ROBUST OBJECT TRACKING BY INTERLEAVING VARIABLE RATE COLOR PARTICLE FILTERING AND DEEP LEARNING

B. Akok, F. Gurkan, O. Kaplan, B. Gunsel

Multimedia Signal Processing and Pattern Recognition Group, Istanbul Technical University, Turkey
{gunselb} @itu.edu.tr

ABSTRACT

We propose an effective combination of discriminative and generative tracking approaches in order to take the benefits from both. Our algorithm exploits the discriminative properties of Faster R-CNN which helps to generate target specific region proposals. A new proposal distribution is formulated to incorporate information from the dynamic model of moving objects and the detection hypotheses generated by deep learning. We construct the generative appearance model from the region proposals and perform tracking through sequential Bayesian filtering by variable rate color particle filtering (VRCPF). Test results reported on CVPR2013 benchmarking data set demonstrate that the interleaving of tracker and detector enables us to effectively update the target distribution that significantly improves robustness to illumination changes, scale changes, high motion and occlusion.
.

Index Terms— Object tracking**,** particle filtering, deep learning,

1. INTRODUCTION

The goal of object tracking is to estimate the state of a target object in a video sequence. Accuracy in object tracking relies heavily on how robust the representation of target appearance is against scale and illumination changes, occlusion, fast motion and background clutter. A number of discriminative and generative methods have been proposed on the robust object tracking problem. The discriminative framework aims to learn a classifier that extracts target from surrounding background. In generative framework, the target appearance is typically described by a statistical model estimated from tracking results in previous frames. Latest work on tracking integrates generative and discriminative methods to take advantages of both and proposes a number of tracking-by-detection methods to improve tracking accuracy [1].

Despite the increased tracking accuracy, the main difficulty of tracking-by-detection algorithms is it is not clear how to couple the estimated (tracked) object position with the detected object labels. This is because the objective for the classifier is not explicitly coupled to the objective for the tracker (estimation of object position). To achieve this, various learning algorithms have been incorporated including online boosting [2], multiple instance learning [3], and online random forest [4]. These approaches are limited to using hand-crafted features for target representation, such as Haar-like features, histogram features and so on, which may not be effective to handle latent challenges imposed on object tracking. In [5] a framework for adaptive visual object tracking is presented based on structured output prediction. By explicitly allowing the output space to express the needs of the tracker, the proposed method, Struct, uses a kernelised structured output support vector machine (SVM), which is learned online to provide adaptive tracking. According to the latest benchmarking papers Struct provides highest performance on CVPR2013 benchmarking video data set [6]. In [7] a new proposal distribution is formulated by integrating Viola Jones detector to particle filter tracking. Inspiring from this work, we propose an effective combination of discriminative and generative approaches that exploits the discriminative properties of deep learning in constructing the generative appearance model which is used to perform tracking through sequential Bayesian filtering. Our motivation is to increase robustness to scale, illumination, pose variations and occlusion.

Since Krizhevsky et al., [8] demonstrated significant performance improvement in ImageNet challenge, Convolutional Neural Network (CNN) has been applied to represent objects in various computer vision tasks including object recognition [9] and object detection [10]. Despite such popularity, there are only few attempts to employ CNNs for visual tracking since offline classifiers are not appropriate for visual tracking conceptually and online learning based on CNN is not straightforward due to large network size. On the other hand, [11] proposes a target-specific CNN for object tracking, where the CNN is trained incrementally during tracking with new examples obtained online. A tracking algorithm is proposed in [12] based on a pre-trained CNN to represent target. On top of the hidden layers in the CNN, an additional layer of an online SVM is put to learn a target appearance discriminatively against background.

In this paper we propose a tracking-by-detection method that combines the strengths of particle filter tracking and deep learning to fulfill the requirements of efficient tracking of non-rigid objects. As we interleave particle filtering with deep learning based detection we call our method interleaving deep particle filtering (IDPF). The deep learning improves the accuracy in detecting objects while the particle filtering enables us to keep track of them. We formulate a novel proposal distribution to incorporate information from the dynamic model of moving objects and the detection hypotheses generated by deep learning. As the deep visual object detector we use faster R-CNN [13] that employs an efficient method to generate candidates which capture the extents of the object accurately, by estimating the best proposal containing the object of interest from several candidates. In order to improve the tracking performance we integrate the variable rate particle filtering [14] into the conventional color based particle filtering [15]. It is shown that the interleaving of deep learning and particle filtering not only increases the tracking performance but also enables us to adaptively update the target distribution. Test results reported on Visual Tracking Benchmark TB-50 data set [1,6] demonstrate that the IDPF allows us to effectively update the target distribution that significantly improves robustness to illumination changes, scale changes, high motion and occlusion.

