	Switch Count
217	3	1	FAT
	0	1	-
218	3	1	FAT
	0	1	-

For the teddy bear object, it can be concluded that the tracking method with the Mask R-CNN RGBD was better because it has fewer number of id switching (Table 5).

B. Box

In the class box, Mask R-CNN RGB had tracking id 2 when it first appears on frame 31 (Fig.15). However, this id can only be maintained for one frame until it finally changes to tracking id three on frame 32. The total number of id switched for the object box was five times (Table 6).

Table 6. Id switching Mask R-CNN RGB for Box

Frame #	Tracking_id rgb	class_id/0	Switch Count
32	3	3	1
33	2	3	2
89	1	3	3
90	2	3	4
116	1	3	5

There was a considerable improvement in the results of object tracking with the early fusion method (Fig.16). It can be seen that there was no id switching happened. However, the number of frames with false alarms increased by one frame (Table 7).

Table 7. Id switching Mask R-CNN RGBD early fusion for Box

Frame #	Tracking_id rgbd early fusion	class_id/0	Switch Count
32	2	3	-
	3	3	FAT

The tracking result was showed in Table 8 had a better consistency for late fusion (Fig.17) as the switching id only happens once in frame 292. It also marks that this method also performed better for object boxes.

Table 8. Id switching Mask R-CNN RGBD late fusion for Box

Frame #	Tracking_id rgbd late fusion	class_id/0	Switch Count
292	0	3	1

C. Person

The best tracking result for Mask R-CNN RGB was class for the person (Fig.18). It can maintain its id until nearing the end of the video at frame 289. It was shown in Table 9 that the total of id switches in this class was only two times, which appear in the next frame (frame 290).

Table 9. Id switching Mask R-CNN RGB for Person

Frame #	Tracking_id rgb	class_id/0	Switch Count
289	3	2	1
290	0	2	2
291	3	2	FAT
	0	2	-
292	3	2	FAT
	2	2	FAT
	0	2	-
293	1	2	FAT
	2	2	FAT
	3	2	FAT
	0	2	-

In the results of early fusion (Fig.19), there was no id switching happened. However, this method also has the same error with Mask R-CNN RGB. It happened in the same four frames: 291, 292, and 293. Detailed information was shown in Table 10.

Table 10. Id switching Mask R-CNN RGBD early fusion for Person

Frame #	Tracking_id rgb early fusion	class_id/0	Switch Count
290	2	2	FAT
	0	2	-
291	2	2	FAT
	0	2	-
292	2	2	FAT
	0	2	-
293	2	2	FAT
	0	2	-
	3	2	FAT

As for Mask R-CNN RGBD (Fig.20), the consistency of the id can be maintained longer by one frame in frame 290. Also, this was the only id switching that happened in the entire tracking for this object. Detailed information was shown in Table 11.

Table 11. Id switching Mask R-CNN RGBD late fusion for Person

Frame #	Tracking_id rgbd late fusion	class_id/0	Switch Count
290	2	2	1
291	1	2	FAT
	2	2	-
292	1	2	FAT
292	2	2	-
293	1	2	FAT
	2	2	-

Even if Mask R-CNN RGB performs the best in this class, only has two id switching, it slightly has worse results than Mask R-CNN RGBD, which only has one id switching. With this result, which performs better for all class, it can be concluded that Mask R-CNN RGBD perform better than Mask R-CNN RGB for object tracking.

D. Switch id summary

The comparison of id switching results for all objects can be seen in Table 12. Overall, object tracking using the Mask R-CNN RGBD has fewer switching ids for the three classes available.

Table 12. Id Switching Comparison

Object	Mask R-CNN RGB[3]	Mask R-CNN Early Fusion[11]	Mask R-CNN Late Fusion
Teddy Bear	10	7	0
Box	5	0	1
Person	1	0	0

Significant results can be seen on the teddy bear object where there were ten times ID switching if only RGB images were detected. Meanwhile, the tracking results can be perfect when the depth image was included because there was no id switching until the last frame was processed. It can be concluded that object tracking with additional depth images using the Mask R-CNN RGBD method was performed better to find the target object in each frame.

