CLASS AND ACTIVE-TIME FILTERING BASED ON DEEP-SORT FOR VISUAL OBJECT TRACKING

Khin Ohnmar Maung

Department of Computer Engineering and Information Technology, Mandalay Technological University,
Mandalay, Myanmar
khinohnmarmg@gmail.com

Theingi Myint

Department of Computer Engineering and Information Technology, Mandalay Technological University,
Mandalay, Myanmar
drtgim@gmail.com

Abstract

Visual object tracking plays an important role in many applications, especially for drone navigation, human-computer interaction, intelligent transportation and robotics. Deep Simple Online and Real-Time Tracking (Deep SORT) is a multi-objects tracking-by-detection algorithm with high tracking accuracy and speed. However, in the case of occlusion and re-entering into the camera's view, the tracking object's ID is changed. To overcome those problems, a class and active-time filtering (CAF) is incorporated into a Deep SORT algorithm in the novel proposed system. Utilizing information of class and active-time, CAF is able to reduce false positive tracks caused by unreliable tracking tends to serious ID switches, mainly for multiple objects with different classes. The novel tracking with the CAF algorithm is determined by two cases: whether the track need to be updated according to its own initial class and whether it exceeds a set of thresholds, so that the algorithm can correctly track the moving objects. As datasets, videos are individually captured in indoor and outdoor environments, and ten videos are used for evaluating performance. According to the experimental results with these videos, the changes in the tracking object's ID of the novel proposed system are less than that of the conventional YOLOv4 Deep SORT approach. Moreover, the evaluation matrices of mean average precision (mAP), Recall and Precision are calculated to prove better performance of the novel approach than the conventional approach.

***Keywords*: YOLO, Deep SORT, CAF, ID switches, Re-entering.**

1. Introduction

Visual object tracking is to assign and keep a unique ID to each object in camera view while tracking the objects is an essential branch of computer vision tasks. Computer vision technology is crucial in various practical applications such as transportation, healthcare, surveillance system, agriculture, and video and sports, among others.

To follow the visual objects effectively, the tracking system must be able to reliably track the objects within the surrounding environment. However, several challenges may occur in tracking system such as illumination, occlusion and increasing the IDs while the tracking objects are re-entering the camera's view. Tracking systems have been broadly studied in numerous published works, and different techniques have been proposed to solve some of these challenges.

The main purpose of this system is to improve the YOLOv4 Deep SORT with CAF approach which could obtain lower identity switch changing and better tracking performance. This paper is organized as follows: Section 2 reviews related works, Section 3 presents the conventional YOLOv4 Deep tracker, Section 4 explains the proposed system in detail, Section 5 shows the experimental results, and Section 6 concludes the manuscript.

2. Related Works

The emergence of deep learning-based object detectors has indeed led to the introduction of novel tracking-by-detection approaches in traditional multi-object tracking (MOT) problems. Specifically, tracking-by-detection algorithms consist of two parts: object detection and data association. Detection results are given by an object detection algorithm, usually in the form of bounding box coordinates, in every frame while a data association algorithm determines whether the newly detected object can be associated with the estimated position of existing tracks. Therefore, taking the benefit of object location knowledge, this approach generates an association model

DOI : 10.21817/indjcse/2023/v14i5/231405017

that would be able to associate objects over time. In this section, tracking approaches with data association will be studied.

Tracking with only spatial data association is considered as the baseline of tracking-by-detection approach, where the input of tracker is the output of detector. In multi-object tracking (MOT), Kalman Filter (KF) is wisely used to estimate the location of the tracked object from last frame. The KF is an algorithm that is able to use measurements from detections and previous states of tracks that contain uncertainty to estimate the current states. To improve the association of objects over time, the KF algorithm is employed as a motion model in recent works.

Bewley et al. [6] proposed Simple Online and RealTime Tracking (SORT), which combines the Kalman Filter (KF) for state estimation and the Hungarian algorithm for data association to associate KF predictions with new object detections in multi-object tracking. It primarily relies on handcrafted features and does not directly integrate deep neural networks for detection or representation. Although this algorithm is robust with a distinguishable appearance of people, when there is full occlusion between multiple people, the algorithm tended to fail by switching ID or assign a new ID to the object after the occlusion.

Wojke et al. [8] proposed an improvement of SORT, the Deep-SORT, by incorporating a deep neural network for feature extraction and similarity metric computation. The deep neural network is used to encode appearance features from object detections, and these features are then used to compute similarity scores for data association. By integrating deep features and the deep association metric, Deep-SORT takes advantage of the rich representation power of deep neural networks, leading to improved object recognition and tracking accuracy. It is particularly effective in handling appearance variations, occlusions, and crowded scenes.

There are recent works which also employ Deep-SORT, as a base-line tracking algorithm. Han Wu et al. [20] proposed SOFT-YM, which combines the YOLOv4-tiny with motion prediction, using Deep SORT as baseline tracker. YOLOv4-tiny is used to improve the tracking accuracy and speed of the model, and a motion prediction strategy is to predict the location of lost objects by effectively reducing the number of ID switches and tracklet segments. In various complex scenes of the MOT-16 dataset, their algorithm achieved high tracking accuracy for both dark and bright objects, and accurately obtained the relatively complete tracklets of most objects. Recently, Dang et al. [14] presented an improved version that was intended to reduce the identity switches where the Dlib tracker is inserted into the YOLOv3 Deep-SORT architecture. When the YOLO cannot detect any object in the frame, the Deep SORT cannot track that missed object. To overcome that issue, their architecture had a discriminative correlation filter to estimate the transitions of the objects, attaining lower identity switches and advanced operating speed compared to the conventional YOLOv3 Deep SORT approach. Furthermore, Hou et al. [11] proposed a Deep-SORT tracking algorithm with the extension of low confidence track filter (DS_LCF). A self-generated UA-DETRAC vehicle re-identification dataset is used to train the convolutional neural network of Deep-SORT for data association. In their approach, the tracks with low average detection confidence in their initial several frames will be deleted. With this extension to Deep-SORT, false positive tracks generated by Deep SORT can be significantly reduced. Moreover, with the growth of deep learning-based Siamese networks in the object tracking community, Jin et al. [18] enhanced the performance of the Deep-SORT on extracting object feature with a Siamese architecture. In addition, optical flow is introduced in the motion module to improve the object association accuracy. According to the MOT-16 datasets [7], their algorithm can reduce the false-negative targets and has a better performance in accuracy. However, identity switches of tracking are more frequent due to the deformation of objects and severe occlusions.