2. TRACKING BY VARIABLE RATE COLOR BASED PARTICLE FILTER

We employ the variable rate color based particle filter (VRCPF) introduced in [15] that integrates the variable rate particle filter [14] into color based particle filter [16] to improve the tracking of highly maneuvering objects.

Conventional particle filter defines each sample vector s_t as

$$s_t = \{x, y, \dot{x}, \dot{y}, H_x, H_y, \alpha\} \quad (1)$$

where x, y stands for the location information of region of interest (ROI), $\dot{x}, \dot{y}$ refers the corresponding velocity components, H_x, H_y denote the length of the half axes of ROI and α denotes scaling parameter.

The system model is used for propagating each sample in order to predict the state of potential target at the next step as,

$$s_{t+1} = A * s_t + \varphi_t \quad (2)$$

where A is a deterministic propagation matrix and φ_t is a multivariate Gaussian noise.

In the conventional particle filters, state variable $\mathrm{s_t}$ depends time index t where t∈N. In variable rate model, the new state vector is defined as $\mathrm{x_k} = (\mathrm{s_k}, \tau_\mathrm{k})$ where k∈N is a discrete state index and τ_k is state arrival time [15,16]. The distribution of time points is given by $\tau_{k+1} - \tau_k \sim \mathrm{G}(\gamma_\tau, \beta_\tau)$ where G refers Gamma distribution and γ_τ, β_τ are its parameters.

Although in variable filtering model state update process is governed by the random time points, observation updates still take place at observation times denoted by t. Observation density function at time k is defined as $p(y_k | x_{N_t})$ and can be calculated by interpolated sample vector $\hat{\mathrm{s}}_\mathrm{t}$ which is a deterministic function of the neighborhood states $x_{N_t} = \{x_k; k \in N_t(1{:}H)\}$. For simplicity we have chosen to use only two states as the neighborhood that is the states just before and after the observation time t. And $\hat{\mathrm{s}}_\mathrm{t}$ is calculated by Eq. 3 with linear interpolation of those two states.

$$\hat{\mathrm{s}}_\mathrm{t} = s_k + \frac{t-\tau_{k-1}}{\tau_k - \tau_{k-1}}(s_k - s_{k-1}) \quad (3)$$

Our observation model is based on RGB color histograms [16]. Initially the target ROI including N pixels is specified in the form of a bounding box. Let $\mathrm{I_t^{tar}}$ refers the target ROI at time t. $\mathrm{p}_{\hat{\mathrm{s}}_\mathrm{t}}(\mathrm{u})$, the probability of a color bin u at time t for the target color model is formulated by Eq. 4,

$$\mathrm{p}_{\hat{\mathrm{s}}_\mathrm{t}}(\mathrm{u}) = \mathrm{f} \textstyle\sum_{\mathrm{i} \in \mathrm{I}_t^{\mathrm{tar}}} \eta_\mathrm{i} \delta(\mathrm{h}(\mathrm{v_i}) - \mathrm{u}) \quad (4)$$

where δ is the delta function, η_i is the normalizing constant which assigns smaller weights to the pixels that are further away from the center of ROI. $\mathrm{h(v_i)}$ assigns the color values of pixel $\mathrm{v_i}$ to the corresponding bin, $\mathrm{v_i}$ is the pixel value at $\mathrm{i} \in \mathrm{I_t^{tar}}$. The normalization factor f is defined as $\sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{1}{\eta_\mathrm{i}}$. And $\mathrm{p}(\hat{\mathrm{s}}_\mathrm{t}) = \{\mathrm{p}_{\hat{\mathrm{s}}_\mathrm{t}}(\mathrm{u}) | \mathrm{u}{:}\, 1, .., \mathrm{N}\}$ denote a kernel density estimate of the color distribution at time t [16].

