Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1010767, 15 pages
https://doi.org/10.1155/2022/1010767

Research Article

Scene-Specialized Multitarget Detector with an SMC-PHD Filter and a YOLO Network

Qianli Liu, Yibing Li, Qianhui Dong, and Fang Ye

College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Qianhui Dong; dongqianhui0203@126.com

Received 26 October 2021; Revised 28 February 2022; Accepted 5 March 2022; Published 28 April 2022

Academic Editor: Bai Yuan Ding

You only look once (YOLO) is one of the most efficient target detection networks. However, the performance of the YOLO network decreases significantly when the variation between the training data and the real data is large. To automatically customize the YOLO network, we suggest a novel transfer learning algorithm with the sequential Monte Carlo probability hypothesis density (SMC-PHD) filter and Gaussian mixture probability hypothesis density (GM-PHD) filter. The proposed framework can automatically customize the YOLO framework with unlabelled target sequences. The frames of the unlabelled target sequences are automatically labelled. The detection probability and clutter density of the SMC-PHD filter and GM-PHD are applied to retrain the YOLO network for occluded targets and clutter. A novel likelihood density with the confidence probability of the YOLO detector and visual context indications is implemented to choose target samples. A simple resampling strategy is proposed for SMC-PHD YOLO to address the weight degeneracy problem. Experiments with different datasets indicate that the proposed framework achieves positive outcomes relative to state-of-the-art frameworks.

1. Introduction

Learning-based detection algorithms have proven important in several subject areas, including smart surveillance systems [1], wireless sensors [2, 3], and secure transportation systems [4]. Over the past several years, convolutional neural networks (CNNs) have achieved excellent results in multiple computer vision assignments. You only look once (YOLO) is an effective visual detection method [5]. Compared with other detection networks, the YOLO network can predict class probabilities and bounding boxes in an assessment directly from the input frame. YOLO detectors, however, are taught with annotated datasets and utilized to attain the highest variability of the target. The distribution of the target captured by the camera may not be a subset of the initial learning set when these detectors are applied to a specific scene, such as in the case of a closed-circuit television (CCTV) camera. Therefore, the resulting Generic YOLO detector may not function effectively, especially for a limited amount of training data [6].

To address this problem, transfer learning with cross-domain adaptation is proposed. A specific training dataset is needed to generate a specific detector. Normally, these positive samples of the specific training dataset are manually selected from the target dataset. However, a large amount of labelled data is needed to tune the detector in each frame, and labelling is a labor-intensive task. A typical solution for reducing the collection time is to automatically provide the sample labels with the target frame. Labelled samples are iteratively collected from the unlabelled sequence and added to the training dataset [7].

We propose a novel transfer learning method with a probability hypothesis density (PHD) filter, which can automatically retrain a YOLO network for a special object. The scene-specific detector is generated with a Generic YOLO detector trained by labelled frames and sequences without labelled information. The parameters of the YOLO detector are estimated by an iterative process. After automatic and iterative training, the final specialized YOLO detector is produced and can run without the SMC-PHD filter. Figure 1 illustrates the structure of our method.

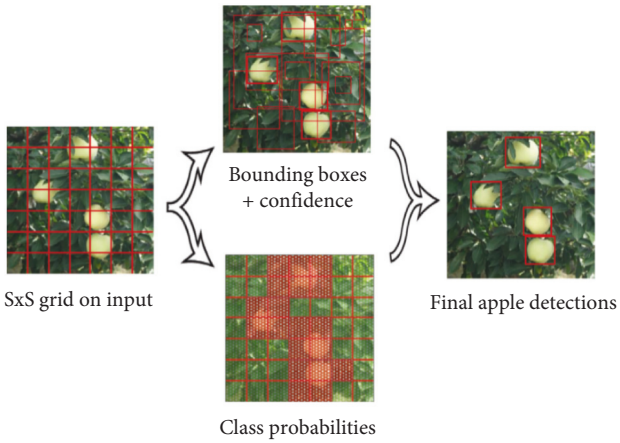


Figure 1: YOLO network. The red line shows the grids of images, and the red box shows the bounding boxes. The pattern-filled boxes show the grids with high probabilities.

Although improving the YOLO with the SMC method has been employed for transfer learning [8], the detection probability and clutter density are not considered in the target sequence. In the updated step of our proposed method, the occluded targets are selected and collected as positive samples for training. The primary benefit of our method is that the recognition model can learn the appearance of occluded targets and clutter. As shown in the experimental results in Section 4, our proposed SMC-PHD YOLO can detect some occluded speakers with the SMC-PHD filter-based occlusion strategy, while the SMC Faster region-based CNN (R-CNN) [8] cannot detect the occluded targets. In addition, when positive samples are collected, some false samples (clutter) may be added to the positive training dataset. The performance of the SMC Faster R-CNN [8] would be affected by the clutter. When there is clutter in the training dataset, the SMC Faster R-CNN produces false detection. Based on the clutter density, this clutter would be assigned a low weight, and our proposed method could disregard false samples. Our proposed PHD YOLO network has four main contributions:

(i) To address the bias between the training dataset and target set, we propose a PHD based transfer learning method for YOLO. For nonlinear tasks, a scene-specialized multitarget detector, SMC-PHD YOLO, is proposed. For linear systems and Gaussian noise tasks, we extend our method to GM-PHD YOLO to eliminate concerns about SMC dependence.

(ii) In SMC-PHD YOLO, we show that the detection probability and clutter density of the SMC-PHD filter improve the performance of the retrained YOLO networks for the occluded targets and multiscale targets. When the image quality of the target scenes is unsatisfactory, even with noise, the specialized YOLO network can still detect the target with the posterior density.

(iii) A novel likelihood is proposed to verify the selected samples in PHD YOLO. To collect positive samples for training, the confidence probability of the YOLO detector and visual context indications are applied.

(iv) For the weight degeneracy problem of SMC YOLO, we also propose a novel and simple resampling strategy that can collect samples from the target sequence based on their weights, and the proposed distribution is assumed to be the target distribution. With the detection distribution, the strategy can function effectively even when a small number of samples is employed.

The remainder of this document is structured as follows: Section 2 introduces the current approach applied in this sector and offers details regarding the benefits of our proposed method over other specialization methods. Section 3 describes our proposed strategy in detail. Section 4 details the configuration of the simulation and presents experimental outcomes, and concluding comments are provided in Section 5. We adhere to the convention that scale variables, such as confidence, are presented in lowercase italics, e.g., f. Symbols for vector-formed states and their densities are shown in lowercase bold italics, e.g., $\mathbf{x}$, and multitarget states are represented by uppercase bold italics, e.g., $\mathbf{X}$. Uppercase nonbold letters represent polynomials. Symbols for matrices, such as the transition matrix, are shown in uppercase bold letters, e.g., $\mathbf{F}$.

2. Background

2.1. Specialization Frameworks. If the distribution of the training samples is different from that of target scenes, then a traditional visual detector may not function effectively [9]. To address this problem, specialization frameworks are utilized to automatically create scene-specific detectors for a target scene. Transfer learning algorithms based on state-of-the-art theories use the annotated model and expertise gained through prior assignments. There are three main types of transfer learning methods [10]. First, by changing the parameters of the source learning model, the model is improved in a target domain [11, 12]. Second, the variation between the source and target distributions is decreased, and the source learning model is adapted to the target domain [13, 14]. Third, the training samples are manually or automatically chosen, and the model is retrained with a subset of selected samples [15]. We focus on the third category because it can automatically label the selected samples and the training parameters remain unchanged.

However, the new training dataset may contain some incorrectly labelled samples because the labels of the samples are not manually verified. With this type of dataset, the accuracy of the detection framework may decrease. To address this problem, various contextual indications, such as the visual appearance of objects, pedestrian movement, road model, size, and place, are used to verify favourable samples for retraining the training dataset; however, this method is sensitive to occlusion [16]. Moreover, some techniques may only use samples from the target domain and waste helpful samples [17]. Htike and Hogg employed a background subtraction algorithm to train a particular detector [9] to

select the target samples from the source and target datasets. To automatically label target information, tracklet chains are utilized to link the proposed samples to tracklets [15] predicted by an appearance-target detector. However, for each target scene, this framework, which includes many manual parameters and thresholds, may affect the specialization performance. Alternatively, Maâmatou et al. [10] collected fresh samples. To train a fresh dedicated retrained sensor, an SMC transfer learning method was employed to create a new dataset [8].

2.2. YOLO Network. In this work, we used the YOLO (V3) network [5] since it passes the image only once into a fully CNN (FCNN), which enables it to achieve real-time performance. YOLO (V3) was developed based on YOLO [18] and YOLO (V2) [19]. The YOLO network considers the detection problem as a regression problem. Therefore, the network directly generates a bounding box for each class via regression without any proposal region, which decreases the computational cost compared to Faster R-CNN.

The YOLO detection model is shown in Figure 1, where the network divides each input image of the training set into $S \times S$ grids. When the grid is filled by the centre of the target ground truth, the grid is used to detect the object. For each grid, several bounding boxes and their confidence scores are predicted. The confidence f_s is defined as

$$f_s = p_r \times \mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}}, \quad p_r \in \{0, 1\}. \tag{1}$$

If the target is in the grid, $p_r = 1$; otherwise, $p_r = 0$. $\mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}}$ (intersection over the union of the prediction and ground truth) is used to present the coincidence between the predicted bounding box and the reference bounding box, which indicates whether the grid contains targets. If several bounding boxes detect the same target, then nonmaximum suppression (NMS) is applied to select the best bounding box.

