Multimedia Tools and Applications (2023) 82:12401–12422
https://doi.org/10.1007/s11042-022-13780-5

UMTSS: a unifocal motion tracking surveillance system for multi-object tracking in videos

Soma Hazra[1] · Shaurjya Mandal[1] · Banani Saha[1] · Sunirmal Khatua[1]

Received: 21 October 2021 / Revised: 4 August 2022 / Accepted: 5 September 2022 /
Published online: 17 September 2022

Abstract

Multiple object detection and tracking play a very crucial role in solving several elementary problems in real-time surveillance video analysis and computer vision. However, it is a challenging problem because real-time surveillance videos are typically affected by a variety of adverse environmental effects. In this work, we propose a novel surveillance framework, called a unifocal motion tracking surveillance system (UMTSS), for multi-object tracking in real-time videos. The proposed UMTSS combines two significant steps. First, a Faster-RCNN with inception-v2 model is employed here to detect multi-objects efficiently in each video frame. Then, a unifocal feature-based KLT (Kanade-Lucas-Tomasi) method is proposed for tracking objects across the video frames based on region proposals generated by the object detector in the previous phase. Also, we have proposed a new tracking parameter, called dynamic tracking accuracy (DTA), to quantify the performance of the tracking algorithms. The performance of our UMTSS has been evaluated on five standard crowd video databases, namely CrowdHuman, PETS, UCSD, AGORASET and CRCV, and compared with state-of-the-art methods in terms of different qualitative and quantitative measures. It has been observed that our UMTSS outperforms the state-of-the-art methods.

Keywords Surveillance system · Multiple object detection · FRI · Tracking · Unifocal features · UFOT-KLT

✉ Banani Saha
bsaha_29@yahoo.com

Soma Hazra
soma.hazra.frnd@gmail.com

Shaurjya Mandal
cannonofspirit@gmail.com

Sunirmal Khatua
enggnimu@gmail.com

[1] Department of Computer Science and Engineering, University of Calcutta, Kolkata, India

1 Introduction

The crowd stampedes and terrorist attacks in public places have now become more serious and dangerous threats due to the rapid increase in the population and scale of cities. Thus, in the age of increasing security needs, crowd monitoring has become more important. Because of the high generation risk, the security of crowd events has been a top priority for the concerned authorities. Nowadays, all crowded public areas are under video surveillance to prevent anomalous or abnormal activities. Each movement of an object, such as a person, vehicle, or animal, is monitored thoroughly for 24 h by the security personnel. However, the continuous monitoring of crowd events is a very difficult and tiresome task for humans, and there may be a possibility of misrecognition. Therefore, automation of the surveillance system is the need of the hour. As a result, research in this domain becomes popular, and much work has been reported in the literature [34] in recent years.

The intelligent monitoring system must be efficient enough to detect crowd movements and let the authorities know if any unintended situation is found. Typically, the functionalities of an intelligent crowd surveillance system can be categorized into three main sections: crowd detection, crowd movement tracking, and abnormal activities detection. Object detection in videos is the first and foremost step of any intelligent surveillance system. After object detection, the next task is to track and monitor the movement of the object in a given scenario. Tracking an object allows us to generate an object's trajectory over time by locating its position in each frame of the video, which can then be used to analyse object behaviour. Over the past years, various methods for object detection, tracking, and activity analysis of objects have been proposed by researchers around the globe [23, 24, 26, 34, 49]. Despite these efforts, developing an intelligence surveillance system remains a difficult task. The primary challenge in object detection and tracking is accounting for target object appearance variation caused by changes in illumination, deformation, and pose. Second, occlusion, motion blur, and camera view angle make it difficult for algorithms to track target objects. Third, some frames may have been missed or hampered due to noise or low video quality or to match the tracker's speed in a real-time scenario, which must be handled properly to track objects efficiently. Furthermore, in spatio-temporal scenarios, there may be multiple CCTV cameras that must be efficiently co-related by the object tracker.

Tracking an object in a video involves detecting the object in the first frame and predicting its state in each subsequent frame of a video sequence. Therefore, every tracking method necessitates an object detection mechanism, either in every frame or when the object first appears in the video. Besides, an object tracker aims to generate the trajectory of an object over time by locating its position in each frame of the video. Various handcrafted features and deep learning-based methods have been developed in this domain over the years [23, 24, 26, 34, 49]. However, each method has its own set of benefits and drawbacks [23, 24, 26, 34, 49]. In this work, we have combined both deep learning and handcrafted features, and proposed a novel surveillance framework, called unifocal motion tracking surveillance system (UMTSS), for multi-object tracking in real-time videos. A Faster-RCNN with inception-v2 model, named Faster-RCNN Inception-v2 (FRI), is employed here to detect multi-objects efficiently in each video frame. Here, we have innovatively used the Inception-v2 model [50] to improve the object detection rate of Faster-RCNN. The most salient feature of the Inception-v2 model is the several parallel convolutions supported by the model. This allows deep to be generated while controlling the overfitting problem. Also, inception-v2 has a lower computational cost than other top-performing successors which motivated us to use it as the backbone network of the

Faster-RCNN to detect objects more efficiently. FRI generates region proposals (bounding boxes) for each object, which are then used to track the object throughout the video. Here, a unifocal feature-based KLT method, called unifocal feature-based object tracking using KLT (UFOT-KLT), is proposed for object tracking. Typically, the traditional KLT method uses multi-feature points to track objects across the frames which leads to an increase in the execution cost and complexity of the tracking system. The proposed unifocal feature-based KLT reduces the overhead of multipoint features, and also maintains the precision of unique labeling of objects for the sequence of frames. Moreover, we have proposed a new tracking parameter, called dynamic tracking accuracy (DTA), to quantify the performance of the tracking algorithms. The system monitors a region of scenario acquired by real-time video streams from a set of CCTV cameras. As we are working with real-time video, the speed of video analysis should match the frames per second (fps) of the captured video. Thus, to get rid of overflow problems, we have selected video frames that contain maximum information without compromising the significant information. A keyframe extraction technique is implemented here using a producer-consumer technique to make the system faster. Also, it helps to maintain the balance between the flow of incoming continuous video frames and their analysis. The performance of the proposed method and its components are validated separately on challenging video sequences of four different datasets and compared with state-of-the-art related methods. In particular, the key contributions of this paper can be outlined as follows.

- *Proposed a novel surveillance framework, called unifocal motion tracking surveillance system (UMTSS), for multiple objects tracking in videos.*
- *A Faster-RCNN with Inception-v2, named FRI, is employed here for detecting multiple objects efficiently. By incorporating Inception-v2 as a feature extractor with Faster-RCNN we are able to achieve a high detection rate.*
- *Proposed a unifocal feature-based object tracking method, called UFOT-KLT, to efficiently track real-time objects.*
- *A new tracking parameter, called dynamic tracking accuracy (DTA), is proposed to evaluate the performance of the tracking algorithms.*
- *The performance of the proposed UMTSS is validated in the presence of missing frames in the given scenarios.*
- *To assess the performance of the proposed framework, we have used CrowdHuman, PETS, UCSD, AGORASET, and CRCV datasets which are publicly available crowd datasets for object detection and tracking problems. The obtained results outperform the existing related methods.*

The remainder of the paper is structured as follows. The related work in this context has been discussed in Section 2. In Section 3, the problem statement is discussed. The methodology and workflow of the proposed system have been described in Section 4. The experimental setups and results have been discussed in Section 5. Finally, Section 6 concludes the paper.

2 Related study

This section presents a brief overview of various methods for multi-object detection and tracking in real-time videos, both conventional and deep learning. Over the past years, significant works have been reported in the literature on object detection and tracking [25, 39, 40], which are briefly discussed in the following subsections.

