Comparing Trackers for Multiple Targets in Warehouses

Syeda Fouzia and Reinhard Klette

School of Engineering, Computer, and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand

Abstract

Multiple object tracking is a complex and challenging computer vision problem. In industrial premises like warehouses, a robust multiple object tracking framework could provide useful information on lift truck and pedestrian movements. This information could be utilized for process improvement. It could also potentially promote greater safety awareness in the facility. We evaluate and analyze selected model-based and deep feature-based tracking mechanisms suitable for warehouse environments. Objects (or targets) are forklift trucks or people. Our experimental comparison and discussion facilitates useful insights into the design of robust multiple target trackers in warehouses, their limitations, and future research directions.

Keywords: Warehouses, Computer vision, Tracking, Forklift-trucks, Pedestrians, Deep-learning, Model-based methods

1. Introduction

Tracking the objects of interest over time is a widely studied computer vision problem. It involves two essential phases: localizing the targets in a video frame and associating each target to a unique identity over time. Comparatively Single object tracking (SOT) involves fewer tasks than multiple object tracking (MOT). MOT includes creation of new track identities, an optimum association measure for track assignment, matching of detection hypotheses to existing tracks based on cost affinity, deletion of a lost track when an object leaves the field of view, output of tracking quality and efficiency, just to name a few challenges.

For warehouses, multiple targets of interest for this study are *pedestrians* and *forklift trucks*, sharing the same warehouse premises. Refer to Figure 1(b), for challenging warehouse scenes. Unlike SOT, the number of objects might vary over subsequent frames in MOT.

The visual representation of a target to be tracked is encoded in an appearance model (i.e. categories of unique features). Typically, any tracking system comprises three main components; target visual representation, its dynamic modelling and a search mechanism employed to look for the matching candidate detection hypothesis (based on some statistical similarity measure). Normalised cross-correlation (NCC) for affinity computation based on raw pixel templates, or the Bhattacharyya distance measure for color histogram models are examples. Target appearance modelling is a vital step as the target may move in various challenging scenarios including clutter, full or partial occlusions, illumination changes or shape deformations.

The respective motion models encode object dynamics to predict target location in subse-

Received: Aug. 22, 2019
 Revised : Sep. 15, 2019
 Accepted: Sep. 20, 2019

Correspondence to: Syeda Fouzia
 (syeda.fouzia@aut.ac.nz)
 ©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Figure 1. Scenarios: (a) possible blind spot, (b) four challenging warehouse scenes, and (c) pedestrian and forklift in a warehouse.

quent frames. Either linear motion models (i.e., constant velocity or constant acceleration assumption) or non-linear motion models are employed. Non-linear motion models are used to represent the complexity of real-world motion. Some composite model-based tracking aims at striking a balance between motion and appearance modelling.

Motivated by the desire to track targets in warehouse scenes, it is essential to use a feature encoding scheme that is suited to match complex industrial environments. According to our analysis, Table 1 lists vital challenges encountered, together with possible solutions.

Based on how objects are initialized, the MOT problem can be classified into two categories. In *detection-based tracking*, objects are first detected and linked into trajectories to form corresponding tracks. This strategy might involve detecting objects based on training an object detector in advance or detecting motion blobs (background modelling). In a background modelling-based tracking approach, a region of interest (ROI) is detected by finding a background representation and then finding deviations from it, for each incoming frame. In *detection-free tracking*, targets of interest are initialized in a first frame and localized in subsequent frames. Processing of the frames is done either *online* using the detection hypothesis from the current frame or *offline* which is based on collection of hypotheses from all the frames in advance to estimate the output track [1].

With advances in computational resources efficiency and availability of bulks of training data, deep learning-based architectures have been used to solve complex computer vision tasks. Representation modelling of target appearance using *deep features* is used as an effective cue in many recent tracking frameworks. Many such methods proved to be robust in tracking under occlusions and varying scales of objects. These features are extracted from a deep neural network (trained for

Table 1. MOT challenges in warehouses

Challenges	Possible solution
Clutter	Using target representation which is robust to identify truck or a person from the background clutter.
Illumination	Using descriptors that are less sensitive to scene illumination changes.
Re-entering objects	Using a sub-module that can handle the truck or person re-identification problem and is able to preserve their identity, after re-entering the scene.
Occlusion	Using an approach robust to occlusions and able to separate the occluder and the actual target.
Similar looking	Sophisticated dynamic object appearance model, might use multiple cues like appearance, shape, colour or texture to handle; the trucks or pedestrians that appears alike.
Scale	Adapting the corresponding truck or person representation dynamically to make it robust to scale changes.

image recognition or classification task) and used in places of conventional hand-crafted features in existing MOT paradigms. This approach was already implemented using two famous deep network architectures (i.e. Faster RCNN and Yolo) in our previous work; see [2, 3].