YOLO has a lower computational cost than Faster R-CNN; however, it has more errors. To address this problem, YOLO uses the "anchor" of the Faster R-CNN to generate suitable prior bounding boxes; YOLO uses k-means cluttering. The adoption of the anchor boxes decreases the mean average precision (mAP). In addition, unlike YOLO, YOLO-V3 uses batch normalization, multiscale prediction, a high-resolution classifier, dimension clutter, direct location prediction, fine-grained features, multiscale training, and other methods that greatly improve the detection accuracy.

2.3. Random Finite Set and PHD Filters. In this subsection, we discuss the random finite set and PHD filters for scene-specialized transform learning. The probability hypothesis density and random finite set are proposed for multitarget tracking [20–22]. The random finite set is a flexible algorithm that can be combined with any object detector to generate positional and dimensional information on objects of interest. Maggio et al. used detectors such as background subtraction, AdaBoost classifiers, and a statistical change detector to track objects associated with a random finite set (RFS) [23, 24]. For handling occlusion problems during tracking, Kim et al. proposed the labelled RFS [25]. As the RFS is a computationally expensive approximation of the multidistribution Bayes filter, the PHD is the first-order moment of the RFS, which is a set of random variables (or vectors) with random cardinality [20]. An alternative derivation of the PHD filter based on classical point process theory was given in [26]. In multitarget research, the Gaussian mixture PHD (GM-PHD) filter [27] and SMC-PHD filter [28] are widely utilized. The GM-PHD filter is a closed-form solution, as it assumes that the model is linear and Gaussian. By limiting the number of considered partitions and possible alternatives, Granstrom et al. proposed a GM-PHD filter for tracking extended targets [29]. Since different objects have different levels of clutter, an N-type GM-PHD filter was proposed for real video sequences by integrating object detector information into this filter for two scenarios [30]. However, the accuracy may decrease for nonlinear problems. To address nonlinear problems, the SMC-PHD filter was proposed based on the Monte Carlo method. With the weights of the samples (particles), the SMC-PHD filter can track a varying number of unknown targets.

The PHD filter is defined as the intensity ψ_k, which is applied to estimate the number of speakers. The PHD filter involves a prediction step and an update step that recursively propagates the intensity function. The PHD prediction step is defined as

$$\begin{aligned}\psi_{k|k-1}(\mathbf{x}_k) &= \xi_k(\mathbf{x}_k) \\ &\quad + \int \phi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})\psi_{k-1}(\mathbf{x}_{k-1})\mathrm{d}\mathbf{x}_{k-1},\end{aligned} \tag{2}$$

where $\mathbf{x}$ is the target bounding box state. $\xi_k(\mathbf{x}_k)$ is the intensity of the birth RFS. $\phi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the analogue of the state transition probability,

$$\begin{aligned}\phi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) &= p_{S,k}(\mathbf{x}_{k-1}) f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) \\ &\quad + \beta_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}),\end{aligned} \tag{3}$$

where $p_{S,k}(\mathbf{x}_{k-1})$ is the survival probability and $f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the transition density. $\beta_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the intensity function of the spawn RFS with the previous state $\mathbf{x}_{k-1}$. The PHD update equation is given as

$$\begin{aligned}\psi_k(\mathbf{x}_k) &= \left[1 - p_{D,k}(\mathbf{x}_k)\right]\psi_{k|k-1}(\mathbf{x}_k) \\ &\quad + \sum_{\mathbf{z}_k \in \mathbf{Z}_k} \frac{p_{D,k}(\mathbf{x}_k) h_k(\mathbf{z}_k \mid \mathbf{x}_k)\psi_{k|k-1}(\mathbf{x}_k)}{\kappa_k(\mathbf{z}_k) + \int p_{D,k}(\mathbf{x}_k) h_k(\mathbf{z}_k|\mathbf{x}_k)\psi_{k|k-1}(\mathbf{x}_k)},\end{aligned} \tag{4}$$

where $h_k(\mathbf{z}_k|\mathbf{x}_k)$ is the likelihood defining the probability of $\mathbf{z}_k$ given $\mathbf{x}_k$. $p_{D,k}(\mathbf{x}_k)$ is the detection probability. The intensity of the clutter RFS $\mathbf{C}_k$ is shown as $\kappa_k(\mathbf{z}_k) = \gamma u(\mathbf{z}_k)$, where γ is the average number of Poisson clutter points per scan and $u(\mathbf{z}_k)$ is the probability distribution of each clutter point. The PHD recursion involves multiple integrals in equations (2) and (4), which have no closed-form solution in general. To address this issue, the SMC-PHD filter has been proposed and widely utilized [28]. In the SMC-PHD filter, at

time $k-1$, the target PHD $\psi_{k-1}(\mathbf{x}_{k-1})$ is represented by a set of particles, $\{\mathbf{x}_{k-1}^i, \omega_{k-1}^i\}_{i=1}^{n_{k-1}}$, where n_{k-1} is the number of particles at $k-1$. To the best of our limited knowledge, this article is the first study to use the PHD filter to train a scene-specialized, multitarget detector. As the number of targets is unknown in our unlabelled dataset and the sample collection is nonlinear and non-Gaussian, the SMC-PHD filter is applied to collect the unlabelled training data and customize the YOLO network.

3. Proposed Framework

This section introduces our proposed framework, which customizes the YOLO model based on the PHD filter. The PHD filter is used to label the target in unlabelled videos based on the YOLO output. The positive samples estimated by the PHD filter are used to build a new custom dataset. The YOLO network is fine-tuned on this custom dataset, which may contain occluded targets and targets of different styles. Since the number of unlabelled videos is large, the bias between the training dataset and the real data decreases. Compared to the state-of-the-art method, our proposed framework is not sensitive to occlusion and target shape. The overall framework of the proposed method is shown in Figure 2.

To be more specific, assume that a Generic YOLO network $\mathbf{Y}_0$ is trained with generic datasets, such as Common Objects in Context (COCO) [31]. For the target sequence, unlabelled frames are represented as $\{\mathbf{I}_k\}_{k=1}^{n_I}$, where k is the index of the frame. The detection output of $\mathbf{Y}^0$ at frame k is $\{\mathbf{Z}_k\}_{k=1}^{n_I}$. $\mathbf{Z}_k = \{\mathbf{z}_k^r\}_{r=1}^{m_k}$ is a detection set at frame k, where $\mathbf{z}_k^r$ is a bounding box state of the detected target. r is the index of the detected target, and m_k is the number of detected targets. Furthermore, the PHD filter updates $\{\mathbf{Z}_k\}_{k=1}^{n_I}$ to the estimated target state $\{\mathbf{X}_k\}_{k=1}^{n_I}$. $\mathbf{X}_k = \{\mathbf{x}_k^j\}_{j=1}^{S_k}$ is an estimated target set, where b_k is the number of estimated targets at k and j is the index of the estimated targets. Note that n_I is not equal to S_K. The PHD filter removes some clutter from $\{\mathbf{Z}_k\}_{k=1}^{n_I}$ and adds some missed targets. The n_I images with an estimated target bounding box set $\{\mathbf{X}_k\}_{k=1}^{n_I}$ are applied to fine-tune the YOLO network. The fine-tuned YOLO is referred to as $\mathbf{Y}_t$, where t is the time of fine-tuning. The training pipeline of the PHD YOLO detector can be found in Figure 3.

The challenge is how to select the samples with the SMC-PHD filter. In this section, the iterative process is divided into three steps: prediction, updating, and resampling. In the following subsections, the details of the three primary steps are outlined. Since the SMC-PHD filter is more robust than the GM-PHD filter in the tracking task, PHD YOLO is mainly implemented as an SMC-PHD filter. To extend our proposed method to linear systems, GM-PHD YOLO is briefly discussed at the end of this section.