E. Further Analysis

In addition to id switching, another exciting thing that happened was the presence of a false alarm trajectory (FAT) in the tracking process (Table 13). FAT is a condition where a detection error happened in a frame (Fig.11). If a false alarm occurs in a frame, then there was a high probability that the event repeated itself at the same position in the next frame [8]. In this experiment, the FATs that occur were treated correctly, using the original Mask R-CNN method and the Mask R-CNN RGBD, by generating a new id for the false alarm object.

Table 13. False Alarm Trajectory

Object	Mask R-CNN RGB[3]	Mask R-CNN Early Fusion[11]	Mask R-CNN Late Fusion
Teddy Bear	1	3	2
Box	0	1	0
Person	4	4	4

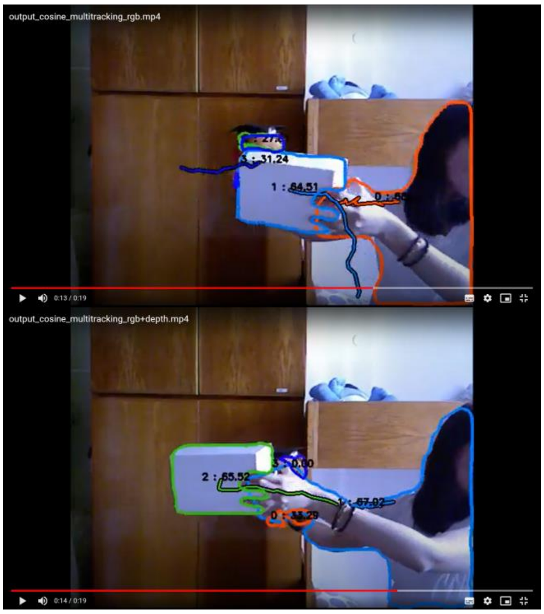

Fig. 10. Examples of False Alarm Trajectory

5. Conclusion

It can be concluded that adding depth images into the multi-object tracking process can improve the ability to retain object ids. The results obtained from the experiment show that object tracking using the R-CNN RGBD Mask has almost perfect results, marked by almost no id switching that occurs. An interesting event occurs when a detection error occurs and generates a false alarm (FAT). Although FAT can be appropriately treated in this study by generating a new id to the object, further research needs to be done on whether the false alarm can affect the tracking results. Of course, the appearance of a false alarm can also be a success rate for the object tracking method. The fewer FAT appears, the better the tracking results.

Besides the improvement, there are some drawbacks that occurred while inserting depth to the experiment. The architectural changes made this method unable to use available pre-trained datasets such as pre-trained COCO, also due to the limited pretrained dataset with RGBD images, this method also only limited to the object used in the experiment.

References

[1] Song, S., & Xiao, J. (2012). Tracking revisited using rgbd camera: Baseline and benchmark. arXiv preprint arXiv:1212.2823.

[2] Yilmaz, A., Javed, O., & Shah, M. (2006). Object Tracking: A survey. Acm computing surveys (CSUR), 38(4), 13-es.

[3] Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2147-2154).

[4] Fausett, Laurene. 1994. Fundamentals of Neural Networks Architectures, Algorithms and Applications. London: Prentice Hall, Inc.

[5] Gandi. R. (2018). R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. (Online) https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-objectdetection-algorithms-36d53571365e (9 Desember 2021)

[6] Gonzalez, R. C., Woods, R. E., & Masters, B. R. (2009). Digital image processing.

[7] Karpathy, A. (2018). Introduction to Convolutional Neural Networks

[8] Megawan, S., & Lestari, W. S. (2020). Deteksi Spoofing Wajah Menggunakan Faster R-CNN dengan Arsitektur Resnet50 pada Video. Jurnal Nasional Teknik Elektro dan Teknologi Informasi, 9(3), 261-267.

