Received: September 17, 2020. Revised: October 7, 2020.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhancement Performance of Multiple Objects Detection and Tracking for Real-time and Online Applications

Nuha H. Abdulghafoor[1] **Hadeel N. Abdullah[2]***

[1,2]*Electrical Engineering Department, University of Technology, Iraq.*
* Corresponding author's Email: 30002@uotechnology.edu.iq

Abstract: Multi-object detection and tracking systems represent one of the basic and important tasks of surveillance and video traffic systems. Recently. The proposed tracking algorithms focused on the detection mechanism. It showed significant improvement in performance in the field of computer vision. Though. It faced many challenges and problems, such as many blockages and segmentation of paths, in addition to the increasing number of identification keys and false-positive paths. In this work, an algorithm was proposed that integrates information on appearance and visibility features to improve the tracker's performance. It enables us to track multiple objects throughout the video and for a longer period of clogging and buffer a number of ID switches. An effective and accurate data set, tools, and metrics were also used to measure the efficiency of the proposed algorithm. The experimental results show the great improvement in the performance of the tracker, with high accuracy of more than 65%, which achieves competitive performance with the existing algorithms.

Keywords: Object detection, Object tracking, Deep learning, Kalman filter, Intersection over union.

1. Introduction

In recent decades, computer vision applications in general, object detection and tracking, in particular, have become one of the most important fields of research, and articles have been prepared for that. The use of Deep Learning (DL) paradigms and bypass neuron networks has achieved much of a shift or a breakthrough in the discovery of many methods of detection and tracking. It has the ability and efficiency to produce systems that work in real-time. The basic principle of these algorithms is to identify targets or objects and predict their location by preparing a list of squares to surround the moving object. The number and size of Bounding box {Bb) are varied with the number and dimensions of the objects discovered within the particular video scene. These algorithms were able to overcome some obstacles and challenges that reduce the accuracy and efficiency of the proposed approach. Which, in turn, may result in new barriers every time a new algorithm is proposed [1].

Any engineering problem or application consists of several steps. One of the most important of these steps is the temporal and spatial prediction, whether these applications are in the areas of navigation or guidance, as well as computer vision [2]. The Kalman filter (Kf) was discovered as its algorithm to solve such problems. The basic principle of this candidate is to take advantage of guesses, available accounts, and previous forecasts to get the best estimate of the current situation, taking into account the occurrence of errors from the system and as a result of the process itself. The DL approaches have been used in detection and tracking applications. It was used to track one or several objects at the same time, and this is done by training a data set that represents video clips of a specific target [3]. Many obstacles reduce the smoothness of tracking objects, the most important of which is the inclusion of objects within the serial video scene. Fig. 1 shows all the pedestrians were accurately detected, including the man in the back, while the next video frame was not discovered. Where identifying the object, it is in the following frameworks and distinguishing it as the

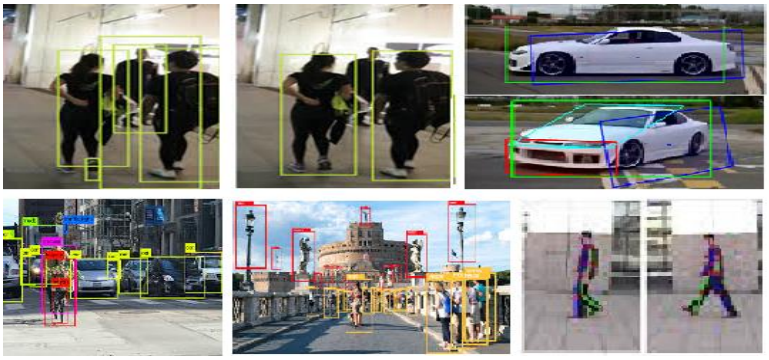


Figure. 1 Some examples of significant such as occlusions, multiple views for the same scenes, motion cameras, and finally, non-sequential datasets

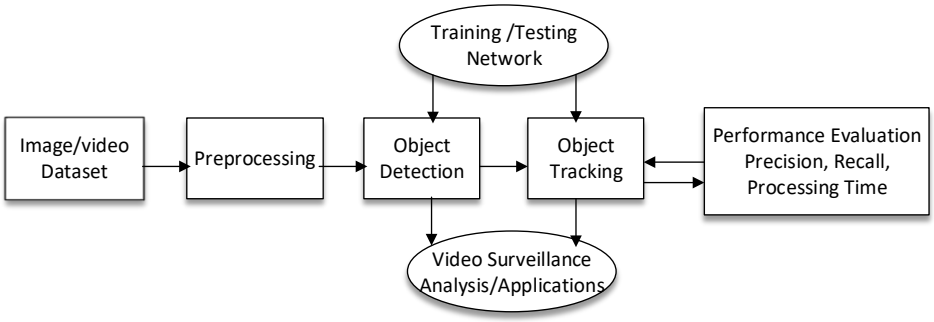


Figure. 2 Object detection and tracking framework

same object and linking its features to the correct path, which is called occlusion. Another problem is that the scene changes from different cameras of the same scene. These features used for tracking are various depending on the width change. So, the proposed algorithm is used to overcome these problems.

In such cases, the features used to track an object become very important as we need to ensure that they are constant for changes in views. We'll see how to overcome this effect. In the case of moving cameras, tracking the object becomes very difficult, as this leads to undesirable but unintended consequences. Trackers often depend on the features of the moving object, so the movement of the camera leads to a change in the object's size or distinction from the factual background, etc. This obstacle is very prominent in drone and navigation applications. Finally, useful data on training or testing any approach that follows is one of the most critical challenges that exist [4-5]. These data should be subsequent collapses to identify and track the object all the time.

