Chapter

Multi-Person Tracking Based on Faster R-CNN and Deep Appearance Features

Gulraiz Khan, Zeeshan Tariq
and Muhammad Usman Ghani Khan

Abstract

Mostly computer vision problems related to crowd analytics are highly dependent upon multi-object tracking (MOT) systems. There are two major steps involved in the design of MOT system: object detection and association. In the first step, desired objects are detected in every frame of video stream. Detection quality directly influences the performance of tracking. The second step involves the correspondence of detected objects in current frame with the previous to obtain their trajectories. High accuracy in object detection system results in less number of missing detection and finally produces less fragmented tracks. Better object association increases the affinity between objects in different frames. This paper presents a novel algorithm for improved object detection followed by enhanced object tracking. Object detection accuracy has been increased by employing deep learning-based Faster region convolutional neural network (Faster R-CNN) algorithm. Object association is carried out by using appearance and improved motion features. Evaluation results show that we have enhanced the performance of current state-of-the-art work by reducing identity switches and fragmentation.

Keywords: face-based tracking, target tracking, object detection, tracking

1. Introduction

We witness the truthfulness of the saying of Greek philosopher Aristotle "Man is by nature a social animal" in our daily life, as we see thousands of humans walk on roads, terminals, shopping malls, and other public places on a daily basis. They intentionally or unintentionally keep interacting with each other. They also make decision on where to go and how to reach their destinations. So their movement is not always straight away. It changes based on external environmental factors. Study and analysis of human dynamics play an important role in public security, public space management, architecture, and design. These tasks are highly dependent upon proper multi-person tracking (MPT) and trajectory extraction procedure. So this thing motivated us to contribute in the development of such system which performs these tasks with real-time speed and high accuracy.

Before moving further we have to know what is meant by multi-object tracking (MOT). The multiple object tracking is the process of localizing multiple moving objects over time using a camera as input or capturing device. A unique identity is

assigned to every detected object. This identity remains specific for that object up to the certain time period. Based upon these identities, we draw motion trajectories of objects that are being tracked. We can also make an analysis of the behavior of objects. **Figure 1** depicts the steps involved in multi-object tracking. Here to be noted that multi-object tracking is different from multi-object detection. Multi-object detection is sub-part of multi-object tracking. In short object detection is the process of locating object of interests in a single frame, while MOT is associated with detecting multiple objects of interests across a series of frames. In tracking the object detected in next frame should be able to relate same object detected in previous frame.

MOT has a variety of uses, some of which are human-computer interaction, surveillance and security, video communication and compression, augmented reality, traffic control, medical imaging, and video editing. Apart from these mentioned uses, there are some certain reasons which describe why tracking is useful.

- First of all, if there are too many objects detected in a video frame, then tracking will make it possible to establish the identity of certain objects across all the frames.

- Second, if there is the case that object detection has failed to detect object, then it may be still possible to track those objects because tracking system extracts and stores the location and appearance features of detected objects from the previous frame.

- Third, tracking methods could perform very efficiently because they perform a local search in place of a global. So, we can achieve a good performance in terms of high frame rate for our proposed hybrid system. The proposed system performs object detection for every nth frame and tracks the target object in intermediate frames based on their position in the frame and appearance features.

The proposed work has many applications in different fields, surveillance, entertainment, gaming, and autonomous vehicles, as shown in **Figure 2**. This system also has applications in crowded scene that enables the analysis of each individual moving opposite to the group movement. Visual surveillance usually requires the detailed human activity of each individual separately. Detecting activity for each person separately demands people to be tracked. Precise information of a person can be obtained by using previous trajectories of each subject. The analysis of drawn trajectories for each area can be helpful to find whether a person has been in forbidden area or not and performing what type of activities (running, walking, and fighting). Combining the tracking information of two or more characters can precisely elaborate the interaction between them.

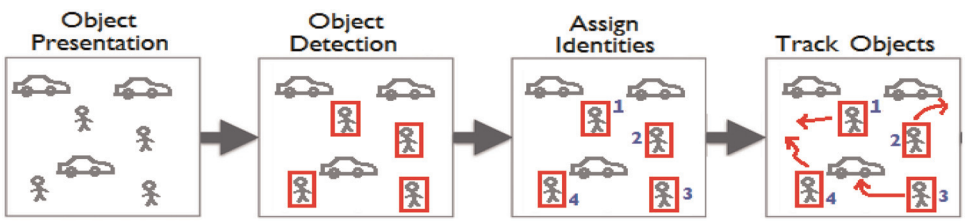

Figure 1.
Flow of steps involved in multi-object tracking.

Figure 2.
Applications of multi-object tracking system.

Multi-human tracking is also useful for multiplayer interactive games. Two humans playing fighting game can be easily tracked using MOT. Similarly, in autonomous vehicle industry, self-driving cars can employ tracking system to follow some specific vehicle. Based on the tracked vehicle, autonomous vehicles can take decisions.

We can track an object continuously after the first-time detection. But a tracking algorithm may sometimes lose track of the objects they are tracking. For instance, when the movement of the target object is too high, a tracking algorithm may not be able to maintain the track of the object. So the solution is to use detection and tracking algorithms together.

In recent years, there has been a lot of focus on MOT because of advancements in object detection techniques, which increased the robustness of tracking algorithms. So, stat-of-the-art techniques are way better than tradition ones. Most of traditional techniques do not perform well in real-time environments. For example, there are some batch-based movement tracking methodologies [1, 2] in which complete batch is required for tracking the human. Some others are probability-based systems for finding the track of the subject [3–5]. These methods require a complete batch of visual frames for processing and tracking the target object. But in real-time scenarios, there is a continuous stream of frames which are being fed to the system, and their number increases by time duration. So it is impossible to convert into batch and perform tracking on real-time bases.

1.1 Challenges in tracking systems

Recent advancements in object detections have made MOT more realistic. Multiple object tracking requires precise tracking of multiple objects based on apparent identity and relative position. MOT paradigm demands entire video batch at the same time and applies global optimization for finding the associations. Individuals in live stream need to be tracked based on history of each individual in the stream. The strong basic emphasis of this work is on the time efficiency and less number of identity switches.

The first and foremost topic of concern in the development of tracking system is time efficiency. Due to limitations of batch-based tracking algorithms, there are certain challenges to implement these systems in real-time scenarios. A plethora of work has been done to compete related challenges to make multi-person tracking and trajectory drawing real time and robust. A lot of efforts are being made to move tracking system from batch based to real time.

