A Real-time and Online Multiple-Type Object Tracking Method with Deep Features

Yi-Hsuan Hsu* and Jiun-In Guo†
* *Graduate Degree Program of College of Electrical and Computer Engineering, National Chiao Tung University, 1001 University Road, Hsinchu City, Taiwan, R.O.C.*
E-mail: piccolo1992514@yahoo.com.tw Tel: +886-03-5712121
† *Institute of Electronics, National Chiao Tung University, 1001 University Road, Hsinchu City, Taiwan, R.O.C.*
Pervasive Artificial Intelligence Research Labs (PAIR Labs),
E-mail: jiguo@nctu.edu.tw Tel: +886-03-5712121

***Abstract*— Object tracking is one of the most important things in intelligent vision system. Meanwhile, the most challenging issue in object tracking is how to keep the target's identity unchangeable with limited power consumption. In this paper, we propose a real-time and online tracking method to track multiple types of objects (e.g. pedestrian and car). Furthermore, to handle the ID switching problem, we provide a lightweight deep learning model which can recognize the similarity of objects. It can effectively solve the ID switching problem resulted from occlusion. Finally, we do some experiments to demonstrate that the proposed method achieves the state-of-the-art performance with less power consumption. The proposed method can solve the problem of high computation of tracking and keep the high accuracy of counting results with low ID switching rate. The experimental result shows that the average counting accuracy of the proposed method can reach more than 93% on pedestrian and vehicle counting applications. Also, it shows that the proposed method improves 68.2% on average of ID switching rate than previous works.**

Keywords—Real-time tracking, Online tracking, Deep learning object detection and tracking

I. INTRODUCTION

With the significantly increasing number of cameras worldwide, providing low power and low computation tracking algorithm along with relatively high accuracy become very important in practice. Thanks to the explosion of AI technology development such as CNN, we have a very good visual feature rather than traditional image processing feature. To make the tracking results better, we extract the deep feature of the image from CNN. In the commercial aspect, the most important thing is to minimize the cost of the product and keep good performance on a real application. It is necessary for us to design a simple model to deal with our application.

In this paper, we propose a tracking-by-detection method that trains a strong and lightweight deep learning model (i.e. pva-lite) and associates the detected objects to the trackers by deep feature from our lightweight model. And we highly reuse the deep learning model to improve detection and tracking performance. By using pva-lite model as the object detector, the counting accuracy of the proposed method can reach more than 93% on both pedestrian and vehicle counting application. Furthermore, we can decrease 68.2% of ID switchinf rate. The system performance can achieve 720x480@10fps on Nvidia 1080 Ti. Our contribution is summarized in the following:

1. We provide a low power, low-cost tracking and counting solution, which achieves 10 fps in tracking algorithm with two deep learning models.
2. Our online tracking method can track multiple-type objects at the same time and it can be widely used in several applications like ADAS, and surveillance without decreasing the performance.
3. We can improve ID switching rate 68.2% on average by the proposed lightweight model.

II. RELATED WORK

One of the challenges in object counting is how to track multiple types of the object accurately. Multiple types of object tracking are way more difficult than single type object tracking because the object under occlusions should correctly remain their IDs after the occlusions are finished. To track the object accurately, a good feature representation is necessary. FAST [1] extracts feature rapidly while having high repeatability. In the literature, there have been different kinds of descriptors proposed, such as SIFT [2], SURF [3], BRIEF [4], ORB [5], BRISK [6], and FREAK [7]. A clear advantage of binary descriptors that they claim is low computational cost algorithm with competitive accuracy. However, keypoint-based methods can hardly work in tracking object since the shape deformation and size variation will change the positions of key point, which will likely lead to a key point matching failure. As shown in Fig. 1, the left part and the right part of the figure show the same people in two adjacent frames, but clearly, it is easier to have wrong key point matching results.

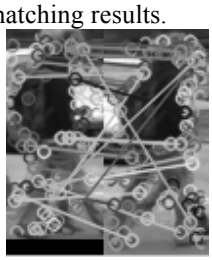

Fig. 1. A case of keypoint matching failure

Using only one feature is not enough in object tracking.

Many existing methods combine different features in their tracking processes. Merad et al. [8] represented a person by combining HSV histograms of the head, torso, and legs. Bousetouane et al. [9] used Haralick texture features and color features as the representation of a person. The tracker will find the best solution by maximizing the Bhattacharyya coefficient and minimizing the Mahalanobis distance between the reference target representation and the candidate target representation. The performance of these methods is easily degraded by occlusions in the crowded scene.