Fig. 2 illustrates the structure of object detection and tracking through video sequencing [6]. More recently, deep learning approaches have emerged as one of the most effective methods in full applications and fields such as computer irrigation applications in general and in applications for detection and tracking of objects in particular. It was used to track one or several objects at the same time, and this is done by training a data set that represents video clips of a specific target. The object is tracked by detecting and positioning it via chained frames. The structure of the bypass neural networks allows the use of two consecutive frames to discover and track the object. The final result is to surround objects with surrounding boxes as well as no. other details such as spatial coordinates. The main contribution of the proposed algorithm as follows:

i. It is proposing an efficient algorithm to detect and track multiple objects and has the ability to address some of the challenges that prevent good results and robust performance—training and testing different Deep learning Networks models of detection stage.
ii. The proposed algorithm was implemented on two simulation devices, a Laptop Computer type MSI GV63 8SE. and Nvidia Jetson TX2 Platform.
iii. Calculation of essential detection and tracking performance factors such as the training and learning efficiency in addition to the impact of the training sample sizes.
iv. The data sets for training and testing the detection and tracking model. A comparison of the digital and visual results of the proposed algorithm with existing detection and tracking systems.

It is an approach from deep learning, and it is a method mainly for the detection object [2, 7]. It was developed by adding some modifications and the technique of repeated low light memory to become one of the effective ways to track the objects based on the temporal and spatial characteristics of the object.

The research document organized as follows: The second section explains similar works, such as research, articles, and others. The third section describes the basic principles and methodology of this article. The fourth section shows the structure of the proposed algorithm and all details of implementation. Section 5 presents laboratory experiments, simulation results, and analytical comparison for all the networks used. As for the sixth and final section, it clarifies the research findings and the future vision for developing the work.

2. Related work

Recently, many papers have been presented interested in the field of object discovery and the relationship between them, video segment analysis, and image processing. The methods based on deep learning showed superior performance, more robust tools, in addition to semantic features, good training, and improvement. Several detection algorithms have emerged from the structure of the bypass neural networks, the most famous of which is the Region of Convolution Neural Network (R-CNN) [8-9], which marks the beginning of the application of these networks in the detection of objects. After that, new systems developed or proposed that had an exceptional performance in detecting objects such as the Single Shot Detector (SSD) [10] and Yolo Only Look Once (YOLO) [11, 12] algorithms. Recently. The methods showed intermediate resolution results and accurate classification of all objects in the video files captured from fixed cameras. Bewley et al. proposed a Simple online and real-time tracking (SORT) [13] as a new algorithm for object detection and tracking. It shows good accuracy and a high apparent positive rate and a minor false positive. It has drawbacks in noise and non-standard vehicles. Then, Wojke et al. modified the previous Algorithm to Deep-SORT [14] is an excellent algorithm for detecting and tracking objects. It relies on deep learning models to discover and identify organisms. It also uses the KF and Hangareen Algorithm in the tracking of detected objects.

Huang et al. [15] proposed an algorithm based on an R-CNN detector. It has a low computing power system and weight constraints. It shows accurate detection but slow and computationally expensive. Then, fast-tracking with less accuracy. Bochinski et al. [16] proposed a new tracking method based on Intersection over Union (IOU) Algorithm. It shows a simple tracker with some drawbacks in multi-object tracking. Andrew G. et al. [17] suggested a compact multi-purpose real-time tracker based on the front-backlight (MobileNets) detector. It has accuracy tradeoffs and shows strong performance with large size and latency. Tijtgat, N. et al. [18] proposed an approach by using the drone's automatic learning algorithm to detect and track objects in real-time using the integrated camera or low-power computer system. They combine a small version of Faster-RCNN with Kernelized Correlation Filters (KCF) tracker to track one object on a drone. This article shows an accurate detection algorithm that is slow and mathematically expensive. While tracking algorithms are very fast, they are not careful, especially when there are fast-moving objects or jumps segmentation.

Raj, M. et al. [19] performed detection by using an SSD detector and made a comparison with other network models like Fast R-CNN and YOLO and knowing the strengths and weaknesses of each system. It shows good accuracy and a high apparent positive rate and a minor false positive. Blanco-Filgueira, B et al. [20] implemented a visual tracing of multiple objects based on deep real-time learning using the NVIDIA Jetson TX2 device. The results highlighted the effectiveness of the algorithm under real challenging scenarios in different environmental conditions, such as low light and high contrast in the tracking phase and not consider in the detection phase. Hossain, S et al. [21] proposed an application based on a DL technique, implemented on a computer system integrated with a drone, to track the objects in real-time. Experiments with the proposed algorithms demonstrated good efficacy using a multi-rotor aircraft. It counted a target of similar features. But it lost track of a counted feature and considered it as a new target. The results showed that these strategis would provide a straight forward methodology for detection and tracking algorithms. Whereas, these algorithms may fail in the face of some challenges found in real-time applications.

3. The basic principles

3.1 Kalman filter

The Kalman filter can be defined as an ideal guess algorithm that works closely with linear systems [22]. It is also assumed that all operations follow the Gaussian system. An example of this is the tracking of objects, whether they are pedestrians or vehicles,

which are linear motion systems, where they include different noise resulting from measurements, as well as from the same process, which applies to the Gaussian hypothes. The average state and covariance are what we want to guess. For example, an average is the coordinates of the bounding boxes. As for the variance, it is the lack of confidence in the surrounding boxes that have these coordinates [3].
In the prediction phase, the Kalman Filter predicts the received locations as shown in Fig. 3. In the update phase, it is a correction of the trajectory and an increase in the reliability of the values predicted by adjusting the uncertainty value. With time, the work of a Kalman candidate tends to be optimal and leads to a better convergence between guess and actual. Prediction is the stage of guessing the current location from its previous value. At the same time, the update phase is a process of correcting the last values to reach new measurements that will improve the filter's work [15].