The target ROI $\mathrm{I_0^{tar}}$ is initialized in the first frame, as in a standard tracking-by-detection setup where $\mathrm{p}(s_0)$ represents corresponding initial color density referred as target density. Similarly $\mathrm{I_t^{par}}$ refers to the hypotheses ROIs provided by the each particle of the filter, where the corresponding color distribution $\mathrm{m}_{\hat{\mathrm{s}}_\mathrm{t}}(\mathrm{u})$ and a kernel density estimate of the color distribution $\mathbf{m}(\hat{\mathrm{s}}_\mathrm{t})$ is formulated similar to Eq.4 but for i∈ $\mathrm{I}_t^{\mathrm{par}}$.

As in [15], we use Bhattacharya distance formulated by Eq. 5 to measure the distance distance between the target and hypotheses distributions,

$$\mathrm{d}[\mathbf{p}(\hat{\mathrm{s}}_\mathrm{t}), \mathbf{m}(\hat{\mathrm{s}}_\mathrm{t})] = \sqrt{1 - \rho[\mathbf{p}(\hat{\mathrm{s}}_\mathrm{t}), \mathbf{m}(\hat{\mathrm{s}}_\mathrm{t})]} \quad (5)$$

where $\rho[\mathbf{p}(\hat{\mathrm{s}}_\mathrm{t}), \mathbf{m}(\hat{\mathrm{s}}_\mathrm{t})]$ refers to Bhattacharyya similarity coefficient which is calculated as $\sum_{\mathrm{u}=1}^{\mathrm{K}} \sqrt{\mathrm{p}_{\hat{\mathrm{s}}_\mathrm{t}}(\mathrm{u}).\, \mathrm{m}_{\hat{\mathrm{s}}_\mathrm{t}}(\mathrm{u})}$, between target and hypotheses distributions at t where K is the number of color bins. The Bhattacharya distance is used to calculate the likelihood distribution for each particle given by Eq. 6,

$$p\left(y_t \middle| \hat{\mathrm{s}}_\mathrm{t} = x_{N_t}\right) \propto \mathrm{e}^{-\lambda(\mathrm{d}^2[\mathbf{p}(\hat{\mathrm{s}}_\mathrm{t}), \mathbf{m}(\hat{\mathrm{s}}_\mathrm{t})])} \quad (6)$$

where λ is a smoothing parameter. $\mathrm{p}(\mathrm{y_t}|\hat{\mathrm{s}}_\mathrm{t})$ constitutes observation model of the color based particle filtering.

Finally weight update rule can be formulated as Eq.7 where $\hat{\mathrm{s}}_\mathrm{t}^\mathrm{z}$ is the state vector of z*th* particle [17].

$$\omega_\mathrm{t}^\mathrm{z} = \omega_{\mathrm{t}-1}^\mathrm{z} \frac{\mathrm{p}(\mathrm{y_t}|\hat{\mathrm{s}}_\mathrm{t}).\mathrm{p}(\hat{\mathrm{s}}_\mathrm{t}^\mathrm{z}|\hat{\mathrm{s}}_{\mathrm{t}-1}^\mathrm{z})}{\mathrm{q}(\hat{\mathrm{s}}_\mathrm{t}^\mathrm{z}|\hat{\mathrm{s}}_{\mathrm{t}-1}^\mathrm{z}, \mathrm{y}_{0:\mathrm{t}})}. \quad (7)$$

In Eq. 7, $\mathrm{q}(\hat{\mathrm{s}}_\mathrm{t}^\mathrm{z}|\hat{\mathrm{s}}_{\mathrm{t}-1}^\mathrm{z}, \mathrm{y}_{0:\mathrm{t}})$ and $\mathrm{p}(\mathrm{y_t}|\hat{\mathrm{s}}_\mathrm{t})$ denote the proposal distribution and state transition distribution, respectively.

By using the proposals of Z particles, the mean state of the object is estimated at each frame as,

$$\mathrm{E}[\mathbf{s}_\mathrm{t}] \sim \textstyle\sum_{\mathrm{i}=1}^{\mathrm{Z}} \mathrm{w}_\mathrm{t}^\mathrm{z}.\, \hat{\mathrm{s}}_\mathrm{t}^\mathrm{z} \quad (8)$$

It is clear that decreasing the distance of target and estimated distributions maximizes the likelihood so as the accuracy. To achieve this, we propose a target update method described in Section 4. As a result of target update we indirectly update the observation model.