People detection is a key building block in most of the state-of-the-art people tracking methods [10, 11, 12]. In recent years, the performance of people detectors has improved tremendously. As more and more powerful computational units become available nowadays, training complicated deep learning models becomes possible. We would like to take advantage of the high accuracy object detector in the proposed tracking-by-detection method. Hong et al. [13] proposed the pvanet, which is an efficient neural networks for real-time object detection. It achieves state-of-the-art accuracy in multi-category object detection task while minimizing the computational cost by adapting and combining recent technical innovations. In the proposed system, we optimize the pvanet model to be a pva-lite model to do object detection for our target applications.

Many of the most successful tracking methods [10, 14, 15] at present perform tracking by detection, i.e., the target is detected by object detection model in every frame independently. The advantages of using an object detector are that it handles re-initialization if a target has been lost and it is not sensitive to the size or lighting change of the target. Some of the current tracking-by-detection methods assign labels to the detected the object in the whole video sequences iteratively by minimizing a predefined energy function [10]. For example, the trajectories of all the detected objects in the whole video sequences are recorded. Then by defining an energy function as the sum of different energy models, an object is more likely to have a smooth trajectory and smaller energy value. The process is illustrated in Fig. 2. The left page is the frame at time t-1. The right page is the frame at time t. $P_{i,t}$ means the i-th object in the frame t. In our proposed method, we adopt the tracking-by-detection method.

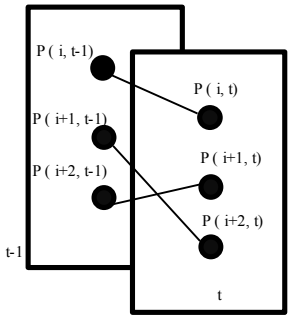

Fig. 2. The adopted tracking-by-detection method

III. PROPOSED METHOD

In this section, we will introduce our proposed method. The overall architecture is shown in Fig 3. First of all, we use deep learning model to do detection for the purpose of initializing the target. Second, we will do the data association to label the identity of the target. Finally, we will predict the next location of the target and feedback the tracking results to detection.

Fig. 3. *Overall architecture*

A. Pva-lite object detection model

Table I shows the simplified Pva-lite model, by reducing the feature extraction layers from 21 to 16. As shown in Table II, Pva-lite model reduces 58% of the execution time at the cost of degrading 4.75% accuracy. It achieves 720x480@40fps on Nvidia 1080 Ti, which is able to meet the performance requirement of the target applications.

Table I PVA-lite structure

Layer Name	Layer Type
Conv1 Conv2 Conv3	Convolutional Layer
Inception3a Inception3b Inception3c Inception3d Inception3e Inception4a Inception4b Inception4c Inception4d Inception4e	Modified Inception
Downsample Upsample Inception3e	HyperNet
RPN Proposal Roi Pooling	FasterRCNN
FC6 FC7	Fully Connected Layer

Table II Performance results

	Pascal VOC 2007		Pascal VOC 2012	
Net work	mAP	speed(fps)	mAP	speed(fps)
ZF-based Faster-RCNN	70.20%	25.3	65.30%	24.8
VGG-based Faster-RCNN	73.20%	8.19	70.40%	7.98
YOLO	63.40%	45	57.90%	43
Fast YOLO	52.70%	155	52.70%	145
SSD300	72.10%	58	70.30%	54
SSD500	75.10%	23	73.10%	22
PVANET	83.85%	18.18	82.50%	17.62
PVANET lite	**79.10%**	**43.47**	**78.42%**	**42.38**
ResNET101 Faster R-CNN	86%	8.63	83.80%	7.518

B. Data association

In Eq. (1), we define IOU between the detection and the trackers. In Fig. 4, the gray dots and are the successfully matching pairs. The black dot doesn't have any match in trackers so it will be initialized as a new tracker. The white dot leaves unconnected, so we will apply Kalman filtering on it to track its trajectory. This process will be deal with Maximum-weighted matching and perfect matching methods.

