Image and Vision Computing 85 (2019) 14–25

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Deep convolution neural network with scene-centric and object-centric information for object detection☆

Zong-Ying Shen [a], Shiang-Yu Han [a], Li-Chen Fu [a,*], Pei-Yung Hsiao [b], Yo-Chung Lau [c], Sheng-Jen Chang [c]

[a] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, ROC
[b] Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung, Taiwan, ROC
[c] Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Taiwan, ROC

ARTICLE INFO

Article history:
Received 13 April 2018
Accepted 5 March 2019
Available online 21 March 2019

Keywords:
Deep learning
Convolutional neural networks
Real-time object detection
Scene information

ABSTRACT

In recent years, Deep Convolutional Neural Network (CNN) has shown an impressive performance on computer vision field. The ability of learning feature representations from large training dataset makes deep CNN outperform traditional hand-crafted features approaches on object classification and detection. However, computations for deep CNN models are time consuming due to their high complexity, which makes it hardly applicable to real world application, such as Advance Driver Assistance System (ADAS). To reduce the computation complexity, several fast object detection frameworks in the literature have been proposed, such as SSD and YOLO. Although these kinds of method can run at real-time, they usually struggle with dealing of small objects due to the difficulty of handling smaller input image size. Based on our observation, we propose a novel object detection framework which combines the feature representations learned from object-centric and scene-centric datasets with an aim to improve the accuracy on detecting especially small objects. The experimental results on MSCOCO dataset show that our method can actually improve the detection accuracy of small objects, which leads to better overall results. We also evaluate our method on PASCAL VOC 2012 datasets, and the results show that our method not only can achieve state-of-the-art accuracy but also most importantly presents in real-time.

1. Introduction

In recent years, deep CNNs have been shown as a powerful solution to resolve the detection problem mentioned above. It is noteworthy that the current state-of-the-art two-stage object detection approaches based on object proposal framework, such as RCNN [1] and Faster-RCNN [2], outperform traditional hand-crafted based methods due to their high performance on feature extraction. Also, many large image datasets, such as ImageNet [3], PASCAL VOC [4], and MSCOCO [5] datasets, have been published to help CNNs learn more object representations. Thus, it is not surprising that CNNs can show an impressive performance on object detection tasks. However, two-stage methods are complex and the involved models are hard to train, because they need to handle proposal generation, object classification, and localization within their framework. Also, for real-world tasks, the computation consumption of the system is a big concern. The overhead of generating object proposals makes the above-mentioned model hard to speed up which consequently makes it hard to be implemented on embedded system. To solve the issue caused by limited computation power, some CNN based one-stage detectors [6,7] are proposed. They reframe object detection as a single regression problem and predict the result by single network forward. Without the complexity of combining two networks, CNN based one shot detectors are more efficient and easier to train.

On the other hand, there are several studies which focus on scene recognition. Similar to the ImageNet, there are some other published large scene-centric image datasets, such as place [8] and sun [9,10], which allow us to train a deep CNN. Besides, many studies show that CNNs can also well perform on scene recognition tasks. Thus, we should have sufficient confidence that CNNs can also learn scene-centric information from large labeled scene image dataset.

As we have mentioned, most of state-of-the-art object detectors are based on the two-stage framework which can achieve high accuracy on several datasets. In addition to the accuracy, time consumption is also an important consideration for us when we would like to apply it to real life scenarios. Based on these factors, we decide to choose one-stage detector as our base framework, which allows our system to run at real time. Nevertheless, these kinds of methods are stuck with insufficient accuracy on small objects due to the lack of feature representations.

Because the Deep CNN has a large amount of parameters which need to be trained, the relevant approaches will not perform well when data is lacking. In order to extract robust feature representations, most of the

☆ This paper has been recommended for acceptance by Sinisa Todorovic.