3.2 Object tracking by detection

The rapid development of the deep learning approach has resulted in an expansion in its use in almost all areas of research. Where good networks are designed and built with a more robust structure and performance [5], several networks have been developed for object detection and classification [7]. After that, its performance was developed and improved for other tasks, such as discovering specific parts that have certain features, such as identifying the face and pedestrians, as well as classifying and distinguishing the object. The most famous of which is the Region of Convolution Neural Network (R-CNN) [8], as shown in Fig. 4, which marks the beginning of the application of these networks in the detection of objects.

4. The proposed algorithm

The primary stages of the proposed algorithm for detecting and tracking multiple objects, as shown in Fig. 5, which consists of several steps: capturing the frame or data set of the video clip, and then pre-processing and deep learning algorithms and their tasks in object discovery and classification. Then, frustrated funds account for the detected objects and finally track the discovered objects with the proposed tracker. This algorithm has shown significant improvement and solution to some challenges facing any proposed method and limits their efficiency. The proposed algorithm is characterized by the use of the Hungarian approach, which consists of several parameters listed as follows:

4.1 The object detection DL network models

Many essential articles emerged in which the bypass neural network models contributed to achieving a shift or improvement in the implementation of detection and tracking methods. To obtain the optimal performance for the proposed algorithm by choosing the efficient deep-learning detector model. So that an SSD, and YOLO object detector network were used, It is characterized by accuracy and efficiency to build systems that can work in real-time or directly. The main goal of this algorithm is to discover the objects inside the following diagram and locate them by enclosing them in the bounding box. The size and number of these boxes are related to the size and number of objects in the scene [10]. The loss function of the SSD model combined two components are:

i. **Confidence Loss (L_{conf})**: calculates the amount of confidence of the network in the objectivity of the Bb calculated using categorical entropy.
ii. **Location Loss (L_{Loc})**: calculates the magnitude of the difference between the position of the predicted Bb from the literal position of the box (which is calculated by using a second Norm).

The several equations can be used to develop a proposed algorithm, the grid model predicts the box updates, which are (x_p, y_p, w_p, h_p) [11]. Then the object displacement is made from the upper-left side of the image by the amount of *xi*, *yi* with prior knowledge of the height and width of the bounding box, as shown in Fig. 6. Therefore, the new dimensions are calculated, as shown below:

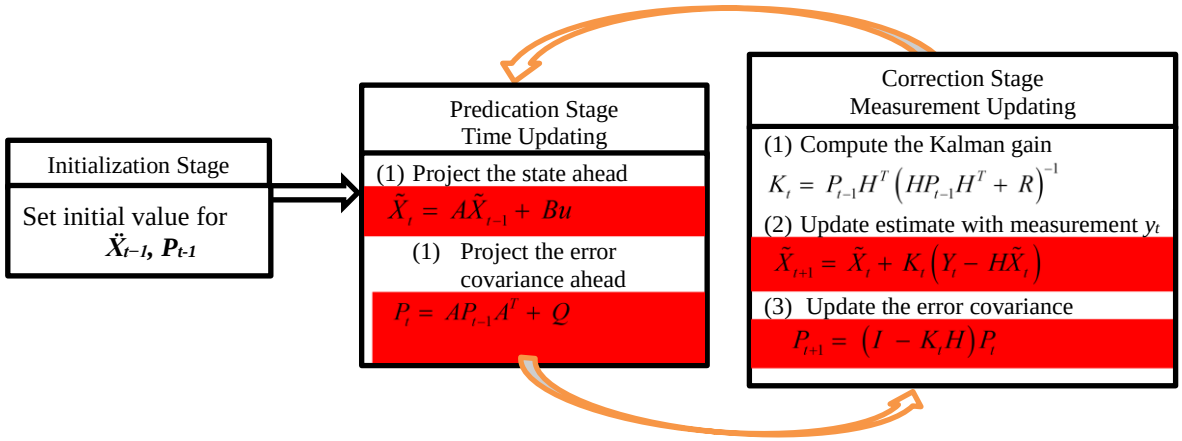
$$x_n = \sigma\left(x_p\right) + x_i \tag{1}$$

$$y_n = \sigma\left(y_p\right) + y_i \tag{2}$$

$$w_n = w_b e^{w_p} \tag{3}$$

$$h_n = h_b e^{h_p} \tag{4}$$

The second critical factor is the loss function, which represents a mixture of the value of the confidence in the predicted degrees and the accuracy of the location. Where the sum of classification and confidence losses represents one measure, as shown in the equation below, it is also an indication of the extent of correlation between the hypothetical square and the ground truth square of an item.



Where A - state transition matrix, B - coverts control input, Q - process noise covariance K- Kalman gain, X- measurement matrix, P - measurement error covariance and H - model matrices. The prediction of the next state X_{t+1} is done by integrating the actual measurement with the pre-estimate of the situation X_{t-1}.

