2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Learn to Detect Objects Incrementally

Linting Guan[1,2], Yan Wu*[1], Junqiao Zhao[1] and Chen Ye[1]

***Abstract*— Intelligent vehicles need to detect new classes of traffic objects while keeping the performance of old ones. Deep convolution neural network (DCNN) based detector has shown superior performance, however, DCNN is ill-equipped for incremental learning, i.e., a DCNN based vehicle detector trained on traffic sign dataset will catastrophic forget how to detect vehicles. In this paper, we propose a novel method to alleviate this problem, our key insight is that the original class of objects also appears in new task data, by utilizing these objects, our method effectively keeps the detection accuracy of original models while incremental learning to detect new classes of objects. Detailed experiments on PASCAL VOC dataset and TSD-max database verified the effectiveness of our method.**

I. Introduction

Intelligent vehicles need real-time perception of the surrounding environment. The perception module needs to detect multiple classes of objects simultaneously, such as vehicles, pedestrians and traffic signs. Deep convolution neural network (DCNN) has made great progress in the field of object detection, while training a DCNN based object detector needs a large amount of annotation data. In reality, these objects are labeled separately in multiple public datasets, e.g., KITTI dataset [1] annotated vehicles and pedestrian, Caltech Pedestrian dataset [2] annotated pedestrians and TSD-Signal dataset [3] annotated traffic signs.

DCNN based models cannot learn to detect them by incremental training across multiple datasets because it will trigger catastrophic forgetting problems [4]. Moreover, intelligent vehicles require detecting new classes of objects when faced with a more complex driving environment. For example, after learned to detect pedestrians and vehicles, intelligent vehicles need to detect new classes of objects, such as traffic lights, traffic signs and lane lines, and even the classes of objects that rarely seen, such as animals or drop boxes on the road and so on. However, current DCNN based object detector did not support learning to detect new classes of objects incrementally.

There are usually two ways to solve this problem. The first way is training a new deep neural network to detect new classes of objects according to the annotated dataset. The advantage of this method is that we can use the existing object detection model. However, this need to run multiple deep neural networks at the same time. These detection models repeat similar convolution calculation on the input data, lead to tremendous waste of computing resources. Moreover, the hardware resource on an intelligent vehicle is very limited. It is hard to meet the requirements of running multiple neural network models.

This work is supported by the National Natural Science Foundation of China (No. U1764261), the Natural Science Foundation of Shanghai (No.kz170020173571), the Fundamental Research Funds for the Central Universities (No. 22120170232) and the Zhejiang Provincial Department of Education Project(No.Y201432357).

*Corresponding author

[1]Linting Guan,Yan Wu, Junqiao Zhao and Chen Ye are with College of Electronics & Information Engineering, Tongji University, Shanghai, 201804, China glinting, yanwu, zhaojunqiao, yechen@tongji.edu.cn

[2]Linting Guan is with Faculty of School of Mathematics, Physics, and Information Science, Zhejiang Ocean University, Zhoushan, Zhejiang 316004, PR China

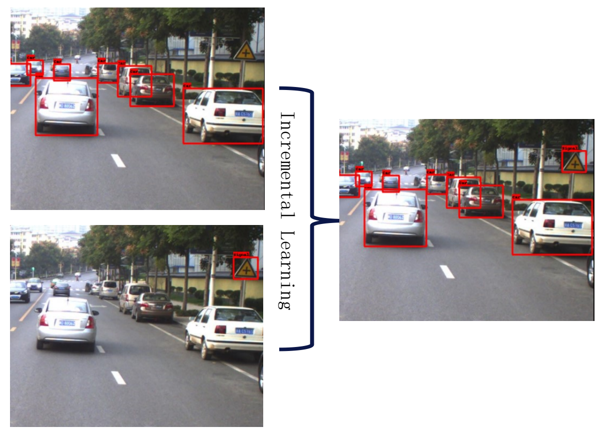

Fig. 1. Illustration of learning to detect object incrementally, firstly, a DCNN based model learned to detect vehicles, then the learned to detect traffic signs incrementally, finally, the detector has the ability of detecting both vehicles and traffic signs.