[9] Nurdiana, O., Jumadi, J., & Nursantika, D. (2016). Perbandingan metode Cosine similarity dengan metode Jaccard Similarity pada aplikasi pencarian terjemah Al-Qur'an dalam Bahasa Indonesia. Jurnal Online Informatika, 1(1), 59-63.

[10] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[11] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[12] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28, 91-99.

[13] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[14] Mai, X., Zhang, H., & Meng, M. Q. H. (2018, May). Faster R-CNN with classifier fusion for small fruit detection. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 7166-7172). IEEE.

[15] Zhu, Y., Urtasun, R., Salakhutdinov, R., & Fidler, S. (2015). segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4703-4711).

[16] Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012, October). Indoor segmentation and support inference from rgbd images. In European conference on computer vision (pp. 746-760). Springer, Berlin, Heidelberg.

[17] Zhou, T., Fan, D. P., Cheng, M. M., Shen, J., & Shao, L. (2021). RGB-D salient object detection: A survey. Computational Visual Media, 1-33.

[18] Zhang, H., Xu, H., Tian, X., Jiang, J., & Ma, J. (2021). Image fusion meets deep learning: A survey and perspective. Information Fusion, 76, 323-336.

[19] Singh, Aishwarya. (2019). Feature Descriptor: Hog Descriptor Tutorial. Retrieved July 16, 2020, from https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-imagesintroduction-hog-feature-descriptor/

[20] Zambounis, O., Fadri, F., Tonci, N., Margarita, G. (2018). does Depth matter ? RGB Instance Segmentation with Mask R-CNN.

[21] Moujahid, D., Elharrouss, O., & Tairi, H. (2018). Visual object Tracking via the local soft cosine similarity. Pattern Recognition Letters, 110, 79-85.

[22] Brown, J., & Sukkarieh, S. (2021). Dataset and Performance Comparison of Deep Learning Architectures for Plum Detection and Robotic Harvesting. arXiv preprint arXiv:2105.03832.

[23] D. Lahat, T. Adali, and C. Jutten. 2015. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. Proceedings of the IEEE.

[24] Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi. 2013. Multisensor data fusion: A review of the state-of-the-art. Inf. Fusion.

[25] Yohanandan, S. (2020, June 9). Map (mean average precision) might confuse you! Medium. Retrieved December 27, 2021, from https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2.

[26] F. (2019, December 6). Frame Rate. CCTVSG.NET. https://www.cctvsg.net/Frame-rate/.

[27] Tan, R. J. (2021, December 9). Breaking Down Mean Average Precision (mAP) - Towards Data Science. Medium. https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52.

[28] Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831.

[29] Wu, B., & Nevatia, R. (2006, June). Tracking of multiple, partially occluded humans based on static body part detection. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 1, pp. 951-958). IEEE.

[30] Zhang, L., Li, Y., & Nevatia, R. (2008, June). Global data association for multi-object tracking using network flows. In 2008 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

[31] Li, Y., Huang, C., & Nevatia, R. (2009, June). Learning to associate: Hybridboosted multi-target tracker for crowded scene. In 2009 IEEE conference on computer vision and pattern recognition (pp. 2953-2960). IEEE.

[32] Han, M., Sethi, A., Hua, W., & Gong, Y. (2004, October). A detection-based multiple object tracking method. In 2004 International Conference on Image Processing, 2004. ICIP'04. (Vol. 5, pp. 3065-3068). IEEE.

[33] Benou, A., Benou, I., & Hagage, R. (2014, December). Occlusion handling method for object tracking using RGB-D data. In *2014 IEEE 28th Convention of Electrical & Electronics Engineers in Israel (IEEEI)* (pp. 1-5). IEEE.

[34] Pan, J., & Hu, B. (2007, June). Robust occlusion handling in object tracking. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8). IEEE.

[35] Gentile, C., Camps, O., & Sznaier, M. (2004). Segmentation for robust tracking in the presence of severe occlusion. IEEE Transactions on Image Processing, 13(2), 166-178.