3.1. Prediction Step. To build the custom dataset, $\{\mathbf{X}_k\}_{k-1}^{n_I}$, several particles are applied. At frame $k-1$, particles are represented as $\mathbf{x}_{k-1}^i, \omega_{k-1}^i$, where ω_{k-1}^i is the particle weight. Our work considers only two kinds of particles: survival particles and birth particles. The spawn particles of the SMC-PHD filter are disregarded. For n_{k-1} survival particles, the particle state is calculated by the transition function F:

$$\mathbf{x}_{k|k-1}^i = F\mathbf{x}_{k-1}^i. \tag{5}$$

For b_k birth particles, the particle state is normally set in the tracking area. The particle weight is calculated by

$$\omega_{k\mid k-1}^i = \begin{cases} \dfrac{\phi_{k|k-1}\left(\mathbf{x}_k^i, \mathbf{x}_{k-1}^i\right)\omega_{k-1}^i}{q_k\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{Z}_k\right)}, & i = 1, \ldots, n_{k-1}, \\ \dfrac{\xi_k\left(\mathbf{x}_k^i\right)}{b_k p_k\left(\mathbf{x}_k^i|\mathbf{Z}_k\right)}, & i = n_{k-1}+1, \ldots, n_{k-1}+b_k. \end{cases} \tag{6}$$

However, if the new birth particle is located near the survival particles, then one target is repeatedly estimated by survival particles and birth particles. Thus, the number of targets would exceed the ground truth. To address this problem, we propose a novel birth density function based on the target state history:

$$\xi_k\left(\mathbf{x}_k^i\right) = \max\left(p_b, p_s \max_{\omega_k^j \in \boldsymbol{\Omega}_k}\left(\delta_{\mathbf{x}_k^i}\left(\mathbf{x}_k^j\right) 1_{\mathbf{X}_{k-1}}\left(\mathbf{x}_k^j\right)\omega_k^j\right)\right), \tag{7}$$

where

$$\delta_{\mathbf{x}_k^i}\left(\mathbf{x}_k^j\right) = \begin{cases} 1, \text{if } \mathbf{x}_k^i = \mathbf{x}_k^j, \\ 0, \text{otherwise}, \end{cases} 1_{\mathbf{X}_{k-1}}\left(\mathbf{x}_k^j\right) = \begin{cases} 1, \text{if } \mathbf{x}_k^j \subset \mathbf{X}_{k-1}, \\ 0, \text{otherwise}, \end{cases} \tag{8}$$

where p_s is the survival probability and p_b is the birth probability. p_s represents the probability that the sample $\mathbf{x}_k^i$ still exists. When $p_s = 1$, a sample still exists in the new dataset. When $p_s = 0$, samples are resampled, and samples in different iterations are independent.

3.2. Update Step. In the update step, the particle states are further updated according to the output of YOLO, $\{\mathbf{Z}_k\}_{k=1}^{n_I}$. The update step of the PHD recursion is approximated by updating the weight of the predicted particles when the likelihood $h_k(\mathbf{z}_k|\mathbf{x}_k^i)$ is obtained. The predicted weights are updated as

$$\omega_k^i = \left[\left[1 - p_D\left(\mathbf{x}_k^i\right)\right] + \sum_{\mathbf{z}_k \in \mathbf{Z}_k} \frac{p_D\left(\mathbf{x}_k^i\right) h_k\left(\mathbf{z}_k|\mathbf{x}_k^i\right)}{\kappa_k\left(\mathbf{z}_k\right) + C_k\left(\mathbf{z}_k\right)}\right]\omega_{k|k-1}^i, \tag{9}$$

where

$$C_k\left(\mathbf{z}_k\right) = \sum_{i=1}^{n_{k-1}+b_k} p_{D,k}\left(\mathbf{x}_k^i\right) h_k\left(\mathbf{z}_k|\mathbf{x}_k^i\right)\omega_{k|k-1}^i. \tag{10}$$

The detection probability $p_D(\mathbf{x}_k^i)$ is simplified as $p_{D,k}^i$ in our following work. The number of targets is estimated as the sum of the weights, $S_k = \sum_{i=1}^{n_k} \omega_k^i$.

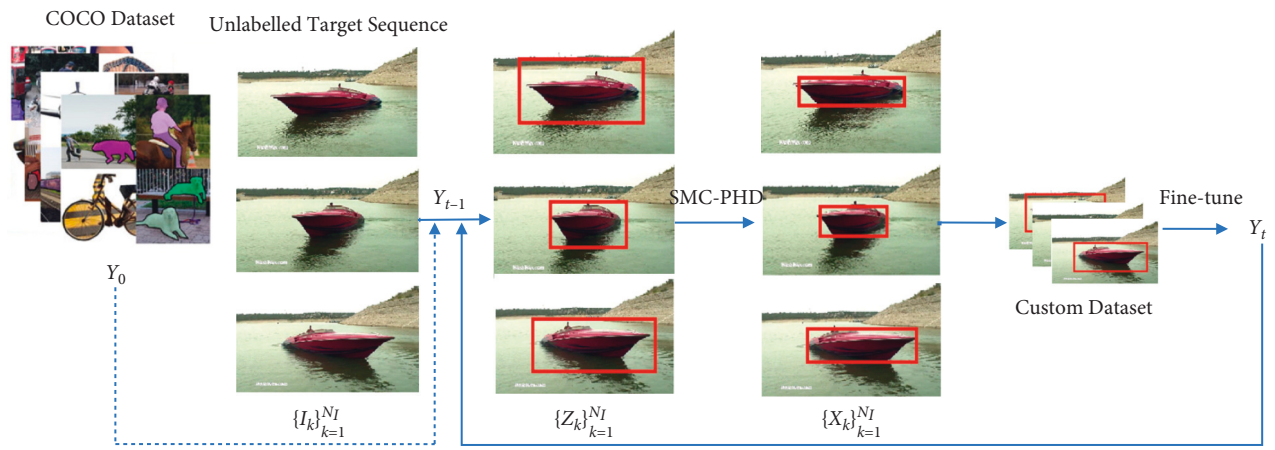


FIGURE 2: Overall framework of the proposed method. The framework input is a generic, fine-tuned YOLO detector. A visual sequence is provided to the scheme without manual labelling. To customize the YOLO network, an iterative method automatically estimates both parameters.

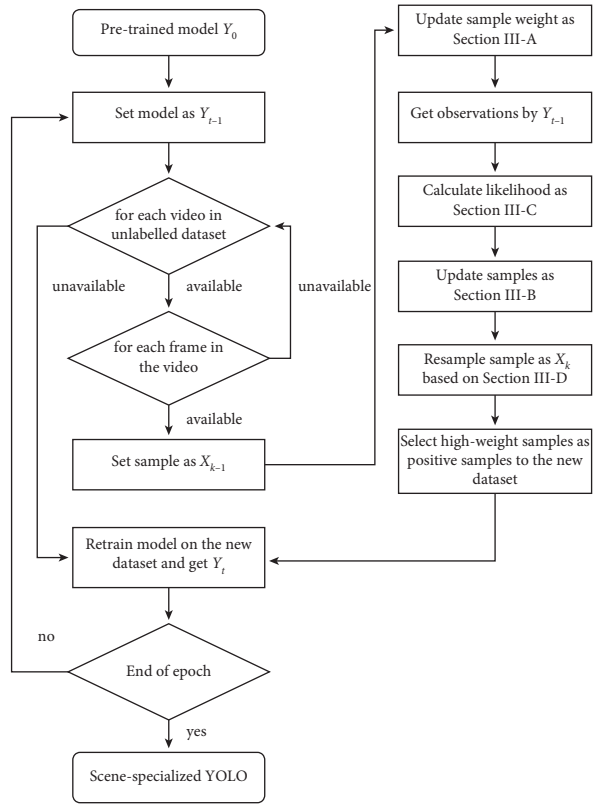


FIGURE 3: The training pipeline of the PHD YOLO detector.

To ignore the clutter, the clutter density function $\kappa_k(.)$ is applied, and the value of $\kappa_k(\mathbf{z}_k)$ is varied for the different detections $\mathbf{z}_k$. $\kappa_k(\mathbf{z}_k)$ indicates the level of clutter and is a set value. When z_k^r has a high probability of being cluttered, $\kappa_k(\mathbf{z}_k)$ is a high value. If the detection is not cluttered, then $\kappa_k(\mathbf{z}_k)$ is given as 0. Normally, $\kappa_k(\mathbf{z}_k)$ is set as a constant or estimated by the Beta-Gaussian mixture model [32].

$p_D(\mathbf{x}_k^i)$ is the detection probability, which is chosen based on the sample and can be estimated by the Gaussian mixture model [32]. If the sample is occluded, then $p_D(\mathbf{x}_k^i)$ would have a low value (near 0). Therefore, the occluded samples have high weights and are selected for retraining the YOLO network. If the sample is not occluded, then $p_D(\mathbf{x}_k^i)$ is equal to 1, and the value $h_k^{i,r}$ is not changed.

3.3. Likelihood Function. In addition to the detected probability and clutter density, the likelihood density determines whether the sample is selected for retraining. Samples with high weights are employed to retrain the YOLO network, while samples with low weights are disregarded. The likelihood density is applied to represent the relationship between the detections of the YOLO network and the samples. Therefore, we define the likelihood as

$$h_k = f_s \max(f_x, \beta_k), \tag{11}$$

where

$$\beta_k = \frac{\beta_0}{k}. \tag{12}$$

During the iterative process, β_k is decreased. When the selected sample applied to retrain the YOLO detector has a high associated score, the sample likelihood is maximized. The confidence scores f_s are provided by the YOLO network output layer. When $f_s = 0$, the weight of the sample is set to 0, and the sample is removed from the specialized dataset. f_x indicates whether the sample was detected by the YOLO network. For visual cues, we calculate the Euclidean distance between the selected sample $\mathbf{x}_k^i$ and the previous sample $\mathbf{X}_{k-1}$.

$$f_x = e^{\sum_{x_k^i \in \mathrm{x}_k} D_k^{i,r}} \alpha_k^i, \tag{13}$$

where

$$D_k^{i,r} = \left(u_k^r - u_k^i\right)^2 + \left(v_k^r - v_k^i\right)^2 + \left(w_k^r - w_k^i\right)^2 + \left(h_k^r - h_k^i\right)^2, \tag{14}$$

where $[u_k^r, v_k^r, w_k^r, h_k^r]$ is the state of the detection z_k^r. To select high-score samples $\mathbf{x}_k^i$, we use a dynamic threshold:

$$\alpha_k^i = \begin{cases} \max\limits_{\mathbf{x}^j \in \mathbf{X}_{t-1}} \delta_{\mathbf{x}_k^i}\left(\mathbf{x}^j\right) 1_{\mathbf{X}_{k-1}}\left(\mathbf{x}^j\right) \delta_{y^j}\left(y_k^i\right) s^j, & \text{if } k \neq 0, \\ \alpha_0, & \text{if } k = 0, \end{cases} \tag{15}$$

where y^j and y_k^i are the target class label s calculated by $\mathbf{Y}_{t-1}$. s_k^j is the associated score, and α_0 is the initial threshold.