For this study, a comparative experimentation analysis is performed. A model-based moving multiple object tracking system using the Gaussian mixture model (GMM) is implemented in our recent work. The GMM is a semi-parametric, high-dimensional density estimation technique. It is used to model complex and non-linear system processes. Model parameters are estimated adaptively with memory efficiency. Moving forklift trucks and pedestrians are segmented on pixel-level into foreground and background regions. Tracking validated motion blobs over time is done using Kalman filtering (constant velocity model). Background and foreground regions are updated, frame by frame, based on motion statistics. For the improvement of resulting foreground maps, pixel saliency based improvement is employed [4].

The paper is structured as follows. Section 1 provides an introduction for this study, warehouse scene challenges and MOT problem formulation in warehousing context. Section 3 describes the motivation and how this study will benefit warehouse process improvements and increase work safety. Section 4 details a few analyzed hand-crafted features schemes. Section 5 outlines different ways of deep neural nets, used to solve MOT problems. Section 6 details a detailed quantitative and qualitative analysis on both groups of methods; applied for warehouse image data. Section 7 concludes the study.

2. Related Work

Hand-crafted features have been used to encode the appearance of targets of interest. Feature descriptor schemes might include optic flow, scale invariant feature transform (SIFT) [5] or gradient based-features, histogram of oriented gradients (HOG) [6], colour, bag-of-features [7], or deformable part based models [8], classic classifiers such as support vector machine (SVM) and random forests. Probabilistic occupancy map (POM) and depth features have also been studied. Other MOT algorithms might use multiple cues to represent application-specific objects of interest.

Based on the used search mechanism, tracking can be categorized into either deterministic or probabilistic approaches. Probabilistic inference is based on a representation of object states as a distribution with some degree of uncertainty. Based on current observations, tracking algorithms are used to estimate the probabilistic distribution of target states. These approaches are based on past and current observations and are most suitable for online tracking. Kalman filter, extended Kalman filter and particle filter frameworks come under this

category [9]. Deterministic optimisation approaches aim to estimate the maximum-a-posterior (MAP) solution for the tracking problem. Observations from a specific time window for all the frames are needed in advance to estimate the trajectory of the target by associating them in a global manner. These methods are more suitable for offline tracking applications. Bipartite graph matching, dynamic programming, min-cost max flow network flow, conditional random field and maximum weight independent set (MWIS) are examples [10, 11].

Deep learning models have been employed to solve MOT problems in a variety of ways. We divide the way deep models have enhanced the solution of MOT problems into three streams.

Using deep network features. Deep features being more discriminative than hand-crafted conventional features; they are used to replace them inside the same MOT framework to enhance the performance of MOT. Training a convolution neural net (CNN) model can be done by using extensive classification datasets already available, or by using tracking data, or pre-training and fine tuning the model. Fully trained CNNs are employed for region proposal generation for object detection [12]. Siamese CNN architecture is employed to learn the matching features for multiple object tracking problems. et al. [13] formulated a deep architecture by fusing deep features and a motion prediction algorithm; a linear programming approach is used to solve the tracking problem [13]. Since optic flow is a useful feature to learn track association, learning deep flow through deep architecture made it more efficient and accurate for MOT. Similarly, pair-wise images are fed to a siamese network to compute cost affinities for tracks. Xiang et al. [14] proposed to learn a triplet-loss based CNN to find the distance metric between trackers and detections. In the tracking process, this distance metric between trackers and detections constructs the cost affinity for bipartite graphs, which is solved by the Hungarian algorithm.

Using core modules learned by a deep neural network. If core MOT modules and processes are learned using a deep neural network, the tracking performance is improved. The study splits the domain into:

Discriminative network learning is known as a tracking-by-detection approach. For instance, a target specific particle filter framework is employed. Features from Faster-RCNN VGG-16 are used, where the top and bottom layer output is used to obtain weights for the particles. A resulting track is estimated through

a particle filter framework [15, 16].