2.1 Object detection

The method of object detection involves identifying the bounding box with the highest detection score for the given input image or video. Typically, object detection methods can be divided into three board categories: motion-based, appearance-based, and deep learning (DL)-based. Motion-based approaches use a sequence of images for the detection of objects. Besides, appearance-based methods use image processing techniques to identify objects directly from images or videos. However, these methods usually fail in detecting objects in complex scenarios. Deep learning (DL)-based methods use either motion features or appearance features or both for object detection in images or videos. DL-based approaches for object detection have gained much attention as compared to either appearance or motion-based approaches. However, object detection using DL-based methods is mainly of two types: two-stage detectors, and one-stage detectors. In the two-stage detection, objects are first localized followed by their classification. Over the past years, various two-stage detectors [10, 16–18, 21, 31, 32, 38, 41, 45, 53] have been reported in the literature. Among those, RCNN [18], Fast RCNN [17], Faster RCNN [45], Mask RCNN [21], RFCN [10], FPN [31], granulated CNN [38], and granulated RCNN [41] are the commonest. On the other hand, one-stage detectors predict bounding boxes over the images, thus increasing the object detection speed. Like two-stage detectors, various models of the one-stage detector, have also been developed in recent years. These models include the YOLO [42–44], SSD [14, 32], RefineDet [58], and DCN [11]. In addition to these methods, hybrid DL-based methods for object detection have been getting attention these days. Hybrid deep learning is the term used to describe a method that combines several DL models or DL models with traditional machine learning techniques to improve the performance of a specific task. Recently, various hybrid DL models [33, 37, 57] are developed for object detection in videos.

RCNN [18]is the first two-stage object detection model. Typically, it consists of four stages. The first stage involves the generation of region proposals in the video frame. Then, a fixed-length feature vector is extracted from each region proposal. The third stage is responsible for the object classification task. The final stage is a bounding-box regressor for accurately bounding box prediction. The RCNN shows significantly better results compared with traditional methods for object detection [18]. Fast R-CNN was suggested by Ross Girshick [17] a year after R-CNN was first introduced. Fast R-CNN [17] takes the entire image as input, extracts the features from it, and then passes the region of interest (RoI) pooling layer to obtain fixed dimension features for the subsequent classification and bounding box regression over the classified object. In comparison to RCNN, a large amount of calculation time is saved for Fast R-CNN. Because it considers the location of pooling features as the possible regions and is used for classification. Another distinction between RCNN and Fast RCNN is that the former employs a multi-stage end-to-end training procedure, whereas the latter does it in a single stage. Faster-RCNN [45] is another two-stage model for object detection which has been developed after three months of the development of Fast-RCNN [17]. It is an improved version of Fast-RCNN in terms of object detection accuracy and execution time. In Fast R-CNN, region proposals are generated using a selective search method that makes the system slow and takes the same amount of execution time as the detection network. Faster RCNN replaces this module with a Region Proposal Network (RPN), which is a fully convolutional network and effectively predicts regions proposals with a broad variety of sizes and aspect ratios. It requires less time to generate region proposals than Fast RCNN. This is because Faster RCNN simultaneously shares with the detection network both the full image convolutional features and a common set of convolutional layers. Anchors are set at each convolution feature point to create region proposals of varying sizes. Anchors are spatial windows of varying sizes and aspect ratios that are inserted at certain locations in the input feature map. Faster RCNN employs anchor boxes with three distinct scales and aspect ratios. Dai et al. [10] developed another two-stage object detector, named R-FCN, which is a modified version of Faster RCNN. In general, R-FCN has been developed to address certain issues in Faster RCNN. Feature pyramid network (FPN) [31] is another popular network for object detection. Many object detection algorithms have implemented feature pyramids, which are based on image pyramids, to increase scale invariance [13, 19]. However, this type of technique requires a lot of training time and memory. This problem was effectively resolved by FPN. In [21], He et al. proposed Mask R-CNN. It is an extension of Faster RCNN that focuses primarily on segmentation tasks. In Mask RCNN, a ResNet-FPN [31] (feature pyramid network) is combined with Faster-RCNN as a backbone to extract region-of-interest features from various layers of the feature pyramid by their scale, thereby achieving high detection accuracy and speed. Recently, granular computing-based CNNs [38, 41], like granulated CNN [38], and granulated RCNN [41], are developed for object detection.

Redmon et al. proposed a one-stage object detector, called You Only Look Once (YOLO) [42–44], after the development of Faster R-CNN [45]. The main contribution is real-time object detection in full images and webcam. Later, many expanded versions of the YOLO were developed. These models include YOLO-v2 [42], YOLO-v3 [43], YOLO-v4 [6], and YOLO-v5 [9]. Typically, these improved YOLO networks were developed to address the flaws in the earlier models. SSD (single-shot detector) [32] proposed by Liu et al. is another one-stage object detector that can detect objects of multiple classes. In [14], a modified version of SSD, called De-convolutional Single-shot Detector (DSSD), has been proposed. In this model, both de-convolution and prediction models are added to SSD, and Res-Net 101 [20] was as a backbone. RefineDet [58] is another kind of one-stage object detector that consists of two interconnected stages, such as refinement, and object detection. These two stages are interconnected through a transfer connection block. Deformable convolutional networks (DCNs) [11] were developed to overcome the issues of regular CNN for object detection. It has two varieties: DCNv1 and DCNv2.

Among the models mentioned above, Faster RCNN has received a tremendous amount of attention from researchers for object detection in images or videos. This may be due to its high accuracy and simplicity. However, in spite of these efforts, object detection in real-time videos remains a challenging problem that needs to be properly addressed for detecting objects efficiently.

2.2 Object tracking

Object tracking is a critical step in locating the moving objects in a video sequence. It is accomplished by locating the target objects in consecutive frames of a video. Over the past years, various methods have been developed in this regard [25, 39, 40]. In [7], a method for face detection and tracking has been proposed. This has been done by using the Viola-Jones face detector which extracts speed-up robust features (SURF) from detected objects. After that, an improved KLT has been used with the Gradient Weighted Optical Flow (GWOF) to track static or moving objects. In [28], a robust multicast multi-object tracking algorithm has been applied. Changing point detection algorithm is used to observe abnormal changes based on spatio-temporal analysis and also a KLT-based motion detector is employed to track the objects. Hamd et al. [4] propose a technique for adaptive block tracking using the Kalman Filter. This approach is more suited for single object tracking in multiple video frames. Thus, this couldn't be used for multiple object tracking in video frames as it is unable to perform the unique labelling achieved in our approach. Li et al. [29] proposes a simple yet effective approach that exploits rich feature information from reliable patches based on the weighted local sparse representation that takes into account the importance of each patch. To achieve this, a reconstruction error-based weight function is also designed to determine patch reliability. But this tracking approach leads to missing important patches which might contain valuable data with respect to crowd tracking scenarios. The proposed approach works with a holistic set of features, thus ensuring better information retrieval. In [54], the authors presented a novel TrackletNet Tracker (TNT) that combines temporal information and appearance information in the same framework. The tracker successfully identifies and deals with the problem relating to fast motion and occlusions. However, measuring separate tracklets for a single object makes the tracker unsuitable for performing real-time analysis in case of a significantly dense crowd. In [48], a detailed comparative study has been done for tracking using the KLT algorithm versus the Camshift algorithm. They have shown that KLT outperforms Camshift for object tracking in videos and especially in tasks involving crowd scenarios. The method in [15], deals with a tracking algorithm based on structure similarity. The performance of this approach is very high in complex scenarios. But it fails to provide adequate veracity in case of videos having missing frames. Our approach performs unique labelling of objects and tracks them efficiently even in the case of missing frames. Zhang et al. [59] use a one-shot approach to obtain very high accuracy in multi-object tracking. They show that a separate re-identification module is usually a heavy load for the system which in turn degrades the tracking accuracy. However, this method is real-time, fast and lightweight but fails to provide any further insight regarding the actions rendering it unsuitable to be used as a part of a surveillance system. The authors of [56] have suggested a definitive approach to better the performance of a deep learning-based multi-object tracker. The approach when integrated with a Deep Hungarian Net (DHN) gives a significant boost in the accuracy. However, these approaches are more aimed toward achieving high MOTA and MOTP scores. These tracking parameters are insufficient to determine the performance of a tracker when used specifically for surveillance purposes. Wang et al. [55] proposed a real-time approach for efficient tracking that can identify detections and corresponding embedding at one go. But this system particularly trades speed and accuracy with the details of a particular object that defies the main purpose of a system with security as its main aim. In [27], the authors proposed a novel tracking approach, in which the tracking has been done in 2 phases. The local periphery of the object is segregated using the optical flow. More precise localization is done with the multi-cue feature fusion around the centroid of the approximate outline obtained in the previous step. In [30], the tracking has been done using two different networks, the tracking and the inspecting networks. This algorithm shows high veracity for scarce distribution of objects but fails in dense crowd scenarios because of the unnecessary overlap between the networks of two or more objects. Recently, in [2], the authors have proposed a method o detect and track multiple objects and address some of the challenges that prevent good results and robust performance. Nuha et al. [1] have combined principal component analysis and a deep learning model for object detection and tracking in real-time scenarios. They have shown that their method outperforms other related methods. Despite these efforts, tracking multiple objects in a real-time video remains a difficult problem. Thus, researchers all over the world are working on developing some robust object trackers to track multiple objects in the videos of unconstrained scenarios.