The second and important area of interest includes finding the solution of the problem of identity switches. Identity switch means how many times the particular object changes its identity number across all the frames or in a specific time duration. Identity switches mostly occur due to missed detection of object in between frames. The other reason is occlusion between different objects which causes identity switches.

The third one is fragmentation issue. Fragmentation occurs when identity switch does not occur but detection of object is missing in some frames, and due to this a fragmented trajectory is generated, meaning tracking breaks where human is not detected.

As we have understood that the problem of identity switches is due to missing detections. So there are two possible solutions for this problem. Solution one is to reduce or eliminate the number of missing object detection in frames. This can be done by improving the object detection algorithm. We have proposed a solution for this problem, which uses the deep learning-based algorithm to detect a person by detecting their shoulder, head, or complete body.

But the problem of identity switches will still persist due to occlusion mechanism. Here comes the second solution, which is to make use of appearance and localization features of objects. As every object have a different location in the frame and different appearance relative to each other, we can build a tracking algorithm which will track multiple objects in a series of frames based on these features. Face recognition can also be used to reduce the number of identity switches. Based upon appearance and motion features, we can relate a trajectory fragmentation of one object with the other fragmentation of that object and can complete the trajectory path. The complete process is shown in **Figure 3**.

One of the recent available tracking systems is simple online and real-time tracking (SORT) [6] that tried to overcome these challenges. It is a simple framework to track persons in real time. SORT utilizes the Kalman filter features on input frames. Hungarian algorithm is employed to find the association in visual tracks. Their proposed system is only applicable for human tracking in different appearance scenes. This system still involves identity switch problem.

Another tracking system, simple online and real-time tracking with a deep association metric (Deep SORT) [7], utilized apparent features extracted from deep convolution neural network (CNN) for tracking the individuals. Deep SORT generates a cost matrix based on motion information and appearance features to avoid missing tracking because of occlusion or missed detection of persons. Their system includes convolution neural network for a person's apparent features trained on person reidentification dataset. Deep SORT has a high rate of missed detection for elevated view, crowded view, and distant view because detected humans are obtained from pre-trained models of object detection.

In this paper, we proposed a novel technique for individual human detection and tracking. We provide a unique real-time tracking using motion information and appearance information. Our system employs convolution neural network to provide the visual appearance features for tracking the individual. The CNN for visual

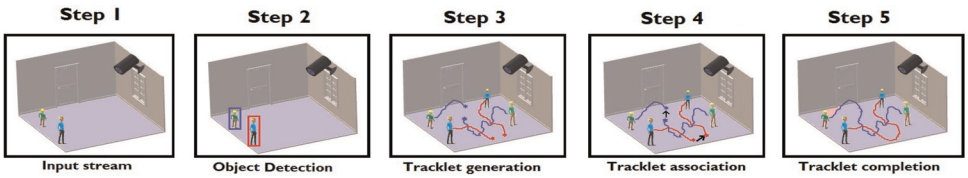

Figure 3.
Fragmentation removal steps.

appearance is trained on person reidentification dataset [8]. Appearance features allow us to improve the tracking results by making it robust for occlusion and detection after multiple frames.

1.2 Major contribution

The proposed system improved the overall detection and tracking problem in MOT problem. Furthermore, we have also improved the detection of human by retraining the Faster region convolutional neural network (Faster R-CNN) on human body parts. Provided better hardware resources, the proposed system outperforms the existing state-of-the-art systems in terms of accuracy and performance. Major contributions of our system are manifold.

- Improved detection by Faster R-CNN trained on human pedestrian dataset from different views.
- We also improved the feature set for tracking the subjects that includes color features, area, mutual distance, and HSV histograms for each region of interest.
- The system behaves better than both SORT and Deep SORT in real-time scenarios for pedestrian tracking.

The rest of the paper is divided into different sections. Section 2 provides the detailed background of previous methodologies for MOT systems. Methodology is described in Section 3 along with complete architecture. Section 4 throws light on experimentation. The last section concludes the proposed system for human tracking in a different environment.

2. Background

With the improvement in multi-object detection, research community has started focusing on tracking of every single object in different environments. The complete MOT problem can be considered as an association problem in which the basic objective is to associate the detected objects. Tracking is carried out after object detection using some object detector. In this section, we will focus on the background of the following systems:

- Object detection algorithms as humans need to be detected before person tracking
- Face recognition systems for target tracking based on recognized faces
- Tracking algorithms for reviewing already available tracking algorithms

2.1 Object detection in past years

In the early 1990s, object detection was carried out using template matching based algorithms [9], where a template of the specific object is slid over the input image to find the best possible match in the input image. In the late 1990s, the focus was shifted toward the geometric appearance-based object detection [10, 11]. In these methods, the basic focus was on height, width, angles, and other geometric properties.

In the 2000s, object detection paradigm was transferred to low-level features based on some statistical classifiers such as local binary pattern (LBP) [12], histogram of oriented gradient [13], scale-invariant feature transform [14], and covariance [15]. Feature extraction-based object detection and classification involved training of machine based on extracted features.

For many years in computer vision field, handcrafted traditional features were used for object detection. But, with the progress in deep learning after accomplishing the remarkable performance in 2012 image classification challenge [16], convolution neural networks are being used for this purpose. After the success of object classification in [16], researchers transferred their attentions toward object detection and classification. Deep convolution neural networks work exceptionally good for extraction of local and global features in terms of edges, texture, and appearance.

In recent years, the research community has moved in the direction of region-based networks for object detection. This type of object detection is being used in different applications like video description [17]. In region-based algorithms for object detection, convolution features are extracted over proposed regions followed by categorization of the region into a specific class.

With the attractive performance of AlexNet [16], Girshick et al. [18] proposed the idea of object detection using convolution neural network. They employed selective search for proposing the areas where the potential objects can be found [19]. They called their object detection network as region convolution neural network (R-CNN). The basic flow of region convolution neural network (R-CNN) can be described as follows:

- Regions are proposed for each object in the input image using selective search [19].
- Proposed regions are resized to same consistent size for classification of the proposal into predefined classes based on extracted CNN features of regions.
- Linear SVM classifier replaced the softmax layer for training the system on fixed length CNN features.
- Finally, a bounding box regressor is utilized for perfect localization of object.