Figure. 3 The block diagram of kalman filter algorithm

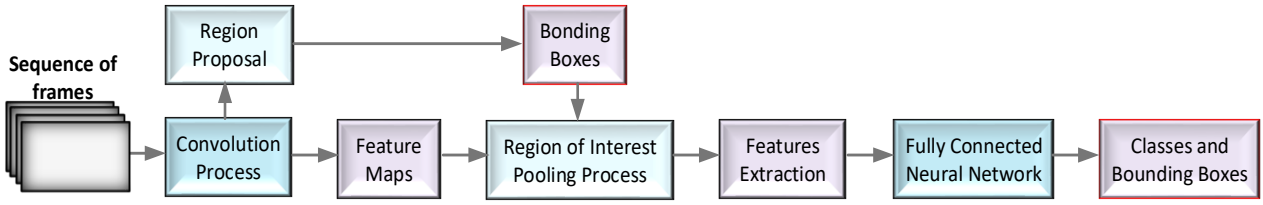


Figure. 4 The general block diagram of object detection by using deep learning

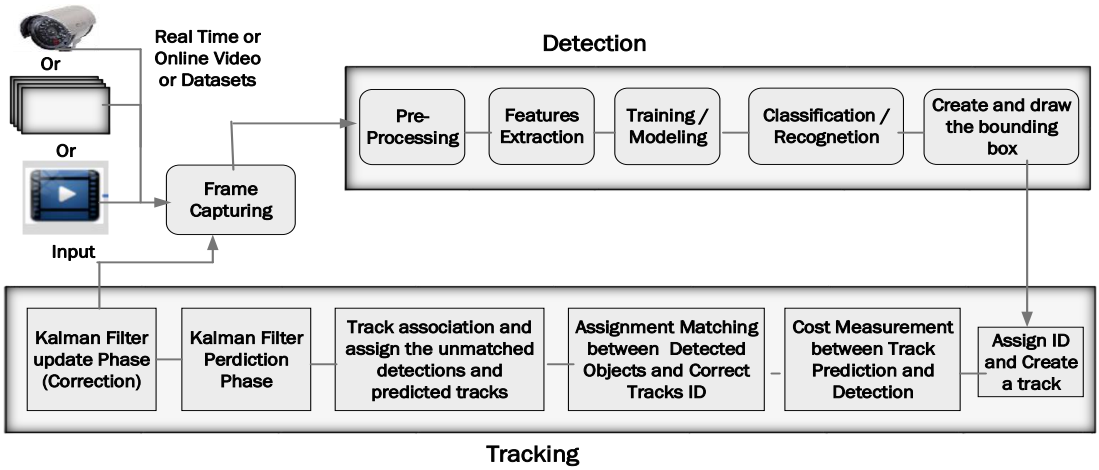
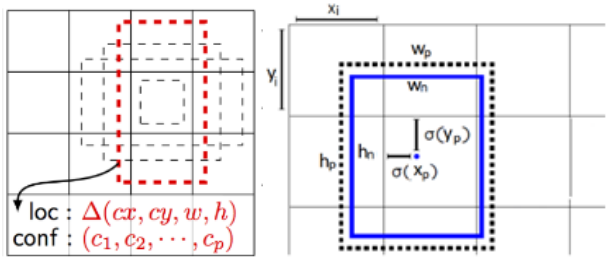


Figure. 5 The general block diagram of proposed algorithm



(a) (b)

Figure. 6 The bounding box transformation between the predication and prior evaluation for: (a) SSD model and (b) YOLO model.

$$L(x,y,bp,bg)=\frac{1}{N}(L_{com}(x,y)+\alpha L_{loc}(x,bp,bg)) \quad (5)$$

Where N equals the number of matching boxes, α represents the balance parameter between the two losses and compensation for the loss of the site. The main goal is to obtain optimal values for the above parameters that reduce the loss function to some extent, to reach predictions closer to the essential truth [12]. The second term represents the soft L1 loss between the shift value of prediction (bp) and ground truth (bg) boxes. x and y is the center of the box. Then

$$L_{loc}(x,bp,bg)=\sum_{i\in pos}\sum x_{ij}Smooth_{L1}(bp_i-bg_j) \quad (6)$$

Where

$$Smooth_{L1}(k)=\begin{cases}0.5k^2 & for\ k>0\\ |k|-0.5 & for\ k<0\end{cases} \quad (7)$$

And

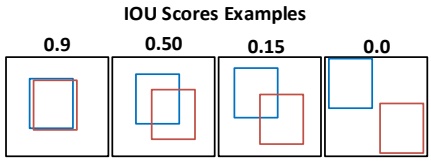
$$L_{con}(x,y)=-\left(\sum_i^N x_i\log(y_i)+\sum_i^N \mathrm{Log}(y_i)\right) \quad (8)$$

4.2 Intersection over the union (IOU)

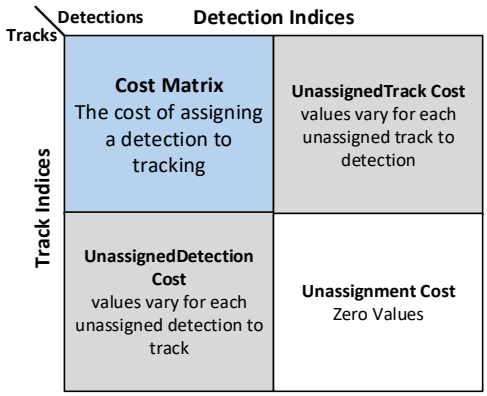
It means that it intersects a bounding box in the current frame with another box in the previous frame, as shown in Fig. 7. The anchors or surrounding squares calculated in advance were created and are of a specific size as well as match the original real squares and correspond to them in the distribution [16]. It was also selected so that the cross-to-union ratio called (IOU) was more significant than 0.5. This term may be formulated, and their dependencies as follow:

$$IOU=\frac{Area\ of\ Overlab}{Area\ of\ Union} \quad (9)$$

i. Form: When there is no significant change in the shape or size of the box surrounding two consecutive frames, the result will increase.
ii. Cost function: - CNN is used to determine the bounding box and compare it with what is in the previous box. When calculating bypass characteristics are the same, that is, the object inside the surrounding merge is the same. If a blockage is found, even partially, the properties remain incomplete, as does the association.



(a)



(b)

Figure. 7 The basic principles: (a) IOU scores calculation and (b) cost matrix

When the algorithm is implemented, two lists that contain squares locations are composed, one for detection and one for tracking. Then we calculate the previous parameters (IOU, shape, and convolutional features). We can also calculate the cost function and use the distance scale, depending on the principle of the sine entirely, and then output the results and the IOU value and then store it as an array [23].

The presence of a match between the bounding squares increases the probability that the object itself. It is possible to draw from the functions present within the Python language to compare the list of detection and tracking lists and to calculate the extent of their interconnection. This matrix is used to calculate unmatched bounding boxes in the detection list with that in the tracking list, as well as the unmatched tracking bounding box, and there is no equivalent in the list.