3 Problem statement

Accurate object detection and tracking in a real-time video are observed to be a difficult and stimulating tasks. Feature extraction of individual entities for object detection plays a crucial role in the real-time scenario. It has to be more effective to determine the intention of the crowd. The identification along with the unique labelling of individuals can also be a challenge. This unique label for each individual can be persisted throughout the sequence of frames in the given situation which massively help the exact object tracking task. The correct unique labelling of the object can increase the potential of the tracker to track every object across the sequence of frames in the given situation. That leads to identifying the movement of the crowd, which in turn helps to detect the unusual or suspicious event if found. In this regard, the accurate and efficient tracking of objects can be produced based on the precise motion flow of the crowd. So, the displacement of the objects can be expressed as their motion feature for the upcoming frames without losing their respective identification marks. This can be expressed as their optical flow. As the video is a real-time scenario, it can contain some missing frames. So, the proposed tracker has to deal with it where the missing frames cannot affect the correctness of object tracking and the generation of crowd optical flow. Thus, based on this information content, the displacement of each object needs to be taken into account. This can be expressed mathematically in the case of the current frame with respect to our previous frame of reference.

Suppose, the input object for the first frame is $I(x, y)$ at time t_1. After the displacement of the object in the next frame at time t_2, the position has changed as $H(x_1, y_1)$ where

$$H(x_1, y_1) = I(x + u, y + v) \tag{1}$$

after $t = t_2 - t_1$

Now, to generate the motion flow of the object, the displacement of the object (u, v)needs to be calculated.

However, object detection and tracking are frequently identified as difficult tasks in various scenarios due to low visibility, poor image quality, or missing frames. Multiple streams of real-time scenarios with high frame rates can be captured by several CCTV sources in an environment. However, processing those frames with the equivalent low-speed data processing module is a difficult task. Because of resource constraints, the video stream analyser may compromise some frames many times in order to resolve the overflow problem. To match the incoming frame rate and data processing frame rate, some frames of the captured videos need to be dropped out. Our proposed model has addressed the aforementioned scenario. It outperforms all related methods considering the missing frames that occur sometimes during the tracing of objects. Thus, this work has aimed at establishing an efficient object detection model that can be able to effectively detect human faces with maximum accuracy along with unique labeling in heterogeneous real-time environments. The model can also be included with an effectual object motion tracker that can correctly track individual objects in the video supported by an optical flow motion generator. This generator leads to detect the doubtful movements of objects if needed.

4 Proposed system

In this section, we describe our proposed unifocal motion tracking surveillance system (UMTSS), which is based on the sequential application of three techniques: a method for multi-object detection, unifocal feature-based object tracking, and a method for keyframe extraction. The proposed surveillance system can monitor the crowd events and investigate their motion for further investigation. Figure 1 illustrates the functionality of the proposed surveillance system.

4.1 Object detection using Faster-RCNN Inception-v2 (FRI) model

In this section, we present our proposed method for multi-objects detection in real-time videos. As stated earlier, object detection in videos is one of the fundamental and challenging tasks of any intelligent surveillance system. In this regard, deep learning (DL)-based techniques have recently been widely adopted in the field of object detection [39]. This is because DL-based methods can perform significantly better compared to handcrafted feature-based methods [39]. Thus, in this work, we employed a Faster-RCNN with Inception-v2 model, called Faster-RCNN Inception-v2 (FRI), for detecting multiple objects in each frame of a video. In this method, the Inception-v2 [50] model is innovatively combined with Faster-RCNN [45] in order to achieve a high object detection rate. Figure 2 illustrates the layered architecture of FRI model.

Faster-RCNN [45] is an enhanced version of Fast-RCNN [17] in terms of object detection accuracy and execution time. In Fast R-CNN, region proposals are generated using a selective search method that makes the system slow and takes the same amount of execution time as the detection network. Faster RCNN, on the other hand, replaces this module with a Region Proposal Network (RPN), which is a fully convolutional network and effectively predicts region proposals with a broad variety of sizes and aspect ratios. In general, Faster-RCNN consists of mainly three components: Convolutional neural network (CNN), Region proposal network (RPN), and Classes and bounding box (BB) prediction. CNN is the backbone of the Faster-RCNN used to extract feature maps from the input. Besides, RPN is a small neural network sliding on the last feature map of the CNN for the generation of region proposals. Finally, there is another fully connected neural network that takes regions proposed by RBN as input and predicts object class and BBs. Typically, the quality of features determines the upper

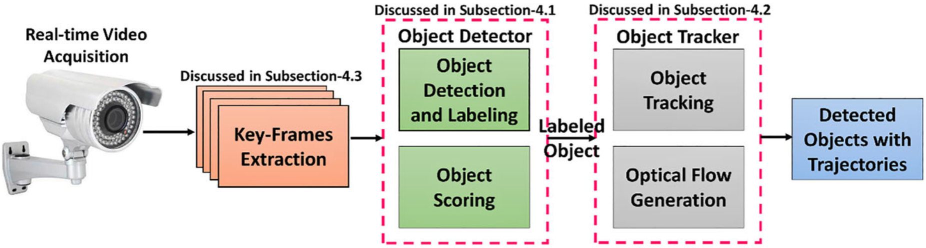

Real-time Video Acquisition
Discussed in Subsection-4.3
Key-Frames Extraction
Discussed in Subsection-4.1
Object Detector
Object Detection and Labeling
Object Scoring
Labeled Object
Discussed in Subsection-4.2
Object Tracker
Object Tracking
Optical Flow Generation
Detected Objects with Trajectories

Fig. 1 Block diagram of the proposed UMTSS

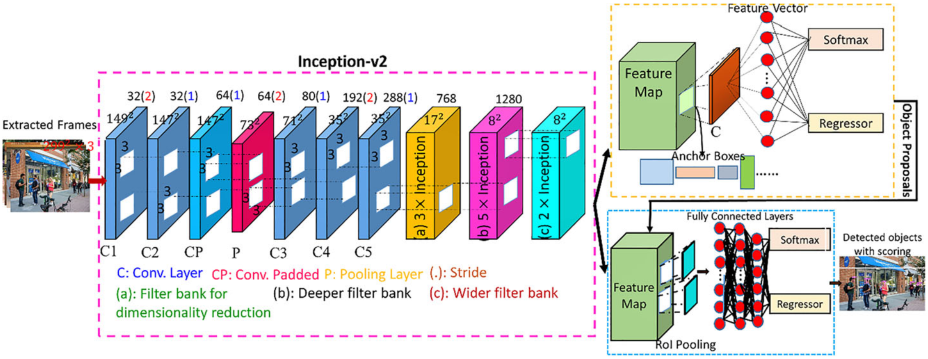


Fig. 2 Layered architecture of FRI model

bound of network performance. Thus, the choice of CNN plays a crucial role in the performance of Faster-RCNN. The common CNNs used in this model are VGG-16 [52] and Alexnet [18]. These CNNs, however, have a monolithic architecture. As a result, the computational cost of these CNNs is very high. Besides, the Inception-v2 model has a lower computational cost than VGG Net, Alexnet, and other top-performing successors. In an inception block, several convolutional layers have been working parallelly, which contains a sparse architecture that helps dimension reduction. This improves the complexity but makes better performance compared to the monolithic architecture. Also, it controls the overfitting problem. In Inception-v2 [50], an efficient approach has been followed to factorize layers with higher convolutions into ones with lower convolutions for more efficient computation. This module makes the convolution network going to be wider than deeper. It is a multi-layered model integrated with the features of auxiliary classification and label smoothing. Label smoothing is a mechanism to regularize the classifier by estimating the level of label dropout during training. Inception-v2 has three different types of filter banks, such as (a), (b), and (c) (see Fig. 2). The decomposition of these three filter banks are shown in Fig. 3. The first filter bank (a), shown in Fig. 3, replaced 5 × 5 convolutions to be 3 × 3 convolutions. This follows the principle said spatial aggregation can be done over lower dimensional embedding without much or any loss in representational power. By conducting the 3 × 3 convolutions, convolution performance has been boosted. Factorizing convolutional filter size $n \times n$ into1 $\times$ n and $n \times 1$ convolutions, we have found their method 33% cheaper than the single 3 × 3 convolutions. That has been