Although the proposed R-CNN was a major breakthrough in the field of object detection, it has some significant weaknesses:

- Training processes is quite slow because R-CNN has different separate stages to train.
- Regions are proposed by selective search that is itself a slow process.
- Training the separate SVM classifier is expensive as CNN features are extracted for each individual region that makes the training of SVM even more challenging.
- Object detection is slow because CNN features are extracted for individual proposal for each testing image.

To overcome the feature extraction issue for each proposal, Kaiming He et al. [20] proposed spatial pyramid pooling (SPP). The basic idea was that the

convolution layers accept the input of any size; fully connected layers force input to be fixed size for making matrix multiplication possible. They used SPP layer after last convolution layer for obtaining the fix-sized features to feed in fully connected layer. Using SPPNet R-CNN performance improved comprehensively. SPPNet extracts convolution features on input image only once for proposals of different sizes. This network improves the performance of testing, but it does not improve the performance of training the R-CNN. Furthermore, weights of convolution layers before SPP layer cannot be changed which limits the fine-tuning process.

The main contributor of R-CNN, Girshik et al. [21], proposed Fast-ECNN to address some problems of R-CNN and SPPNet. Fast R-CNN employs the idea of computation sharing of convolution for different proposed regions. It adds region of interest (ROI)-pooling layer after the last convolution layer for generating fix-sized features of individual proposals. The fix-sized features from ROI-pooling layers are fed to the stack of fully connected layers that further split down into two branch networks: one acts as the object classification network and the other for bounding box regression. They claimed that the overall performance of training step of R-CNN is enhanced by three times and ten times for testing.

Although Fast R-CNN improved the performance of R-CNN notably, it still uses selective search as a region proposal network (RPN). Region proposal step consumes the time comprehensively that acts as the bottleneck in Fast R-CNN. Modern advancements in object localization using deep neural network [22] motivated Ren et al. [23] to employ CNN for replacing slow process of region proposal using selective search. They proposed efficient RPN for proposing proposals for objects. In Faster R-CNN, RPN and Fast R-CNN share the convolution layers for region proposal and region classification, respectively. Faster R-CNN is a purely convolution neural network without any handcrafted features that employ fully convolution neural network (FCN) for region proposal. They claimed that Faster R-CNN can work at 5fps for testing phase.

Redmon et al. [24] proposed You Only Look Once (YOLO) for object detection. They completely dropped the region proposal step; YOLO splits the complete image into grids and predicts the detection on the bases of candidate regions. YOLO divides the complete image into $S \times S$ grids. Each grid has a class probability C, B as the bounding box locations and a probability for each box. Removing the RPN step enhances the performance of the detection; YOLO can detect the objects while running in real time with about 45 fps.

2.2 Face recognition over past years

In current era, biometric identification systems are required more than ever because of the improved security requirement in the globe. There have been a lot of efforts by researchers for face recognition technology (FRT). The basic division of FRT can be the traditional handcrafted feature-based identification and deep learning-based identification.

2.2.1 Handcrafted feature-based identification

Eigenface [25] and Fisherface [26] were commonly used approaches in the last decade for face identification. Eigenfaces reduced the feature points for measuring maximum change in face features using minimum set of features. For reducing the features, they used principal component analysis (PCA). Linear face can be recognized based on linear structure of the face using Eigenfaces. In contrast with the Eigenfaces, Fisherfaces are a supervised learning-based face identification method based on traditional texture features. Fisherfaces employ linear discriminator

analysis for finding the uniquely describing data points. Both of these methodologies extract features in terms of Euclidean distance to identify the face.

Researchers have also used LBP for facial recognition [27, 28]. Hadid et al. [27] exploited LBP features for face recognition. They worked on face detection and recognition. Face detection was achieved by training a support vector machine of second degree on extracted features. Face recognition was achieved using LBP-based texture descriptor. Machine was trained on these descriptors for face recognition.

2.2.2 Deep learning-based identification

Advancement in convolution neural networks has achieved remarkable performance by increasing accuracy and efficiency. The very basic assumption in deep neural networks is to feed as much data as possible for getting better results. Requirement of huge data makes deep learning-based approaches data hungry.

Lu et al. [29] implemented a residual network (ResNet)-based model for face recognition. They divided their complete network into three networks: one backbone network called trunk network and two other networks called branch networks that emit from trunk network. The central network is trained once for learning the deep features for face identification. The central network is generated using residual blocks. Resolution-specific coupled mapping is employed in branch network for training. Input image and comparison image from gallery are transformed to same representation for comparing. Based on distance the decision is made about identified face.

Schroff et al. [30] developed a deep neural network based on convolution neural network and then named it as FaceNet. Their proposed system extracts the feature space in terms of Euclidean space. They optimized the feature mapping of facial structure using deep convolution neural network. Their proposed system, FaceNet, generates a feature vector of 128 dimensions that is optimized using triplet loss. Their proposed triplet loss comprises three face images: two from the same pair and one from a separate individual. The loss function tries to separate the same individual faces from different individual faces. Their triplet loss function is trained to minimize the distance between the same identity faces and maximize the distance between different identities. Inception model with little modification is employed in FaceNet for extracting convolution features. They tested their system on LFW dataset [31].

A research group from Facebook, Taigman et al. [32], developed a state-of-the-art system for face alignment and face recognition, named as DeepFace. They used deep convolution neural network having nine convolution layers for extracting facial features. Facial landmarks are used in their system for face alignment. The facial landmarks are estimated using support vector regressor (SVR). Extracted features from nine-layered network are passed to Softmax layer for classification. They employed cross-entropy to reduce the loss of correct labels. They also proposed a huge face recognition dataset named as Social Face Dataset [32]. They used their dataset for training the system for face identification.

2.3 Multi-object tracking

Multiple researchers have focused on movement and spatial features for tracking the multiple objects [33, 34]. Some of the researchers have focused on appearance features for capturing the associations between different detections [2, 35].

There are some traditional methods that make prediction on frame-by-frame basis. These traditional approaches involve multiple hypothesis tracking (MHT) [36] and joint probability data association filter (JPDAF) [37]. Both of these old methodologies require a lot of computation for tracking the detected objects. The complexity of these methodologies increases exponentially with increasing the number of trackable objects that makes them really slow to be used for online applications in complex environment. In JPDAF hypothesis of single state is generated based on relation between individual measurement and association likelihood. In MHT, a complete set of hypotheses is taken into consideration for tracking followed by post pruning for tractability.