Deep metric learning for MOT is learning affinity or a distance measure of detection pairs for the tracklet association problem. One of the main processes in MOT is the association of detection hypotheses over subsequent frames with a correct track. An optimum association measure is constructed by using learned appearance features from a person re-identification dataset [17, 18]. To handle track identity switches, an *appearance saliency map guided data association measure* is exploited to verify the track identity. A saliency distribution dissimilarity measure between a detected ROI and predicted candidate track locations was described by the Bhattacharyya coefficient in our previous work.

Extending deep models for generative learning is used to learn MOT vital parameters for data distribution and can be learned through generative adversarial networks (GAN). Fang et al. proposed to model object motion and appearance using a posterior probability Gaussian distribution, using an auto-linear regression method. A generative long short-term memory (LSTM) model is used to generate a confidence map output. A pixel-wise probability map is generated through a decoder following an LSTM layer [19].

Using an end-to-end deep network. This is applied to model the whole process of target tracking. Since many related sub-processes are intertwined inside the framework, the model relies on few assumptions like Markov properties, or fixed distribution. For the online MOT problem, a recursive Bayesian filter comprises prediction and update modules. Object states, observations, matching matrix and existing probabilities are fed as input into the recurrent neural net (RNN). Predicted states, updated results and existence probabilities are the output. A set of LSTMs is used to compute a matching matrix. Either one is used to find a match among one of the object states and the current observations. Multiple tracking segments are used to train a group of LSTMs and an RNN [20]. Deep networks are also used to learn regression models in SOT and also a few instances in MOT for regression learning. In comparisons with CNNs, RNNs are more suitable for target sequence modelling and for predicting the target's next state according to its historical information. However, to simplify learning the appearance features, some existing CNNs are used to extract deep features and fed as input for RNNs [21].

A typical warehouse scene. A typical warehouse scene is comprised of blocks of goods stacked in storage racks and

shelves forming aisles; concrete floor stripped into zones using safety tapes of different color codes distinguishes work zones. Components are forklift traffic lanes, pedestrian crossings, equipment placement or guard rails for safety and the *warehouse furniture* including pallets, racks, instruction signs (directing traffic and hazard warnings), with various types of *obstacles* like other forklifts trucks, pedestrians or parked trucks. Figure 1(b) is depicting four warehouse scenes.

Multiple object tracking problem formulation. A monocular camera C_1 is fixed at a main location in a warehouse capturing images (i.e. video frames) denoted as $I_1, I_2, I_3, \dots, I_N$. Every frame has varying number of m multiple objects, where these objects are described by notation $A_1, A_2, A_3, \dots, A_m$ and named by their respective class identifier (in our case it is either a person or a truck). Each frame can have either a person/truck or both classes of multiple objects present. Number m of objects in every frame can vary as new objects may enter and old objects may leave the scene.

We denote a state vector S_i to represent locations for m multiple objects $A_1, A_2, A_3, \dots, A_m$ seen in frame i , i.e. $S_i = (l_i^1, l_i^2, l_i^3, \dots, l_i^m)$. We denote the sequential locations of those multiple objects in a range as $S_{1:t} = \{S_1, S_2, S_3, \dots, S_t\}$.

Also, $l_{s:f}^j = (l_s^j, \dots, l_f^j)$ denotes the initial to final sequential locations of the j -th object in a range s to f (where it may not be the single frame range and may involve split of ranges, as target may leave the scene and reappear at a later stage). Moreover, using tracking by detection approach, we measure the states of these objects at every frame using some detection algorithm. We denote the collected observations from multiple objects for frame i as $O_i = (o_i^1, o_i^2, o_i^3, \dots, o_i^m)$. Sequential observations of those multiple objects over 1 to t frames are in a range $O_{1:t} = \{O_1, O_2, O_3, \dots, O_t\}$.

The objective of multiple object tracking is to estimate, for every corresponding j -th object belonging to frame i , i.e. A_i^j , a location l_i^j for all sequential frames in which it appears, which should be as close to the ground truth location as possible (i.e. $[l_i^j]^{gt}$). Depending on the approach, this location estimate could be the 2D centroid location of the target (x, y) or it could be the world coordinates (x, y, z) if a target is mapped from 2D into the 3D world. When seeking a solution to the multiple object tracking problem, some physical constraints need to be considered to avoid conflicts or collisions. Two different objects cannot occupy the same physical space in the real world. Since we deal with multiple detection hypotheses, a constraint is that in the same frame, two detection-hypotheses cannot be assigned

to the same track.

