Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 7848-7859
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.889

Comparison and study of Pedestrian Tracking using Deep SORT and state of the art detectors

Farah Jamal Ansari, Section of Computer Engineering, University Polytechnic, Jamia Millia Islamia, New Delhi. (India), fansari@jmi.ac.in
Anushka Dhiman, Delhi University, New Delhi. (India), dhiman.anushka23@gmail.com
Aleem Ali, Department of Computer Science & Engg., Glocal University, Saharanpur-247122. (India), aleem@theglocaluniversity.in

Abstract- Object Tracking is becoming very popular these days in the computer vision field. It is the process of tracking an object across a sequence of frames. Deep Sort is a very fast and powerful tracking algorithm. It has a practical way of approaching multiple object tracking problems. It uses the appearance information to track objects through occlusions and thereby reducing the identity switches. Performance evaluation and comparison have been performed on pedestrian tracking using the Deep Sort algorithm in conjunction with the various state-of-the-art object detectors: YOLO, SSD and FasterRCNN. Criteria for Evaluation, datasets used for evaluation, along with the quantitative results have been described and discussed in this work.

Keywords: **Pedestrian Tracking, Deep Sort algorithm, Faster R-CNN, SSD, YOLO.**

I. INTRODUCTION

Videos are sequences of images or frames that are viewed at a high enough pace for human eyes to perceive content continuity. Object detection in video and images entails identifying and locating an object in the image. In a video series, object tracking involves keeping track of an object's spatial and temporal shifts, such as its presence, position, scale, and shape. Since tracking starts with object detection, which iteratively verifies the tracking, the two are inextricably connected. There are some frames where the visual appearance of the object is not clear, while there is a moving object in a video. In that case, the detection fails but is overcome by tracking as it also contains the motion pattern and the history of the object. Some challenges in object tracking:

1. Occlusion: It occurs when an object that is being tracked is hidden by another object. Like two people walking past each other or a car entering into a tunnel. The problem, in this case, is what to do when the object reappears.
2. Background clutter: Background clutter occurs when the background near the object is the same color or texture as the object. As a result, tracking the object in a cluttered background becomes difficult.
3. Appearance change: A point of view of an object can look very different visually without the context. Therefore, it becomes very difficult to identify the object using only visual detection.

The goal of this paper is to compare the state-of-the-art tracking algorithm using the Kalman filter and deep associations with a variety of state-of-the-art deep learning-based detectors for the pedestrian dataset. We have also discussed the various shortcomings of each detector when implemented on the pedestrian dataset.

The outline of the paper is in this manner. The literature review and various methods for pedestrian monitoring are discussed in Section 2. The different monitoring methods are discussed in section 3. The paper's Simple Online Real-time Tracking with a Deep Association Metric was defined in section 4. In Section 5, we compare and contrast the various deep learning-based detectors. In section 6 and section 7 pedestrian dataset and evaluation metrics have been explained. In 8 we have discussed our results and 9 is Evaluation Analysis & Conclusions.

II. LITERATURE SURVEY:

2.1 Multiple Object Tracking with Mean Shift and CAMShift:

Wang. M., Li. W. et. al., [1] present a tracking system for multiple targets in occlusion and illumination-sensitive scenes. The system made use of the CFMS (Combination Feature and Mean Shift) algorithm, which is an extension of Mean Shift. Center position, distance, height, area, and Harris corners are the five features

used for multiple object tracking. The proposed double threshold Harris corner detection algorithm is ideal for multiple tracking videos with occlusions because it extracts Harris corner information. The corners in the occluding area were identified using the K-NN classifier. As a consequence, a system based on feature fusion and the mean shift algorithm is developed to track multiple objects more efficiently.

S. S., Kondo T. Et. al., [2] proposed the CAMShift algorithm for multiple tracking targets. This technique searches objects with the same hue value and pattern shape recognition for one selected object as a template. For missing objects in the frame, the frame is searched for most similar-looking objects and tracks them. This technique separates a target object from the background containing noise. The object recognition method counts objects to be tracked.

To solve the occlusion issue, [3] proposes an improved mean-shift tracking method. The theory of occlusion layers is implemented to form a relationship between the occluding and non-occluding regions of pedestrians to overcome occlusion. The occluded pedestrians are progressively modified to remove the impact of occlusion. The proposed algorithm is efficient for tracking occlusion amidst pedestrians where the traditional tracking algorithm fails.

In [4], Yan et al. proposed an algorithm based on Camshift and Kalman filtering for pedestrian tracking. iteratively calculating the best matching window and spatial information is done using the Camshift algorithm. To predict the state of a moving object, the Kalman filter is used. The method is experimentally suitable for pedestrian tracking.