3.4. Resampling Step. The SMC-PHD filter is utilized to construct a new, specific dataset for retraining, according to the resampling approach, in which resamples from the weighted dataset are included in the generated dataset $\{\mathbf{x}_k^i\}_{i=1}^{n_k}$. However, the traditional SMC-PHD meets the weight degeneracy problem and the number of samples decreases during the retraining step. To generate a new, unweighted dataset with the same number of samples as the weighted dataset, a sampling strategy is employed. Moreover, the effective sample size (ESS) of $\{\mathbf{x}_k^i, \omega_k^i\}_{i=1}^{n_k}$ is calculated:

$$\mathrm{ESS} = \frac{\left(\sum_{i=1}^{n_k} \omega_k^i\right)^2}{\sum_{i=1}^{n_k} \left(\omega_k^i\right)^2}. \tag{16}$$

When the ESS is greater than 0.5, the particles can be considered to be positive samples for the special training dataset. When the ESS is less than 0.5, the particles should be resampled via the Kullback–Leibler distance (KLD) sampling [33]:

$$\left\{\mathbf{x}_k^i\right\}_{i=1}^{n_k} \leftarrow \left\{\mathbf{x}_k^i, \boldsymbol{\omega}_k^i\right\}_{i=1}^{n_K}. \tag{17}$$

An extra k-means method is used to estimate $\mathbf{X}_k$ based on the particles $\mathbf{x}_{kk=1}^{i\,n_k}$. Note that the aspect ratio of the positive training sample may differ from the initial anchors $\mathbf{A}_{t-1}$, as we use the IoU overlap as the positive sample. We employ the k-means method to cluster the aspect ratio of samples to update the anchors. To decrease the computational cost, only three anchors are used to retrain the YOLO network; they are set to $\mathbf{A}_t$. These proposals are employed to retrain the YOLO network, which is produced by fine-tuning the specific dataset. In the next iteration, these networks will become the input of the forecast phase and be used to create target proposals (bounding boxes) in the target scene.

3.5. GM-PHD YOLO. SMC-PHD is mainly discussed and applied to improve the YOLO network since it is more robust than the GM-PHD filter for nonlinear systems. However, for linear systems, the GM-PHD filter can provide a higher accuracy rate than the SMC-PHD filter. Therefore, in this subsection, we briefly discuss how to use the GM-PHD filter to improve the YOLO network. The pipeline of the GM-PHD YOLO is similar to that of SMC-PHD YOLO. YOLO is pretrained on the generic dataset, and GM-PHD assists in building the custom dataset from the unlabelled target sequences. YOLO is fine-tuned on this custom dataset. When the GM-PHD filter selects the samples, the steps include the prediction step, update step, and pruning.

In the GM-PHD filter, $\mathbf{x}_{k-1}^i$ is distributed across the state space based on Gaussian density $N(\mathbf{m}_{k-1}^i, \mathbf{P}_{k-1}^i)$, where $\mathbf{m}_{k-1}^i$ and $\mathbf{P}_{k-1}^i$ are the mean and variance, respectively. In the prediction step, for existing targets, N ($\mathbf{m}_{k-1}^i$ and $\mathbf{P}_{k-1}^i$) are predicted as $\mathbf{m}_{k|k-1}^i = F\mathbf{m}_{k-1}^i$ and $\mathbf{P}_{k|k-1}^i = \mathbf{Q} + \mathbf{F}\mathbf{P}_{k-1}^i F^T$, respectively, where $\mathbf{Q}$ is the transition noise variance. Their weight is calculated as $\omega_{k|k-1}^i = p_s \omega_k^i$. Birth targets are randomly chosen in the tracking area. In the update step, for undetected targets, the mean and variance retain their values, and their weights are calculated as $\omega_k^i = (1 - p_D)\omega_{k|k-1}^i$. For detected targets, the mean is calculated as

$$\begin{aligned} \mathbf{m}_k^i = {} & \mathbf{m}_{k|k-1}^i \\ & + \mathbf{P}_{k|k-1}\mathbf{H}^T \left[\mathbf{R} + \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T\right]^{-1} \left(\mathbf{z}_k - \mathbf{H}\mathbf{m}_{k|k-1}^i\right). \end{aligned} \tag{18}$$

The variance is updated as

$$\mathbf{P}_k^i = \left[\mathbf{I} - \mathbf{P}_{k|k-1}\mathbf{H}^T \left[\mathbf{R} + \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T\right]^{-1} \mathbf{H}\right] \mathbf{P}_{k|k-1}^i. \tag{19}$$

The particle weight is updated as

$$\omega_k^i = \mathbf{p}_D \omega_{k|k-1}^i N\left(\mathbf{z}_k; \mathbf{H}\mathbf{m}_{k|k-1}^i, \mathbf{R} + \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T\right). \tag{20}$$

The weight is normalized as

$$\omega_k^i = \frac{\omega_k^i}{\kappa_k(\mathbf{z}_k) + \sum_{i=1}^{n_k} \omega_k^i}. \tag{21}$$

A simple pruning procedure is further employed to reduce the number of Gaussian components. The high weight targets are set to $\mathbf{X}_k$ and are utilized to build the custom dataset.

4. Experimental Results

This section introduces the test results obtained on several public and private datasets. First, the implementation details of our proposed method are given. Second, the dataset and baseline algorithms are introduced. Third, the ablation study of the SMC-PHD YOLO filter is discussed. Our proposed SMC-PHD YOLO detector and several baseline methods are compared.

4.1. Implementation Details. The initialized YOLO in our proposed SMC-PHD YOLO filter is pretrained on the COCO dataset [31]. The Adam optimizer is applied, where the weight decline is 0.0005 and the momentum is 0.9. Although the transition matrix F differs substantially across the different object classes in the different datasets, to simplify the problem, we assume F to be

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{22}$$

The YOLO network is fine-tuned on our evaluation dataset for the different tasks with the help of the SMC-PHD-based transforming method. The YOLO detector is tuned with a 64 GB NVIDIA GeForce GTX TITAN *X* GPU.

4.2. Evaluation Methodology and Dataset. We train the YOLO detector on a training collection containing 80k training frames and 500k example annotations from the COCO dataset, which contains 2.5 million labelled instances among 328k images of only 91 objects. Although the COCO dataset does not contain continuous frames, it is only used to pretrain the YOLO network before the experiments. In the evaluation step, datasets should contain continuous frames. The evaluation was performed with three different datasets.

GOT-10k [34] is a large-scale, visual dataset with broad coverage of real-world objects. It contains 10k videos of 563 categories, and its categories are more than one order of magnitude wider than those of counterparts of a similar scale. Some of its categories are not included in the COCO dataset. Therefore, GOT-10k is suitable for fine-tuning the YOLO network pretrained on the COCO dataset. The annotations that we tested include birds, cars, tapirs, and cows. YouTubeBB [35] is a large, diverse dataset with 380,000 video sections and 5.6 million human-drawn bounding boxes in 23 classifications from 240,000 distinct YouTube videos. Each video includes time-localized, frame-level features, so classifier predictions at segment-level granularity are feasible. The annotations that we tested include cars and zebras. In the MIT Traffic dataset [36], a 90-minute video is provided. A total of 420 frames from the first 45 minutes are employed for specialization, and 420 images from the last 45 minutes are utilized for testing. The video was recorded by a stationary camera. The size of the scene is 720 by 480, and it is divided into 20 clips. The annotation that we tested includes only the cars. False-positive curves per frame (FPPI) and receiver operating characteristic (ROC) curves are used to evaluate our proposed detector and baseline methods. The pipeline of the data preparation for the PHD YOLO experiment is shown in Figure 4.

4.3. Baseline Method. The algorithms compared with the SMC-PHD YOLO algorithm are Generic YOLO [5], Generic Faster R-CNN [37], SMC Faster R-CNN [8], that of Singh et al. [38], that of Deshmukh and Moh [39], that of Kang et al. [40], that of Maâmatou et al. [10], spatiotemporal sampling network (STSN) [41], salient object detection (SOD) [42], that of Lee et al. [43], that of Jie et al. [44], and that of Ghahremani et al. [45]. Table 1 shows the comparison between baseline methods and our method. The detector pretrained on the general dataset is presented in the second column. Some methods automatically fine-tune the network with the target dataset collected by the methods shown in the third column. For example, the algorithm of Kang et al. [40] does not include a fine-tuning step, and there is no information in its block. The computational complexity of fine-tuning with the target dataset is shown in the last column, where n_I is the number of frames in the video, n is the number of particles for the SMC method, m is the average number of targets in each frame, $l * h$ is the size (length-$*$width) of the frame, and a is the number of auxiliary networks.