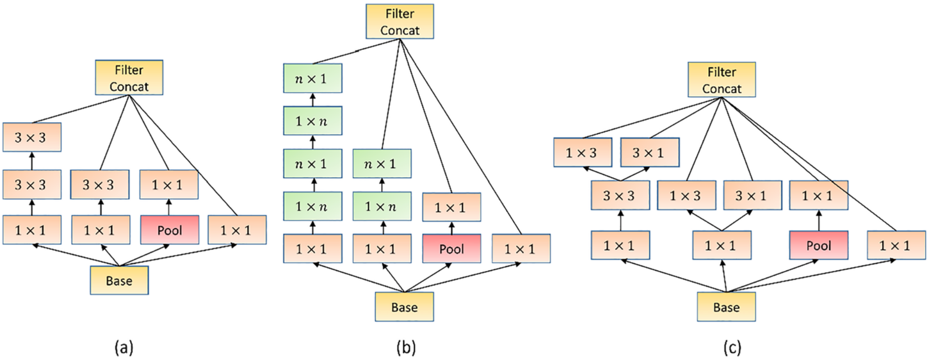


Fig. 3 Decomposition of filter banks (**a**), (**b**), and (**c**) of Inception-v2 model

shown in (b). Moreover, the filter has been expanded that follows the principle of higher dimensional representations can be easier to process locally within a network. The expanded module has been shown in (c).

It has been observed that input resolution can significantly impact detection accuracy. From our experiments, we observe that decreasing resolution by a factor of two in both dimensions consistently lowers accuracy (by 15: 88% on average) but also reduces inference time by a relative factor of 27: 4% on average [50]. We have also performed the score conversion function on the identified objects which represents the probability of detecting objects of being human.

After identifying the objects, the task of object labelling and tracking (see Subsection 4.2) of the identified objects has been performed. Each of the identified objects can be labelled by assigning a unique id, which remains specific to a particular object as long as it remains in the frame of reference. In order to track the objects correctly across the frames, unique object labelling plays a crucial role to keep the spatio-temporal data of individual moving objects across the frames. We have considered the situation where some of the identified objects may not be detected in a few consequent key-frames due to different postures or deteriorated image quality. But as all of them have been already labelled by unique ids, those objects will again correctly be tracked as soon as they appear in the next reference frames.

In this work, the pre-trained convolutional blocks and the weights of Inception-v2 model are utilized in Faster-RCNN, which is further trained end-to-end using our experimental datasets. Inception-v2 was originally trained on ImageNet dataset [46], which is a large dataset consisting of near about 200 object classes. Here, we used the transfer learning strategy to fine-tune our FRI model to detect multi-objects in videos. In the transfer learning stage, FRI model is trained for 100 epochs for each training set at a learning rate of 10^{-3}. The SGD optimization function is used for backpropagation to minimize cross-entropy loss. The value of 0.99 is used for the momentum decay rate. A 3.5 GHz AMD Ryzen 3 1300 quad-core processor with 128 GB memory and a Nvidia GeForce GTX 1060 6GB GPU were used to train the network. Table 1 illustrates the hyper-parameters used to train our FRI network. The network is implemented in Python using the tensorflow framework.

4.2 Object tracking

This section presents our proposed object tracking method. After the detection of objects using the method stated above (See Subsection 4.1), they are need to track across the frames. In this regard, we have proposed here a unifocal feature-based KLT method, called unifocal feature-based object tracking using KLT (UFOT-KLT), for robust object tracking in real-time videos. KLT [51] is a tracking algorithm which tries to find the shift of point of interest, that might have taken. The framework is based on local optimization. When the movement is minor, it is

Table 1 Hyper-parameters and their values used to train the FRI model

Hyper-parameters	Values
Loss Function	Cross-entropy
Optimizer	SGD
Initial Learning Rate	0.01
No. of Epochs	100
Batch Size	4
Momentum	0.99

easy to follow the movement. But for real-time motion, it is difficult to figure out the features that can be tracked. There are several challenges present while tracking the object in motion. Some points may change their appearance over time due to rotation or moving into shadows. Also, some points may appear or disappear at some point of time. Considering all the aforementioned challenges, the KLT tracker tracks the feature points with some assumptions:

- Brightness Constancy- Projection of the same points looks like same in every possible frame.
- Small Motion- Feature points do not move very far.
- Spatial Coherence- Feature points move like they are neighbours.

However, feature extraction plays an important role for a KLT-based object tracking. In this regard, Shi-Tomasi [51] introduced some 'good' features points to track object motion using Lucas-Kanade algorithm. Those feature points are extracted using eigen values of the second-moment matrix (e.g. Haris Corner Detection). Those good feature points to track are the ones whose motion can be estimated reliably. The KLT model tracks those features from frame to frame with Lucas-Kanade algorithm [51]. The consistency of tracking has been checked by affine registration to the first observed instance of the feature. For the larger displacements, the affine model is more accurate. Compared to the first frame, it helps to minimize drift. It uses intensity second-moment matrix and differences across frames to find displacement.

It is observed that, using KLT, the feature extraction of the individual object (human) is a difficult task for a real-time crowded scenario. The extracted multi-feature points do not validate the unique identity of individual objects. It leads to being a challenge for the tracker to get the precise motion flow of every detected object uniquely across the frames without losing their respective identification marks. In this regard, a distinctive monopoint feature identification technique is introduced in the proposed tracking model. Each identified human face has been identified by bounding boxes. The centre point of each face has been calculated and labelled by unique ids which can be persisted throughout the sequence of frames. Based on this feature, we have proposed UFOT-KLT. That extracted monopoint feature massively helps the unique object tracking task. The multipoint feature extraction dependency moving down to unipoint distinctive features that extensively decrease the overheads of multipoint features tracking techniques and also maintains the precision of unique labelling of objects for the sequence of frames. Therefore, instead of multi-tracking feature points, the tracker can trace the unique object trajectories depending on a single unique point for each. So, this method also minimizes the execution cost and time of the tracking system and also maintains the correctness of the required objectives. Another advantage of our technique is that, since we are labelling the objects after the identification, we can successfully track them even if their position changes over time. Even if the object has changed its orientation to some other angle for which it cannot be detected, still the UFOT-KLT will continue to track it throughout the consecutive frames. It provides the discrete missing points of the movement of the pedestrian that can help to produce the correct optical flow of the identified objects across the video. A moving object in a single frame at position p is represented as:

$$\sum_{x}\left([I(w(x:p))-T(x)]^2\right) \tag{2}$$

$$\sum_{x}\left([I(w(x:p))-T(x)]^2\right) \tag{3}$$

Considering the object moves by p in the consequent frame, it can give by

$$\sum_{x}\left([I(w(x:p+\Delta p))-T(x)]^{2}\right) \tag{4}$$

Thus, the change in position or the update in the frame with respect to its predecessor is given by:

$$\alpha=\sum_{x}\left([I(w(x:p+\Delta p))-T(x)]^{2}-[I(w(x:p))-T(x)]^{2}\right) \tag{5}$$

where, $w(.)$ represents the position of objects x in the given frame, $I(.)$ denotes the image where the given frame belongs to, and $T(x)$ represents the dynamic of the reference frame for object x. Extrapolating it for k number of objects, the mean distance between the frames is represented by τ, where,

$$\tau=\frac{\sqrt{\sum_{i=1}^{k}(\alpha_i)^2}}{k} \tag{6}$$

This added feature can be extensively helpful to evaluate the motion flow of the objects to investigate the intention of the crowd. After tracking the unique identified objects, their respective displacements can be expressed in terms of optical flow using the Lucas-Kanade algorithm. Lucas-Kanade optical flow generation technique has some limitations. If the motion is large or some intermediate frames have been lost due to noise, the movement of the distinct object is estimated at an enormous precision risk. Thus, to fix this challenge, the mean shift algorithm [32] is applied to the tracking model which can efficiently predict all the missing points using the maxima of a density function. This mode–seeking algorithm can estimate each lost point by applying the density function to the identified movements. The predictive key points of the object can be obtained by taking into account both the previous as well as future frame of reference of the distinct objects. So, the optical flow leads to generating self-reliant movements of the objects without compromising their accuracy.

4.3 Keyframe extraction

The continuous unique object labelling and tracking across the frames is a vital task. Otherwise, the object information can be missed and generate incorrect motion details. As we are working with real-time videos, the speed of coming videos must match the frames per second. Otherwise, the overflow of video frames can happen. To address this issue, a keyframe extraction technique [36] is employed in this work. In [36], the intensity level entropy difference between consecutive frames was used to extract the keyframe. The frames with a difference greater than a certain threshold value are considered keyframes for the given scenario. Using this technique keyframes are effectively selected from all the incoming videos and used for further analysis.

In the present work, an efficient producer-consumer technique using Kafka has been implemented for the method [36] to maintain the balance between the speed of the incoming real-time video and the process of frames per second. In the Kafka implementation, the selected keyframes are converted into JSON for light-weighted data transmission. Data has been transmitted efficiently using the producer-consumer method. On the producer end, the video stream has been continuously generated through different sources like CCTV and on the consumer end, the data will consume accordingly and converted back to the image. In between these functions, the broker plays the role of a mediator which will protect the system from any data overflow situations. The broker keeps the JSON data until the consumers are up to consumption. The consumer has been trained with object detection and a unique labelling algorithm. If any object is found, then it will store the relevant information of the identified object. In Fig. 4, the proposed key frame extraction technique using Kafka has been depicted. For this implementation, we have used the OpenCV platform in python programming.

5 Experimental result

In this section, the details of the experimental results of our proposed system have been presented. The targeted situation relevant to the work includes tracking multiple objects/people in real-time videos. Thus, we have considered different crowded scenarios taken from different publicly available standard datasets. The videos contain various acts like war scenes, the state of busy road crossing, disturbed crowds at crossroads, and the running of the bulls scenario. In Subsection 5.1 of this section, the database used for experimentation is described. The quantitative evaluation of object detection and tracking modules of UMTSS is demonstrated in Subsections 5.2 and 5.3. In Subsection 5.4, the performance evaluation on keyframe extraction-based object tracking is presented.

5.1 Databases for surveillance

The proposed UMTSS is evaluated on challenging video sequences taken from the CrowdHuman [47] dataset, Performance Evaluation of Tracking and Surveillance (PETS) dataset [12], University of California San Diego (UCSD) Pedestrian Dataset [35], AGORASET: A dataset for crowd video analysis [3], and CRCV: Centre for Research in Computer Vision- Tracking in High-Density crowd dataset [22]. CrowdHuman is a large human object detection dataset that contains over 25,000 images. It has been expertly annotated and covers a wide range of scenes. It has more complex and crowded scenes, making conventional duplicate removal difficult. The average number of people in an image is

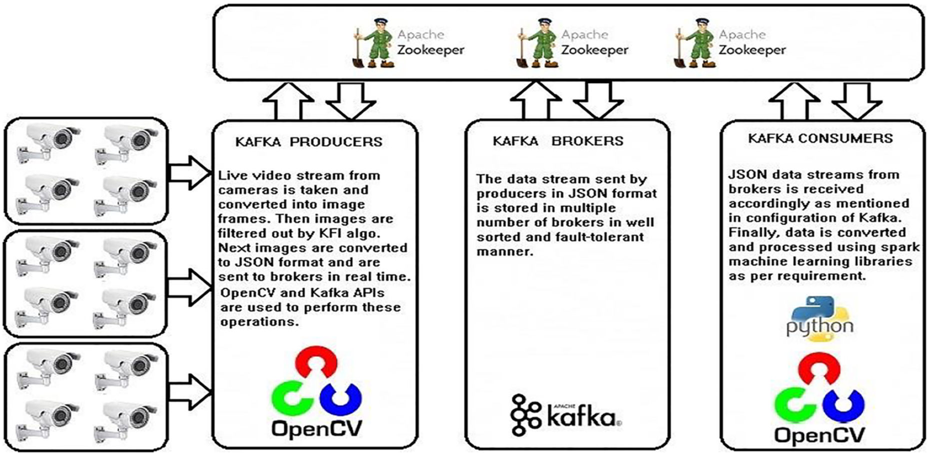

Apache Zookeeper
Apache Zookeeper
Apache Zookeeper
KAFKA PRODUCERS
Live video stream from cameras is taken and converted into image frames. Then images are filtered out by KFI algo. Next images are converted to JSON format and are sent to brokers in real time. OpenCV and Kafka APIs are used to perform these operations.
OpenCV
KAFKA BROKERS
The data stream sent by producers in JSON format is stored in multiple number of brokers in well sorted and fault-tolerant manner.
kafka
KAFKA CONSUMERS
JSON data streams from brokers is received accordingly as mentioned in configuration of Kafka. Finally, data is converted and processed using spark machine learning libraries as per requirement.
python
OpenCV

Fig. 4 Keyframe extraction of real-time video using Kafka implementation technique

23. The DV cameras wear used to film all datasets (except CrowdHuman) with frame rate ~ 7. Camera models used are Axis 223 M, and PTZ Axis 233D, with the resolution used, and Sony DCRPC1000E 3xCMOS, and Canon MV-1 1xCCD with the resolution used. PETS dataset consists of video sequences of the medium-density crowd, high-density crowd and sparse crowd. It has 5 subsets. Each subset contains several sequences and each sequence contains different views. UCSD dataset contains 98 sparse to very high crowd videos with five well-defined abnormal categories. Consisting videos have been categorized into two subsets where each scene video footage is sliced into clips of 120–200 frames. The videos have been captured by stationary cameras with a resolution of 740 × 280 at 30 fps. AGORASET dataset consists of simulation-based crowd videos composed of 8 scenes. All the videos correspond to various situations like the stress of the crowd, viewing angles etc. The disposal of the same scene with and without genuine visual effects allows the user to detect disruption to the analytical procedure by light conditions. The images were captured with the parameters of $sc ale = 4$ and pixel $format = 422p$. CRCV dataset contains three categories of subsets. These are: the crowd counting dataset, crowd segmentation data set, and tracking in high-density crowd dataset. All those videos have been taken from FLICKR, the Getty Images website and the BBC motion library. The dataset consists of extremely high-density crowd videos. The sequence length of the frames varies from 120 to 492 frames on an average.

5.2 Object detection performance evaluation

The efficacy of the FRI model has been evaluated on the CrowdHuman dataset [47] concerning various performance measures. The CrowdHuman is a commonly used dataset for object detection performance evaluation. It consists of the largest number of persons per image and the largest number of pairs of intersecting bounding boxes among all datasets for human detection [47]. Thus, we consider this dataset to evaluate the efficacy of our FRI model. The performance of FRI model is compared against Fast-RCNN [17], and different backbone CNNs with Faster-RCNN concerning three standard performance metrics. These metrics include precision [49], recall, and F1-score [49] with criteria Intersection-over-Union (IoU) [49] threshold of 0.5. The idea of IoU is used to assess localization correctness. Typically, it computes the ratio of an overlapping region between model predicted and ground truth boundaries to the total or region of union between the two boundaries. Figure 5 demonstrates some examples of objects detected in images of the CrowdHuman test dataset. It can be observed that FRI model detected almost all objects in images.