Rezatofighi et al. [1] made an effort to improve the JPDAF performance by providing approximation of JPDA. They exploited recent advancement in solving m-best solution for an integer program. The main advantage of this system is to make JPDA less complex and more tractable. They redefined the method for calculating individual JPDAF assignment in terms of a solution to a linear program. Another group of researchers Kim et al. [2] used appearance-based features for tracking the target. They improved the MHT by pruning the graph of MHT for achieving state-of-the-art performance. They employed regularized least squares for increasing the efficiency of the MHT methodology.

These two improvements perform quite well as compared to the legacy implementations, but these two methods still have much delay in the decision-making step which makes these methods inappropriate for real-time applications. These methods require large computational resources with increasing the individual density.

Some researchers worked on graph theory for tracking human. Kayumbi et al. [38] proposed an algorithm to find football players' trajectories based on distributed sensing algorithm in multi-camera view. Their algorithm starts with mapping of camera view plane to virtual top-view of the ground plane. Finally, they exploited graph theory for tracking each individual in the ground plane.

Some online tracking methods utilize appearance features of individuals for tracking [39, 40]. These models extract apparent look features of individuals. Both of the systems provide accurate appearance descriptors for providing guidance to data association. First system incorporates temporal appearance of individuals along with the spatial appearance features. Their appearance model is learned by applying incremental evaluation after tuning the parameters in each iteration. In the second system, Markov decision process (MDP) is employed to map the age of the detected object in terms of Markov chain. MDP decides the tracks based on current status and history of the target.

Recently, some of the researchers worked on simple online tracking and tried to make tracking real time in live stream [6, 7]. These systems are named as simple online and real-time tracking and simple online and real-time tracking with a deep association metric, respectively. Both of these systems are two successive versions of the same methodology. In both systems Kalman filter is employed to find the movement features of the target. These systems used intersection over union, central position, height, width, and velocity as the core features for tracking. In Deep SORT, convolution features for targets appearance are also used along with motion features to reduce the missing tracks after occlusions and missed detections in multiple frames. Despite the real-time performance, these systems miss tracks after the changed posture and missed detection in a large number of frames.

Our proposed system reduces the limitation of missed detection of the human body, and it also reduces the track missed by incorporating extra features and better human detection system.

3. Methodology and framework

Multi-object tracking (MOT) in real time with good accuracy has been a challenge from decades. Many systems have been developed for this task during the last few decades, using tradition computer vision techniques. But due to rebirth of deep learning, object tracking has been robust. As object detection is the backbone of object tracking systems and deep learning techniques are good at object detection problem with real-time speed and accuracy, it is a better choice to use deep learning algorithm for detection purpose.

We have solved the MOT problem using state-of-the-art techniques. The proposed method is explained by the key components of human detection, position prediction of objects in future frames, tracklet associations, and managing the life span of identities for tracked objects. The mostly used state-of-the-art object detection algorithm is YOLO [24] which is fast enough to detect multiple objects in real time, but it has the problem of missed detections which leads to fragmentation and identity switch problems. So we conduct proper survey to choose the best detection algorithm for the problem. We have divided this methodology into sub-components for detection, track handling, and association as follows:

- Faster R-CNN for human detection
- Kalman filter
- CNN for appearance features
- Hungarian algorithm for tracking nearby rectangles
- Additional features like area, relative distance, color, nearest color, and HSV

The basic modules of the proposed system are described in the following sections.

3.1 Person detection using Faster R-CNN

As mentioned above, with the advancement in deep learning-based algorithms, real-world object detection has been a lot easier. So we have employed Faster Region Convolutional Neural Network (Faster R-CNN) detection network [23].

There are two stages of Faster R-CNN. In the first stage, region proposal network (RPN) generates the anchors on the regions present in the image where there might be a high possibility of the presence of an object. This process is further divided into three steps:

1. First step involves the process of feature extraction by using convolution neural network. Convolution feature maps are generated at the end of last layer.
2. In second step, a sliding window approach is used on these feature maps to generate anchor boxes. These anchor boxes are further refined in the next step to indicate the presence of objects.
3. Finally, in the third step, generated anchors are refined using a smaller network which calculates the loss function to select top anchors containing objects.

For region proposal network, prerequisite step is extraction of convolution features that are extracted using backbone network.

3.1.1 Residual Network-30 (backbone network)

As object detection problem is dependent upon feature extraction process to produce good quality proposals. So we used ResNet-based model containing 30 layers named as ResNet-30. ResNet or residual networks are special type of convolution neural networks which have residual connections in between layers. The benefit of these residual connections is that the network is able to learn local, global, and intermediate features in parallel, making it more efficient as compared to simple CNN. Residual connections also help in avoiding vanishing gradients problem, which is a major issue in networks containing high number of layers. So, ResNet-30 is able to learn more patterns than simple CNN by grasping more information. There are two types of short connections used in ResNet in different scenarios as described below:

1. In the first case, when the inputs and outputs are of the same dimensions, shortcuts (x) can be used directly. As illustrated in Eq. 1

$$l = F(x, W_i) + x \tag{1}$$

2. In the second case, we have changed dimensions, and the identity mapping is performed by padding extra zero entries to make dimension suitable. Another option is to use the projection shortcut to match the dimension (done by 1×1 conv) using Eq. 2:

$$l = F(x, W_i) + W_j x \tag{2}$$

where W is the weight matrix, x is the feature vector from previous layer, and F is the convolution function. Pictorial representation for residual block is given in **Figure 4**.

3.1.2 Anchor generation

Now to propose the regions in image which contains the high probability of presence of objects, the sliding window approach is used. A sliding window moves across the feature maps to generate anchors. The sliding window has the size of

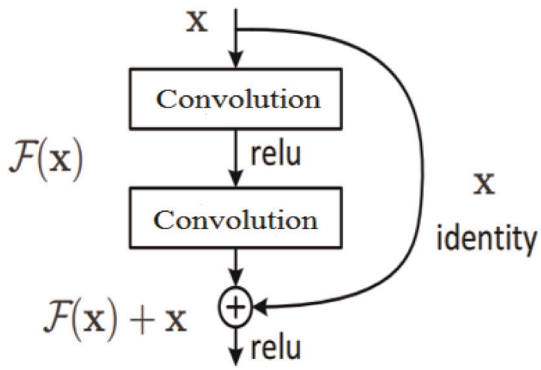

Figure 4.
Basic building block of residual learning.