* Corresponding author.
E-mail address: lichen@ntu.edu.tw (L.-C. Fu).

https://doi.org/10.1016/j.imavis.2019.03.004
0262-8856/

Fig. 1. Image samples from ImageNet dataset for "dog" categories.

state-of-the-art CNN object detectors, no matter if it is one-stage framework or two-stage detection framework, use model that pre-trained on ImageNet dataset as their base network and fine-tune for target data. However, feature representations learned from object-centric data (e.g., ImageNet) may not be sufficient for all objects in object detection tasks.

In this research, we propose a novel real-time object detection framework which can combine both object-centric and scene-centric information. To validate our work, we will evaluate it on MSCOCO and PASCAL datasets, which contain objects in general scenes. The experimental results show that feature representations extracted by the CNN trained on image-level scene-centric dataset can improve overall object detection performance, especially the small objects.

2. Related work

In this research, we propose a novel object detection network architecture which combines two stream CNNs, one for object-centric feature representations and the other for scene-centric ones, to improve the detection performance. In this section, we will first describe the evolution of CNNs and some state-of-the-art CNN based object detection framework. Then, we will also describe what CNN architecture can aid detection performance by learning scene-centric information.

2.1. Convolutional neural networks for object detection

Machine learning methods are widely used for solving computer vision problems. Traditional machine learning methods can solve simple recognition and detection tasks on small datasets. As the scale of dataset is growing quickly in recent years, a large annotated image dataset, called ImageNet [3], is published. ImageNet contains over 15 million images and each image is annotated with an image-level object category label. Due to the powerful learning capacity, CNNs start to become a popular approach to model this amount of data. The first deep CNN, called AlexNet, was proposed by Krizhevsky et al. [11] which outperformed traditional machine learning methods on ILSVRC-2010 [12] object classification benchmark. They implemented the CNN on the graphics processing unit (GPU) which allowed the network to be trained and tested efficiently. They also used a regularization method, called dropout, which forced the model to learn more robust features and helped to prevent overfitting.

Due to the success of AlexNet on image classification tasks, some researchers started to apply the CNNs to object detection system. Girshick et al. [1] proposed an object detection framework, called RCNN, which was the first CNN based object detector. The RCNN approach extracted high-level CNN feature from the Region of Interest(ROI) generated by selective search [13] method. Then, a multi-class support vector machine (SVM) [14] was trained on the CNN features to classify the object category of each object proposal. Although RCNN outperformed traditional machine learning methods on PASCAL VOC [4] dataset, it was quite computation consuming due to the overhead of applying CNN to each proposed region.

To speed up the overall testing time, several approaches are proposed to improve the detection framework. Fast RCNN is proposed by Girshick [15] which introduced the ROI pooling layer to pool the features in each ROI into a fixed dimension feature vector. With ROI pooling layer, Fast RCNN can reduce the overhead of computing CNN operation on the overlap of each region proposal. Furthermore, they also compress the fully connected layers with truncated Singular Value Decomposition (SVD) [16,17] to accelerate the network inference time. Due to these improvements, Fast RCNN achieves better results than RCNN with 10× faster runtime.

However, Fast RCNN still used selective search for generating the object proposal, which took about 2 s on the single core CPU. To reduce the extra computation consuming and use the CNN features more effectively, Faster RCNN [2] introduced Region Proposal Network (RPN) to combine the object proposal process into the CNN network. Benefited by the parallel computation power of GPU, Faster RCNN could find out object candidates more efficiently. Due to the high detection performance and "near" real-time speed, this two-stage framework was widely used by the later developed object detection methods. For example, Inside-Outside Net used 4-directional IRNN [18] for modeling the context of the feature maps. And, Hyper-Net [19] aggregated the feature maps generated by different convolution layers and combined them into more semantic features. Both of them were based on Faster RCNN framework and they all achieved leading results on PASCAL VOC and MSCOCO datasets.

Despite this, two-stage methods were still hard to achieve real time speed. Different from these methods, You Only Look Once (YOLO) [6] and Single Shot Detector (SSD) [7] were proposed. Both of them were proposal free one-stage framework, meaning that they could detect the objects by single network forwarding. They model bounding boxes prediction and object classification into a regression problem and trained a single convolutional neural network to solve it. Without region proposal and ROI pooling, these methods used the feature in each gird of the feature map for predicting the objects directly. In this thesis, our proposed object-scene net (OS-Net) architecture is inspired by the Single Shot Detector [7] framework, which allows us to detect the objects in real-time.