2.2 Multiple Object Tracking with Optical Flow:

Yamamoto et al. proposed in [5] a method for extracting the optical flow from a series of images and tracking moving objects in real-time using a flow of vectors. The generalized gradient model [6], [7], which calculates spatial and temporal intensity gradients, is used to calculate optical flow. In each image, a region with similar flow vectors is extracted, modified, and tracked. Optical flow is extracted using a special image processor [8], [9] to track the object in real-time. With this method, the system tracks two overlapping objects. The disadvantage is that the motion of the object obtained is not precise if the shape of the object is not rectangular.

Urban Tracker (UT) [10], [11], and Multiple Kernelized Correlation Filter Tracker (MKCF) [12] are two trackers that use background subtraction to obtain foreground blobs to track multiple objects in a frame. The presence of foreground blob merging, fragmentation, and background subtraction induced by shadows are all disadvantages of images generated by background subtraction methods in the scene. To address these issues, J. P. Jodoin et al. proposed a system that combines context subtraction [13], optical flow [14], and edge detection to form a binary image of foreground blobs. For each blob in the image, the dense optical flow is then computed. To get the edges of the foreground objects, they use the Canny edge detector [15] on both the blobs and the background image. Finally, it changes the object sizes, separates close objects that appeared as a single blob during background subtraction, and removes noise. With all of this data, the authors' method creates a binary image addressing fragmentation, merging, and noise removal, resulting in a more precise result.

2.3 Multiple Object Tracking with Kalman Filtering

Li et al. introduced a Kalman filter for object tracking in [16], which uses the object's current location and bounding box to predict the position of the object in the next frame, minimizing the time spent looking for a moving object. It also establishes corresponding relationships through features to handle separation after the objects are merged. The experiments show that the proposed approach achieves reliable monitoring.

In [17], Xi Chen et al. proposed an unscented Kalman filter (UKF) for a more accurate and reliable detection cum tracking framework using a nonlinear tracking algorithm. This approach differs from traditional Kalman filtering (KF), which fails to achieve optimal estimation in nonlinear tracking situations. UKF is used for linear and nonlinear tracking because of the unscented transform. It estimates the temporal information for each detected object and tracks many moving objects even while in occlusion. As a result, the proposed method is very accurate for detecting multiple objects having non-linear motion and also suffering from occlusion.

In [18], M. Meuter et al. present a camera-based pedestrian tracking, which is a time-efficient estimation framework. Image processing techniques are used to identify objects of interest in each frame. They created a new approach for estimating target movement while taking into account host movement and intrinsic and

extrinsic camera parameters. The spatio-temporal model is combined in an unscented Kalman filter. Therefore, it is well equipped to tracking moving objects even if suffering from occlusion.

III. TRACKING TECHNIQUES

3.1 Mean Shift

It is an algorithm that moves data points in a neighboring region towards the mean of data points in that region iteratively[19].

Consider a set S of n data points x_i in a d dimentional Euclidean space X. Let K(x) be a kernel function that represents the number of points that contribute to the mean estimation.

Then, the mean m at location x of the kernel K is given by

$$m(x) = \frac{\sum_{i=1}^{n} K(x-x_i)x_i}{\sum_{i=1}^{n} K(x-x_i)} \quad (1)$$

where the mean shift is denoted by m(x) -x.

The mean shift algorithm moves to its mean iteratively.

m(x) moves closer to x with each iteration.

The algorithm comes to a halt when m(x) = x.

The x trajectory is generated by the sequence of x, m(x), m(m(x)),...

If the mean is estimated at several points, all of the points are changed at the same time during each iteration.

So, kernel K a function of $\|x\|^2$ is

$$K(x) = k(\|x\|^2) \quad (2)$$

where, k is the profile of K and is nonnegative, nonincreasing i.e. k(x) ≥ k(y) if x < y and piecewise continuous and

$$\int_0^\infty k(x)dx < \infty \quad (3)$$

The probability density is estimated using kernel density estimation (Parzen window technique).

The mean squared error between the estimate and the actual density is minimized by the Epanechnikov kernel and is used to estimate the validity of a kernel density estimator. Objects are detected by matching the color probability. Hence, mean shift is used to estimate the color probability and target location.

3.2 Continuous Adaptive Mean Shift (CAMSHIFT)

Continuously Adaptive Meanshift (Camshift) [20] is an extented version of the meanshift algorithm which provides more accuracy and robustness to the model. With the Camshift algorithm, the size of the window keeps updating when the tracking window tries to converge. It uses the color information of the object for tracking. It also provides the best fitting tracking window for object tracking. It first applies meanshift, then adjusts the window size as follows:

$$s = 2 \times \sqrt{\frac{M_{00}}{256}} \quad (4)$$

It then determines the ellipse that fits it the best. Then, with the newly scaled search window and the previous window, apply the meanshift once again. This procedure is repeated until the accuracy is good.