There are also other tracking algorithms to improve the tracking robustness by deep feature learning. He et al. [22] proposed the graph matching - continuous tracker (GM-CT) algorithm that incorporates graph partitioning with deep feature learning. These approach constructed a graph through the extracted object appearance features and used it in the association step to model the relationship between measurements and tracks with higher accuracy. Next, Zhao et al. [1] combined the Correlation Filter tracker with CNN features to enable re-identification (ReID) when tracked objects are lost, and this approach is effective in tracking small objects with lower false negatives. Moreover, Han et al. [13] designed a scale estimation strategy for multi-channel feature fusion to characterize the appearance features of objects, proposing DSCF. In their proposed system, the color names (CNs), HOG, and gray features are fused for improving the tracking robustness. In addition, it estimated the scale of objects based on the correlation filter and then utilized the appearance feature to perform the data association. However, the DSCF cannot deal with multiple similar objects, such as vehicles and pedestrians. Next, Bae et al. [2] proposed confidence multi-object tracking (CMOT) using an ensemble learning algorithm to learn tracklet features online. The CMOT utilized the incremental linear discriminant analysis learning model to learn the appearance of objects and combine the similarity between the tracklet and the detection. As a result, the CMOT achieved a high tracking accuracy with a speed of 5 FPS on a 3.07 GHz CPU. Furthermore, Xiang et al. [4] regarded the generation and termination of tracklets as state transitions in the Markov decision process (MDP). They utilized the reinforcement learning algorithm to learn the correlation of data. According to the experimental results, MDP outperformed the state-of-the-art methods on the MOT-15 dataset [3].

A method similar to Deep-SORT was proposed by Chen et al., the MOTDT [9], which introduced a tracklet scoring mechanism to prevent tracking drifts in the long term. In the MOTDT, a deeply learned appearance representation was applied to enhance the identification capability. MOTDT uses a fully CNN-based scoring function for an optimal selection of candidates and Euclidean distances between extracted object appearance features for improve the association step. The experimental results showed that the MOTDT achieved state-of-the-art tracking performance on the MOT-16 datasets [7]. Based on the YOLOv3 and the MOTDT, Wang et al. [17] proposed joint detection and embedding (JDE), which performed object detection and feature extraction in a network. Therefore, the JDE significantly reduced the computational overhead. Experimenting it on the MOT-16 dataset [7], it was showed that the JDE became the first real-time MOT method, with a tracking speed of 30.3 FPS.

However, realizing a trade-off between the tracking accuracy and speed in the MOT task is still challenging. On the one hand, since the previous methods did not fully use the information of objects in the past video frames, the tracking performance was seriously affected by occlusions and re-entering. Thus, we propose a tracking system with more information of objects' classes, active- and occluded- times for tracking multiple-classes multiple-objects continuously, keeping their ID unchanged. Our proposed system utilizes YOLOv4 as the detector to improve the detection accuracy. Moreover, Deep SORT algorithm is used as a base-line tracker to track the moving objects. Specifically, CAF algorithm is added to the base-line accompanying the information of class, active- and occluded- times in the tracking stage. Through the CAF approach, the contribution of the proposed system is to reduce the number of ID switches and false-positive tracklet segments effectively.

3. Conventional YOLOv4 Deep SORT

The conventional tracking system YOLOv4 Deep SORT approach is presented in this section. The overall tracking system composed of object detection, feature extraction and tracking. An input video is captured by Pi camera that is mounted on the robot and it is divided into several frames and sent to the detector to detect the objects. These detected objects are tracked using the tracker.

3.1. *Object detection with YOLOv4*

For detecting the objects in the video frames, YOLOv4 detector is used in this system. YOLO [5] is an algorithm that utilizes neural networks for real-time object detection. It employs convolutional neural networks (CNN) to detect objects in real-time, which requires only a single forward propagation through a neural network to detect objects [19] and is used to predict various class probabilities and bounding boxes contemporaneously. The YOLO algorithm operates in three steps as follows:

- The input image is divided in S × S grid of cells, which each detects only one object by predicting bounding boxes for the object. According to the number of classes, the cell also predicts conditionals class probabilities.
- Each bounding box has five elements (x; y; w; h; confidence) where x and y are the coordinates of the bounding box, confidence is the probability that the box contains an object, and w is the width and h is the height of the bounding box.
- To obtain the class-specific confidence score for each bounding box, the confidence is multiplied with the conditional class probabilities.

YOLOv4 [15] outruns the existing methods significantly in both terms detection performance and superior speed. It consists of backbone, neck and head as shown in Fig. 1. Backbone, acts as a feature extractor, is a deep learning architecture. Basically, neck collects feature maps from different stages of the backbone, and head finds the region where the object might be presented but does not tell about which object is presented in that region, which is also known as the object detector. CSPDarknet53 acts as a Backbone, SPP (Spatial pyramid pooling) and PAN (Path Aggregation Network) as a Neck and YOLOv3 for the Head. In this system, the input image size is 416 × 416 according to the training image data size.

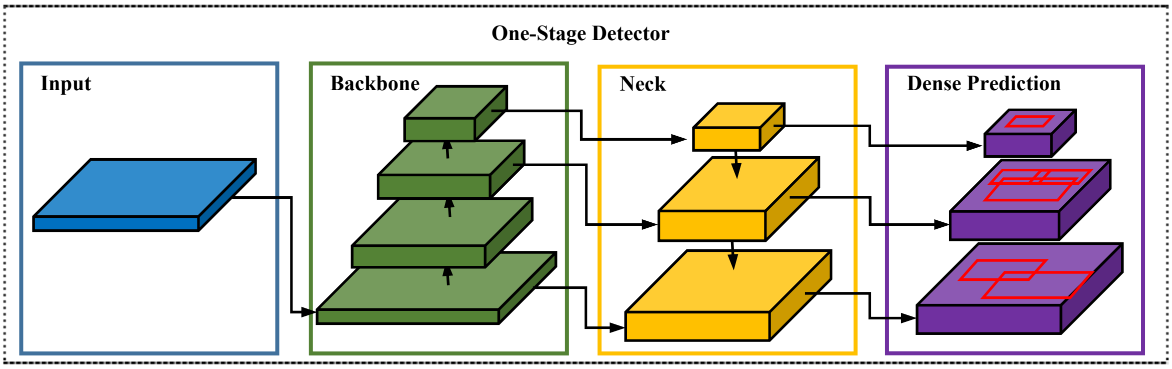


Fig. 1. YOLOv4 One-Stage Detector [15]

3.2. *Deep SORT algorithm*

Object tracking is the next process after object detection, which receives the detected objects with bounding boxes, assigns a unique identification (ID) for each of the initial detections, and also tracks the detected objects along the moved between frames. Bewley et al. [6] proposed Simple Online and Real-time Tracking (SORT), which is composed of a Kalman filter (KF) to estimate object states, and by the Hungarian algorithm to associate the KF predictions with new object detections. SORT is an extremely simple, effective and practical multi-target tracking algorithm. Using only IoU for matching, SORT is very fast, but the ID switch is still very large [12]. To handle the occlusion and ID switching problems and to improve the performance of SORT, apparent information is added which is the feature corresponding to the target [23]. Wojke et al. [8] proposed an improvement of SORT, the Deep-SORT, by including a cascading association step that uses CNN-based object appearance features. The process flow of Deep SORT that adds cascade matching and new trajectory confirmation on the basic of the SORT algorithm is as shown in Fig. 2.