$n \times n$. In our case $n = 3$; this means a 3×3 window is used. A set of nine anchors is generated for each pixel, having the same center (x, y) for all anchors. All nine anchors have three multiple aspect ratios and three varieties in scales. **Figure 5** represents the nine anchors having the same center point. Anchors with the same color have the same aspect ratio but different scaling. An intersection of union (IoU) approach is used to determine how much of these anchors overlapped with ground-truth bounding boxes. A threshold value is set based on IoU. Mostly anchors are discarded and some are selected using threshold. Anchors having IoU value >0.7 are considered as object-containing regions and anchors with value <0.3 considered as background. Eq. 3 represents the formula to find the probability of object based upon IoU value.

$$IoU = \frac{Anchor \cap Gt}{Anchor \cup Gt} \begin{cases} > & 0.7 & = & Object \\ < & 0.3 & = & NotObject \end{cases} \tag{3}$$

3.1.3 Loss function

The selected anchors are further fine-tuned using the loss function at the end of region proposal network. A shallow network is used for this purpose which performs two tasks: classification and regression. The classification performed here is binary classification which classifies anchors in one of two classes. The first class is object and the second is background.

The output of regressor determines the position of predicted bounding box in terms of four parameters (x, y, w, h), where x and y indicate the center point of anchor, w stands for width, and h represents the height of anchor box. The formula of loss function which calculates the loss for both regression and classification is given in Eq. 4:

$$L(p_i, r_i) = \frac{1}{N_{cls}} \sum_i G_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i P_i^* G_{reg}(r_i, r_i^*) \tag{4}$$

RPN is trained to propose regions of interest (ROIs) on feature maps which are obtained from input image. These RoIs are enclosed in bounding boxes. RPN outputs different scales of bounding boxes, on feature maps. These bounding boxes contain high probability of presence of objects.

Now comes the second stage of Faster R-CNN, which is a classification of ROIs obtained from RPN network. To bring the ROIs in feedable format for the classifier, a ROI-pooling method is used which uses the pooling mechanism to shape all RoIs in the same scales. Its purpose is to perform max pooling on inputs of nonuniform sizes to obtain fix-sized feature maps for each RoI.

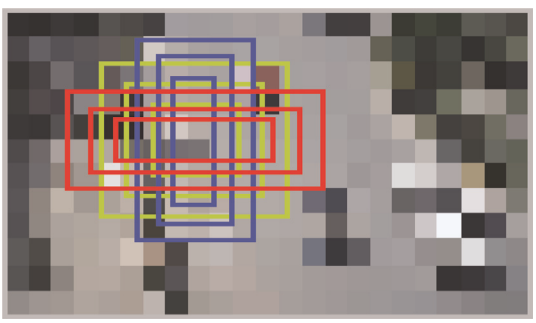

Figure 5.
Nine different proposed anchors for each single point.

3.1.4 ROI classification and regression

Now same-sized feature maps or RoIs obtained from RoI-pooling are further proceeded for classification and regression purpose. This step runs two stages in parallel. Bounding box classification and regression loss are calculated based on the optimization of the loss function. Classification head results in the class score for each individual category, and regression head resizes the bounding box values (x, y, w, h) to cover complete object. Overall performance and accuracy of Faster R-CNN is better than all the traditional object detectors. A diagram for Faster R-CNN is given in **Figure 6**.

We have trained Faster R-CNN on 4000 annotated images of human heads, shoulders, and complete bodies which improved detection accuracy efficiently having only few numbers of miss rate.

3.2 Track handling and state estimation of future frames

Kalman filter is used in its standard form as proposed in [41]. We have defined tracking scenario on the multidimensional state space $(x, y, \gamma, h, s, t, u, v)$ that consists of the bounding box center location (x, y), with height h, their respective velocities (s, t, u, v) in image coordinates, and aspect ratio γ. We keep on calculating the total count of frames for every individual track T, starting from previous correct association S_T. When Kalman filter predicts opposite features then this counter is incremented. When the track is assigned with a previous list, then the counter is reset to 0. In those tracks that are newly detected and cannot be assigned to any of the current list, then new track prediction is initiated. For the first three frames, these tracks are classified as tentative. If the association of a measurement at every time step t is found, the tracks are kept for further processing and otherwise deleted.

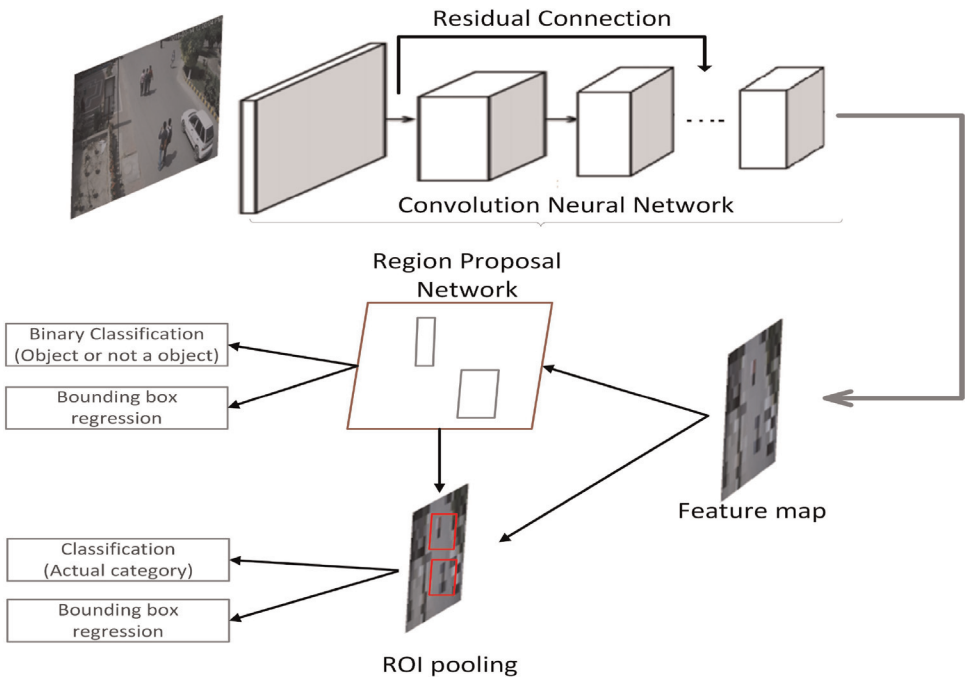

Figure 6.
Complete framework diagram of Faster R-CNN.