3. OBJECT DETECTION BY FASTER R-CNN

Recent object detection algorithms rely on region proposal methods and region based convolutional neural networks, such as; SPPNet [9], RCNN [10], Fast-RCNN [19] and more recently Faster R-CNN [13]. In this work, Faster R-CNN, that received state-of-the-art accuracies in object detection task on various datasets like; PASCAL VOC 2007 [18], and Microsoft COCO [11] datasets, is used to detect objects in an image.

Faster R-CNN is a convolutional neural network (CNN) architecture which is used to predict locations and classes of multiple objects in an image. Faster R-CNN consists of two modules; a fully convolutional neural network which is called Region Proposal Network (RPN) to propose candidate rectangular object locations with an objectness score and a CNN to estimate class of object and refine its location for each proposal of RPN, named as Fast R-CNN [19].

In training and inference of Faster R-CNN, images are input of the network. Input images are processed by several convolutional layers and same convolutional feature maps are used for both RPN and Fast R-CNN. Hence, RPN and Fast R-CNN share parameters of these convolutional layers that produce convolutional feature maps and this parameter sharing between two modules decreases the number of parameters of Faster R-CNN.

4. INTERLEAVING VRCPF AND FASTER R-CNN

We propose a novel tracking framework that the VRCPF tracker follows the object from frame to frame while output of the faster R-CNN detector is integrated into the tracking to localize the object. This is achieved by replacing the conventional particle filtering proposal distribution as in Eq.9.

$$q(s_{t+1}|s_t,\, y_{0:t}) = \alpha.\, \mathcal{E}\left(s_t^{dl}, s_t^{pf}\right) q_{dl}(s_t^{dl}, y_t) + \left(1 - \alpha\, \mathcal{E}\left(s_t^{dl}, s_t^{pf}\right)\right) p_{pf}(s_{t+1}^{pf}\left|s_t^{pf}\right) \qquad (9)$$

where α is set to 1 when the faster R-CNN detects an object and 0 otherwise. When the deep detector fails because of blur, abrupt illumination changes or pose changes, α is set to 0 while VRCPF continues tracking.

When a new video frame y_t is observed, the first additional term of Eq.9 will be active if the deep detector provides an object proposal with a high objectiveness score where $q_{dl}(s_t^{dl}, y_t)$ represents the proposal, s_t^{dl} can be interpreted as the state vector corresponding to the detected object. The second additional term of Eq.9 models the contribution of particle filter tracking where s_t^{pf} denotes the state vector estimated by VRCPF at frame t. $p_{pf}(s_{t+1}^{pf}\left|s_t^{pf}\right)$ corresponds to the proposal distribution of the VRCPF. The function $\mathcal{E}\left(s_t^{dl}, s_t^{pf}\right)$ controls the interleaving and is formulated as in Eq. 10,

$$\mathcal{E}\left(s_t^{dl}, s_t^{pf}\right) = u(th_g - g_t).\, u\left(th_{dd} - d\left[\mathbf{p}\left(s_t^{pf}\right), \mathbf{m}(s_t^{dl})\right]\right) \quad (10)$$

Following, we explain the idea behind introducing Eq. 9 and Eq. 10 and give details of the proposed object tracker, IDPF.

The tracker monitors the Bhattacharya distance at each step t. If the distance to current target distribution is low enough ($d\left[\mathbf{p}\left(s_t^{pf}\right), \mathbf{m}\left(s_t^{pf}\right)\right] < 0.1$ to guarantee high tracking accuracy) the object proposal estimated by VRCPF is accepted and the tracker propagates the samples to the next state, *t*+1.

Because of partial self-occlusion, high object motion, or illumination changes, the distance may increase even though the tracker works properly. When the distance increases we check the level of agreement between tracker and deep detector. If the overlap between the ROI tracked by VRCPF ($I_{t,dl}$) and the object proposal of faster R-CNN ($I_{t,pf}^{est}$) is high, VRCPF keeps track of propagating states since both the tracker and detector agree on the tracked object of interest. The first multiplicative term in Eq.10 controls this case where g_t denotes the overlapping ratio calculated as in Eq.11 and th_g is the threshold imposed on g_t. We control the priority given to deep detector by changing the threshold th_g.

$$g_t = \frac{area(I_{t,dl} \cap I_{t,pf}^{est})}{area(I_{t,dl} \cup I_{t,pf}^{est})} \quad . \qquad (11)$$