The second way is to design a unified DCNN detector to share multi-layer convolution operations and detect many classes of objects at the same time. However, it requires that all classes of objects should be annotated in the same dataset in order to train a DCNN model to detect them simultaneously, e.g., we need to label all the objects of desired classes such as vehicles, pedestrians, traffic signs in the training dataset. When DCNN model need to detect new classes of objects, it requires a lot of human resources to re-label the existing datasets, so it is costly for the DCNN based detector to continually learn to detect new classes of objects with this method.

In this paper, we propose a novel method enabling the neural network model incremental learning to detect new classes of objects and avoid catastrophic forgetting problem. The core of our method is using the objects of original classes existed in new task data. With hundreds of original class objects in the new task data, we can effectively keep the detection accuracy of original classes. Fig. 1 illustrates the purpose of our method.

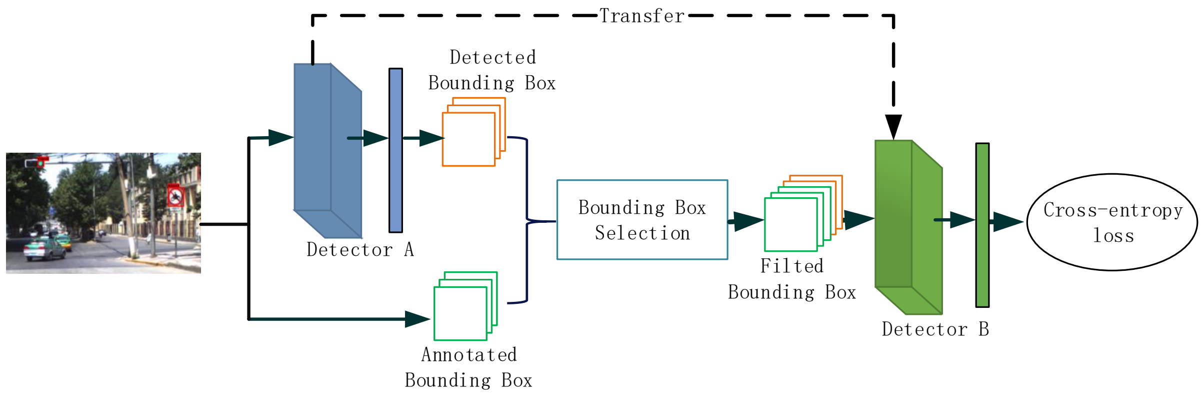

Fig. 2. Overview of our method for learning to detect objects incrementally. We export the knowledge of original model (Detector A) by detecting objects on new task data, then using the detection results to force the model not forgetting previous knowledge, while learning to detect new classes of objects. We further improved the performance by transferring the convolution layers of the original model to the next model (Detector B), see text for more details.

II. Related Work

A. Object Detection Algorithms

In recent years, deep convolution neural network has developed rapidly in the field of object detection. In 2013, Sermanet *et al.* [5] proposed a multi-scale sliding window algorithm using convolution neural network (CNN), which first used deep convolution neural network for object detection. Ross Girshick *et al.* [6]proposed a method (R-CNN) used the region proposal algorithm [7] to extract regions containing possible targets, and used CNN to extract features at each region, finally classified the regions using support vector machines. This method achieved a 50% performance improvement on PASCAL VOC dataset over other methods. Fast R-CNN [8] improved R-CNN using CNN to extract features on the whole image, and then used region interest pooling (RoI) on the feature map, finally classified and regressed bounding boxes using neural networks, which speed up the algorithm by reusing convolution operation on images. Faster R-CNN [9] added the Region Proposal Network (RPN), allowing the model to be fully end-to-end training.

YOLO [10] and YOLOv2 [11] integrated bounding box location prediction and class prediction into a single neural network model to achieve fast object detection with high accuracy and speed. Single Shot Detector (SSD) [12] achieved better detection accuracy by using multi-scale convolution features. Variants of Faster R-CNN models [13], [14] and SSD models [15] further improved the detection accuracy. In this paper, we focused on incremental learning to detect new classes of objects. Our proposed method enables object detection models to have the ability of incremental learning.