3.2.1 Association of newly predicted states and current states

A traditional approach to find association between the current Kalman states and newly arrived detections is to use the Hungarian algorithm. We integrated spatial displacement and apparent features by creating two different metrics. For motion information, we used Mahalanobis distance between current list of states and newly arrived states. The Mahalanobis distance removes state estimation uncertainty by measuring deviations between detection and mean track location. Further, false associations can be excluded by thresholding at a 90% confidence interval computed from the inverse x^2 distribution. We have set the value of threshold t as 9 for the decision based on Mahalanobis distance.

Mahalanobis distance matrix provides robust association metric when the overall motion transition is not high and the Kalman filter framework supply only a vague approximation of the object position. Specifically, rapid movement of capturing device can lead to displacements, making it uninformed metric for tracking in the presence of occlusions. Therefore, we integrate a second metric for tracking the pedestrians; we have utilized a pre-trained convolution network to extract the bounding box appearance features. The complete architecture of the proposed convolution neural network is shown in **Table 1**.

In combination, both techniques support each other by handling different aspects of association problem. In particular, the Mahalanobis distance matrix is employed to extract information about object positions based on movement of the objects for a short period. Along with the distance matrix, we have employed convolution for appearance feature descriptor; the CNN considers appearance information for those long-term occluded detection that are not possible to be captured through motion features. Moreover, we have used some additional features like area of human, relative distance between tracked humans, color or nearest color of object, and HSV to handle occlusions quite efficiently.

Area of human can accommodate the occlusion problem quite well because the area mostly remains the same during the whole tracking, for example, if a person is short, they will remain short before and after occlusion which makes it easy to reidentify a person after a long-time occlusion. Similarly, if someone is sitting on a wheel chair, their area will be the same throughout the time of tracking.

Name	Filter size	Stride	Output size
Conv 1	3×3	1	$32 \times 128 \times 64$
Conv 2	3×3	1	$32 \times 128 \times 64$
Max pool 1	3×3	2	$32 \times 64 \times 32$
Residual block 1	3×3	1	$32 \times 64 \times 32$
Residual block 2	3×3	1	$32 \times 64 \times 32$
Residual block 3	3×3	2	$64 \times 32 \times 16$
Residual block 4	3×3	1	$64 \times 32 \times 16$
Residual block 5	3×3	2	$128 \times 16 \times 8$
Residual block 6	3×3	1	$128 \times 16 \times 8$
Dense layer 1		—	128
Batch norm		—	128

Table 1.
Complete architecture for appearance feature extractor network.

Relative distance is another important appearance feature to keep track updated. Let us assume if the targeted person is moving in a group with few other people, their relative distance can be used to keep track of target even after the long-term occlusion.

Color or nearest color can be helpful to reidentify after occlusion because humans mostly keep the same clothes within a session. Similarly, if the nearest color is predicted, the person can be reidentified even after a sudden change of lighting or contrast.

HSV and RGB histograms are employed for comparing the histogram based on object appearances. We compare the histogram appearance model in both color spaces using cumulative brightness transfer function (CBTF) as mapping function between the two fields of views, which helps us handle occlusions in a better way.

3.3 Tracking options

In this system we have provided multiple options for enabling the tracking. Major tracking options include (1) face recognition-based tracking and (2) target tracking.

3.3.1 Face recognition-based tracking

As the object tracking system trained by us also detects faces, so we take benefit of this approach and make use of these detected faces in our tracking system. We used the face recognition proposed by Lu et al. [29] to recognize the detected faces. When any new face enters in frame, then the system extracts and stores these features by assigning a specific ID for later use. These feature maps and IDs are used in the future to associate the face detected with saved faces. That's how data association accuracy increased and helped in better tracking. But the limitation of this system is that it works only in case-detected face that is visible enough. This system works within 15 to 20 feet distance from camera. It also depends upon camera resolution; as the resolution is high, the better feature will be extracted.

3.3.2 Target tracking

Developed system can also perform specifically selected object tracking task. We have to basically select the object which we want to track by clicking on the detected object. This system enables us to perform analysis of only desired object by hiding the tracking details of other objects. The system is a practical implementation of human-computer interaction facilitated by the user. The overall tracking of pedestrians is improved based on robust detection of human using multiple views and body parts (head, shoulder, and complete body). Furthermore, the problem of identity switches and fragmentation is addressed by appearance features and increased spatial features (area, relative distance, color, and histograms).

4. Evaluation

Training of proposed detection system is performed on self-generated dataset, and for tracking purpose, we employed standard tracking dataset for evaluating the overall system.

4.1 Environment setup

To setup environment, we used GeForce GTX 1080 Ti GPU with Ubuntu OS installed in the system. We chose Python programming language to perform the experimentation steps. System is built by using Tensor Flow framework. We evaluated five different models of ResNet integrated with Faster R-CNN architecture on self-generated dataset, and the results are further discussed in Section 4.4.

4.2 Self-generated dataset

For training the detection system, we have utilized self-generated dataset having 4000 image of human in different postures. Each image in the dataset contains on average five different subjects with limited repetition in other images. The dataset has images from different environment conditions (rain, snow, and shadow) and in different lighting conditions (day and night). Some of the images are collected over the Internet, and some are generated within university premises using surveillance cameras. The dataset is comprehensive in terms of human densities, view angle, posture, and scale. The dataset covers different scenes: streets, bazars, buildings, malls, parks, roads, and stadium. **Figure 7** shows some images from self-generated dataset.

We have annotated human body parts in different categories. The visible human body can be categorized into three classes based on occlusion and density of the crowd. These classes include the head, shoulder, and complete body. Based on the visible category, we have annotated the images. The details of each annotated category in the dataset are shown in **Table 2**.

4.3 MOT benchmark dataset

For evaluating our proposed tracking system, we have employed multiple object tracking (MOTChallenge) benchmark dataset [42] that contains a variety of

Figure 7.
Sample images from self-generated dataset.

Category	Instances	Number of images
Head	4325	987
Shoulder	3130	2031
Complete body	12,690	3411

Table 2.
Each category division in proposed dataset.

sequences with dynamic camera and static camera. In this dataset, they have combined 22 different available datasets. Some sample images from MOTChellange dataset are shown in **Figure 8**.

They have provided a 10-minute video of individuals with 61,440 rectangles of human detection. This dataset is composed of 14 different sequences with proper annotations by expert annotators. They also annotated different objects like chair and car for better representation of the occlusions. Three different aspects of MOTChallenge are described below:

1. Dynamic or static stream: A camera while capturing can have multiple states, placed on a stroller, in a car or a person holding the camera that makes it dynamic and static.