It is common practice that particle filter tracking fails because of an abrupt illumination change, object scale change, or occlusion. To alleviate this drawback, when the overlapping ratio g_t is small, we give priority to the faster R-CNN detector. In this case, the first multiplicative term of Eq.10 will be equal to one and the second multiplicative term controls the level of priority given to deep learning. Decision on whether to switch to the faster R-CNN proposal is given by checking the Bhattacharya distance between the current target proposal and the deep proposal provided by faster R-CNN. If the distance $d\left[\mathbf{p}\left(s_t^{pf}\right), \mathbf{m}(s_t^{dl})\right]$ is lower than a pre-specified threshold th_{dd}, than $u\left(th_{dd} - d\left[\mathbf{p}\left(s_t^{pf}\right), \mathbf{m}(s_t^{dl})\right]\right)$ will be equal to 1 and the second additive term of Eq. 9 will be equal to zero that guarantees switching to the faster R-CNN proposal. Consequently we update the target proposal by the color distribution of the detected ROI. This yields an update on the observation model given by Eq.6. The proposed target update scheme highly improves the tracking performance because of the excellent localization capability of faster R-CNN.

5. TEST RESULTS

Faster R-CNN is trained on Microsoft COCO [11] dataset using implementation given in [20]. VGG-16 [12] pre-trained on ImageNet Classification-Localization [21] dataset is used as CNN architecture. Microsoft COCO, contains 328k images and 91 object categories which also correspond to the object categories included in Visual Tracker Benchmark(VTB)[6] videos. VTB contains 50 video sequences. Due to the fact that "Face" category is exist only in test dataset, training dataset is augmented with manually annotated face ROIs using PASCAL VOC 2007 dataset [18]. We tested IDPF on 32 videos of TB-50 dataset [1,6].

Test results demonstrate that integration of particle filter tracking with deep learning by the proposed IDPF significantly improves robustness to illumination, scale and pose changes. Figure 1 shows frames from the benchmark video sequences where the objects tracked by the proposed IDPF, VRCPF, faster R-CNN and ground truth are overlaid on the frames. The sequence in Fig 1.a has considerable amount of scale change, partial occlusion and pose change. It can be seen that while all methods track the car accurately at frame 4, VRCPF cannot keep up with the scale changes. Even though faster R-CNN fails to detect car at frame 164, whenever it tracks it gives accurate location of the object regardless of the scale and incorporating this accuracy IDPF becomes robust against scale change thus accurately tracks the object at all the frames. Fig 1.b features a low resolution sequence with illumination changes and blurring that makes tracking harder for VRCPF as target distribution changes rapidly. It is also difficult for faster R-CNN to detect the object in every frame due to blurriness and low resolution. However, IDPF tracks the object accurately as detections from faster R-CNN update the target distribution that improves target appearance model. It can be seen from the frames of Fig.1.b, while VRCPF begins to diverge from the ground truth IDPF keeps on the object. Fig 1.c illustrates frames of a sequence with object shape change and busy background. VRCPF tracker fails quickly as color information is cluttered by the busy background but IDPF stays on the object as target distribution is corrected with the results from faster R-CNN. This can also be seen from the success rate plot for that same sequence in Fig.2. It is clearly seen that our method IDPF which combines the deep learning and VRCPF outperforms both by a large margin in this sequence. .

Tracking performance is evaluated by the overlap ratio which is defined as the area of intersection over union (IoU) between two regions of interest. If the overlap ratio between the tracked ROI and ground truth is higher than a given threshold, the object is said to be successfully tracked in that frame. Success rate is the ratio between number of successful frames and total number of a)

b)

c)