4.4. SMC-PHD Filter YOLO for Multitarget Detection. In this subsection, we discuss the contribution of the SMC-PHD in our proposed method via three experiments. In these three experiments, we evaluate the performance of the detection probability and clutter density. Note that for a fixed label dataset and fixed YOLO, these parameters are also fixed and can be measured from the dataset. To show the contribution of the detection probability and clutter density, we set different values in the experiments.

4.4.1. Detection Probability. To evaluate the detection probability performance, we set the detection probability as different constants. The detection probability in the SMC-PHD is incrementally increased from 0 to 1, and six situations are considered: 0, 0.2, 0.4, 0.6, 0.8, and 1. The YouTubeBB dataset is selected since it includes several situations. For example, the vehicles in traffic videos are frequently occluded by other vehicles, while airplanes at an airport always appear in the scene.

Table 2 shows the FPPI of the SMC-PHD YOLO network versus the detection probability and category. A correctly

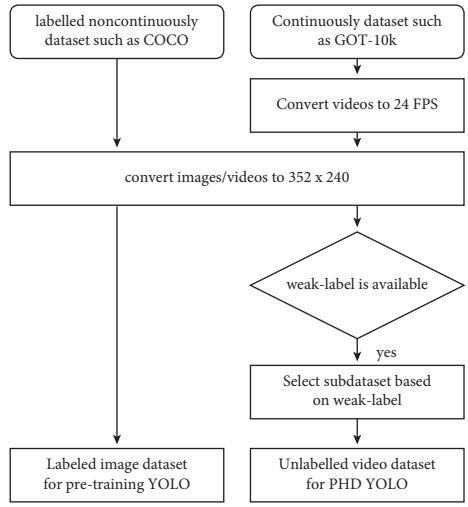


Figure 4: The pipeline of the data preparation for the PHD YOLO experiment.

Table 1: Comparison between baseline methods and our method.

Baseline	Detector	Fine-tuned	Computational complexity
YOLO [5]	YOLO	—	—
R-CNN [37]	R-CNN	—	—
SMC R-CNN [8]	R-CNN	SMC	$\mathcal{O}(n_I * n * m)$
Singh et al. [38]	R-CNN	Track and segment [46]	$\mathcal{O}(n_I * l * h)$
Deshmukh and Moh [39]	CNN	Edge detectors	$\mathcal{O}(n_I * l * h)$
Kang et al. [40]	Contextual R-CNN	—	—
Maâmatou et al. [10]	SVM	SMC	$\mathcal{O}(n_I * n * m)$
STSN [41]	STSN	—	—
SOD [42]	SOD	R101 FPN	$\mathcal{O}(n_I * n * m)$
Lee et al. [43]	R-CNN	Auxiliary network	$\mathcal{O}(n_I * n * m * a)$
Jie et al. [44]	R-CNN	Online supportive sample harvesting [44]	$\mathcal{O}(n_I * n * m)$
Ghahremani et al. [45]	CNN	F1 score threshold	$\mathcal{O}(n)$
SMC-PHD YOLO	YOLO	SMC-PHD	$\mathcal{O}(n_I * n * m)$

Table 2: FPPI of the SMC-PHD YOLO network versus detection probability for the "airplane" and "car" categories of the YouTubeBB dataset.

$p_{D,k}$	0	0.2	0.4	0.6	0.8	1
Airplane	0.80	0.81	0.78	0.75	0.72	0.69
Car	0.80	0.83	0.86	0.88	0.85	0.81

estimated detection probability can produce a high FPPI. For example, since the airplanes are always shown in the centre of the scene in the airplane sequences, the lowest FPPI for the airplane category is $p_{D,k} = 0.2$. The best results for the car category are $p_{D,k} = 0.6$ due to the occluded cars. Therefore, if targets are frequently occluded, then the detection probability should be of high value. Furthermore, for the airplane category, the FPPI at $p_{D,k} = 1$ is only 85% of that at $p_{D,k} = 0.2$. Thus, if the detection probability is too high, such as 1, then the FPPI of the detection would decrease.

4.4.2. Clutter Density Function. The clutter density function is employed to address the clutter problem. For the PHD filter, the clutter density function is varied based on the detection results, and it is given a constant value in many references [26, 28, 32, 47]. In these experiments, clutter density is a constant value for all detections. However, a large $\kappa_k(\mathbf{z}_k^r)$ may decrease the weights of the targets, which causes an insufficient number of samples to be included in the training dataset. A low $\kappa_k(\mathbf{z}_k^r)$ cannot address the clutter problem, and the retrained YOLO model is still sensitive to clutter. Since $\kappa_k(\mathbf{z}_k^r)$ is normally set to a value from 0 to infinity, we test 8 different values on the boat and bicycle sequences of the YouTubeBB dataset. Distant buildings may be detected as boats, and the bicycle detection performance is also easily affected by the surroundings. The results are shown in Table 3. The highest FPPIs for the boat sequence and bicycle sequence are 0.3 and 0.1, respectively, since the level of clutter varies for different categories. For "boat," if κ_k is lower than 0.3, the FPPI would slightly decrease since clutter is added to the specialized training data and the

Table 3: FPPI of the SMC-PHD YOLO network versus detection probability for the "boat" and "bicycle" categories of the YouTubeBB dataset.

κ_k	0	0.1	0.3	0.7	0.9	1	5	10
Boat	0.81	0.82	0.84	0.82	0.74	0.68	0.58	0.51
Bicycle	0.67	0.69	0.65	0.59	0.51	0.48	0.39	0.34

retrained model is still sensitive to the clutter. If κ_k exceeds 0.3, the FPPI also decreases since the weight of the target samples decreases and the retraining dataset does not include sufficient training samples.

4.5. Error Analysis of the SMC-PHD YOLO Network. Since the target dataset is automatically generated by an SMC-PHD filter, it may include some error samples with uncorrected labels. To analyse whether the error samples affect the final performance, we test our SMC-PHD YOLO network with the YouTube dataset. The annotations that we employ comprise cars and zebras. The video length for each annotation, which contains 36000 frames, is 20 min. These frames are manually labelled by researchers and automatically labelled by our methods. After manually labelling these videos, 831,615 and 88,234 positive target samples were obtained for cars and zebras since multiple targets may appear in the same frame. For labels labelled by our methods, "cars" includes 797,660 true-positive samples and 212 false-positive samples, while "zebras" includes 69,821 true-positive samples and 17 false-positive samples. These results show that algorithms assign fewer labels than humans because some tiny targets and low-possibility targets are considered clutter to be disregarded. "Car" has a higher recall rate (96%) than "zebra" (79%) since cars with a regular profile are easier to detect. To further analyse these error samples, we print these data distributions. The selected features comprise the input of the last fully connected layer of YOLO. Two main dimensions are selected by t-distributed stochastic neighbour embedding. Figure 5 shows the data distribution of true positives, false positives, and false negatives. This finding proves that tiny targets are considered to be outliers and are disregarded. We also discovered that some clutter (green points) in the target dataset is considered positive samples (false positives). After the clutter is manually disregarded in the target dataset, the YOLO performance does not change. The main potential reason for this is the high threat score (99%), and the SMC-PHD filter disregards the most uncertain samples. However, this approach does not fundamentally solve the problem of clutter since some low-possibility positive samples are considered to be false negatives (red points). Some researchers suggest the use of extra information, such as audio information, to address the clutter problem [48]. Addressing the clutter problem will be one of our future research topics.

4.6. Scene-Specialized Multitarget Detector. To show the performance of the PHD method for transfer learning, we compare the baseline YOLO network, SMC YOLO network, SMC R-CNN, and our proposed SMC-PHD YOLO network and GM-PHD YOLO on the YouTubeBB dataset. Since SMC R-CNN cannot address occluded samples, we propose SMC-PHD R-CNN with SMC-PHD to improve the performance of Faster R-CNN and show the effect of the PHD method. We train the YOLO network with a general training set (COCO dataset), which contains a limited amount of target data. SMC-PHD then augments a dataset containing unseen data. The unseen data in augmented data are assigned labels that may contain errors. YOLO is fine-tuned on this target dataset, and YOLO is applied without an SMC-PHD filter. The SMC-PHD filter is only applied to augment data in this work. The parameters of the PHD filter are chosen according to the Beta-Gaussian mixture model [32]. We test these methods for the airplane, bicycle, boat, and car categories of the YouTubeBB dataset. For different categories, we train the different SMC-PHD YOLO networks where parameters are independent. The YOLO network and R-CNN fine-tuned by the SMC-PHD, GM-PHD, and SMC filters are shown in Table 4. After fine-tuning YOLO, filters are not employed for target detection. Our proposed method has the highest FPPI value of all methods for the boat and car categories, and SMC-PHD YOLO performs similarly to SMC-PHD R-CNN. According to the results, SMC improves the performance of YOLO and R-CNN by approximately 8%, and PHD further improves their performance by approximately 6%. Although GM-PHD YOLO has an 8% higher FPPI than YOLO, it is still lower than that of SMC-PHD YOLO. We speculate that the reason for this is that the number of bounding boxes identified by GM-PHD YOLO is 4% more than that identified by SMC-PHD YOLO. It is proven that SMC-PHD YOLO is more robust than GM-PHD YOLO. Therefore, in the following experiment, we mainly test SMC-PHD YOLO.