The process of Deep SORT is

- The trajectory tracks are predicted by the Kalman filter
- To match the predicted trajectory tracks with the detections in the current frame (cascade matching and IOU matching) by using the Hungarian algorithm
- Updates the Kalman filter

The process of Deep SORT has several situations. These are matched tracks, unmatched detections and unmatched tracks [27]. Matched track means the detection and track are match. Ordinary continuous tracking targets belong to this situation and there are targets in the two frames before and after, which can be matched, so called 3 consecutive hits. Unmatched detection means the detection did not find a matching track; when a new target suddenly appears in the image, detection cannot find a matching that target in the previous track. Unmatched track means track did not find a matching detection. If the continuously tracked target exceeds the image area, track cannot match any current detection. In experiments, seven consecutive hits and 180 maximum ages are equally set-up for comparison with CAF.

With this extension, Deep SORT is better able to handle situations where the target is occluded for a long time, reducing the ID switch metric, but still cannot handle increasing IDs in re-entering cases. Although the original Deep SORT algorithm can track multiple objects single class, it cannot solve the ID switching problem completely. Therefore, we will propose the system including Class and Active-time Filtering module (CAF) which is extended to the baseline Deep SORT tracker to reduce false positive track that causes the ID switching problem.

4. Proposed class and active-time filtering

In order to acquire the sequences of image, the video frames are captured by Pi-camera that is mounted on the robot, which is used as a static in this system. In implementing the robot, the Raspberry Pi 3 B+, Raspberry Pi Camera B Rev 2.0, Ultrasonic Sensor and Arduino Uno are setting up on the robot [24]. The design of proposed system for object tracking (Sports Ball and Person) is illustrated in Fig. 3.

This proposed system uses YOLOv4 to detect the generic objects such as Sports Ball and Person using 80 object classes of COCO dataset [21] and Deep SORT algorithm which applied to track the object. Moreover, CAF is added to the baseline Deep SORT which is intended to reduce the false positive tracks.

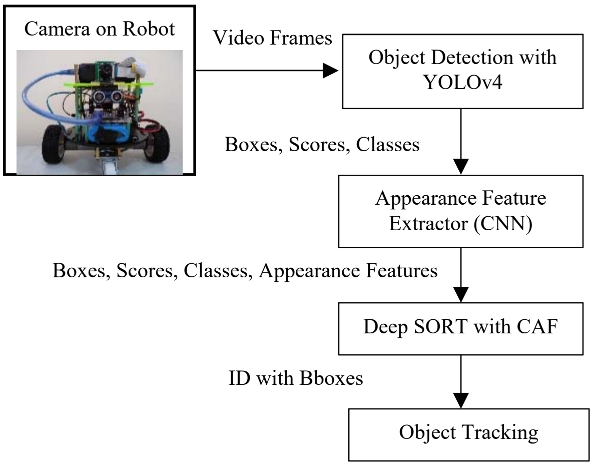

Camera on Robot
Video Frames
Object Detection with YOLOv4
Boxes, Scores, Classes
Appearance Feature Extractor (CNN)
Boxes, Scores, Classes, Appearance Features
Deep SORT with CAF
ID with Bboxes
Object Tracking

Fig. 2. Design of Proposed System

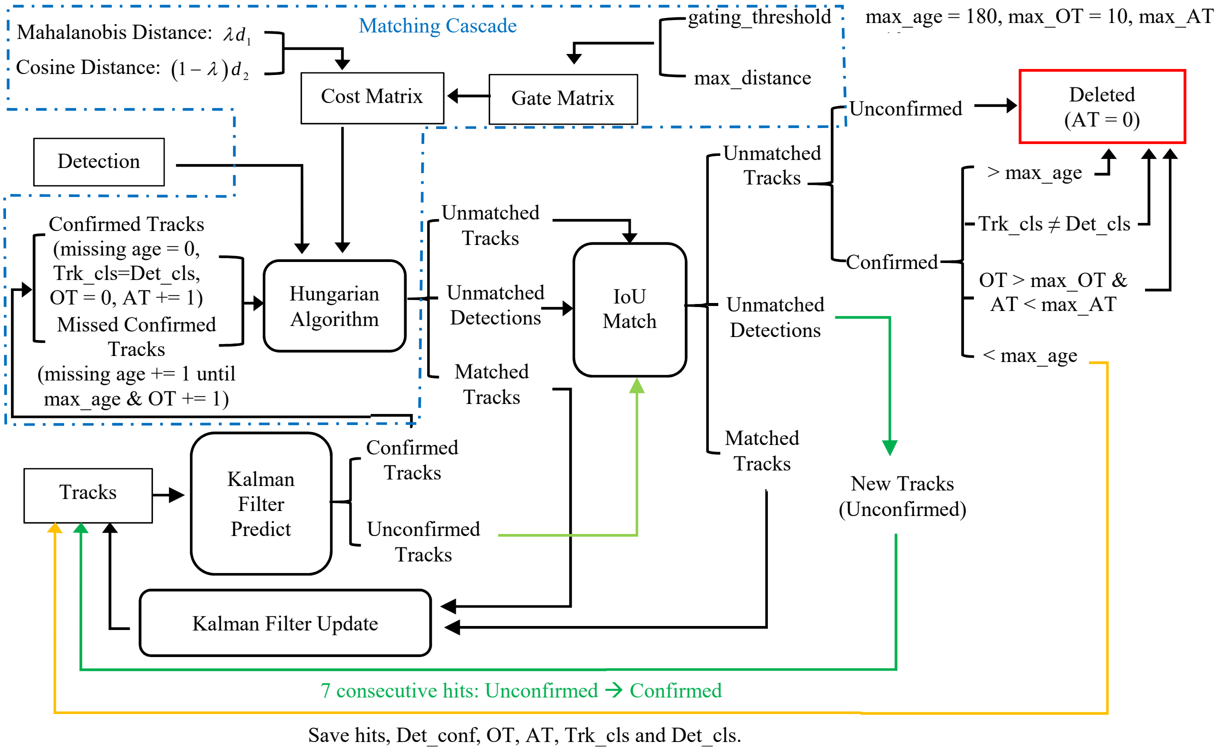


Fig. 3. Process Flow of Baseline Deep SORT Tracker with CAF

To handle the ID switching problem in the case of occlusion and re-entering into the camera view, the class and active-time filtering (CAF) algorithm is added to the baseline Deep SORT tracker. When an input video frame is processed, and object is detected by means of YOLO v4 architecture, the object is not tracked at first. Then, even after its detection matched in specific consecutive frames, it will be a track.