Table 2 shows precision, recall, and, F1-scores for FRI, and three other models mentioned earlier at an IoU of 0.5. It can be seen that the FRI model consistently retained significant improvement over the other models. FRI model improves precision by more than 1.2%, recall by 2.56%, and F1-score by 1.84% with respect to the other models listed in Table 2. Also, a higher IoU threshold, i.e., 0.7, is used to examine how the model adapts when more localization is needed. Table 3 shows precision, recall, and F1-scores at an IoU threshold of 0.7 for FRI, and three other models. As can be observed, FRI still outperformed other models.

5.3 Tracking performance evaluation

In this section, the tracking performance of the proposed UMTSS has been evaluated. The tracking performance of UMTSS is evaluated on video sequences of four standard publicly available datasets: PETS, UCSD, AGORASET, and CRCV, discussed in Section 5.1. Figure 6

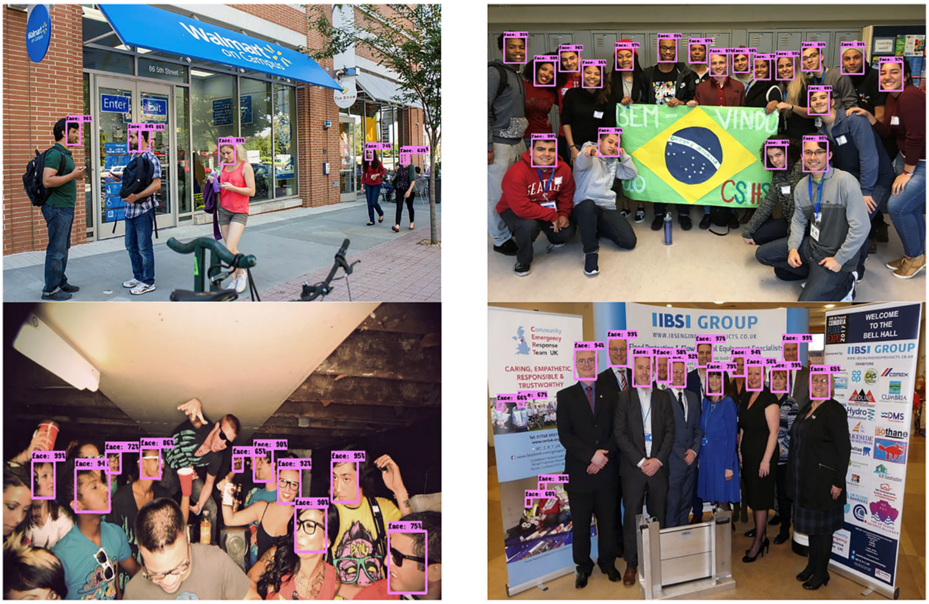


Fig. 5 Examples of objects detected in images of CrowdHuman test dataset

shows the optical flow of objects in video frames from the PETS, UCSD, AGORASET, and CRCV datasets using our proposed method. Also, we compared our method with six state-of-the-art related methods. The state-of-the-art methods considered for comparison are Kalman Tracker [4], conventional KLT [51] Tracker, FAIR-MOT [59], JDE [55], DEEP-MOT [56], TNT [54], AVFOT [2], and PCA-DLM [1]. For the sake of fair comparison, all these methods are implemented according to the description specified in the respective original papers. The state-of-the-art tracking algorithms used in crowd detection are Kalman Tracker [4] and the conventional KLT [51] Tracker. These algorithms are well known for mass object tracking. But, the results are unsatisfactory in the case of random motion of the dense crowd and do not yield accurate results when the video frames passed are non-consecutive. The unique identification of the objects is also not possible in the Kalman tracker. However, the proposed UMTSS can overcome all those limitations. This is a major advantage of performing tracking after object detection, which we have obtained in our work. The corresponding coordinates of the detected objects can be derived from each frame which can provide a more meaningful insight regarding the displacement of the detected objects and also aiding us in finding the trajectories.

As UMTSS deals with real-time crowd scenarios, thus in this experiment, the system performance has been measured on two parameters- Track length of the pedestrian [56], and dynamic tracking Accuracy.

Table 2 Performance comparison of different models at an IoU threshold = 0.5

Methods	Precision	Recall	F1-score
Fast RCNN	81.79	86.91	84.27
Faster-RCNN VGG-16	83.47	89.38	86.32
Faster RCNN Inception-v1	85.28	89.59	87.38
FRI	***86.48***	***92.15***	***89.22***

Table 3 Performance comparison of different models at an IoU threshold = 0.7

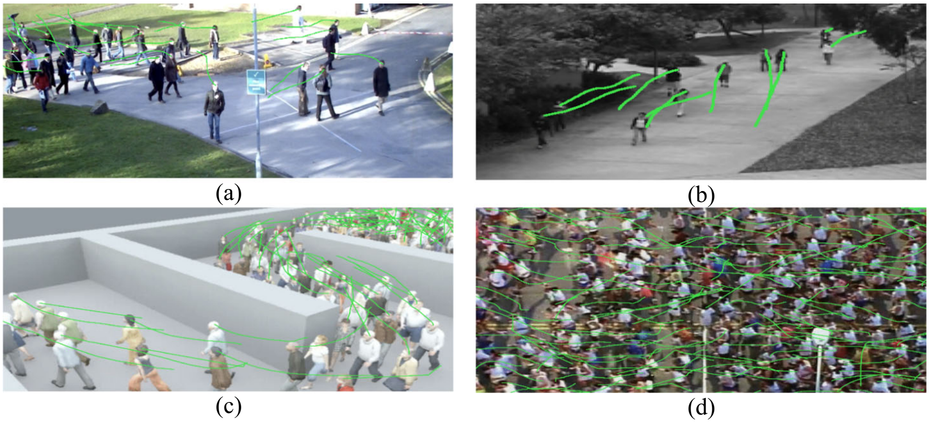
Methods	Precision	Recall	F1-score
Fast RCNN	55.19	60.31	57.63
Faster-RCNN VGG-16	57.77	63.68	60.58
Faster RCNN Inception-v1	61.88	66.29	64
FRI	***62.98***	***68.55***	***65.64***

- **Track Length (TL)**: Tracking length [8] is a measure parameter that is used to compare trackers in tracking length. The role of this measure is to report the number of successfully tracked frames from the tracker's initialization to its (first) failure. The importance of tracking length is significantly high, especially in a surveillance system. However, there are various methods for automating the failure criterion, which affects the comparison results. Usually, this is accomplished by applying a threshold to the centre error or the overlap region. However, when the problem is the simultaneous detection of multiple objects across real-time video frames, it is inefficient [8]. Hence, in this work, the threshold for each frame has been determined using the following metric $\frac{No.ofcorrectlydetectedobjects}{Totalno.ofobjectsintheframe}$.

  Tracking length serves as a crucial aspect to bolster the claim for the performance of our tracker. The virtue of being able to track a labelled object for a higher number of frames gives an opportunity for better feature extraction and behaviour analysis.
- **Dynamic Tracking Accuracy (DTA)**: The most commonly used parameter to observe the tracking accuracy of any state-of-the-art tracker is MOTA (multi-object tracking accuracy) [5]. This quantification is usually achieved on large annotated video datasets. However, this parameter alone is not enough to form a conclusion about tracking systems specifically for surveillance purposes. The objective of a surveillance system is not merely to observe the trajectory of each identified entity but also to monitor them. Hence, in this work, we proposed a new parameter, called dynamic tracking accuracy (DTA), that measures the tracking accuracy in a dynamic manner. It is dependent on the factors stated below:

– The ratio of total objects detected in a frame to the actual number of objects present.

– Number of common objects in one particular frame and its following frame.