Some results of the proposed method and baseline methods are shown in Figure 6. The first line and second line of each subfigure are detected by Generic YOLO and specific YOLO, respectively. In Figure 6(a), the flapping bird is detected only by the specialized YOLO detectors. Thus, our proposed method can customize the detector for a moving target because the dataset is selected from a sequence with the likelihood function. In addition, some occluded cars are detected by our proposed method due to the detection probability. In Figure 6(b), cars and zebras are successfully detected by the specialized YOLO detector, even though only parts of the vehicles and zebras are shown in the images. For the traffic sequences shown in Figure 6(c), the number of cars detected with the specialized YOLO detector is higher than that detected with the Generic YOLO detector. With the SMC-PHD filter, our proposed method can detect occluded cars and certain small vehicles.

To further evaluate our proposed method, we further compare our methods with other baseline methods, such as that of Singh et al. [38], that of Deshmukh and Moh [39],

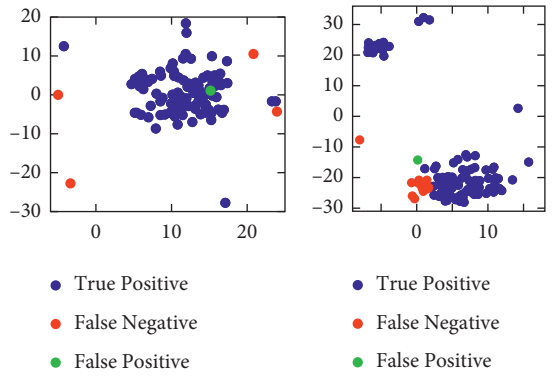


Figure 5: Data distribution of true positives, false positives, and false negatives for "car" and "zebra" of the YouTube BB dataset.

Table 4: FPPI of our proposed SMC-PHD YOLO, SMC YOLO, YOLO, SMC-PHD R-CNN, and SMC R-CNN on the YouTubeBB dataset.

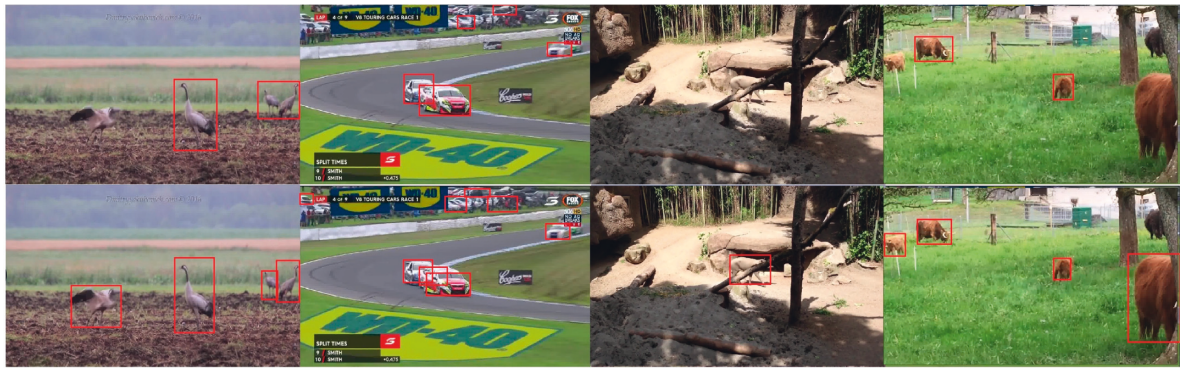
Method	Airplane	Bicycle	Boat	Car
SMC-PHD YOLO	0.81	0.69	0.84	0.88
GM-PHD YOLO	0.79	0.65	0.82	0.84
SMC YOLO	0.76	0.63	0.76	0.81
YOLO	0.71	0.57	0.68	0.76
SMC-PHD R-CNN	0.82	0.70	0.83	0.88
SMC R-CNN	0.79	0.67	0.83	0.89

that of Kang et al. [40], that of Maâmatou et al. [10], STSN [41], SOD [42], that of Lee et al. [43], that of Jie et al. [44], and that of Ghahremani et al. [45].

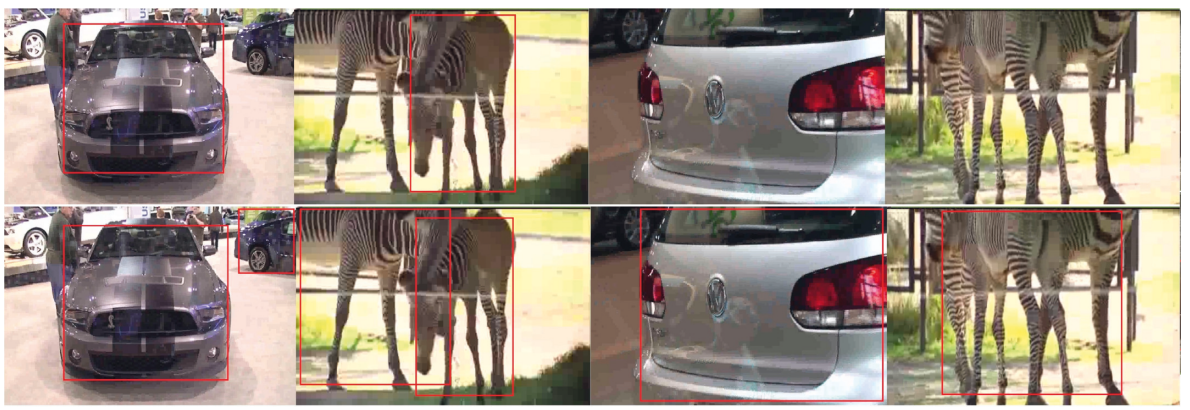
Figure 7 shows the ROC curves of the filters for the different annotations. In this experiment, we chose the bird and boat categories from the GOT-10k and YouTubeBB datasets and the car category from the MIT Traffic dataset. Due to the page limitation, Figures 7(a) and 7(b) only show a comparison between SMC-based detectors, such as SMC-PHD YOLO, and generic detectors, such as YOLO. The comparison between our proposed method and state-of-the-art methods is shown in Figures 7(c)–7(e). In Figure 7(a), the method of Kang achieves a higher true-positive rate than that of Kumar and Dalal because the former is specially designed for boat detection. Compared with the Generic YOLO for boat detection, the SMC-PHD YOLO detector achieves an ROC improvement of 13%. As the boat is often occluded in the bay, the SMC-PHD YOLO detector with the detection probability performs better than the other methods. The boat detection results on the YouTubeBB dataset are similar to those on the GOT-10k dataset. Compared with generic methods, specialized methods achieve ROC improvements of approximately 10%. More baseline transform learning methods are considered in Figure 7(c), which are shown as dashed lines. The transform methods achieve better performance than the generic R-CNN or YOLO methods. SMC based on R-CNN achieves a similar ROC value as other transform detectors. Based on SMC, the SMC R-CNN detector and SMC-PHD YOLO detector achieve increases in the ROC values of 3.8% and 5.8%, respectively, compared with their baseline methods. For car detection, we test the methods only on the MIT Traffic dataset. As shown by the ROC curves in Figure 7(e), the YOLO SMC-PHD sensor outperforms all other car detection frameworks. The SMC-PHD YOLO detector also outperforms the four other specialized detectors, i.e., SMC Faster R-CNN, that of Kumar, that of Dalal, and that of Maamatou, by 5%, 6%, 9%, and 2%, respectively.

Table 5 reports the average detection rate of our proposed method and other state-of-the-art methods for the different datasets. We list the ten annotations on GOT-10k and YouTubeBB. As the Kang and Maamatou methods are designed for boat and traffic detection, they are not included in this table. Our proposed method achieves the highest detection rate, especially for the MIT Traffic dataset. SMC-PHD YOLO can detect occluded targets, such as cars. Although SMC R-CNN achieves a detection rate similar to that of the SMC-PHD YOLO detector, the number of frames per second (FPS) of the SMC-PHD YOLO network is 100 times that of SMC R-CNN. Therefore, the SMC-PHD YOLO detector considerably outperforms the generic detector with several annotations on all government datasets. Compared to the baseline YOLO detector, the SMC-PHD YOLO detector achieves a 12% higher detection rate.