3. Motivation and Significance of the Tracking Framework

Human detection for forklift applications will help to improve work safety levels in warehouses [22]. A robust computer vision-based MOT framework will track the pedestrians in various poses like standing, sitting, walking, facing backwards, possibly lifting goods of various sizes and shapes – all of which may be partially occluded by background clutter. Typically, forklift trucks operate in constrained environments and their motion may include frequent accelerations, decelerations, reversing and turns in the vicinity of other trucks or pedestrians [23].

Through the installation of cameras inside warehouses, the tracked movement of pedestrians and lift trucks could be utilized in a variety of ways. Tracking multiple lift trucks, and hence the associated goods, can aid in product complaint handling applications and recording any deviations. The number of objects, either trucks or persons passing within the field of view, will benefit the processes involving inventory counts. It will also help in identifying abandoned or misplaced objects.

The detection of the various categories of lift trucks (reach trucks, stock pickers, etc.) could contribute in optimising resource usage inside warehouses. Warehouses and manufacturing plants can be cluttered and busy, with trucks and pedestrians working in close proximity. Racking infrastructure can make blind spots as shown in Figure 1(a).

Needless to say that requirements for deployment of robust tracking modules for these vehicles are challenging. But taking an initiative in the exploration of multiple object tracking problems in warehouses and experiments with various state of art frameworks will open new avenues of research into industrial process improvements. Moreover, such steps will help to get an insight into potentials of computer vision algorithms and deep learning models in warehouses.

4. Model-based GMM Multi-object Tracking

To model object appearance in warehouses, a study is performed for various cues or features that can be employed for targets. Table 2 enlists the studied observations regarding stated feature schemes. It was concluded that an adaptive detection algorithm is needed which can detect pedestrians working behind the warehouse clutter, which are often occluded and challenging to detect.

Table 2. Examples of appearance modelling schemes

Features	Challenges
Gradient-based feature	Can encode target shape robust to illumination variations but cannot handle occlusions and shape deformations.
Colour-based feature	For instance color-histogram is discriminative but spatial contents of the image is ignored.
Local features	Out of plane rotation is a challenge but robust to shape deformation (optic flow, KLT).
Region features	Involve wider region but computationally expensive (co-variance matrix).
Depth feature	Depth measurement needs multiple views for the same scene for depth computation.

Due to forklift trucks with loading and unloading pallets, people working in aisles might be occluded partially with various poses. It is vital to handle illumination variations, repetitive motions from clutter, and long-term scene changes. A GMM based approach to detect pixel-wise moving segmented foreground and background regions is selected for target detection and onward tracking.

For each video frame $I_1, I_2, I_3, \dots, I_t$, the value of every pixel over time is modelled as a mixture of adaptive Gaussian distributions. A pixel process of a pixel (x, y, X) has the history of its previous t values, say from Frame 1 to Frame t . This can be represented by the set $\{X_1, \dots, X_i, \dots, X_t\}$, where $1 \leq i \leq t$.

The probability of observing a specific mixture component at Frame t is given by products of probability density functions with their weight. For more than one density function, we have a multivariate case, such as

$$P(X_t) = \sum_{i=1}^k w_{i,t} \cdot G(X_t, \mu_{i,t}, \sigma_{i,t}). \quad (1)$$

Here, $w_{i,t}$ is the weight of the i -th Gaussian distribution at Frame t . The method for updating adaptive mixtures of multimodal Gaussian distributions and heuristics for determining the background model is done as in [24]. The learning rate was set adaptively according to sequence challenging attributes.

Figure 2. For three sequences, resulting tracked bounding boxes are shown. Top row depicts the tracked bounding boxes for each sequence and bottom row shows the corresponding foreground regions.

Once foregrounds are obtained by the GMM method, they are postprocessed to refine the quality of foreground regions. Subsequent saliency-maps for this purpose are used. The resultant foreground quality is optimal. These selected motion blobs were tracked by a linear Kalman filter. Refer to Figure 2 for qualitative results.