B. Incremental Learning Algorithm

Although DCNN based models have achieved great success in the field of image perception, incremental learning remains a huge challenge for it. McCloskey *et al.* [16] described the phenomenon that training neural networks incrementally with new data will lead to overwriting its knowledge of previous data acquisition as the catastrophic forgetting problem. Researchers have tried to alleviate this problem with various ways [17], [18], but the achievements are limited. Essentially, the catastrophic forgetting problems in neural networks face stability/plasticity dilemma [19]: neural network models need enough plasticity to learn a new task, but increasing plasticity of neural network leads to forgetting features that have been learned on previous tasks. Kemker *et al.* [20] divided the methods of previous research into two categories. The first category is to keep old features by regularization [21]. The second category is combining the old and new tasks data for training to prevent the old features from being forgotten.

A representative recent achievement of the first category is the elastic weight consolidation (EWC) algorithm [22] and the following works [23], [24]. EWC calculates fisher information from the previously trained model and use this information as a constraint that reduces the plasticity of the important weights for those previous tasks. However, this method performs not very good in object detection [25]. The second class methods prevent the agent to continually learn new tasks throughout its life cycle due to the need of storing all previously trained data. Sarwar *et al.* [26] increase convolutional kernels in the last layer of original network to raise the network capacity for incremental learning. Li *et al.* [27] propose to keep the model response for old tasks while learning a new task, but their approach is limited to classification problem. We focus on incremental learning of object detection, our proposed method combine the data of detected old classes of objects and annotated new classes of objects in the new task for training, it belongs to the second class but does not need to store previously trained data.

C. Object Detection Algorithm Based on Incremental Learning

Our method tries to solve the catastrophic forgetting problem while incremental learning to detect new classes of objects with DCNN. The most related work with ours is from Shmelkov *et al.* [25], which used a loss function balancing the interplay between predictions on new classes and a distillation loss minimizing the discrepancy between responses for old classes from the original and the new networks. Specifically, they used the EdgeBox [28] to propose RoIs from the data of the new task and randomly sampled 64 RoIs out of 128 object frames with the smallest background score. Then, a loss function was designed to make the new model keep the similar classification results of these proposals with the original model. Essentially, this method converts the incremental learning of object detection problem to classification problem, it is restricted to use category-agnostic proposals because it needs to get the same RoI candidates from each image and try to keep the same classification results with original model. But category-agnostic proposals are inefficient [29], so this method cannot be used in scenes requiring real-time detect speed, such as autopilot environment. We take advantage of the fact that learned classes of objects are also existed in new task data. We use original model detect the new task data and combining the detection objects and labeled objects for incremental training the detector. Our proposed method does not use class-independent region proposal method, so it is suitable for high-performance end-to-end object detection models.

III. Method

Our approach for incremental learning of object detection model is illustrated in Fig. 2. We keep the detection accuracy of original detector by two steps. The first step is exporting the knowledge of original detector by getting the detection results from the new task data. As the experiment shows that with hundreds of original classes of objects existed in the new task data, we can effectively prevent the model from catastrophic forgetting of learned knowledge. This requirement is satisfied in traffic environment, i.e., we can collect a lot of data containing learned classes of objects, like vehicles and new classes of objects, such as traffic signals. These data can be used to train the model incremental learning to detect new classes of objects. The second step is transferring the parameter of the original model to the new model, it is important to incremental learning to detect new classes of objects based on the original model. In the remainder of this section, we provide details of the object detection network and the learning algorithm.

A. Object Detection Network

We choose YOLOv2 [11] as the base object detection network, which is one-stage object detection model with anchor boxes to predict objects classes and locations on the top of convolution features. We chose this model instead of two-stage models like the variants of Faster R-CNN models [13], [14], because YOLOv2 can run in real time with less computing resources while retaining the state of the art performance.