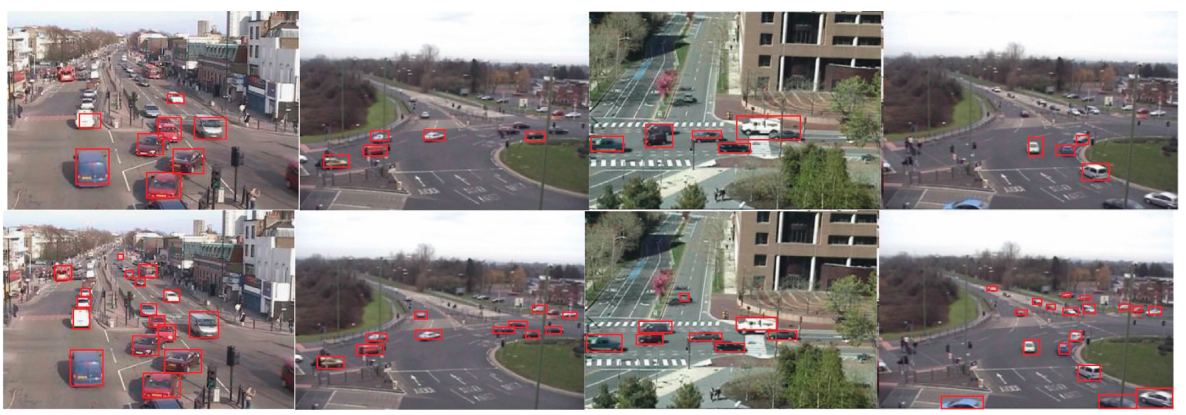
Although our proposed method has the highest detection rate and large ROC values among all methods, the proposed SMC-PHD YOLO performance depends on the hyperparameters, such as the detection probability and clutter density. These parameters should be established at the beginning of training based on previous experience. Some researchers have proposed solutions for estimating the parameters of the SMC-PHD filter. For example, Lian et al. [49] used the expectation maximum to estimate the unknown clutter probability, and Li et al. [50] used the gamma Gaussian mixture model to estimate the detection probability. Applying this kind of estimation method to improve the SMC-PHD YOLO filter will be addressed in our future work.



(a)



(b)



(c)

Figure 6: Improvement of the scene-specific detector for GOT-10k (a), YouTubeBB (b), and MIT Traffic (c). The first line of each subfigure indicates the Generic YOLO, and the second line of each subfigure indicates the SMC-PHD YOLO detector.

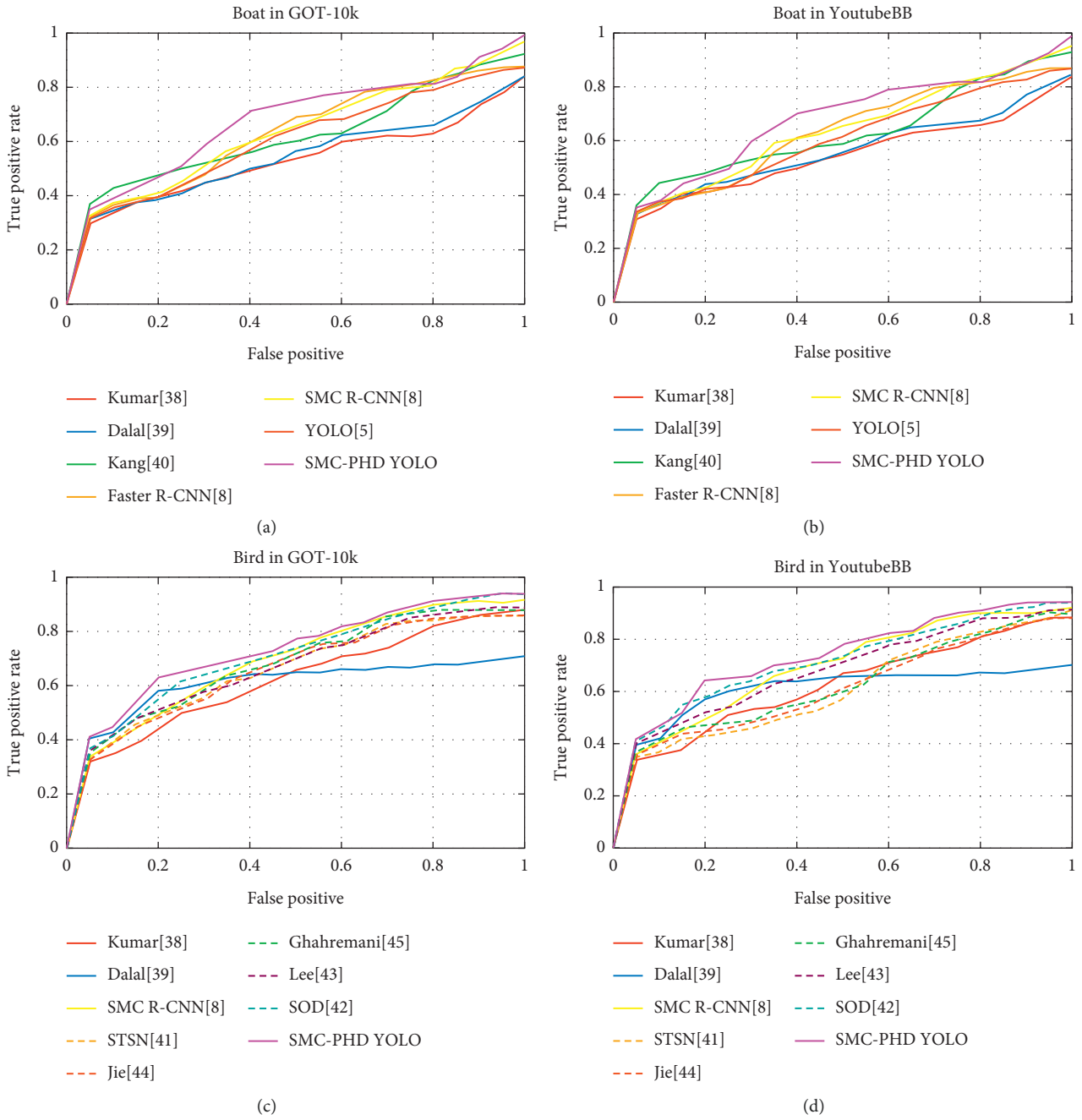
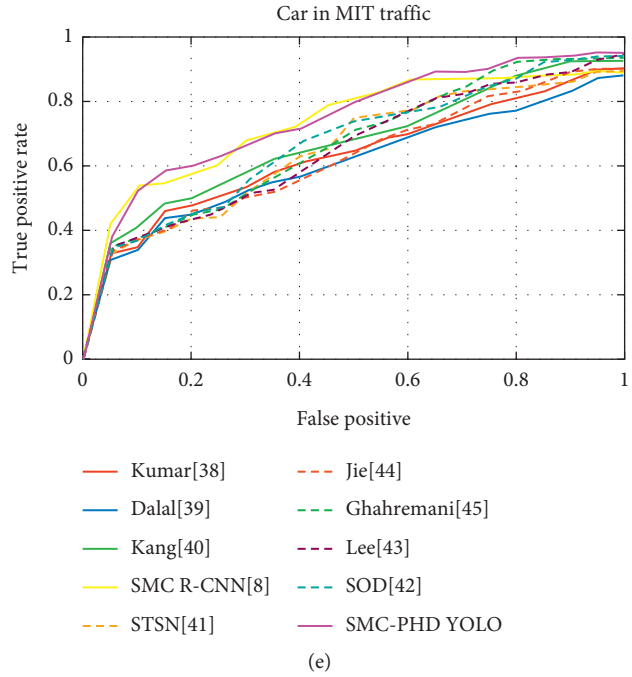


Figure 7: Continued.



(e)

Figure 7: ROC curves for the Kumar, Dalal, Faster R-CNN, SMC Faster R-CNN, YOLO, and SMC-PHD YOLO methods with the bird (a) and boat (b) annotations of GOT-10k, the bird (c) and boat (d) annotations of YouTubeBB, and the car (e) annotation of the MIT Traffic dataset.

Table 5: Detection rate for the different datasets with different detections (at 1 FPPI).

Data		SMC-PHD YOLO	YOLO [5]	SMC R-CNN [8]	Kumar [38]	Dalal [39]	STSN [41]	Jie [44]	Ghahremani [45]	Lee [43]	SOD [42]
YoutubeBB	Airplane	0.91	0.81	0.87	0.8	0.8	0.83	0.85	0.86	0.83	0.89
	Bicycle	0.89	0.77	0.86	0.78	0.75	0.82	0.83	0.82	0.84	0.87
	Bird	0.94	0.85	0.92	0.82	0.84	0.87	0.86	0.86	0.88	0.89
	Boat	0.98	0.87	0.96	0.83	0.84	0.89	0.91	0.92	0.93	0.95
	Bus	0.96	0.83	0.95	0.82	0.81	0.89	0.86	0.9	0.92	0.93
	Car	0.98	0.86	0.95	0.84	0.83	0.91	0.92	0.92	0.93	0.94
	Cat	0.94	0.85	0.92	0.82	0.83	0.93	0.91	0.93	0.92	0.95
	Cow	0.98	0.87	0.95	0.86	0.88	0.95	0.96	0.94	0.95	0.96
	Dog	0.92	0.81	0.89	0.80	0.82	0.88	0.91	0.89	0.9	0.88
	Horse	0.96	0.85	0.94	0.86	0.86	0.92	0.9	0.89	0.93	0.95
GOT-10k	Anteater	0.53	0.39	0.41	0.37	0.42	0.52	0.48	0.51	0.49	0.52
	Bird	0.94	0.88	0.92	0.87	0.79	0.86	0.86	0.88	0.89	0.93
	Cat	0.91	0.83	0.90	0.84	0.79	0.84	0.86	0.88	0.9	0.87
	Elephant	0.88	0.73	0.86	0.75	0.70	0.82	0.84	0.87	0.89	0.85
	Boat	0.98	0.87	0.97	0.84	0.84	0.87	0.89	0.92	0.94	0.97
	Goat	0.88	0.72	0.87	0.76	0.69	0.78	0.8	0.83	0.85	0.87
	Horse	0.87	0.71	0.85	0.73	0.75	0.81	0.83	0.84	0.86	0.85
	Lion	0.86	0.73	0.84	0.71	0.77	0.81	0.83	0.84	0.85	0.83
	Car	0.95	0.85	0.91	0.86	0.87	0.85	0.87	0.93	0.94	0.94
	Tank	0.74	0.61	0.68	0.63	0.61	0.63	0.66	0.69	0.71	0.73
MIT	Pedestrian	0.97	0.85	0.93	0.86	0.82	0.91	0.93	0.95	0.94	0.96
	Car	0.95	0.88	0.89	0.93	0.89	0.9	0.92	0.93	0.95	0.96
Average		0.9	0.79	0.87	0.79	0.78	0.84	0.85	0.86	0.87	0.89

5. Conclusion

To customize the YOLO detector for unique target identification, we suggested an effective and precise structure based on the SMC-PHD filter and GM-PHD filter. On the basis of the proposed confidence score-based likelihood and novel resampling strategy, the framework can be employed by choosing appropriate samples from target datasets to train and then detect a target. This framework automatically offers a strong specialized detector with a Generic YOLO detector and some target videos. The tests showed that the proposed framework can generate a specific YOLO detector that considerably outperforms the Generic YOLO detector on a distinct dataset for bird, boat, and vehicle detection. Correlated clutter is still challenging for SMC-PHD filters. Our future research will focus on expanding the algorithm with multimodal information to address the correlated clutter problem.