5. Deep Learning-based Multi-object Tracking

For training the first tested deep model Faster-RCNN, a pre-trained Alex-Net model is used [25]. It is a 25 layered architecture pre-trained on 1,000 image categories. We re-trained the last three layers of the model with acquired forklift truck recorded data. During model training, image patches are extracted from training data. The two vital training parameters, the positive overlap range and the negative overlap range, control which image patches are used for training. The best values for these parameters are chosen by testing the trained detector on a validation data set (set separate from training and testing dataset). Detection results are used to initialize a Kalman filter, used for the prediction of an object's future location. Based on localization coordinate feedback from the Faster-RCNN trained

network, the Kalman filter corrects the predicted trajectory. We also improve the trajectory results by using a saliency map of the ROI (as detected by Faster-RCNN). Using an ROI saliency map [26], the Kalman filter performs state correction for every frame. The resulting trajectory is improved and close to ground truth trajectory.

A detection association method is used based on the Hungarian algorithm in which for every subsequent frame, new RCNN detections are assigned to corresponding tracks [27]. The assignment cost is minimised and calculated based on the Euclidean distance d between the centroids of each pair of bounding boxes for the targets [28]:

$$d = \|p_c(j) - p_c(i)\|_2, \quad (2)$$

where $p_c(i)$ and $p_c(j)$ are centroids of boxes i and j , respectively. A maximum distance threshold is used to rate the assignment such that matches are discarded if $d > d_{max}$. With new detections being assigned to existing tracks, the track is updated by estimating the state using the new observation. If a track is not assigned (i.e. a new detection in the current frame), the new bounding box is predicted by the Kalman filter. Matches between new detections in two consecutive frames create a new track. In case of false detections, showing up for a few frames, or no detections associated to a track for threshold track age, the track is deleted [29].

For training the second model 30, CNN weights that are pre-trained on the ImageNet dataset are used, i.e. weights from the DarkNet53 model [30]. Our self-collected warehouse image dataset, including 3,077 training images with manually labelled ground truth bounding boxes for forklift trucks and pedestrians, defining 8,242 ROIs, was annotated. Each image contains 1 – 7 labelled instances of a forklift truck or pedestrian. Subsets of image data from the Caltech pedestrian [31] and INRIA pedestrian datasets [32] are also added to obtain sufficient training samples for pedestrians. The pedestrians present in these subsets, with different degrees of occlusion, wearing different costumes, have many kinds of scales and changing postures.

DarkNet-53 model inputs a text file for each image and every ground truth object in the image in format $x, y, width, height$ where these parameters are relative to the image's width and height. A text file with names and paths of the images to be trained was supplied to DarkNet to load the images to be trained on. A test text file for testing and validation text file for validation images is also created. After training the pretrained DarkNet53 model for about 20k iterations (13,500 images), a

Table 3. Sequences used for comparison

	Attribute	Frames	Tracks
Seq1	Low illumination areas	150	7
Seq2	Occlusions	614	21
Seq3	Cluttered areas	614	9

validation accuracy of 91.2% and testing accuracy of 90% is obtained. This was without any data augmentations or other transformations. A GeForce GTX 1080 Titanium GPU with compute capability 6.1 and 12 GB memory is used for model training.

For the detection to track associations, two distance measures were employed, the Mahalanobis distance and the Bhattacharyya distance. The Mahalanobis distance, which is the squared distance between predicted object states (by a Kalman filter) and incoming new measurements, can be written as follows:

$$\mathbf{d}^1(i, j) = (d_j - y_i)^T \mathbf{S}_i^{-1} (d_j - y_i). \quad (3)$$

We denote by $\mathbf{p}^*$ the reference track saliency-color model; $\mathbf{p}(x_t)$ is the candidate detection model. Then, the distance between the two is described by the Bhattacharyya distance (bdist) [33] and is described by

$$\begin{aligned} \mathbf{d}^2(i, j) &= \text{bdist}[\mathbf{p}^*, \mathbf{p}(x_t)] \\ &= \left[1 - \sum_{u=1}^M \sqrt{\mathbf{p}_u^*(x_o) \mathbf{p}_u(x_t)} \right]^{\frac{1}{2}}. \end{aligned} \quad (4)$$

6. Experimental Comparison and Discussion

We perform an experimental comparative study for model based GMM Kalman tracker and tracking by detecting using two deep learning models. Recorded video data using four monocular uncalibrated cameras installed in a real production warehouse are used for deep learning model training for this work. For a better analysis of the strength and the weakness of the tracking algorithms, we categorised the sequences according to the three attributes which are low illumination areas, occluded scenes and cluttered warehouse areas (Table 3).