Process flow of baseline Deep SORT tracker with CAF is shown in Fig. 4. Firstly, objects have to be scanned for detection through the input video frames. In the detection generation, YOLOv4 is applied to produce the detections for each video frame. In initial states of detections, tracklets will be unmatched detections and there are no confirmed tracks. Only after the detection is matched in seven consecutive video frames, the next detection is counted as a track. After detection, appearance features have to be extracted in each bounding box of detected objects. In the appearance feature extraction, the appearance features of each detection are extracted through a convolutional neural network in DeepSORT. Then, for the tracklet location prediction, the predicted location of every tracklet in the next frame can be obtained by utilizing the Kalman filter. Combining these two information, association data of confirmed tracklets and detections are got in the matching cascade. In this process, the appearance feature similarity and location distance between the confirmed tracklets and detections are calculated, and the association results of the confirmed tracklets and detections are obtained through the Hungarian algorithm. After the matching cascade, there are three states: unmatched tracks, unmatched detections and matched tracks. For unmatched tracks and unmatched detections, the intersection-over-union (IOU) between the detection boxes and predicted bounding boxes of candidate tracklets are computed in the IOU matching, and the association results of the candidate tracklets and detections are obtained through the Hungarian algorithm. After the IOU matching, there are also three states: unmatched tracks, unmatched detections and matched tracks. Tracks will have to be processed depending on two cases: associated and unassociated detections. In associated detection cases, the classes of the tracklets are saved for later matching in re-entering cases, and their active- and occlusion- times are updated according to their occurrences or occlusions in video frames. Otherwise, new tracklets for unassociated detections are initialized. Therefore, two matched tracks of the matching cascade and IOU are used for updating track in Kalman Filter Update. And unmatched detections are candidate tracklets for new tracks. Then, unmatched tracks are classified as unconfirmed and confirmed. Unconfirmed unmatched tracks will be deleted, while confirmed unmatched tracks will be processed as two actions: deletion or consecutive tracking. They will be deleted if they are missed greater than 180 maximum ages, or if each track class is not equal to each own detect class, or if the track is either missed or not active track between ten frames. Otherwise, they will be consecutively tracked as long as less than of 180 maximum ages.

Detailed algorithm of CAF for tentative tracks (T_t) is as follows:

Algorithm: Class and Active-time Filtering

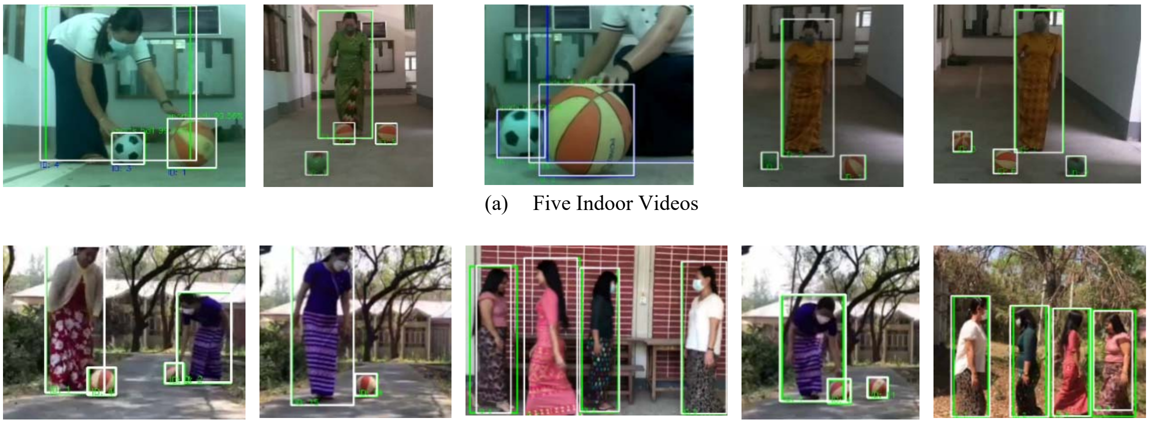
```
Input: Tentative tracks $T_t$; Tentative track class $t_{cls}$; Associated
        detection class $T_{det\_cls}$; Occlusion threshold $t_{OT}$; Active
        threshold $t_{AT}$; Active tracks $T_a$
Output: Confirmed tracks $T_c$ ; Deleted tracks $T_d$

Step 1: for sequential frames do
Step 2:     for $t \in T_t$ do
Step 3:         if $t$ is new in $T_t$ then
Step 4:             $OT = 0$
Step 5:             $AT = 0$
Step 6:             $t_{cls} = T_{det\_cls}$
Step 7:         if $t$ is missed in $T_a$ then
Step 8:             $OT = OT + 1$
Step 9:         else if $t$ is active in $T_a$ then
Step 10:            $OT = 0$
Step 11:            $AT = AT + 1$
Step 12:        if $t_{cls} \neq T_{det\_cls}$ or ($OT > t_{OT}$ and $AT < t_{AT}$) then
Step 13:            $T_d = T_d \cup t$ and $T_t = T_t \setminus t$
Step 14:            $AT = 0$
Step 15:        else
Step 16:            $T_c = T_c \cup t$ and $T_t = T_t \setminus t$
```

In the proposed system, tracks are filtered by their initial own detection classes and active occurring times as well as occlusion/missed times. Therefore, initial class that belongs to the track and how long it appeared in video frames have to be saved for each track. In class filtering, a tentative track (T_t) will be deleted if the tracking class is unequal to the initial detection class. And in active-time filtering, a tentative track (T_t) will be deleted if occluded time is greater than the predefined occlusion threshold (t_{OT}) and active time is less than the predefined active threshold (t_{AT}). Otherwise, the tentative track (T_t) can be updated to confirmed track (T_c).