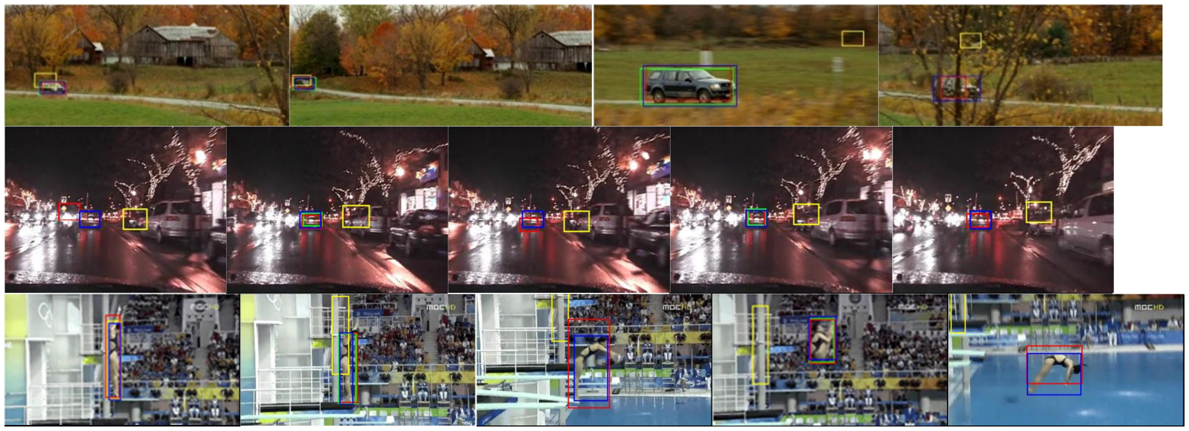

Fig 1. Examples of tracking results. a) CarDark (frames 4, 83, 164, 201), b) CarScale (frames 5, 35, 80, 140, 195) (c) Diving (frames 5, 35, 80, 140, 195). VRCPF(yellow), Deep Learning(green), IDPF-1.0(red) and Groundtruth(blue)

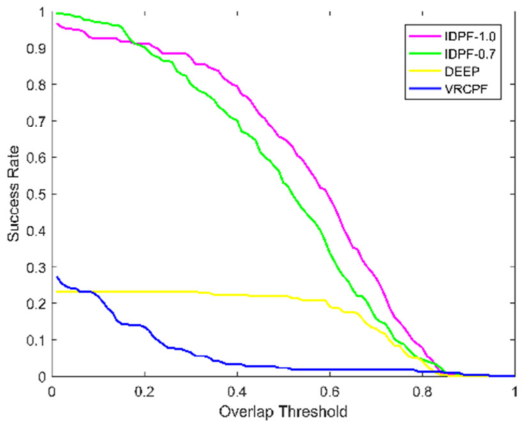

Fig 2. Success rates for Diving sequence.

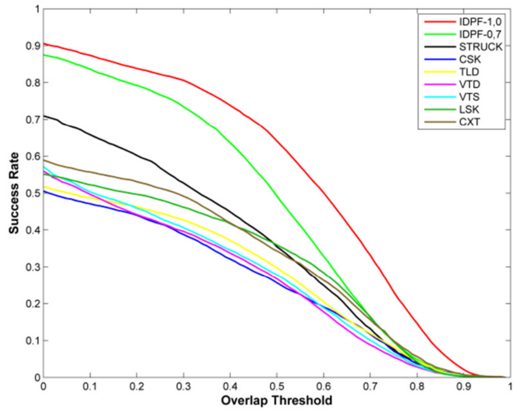

Fig 3. Average success rates versus overlap threshold.

frames in the sequence. This evaluation is preferred as it shows how much of the sequence is successfully tracked for the desired margin of overlap which is an important metric in object tracking and 0.5 is a commonly used overlap threshold in literature [1]. After finding individual success rates versus to overlap threshold we report the average of them on Fig. 3. IDPF is evaluated at different th_g values to observe the effect of agreement level controlled by g_t on tracking performance. We first set th_g to 1.0 means the tracker remains at VRCPF mode only the agreement is full otherwise gives the priority to deep detector (IDPF-1.0). Performance gain provided by IDPF is also evaluated at a moderate agreement level by setting th_g to 0.7 (IDPF-0.7). Note that a higher threshold means we reset our tracker more frequently.

Fig.3 shows the overall performance of IDPF compared to baseline tracker results from [1]. Success rates are plotted against increasing overlap thresholds for IDPF and top seven trackers of benchmark, STRUCK[5], CXT[22], CSK[23], TLD[24], VTD [25], VTS [26], LSK[27]. It is observed that success ratios achieved by IDPF are better than the baseline trackers. Specifically at overlap threshold 0.5 our method achieves approximately 15% percent higher accuracy when th_g is set to 0.7. The gain of IDPF increases to 30% when th_g is set to 1.0 means the target model is updated by deep detector for all the frames. It can also be seen that while other methods converge very quickly at higher overlap thresholds. This success is resulting from the superior localization of the deep learning based detection, thus interleaving deep learning effects the results dramatically especially when there is high motion and large scale changes in the sequence. As the localization performance of the deep learning based detection is high it is appealing to select a higher th_g but picking a higher th_g also means that we reset our filter much more frequently and this can be undesirable in some situations as resetting the filter loses smoothness of the tracking, disrupts hidden states of the filter like speed of the object and adds computational cost. Average percentage of resetting the filter of our algorithm is 66% at priority threshold 1, and 45% at threshold 0.7 for all videos.