B. Learning Algorithm

Our key finding is with hundreds of original classes of objects, we can effectively keep the accuracy of original detector when incremental learning to detect new classes of objects. We achieve this by detecting the objects of old classes from new task data with the original neural network (Detector A). However, the original model will detect many incorrectly classified objects when there are exists similar classes of objects in the first and second stage data. In this condition, the second stage classes of objects will be misclassified as the first stage classes. Fig. 3 shows several misclassified examples. The detector was trained on the first ten classes of data in PASCAL VOC dataset [30] in alphabetical order, then detected on the data of last ten classes. The model recognized many objects of last ten classes as first ten classes, such as recognized dogs and sheep as cows, recognized a man with a textured hat as a bird and recognized a sofa as a chair.

As we can see from Fig. 3 that the locations of these objects are accurate, but the class of object is wrong, so the incorrectly recognized objects will have high intersection over union (IoU) with the annotated objects. We solve this problem by computing the IoU of bounding boxes from detected objects with the bounding boxes of annotated objects in the second stage, and remove the detected objects which have high IoU with the annotated objects of second stage classes. These remaining objects are then combined with the annotated objects for each image in second stage training dataset. The detailed procedure of our approach is described in Algorithm1.

Algorithm 1 Incremental learning of object detection algorithm

Input: Training Set X, Ground Truth Y, Original Detection Network N_A, IoU Threshold h_i, Object Probability Threshold h_p

Output: Object Detection Network N_B

```
for each x in X do
    Ŷ_x ← N_A(x, h_p) //get original classes of objects with probability big than h_p in new task data x with original detector N_A
    Y_x ← subset(Y, x) //get annotated objects in x from Y
    for each ŷ_x in Ŷ_x do
        for each y_x in Y_n do
            if IoU(ŷ_x, y_x) < h_i then
                Y_x ← Y_x ⋃ ŷ_x
            end if
        end for
    end for
    Y ← Y ⋃ Y_x
end for
Initialize N_B with the parameter of N_A
Train N_B with dataset Y
return N_B
```

IV. Experiments

We validated our algorithm on the PASCAL VOC dataset [30] and traffic scenario TSD-max dataset [3] respectively.

TABLE I
VOC 2007 TEST PER-CLASS AVERAGE PRECISION (%) UNDER DIFFERENT SETTINGS.

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP(1-10)	mAP
$A(20)$	76.1	82.3	74.0	64.5	48.6	81.2	78.9	87.0	57.2	78.1	75.9	84.9	85.7	82.2	75.2	50.4	73.1	78.7	84.0	75.0	72.8	74.6
$A(1-10)$	77.6	82.9	71.7	65.2	50.1	77.8	82.9	80.2	56.6	57.8	-	-	-	-	-	-	-	-	-	-	70.3	70.3
$B(11-20)$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	68.0	69.5	78.7	78.2	70.4	41.9	61.1	66.5	78.2	69.7	0.0	34.1
$B_I(11-20)$	66.3	79.2	49.4	53.6	44.5	79.6	78.5	79.7	54.1	62.5	75.8	74.6	82.4	82.0	74.9	48.5	63.1	74.5	80.7	72.8	64.8	68.8

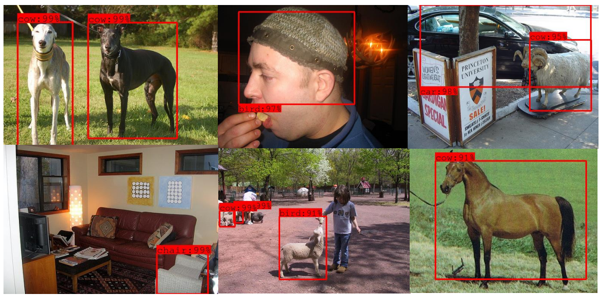

Fig. 3. Illustration of the detection results from the DCNN based model trained with the first ten classes of PASCAL VOC dataset, and detect on the last ten classes of data. These detected objects are located correctly but wrongly classified.