Data Availability

The data used to support this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 51879055) and Heilongjiang Touyan Innovation Team Program.

References

[1] A. Hampapur, L. Brown, J. Connell et al., "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 38–51, Mar. 2005.

[2] S. Javadi, H. Moosaei, and D. Ciuonzo, "Learning wireless sensor networks for source localization," *Sensors*, vol. 19, no. 3, p. 635, 2019.

[3] L. Zhao and Z. Huang, "A moving object detection method using deep learning-based wireless sensor networks," *Complexity*, vol. 2021, p. 2021.

[4] R. Thiolliere, E. Dunbar, G. Synnaeve, M. Versteegh, and E. Dupoux, "A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling," *Proc. INTERSPEECH*, pp. 3179–3183, 2015.

[5] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[6] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proceedings of the British Machine Vision Conference*, 7 September 2009.

[7] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 3457–3464, IEEE, CO, USA, 20 June 2011.

[8] A. Mhalla, T. Chateau, H. Maâmatou, S. Gazzah, N. E. Ben Amara, and R.-C. N. N. Smc faster, "SMC faster R-CNN: toward a scene-specialized multi-object detector," *Computer Vision and Image Understanding*, vol. 164, pp. 3–15, 2017.

[9] K. K. Htike and D. C. Hogg, "Efficient non-iterative domain adaptation of pedestrian detectors to video scenes," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 654–659, Springer, Stockholm, Sweden, 24 Auguust 2014.

[10] H. Maâmatou, T. Chateau, S. Gazzah, Y. Goyat, and N. E. B. Amara, "Transductive transfer learning to specialize a generic classifier towards a specific scene," in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 411–422, IEEE, Rome, Italy, 27 February 2016.

[11] Y. Aytar and A. Zisserman, "Tabula rasa: model transfer for object category detection," in *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 2252–2259, Springer, Barcelona, Spain, 6 November 2011.

[12] T. Tommasi, F. Orabona, and B. Caputo, "Learning categories from few examples with multi model knowledge transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 928–941, 2014.

[13] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[14] B. Quanz, J. Huan, and M. Mishra, "Knowledge transfer with low-quality data: a feature extraction issue," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1789–1802, 2012.

[15] Y. Mao and Z. Yin, "Training a scene-specific pedestrian detector using tracklets," in *Proceedings of the IEEE Conference on Applications of Computer Vision (WACV)*, pp. 170–176, IEEE, Waikoloa, HI, USA, 5 January 2015.

[16] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *Proceedings of the IEEE Conference on Applications of Computer Vision (WACV)*, pp. 29–36, IEEE, Breckenridge, CO, USA, 5 January 2005.

[17] K. All, D. Hasler, and F. Fleuret, "Flowboost appearance learning from sparsely annotated video,"pp. 1433–1440, IEEE, Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR) Colorado Springs, CO, USA, 20 June 2011.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, IEEE, Las Vegas, NV, USA, 27 June 2016.

[19] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, IEEE, Honolulu, Hawaii, USA, 21 July 2017.

[20] R. P. S. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[21] R. Mahler, "Phd filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, 2007.

[22] R. P. Mahler, "Statistical multisource-multitarget information fusion," *Artech House Norwood*, vol. 685, MA, 2007.

[23] E. Maggio, M. Taj, and A. Cavallaro, "Efficient multitarget visual tracking using random finite sets," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1016–1027, 2008.

[24] E. Maggio and A. Cavallaro, "Learning scene context for multiple object tracking," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1873–1884, 2009.

[25] D. Y. Kim, B.-N. Vo, and B.-T. Vo, "Online visual multi-object tracking via labeled random finite set filtering," 2016, https://arxiv.org/abs/1611.06011.

[26] S. S. Singh, B.-N. Vo, A. Baddeley, and S. Zuyev, "Filters for spatial point processes," *SIAM Journal on Control and Optimization*, vol. 48, no. 4, pp. 2275–2295, 2009.

[27] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[28] B.-N. Ba-Ngu Vo, S. Singh, and A. Boucet, "Sequential Monte Carlo methods for multi-target filtering with random finite sets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, 2005.

[29] K. Granstrom, C. Lundquist, and O. Orguner, "Extended target tracking using a Gaussian-mixture phd filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3268–3286, 2012.

[30] N. L. Baisa and A. Wallace, "Development of a N-type GM-PHD filter for multiple target, multiple type visual tracking," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 257–271, 2019.

[31] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 740–755, Springer, Zurich, Switzerland, 6 September 2014.

[32] R. P. S. Mahler, B.-T. Vo, and B.-N. Vo, "Cphd filtering with unknown clutter rate and detection profile," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3497–3513, 2011.

[33] D. Fox, "Kld-sampling: adaptive particle filters," in *Advances in Neural Information Processing Systems*, pp. 713–720, 2002.

[34] L. Huang, X. Zhao, and K. Huang, "Got-10k: a large high-diversity benchmark for generic object tracking in the wild," 2018, https://arxiv.org/abs/1810.11981.

[35] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "Youtube-boundingboxes: a large high-precision human-annotated data set for object detection in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5296–5305, 19 June 2017.

[36] X. X. Wang, X. Xiaoxu Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 539–555, 2009.

[37] S. Ren, K. He, R. Girshick, J. Sun, and R.-C. N. N. Faster, "Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[38] K. K. Singh, F. Xiao, and Y. Jae Lee, "Track and transfer: watching videos to simulate strong human supervision for weakly-supervised object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3548–3556, IEEE, Las Vegas, NV, USA, 27 June 2016.

[39] S. Deshmukh and T.-S. Moh, "Fine object detection in automated solar panel layout generation," in *Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1402–1407, IEEE, Orlando, FL, USA, 17 December 2018.

[40] M. Kang, K. Ji, X. Leng, and Z. Lin, "Contextual region-based convolutional neural network with multilayer fusion for sar ship detection," *Remote Sensing*, vol. 9, no. 8, p. 860, 2017.

[41] G. Bertasius, L. Torresani, and J. Shi, "Object detection in video with spatiotemporal sampling networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 331–346, Springer, Munich, Germany, 8 September 2018.

[42] Y. Li, D. Huang, D. Qin, L. Wang, and B. Gong, "Improving object detection with selective self-supervised self-training," *Computer Vision - ECCV 2020*, Springer, in *Proceedings of the European Conference on Computer Vision*, pp. 589–607, 8 September 2020.

[43] W. Lee, J. Na, and G. Kim, "Multi-task self-supervised object detection via recycling of bounding box annotations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4984–4993, IEEE, Long Beach, CA, USA, 15 June 2019.

[44] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu, "Deep self-taught learning for weakly supervised object localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1377–1385, IEEE, San Juan, PR, USA, 21 July 2017.

[45] A. Ghahremani, E. Bondarev, and P. H. N. de With, "Towards multi-class detection: a self-learning approach to reduce inter-class noise from training dataset," *International Society for Optics and Photonics*, vol. 11041, Article ID 110411M, 2019.

[46] F. Xiao and Y. Jae Lee, "Track and segment: an iterative unsupervised approach for video object proposals," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 933–942, IEEE, Las Vegas, NV, USA, 27 June 2016.

[47] T. Li, S. Sun, M. Bolić, and J. M. Corchado, "Algorithm design for parallel implementation of the SMC-PHD filter," *Signal Processing*, vol. 119, pp. 115–127, 2016.

[48] Y. Liu, A. Hilton, J. Chambers, Y. Zhao, and W. Wang, "Non-zero diffusion particle flow smc-phd filter for audio-visual multi-speaker tracking," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4304–4308, IEEE, Calgary, AB, Canada, 15 April 2018.

[49] F. Lian, C. Han, and W. Liu, "Estimating unknown clutter intensity for phd filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 4, pp. 2066–2078, 2010.

[50] C. Li, W. Wang, T. Kirubarajan, J. Sun, and P. Lei, "Phd and cphd filtering with unknown detection probability," *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3784–3798, 2018.