In the YOLOv4 Deep SORT with CAF approach, a suitable set of parameters is as follows:

- The ROI size of the input frame is 640×480 and that size is sufficient for objects detection and tracking.
- Consecutive hits = 7: Hits represents the number of consecutive confirmations and is used when changing from an uncertain state to a certain state. Every time track is updated, hits will be + 1, if hits>n_init (default is 7), that is, the trajectory of seven consecutive frames has been matched, and then the uncertain state is changed to the definite state. The hits value is set to greater than 7 in this system, the tracks is missed and the accuracy of the tracking is desecrated.
- IoU_thres = 0.7: This threshold is used to track the objects by the positions of bounding box intersections. If this value is too small or too big, the object could not be tracked.
- Maximum cosine distance = 0.3.
- Maximum age = 180: Max_age represents the maximum number of consecutive misses before the track state is set to delete.
- Maximum occlusion time (OT) = 10, and Maximum active time (AT) = 10. (If the track is either missed or not active track between ten frames, that track will be deleted.)

In this object tracking system, consecutive hits are set to seven for confirming the tracks to be avoid of serious ID switches. In order to delete the track that is missed in the active track between ten frames and is not active in the active track between ten frames, occlusion time and active time are set to ten. For maintaining a track ID in an experimental video as long as possible, Max_age is set to 180, which will delete the track state after the maximum number of consecutive misses (180). And if the detection class is not equal to the tracking class, that frame will be discarded. Therefore, CAF can filter out a tracklet when detection class is not equal to the tracking class. After adding CAF to the baseline Deep SORT tracker, the proposed system can significantly reduce the ID switches of tracking objects as well as false positive IDs of tracks with unequal detection classes.



(a) Five Indoor Videos

(b) Five Outdoor Videos

Fig. 5. Experimental Videos

5. Experimental Results and Analyses

In this section, the details of our experiments are described. The performance of the proposed algorithm is evaluated on our own datasets, comparing between the conventional and proposed methods. Moreover, we compare our approach with state-of-the-art methods.

5.1. *Experimental settings and datasets*

In order to acquire the sequences of image, the video frames are captured by Pi-camera that is mounted on the robot, which is used as a static in this system. The HP i7 PAVILION with 8GB RAM, Single GPU (4G) Processor are used as hardware requirements and the software requirements such as the Python 3, Tensorflow 2 and Keras are applied in this proposed system. The proposed YOLOv4 Deep SORT with CAF architecture is tested with video files within the University's campus.

In this system, experiments are conducted to investigate the identity switches in indoor and outdoor conditions. Ten different experiments for indoor and outdoor environments are shown in Fig. 5 (a) and (b). To investigate different situations, five different videos in indoor and another five different videos in outdoor are recorded. Each video file is 30_second long and 720 frames. In all video files, there are tracking challenges concerning the illumination, occlusion, re-entering and identity switching problems. These video files including three objects (such as a person and two balls) or four objects (such as two persons and two balls) are experimented with both conventional and proposed Deep SORT. In order to identify the problems of re-entering and occlusion, all videos have multiple objects of two different classes which re-entered and/or occluded in video frames. Number of objects, and re-entering- and occlusion- times of the tracking objects in experimental videos are illustrated in Table 1.

	Re-entering Times	**Occlusion Times**	**Objects**
Indoor: Video 1	0	9	4
Indoor: Video 2	2	5	4
Indoor: Video 3	6	6	3
Indoor: Video 4	4	1	3
Indoor: Video 5	4	3	4
Outdoor: Video 1	5	6	3
Outdoor : Video 2	7	7	4
Outdoor: Video 3	4	2	4
Outdoor: Video 4	4	3	3
Outdoor: Video 5	2	7	3

Table 1. Objects, Re-entering and Occlusion

Max_age	30	70	120	180
ID Accuracy	44.4	57.7	69.3	78.4

Table 2. Influence of max_age on ID Accurascy

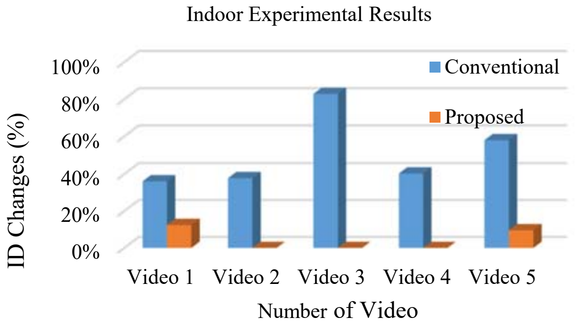
In case of removing a trajectory from association candidates, if the tracklet has not been associated with any detections in specified frames (max_age), the tracklet will be removed and not be matched any more. In order to deal with long-term tracklet associations better, max_age has to be assigned for experimental set-up. We compared several number of max_age as illustrated in Table 2. In our experiments, when the max_age is increased to 180 frames, which can successfully match the tracklets disappeared about six seconds ago, the ID accuracy is 78.4. Therefore, this max_age can deal with long-term tracklet asssociations. Furthermore, consecutive frames also play important role in tracking performance, especially breeding false positive error. If tracklets are in uncertain condition, the system shouldn't count as a new track. Therefore, we implement seven consecutive frames for a newly initialized tracklet. In both approaches, seven consecutive hits and 180 maximum ages are equally set-up for comparing between the conventional and the proposed approaches to see misidentification rate.

5.2. *Experiments of Identity switching experiment*

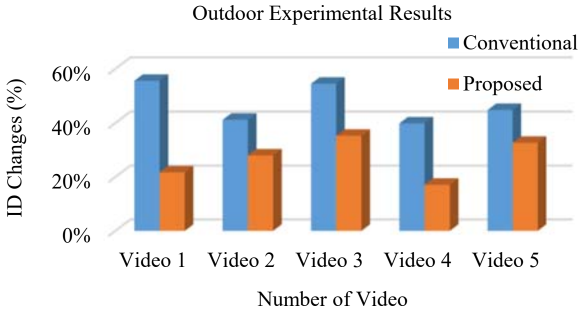
There are two types of ID switching problems in occlusion and re-entering cases: (i) formerly same class but different IDs, and (ii) different classes but formerly defined ID. In the experimental videos, there are two or three or four real objects but two different classes (person and sports ball in this study) entering and departing the video clips. In tracking processes, an object is specified as matched with its unique ID number when it is detected in seven consecutive frames of the video. In case of missing the object in many video frames (less than 180 frames), it will have to be re-identify with its ID number when it occurred again in a current frame. The entering object is labelled as their formerly defined ID number to know it as an existing object.