Data association through IOU method can handle normal case of tracking issue. But, if the occlusion issue happens, the ID switch problem will become very serious. We provide an efficient model, a.k.a IVS-ComNet, to deal with the tricky issue. This method can obviously drop the ID switch number.

$$\boldsymbol{IOU} = \frac{\boldsymbol{Area\ of\ Overlap}}{\boldsymbol{Area\ of\ Union}} \qquad (\mathbf{1})$$

$$\boldsymbol{IOU_{max}} = \mathbf{max}\big(\boldsymbol{IOU_{(di,ti)}}\big) \qquad (\mathbf{2})$$

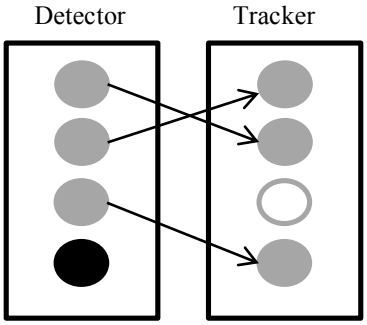

Fig. 4. Data association in the proposed algorithm

C. Maximum-weighted matching and perfect matching

The Maximum-weighted matching can be used to find the perfect matching of the bipartite graph in order to make the data association between detector and tracker efficiently. First, we can use maximum-weighted matching to find the best matching, where a perfect matching means that every vertex just has one best matching.

We suppose that each edge $\boldsymbol{e}$ in the bipartite $\boldsymbol{B}$ connects two points, and every vertex $\boldsymbol{v}$ is covered one time. By this rule, we get the inequality (3), where $\boldsymbol{M'}$ is any perfect matching in $\boldsymbol{B}$ given by the assignment of each edge $\boldsymbol{e}$, and $l(x)$ is a numeric label to point $\boldsymbol{x}$. That is means $\sum_{v\in V} l(v)$ is the upper boundary on the cost of perfect matching. This step can assign a weight to each candidate of the detector and tracker in the bipartite $\boldsymbol{B}$. And we can find all available labeling in the inequality (4), $\forall x \in X, y \in Y$, where X is the group of the vertex on the one side of the bipartite, Y is on the other side. Second, if the matching is perfect, the data association is done. Otherwise, there will be some vertices that are not connected to any other vertex. Start with some matching M and a labeling l, search for another path in M. If the path does not exist, M is the perfect matching. And find the other matching until a perfect matching is found.

$$w(M') = \sum_{e\in E} w(e) \le \sum_{e\in E} \big(l(e_x) + l(e_y)\big) = \sum_{v\in V} l(v) \quad (3)$$

$$l(x) + l(y) \ge w(x, y) \qquad (4)$$

D. IVS-ComNet

In this section, we will introduce the proposed method which can use deep learning feature to compare the similarity between two images. First, we distill the feature map which is from a convolution neural network in Resnet-50. The Resnet-50 model was trained by Imagenet. Second, we normalize the feature onto unit hypersphere, and it will be calculated by cosine similarity to get the distance between two images according to the equation shown in equation (5). If the distance is smaller, the pairs of the image are more similar. Third, we use maximum-weighted matching and perfect matching method to match candidates with each other. Although the result is very good to find a similar image, it cost lots of computation. So we provide a lightweight model to further reduce the computational complexity, where the proposed model is shown in Table III. The proposed model contains fourteen layers of convolution. The input size of the model is 3x128x64 and the output size is 128-dimension.

$$\text{cosine(a, b)} = \frac{\sum_{i,j}^{N} s_{i,j} a_i b_j}{\sqrt{\sum_{i,j}^{N} s_{i,j} a_i a_j}\sqrt{\sum_{i,j}^{N} s_{i,j} b_i b_j}} \qquad (5)$$

Table III IVS-ComNet

Layer	Input size	Output size	Kernel
Con2d Batch Normalization ELU	3x128x64	32x128x64	3x3, s = 1,p = 1
Con2d Batch Normalization ELU Maxpool2d	32x128x64	32x64x32	3x3, s = 1, p = 1
Con2d Batch Normalization Relu Con2d Batch Normalization	32x64x32	32x64x32	3x3, s = 1, p = 1
Con2d BN Relu Con2d BN Con2d BN	64x32x16	64x32x16	3x3, s = 2, p = 1
Con2d Batch Normalization Relu Con2d Batch Normalization	64x32x16	64x32x16	3x3, s = 1, p = 1
Con2d BN Relu Con2d BN Con2d BN	64x32x16	128x16x8	3x3, s = 2, p = 1
Con2d Batch Normalization Relu Con2d Batch Normalization	128x16x8	128x16x8	3x3, s = 1, p = 1
Dense layer Batch Normalization	128x16x8	128	

E. Feedback to region proposal candidates

The region proposal of Faster R-CNN [27] is giving some fixed anchors and doing the regression to propose the regions that do not belong to the background. In our experience, there are some losing targets which are the same targets in a sequence. It will degrade the accuracy of detection. To resolve this problem, we can reuse the tracking results to increase the valid candidates. In Fig.5, we feedback the tracking results into the next frame region proposal candidates. In our experiment, the detection results become much better and stable.