2. Viewpoint variation: Video camera can be elevated, at same height as pedestrian, or at low position.

3. Weather conditions: The weather condition of the captured stream is also provided with sequences to get the idea of lighting, shadows and blurring of the pedestrians.

4.4 Results

We integrated and tested multiple ResNet backbone network architectures in Faster R-CNN-based object detection. We evaluated the networks based on detection accuracy and performance. After proper evaluation, we found that ResNet-30

Figure 8.
Samples from MOT dataset. Top, training images; bottom, test sequences.

Model (residual networks)	Layers	Top-1 error	Top-5 errors (avg.)	Runtime (ms)
ResNet-34	34	25.27	8.51	52.09
ResNet-50	50	23.93	7.82	104.13
ResNet-101	101	22.81	7.11	158.35
ResNet-152	52	22.52	6.63	219.06
ResNet-30 (proposed)	30	26.02	8.04	48.93

Table 3.
Comparison table of different ResNet architectures as backbone network for object detection.

		MOTA	MOTP	MT	ML	ID	FM	FP	FN	Runtime
KDNT [43]	Batch based	68.2	79.4	41.0%	19.0%	933	1093	11,479	45,605	0.7 Hz
LMP p [44]	Batch based	71.0	80.2	46.9%	21.9%	434	587	7880	44,564	0.5 Hz
MCMOT HDM [45]	Batch based	62.4	78.3	31.5%	24.2%	1394	1318	9855	57,257	35 Hz
NOMTwSDP16 [46]	Batch based	62.2	79.6	32.5%	31.1%	406	642	5119	63,352	3 Hz
EAMTT [47]	Real time	52.5	78.8	19.0%	34.9%	910	1321	4407	81,223	12 Hz
POI [43]	Real time	66.1	79.5	34.0%	20.8%	805	3093	5061	55,914	10 Hz
SORT [6]	Real time	59.8	79.6	25.4%	22.7%	1423	1835	8698	63,245	60 Hz
Deep SORT [7]	Real time	61.4	79.1	32.8%	18.2%	781	2008	12,852	56,668	40 Hz
Proposed system	Real time	75.2	81.3	33.2	17.5	825	1225	4123	52,524	42 Hz

Table 4.
Evaluation table.

performed better in our case as our major concern is minimum runtime with maintaining the accuracy. Response time of our system is 25 frames per second. So, we decided to use ResNet-30 in designed system for better speed and accuracy. The evaluation matrix for different tested models is given in **Table 3**.

As **Table 3** depicts, the runtime for ResNet-30 is lower than other ResNet models, and Top-5 error rate is not significantly low. Based on this evaluation, ResNet-30 was our ultimate choice for backbone architecture of Faster R-CNN.

Table 4 provides the evaluation results of our complete tracking system on MOT dataset. This evaluation provides results of our designed system on seven challenging test sequences, on human eye level and elevated view of camera scenes. The tracking system highly relies on detection mechanism to perform better detection followed by better tracking; we used Faster R-CNN trained on our self-collected dataset. We rerun Deep SORT on the same evaluation dataset for fair comparison.

We set threshold of 0.7 for detection confidence score. We further fine-tuned the other parameters of network to produce better model. Following metrics are used for comparison purpose:

- Multi-object tracking accuracy (MOTA): It provides complete accuracy of system incorporating with false negatives, identity switches, and false positives.

- Multi-object tracking precision (MOTP): It gives complete tracking precision for bounding boxes overlapping between predicted location and ground-truth value.

- Mostly tracked (MT): It is the percentage of ground-truth tracks that do not change their labels at least during 80% of their life span.

- Mostly lost (ML): It provides the percentage of actual tracks that are being tracked by the system at most 20% of their life span.

- Identity switches (ID): It defines the total reported identity changes of ground-truth tracks.

- Fragmentation (FM): It provides the detail of how many times the track is interrupted by missed detection of person.

The results of our evaluation are shown in **Table 4**. The numbers of identity switches have been reduced due to our alteration in detection network. As compared to Deep SORT [7], MOTA is increased from 61.4 to 75.2.

5. Conclusion

The goal of this work was to implement a fast and competitive MOT system. We presented a multiple object tracker that combines a deep learning-based object detection network named as Faster R-CNN with the tracking algorithm. The proposed system performed tracking by detecting multiple objects followed by assigning each object a unique ID and generating their tracklets. In the case of fragmentation in tracklet of any object, the system uses tracklet association mechanism to generate a complete trajectory. Tracking is performed based upon appearance and motion features of objects. When in any frame object detection network fails to detect objects, these features are used to track object again with the same ID. Kalman filter and Hungarian algorithm both collectively used to predict the position of object in the frame. Other features like area, color, relative distance, nearest color, and HSV histograms are also used to increase the tracking accuracy. Overall the system performed very well, and it has shown improvement in MOTA, MOTP, ML, and FP fields as shown in comparison in **Table 4**. But considering environmental constraints and hardware limitations, our system has some pros and cons. We have listed some strengths and weaknesses as follows.

5.1 Pros

- Efficient in terms of response time because of less number of layer of residual network

- Have minimum number of missing detections because of improved object detection process

- Less fragmentation in drawn trajectories because of continuous detection of persons in consecutive frames

- Introduction of visual and motion feature-based tracking for reducing identity switches
- Trajectory completion in the case of fragmentation

5.2 Cons

- It requires GPU-based hardware for enabling real-time tracking.
- For highly dense crowd identity, switches may occur because of similar features of the head for each person.
- Performance reduces in dark environmental condition.

Acknowledgements

This work is carried out at UET Lahore under Intelligent Criminology Lab, National Center of Artificial Intelligence.