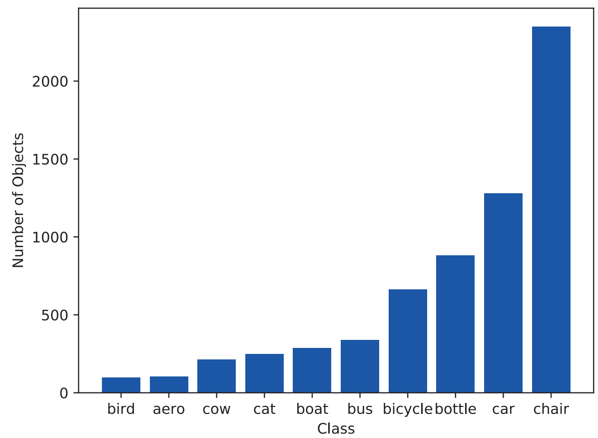

Fig. 4. The number of first ten classes of objects detected on VOC last ten classes of images.

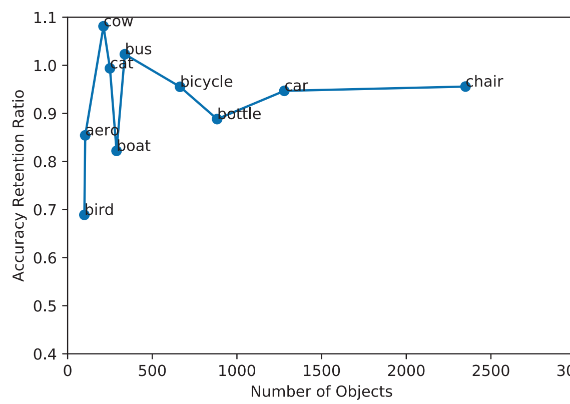

Fig. 5. The accuracy retention ratio relative to the number of objects.

We use YOLOv2 as the detection model, set the input resolution to 416 x 416. We train the network using SGD and Nesterov momentum [31], set the momentum to 0.9, set the learning rate of the first 25K iterations to 0.001, then decay to 0.0001 in following 10K iterations, and decay to 0.00001 in the last 10K iterations. In the incremental stage of training to detect new classes of objects, we continued to train model 25K iterations using learning rate of 0.001, and decay to 0.0001 for next 10K iterations. All the experiments were performed using a Maxwell-based NVIDIA TITAN X GPU.

A. Experiment on PASCAL VOC Dataset

In the first experiment, we check our method on the PASCAL VOC dataset, which is a benchmark in visual object category recognition and detection. The VOC2007 and VOC2012 dataset consist of annotated consumer photographs collected from the Flickr. We use a total of 16K images from the PASCAL VOC2007, VOC2012 training, and validation sets as training dataset and 5K VOC2007 test split as the test dataset. We use the mean average precision (mAP) at 0.5 IoU threshold as the evaluation metric. Tab. I shows the result, where $A(1-20)$ is the model trained with all 20 classes of VOC train-val dataset, $A(1-10)$ trained with the first ten classes (in alphabetical order) in the subset of VOC train-val dataset. In the second stage, we continue training $B(11-20)$ with the last ten classes of data in the VOC dataset based on the model of $A(1-10)$. The mAP of the model for the first ten objects dropped from 70.3% to 0%, because the objects of first ten classes not be labeled in the last ten classes of data, so the learning to detect new classes of objects on the last ten data lead to catastrophic forgetting of learned classes. Model $B_I(11-20)$ is our proposed method, we set both object probability threshold h_p and IoU threshold h_i to 0.5, the total mAP keep 68.8%, the mAP of the first ten classes keep 64.8% ,so there is a large improvement compared to 0.0% of $B(11-20)$, this shows that our method can effectively retain previously learned knowledge.