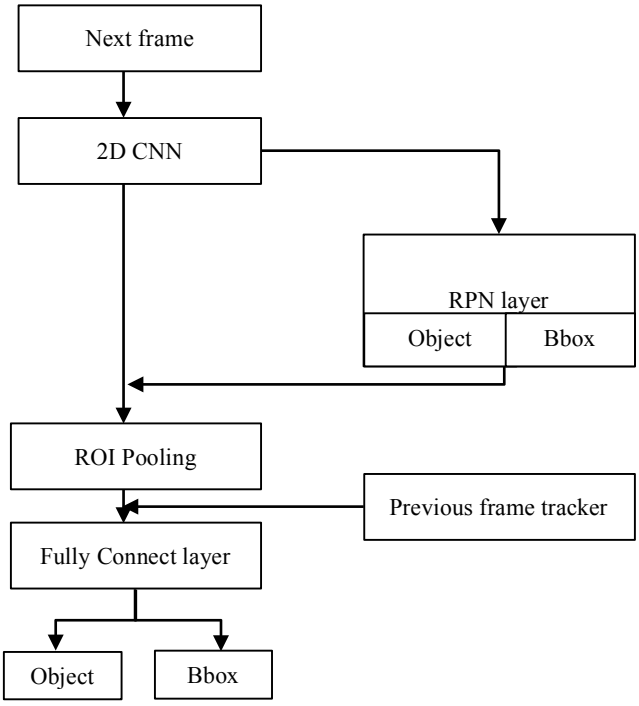

Fig. 5. Feedback to region proposal candidate

F. Object tracking process

The proposed tracking mechanism consists of three essential parts. The first one is the initialization, which is given by the detector. The second one is data association, which is the process of matching detection results to the current trackers. The third one is the prediction, which predicts the position of the lost tracker by applying Kalman filter. In the data association step, each pair of detection and tracker will be given its IOU and deep appearance feature to match their ID. Afterward, if there is still an unmatched tracker, Kalman filter will be applied to predict the target. The bounding box will be predicted by a pre-trained classification model to check the bounding box is background or not. If the tracker is background, the tracker will be deleted, otherwise, it will feedback to region proposal candidates. The details of the tracking process are shown in Fig. 6.

G. Counting strategy

We propose a counting strategy that can judge not only the object passing the line but also the direction of the object. Additionally, our method can handle a polyline situation. The following example will be a polyline build by three points:

We define a polyline shown in Fig. 7 for counting the moving objects that cross the polyline. With the crossing line, we can count the moving objects by using the positive and negative value of crossing value shown as Eq. (6) and (7):

$$\overrightarrow{P_1P'} \times \overrightarrow{P_1P_2} > 0 \rightarrow OUT \quad (6)$$

$$\overrightarrow{P_1P'} \times \overrightarrow{P_1P_2} < 0 \rightarrow IN \quad (7)$$