- maintains a list of currently tracked objects.
- Process current frame to obtain new detections
- Assign current detections to exist trackers using the Hungarian Algorithm which results in matches, unmatched detections, and unmatched trackers
- Assign new trackers to unmatched detections. Keep old trackers for consecutive unmatched detections for `max_age` frames.
- Update tracker's state using tracking algorithm (currently KF)

4.3 Modified KF and adaptive tracking method

After identifying the surrounding boxes, calculating their location, and completing the pairing, the role of a Kalman Filter is performed to make predictions, measurements, and updates. Where a Kalman Filter is composed of a set of mathematical functions to calculate the state and the amount of covariance in the process and measurement phase. In the beginning, a path is renewed and plotted for each bounding box by a KF, which has a confidence value that exceeds the threshold limit and assigns a unique identifier to it. Then the current paths are linked to the object detection operations in the next frame using cross-over calculations and the cost function and its reach to the minimum [15]. The cost function includes the spatial distance (d_s) between the detected box by predicting the new position and the previously defined square of the same object. In addition to the visual displacement (d_v) that takes into account the date of the appearance of the detected object and its comparison with the appearance of the tracked object. The cost function is formulated as follows:

$$C_{ij} = \lambda d_{ij}^{S} + (1-\lambda) d_{ij}^{V} \quad (10)$$

Where λ represents the tradeoff control parameter between spatial and visual distance. The spatial term can be obtained by:

$$d_{ij}^{S} = \left(Bp_j^d - Mean_i\right)^T Cov_i^{-1} \left(Bp_j^d - Mean_i\right) \quad (11)$$

Where Bp represents the bounding box position of the detection stage. While Mean and Cov represent the mean and covariance matrix of the tracking stage. Then, the visual term can be calculated as follow:

$$d_{ij}^{V} = \min(1 - Ap_j^T Ap_k^i \mid Ap_k^i \in \mathcal{R}_i) \quad (12)$$

Where Ap_j represents the appearance description part from the detected bounding box, and Ap_k represents the tracked description part from a set of appearance descriptor $\mathcal{R}$.

To improve the performance of the proposed algorithm, we present a new adaptive tracker method that mainly relies on adding an intersection tracker over the visual Union along with a Kalman filter tracker. The IoU tracker [24] can link and overlap the detections in consecutive frames, which in turn create unique paths for each identifier finder. The parent follower suffers that the trace depends on having a high IoU value to set the object. The emergence of false positive and negative detection, which in turn interferes with these discoveries with accurate tracking. Consequently, tracks that contain at least one detected object and possess the highest IoU value in addition to the minimum length through successive frames are preferred. Which, in turn, increases the emergence of false positive or false negative detections that may stop the prosecution path.

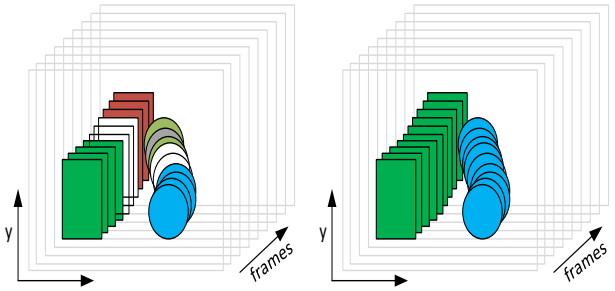


Figure. 8 A general principle for the tracker: (a) the paths of the objects are fragmented as a result of false detections, which in turn lead to an identifier change for the same object and (b) compensating for lost paths using the adaptive tracker

Fig. 8 illustrates the new adaptive tracker where a visual tracking of the object takes place in the absence of a link between detection and tracking. That is if the tracked object does not have an IOU that exceeds the threshold. The visual tracking is relied upon, i.e., whether or not the object appeared in the previous frames and to determine its location. Thus the tracking is used for the object with a certain number of frames. When there is a new detection that exceeds the threshold limit, the visual tracking is stopped, and the original Kalman tracker is returned. Increasing the successive visual frames causes the tracker to lose track or jump to another object. To reduce this, the object is tracked by visual tracking. Tracking backward through the previous frames is also performed for each new object with matching interference criteria, and upon meeting them are combined. It adds talking tracks to the tracker throughout the entire video, in addition to closing gaps and cutting the strings that visually track the surrounding boxes outside the specified path.

4.4 The hardware implementation of the proposed algorithm

In this article, a proposed algorithm is designed and implemented to create a real-time object detection and tracking system and test its features for various models of deep learning. Many CNN models have been used on artificial intelligence (AI) computing algorithms. These models have been trained and tested as object detectors on a different data set, n building the proposed algorithm, as shown in Fig. 5. It achieved high throughput for applications,

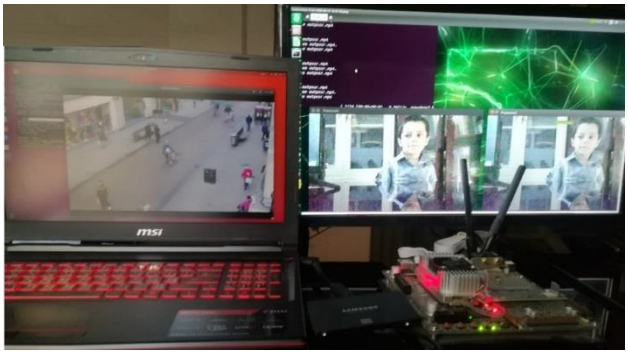


Figure. 9 The hardware implementation of the proposed system

which includes detection, tracking, and categorizing objects. The NVidia Jetson TX2 [25] platform has been used to create a practical application to detect and track objects in real-time, as shown in Fig. 9.

It represents a miniature computer that has a powerful processing unit and fast performance, especially in applications of artificial intelligence [26], also that it has a graphics processing unit (GPU) [27], type NVIDIA Pascal™ 256 Compute Unified Device Architecture (CUDA) Core, which makes it suitable for to that it works at low capacity. From technical specifications for this platform, we noted applications based on the neural network. It is also supported by modern software, including NVidia Jetpack and Software Development Kit (SDK) development kit, including comprehensive libraries and deep-learning software and GPUs [27]. The model has a relatively lower configuration compared to the computers used, and it is also small and lightweight that can perform in real-time with reasonably good accuracy and software and technology researchers.