The experimental results are as shown in Fig. 6; the first five videos are indoor experiment results for ID switching cases and the rest five videos are outdoor experiment results. In the bar charts, the number in each cell is the number of identity switches divided by the total number of frames.

As shown in Fig. 6 (a), it can be clearly seen that the indoor experiment results of our approach are less ID changes than the conventional, absolutely in video 3 of indoor. In this video, person is partially appeared and this causes serious ID changes. In this experiment, the original Deep SORT approach has 83 % ID changes in 720 frames. Meanwhile, in our approach, there is no ID changes of three objects as shown in Fig. 6 (a). And in outdoor experiments, there are challenges of illumination changes, clothes colour and sun-drops which are badly effect on ID switches. In these cases, our approach can overcome these challenges with less ID changes than the conventional as shown in Fig. 6 (b) since the CAF architecture is mainly implemented to solve the problem of identity switches. According to the experimental results, we found that the overall average percentages of ID switch changes are 15.68% with CAF and 49.08% with the conventional approach. Therefore, the proposed system could overcome the problem of the changes of tracking object's ID about 68 %. According to the experimental results, the proposed system is less ID switching.



(a) Indoor Results



(b) Outdoor Results

Fig. 6. Experimental results for ID switching

	Conventional	Proposed CAF
Video 1	7	6
Video 2	7	0
Video 3	14	0
Video 4	3	0
Video 5	7	2
Average	7.6	1.6

Table 3. Number of Misidentification in Indoor

	Conventional	Proposed CAF
Video 1	15	7
Video 2	8	4
Video 3	7	11
Video 4	3	1
Video 5	4	3
Average	7.4	5.2

Table 4. Number of Misidentification in Outdoor

	Conventional (%)	Proposed CAF (%)
Video 1	63.68	71.15
Video 2	61.98	100.00
Video 3	62.08	93.06
Video 4	59.54	87.45
Video 5	41.88	90.38
Average	57.83	88.41

Table 5. ID Accuracy of Indoor Experiments

	Conventional (%)	Proposed CAF (%)
Video 1	63.43	68.19
Video 2	56.67	67.57
Video 3	45.28	60.24
Video 4	40.00	78.84
Video 5	55.05	67.18
Average	52.09	68.40

Table 6. ID Accuracy of Outdoor Experiments

Furthermore, number of misidentification are checked out for comparative purpose in tracking. Total number of ID error which isn’t equal to initial ID number are illustrated in Table 3 for indoor experiments and Table 4 for outdoor experiments. Normally, misidentification errors of indoor experiments are less than that of outdoor experiments since there are challenges of illumination changes, clothes colour and sun-drops in outdoor environments. In the experiments, average misidentification number of indoor experimental results for the proposed system is 1.6 while that of the conventional approach is 7.6 as illustrated in Table 3. And, average misidentification number of outdoor experimental results for the proposed system is 5.2 while that of conventional approach is 7.4 as illustrated in Table 4. As these results, the proposed system is less misidentification rate than the conventional Deep SORT.

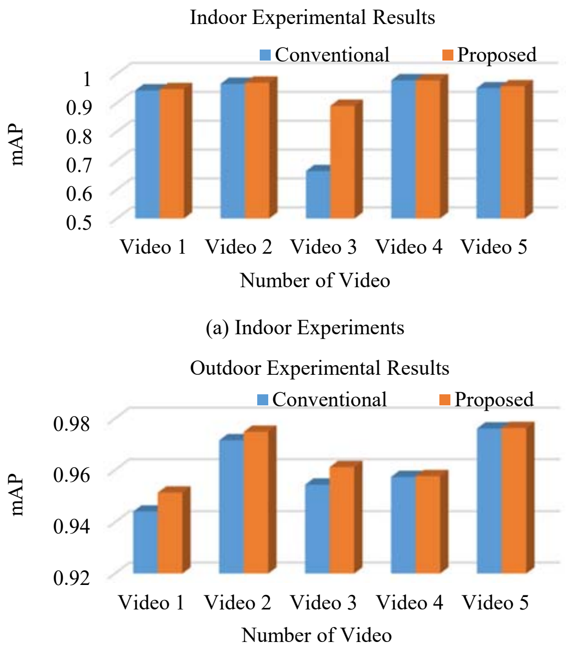
Then, ID accuracy for indoor and outdoor experiments are evaluated. In the indoor experiments, the overall ID accuracy of the proposed system is 88.41% while that of conventional approach is 57.83% as illustrated in Table 5. And in the outdoor experiments, that of outdoor experiments is 68.40% while that of conventional approach is 52.09% as illustrated in Table 6. Therefore, it can be concluded that the proposed system is obtained better overall ID accuracy as well as less misidentification rate.

5.3. *Experiments of recall, precision and mean average precision*

The experiments for recall, precision and mean average precision (mAP) of person and sport balls are conducted in both conventional approach and proposed approach. In the process of mAP, ground-truth bounding box is compared with the tracking one and then the score is returned. In calculating the mAP, Average Precision(AP) is found for each class and then average over a number of classes [16]. To find the mAP value, firstly the AP is calculated for each class and the mean value for all classes is estimated using the following Eq. (1) [25].

$$\mathrm{mAP}=\frac{1}{\mathrm{n}}\sum_{\mathrm{k}\text{-}1}^{\mathrm{n}}\mathrm{AP}_{\mathrm{k}} \qquad (1)$$

where, n = the number of classes, and AP_k = the average precision of class k



(a) Indoor Experiments

(b) Outdoor Experiments

Fig. 7. Experimental Results for mAP

Precision measures how accurate the system's predictions are. i.e. the percentage of its predictions that are correct. It calculates how many of the predictions that the system made were actually correct. Recall measures how well the system finds all the positives [26]. Precision and recall can be calculated by the following Eq. (2) and Eq. (3).

$$\text{Precision}=\frac{\text{True Positive}}{\text{True Positive+False Positive}} \quad (2)$$

$$\text{Recall}=\frac{\text{True Positive}}{\text{True Positive+False Negative}} \quad (3)$$

True positive means the system predicted as positive and was found to be correct. False positive means the system predicted as positive but was found to be incorrect and false negative means the system failed to predict an object that was there. In this system, precision, recall and mean average precision for two classes (person and sports ball) are calculated for evaluating the performance. The experimental results for mean average precision are shown in Fig. 7, and for recall and precision are illustrated in Table 7, 8, 9 and 10. In video 3 of indoor condition, one object class is not included the whole part and it appeared some part in many frames. Therefore, the conventional approach cannot solve this problem and causes serious ID switching, but the proposed approach can be with better mAP (0.8896), Recall (0.4735) and Precision (0.6637) as shown in Fig. 7 (a) and illustrated in Table 7 and 9. The other four videos in indoor condition have better results of mAP, Recall and Precision in the proposed approach than the conventional approach as shown in Fig 7. (a) and illustrated in Table 6 and 8. In the outdoor condition with many tracking challenges, the proposed approach has also better results of mAP, Recall and Precision in all video files than the conventional approach as shown in Fig 7. (b) and illustrated in Table 8 and 10. Therefore, according to the experimental results, our approach is better performance in the evaluation matrices of mAP, Recall and Precision than the conventional approach.