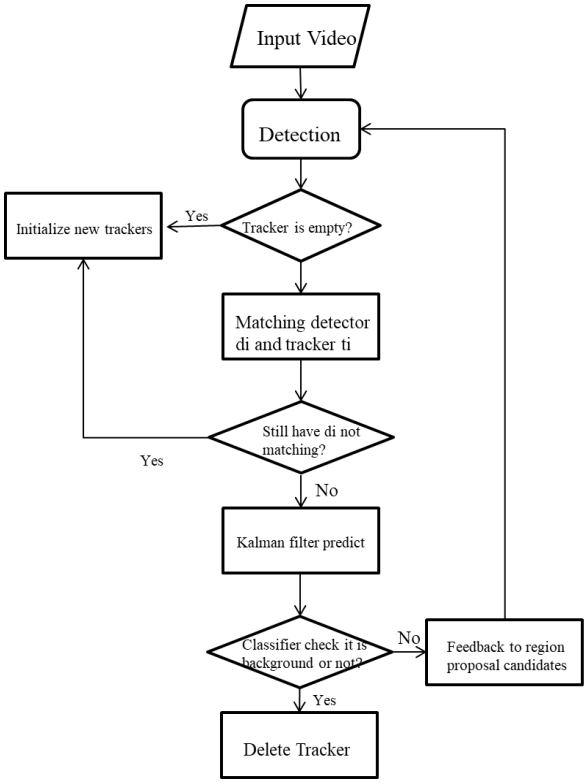

Fig. 6. The proposed object tracking algorithm

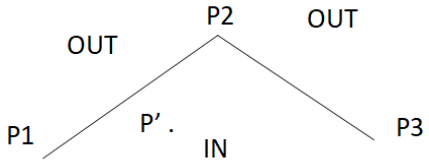

Fig. 7. Counting strategy

IV. EXPERIMENTS

In this section, we design three experiments to verify the advantages of the proposed method.

The first experiment, we design an experiment to evaluate SIFT, the proposed model and the state-of-the-art model, Resnet-50. First, we collect some test data which the class type is the same, as shown in Fig. 8. The data consist of 21 different people, including similar appearance and different background, where each people has three frames of image. Second, these images will be fed into the proposed CNN model and Resnet-50. Third, the output features are projected onto the unit hypersphere and its cosine similarity is calculated to find the most similar target. The results of the proposed model compared to SIFT and Resnet-50 are shown in Table IV. We find that the proposed model outperforms the SIFT in both the processing speed and accuracy. Compared to Resnet-50, the proposed model outperforms in processing speed at the cost of 1% accuracy drop in object identification.

Fig. 8. Test data of people identification

Table IV Results of SIFT, Resnet-50 and IVS-ComNet

	SIFT	Resnet-50	IVS-ComNet
Dataset	X	ImageNet	ImageNet
Layer number	X	50	14
Execution time	3 sec	1.86 sec	0.83 sec
Object identification accuracy	81%	95%	94%

The second experiment, we want to show that our tracking algorithm is very good at counting application. And we prove that feedback tracking results to detection can improve the accuracy of detection. We test the proposed algorithm on some video clips. In Table V, it shows the average counting accuracy of the proposed system can reach about 93%. Fig. 8 shows some demonstration. In Table VI, it shows the improvement of detection results, we can find that the improvement of the car is much better than the person because the motion of the car is more regular. The average improvement rate is 2.5% in our experiment. It is a great improvement in the detection domain.

Table V Counting results

Name	Scene	Ground Truth	Counting Result	Accuracy Rate
Linkou1	Indoor	51	51	100%
Linkou2	Indoor	47	44	94%
Simon1	Outdoor	51	47	92.20%
Simon2	Outdoor	56	51	91.10%
GPO10122	Outdoor	42	50	84%
IronPD	Outdoor	176	165	94%
cap816	Indoor	10	9	90%
cap316	Indoor	15	14	93%
cap817	Indoor	11	10	91%
cap337	Indoor	20	18	90%
5F_1	Indoor	18	17	94%
5F_2	Indoor	27	25	93%
High3	Outdoor	63	63	98.40%
High4	Outdoor	52	52	100%
Rain_High	Outdoor	94	87	100%
Average of Counting Accuracy				93%

Table VI Improvement of detection results

Name	Type	mAP (detection)	mAP (detection+ tracking)	Increasing rate
Linkou1	people	65.3%	67.9%	2.6%
Linkou2	people	62.1%	63.3%	1.2%
High3	car	77.4%	80.2%	2.8%
High4	car	78.5%	81.4%	2.9%
Average improvement (mAP)				2.5%

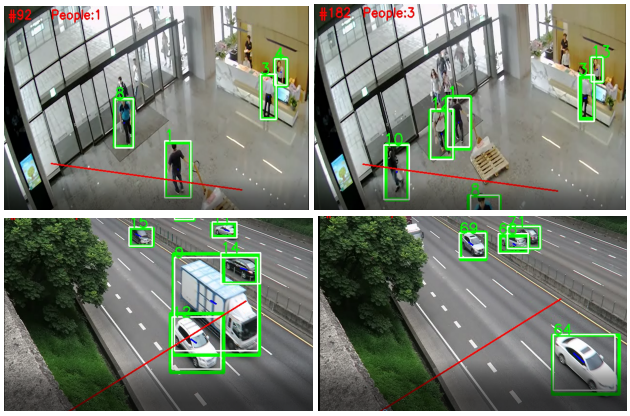

Fig. 9. Demonstration of object counting

The third experiment, we want to show that our model can handle ID switching problems by our tracking algorithm. We find some video clips which are very challenging because these video clips (shown in Fig 10)contain several people who wear similar clothes and they wander randomly, it will lead to a lot of occlusion issue happened. In Table VII, our tracking algorithm with only IOU feature has bad performance on ID switching issue. And we can find that our tracking algorithm with deep appearance feature improves obviously. The best one can improve 81.2%, and the average improvement rate also reaches 68.2%.