5. Experimental testing and results

The performance of the proposed algorithm has been examined by applying it to a different data set containing scenes of multiple objects such as vehicles and pedestrians. More than one explorer was used to test the efficiency of the algorithm. The results showed the effectiveness of the proposed approach to the different exploration used in the detection phase. However, the results were mixed, reasonable, and acceptable. In the tracking phase, the proposed method was tested several times with a different degree of detection to know its effect on the recall path. We note that the lower the detection threshold, the more efficient and suggested tracking capacity.

There are several metrics for measuring object performance algorithms. Tracking a single object is somewhat simple while measuring the performance efficiency of algorithms for multiple organisms is more complicated. Where it needs to apply a delicate design to create different sets of correspondence or paths for each object, as in Fig. 10.

Various deep learning network models have also been tried to test the effectiveness and efficiency of the proposed algorithm to obtain the optimal performance. So that an SSD and YOLO object detector network was used, as shown in Fig. 11. The results obtained showed the SSD network's ability to detect objects with high accuracy. So it can also be used in real-time applications and low-resolution video data. The arithmetic time for SSD is higher than YOLO algorithms. We test the efficiency of the proposed algorithm in different circumstances and challenges [7-8]; it was tested on various datasets that include different environments, whether internal or external scenes, as shown in Fig. 12 It shows that the proposed algorithm can be processed the different challenges from multiple video streams sources such as offline, online broadcasting, real-time monitoring source, and others. Then extract useful information with high smoothness, speed, and accuracy where good results were obtained compared to previous existing algorithms, as shown in Fig. 13.

In recent decades, various methods and measures have been proposed, and not a single specific method has been agreed upon. Recently, several measures have emerged that have received the approval and attention of researchers in the field of performance measurement [24]. An example of this is a minimal cost to detect and track actual objects and all forecasts for each frame on a scale called Clear-MOT. In comparison, ID-Measure works to find tha0t cost for all frames at the same time. Multiple objects tracking Accurate (MOTA) and multiple objects Tracking precision (MOTP) (standards provide comparability to prove the efficiency of the proposed algorithm. Including the identification of correspondence and movements, as well as false alarms. Where these events and findings are used in building data structures and analyzing them, as well as using these derived values to find other new measures, the Precision criterion was also calculated, which is the ratio of the number of real discovered objects to the sum of positive and negative detected objects. Likewise, the Recall criterion represents the ratio of the number of positive detected objects to the sum of the original organisms within the framework.

It was measuring of clarity and accuracy of the trajectory and finding the renewing paths for all objects in addition to guessing or predicting paths resulting from hypotheses. Whereas, the target object may be surrounded by multiple outputs. The measure of tracking the target to determine the efficiency of the tracer and for each hypothetical outcome, whether

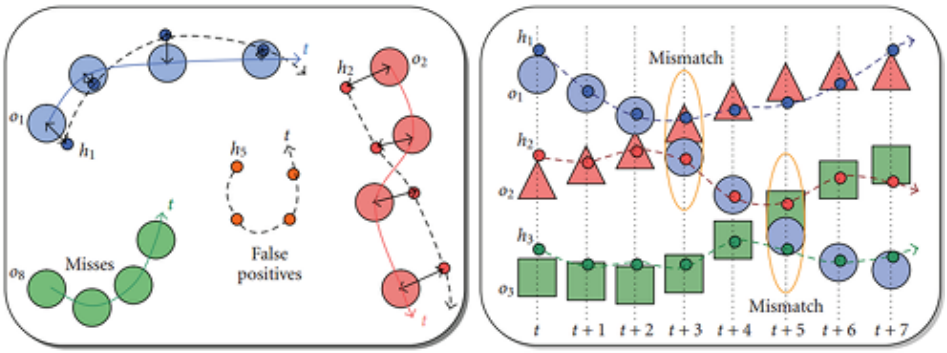


Figure. 10 The estimated paths and correspondence of multiple objects

(a) (b) (c) (d)

Figure. 11 The experimental result of the proposed algorithm for different applications (a) original frame, (b) ground truth, (c), and (d) the proposed algorithm using SSD and YOLO, respectively.

(a)

(b)

(c)

Figure. 12 Visual experimental results for different challenges: (a) variation of views, (b) crossing, and intersection the objects, and (c) speed variation

(a) (b) (c) (d) (e) (f)

Figure. 13 The visual results comparison between different tracking algorithms: (a) ground truth, (b) DSORT [14], (c) IOU tracker, [16] (d) KCF tracker [18], (e), and (f) the proposed algorithm using SSD and YOLO respectively

Table 1. The evaluation performance matrices for the several different trackers by using MOT16 challenges dataset

Tracker	MOTA	IDF1	MOTP	MT	ML	FP	FN	IDR	IDP	FM	Rcll	Prcn
SimpleMOT[17]	62.44	58.46	78.33	205	242	5909	61981	66.01	95.32	1361	66.01	95.32
KCF [19]	48.80	47.19	75.66	120	289	5875	86567	52.52	94.22	1116	52.52	94.22
IOU[15]	53.22	45.03	75.54	150	238	8123	60728	55.52	90.31	2302	59.12	90.56
V-IOU [18]	57.14	46.90	77.12	179	250	5702	70278	61.45	95.16	3028	61.45	95.16
Tracker[16]	57.01	58.24	78.13	167	262	4332	73573	59.65	80.17	0.73	475	859
DeepSORT[14]	61.44	62.22	79.07	249	138	12852	56668	68.92	90.72	2008	68.92	90.72
Our Tracker1	64.50	60.01	76.19	300	142	15098	42286	76.81	90.27	1678	76.81	90.27
Our Tracker2	66.57	57.23	78.40	250	175	10034	42305	73.51	93.75	3112	73.51	93.75