Fig. 4 shows the number of first ten classes of objects detected on VOC last ten classes of images varies greatly between each class. In order to analyze the influence of the detected object number on the detection accuracy in each class, we propose a new metric named accuracy retention ratio (ARR), which is defined as:

$$r_c = \frac{P_c^I}{P_c^O}$$

where r_c is the ARR of class c, P_c^I is the accuracy of incremental detection model I on class c, and P_c^O is the

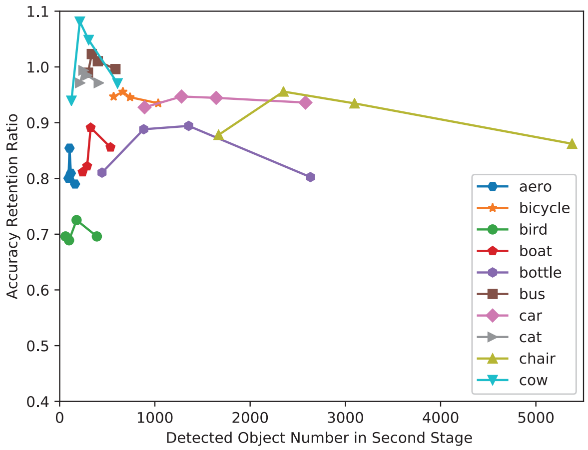

Fig. 6. Accuracy retention ratio of each class with the number of detected objects by setting h_p to 0.1, 0.3, 0.5 and 0.7.

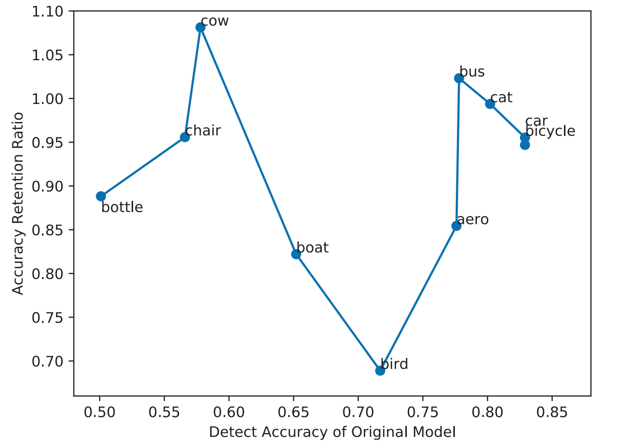

Fig. 7. Accuracy retention ratio relative to the detect precision of original model.

Fig. 8. Examples of false detected birds, the original model detects part of other objects as birds.

accuracy of original detection model O on class c. In our experiments, we compute the ARR values with:

$$r_c = \frac{P_c^{B_I(11-20)}}{P_c^{A(1-10)}}$$

we can see from the Fig. 5 that when the detected object number is small, the ARR increases with the number of detected objects, after there are hundreds of detected objects, the detection accuracy of the original model can be effectively preserved, so continue to increase the number of detected objects does not significantly improves ARR. We also examined the effect of different number of objects on each class, from second stage data by setting h_p to 0.1, 0.3, 0.5 and 0.7 respectively. Fig. 6 shows the ARR of each class with different number of objects, we found that the middle level of h_p which balances the number of objects and the detection accuracy get better results in all classes.

We also check the relationship of ARR and the detection accuracy of the original model. Fig. 7 shows that the ARR in the second stage does not increase with the detection accuracy of the original model. Our method keeps high ARR for the most classes regardless of the accuracy of the original model, it means that the classes of ARR are not strongly related to the detection accuracy of original model. We argue that ARR is more affected by other classes objects, i.e., the detection accuracy of cow increased from 57.8% to 62.5% in the second stage. This is because there are many animals such as dogs and houses are mistakenly identified as cows in first stage as shown in Fig. 3, but these objects have been annotated in the second stage data. Our proposed method filtered these wrong detection results, so the model improved the detection accuracy of cow by learning to correctly identify these objects in the second stage data. Another example is the accuracy of bird dropped from 71.7% to 49.4%, the main reason is that some objects such as watering pots or part of plants in the second stage look like birds, so the original detection model is easy to detect these objects as birds. But there are no annotation information for these objects, so these objects cannot be filtered out by the IoU between the detection object and these object. These wrong detected objects strongly interfere the detection accuracy of birds in the second stage. Fig. 8 visualizes examples of false detected birds.

In summary, we argue that by keeping hundreds of objects with each original class in the new task data, our proposed method can effectively keep the detection precision of original model. Experiments show that our method can effectively keep the knowledge of original model by utilizing the objects of original classes in the new task dataset.