3.3 Optical Flow

The primary method for measuring motion frame intensity, which can be related to the motion of objects in a scene, is optical flow. It provides a brief overview of both the areas of the frame that are moving and the speed at which they are moving. In practice, computation of optical flow is sensitive to multiple objects tracking with occlusion and illumination changes.

The problem of optical flow may be expressed as [21]:

Consider a pixel I (x,y,t) in the initial frame that moves by a distance (dx, dy) at dt intervals in the next frame. Assuming that the pixel strength is constant,

$$I(x,y,t) = I(x+dx, y+dy, t+dt) \quad (5)$$

After eliminating the generic terms, divide by dt using the Taylor series approximation,

$$f_x u + f_y v + f_t = 0 \quad (6)$$

where,

$$f_x = \frac{\partial f}{\partial x} ; f_y = \frac{\partial f}{\partial x} \quad (7)$$

$$u = \frac{dx}{dt} ; v = \frac{dy}{dt} \quad (8)$$

This equation is called the Optical Flow equation. where f_x and f_y are image gradients, f_t is the time gradient, and (u,v) is unknown.

3.4 Kalman Filtering

Based on a dynamic model, the Kalman filter tracks moving objects by estimating a state vector containing the target's parameters, such as location and velocity. Since different movement conditions and occlusions can impede an object's vision tracking. It is considered to employ the Kalman filter technique, which allows for minor occlusions and complex object movements. The Kalman filter is a recursive estimator that calculates the current state based on previous states and measurements.

In 1960 R. E. Kalman presented the Kalman Filter (KF) [22], which extracts the useful signal from noisy measurement variables. The measurement variables are used as input signals, and it is based on the system's statistical characteristics and measurement noise. A prediction equation and an update equation describe the entire operation.

$$x(n) = F \cdot X(n-1) + V_q(n-1) \quad (9)$$

$$y(n) = H \cdot X(n-1) + V_p(n-1) \quad (10)$$

where x(n) is the state variable and y(n) is the measurement variable . F is the state transition matrix and H is the measurement matrix. $V_q(n)$ is sytem noise and $V_p(n)$ the measurement noise.

IV. Simple Online Realtime Tacking with a Deep Association Metric (DeepSORT)

Deep SORT [23], an extension of [24] SORT, is one of the most widely used frameworks (Simple Real time Tracker). Nicolai et al. [23] used appearance information to improve SORT's performance. This extension allows objects to be tracked over longer duration of occlusion, and ttherefore reduces the number of identity changes.

4.1 Sort with Deep Association Metric

A hypothesis by Nicolai et al. [23] using recursive Kalman filter along with data association from frame to frame.

Nicolai et al. uses the Kalman Filter for multiple object tracking benchmarks. Nicolai et al. described a tracking scenario on an eight-dimensional state space (u, v, γ, h, x', y', γ', h') representing a bounding box with centre position (u, v), aspect ratio γ, height h. , which represents a bounding box with a centre location (u, v), aspect ratio h, and height h. The bounding box coordinates (u, v, γ, h) are passed as observations in a Kalman filter, which is based on the constant velocity linear model. It also has a parameter that monitors and deletes tracks of objects that are no more the contenders. The Kalman filter trackes the new bounding boxes, and associates

new detections with new predictions. The squared Mahalanobis distance is used to incorporate the motion information and calculate the associations.

The Mahalanobis distance is a multivariate distance metric used to calculate the distance between two points. Formula to compute Mahalanobis distance:

$$D^2 = (x-m)^T \cdot C^{-1} \cdot (x-m) \quad (11)$$

where, D^2 is the squared Mahalanobis distance, x is the observation vector, m is the vector of mean values of independent variables, C^{-1} is the vector of independent variable mean values, (x – m) s the distance of the vector from the mean divided by the covariance matrix, and (x – m) is the distance of the vector from the mean divided by the covariance matrix (or multiplied by the inverse of the covariance matrix). The squared Mahalanobis was used by Nicolai et al. to determine the difference between expected Kalman states and newly arrived measurements:

$$d^{(1)}(i, j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i), \quad (12)$$

where (y_i, S_i) denotes the projection of the i-th track distribution into measurement space and d_j denotes the j-th detection.. It how much the detection is away from the mean track location. As in the paper, to remove unwanted associations a threshold is put on the Mahalanobis distance to be at a confidence interval of 95% calculated from the inverse chi-square distribution .

$$b^{(1)}_{i,j} = 1[d^{(1)}(i, j) \leq t^{(1)}] \quad (13)$$

The Mahalanobis distance is an appropriate association metric when motion uncertainty is minimal. The expected state distribution is obtained using the Kalman filtering framework, which gives a rough idea of the object position. It makes the Mahalanobis distance ambiguous for tracking through occlusions. Therefore, Nicolai et al. clubbed a second metric.

$$d^{(2)}_{(i,j)} = \min\{1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in R_i\} \quad (14)$$

$$b_{i,j}^{(2)} = 1[d^{(2)}_{(i,j)} \leq t^{(2)}] \quad (15)$$

The equation gives as a binary variable to indicate a valid association.

$$c_{i,j} = \lambda\, d^{(1)}_{(i,j)} + (1 - \lambda) d^{(2)}_{(i,j)} \quad (16)$$

$$b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)} \quad (17)$$

As a result, (16) is a weighted sum to construct the association problem, with the association admissible if it is within the gating region of both metrics (17).