[36] Lee, B. Y., Liew, L. H., Cheah, W. S., & Wang, Y. C. (2014, February). Occlusion handling in videos object tracking: A survey. In IOP conference series: earth and environmental science (Vol. 18, No. 1, p. 012020). IOP Publishing.

[37] Chen, Y. T., Liu, X., & Yang, M. H. (2015). Multi-instance object segmentation with occlusion handling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3470-3478).

[38] Elgammal, A. E., & Davis, L. S. (2001, July). Probabilistic framework for segmenting people under occlusion. In Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001 (Vol. 2, pp. 145-152). IEEE.

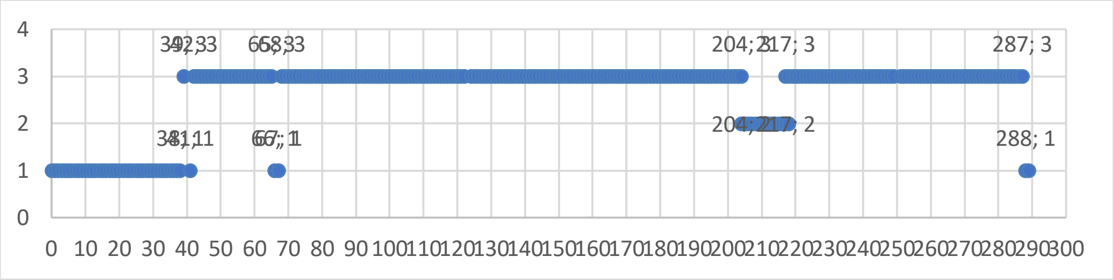

Fig. 11. Tracking id teddy bear using Mask R-CNN RGB

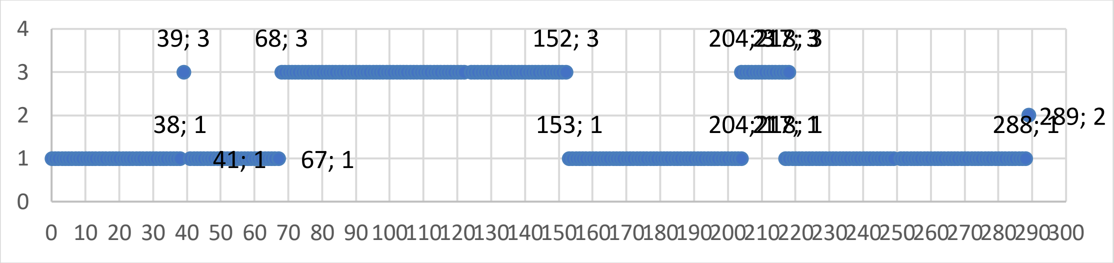

Fig. 12. Tracking id teddy bear using Mask R-CNN RGBD early fusion

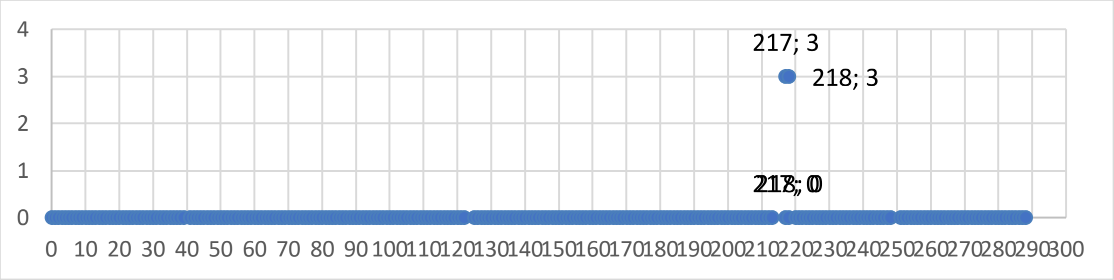

Fig. 13. Tracking id teddy bear using Mask R-CNN RGBD late fusion

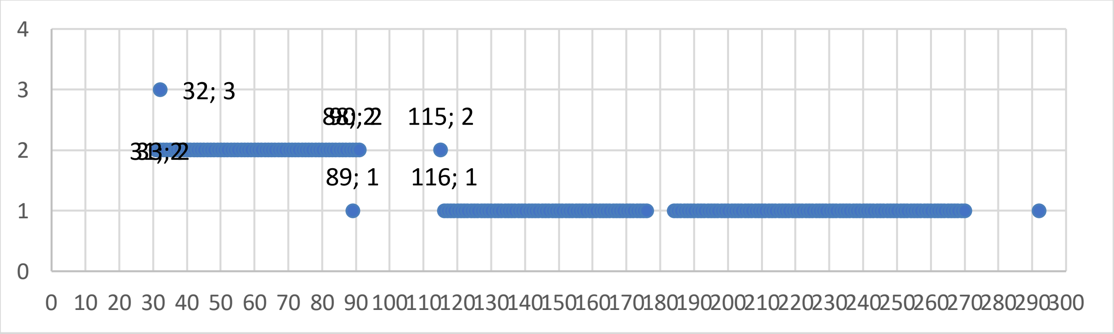

Fig. 14. Tracking id box using Mask R-CNN RGB