Author details

Gulraiz Khan[1], Zeeshan Tariq[1] and Muhammad Usman Ghani Khan[2*]

1 Al-Khawarizmi Institute of Computer Science, UET Lahore, Pakistan

2 Department of Computer Science and Engineering, UET Lahore, Pakistan

*Address all correspondence to: usman.ghani@kics.edu.pk

References

[1] Rezatofighi AM, Zhang Z, Shi Q, Dick A, Reid I. Joint probabilistic data association revisited. In: Proceedings of the IEEE International Conference on Computer Vision. 2015. pp. 3047-3055

[2] Kim C, Li F, Ciptadi A, Rehg JM. Multiple hypothesis tracking revisited. In: Proceedings of the IEEE International Conference on Computer Vision. 2015. pp. 4696-4704

[3] Yang B, Nevatia R. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE. 2012. pp. 1918-1925

[4] Andriyenko A, Schindler K, Roth S. Discrete-continuous optimization for multi-target tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE. 2012. pp. 1926-1933

[5] Milan A, Schindler K, Roth S. Detection-and trajectory-level exclusion in multiple object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013. pp. 3682-3689

[6] Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP); IEEE. 2016. pp. 3464-3468

[7] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP); IEEE. 2017. pp. 3645-3649

[8] Zheng L, Bie Z, Sun Y, Wang J, Su C, Wang S, et al. Mars: A video benchmark for large-scale person re-identification. In: European Conference on Computer Vision; Springer. 2016. pp. 868-884

[9] Jain AK, Zhong Y, Lakshmanan S. Object matching using deformable templates. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1996;**18**(3):267-278

[10] Mundy JL. Object recognition in the geometric era: A retrospective. In: Toward Category-Level Object Recognition. Springer; 2006. pp. 3-28

[11] Ponce J, Hebert M, Schmid C, Zisserman A. Toward Category-Level Object Recognition. Vol. 4170. Springer; 2007

[12] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002;**24**(7):971-987

[13] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005; volume 1; IEEE. 2005. pp. 886-893

[14] Lowe DG. Distinctive image features from scale-invariant key points. International Journal of Computer Vision. 2004;**60**(2):91-110

[15] Tuzel O, Porikli F, Meer P. Region covariance: A fast descriptor for detection and classification. In: European Conference on Computer Vision; Springer. 2006. pp. 589-600

[16] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. 2012. pp. 1097-1105

[17] Khan G, Ghani MU, Siddiqi A, Seo S, Baik SW, Mehmood I, et al. Egocentric visual scene description based on human-object interaction and

deep spatial relations among objects. Multimedia Tools and Applications. 2018:1-22

[18] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014. pp. 580-587

[19] Uijlings JRR, Van De Sande KEA, Gevers T, Smeulders AWM. Selective search for object recognition. International Journal of Computer Vision. 2013;**104**(2):154-171

[20] He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European Conference on Computer Vision; Springer. 2014. pp. 346-361

[21] Girshick R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. 2015. pp. 1440-1448

[22] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. pp. 2921-2929

[23] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards realtime object detection with region proposal networks. In: Advances in Neural Information Processing Systems. 2015. pp. 91-99

[24] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. pp. 779-788

[25] Turk MA, Pentland AP. Face recognition using eigenfaces. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91; IEEE. 1991. pp. 586-591

[26] Kwak K-C, Pedrycz W. Face recognition using a fuzzy fisherface classifier. Pattern Recognition. 2005; **38**(10):1717-1732

[27] Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006;**28**(12):2037-2041

[28] Hadid A, Pietikainen M, Ahonen T. A discriminative feature space for detecting and recognizing faces. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004; volume 2; IEEE. 2004

[29] Lu Z, Jiang X, Kot ACC. Deep coupled resnet for low-resolution face recognition. IEEE Signal Processing Letters. 2018

[30] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. pp. 815-823

[31] Huang GB, Mattar M, Berg T, Learned-Miller E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition. 2008

[32] Taigman Y, Yang M, Ranzato MA, Wolf L. Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2014. pp. 1701–1708

[33] Dicle C, Camps OI, Sznaier M. The way they move: Tracking multiple targets with similar appearance. In: Proceedings of the IEEE International Conference on Computer Vision. 2013. pp. 2304-2311

[34] Yoon JH, Yang M-H, Lim J, Yoon K-J. Bayesian multiobject tracking using motion context from multiple objects. In: 2015 IEEE Winter Conference on Applications of Computer Vision (WACV); IEEE. 2015. pp. 33-40

[35] Bewley A, Ott L, Ramos F, Upcroft B. Alextrac: Affinity learning by exploring temporal reinforcement within association chains. In: 2016 IEEE International Conference on Robotics and Automation (ICRA); IEEE. 2016. pp. 2212-2218

[36] Reid D et al. An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control. 1979;**24**(6):843-854

[37] Fortmann T, Bar-Shalom Y, Scheffe M. Sonar tracking of multiple targets using joint probabilistic data association. IEEE Journal of Oceanic Engineering. 1983;**8**(3):173-184

[38] Kayumbi G, Mazzeo PL, Spagnolo P, Taj M, Cavallaro A. Distributed visual sensing for virtual top-view trajectory generation in football videos. In: Proceedings of the 2008 International Conference on Content-Based Image and Video Retrieval; ACM. 2008. pp. 535-542

[39] Yang M, Jia Y. Temporal dynamic appearance modeling for online multi-person tracking. Computer Vision and Image Understanding. 2016;**153**:16-28

[40] Xiang Y, Alahi A, Savarese S. Learning to track: Online multi-object tracking by decision making. In: Proceedings of the IEEE International Conference on Computer Vision. 2015. pp. 4705-4713

[41] Kalman RE. A new approach to linear filtering and prediction problems. Journal of Basic Engineering. 1960; **82**(1):35-45

[42] Leal-Taixé L, Milan A, Reid I, Roth S, Schindler K. Motchallenge 2015: Towards a benchmark for multi-target tracking. 2015; arXiv preprint arXiv: 1504.01942

[43] Yu F, Li W, Li Q, Liu Y, Shi X, Yan J. Poi: Multiple object tracking with high performance detection and appearance feature. In: European Conference on Computer Vision; Springer. 2016. pp. 36-42

[44] Keuper M, Tang S, Zhongjie Y, Andres B, Brox T, Schiele B. A multi-cut formulation for joint segmentation and tracking of multiple objects. arXiv preprint arXiv:1607.06317; 2016

[45] Lee B, Erdenee E, Jin S, Nam MY, Jung YG, Rhee PK. Multi-class multi-object tracking using changing point detection. In: European Conference on Computer Vision; Springer. 2016. pp. 68-83

[46] Choi W. Near-online multi-target tracking with aggregated local flow descriptor. In: Proceedings of the IEEE International Conference on Computer Vision. 2015. pp. 3029-3037

[47] Sanchez-Matilla R, Poiesi F, Cavallaro A. Online multi-target tracking with strong and weak detections. In: European Conference on Computer Vision; Springer. 2016. pp. 84-99