6. CONCLUSIONS

In this paper we propose an effective combination of discriminative and generative approaches that exploits the discriminative properties of deep learning in constructing the generative appearance model which is used to perform tracking through sequential Bayesian filtering. Our motivation is to increase robustness to scale, illumination, pose variations and occlusion. Test results demonstrate that the IDPF method improve upon benchmark results.

7. REFERENCES

[1] Y. Wu, L. Lim, M. Yang, "Object tracking benchmarking," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol.37, no.9, pp.1834-1848, Sept. 2015.

[2] H. Grabner, M. Grabner, and H. Bischof. "Real-Time Tracking via On-line Boosting". in *Proc. British Mach. Vis. Conf.*, pp.6.1-6.10, 2006.

[3] B. Babenko, M. Yang, S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 33, no.7, pp.1619-1632, Aug. 2011.

[4] J. Gall, N. Razavi, L. van Gool, "An introduction to random forest for multi-class object detection,". In Dellaert F., Frahm J.-M., Pollefeys M., Leal-Taixe L., and Rosenhahn B., editors, *Outdoor and Large-Scale Real-World Scene Analysis,* volume 7474 of Lecture Notes in Computer Science, pp. 243–263. Springer Berlin Heidelberg, 2012

[5] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng,Stephen L. Hicks, and Philip H. S. Torr, "Struct: Structured output Tracking with Kernels," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol.38, issue.10, Oct. 2016.

[6] Yi Wu, Jongwoo Lim, Ming-Hsuan Yang. Online Object Tracking: A Benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* pp.2411-2418, 2013.

[7] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. "A boosted particle filter: Multitarget detection and tracking, " in *Proc. European Conference on Computer Vision (ECCV),* pp.28-39, 2004

[8] Krizhevsky, A., Sutskever, I., Hinton, G.E:, "ImageNet classification with deep convolutional neural networks," in *Neural Information Processing Systems NIPS*, pp."1106-1114, 2012.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol.37, no.9, pp.1904-1916, Sept. 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*. Pp.580-587, 2014.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Proc. European Conference on Computer Vision (ECCV),* pp.740-755, 2014.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. International Conference on Learning Representations (ICLR),* 2015.

[13] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Advances in neural information processing systems,* pp.91-99, 2015.

[14] S. Godsill and J. Vermaak, "Variable rate particle filters for tracking applications," in *Proc. IEEE Statist. Signal Processing*, pp.1280-1285, 2005

[15] D.Kumlu and B.Günsel. "Variable rate adaptive color-based particle filter tracking," in *Proc. IEEE International Conference on Image Processing (ICIP),* 2016

[16] Nummoiaro, Katja, Ester Koller-Meier and Luc Van Gool. "An adaptive color-based particle filter," *Image and vision computing*, 21(1),99-110, 2003.

[17] Ulker Y., Gunsel B., "Multiple model target tracking with variable rate particle filters," *Digital Signal Processing*, 2012, 22(3):417-429.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *"The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,"* 2007

[19] R. Girshick, "Fast R-CNN," *in Proc. IEEE International Conference on Computer Vision (ICCV),* pp.1440-1448 2015

[20] Faster R-CNN*: https://github.com/rbgirshick/py-faster-rcnn*

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," in *International Journal of Computer Vision (IJCV),* vol.115, no.9, pp.211-252, 2015

[22] T. B. Dinh, N. Vo, and G. Medioni, , A. "Context tracker: Exploring supporters ans distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp.303-338, 2011.

[23] J. A. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels," in *Proc. European Conference on Computer Vision (ECCV),* pp.702-715, 2012.

[24] Z. Kalal, J. Matas, and K. Mikolajczyk. **"**P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., pp.49-56, 2010.

[25] J. Kwon and K. M. Lee. "Visual Tracking Decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., pp. 1269–1276, 2010.

[26] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Proc. IEEE Int. Conf. Comput. Vis.,* , pp. 1195–1202, 2011.

[27] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. "Robust Tracking using Local Sparse Appearance Model and K-Selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., pp. 1313–1320, 2011.