When object occlusion persists for a longer time, it gives rise to an uncertainty in the object position due to which the probability mass spreads out and the observation likelihood becomes uncertain. Matching Cascade is used to solve the assignment problem while calculating measurement-to-track associations. When two tracks show same detection, the Mahalanobis distance becomes more uncertaint as the standard deviation of any detection is levrages to the average of the predicted track. As a result, Nicolai et al. [23] established a matching cascade that prioritises frequently occurring objects.

In the whole framework detections are provided from an object detector , Kalman filter which does the tracking and Matching Cascade to solve the association problem.

Nicolai et al. introduced the appearance feature vector. So, first create a classifier and train it with high accuracy on the dataset, and then remove the final classification layer. Then, with a dense layer produces a feature vector, which is to be classified. The object's appearance descriptor is represented by this feature vector.

The CNN architecture of the network is shown in Figure 1. The deep association clock consists of a residual network [25], two convolutional layers and six residual blocks. The dimentionality of the features map computed by the dense layer is 128 . A batch layer and l2 normalization projects feature onto the unit hypersphere to be compatible with the cosine appearance metric.

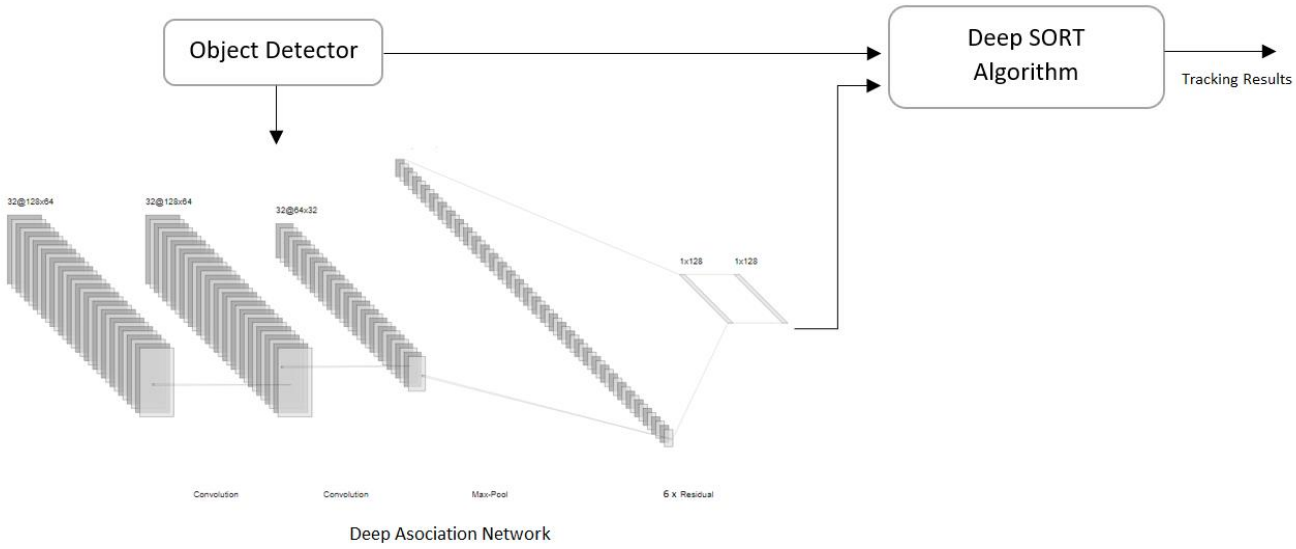


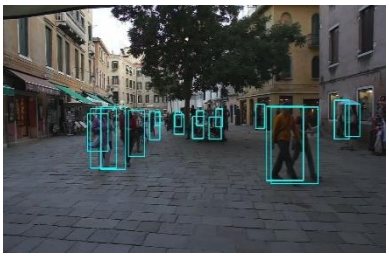
Figure 1: Deep SORT Architecture

V. STATE-OF-THE-DETECTORS

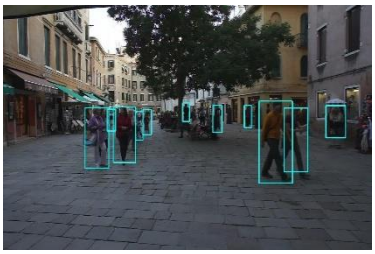
There has been a significant advancement in the area of object detection following the discovery of CNN and deep learining. Graphic Processing Units (GPUs) through parallelization has significantly helped in reducing the problems related to the the real-time processing associated with complex CNN computations. Thereby CNN can be easily used for real time video processing.