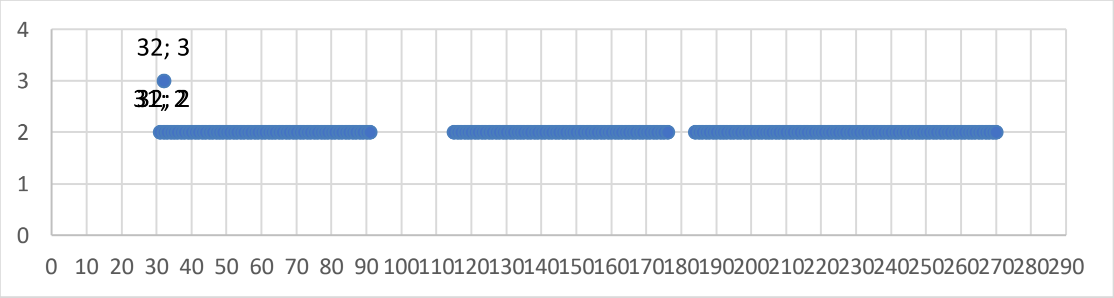

Fig. 15. Tracking id box using Mask R-CNN RGBD early fusion

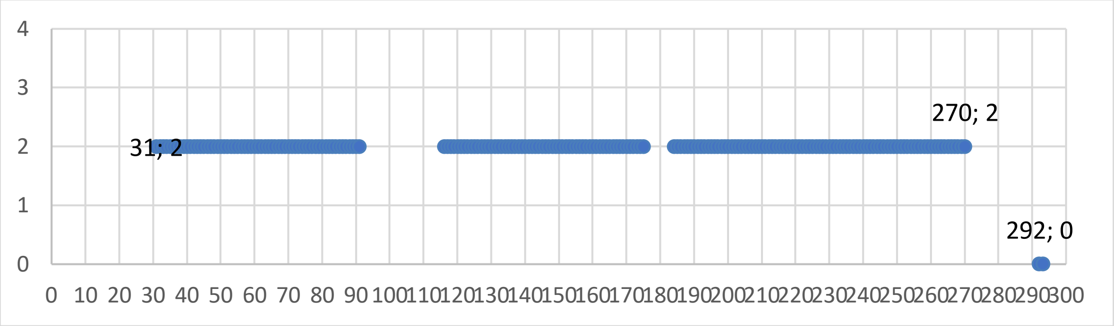

Fig. 16. Tracking id box using Mask R-CNN RGBD Late Fusion

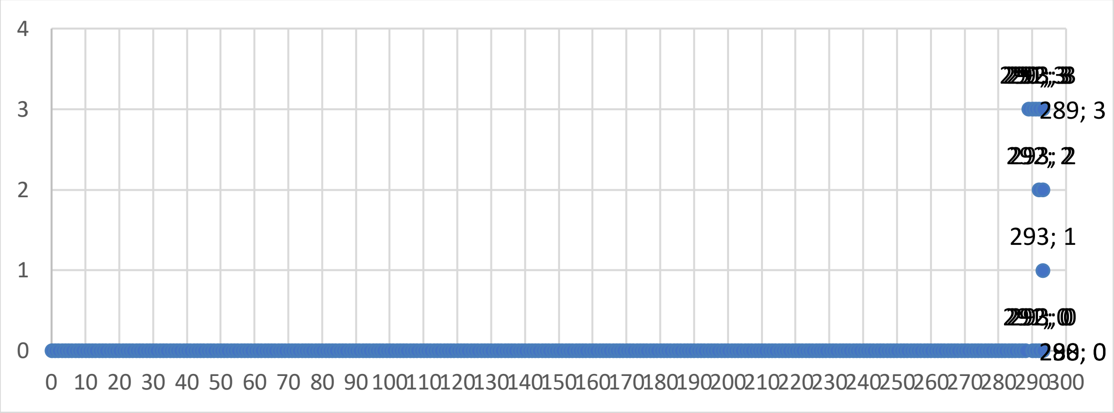

Fig. 17. Tracking id person using Mask R-CNN RGB

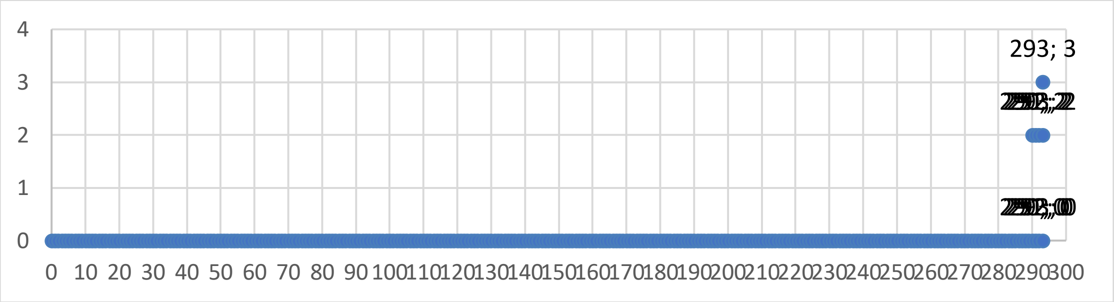

Fig. 18. Tracking id person using Mask R-CNN RGBD early fusion

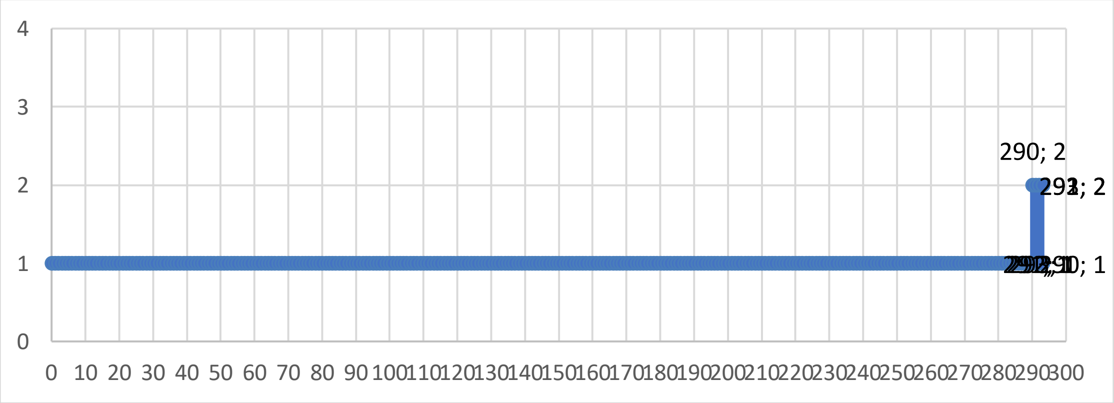

Fig. 19. Tracking id person using Mask R-CNN RGBD late fusion