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

Fig. 6 Optical flow of objects using the proposed UMTSS. Frames taken from **a** PETS dataset, **b** UCSD dataset, **c** AGORASET dataset, and **d** CRCV dataset

Table 4 Performance evaluation of UMTSS against KLT and Kalman tracker

Index	Dataset	Parameter	KLT tracker	Kalman tracker	UMTSS
1	PETS	TL	0.382	0.411	**0.639**
2	PETS	DTA	0.563	0.521	**0.844**
3	UCSD	TL	0.322	0.213	**0.672**
4	UCSD	DTA	0.568	0.379	**0.871**
5	AGORASET	TL	0.289	0.315	**0.663**
6	AGORASET	DTA	0.474	0.452	**0.855**
7	CRCV	TL	0.261	0239	**0.550**
8	CRCV	DTA	0.313	0.262	**0.608**

- Count of new objects entering the frame of reference and being detected.
- Count of old objects leaving the frame of reference.

  The proposed DTA is defined as follows. First, we have calculated DTA for a single frame transition (i.e., DTA $(i+1, i)$), which is defined as:

$$DTA(i+1,i) = \frac{no.\ of\ new\ trackids\ generated}{2 \times no.\ of\ new\ object\ in\ the\ frame} + \frac{no.\ of\ trackids\ lost}{2 \times no.\ of\ object\ left\ in\ the\ frame} \quad (7)$$

Table 5 Performance evaluation of UMTSS with other related methods

Index	Dataset	Tracking Algorithm	TL	DTA
1	PETS	FAIR-MOT	0.489	**0.884**
		JDE	0.623	0.831
		DEEP-MOT	0.596	0.832
		TNT	0.572	0.764
		AVFOT	0.611	0.803
		PCA-DLM	0.625	0.831
		UMTSS	***0.639***	0.844
2	UCSD	FAIR-MOT	0.537	**0.921**
		JDE	0.674	0.842
		DEEP-MOT	0.649	0.825
		TNT	0.651	0.810
		AVFOT	0.612	0.860
		PCA-DLM	0.662	0.863
		UMTSS	***0.672***	0.871
3	AGORASET	FAIR-MOT	0.618	**0.872**
		JDE	0.594	0.856
		DEEP-MOT	0.627	0.852
		TNT	0.601	0.839
		AVFOT	0.605	0.833
		PCA-DLM	0.633	0.859
		UMTSS	***0.663***	0.855
4	CRCV	FAIR-MOT	0.438	**0.631**
		JDE	0.522	0.534
		DEEP-MOT	0.532	0.611
		TNT	0.511	0.572
		AVFOT	0.512	0.588
		PCA-DLM	0.550	0.594
		UMTSS	***0.558***	0.608

Now, the final DTA for a tracking video is defined as:

$$DTA = \frac{\sum_{i=1}^{N-1} DTA(i+1, i)}{N-1} \tag{8}$$

where, N= total number of frames.

Table 4 shows the performance of the proposed UMTSS, Kalman tracker and KLT tracker concerning Track Length and DTA. It can be observed that our UMTSS outperforms the conventional tracking methods listed in Table 4 with a significant margin.

Table 5 shows a comprehensive comparison of the proposed method and six other related methods like FAIR-MOT, JDE, DEEP-MOT, TNT, AVFOT, and PCA-DLM concerning TL and DTA measures, where the best result is shown in bold font. It can be seen that our proposed UMTSS outperforms all the methods in terms of TL measure for all datasets. Besides, FAIR-MOT achieved the highest score in terms of DTA for all datasets, whereas our method achieved the second highest score for all datasets. However, it can also be observed the DTA score of our method is very close to FAIR-MOT. These results ensure that the performance of our method is consistent and credible compared to the other related methods listed in Table 5.

Table 6 Performance evaluation of UMTSS with other related methods on missing frames

Index	Dataset	Tracking Algorithm	TL	DTA
1	PETS	FAIR-MOT	0.185	0.389
		JDE	0.215	0.328
		DEEP-MOT	0.241	0.319
		TNT	0.194	0.264
		AVFOT	0.310	0.344
		PCA-DLM	0.324	0.372
		UMTSS	***0.338***	***0.385***
2	UCSD	FAIR-MOT	0.246	0.429
		JDE	0.317	0.421
		DEEP-MOT	0.299	0.458
		TNT	0.291	0.341
		AVFOT	0.316	0.469
		PCA-DLM	0.363	0.474
		UMTSS	***0.374***	***0.482***
3	AGORASET	FAIR-MOT	0.293	0.471
		JDE	0.217	0.391
		DEEP-MOT	0.252	0.452
		TNT	0.317	0.420
		AVFOT	0.285	0.443
		PCA-DLM	0.317	0.470
		UMTSS	***0.327***	***0.476***
4	CRCV	FAIR-MOT	0.125	0.329
		JDE	0.163	0.282
		DEEP-MOT	0.172	0.341
		TNT	0.155	0.283
		AVFOT	0.124	0.315
		PCA-DLM	0.162	0.332
		UMTSS	***0.211***	***0.356***

5.4 Performance evaluation on key-frame extraction-based object tracking

In this section, the performance of the proposed system is evaluated in the missing frames scenario. In this experiment, we performed a keyframe extraction technique, as described in Section 4.3, on given datasets and merged the video from the extracted key-frames. After that, the experiment has been performed using UMTSS and other six related methods mentioned earlier (see Section 5.3). Table 6 shows the performance of the proposed method and other related state-of-the-art methods in missing frames scenarios concerning TL and DTA. It can be observed that the proposed UMTSS outperforms other related methods concerning TL measure by a significant margin for all datasets. Also, it can be seen that our method performs notably better compared to other methods in terms of DTA measure for UCSD, AGORASET, and CRCV datasets. However, for PETS dataset, our method achieved the second highest DTA score, which is very close to the highest DTA score obtained by the FAIR-MOT method. From these results it can be conclude that our method can also performs significantly better on missing frames.

6 Conclusion and future work

In real-time surveillance video analysis and computer vision, multiple object tracking is crucially significant in resolving a number of elementary issues. The approach of tracking an object in a video, however, entails identifying the object in the first frame and determining its state in each subsequent frame. In this work, we have proposed a novel surveillance framework coined as a unifocal motion tracking surveillance system (UMTSS), for multi-object tracking in real-time videos. The proposed UMTSS combines two significant steps. First, a deep learning-based model, called FRI, is employed to detect multiple objects efficiently in each video frame. Then, a proposed unifocal feature-based KLT method is applied for tracking objects in each frame of the video based on region proposals generated by the object detector in the previous phase. Also, a new tracking parameter, called dynamic tracking accuracy (DTA), is proposed to quantify the performance of the tracking algorithms. The performance of our UMTSS has been evaluated on various standard crowd video databases, and compared with state-of-the-art related methods concerning different qualitative and quantitative measures. It has been observed that our UMTSS outperforms the state-of-the-art methods.

Although our approach obtains concrete results in the detection, labelling and tracking domains, there are a few areas that can be improved further. As the initial tracking points are identified by a trained neural network model, the accuracy diminishes greatly when handling objects from different orientations in the frame of reference. However, object detection and tracking are necessary in the presence of multiple CCTV cameras at times. The image resolutions or image quality of each of those cameras may differ. Thus, object detection and tracking across multiple CCTV cameras of varying quality is a significant future improvement opportunity for us. Further, in the crowd scenario, the tracking fails in case of high occlusion. Thus the accuracy of labelling of each object has been obtained at the cost of a minute error in the trajectory of the same object. Another major future improvement of our process lies in the high interdependency of the detecting, labelling and tracking modules, i.e. failure in one module will reflect an adverse effect on the others. So, the future direction of the work is to overcome all the mentioned limitations that will significantly increase the performance and precision of UMTSS.

Acknowledgements This project work is supported by DRDO under the title 'Object Identification through Syntactic as well as Semantic Interpretation from given Spatio-Temporal Scenarios'. The project was reviewed and sanctioned by the revıew committee under the project grant number ERIP/ER/1404742/M/01/1661. We would like to express our sincere gratitude to DRDO members for this opportunity.