Table 2. The evaluation performance matrices for the several different trackers by using CVPR19 challenges

Tracker	MOTA	IDF1	MOTP	MT	ML	FP	FN	IDR	IDP	FM	Rcll	Prcn
SimpleMOT[17]	53.63	50.56	80.09	376	311	6439	231298	55.30	90.80	4335	55.30	97.80
KCF[19]	50.79	52.15	76.83	472	240	58689	193199	62.66	84.67	4233	62.66	84.67
IOU[15]	43.94	31.37	76.13	312	284	50710	235059	54.57	84.78	5754	54.57	84.78
V-IOU[18]	42.66	45.13	78.49	208	326	27521	264694	48.84	90.18	17798	48.84	90.18
Tracktor[16]	54.46	50.06	77.35	415	245	37937	195242	52.56	70.45	2580	62.27	89.47
DeepSORT[14]	53.54	49.30	79.78	383	321	7211	230862	55.38	97.55	4673	55.38	97.55
Our Tracker1	56.84	42.15	78.00	312	189	6014	206176	64.36	88.93	7207	66.36	88.93
Our Tracker2	61.81	67.28	78.62	455	123	5440	158901	72.82	90.56	5874	72.82	91.56

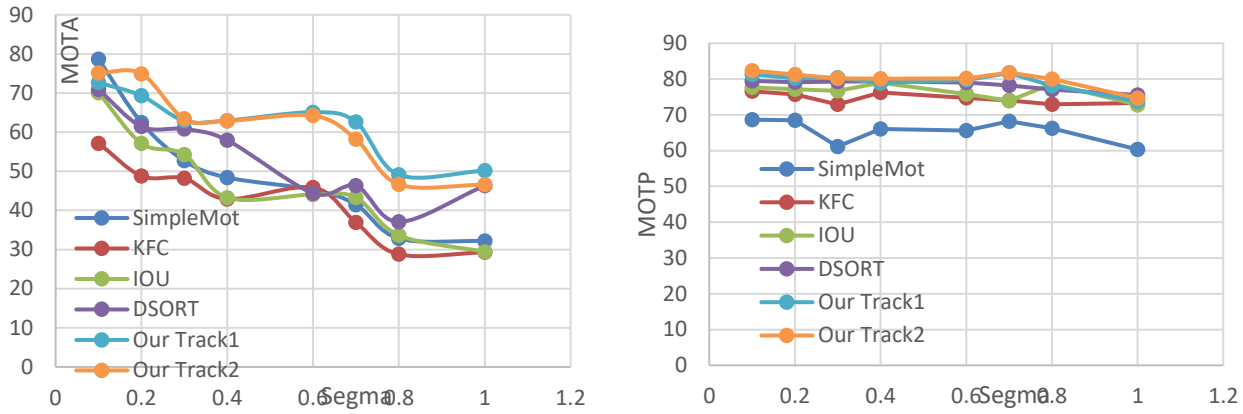


Figure. 14 The comparison results between different tracking algorithms: (a) MOTA and (b) MOTP

it was a real positive (TP), meaning the actual object was present, or a false alarm, i.e., a false positive. The proposed algorithm may fail to recognize the object and be False Negative (FN)).

The result should be right concerning false alarms and are close to each other [23]. Table (1) and (2) show the results and values for some of the performance measures of the proposed tracking algorithm and its efficiency for different Datasets challenges, as it was compared with some other tracking algorithms. Also, some measures of identification of matching were calculated, where Fp and Fn, which represent the number of false-positive and negative matches, respectively. Then, Identification Precision (IDP), Identification Recall (IDR), and Identification F-measure (IDF) for global minimum cost. Fig. 14 represents the comparison results between different Tracking algorithms for different Thresholding detection Values.

6. Conclusion

We presented a method to integrate an intersection-based visual tracking over the union (IOU) with a Kalman filter tracker. He completed several simulations as well as the practical implementation of the proposed algorithm. The results showed the possibility of compensating for the false negative detections, the number of switches and hashes are reduced, and thus the quality of the tracks is greatly improved, whether in the case of single or multiple objects tracking. The results show the speed of the proposed algorithm in detecting and tracking objects and their accuracy, providing a set of available capabilities It has an ability and features to deal with tracking challenges such as blockage, speed change, etc. We propose several future proposals to arrive at a comprehensive algorithm capable of detection and tracking.

Conflicts of Interest

We confirm that there is no conflict of interest in this work.

Author Contributions

“Conceptualization, and Methodology, N. H. Abdulghafoor and H. N. Abdullah; Software, N. H. Abdulghafoor; Validation, N. H. Abdulghafoor and H. N. Abdulla; Formal analysis, N. H. Abdulghafoor; investigation, N. H. Abdulghafoor; resources, N. H. Abdulghafoor; data curation N. H. Abdulghafoor; Writing—original draft preparation, N. H. Abdulghafoor; Writing—review and editing, N. H. Abdulghafoor and H. N. Abdullah; visualization, N. H. Abdulghafoor; supervision, H. N. Abdullah; project administration, H. N. Abdullah; funding acquisition, N. H. Abdulghafoor”.

References

[1] D. Hemanth and V. Estrela, eds. *Deep learning for image processing applications*. Vol. 31. IOS Press, 2017.

[2] N. H. Abdulghafoor and H. N. Abdullah, “Real-Time Object Detection with Simultaneous Denoising using Low-Rank and Total Variation Models”, In: *2nd International Cong. on Human-Computer Interaction, Optimization and Robotic Applications (HORA2020)*, Anqara Turkey, pp.1-10, IEEE, 2020,

[3] H. N. Abdullah and N. H. Abdulghafoor, “Objects detection and tracking using fast principle component purist and kalman filter”, *International Journal of Electrical & Computer Engineering*, Vol. 10, No.2, pp. 1317-1326, 2020.