Since 2012, many deep learning algorithms and CNN architectures have been proposed such as R-CNN and its various other versions such as [26], [27], [28], [29]. Similarly variants of You Only Look Once (YOLO) are also available [30], [31], [32], [33]. For detecting object, we applied different object detection framework namely, Faster Region CNN (FRCNN) , Single Shot Detector (SSD) [34] and YOLOv4 [33]. The FRCNN model is comprised of two steps . In the first step a deep convolutional network is used for generating region proposal which are classified into different objects in the second step. On the other hand in SSD only a single shot is required to detect objects. Therefore detection is SSD is quite fast as compared to FRCNN. SSD extracts feature maps and uses , a 3 × 3 convolution filter to each cell for prediction. In 2016, YOLO Joseph Redmon proposed YOLO. Unlike the other region based detectors, YOLO passes the entire image only once to a CNN , and this makes it very fast. The image is split into grid of m by m, and bounding boxes and their class probabilities are generated. YOLOv2 introduced batch-normalization and a reduced localization error and better recall when compared to region-based detectors. YOLOv3 was released in 2013 and is known for its high accuracy. It has replaced softmax with logistic regression. Recently, YOLOv4 is proposed by Bochkovskiy et al. with a great improvement on YOLOv3. It has an improved MAP (mean Average Precision) by 10% and FPS (Frame per Second) by 12%.

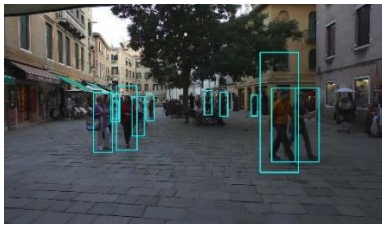
In this paper, we compare tracking results, with three different detectors : Faster Region CNN (FRCNN) , SSD and YOLOv4. Throughout this work, we applied the Faster Region CNN (FRCNN), SSD and YOLOv3, YOLOv4 with default parameters. We disregard all other classes and only pass pedestrian detection results to the tracking framework if the performance probability is greater than 50%.



Pedestrians Detection with SSD.



Pedestrians Detection with YOLO.



Pedestrians Detection with FRCNN.

Figure 2: Pedestrians Detection Results with SSD, YOLO

VI. DATASETS

Multiple benchmarks exist for evaluating tracking models, the most commonly used is multiple object tracking (MOT) [35] . MOT Challenge is a competition used to benchmark multiple object tracking models. The dataset has video sequences labelled with bounding boxes for each pedestrian and is collected from multiple sources, which differ in resolution, frame rate, illumination etc. There are two variations of the challenge. In one task, there is the raw video sequences, and thus need to do both: detection and tracking. In the other, there is the sequences along with a set of detections, and the task is to make an as accurate tracker as possible using these detections.

We have evaluated and compared the tracking results arising from differens state of the art detectors on MOT 2016. The MOT16 data set is a data set proposed in 2016 to measure the standards of multi-target tracking detection and tracking methods, specifically for pedestrian tracking. The official website address is https://motchallenge.net/

VII. EVALUATION METRICS

The metrics of MOT16 [35] are based on [36][37] CLEAR MOT and MTMC. During the evaluation starting point, all targets that appear must be found in time; the target position should be as consistent as possible with the real target position; each target should be assigned a unique ID, and the ID assigned by the target remains unchanged throughout the sequence.

Steps in the evaluation process are:

1. Establish an optimal one-to-one correspondence between the target and the hypothetical optimal, called correspondence.

2. Calculate the position offset error for all correspondences.

3. Calculate cumulative structural error a, the number of missed detections b, the number of false alarms c and the number of times the tracking target jumps.

The meaning of each alias is given by:

Rcll(Recall):
The ratio of TP boxes to GT boxes.

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

Prcn(Precision):
The ratio of TP boxes to all detected boxes.

$$Precision = \frac{TP}{TP+FP} \quad (19)$$

GT:
The number of ground truth trajectories.

MT:
The proportion of tracks that meet ground truth that matches successfully at least 80% of the time in all tracking targets.

PT:
The number of trajectories that have 20% to 80% target tracked.

$$PT = GT - MT - ML \quad (20)$$

ML:
The proportion of tracks that meet ground truth that matches successfully in less than 20% of the time in all tracking targets. Total false positive number among all frames.

$$FP = \sum_t \sum_i fp_i, t \quad (21)$$

FP:

The track and detection predicted by the current frame do not match, and the track point that is incorrectly predicted is called False Positive. Whether the match is successful is related to the threshold set during the match.