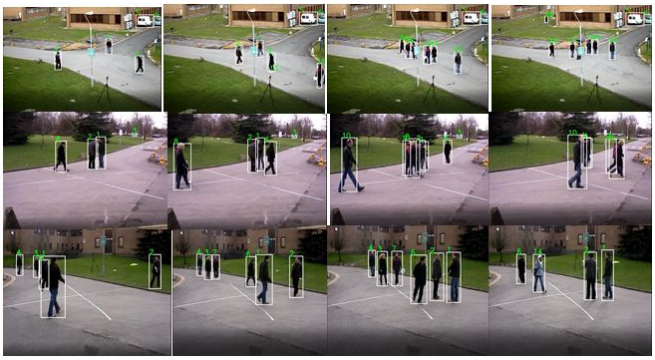

Fig. 10. Demonstration of ID switching issue

Table VII Improvement of ID switching results

Name	Type	ID switching #(only IOU)	ID switching #(with deep feature)	Improvement rate
video001	people	20	11	45%
video005	people	16	5	68.8%
video006	people	16	4	75%
video007	people	16	3	81.2%
video009	people	14	4	71.4%
Average improvement rate				68.2%

V. COMPARISON

This section shows two comparisons of state-of-the-art method. In the first part, we show the counting results and performance. In the second part, we show the ID switching results.

The first part shows a comparison between the proposed system and some previous works about object counting as illustrated in Table VIII. Cetinkaya et al. [16] used motion history of the detected faces to identify the same face in different frames. They believe that if the positions of the same detected face could be quite close among different frames, we can say they are quite likely to be in the same connected component in motion history. But the problem is that their matching step takes only the positions into account. In complicated scenarios, the accuracy will drop. Yoshinaga et al. [17] extracted blob features of people and place them into a trained neural network to estimate the number of the pedestrian. The counting accuracy is more than 80%. Using the neural network only to decide how many people appears inside the blob instead of detecting the position of every single person is a good idea to estimate the number of people. But it is also weak in analyzing the people flows because the system cannot give every person a unique identification and track them. Chang et al. [18] proposed a people counting method based on multiple cameras information fusion. Perng et al. [19] proposed a vision-based people counting system in buses, which adopts a top-view camera to capture the image.

The proposed method owns better counting accuracy and achieves real-time performance in high resolution. If we only calculate the tracking time, it can achieve 400 fps (without deep feature).

The second part shows the comparisons between our method and some state-of-the-art methods. The results are shown in Table IX, we can find that our method with deep feature performs the best, and our frame rate can reach 10fps, this method can handle some complex scenes. If the scene is simple, we also provide a very fast method; its frame rate can achieve 40fps.

Table VIII comparison in terms of counting accuracy

Design	Design [16]	Design [17]	Design [18]	Design [19]	Proposed method
Size	352x288	320x240	352x288	320x240	**720x480**
Clock	2.4GHz	3.2GHz	2.8GHz	2.5GHz	3.3GHz
Memory	3GB	2GB	512MB	n/a	8GB
Speed	15fps	10fps	5fps	66fps	**40fps**
Accuracy	83.74%	80%	87.4%	87%	**93%**

Table IX Comparison in terms of ID switching

Method	Design [5]	Design [20]	Proposed method without deep feature	Proposed method with deep feature
Size	720x480	720x480	720x480	720x480
ID switching #	53	68	82	**27**
Speed	1fps	30fps	**40fps**	10fps

VI. CONCLUSIONS

We have proposed a real-time and online tracking method with low power consumption for multiple types of object. Our method effectively solves the ID switching problem. We provide a lightweight a deep learning model for tracking and feedback to detection. The proposed method can improve 68.2% on average. The overall counting accuracy can reach 93% and the system performance can achieve 720x480@40fps or 720x480@10fps with deep feature on Nvidia 1080Ti. The feedback method can improve 2.5% mAP in detection accuracy. The next target will focus on cross camera tracking.