[4] A. Milan, L. Laura, R. Ian, R. Stefan, and S. Konrad, “MOT16: A benchmark for multi-

object tracking", *arXiv preprint arXiv:1603.00831,* 2016.

[5] P. Dendorfer, R. Hamid, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixe. "CVPR19 Tracking and Detection Challenge: How crowded can it get?", *arXiv preprint arXiv:1906.04567,* 2019.

[6] N. H, Abdulghafoor and H. N. Abdullah, "Real-Time Moving Objects Detection and Tracking Using Deep-Stream Technology", *Journal of Engineering Science and Technology*, Vol. 16, No. 1, 2021.

[7] A. Gad and S. John, *Practical Computer Vision Applications Using Deep Learning with CNNs*. Apress, 2018.

[8] R, Shaoqing, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 39, No 6, pp. 1137 - 1149, 2017.

[9] Z. Zhong-Qiu, Z. Peng, X. Shou-Tao, and W. Xindong, "Object detection with deep learning: A Review", *IEEE Transactions on Neural Networks and Learning Systems,* Vol. 30, No. 11, pp. 3212-3232, 2019.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector", In: *Proc. of European Conf. on Computer Vision*, Lecture Notes in Computer Science, Vol. 9905. Springer, Cham. Springer, Cham, pp. 21-37, 2016.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection", In: *Proc. of* 2016 *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 779-788, 2016.

[12] J. Redmon, and A. Farhadi, "Yolov3: An Incremental Improvement", *arXiv preprint arXiv: 1804.02767,* 2018.

[13] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and real-time tracking", In: *Proc. of IEEE International Conf. on Image Processing (ICIP*), Phoenix, AZ, USA, pp. 3464-3468, 2016.

[14] N. Wojke, A. Bewley, and D. Paulus, "Simple online and real-time tracking with a deep association metric", In: *Proc. of IEEE International Conf. on Image Processing (ICIP),* Beijing, China, pp. 3645-3649, 2017.

[15] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/ accuracy tradeoffs for modern convolutional object detectors", In: *Proc. of the IEEE Conference on Computer Vision and Pattern Rrecognition*, Honolulu, HI, USA, pp. 7310-7311, 2017.

[16] G. Chandan, A. Jain, and H. Jain, "Real-time object detection and tracking using Deep Learning and OpenCV", In: *International Conf. on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, pp. 1305-1308, 2018.

[17] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information", In: *Proc. of 14th IEEE International Conf. on Advanced Video and Signal Based Surveillance (AVSS*), Lecce, Italy, pp. 1-6, 2017.

[18] E. Bochinski, S. Tobias and S. Thomas, "Extending IOU based multi-object tracking by visual information", In: *Proc. of 15th IEEE International Conf. on Advanced Video and Signal Based Surveillance (AVSS)*, Auckland, New Zealand, pp. 1-6, 2018.

[19] M. Raj and S. Chandan, "Real-Time Vehicle and Pedestrian Detection Through SSD in Indian Traffic Conditions", In: *Proc. of IEEE International Conf. on Computing, Power, and Communication Technologies (GUCON),* Greater Noida, Uttar Pradesh, India, pp. 439-444, 2018.

[20] S. Hossain and D. J. Lee, "Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices", *Sensors*, Vol. 19, No.15, pp. 3371-3395, 2019.

[21] B. Blanco-Filgueira, D. García-Lesta, M. Fernández-Sanjurjo, V. Brea, and P. López, "Deep learning-based multiple object visual tracking on embedded system for IoT and mobile edge computing applications", *IEEE Internet of Things Journal,* Vol. 6, No.3, pp. 5423-5431, 2019.

[22] H. N. Abdullah and N. H. Abdulghafoor, "Automatic Objects Detection and Tracking Using FPCP, Blob Analysis, and Kalman Filter", *Engineering and Technology Journal*, Vol. 38, No. 2 part (A) Engineering, pp. 246-254, 2020.

[23] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression. In: *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 658-666, 2019.

[24] P. Kaustubh, "NVIDIA DeepStream SDK 4.0.2 Release, Quick Start Guide", https://docs. Nvidia.com/ metropolis/deepstream/dev-guide/, 2018.
[25] S. Han, H. Shen, M. Philipose, S. Agarwal, A. Wolman, and A. Krishnamurthy, "Mcdnn: An approximation-based execution framework for deep stream processing under resource constraints", In: *Proc. of the 14th Annual International Conf. on Mobile Systems, Applications, and Services*, Singapore Singapore, pp. 123-136, 2016.
[26] Hamieh, R, Myers, H. Nimri, T. Rahman, A. Younan, B. Sato, A. El-Kadri, S. Nissan, and K. Tepe, "LiDAR and Camera-Based Convolutional Neural Network Detection for Autonomous Driving", *SAE Technical Paper*, No. 2020-01-0136, 2020.
[27] M. Colbert and J. Krivanek, "GPU-based importance sampling", *GPU Gems 3*, pp. 459-476, 2007.

A notation list

Symbol	Abbreviation
A	state transition matrix
Ap	Appearance descriptoe part
b_g	Ground Truth position of a bounding box
b_p	Predicated position of a bounding box
B	coverts control input
Bp	Bounding box position
C_{ij}	The cost function
d_v	visual displacement
e	Exponential function
g	Intensity
h_k	Height of bounding box
H	model matrices
I,J	Curves
K	Kalman gain
L	Loss function
L_{con}	Confidence loss function
L_{loc}	Location loss function
Log	Logarithm function
N	The number of matching boxes
P	measurement error covariance
Q	process noise covariance
$\mathscr{R}$	a set of appearance descriptor
w	domain
w_k	Width of a bounding box
x	Centre coordinates
x_k	Centre horizontal position value
X	measurement matrix
y_k	Centre vertical position value
α	Balance parameter between the losses
λ	Tradeoff control parameter
σ	Variance function