FN :
The track and detection of the current frame prediction do not match, and the unmatched ground truth point is called False Negatives. (also, can be called Miss)

$$FN = \sum_t \sum_i fn_i, t \quad (22)$$

IDs:
ID switch number, indicates the number of times the ID assigned by ground truth has changed i.e.; the times of ID jumps.

$$IDs = \sum_t ids_{i,t} \quad (23)$$

FM :
FM calculates number of times the tracking has been interrupted i.e.; the ground truth track is not matched. In other words, whenever the track changes its state from the tracking state to the untracked state, and the tracking is the same at a next point in time. Fragmentation reflects the continuity of trajectories. When trajectories are determinate, it counts all missed target in each frame.

MOTA :
This metric describes the tracking accuracy. It takes into account FN, FP, and IDS and provides a simple way to monitor how well it detects objects and keeps track of them. It has nothing to do with target detection accuracy.

MOTA has a value of less than 100, but when the tracker's error exceeds the number of objects in the scene, MOTA will become negative.

$$T = \sum_t \sum_i gt_{i,t} \quad (24)$$

$$MOTA = 1 - \frac{FN + FP + IDS}{T} \quad (25)$$

MOTP :
A metric reflects the tracking precision.

$$MOTP = \frac{\sum_{i,t} IoU_{t,i}}{TP} \quad (26)$$

Hence, MOTA and MOTP calculates all frame-related indicators before averaging, not calculating the rate of each frame and then averaging the rate.

Detectors	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP
FRCNN(Baseline)[1]	53.40%	96.80%	517	203	254	60	**2629**	69329	723	28038	**51.10%**	22.50%
FRCNN(Inception ResNet v2)	40.40%	78.80%	517	130	187	200	16154	88590	1438	17151	28.60%	31.60%
SSD(ResNet v2)	21.40%	83.80%	517	28	156	333	6154	116779	**539**	**10340**	16.90%	28.50%
YOLOv3	38.80%	93.10%	517	135	231	151	4291	90987	2121	16428	34.50%	**33.20%**
YOLOv4	54.10%	83.30%	517	222	245	50	16156	**68189**	7371	18394	**38.30%**	30.00%

Table 1: Tracker result on train set at IOU Threshold = 0.10

Detector	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOT
FRCNN(Baseline)[1]	52.80%	95.80%	517	198	257	62	**3441**	**70141**	750	27809	**50.00%**	20.90
FRCNN(Inception ResNet v2)	38.20%	74.60%	517	118	170	229	19373	91809	1261	16512	24.40%	**27.50**
SSD(ResNet v2)	20.40%	79.80%	517	27	140	350	7669	118294	**463**	**10138**	14.90%	25.10
YOLOv3	37.70%	90.50%	517	125	230	162	5891	92587	2166	16118	32.30%	30.30
YOLOv4	51.70%	79.50%	517	196	249	72	19807	71840	7001	17569	**33.60%**	26.60

Table 2: Tracker result on train set at IOU Threshold = 0.30

Detector	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP
FRCNN(Baseline)[1]	50.30%	91.20%	517	170	269	78	**7231**	**73931**	764	26230	**44.90%**	18.50%
FRCNN(Inception ResNet v2)	34.30%	66.90%	517	168	157	251	25232	97668	1026	15112	16.60%	22.90%
SSD(ResNet v2)	18.50%	72.50%	517	17	141	359	10462	121087	**374**	**8942**	11.20%	20.10%
YOLOv3	33.50%	80.50%	517	89	238	190	12085	98781	1709	14984	**24.30%**	**26.00%**
YOLOv4	46.40%	71.40%	517	133	278	106	27604	79637	6259	16349	23.60%	22.50%

Table 3: Tracker result on train set at IOU Threshold = 0.50

VIII. DISCUSSION

From the above qualitative results, it can be noticed that Faster R-CNN is detecting many false positives while detecting the pedestrians. However, SSD doesn't have the problem of false positives, but it has high miss rate. The reason being many false positives with Faster R-CNN are due to low quality of images. Since, Faster R-CNN has more problems with hard negatives in low-resolution images, so it gives high False Positives.

However, SSD can handle these hard negatives and small objects better, but it has higher miss rate. While comparing YOLOv3 and YOLOv4, YOLOv4 gives high False Positives and YOLOv3 has high miss rate. That means, YOLOv4 found to be much efficient that it can achieve high MOTA in accordance with the MS-COCO[38]. In our opinion, these results are indicative but not very accurate, because these state-of-the-art detectors are trained on MS-COCO [38]. To have good results, we need to train all of them on certain pedestrian datasets and then evaluate.

IX. Evaluation Analysis & Conclusions

In this paper, we evaluate the Deep Sort tracker plugged in with the various state-of-the-art detectors that are FRCNN Inception ResNet v2, SSD ResNet v2, YOLOv3, and YOLOv4 on the MOT16 benchmark on train set including seven train sequences. The performance of the tracker has been assessed on the MOT16 benchmark on the train set. Since, ground truth files are not available on the test set, therefore results are evaluated on the train set to evaluate the results with different detectors. Py-motmetrics library [39] has been used to evaluate the results which is a Python implementation of metrics for benchmarking multiple object trackers (MOT). The results have been summarized in Table1, Table 2, and Table 3. We have compared all detectors mentioned above and also on detector provided by Yu et al. [40], [23] Nicolai et al. based on FRCNN.