Video	Conventional			Proposed CAF		
	Person	Sports Ball	Recall	Person	Sports Ball	Recall
Video 1	0.5024	0.4787	0.4906	0.5028	0.4844	0.4936
Video 2	0.4932	0.4903	0.4918	0.4934	0.4938	0.4936
Video 3	0.2729	0.4409	0.3569	0.4513	0.4956	0.4735
Video 4	0.4945	0.4979	0.4962	0.4977	0.4912	0.4945
Video 5	0.4932	0.4881	0.4907	0.4964	0.4910	0.4937
Average			0.46524			0.48978

Table 7. Experimental Results of ID Recall for Indoor Condition

Video	Conventional			Proposed CAF		
	Person	Sports Ball	Recall	Person	Sports Ball	Recall
Video 1	0.4817	0.4944	0.4881	0.4876	0.4953	0.4915
Video 2	0.4945	0.4928	0.4937	0.4949	0.4948	0.4949
Video 3	0.4804	0.4949	0.4877	0.4910	0.4982	0.4946
Video 4	0.4935	0.4842	0.4889	0.4944	0.4839	0.4892
Video 5	0.4976	0.4673	0.4825	0.4977	0.4952	0.4965
Average			0.48818			0.49334

Table 8. Experimental Results of ID Recall for Outdoor Condition

Video	Conventional			Proposed CAF		
	Person	Sports Ball	Precision	Person	Sports Ball	Precision
Video 1	0.9820	0.9041	0.9431	0.9823	0.9126	0.9475
Video 2	0.9702	0.9616	0.9659	0.9713	0.9679	0.9696
Video 3	0.4533	0.8740	0.6637	0.7984	0.9808	0.8896
Video 4	0.9814	0.9751	0.9783	0.9876	0.9686	0.9781
Video 5	0.9705	0.9305	0.9505	0.9784	0.9367	0.9576
Average			0.9003			0.94848

Table 9. Experimental Results of ID Precision for Indoor Condition

Video	Conventional			Proposed CAF		
	Person	Sports Ball	Precision	Person	Sports Ball	Precision
Video 1	0.9079	0.9804	0.9442	0.9195	0.9831	0.9513
Video 2	0.9755	0.9678	0.9717	0.9760	0.9739	0.9750
Video 3	0.9529	0.9558	0.9544	0.9652	0.9572	0.9612
Video 4	0.9839	0.9309	0.9574	0.9849	0.9304	0.9577
Video 5	0.9851	0.9671	0.9761	0.9855	0.9674	0.9765
Average			0.96076			0.96434

Table 10. Experimental Results of ID Precision Outdoor Condition

5.4. *Experiments of comparing with other state-off-the-art methods*

The proposed tracker was compared with other state-of-the-art methods on the corresponding datasets such as UA-DETRAC, MOT 16 and own dataset. However, former datasets are labelled for human or vehicle tracking only. To evaluate the tracking performance, the evaluation matrices are MOTA and MOTP.

The multiple object tracking accuracy (MOTA) comprehensively considers three types of tracking errors such as FN, FP and IDs. FN is the number of false negatives representing the ground-truth objects that are not detected by the algorithm, while FP is the number of false positives representing tracked objects that are not related to any ground-truth object. IDs indicates the number of ID switches for all objects. It is defined as:

$$\mathrm{MOTA}=1-\frac{\sum_t(\mathrm{FN_t}+\mathrm{FP_t}+\mathrm{IDs_t})}{\sum_t \mathrm{GT_t}} \quad (4)$$

where, *t* is the index of the video frame, and *GT* is the number of ground-truth objects in all video sequences.

Then, the tracking precision is evaluated through the multiple object tracking precision (MOTP), which is defined as:

$$\mathrm{MOTP}=\frac{\sum_{t,i} \mathrm{d_{t,i}}}{\sum_t \mathrm{c_t}} \quad (5)$$

where, c_t is the number of the objects tracked correctly in frame *t*, and $d_{t,i}$ indicates the bounding box overlap of the *i*-th successfully tracked object with the ground-truth object in frame *t*. In Table 11, the upper arrow ↑ means that larger value of this metric shows better performance, and the down arrow ↓ means that smaller value of this metric shows better performance. As illustrated in Table 11, mostly state-of-the-art methods are multi-object tracking but only-one-class. Our approach is for tracking multiple objects with multiple classes. Therefore, this table shows that our method gets better performance in MOTA and MOTP (78.4% and 95.6 %, respectively) than the conventional approach. The results demonstrate the effectiveness of our CAF filtering and the importance of object classes in tracking objects.

Methods	Detector	Dataset	Class	MOTA↑	MOTP↑	FP↓	FN↓	IDs↓
DS_LCF	Mask R-CNN	UA-DETRAC	Vehicle	30 %	36.3%	20263.4	166384.9	388.8
GMT_CT	Tracktor	MOT 16	Person	66.2 %	-	6355	54560	701
SORT-YM	YOLO v4-tiny	MOT 16	Person	63.4%	81.4%	-	21,439	707
Deep SORT	Faster R-CNN	MOT 16	Person	61.4%	79.1%	-	56,668	781
Deep SORT	YOLO v4	Own Dataset	Person , Sports Ball	54.96%	93.1%	2976	12239	75
Deep SORT + CAF	YOLO v4	Own Dataset	Person , Sports Ball	78.4%	95.6%	1374	4001	34

Table 11. Comparison with State-of-the-Art Methods

6. Conclusion

This paper is implemented a real-time object tracking using YOLO detector and Deep SORT along with class and active-time filtering. Detection in YOLOv4 is able to solve for illumination cases. For long-term occlusions and ID switching in tracking process, the proposed tracking system can re-identify the objects by setting maximum-ages of track to 180, and including information of object's class, active- and occluded- times. Due to CAF, our system can delete unreliable tracks with unequal detection class. Experimental results have demonstrated that the proposed system can significantly reduce ID switches and false-positive IDs of tracks. According to the experimental results, the evaluation matrices of proposed approach is better than the conventional approach. We will extend the tracking system using hybrid approach combining Deep SORT CAF with LCF in our next research. The proposed system will be tested on mobile robot in real-time to track the selected target moving object as future research.