Data availability Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Competing Interests The authors state that they have no conflicting financial interests or personal connections that may have influenced the work reported in this paper.

References

1. Abdulghafoor NH, Abdullah HN (2022) A novel real-time multiple objects detection and tracking framework for different challenges. Alexandria Eng J 61(12):9637–9647
2. Abdulghafoor NH, Abdullah HN (2022) Enhancement performance of multiple objects detection and tracking for real-time and online applications. Int J Intell Eng Syst 13:533–545
3. Allain P, Courty N, Corpetti T (2012) AGORASET: a dataset for crowd video analysis. In: 1st ICPR international workshop on pattern recognition and crowd analysis, pp 1–6
4. Ait Abdelali H, Essannouni F, Essannouni L, Aboutajdine D (2016) An adaptive object tracking using Kalman filter and probability product kernel. Model Simul Eng 2016
5. Bernardin K, Stiefelhagen R (2008) Evaluating multiple object tracking performance: the clear mot metrics. EURASIP J Image Video Process 2008:1–10
6. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934
7. Buddubariki V, Tulluri SG, Mukherjee S (2015) Multiple object tracking by improved KLT tracker over SURF features. In: 2015 fifth national conference on computer vision, pattern recognition, image processing and graphics (ncvpripg). IEEE, pp 1–4
8. Čehovin L, Leonardis A, Kristan M (2016) Visual object tracking performance measures revisited. IEEE Trans Image Process 25(3):1261–1274
9. Couturier R, Noura HN, Salman O, Sider A (2021) A deep learning object detection method for an efficient clusters initialization. arXiv preprint arXiv:2104.13634
10. Dai J, Li Y, He K, Sun J (2016) R-fcn: object detection via region-based fully convolutional networks. Adv Neural Inf Process Syst 29:379–387
11. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y (2017) Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 764–773
12. Ellis A, Ferryman J (2010) PETS2010 and PETS2009 evaluation of results using individual ground truthed single views. In: 2010 7th IEEE international conference on advanced video and signal based surveillance. IEEE, pp 135–142
13. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645
14. Fu CY, Liu W, Ranga A, Tyagi A, Berg AC (2017) Dssd: deconvolutional single shot detector. arXiv preprint arXiv:1701.06659
15. Fu C, Duan R, Kayacan E (2019) Visual tracking with online structural similarity-based weighted multiple instance learning. Inf Sci 481:292–310
16. Gani MO, Kuiry S, Das A, Nasipuri M, Das N (2021), January Multispectral object detection with deep learning. In: International conference on computational intelligence in communications and business analytics. Springer, Cham, pp 105–117
17. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
18. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
19. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916

20. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
21. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
22. Idrees H, Saleemi I, Seibert C, Shah M (2013) Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2547–2554
23. Jha S, Seo C, Yang E, Joshi GP (2021) Real-time object detection and trackingsystem for video surveillance system. Multimedia Tools Appl 80(3):3981–3996
24. Jiao L, Zhang F, Liu F, Yang S, Li L, Feng Z, Qu R (2019) A survey of deep learning-based object detection. IEEE Access 7:128837–128868
25. Jiménez-Bravo DM, Murciego ÁL, Mendes AS, Blás S, Bajo J (2022) Multi-object tracking in traffic environments: a systematic literature review. Neurocomputing
26. Khan MA, Mittal M, Goyal LM, Roy S (2021) A deep survey on supervised learning based human detection and activity classification methods. Multimedia Tools and Applications 80(18):27867–27923
27. Kumar A, Walia GS, Sharma K (2020) A novel approach for multi-cue feature fusion for robust object tracking. Appl Intell 50(10):3201–3218
28. Lee B, Erdenee E, Jin S, Nam MY, Jung YG, Rhee PK (2016) Multi-class multi-object tracking using changing point detection. In: European conference on computer vision. Springer, Cham, pp 68–83
29. Li Z, Zhang J, Zhang K, Li Z (2018) Visual tracking with weighted adaptive local sparse appearance model via spatio-temporal context learning. IEEE Trans Image Process 27(9):4478–4489
30. Li T, Wu P, Ding F, Yang W (2020) Parallel dual networks for visual object tracking. Appl Intell 50(12): 4631–4646
31. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
32. Liu J, Zhang S, Wang S, Metaxas DN (2016) Multispectral deep neural networks for pedestrian detection. arXiv preprint arXiv:1611.02644
33. Lu Y, Chen Y, Zhao D, Li H (2018) Hybrid deep learning based moving object detection via motion prediction. 2018 Chinese Automation Congress (CAC). IEEE, pp 1442–1447
34. Luna E, San Miguel JC, Ortego D, Martínez JM (2018) Abandoned object detection in video-surveillance: survey and comparison. Sensors 18(12):4290
35. Mahadevan V, Li W, Bhalodia V, Vasconcelos N (2010) Anomaly detection in crowded scenes. In: 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 1975–1981
36. Mentzelopoulos M, Psarrou A (2004) Key-frame extraction algorithm using entropy difference. In: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval, pp 39–45
37. Mukilan P, Semunigus W (2022) Human and object detection using hybrid deep convolutional neural network. Signal Image Video Process 1–11
38. Pal SK, Bhoumik D, Bhunia Chakraborty D (2020) Granulated deep learning and Z-numbers in motion detection and object recognition. Neural Comput Appl 32(21):16533–16548
39. Pal SK, Pramanik A, Maiti J, Mitra P (2021) Deep learning in multi-object detection and tracking: state of the art. Appl Intell 51(9):6400–6429
40. Park Y, Dang LM, Lee S, Han D, Moon H (2021) Multiple object tracking in deep learning approaches: a survey. Electronics 10(19):2406
41. Pramanik A, Pal SK, Maiti J, Mitra P (2021) Granulated RCNN and multi-class deep sort for multi-object detection and tracking. IEEE Trans Emerg Top Comput Intell 6(1):171–181
42. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263–7271
43. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767
44. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
45. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 28
46. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, … Fei-Fei L (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
47. Shao S, Zhao Z, Li B, Xiao T, Yu G, Zhang X, Sun J (2018) Crowdhuman: a benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123
48. Sharma P, Kokare PM, Kolekar MH (2019) Performance comparison of KLT and CAMSHIFT algorithms for video object tracking. Recent trends in communication, computing, and electronics. Springer, Singapore, pp 323–331

49. Sharma V, Mir RN (2020) A comprehensive and systematic look up into deep learning based object detection techniques: a review. Comput Sci Rev 38:100301
50. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
51. Shi J (1994) Good features to track. In: 1994 Proceedings of IEEE conference on computer vision and pattern recognition. IEEE, pp 593–600
52. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
53. Takumi K, Watanabe K, Ha Q, Tejero-De-Pablos A, Ushiku Y, Harada T (2017) Multispectral object detection for autonomous vehicles. In: Proceedings of the on thematic workshops of ACM multimedia 2017, pp 35–43
54. Wang G, Wang Y, Zhang H, Gu R, Hwang JN (2019) Exploit the connectivity: multi-object tracking with trackletnet. In: Proceedings of the 27th ACM International Conference on Multimedia, pp 482–490
55. Wang Z, Zheng L, Liu Y, Li Y, Wang S (2020) Towards real-time multi-object tracking. In: European conference on computer vision. Springer, Cham, pp 107–122
56. Xu Y, Osep A, Ban Y, Horaud R, Leal-Taixé L, Alameda-Pineda X (2020) How to train your deep multi-object tracker. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6787–6796
57. Xu Y, Li Z, Wang S, Li W, Sarkodie-Gyan T, Feng S (2021) A hybrid deep-learning model for fault diagnosis of rolling bearings. Measurement 169:108502
58. Zhang S, Wen L, Bian X, Lei Z, Li SZ (2018) Single-shot refinement neural network for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4203–4212
59. Zhang Y, Wang C, Wang X, Zeng W, Liu W (2021) Fairmot: on the fairness of detection and re-identification in multiple object tracking. Int J Comput Vision 129(11):3069–3087

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.