To evaluate the performance of the detectors for the task of tracking, we evaluate them using all bounding boxes considered for the tracking evaluation. The iou_cost is used to calculate the IOU distance matrix between track and detection. Tracking results have been evaluated at three different IOU thresholds i.e. 0.10, 0.30, and 0.50. By evaluating the results at these different IOU thresholds, it was found that the tracker at the lowest threshold i.e. 0.10 has the highest MOTA and vice versa.

The detector provided by Yu et al. [40], [23] Nicolai et al., has provided the best performance in terms of accuracy.YOLOv4 at two different IOU Threshold i.e. 0.10 and 0.30, has the next highest accuracy compared to other detectors with the best MOTA being 38.30% and 33.60% respectively. Only at IOU Threshold, 0.50 YOLOv3 has the best MOTA being 24.30%.

The proposed detector used by [23] Nicolai et al. have the best MOTA at all three different thresholds. The best MOTA is 51.10% at 0.10 IOU as compared to other detectors. This is because they have fine-tuned the CNN-model on the VGG-16 on ImageNet and have also trained on the ETHZ pedestrian dataset [41], Caltech pedestrian dataset [42], and the self-collected surveillance dataset. They adopted the multi-scale training strategy and also use skip pooling [43] and multi-region [44] strategies to combine features at different scales and levels. This kind of pipeline boosts the performance of the detectors which can fetch the maximum pedestrian despite a scale change or in-plane rotation.

In the future, we propose to fine-tune the state-of-the-art detectors to improve the MOTA further on two most popular pedestrian detection datasets: caltech-usa [42] and inria[45]. This will lead to decrease the number of False Negatives and False Positive and hence, increases the MOTA rate. We will also compare the fine-tuned detectors with other tracking algorithms other than Deep SORT [46-50].

Conflict of Interest: The authors declare that they have no conflict of interest.

Referrences:

1. Mengmeng Wang, Xiaofeng Li, Peixin Liu, Kai Xu, Zhizhong Fu, "Multiple object tracking by multi-feature combination based on min-cost network flow", Signal Processing (ICSP) 2016 IEEE 13th International Conference on, pp. 714-718, 2016.
2. Sooksatra S., Kondo T. (2013) CAMSHIFT-Based Algorithm for Multiple Object Tracking. In: Meesad P., Unger H., Boonkrong S. (eds) The 9th International Conference on Computing and InformationTechnology (IC2IT2013). Advances in Intelligent Systems and Computing, vol 209. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37371-8_33
3. Li, Z.; Tang, Q.L.; Sang, N. (2008). Improved mean shift algorithm for occlusion pedestrian tracking. Electronics Letters, 44(10), 622–. doi:10.1049/el:20080064
4. Yan Zhang, Lijie Li, "A Pedestrian Tracking Algorithm Based on Camshift and Kalman Filtering", International Journal of Science and Research (IJSR),
5. Yamamoto, S.; Mae, Y.; Shirai, Y.; Miura, J. (1995). [IEEE 1995 IEEE International Conference on Robotics and Automation - Nagoya, Japan (21-27 May 1995)] Proceedings of 1995 IEEE International Conference