Conflict of interest: "All authors declare no conflicts of interest in this paper".

References

[1] D. Zhao, H.Fu, L. Xiao, T. Wu and B. Dai, "Multi-object Tracking with Correlation Filter for Autonomous Vehicle", Sensors (Basel) 2018, Vol. 18, No. 7, 2004.

[2] S. H. Bae and K. J. Yoon, "Robust Online multi-object Tracking Based on Tracklet Confidence and Online Discriminative Appearance Learning", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, pp. 1218–1225, 2014.

[3] L. Leal-Taixe, A. Milan, I.Reid, S. Roth, K. Schindler, MOTChallenge 2015, "Towards a Benchmark for Multi-Target Tracking", arXiv 2015, arXiv:1504-01942, 2015.

[4] Y. Xiang, A. Alahi, S. Savarese, "Learning to Track: Online Multi-Object Tracking by Decision Making", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santiago, Chile, pp. 4705–4713, 2015.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016.

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple Online and Real-Time Tracking", 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464-3468, 2016.

[7] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, K. Schindler, Mot16, "A Benchmark for Multi-Object Tracking", arXiv 2016, arXiv:1603.00831, 2016.

[8] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Real-time Tracking with a Deep Association Metric", IEEE International Conference on Image Processing (ICIP), IEEE 2017, pp. 3645–3649, 2017.

[9] L. Chen, H. Ai, Z. Zhuang, C. Shang, "Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification", Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, pp. 1-6, 2018.

[10] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement", Computer Science, arXiv: 1804. 02767, http://arxiv.org/abs/1804.02 767, 2018.

[11] X. Hou, Y. Wang and L. P. Chau, "Vehicle Tracking using Deep SORT with Low Confidence Track Filtering", Proc. of 16th IEEE International Conf. on Advanced Video and Signal Based Surveillance (AVSS), pp. 1-6, 2019.

[12] K. Chou, N. Kaothanthong and C. Jeenanunta, "Simple Online and Real-time Tracking with Feature Matching Enhancement for Re-identification after Occlusion", International Scientific Journal of Engineering and Technology (ISJET), Vol. 3, No. 2, pp. 34–41, 2019.

[13] X. W. Han, Y. W. Wang, Y. H. Xie, Y. Gao, Z. Lu, "Multi-channel Scale Adaptive Target Tracking Based on Double Correlation Filter", Chin. J. Sci. Instrum, Vol. 40, pp. 73–81, 2019.

[14] T. L. Dang, G. T. Nguyen and T. C. Cao, "Object Tracking using Improved Deep SORT YOLOV3 Architecture", ICIC International @2020, Vol. 14, No. 10, pp. 961-969, 2020.
[15] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", Computer Vision and Pattern Recognition (cs.CV); Image and Video Processing (eess.IV), 2020.
[16] S. Yohanandan, "What is Mean Average Precision (MAP) and How Does It Work", https:// xailient. com/ blog/what-is-mean-average-precision-and-how-does-it-work, 2020.
[17] Z. Wang, L. Zheng, Y. Liu, Y.Li, S. Wang, "Towards Real-Time Multi-Object Tracking" Proceedings of the European Conference on Computer Vision – ECCV 2020, Glasgow, UK, pp. 107–122, 2020.
[18] J. Jin, X. Li, X. Li, S. Guan, "Online Multi-Object Tracking with Siamese Network and Optical Flow", 2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC), Beijing, China, pp. 193-198, 2020.
[19] A. Kumar and S. Srivastava, "Object Detection System Based on Convolution Neural Networks Using Single Shot Multi-Box Detector", Procedia Computer Science 171 (2020), pp. 2610-2617, 2020.
[20] H. Wu, C. Du, Z. Ji, M. Gao and Z. He, "SORT-YM: An Algorithm of Multi-Object Tracking with YOLOv4-tiny and Motion Prediction", Electronics 2021, Vol. 10, No. 18, 2021.
[21] T. Petrosyan, https://opencv.org/introduction-to-the-coco-dataset, 2021.
[22] J. He, Z. Huang, N. Wang, Z. Zhang, "Learnable Graph Matching: Incorporating Graph Partitioning with Deep Feature Learning for Multiple Object Tracking", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, pp. 5295-5305, 2021.
[23] R. Pereira, G. Carvalho, L. Garrote and U. J. Nunes, "SORT and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics", Applied Sciences 2022, Vol. 12, No. 3, 2022.
[24] A. Jazmati, "Implementation of Mobile Robot Based on Raspberry-pi Part-1, https://www. hackster.io/ aula-jazmati 674b52, 2022.
[25] D. Shah, "Mean Average Precision (mAP) Explained: Everything You Need to Know", https:// www. v7labs.com/blog/mean-average-precision, 2022.
[26] P. Huilgol, "Precision vs. Recall – An Intuitive Guide for Every Machine Learning Person", https:// www. analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning, 2022.
[27] Programmer Sought, "Deep SORT Multi-Target Tracking Algorithm Code Analysis", https://www. programmersougth.com/article/17005126187.
[28] B. T. Tung, "SORT – Deep SORT: A Review of Object Tracking (part 1 and 2)", https:// www.viblo.asia/p/sort-deep-sort-mot-goc-nhin-ve-object-tracking-phan-2-djeZim78zWz.

Authors Profile

Khin Ohnmar Maung, is a PhD candidate at Mandalay Technological University, Mandalay, Myanmar. She completed M.E in Computer Engineering and Information Technology in 2010 from Mandalay Technological University. She is working as a Lecture in the Department of Computer Engineering and Information Technology at Mandalay Technological University. Image processing, Object detection and tracking, Deep learning, and Artificial Intelligence are some of her research interests.

Theingi Myint, received Ph.D (IT) at Mandalay Technological University, Mandalay, Myanmar, in 2011, and Doctor of Engineering (Computer Science) at Kumamoto University, Japan, in 2020. Currently, she is working as a Professor in Department of Computer Engineering and Information Technology at Mandalay Technological University. Her research interest areas are Computing System, Reconfigurable Architecture, and Deep Learning.