ACKNOWLEDGMENT

We appreciate the partially support from the “Center for mmWave Smart Radar Systems and Technologies” under the ‘Featured Areas Research Center Program’ within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE), Taiwan, R.O.C., and partially supported under MOST projects with grants MOST 108-3017-F-009-001 and MOST 108-2634-F-009-008 through Pervasive Artificial Intelligence Research Labs (PAIR Labs) in Taiwan, R.O.C. as well as the partial support from Advantech.

REFERENCES

[1] E. Rosten, R. Porter, and T. Drummond, “Fusing points and lines for high performance tracking,” *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2005.

[2] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Proc. International Journal of Computer Vision*, pp 91–110, 2004.

[3] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” *Proc. European Conference on Computer Vision (ECCV)*, 2006.

[4] M. Calonder, V. Lepetit, C. Strecha, P. Fua, “BRIEF: Binary robust independent elementary features,” *Proc. European Conference on Computer Vision (ECCV)*, 2010.

[5] E. Rublee, V. Rabaud, K., Konolige and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2011.

[6] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” *Proc. IEEE International Conference on Computer Vision*, 2011.

[7] A. Alexandre, R. Ortiz, and P. Vandergheynst, “Freak: Fast retina keypoint,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[8] D. Merad, K.-E. Aziz, R. Iguernaissi, B. Fertil, and P. Drap, "Tracking multiple persons under partial and global occlusions: Application to customers' behavior analysis," *Pattern Recognition Letters (PRL)*, vol. 81, pp. 11-20, 2016.

[9] F. Bousetouane, L. Dib and H. Snoussi, "Improved mean shift integrating texture and color features for robust real time object tracking," *The Visual Computer*, vol. 29, pp. 155-170, 2013.

[10] A. Andriyenko and K. Schindler, “Multi-target tracking by continuous energy minimization,” *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[11] B. Yang and R. Nevatia, “Online learned discriminative partbased appearance models for multi-human tracking,” *Proc. European Conference on Computer Vision (ECCV), 2012.*

[12] A. R. Zamir, A. Dehghan, and M. Shah, “GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs,” *Proc. European Conference on Computer Vision (ECCV), 2012.*

[13] S. Hong, B. Roh, K.-H. Kim, Y. Cheon, and M. Park, "PVANet: Lightweight Deep Neural Networks for Real-time Object Detection," *Proc. 1st International Workshop on Efficient Methods for Deep Neural Networks (EMDNN)*, 2016.

[14] S. Tang, M. Andriluka, and A. Milan, "Learning People Detectors for Tracking in Crowded Scenes," *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2013.

[15] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-Continuous Optimization for Multi-Target Tracking," *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[16] H. Cetinkaya and M. Akcay, "People Counting at Campuses," Proc. 4th *World Conference on Educational Technology Researches (WCETR)*, 2014.

[17] S. Yoshinaga, A. Shimada, and R. Taniguchi, "Real-time people counting using blob descriptor," *Procedia Social and Behavioral Sciences, (PSBS)*, vol. 2, pp. 143-152, 2010.

[18] Q. Chang, Z. Song, R. Shi, and J. Xu, “A People Counting Method based on Multiple Cameras Information Fusion,” *Proc. IEEE International Conference on Systems, Man and Cybernetics, (ICSMC)*, 2015.

[19] J.-W. Perng, T.Y. Wang, Y.-W. Hsu and B.-F. Wu, “The Design and Implementation of a Vision-based People Counting System in Buses,” *Proc. International Conference on System Science and Engineering (ICSSE)*, 2016.

[20] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. “Visual object tracking using adaptive correlation filters,” *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[21] M. Brown, S. Winder, and R. Szeliski. “Multi-image matching using multi-scale oriented patches,” *Computer Vision and Pattern Recognition, (CVPR)*, pages 510–517, 2005.

[22] A. Weimert, X. Tan, and X. Yang. “Natural feature detection on mobile phones with 3D FAST,” *Int. J. of Virtual Reality (IJVR)*, 9:29–34, 2010.

[23] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. “Imagenet large scale visual recognition challenge,” arXiv:1409.0575, 2014

[25] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In CVPR, 2015.

[26] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4, 7

[27] S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Neural Information Processing Systems (NIPS)*, 2015.