on Robotics and Automation - Realtime multiple object tracking based on optical flows. , 3(), 2328–2333. doi:10.1109/robot.1995.525608
6. M.V. Srinivasan : Generalized gradient schemes for the measurement of two-dimensional image motion, Biol. Cybern., Vo1.63, pp.421-431 (1990)
7. P. Sobey and M.V. Srinivasan : Measurement of optical flow using a generalized gradient scheme, Journal of the Optical Society of America, pp.1488-1498 (1991).
8. M. Shiohara, H. Egawa, S. Sasaki, M. nagle, P. Sobey, M.V. Srinivasan : Fteal-Time Optical Flow Processor ISHTAR, Proc. ACCV, pp.790-793 (1993).
9. Y. Shirai, J. Miura, Y. Mae, M. Shiohara, H. Egawa, S. Sasaki : Moving Object Perception and Tracking by Use of DSP, Proc. CAMP, pp.251-256 (1993).
10. J. Jodoin, G. Bilodeau, and N. Saunier, "Urban tracker: Multiple object tracking in urban mixed traffic," in IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, March 24-26, 2014, 2014, pp. 885–892.
11. J. P. Jodoin, G. A. Bilodeau, and N. Saunier, "Tracking all road users at multimodal urban traffic intersections," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 11, pp. 3241–3251, Nov 2016.
12. Y. Yang and G. Bilodeau, "Multiple object tracking with kernelized correlation filters in urban mixed traffic," CoRR, vol. abs/1611.02364, 2016. [Online]. Available: http://arxiv.org/abs/1611.02364
13. O. Barnich and M. V. Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," IEEE Trans. Image Processing, vol. 20, no. 6, pp. 1709–1724, 2011.
14. T. Kroeger, R. Timofte, D. Dai, and L. V. Gool, "Fast optical flow using dense inverse search," in Proceedings of the European Conference on Computer Vision (ECCV), 2016.
15. J. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986.
16. Li, Xin; Wang, Kejun; Wang, Wei; Li, Yang (2010). [IEEE 2010 International Conference on Information and Automation (ICIA) - Harbin, China (2010.06.20-2010.06.23)] The 2010 IEEE International Conference on Information and Automation - A multiple object tracking method using Kalman filter. 1862–1866.
17. Xi Chen, Xiao Wang and Jianhua Xuan, "Tracking Multiple Moving Objects Using Unscented Kalman Filtering Techniques", 2018.
18. M. Meuter, U. Iurgel, S. Park and A. Kummert, "The unscented Kalman filter for pedestrian tracking from a moving host," 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, 2008, pp. 37-42, doi: 10.1109/IVS.2008.4621191.
19. http://www.cse.psu.edu/~rtc12/CSE598G/introMeanShift.pdf
20. https://en.m.wikipedia.org/wiki/Optical_flow
21. French Wikipedia page on Camshift.
22. Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 1960(82), pp. 35-45 doi:10.1109/ICINFA.2010.5512258
23. Nicolai Wojke, Alex Bewley, Dietrich Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric", 2017.
24. Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft, "Simple Online and Realtime Tracking", 2016.
25. S. Zagoruyko and N. Komodakis, "Wide residual networks," in BMVC, 2016, pp. 1–12.
26. R. Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448,2015.
27. S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
28. J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In Advances in neural information processing systems, pages 379–387, 2016.
29. Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Lighthead R-CNN: in defense of two-stage object detector. CoRR, abs/1711.07264, 2017.
30. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Pro-ceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.
31. J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017.
32. J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

33. A.Bochkovskiy, C.Y.Wang, H.M.Liao, and et al. Yolov4: Optimal speed and accuracy of object detection. IEEE Conference on Computer Vision and Pattern Recognition, 2020.
34. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector.In European conference on computer vision, pages 21–37.Springer, 2016.
35. Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. "MOT16: A Benchmark for Multi-Object Tracking", 2016.
36. Bernardin, Keni, and Rainer Stiefelhagen. "Evaluating multiple object tracking performance: the CLEAR MOT metrics." Journal on Image and Video Processing 2008 (2008)
37. Ristani, Ergys, et al. "Performance measures and a data set for multi-target, multi-camera tracking." European Conference on Computer Vision. Springer, Cham, 2016.
38. Lin TY. et al. (2014) Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48
39. https://github.com/cheind/py-motmetrics
40. F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Poi: Multiple object tracking with high performance detection and appearance feature," in ECCV. Springer, 2016.
41. Ess, A., Leibe, B., Schindler, K., Gool, L.J.V.: A mobile vision system for robust multi-person tracking. In: CVPR (2008).
42. Doll´ar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: CVPR (2009).
43. Bell, S., Zitnick, C.L., Bala, K., Girshick, R.B.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. CoRR (2015).
44. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware CNN model. In: ICCV (2015) https://www.ijsr.net/search_index_results_paperid.php?id=ART20171961, Volume 6 Issue 3, March 2017, 2256 – 2258.
45. http://pascal.inrialpes.fr/data/human/
46. Nazia Parveen, Ashif Ali, Aleem Ali, IOT Based Automatic Vehicle Accident Alert System, 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), pp. 330-333, 30-31 Oct. 2020, Greater Noida, DOI: 10.1109/ICCCA49541.2020.9250904.
47. Iftikhar Husain, Aleem Ali, Fuzzy Matrix Approach to Study the Maximum Age Group of Stressed Students Studying in Higher Education, International Journal on Emerging Technologies, 12(1), pp. 31-35, 2021.
48. Aleem Ali, Neeta Singh "M/M/1/n+Flush/n model to enhance the QoS for Cluster Heads in MANETs" published in "International Journal of Advanced Computer Science and Applications (IJACSA)", U.K. 2018.
49. Rasmeet Kaur, Aleem Ali, A Novel Blockchain Model for Securing IoT Based Data Transmission, International Journal of Grid and Distributed Computing, Vol. 14, No. 1, pp. 1045-1055 1045, May 2021.
50. Aleem Ali, Pooja Malik, Conditional GANS as a Solution to Image to Image Rendering Problems, IT in Industry, Vol. 9, No.2, pp. 1106-1111, 13 April 2021.