There is no single approach that will handle all of these challenging scenarios fully successfully. We evaluated the results under considered attributes for each technique separately. It helped to obtain an insight into methods suitable according to scene statistics. A ground-truth target location dataset for

Figure 3. Top row depicts the trajectory computed by Faster-RCNN based tracker. Bottom row, (left) saliency map with detected centroid in red, (middle) trajectory estimated by Kalman, (right) ground truth trajectory.

Figure 4. Multiple target tracking. Left to right. Frames 35, 54 and 88 using 30 tracker with bounding box shown for each tracked target.

corresponding target's ground-truth positions and ground-truth trajectories are formulated. For fair comparisons, the parameters of each tracker were fixed for all the considered sequences. Refer to Figure 2, Figure 3 and Figure 4 for qualitative results for GMM, Faster-RCNN and 30 based trackers.

Many vital factors such as target position accuracy, robustness over different appearance changes, tracking efficiency, memory consumption, and ease of usage are studied. Even in one frame with the tracking output and ground-truth object state, there can be several metrics to measure accuracy. For the multiple object tracking problem, quantitative performance is mainly evaluated using two important evaluation measures, i.e. multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) [34, 35]. MOTA takes into account all configuration errors made by the tracker (i.e. false-positives, misses, number of mismatches), averaged over all frames:

$$\text{MOTA} = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t}, \quad (5)$$

where FN_t is the number of false-negatives or misses, FP_t is the number of false-positives, and $IDSW$ is the number of ID mismatches/switches, where t is the frame index and GT is the number of ground truth objects. The MOTP is the average dissimilarity between all true-positives and their corresponding

Table 4. Average quantitative comparison for model and deep-learning based tracker

Average quantitative evaluation results for three sequences										
Technique	FPS	FNs	MOTA	MOTP	FAF	MT	ML	ld-sw	frag	Comments
GMM-based	469	236	28.2	26.9	5.7	21.5	34.4	432	411	Target detection efficiency behind cluttered warehousing scenes is comparable to Faster-RCNN detection outputs for the same. The reason that even hidden segmented motion blobs are detected with GMM but might be missed by deep models. To make the algorithm adaptive to scene changes and to prevent merging in the background, parameter tuning is very important. Adaptive parameter adjustment to handle these challenges might make model based GMM method potentially comparable to any deep model based detection tracker.
Faster-RCNN based	444	394	47.1	42.6	5	23.5	19.1	254	336	The model was prone to miss-classification errors due to lack of training data. Miss-classification errors in sequence 2 and 3 i.e., occluded and cluttered scenes, caused many identity switches in tracking outputs. Using non-linear motion model i.e., extended Kalman filter will benefit the robustness.
30 based	174	109	59.9	54.1	1.1	35.1	16.2	231	156	Number of miss-classifications were minimal. Target trajectory is mostly tracked with yolo-based tracking output. Incorporating appearance cue information for track re-identification; once target leave the camera FOV or scene improved the detection accuracy and reduced track fragmentations. Adding more training data can make the framework robust to miss-classifications.

ground truth targets:

$$\text{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}, \quad (6)$$

where c_t denotes the total number of matches in frame t and $d_{t,i}$ is the bounding box overlap of target i with its assigned ground truth object.

Quantitative comparison results have been summarised in Table 4 for each method for the three considered sequences. Targets are consistently tracked over large parts of Sequences 1 and 3 for the model-based GMM method and over considerable parts of Sequences 1, 2 and 3 using tracking by deep model-based detection. Table 5 lists the observations regarding model and deep-learning based tracking methods. It also shows the quantitative comparison of the two category of methods com-

pared based on MOT evaluation metrics. It can be seen that 30 based tracking by detection techniques gave the best tracking quality and best quantitative results for the considered warehouse scenes.

7. Conclusions

A comprehensive evaluation of model and deep learning-based tracking frameworks in industrial environments is carried out. When supplied with a bulk of training data, deep models are better at target localization accuracy and avoiding tracking errors than model-based methods. A single trained model might be good enough for only specific custom tasks it is trained for. Moreover, it cannot handle video scenarios with a variety of challenging attributes for the same warehouse environment.