B. Experiment on TSD-max Dataset

In the second experiment, we check our method on the TSD-max dataset [3], a dataset annotated traffic element attributes using vehicle-mounted road video provided by Artificial Intelligence and Robotics Research Institute, Xi'an Jiaotong University. We chose the traffic signal dataset named TSD-Signal and front vehicle dataset named TSD-Vehicle, the off-line subset of TSD-Signal contains 3174 images annotated with traffic signals, the off-line subset of TSD-Vehicle contains 7872 images and annotate with vehicles, these images are collected from the traffic road in China. Besides traffic signals, TSD-Signal dataset also contains many unlabeled vehicles. We trained a neural network model to detect vehicle and traffic signals simultaneously with both datasets using our proposed method.

In TSD-Signal dataset, there are only a few objects in each class, so the object number of each single class cannot meet

Fig. 9. Illustration of the detection results on TSD-Signal validation split, the detector is first trained on the TSD-Vehicle and then on TSD-Signal training split with our proposed method, finally, the detector learned to simultaneously detect both vehicles and traffic signs.

the needs of deep neural network training. Thus, we simplify the task by converting all traffic signals into one class, and treat the vehicles as another class. We set the network input resolution to 640 x 640 due to the high precision of original image (1280 x 1024) and the small area of traffic signal(less than 50 x 50). We first train the vehicle detection model on the TSD-Vehicle dataset and then use this model to detect vehicles in the TSD-Signal dataset. The detection results are integrated with the annotated traffic signals in TSD-Signal dataset, the fused annotation data is split into training set and testing set. For there are no ground truth labels for vehicles in TSD-Signal dataset, we only visualize the detection results of the trained model on the test dataset. As shown in Fig. 9, the model learned to simultaneously detect both vehicles and traffic signs on TSD-Vehicle dataset and TSD-Signal dataset using our proposed method.

V. Conclusion

In this paper, we presented a method for incremental learning to detect new classes of objects without the need to access the training set of previous tasks. We alleviate the problem of catastrophic forgetting by detecting learned objects on the new task data. Extensive experiments on PASCAL VOC dataset and TSD-max dataset verify the effectiveness of our method. Our method can be combined with light-weight real-time object detection methods to incremental learning to detect new classes of objects.

References

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[2] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 304–311.

[3] Tsd-max traffic scene video dataset. [Online]. Available: http://trafficdata.xjtu.edu.cn/index.do

[4] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[5] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *International Conference on Learning Representations (ICLR)*, 2014.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[7] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[8] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

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

[10] 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*, 2016, pp. 779–788.

[11] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *arXiv preprint arXiv:1708.02002*, 2017.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *arXiv preprint arXiv:1703.06870*, 2017.

[15] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[16] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of learning and motivation*, vol. 24, pp. 109–165, 1989.

[17] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[18] B. Goodrich and I. Arel, "Unsupervised neuron selection for mitigating catastrophic forgetting in neural networks," in *Circuits and Systems (MWSCAS), 2014 IEEE 57th International Midwest Symposium on*. IEEE, 2014, pp. 997–1000.

[19] W. C. Abraham and A. Robins, "Memory retention–the synaptic stability versus plasticity dilemma," *Trends in neurosciences*, vol. 28, no. 2, pp. 73–78, 2005.

[20] R. Kemker, A. Abitino, M. McClure, and C. Kanan, "Measuring catastrophic forgetting in neural networks," *arXiv preprint arXiv:1708.02072*, 2017.

[21] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," *arXiv preprint arXiv:1701.08734*, 2017.

[22] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, p. 201611835, 2017.

[23] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Advances in Neural Information Processing Systems*, 2017, pp. 4655–4665.

[24] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*, 2017, pp. 3987–3995.

[25] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," *arXiv preprint arXiv:1708.06977*, 2017.

[26] S. S. Sarwar, A. Ankit, and K. Roy, "Incremental learning in deep convolutional neural networks using partial network sharing," *arXiv preprint arXiv:1712.02719*, 2017.

[27] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[28] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*. Springer, 2014, pp. 391–405.

[29] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 814–830, 2016.

[30] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[31] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.