Table 5. Comparison of tracking results on three challenging sequences using MOT evaluation measures [34]

Sequence	GMM-based			Faster-RCNN based			30 based		
	#1	#2	#3	#1	#2	#3	#1	#2	#3
FPs	413	784	211	389	401	544	165	203	155
FNs	163	312	234	213	503	466	76	98	154
MOTA	45.1	39.4	49.9	50.3	47.8	43.2	61.2	59.8	58.8
MOTP	39.6	1.3	39.8	41.3	45.3	41.2	53.1	55.2	54.2
FAF	4.5	9.3	3.3	3.6	4.1	7.5	1.7	0.8	1
MT	21.8	19.8	23	24.5	25.3	20.8	32.4	36.6	36.5
ML	32.7	33.5	37	19.6	16.7	21	18.9	21.9	25.3
ld-sw	173	453	233	154	123	233	101	176	133
frag	254	189	203	156	342	511	103	132	233

A model trained for one set of attributes and video frames might not be good enough for a set of others. Robustness is a relative measure and highly variable as per the scene statistics. Using deep features and models for simplifying vital tracking sub-tasks other than detection and localization is also valid. It might include cost affinity computation and handling track re-identification problem in MOT.

Model based methods are better at finding regions of interest but comparably “see” less variables according to scene attributes. Possibly, a hybrid model and deep feature-learned network can work best in warehouse’s challenging scenarios. We assume that an optimum quality of MOT results will lay the ground for subsequent higher-level computer vision tasks such as people action recognition and process improvement. Furthermore, we suggest that such a detection and tracking framework would be useful for increasing work safety at the warehouses.

Acknowledgment

Authors acknowledge support by Crown Equipment Limited, Auckland, for collecting used sequences and for problem discussions.

References

- [1] W. Luo, J. Xing, A. Milan, Z. Zhang, W. Liu, Z. Zhao, and T. K. Kim, “Multiple object tracking: a literature review,” 2014, Available <https://arxiv.org/abs/1409.7618>
- [2] S. Fouzia, M. Bell, and R. Klette, “Tracking of load handling forklift trucks and of pedestrians in warehouses,” in *New Trends in Computer Technologies and Applications*. Singapore: Springer, 2018, pp. 688-697. https://doi.org/10.1007/978-981-13-9190-3_76
- [3] S. Fouzia, M. Bell, and R. Klette, “Saliency guided data association measure for multiple object tracking,” in *Proceedings of the 16th International Symposium on Pervasive Systems, Algorithms, and Networks*, Naples, Italy, 2019.
- [4] S. Fouzia, M. Bell, and R. Klette, “Deep learning-based improved object recognition in warehouses,” in *Image and Video Technology*. Cham: Springer, 2017, pp. 350-365. https://doi.org/10.1007/978-3-319-75786-5_29
- [5] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- [6] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of 2005 International Conference on Computer Vision and Pattern Recognition (CVPR’05)*, San Diego, CA, 2005, pp. 886-893. <https://doi.org/10.1109/CVPR.2005.177>
- [7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” in *Workshop on Statistical Learning In Computer Vision (ECCV)*, 2004.
- [8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained

- part-based models,” *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, 2009. <https://doi.org/10.1109/TPAMI.2009.167>
- [9] Z. Han, Q. Ye, and J. Jiao, “Online feature evaluation for object tracking using Kalman filter,” in *Proceedings of 2008 19th International Conference on Pattern Recognition*, Tampa, FL, 2008, pp. 1-4. <https://doi.org/10.1109/ICPR.2008.4761152>
- [10] H. Jiang, S. Fels, and J. J. Little, “A linear programming approach for multiple object tracking,” in *Proceedings of 2007 IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007, pp. 1-8. <https://doi.org/10.1109/CVPR.2007.383180>
- [11] P. Lenz, A. Geiger, and R. Urtasun, “FollowMe: efficient online min-cost flow tracking with bounded memory and computation,” in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 4364-4372. <https://doi.org/10.1109/ICCV.2015.496>
- [12] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-CNN: towards real-time object detection with region proposal networks,” *Advances in Neural Information Processing Systems*, vol. 28, pp. 91-99, 2015.
- [13] L. Leal-Taixe, C. Canton-Ferrer, and K. Schindler, “Learning by tracking: Siamese CNN for robust target association,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Las Vegas, NV, 2016, pp. 33-40. <https://doi.org/10.1109/CVPRW.2016.59>
- [14] J. Xiang, G., Zhang, J. Hou, N. Sang, and R. Huang, “Multiple target tracking by learning feature representation and distance metric jointly,” 2018, Available <https://arxiv.org/abs/1802.03252>
- [15] L. Chen, H. Ai, C. Shang, Z. Zhuang, and B. Bai, “Online multi-object tracking with convolutional neural networks,” in *Proceedings of 2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, pp. 645-649. <https://doi.org/10.1109/ICIP.2017.8296360>
- [16] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, “Deep affinity network for multiple object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. <https://doi.org/10.1109/TPAMI.2019.2929520>
- [17] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proceedings of 2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, pp. 3645-3649. <https://doi.org/10.1109/ICIP.2017.8296962>
- [18] K. Yoon, D. Y. Kim, Y. C. Yoon, and M. Jeon, “Data association for multi-object tracking via deep neural networks,” *Sensors*, vol. 19, no. 3, article ID. 559, 2019. <https://doi.org/10.3390/s19030559>
- [19] Y. Xu, X. Zhou, S. Chen, and F. Li, “Deep learning for multiple object tracking: a survey,” *IET Computer Vision*, vol. 13, no. 4, pp. 355-368, 2019. <https://doi.org/10.1049/iet-cvi.2018.5598>
- [20] K. Fang, Y. Xiang, X. Li, and S. Savarese, “Recurrent autoregressive networks for online multi-object tracking,” in *Proceedings of 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, 2018, pp. 466-475. <https://doi.org/10.1109/WACV.2018.00057>
- [21] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, Available <https://arxiv.org/abs/1409.1556>
- [22] V. Danielsson and G. Smajli, “Improving warehousing operations with video technology,” Master’s thesis, Lund University, Sweden, 2015.
- [23] M. Drulea, I. Szakats, A. Vatavu, and S. Nedeveschi, “Omnidirectional stereo vision using fisheye lenses,” in *Proceedings of 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj Napoca, Romania, 2014, pp. 251-258. <https://doi.org/10.1109/ICCP.2014.6937005>
- [24] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proceedings of 1999 International Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, Fort Collins, CO, 1999, pp. 246-252. <https://doi.org/10.1109/CVPR.1999.784637>
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012.

- [26] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," *Advances in Neural Information Processing Systems*, vol. 20, pp. 545-552, 2007.
- [27] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83-97, 1995. <https://doi.org/10.1002/nav.3800020109>
- [28] T. Cane and J. Ferryman, "Saliency-based detection for maritime object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Las Vegas, NV, 2016, pp. 18-25. <https://doi.org/10.1109/CVPRW.2016.159>
- [29] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Proceedings of 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, 2017, pp. 1-6. <https://doi.org/10.1109/AVSS.2017.8078516>
- [30] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, Available <https://arxiv.org/abs/1804.02767>
- [31] Caltech pedestrian detection benchmark, http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/index.html
- [32] INRIA person dataset, <http://pascal.inrialpes.fr/data/human/>
- [33] D. Sidibe, D. Fofi, and F. Meriaudeau, "Using visual saliency for object tracking with particle filters," in *Proceedings of 2010 18th European Signal Processing Conference*, Aalborg, Denmark, 2010, pp. 1776-1780.
- [34] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: a benchmark for multi-object tracking," 2016, Available <https://arxiv.org/abs/1603.00831>
- [35] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the CLEAR MOT metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, article no. 246309, 2008. <https://doi.org/10.1155/2008/246309>

Syeda Fouzia is a Ph.D. student at the Department of Electrical and Electronic Engineering, in the School of Engineering, Computer and Mathematical Sciences at Auckland University of Technology, New Zealand. Her research interests include computer vision and machine learning.

E-mail: syeda.fouzia@aut.ac.nz

Reinhard Klette is a Fellow of the Royal Society of New Zealand and a professor at Auckland University of Technology. He (co-)authored more than 400 publications in peer-reviewed journals or conferences, and books on computer vision, image processing, geometric algorithms, and panoramic imaging. He was an associate editor of IEEE PAMI (2001-2008).

E-mail: rklette@aut.ac.nz