Fig. 2. Image samples from ImageNet dataset for "crosswalk" scene categories.

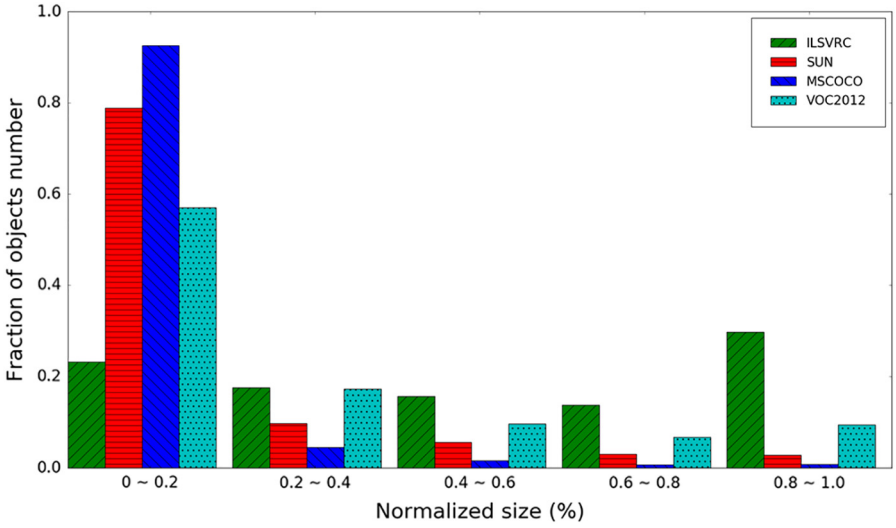


Fig. 3. Distribution of objects size.

2.2. Convolutional neural networks for scene recognition

Different from recognizing the object category of the image, there was some research focusing on using CNNs for recognizing the scene of the image. Similar to ImageNet object-centric dataset, places [8] was a scene-centric dataset which contained over 7 million images with their scene labels. Zhou et al. [8] showed that the approach of CNN trained on Places dataset could perform better than SVM trained on ImageNet-CNN feature for scene recognition. By applying visualization of the activation of each neuron, they realized that CNNs pre-trained on Places dataset could learn different feature representations as compared with ones pre-trained on ImageNet dataset.

To understand what CNNs could learn from the datasets, Zhou et al. [20] proposed a visualization technique called Class Activation Mapping (CAM) to figure out the discriminative regions where CNNs used for recognizing the object, scene or other high-level concepts. In their experiments, they showed that CNNs trained for scene recognition could localize the informative objects in the images. The capacity of discriminative localization of the CNN trained for scene recognition inspires us to adopt it for object detection tasks.

To adopt the well-trained ImageNet-CNN features to the scene recognition tasks, Herranz et al. [21] studied the differences between CNNs trained on ImageNet and those trained on Places dataset. They applied SVMs on the features extracted by ImageNet-CNN and Places-CNN separately to perform object classification. The experimental results showed that the SVM trained on ImageNet-CNN features could perform better than the one trained on Places-CNN when the background was removed and the object size was large. However, it is contrary when the background was included and the object was small, SVM plus Places-CNN features performed better. These results showed that there was a bias between object-centric dataset (e.g., ImageNet) and scene-centric dataset (e.g., Places).

Based on these analyses, Herranz et al. [21] proposed a multi-scale CNN architecture to combine the features learned from both ImageNet and Places datasets. In their architecture, ImageNet-CNN and Places-CNN were used for extracting the features from different scales of the input patches, and these features were further concatenated into a feature vector for scene classification. Their experimental results showed that the hybrid features could achieve the best performance on scene recognition. On the other hand, X. Zeng et al. [33] added neighboring support regions to improve classification accuracy. This concept showed the importance of contextual visual information in object detection, inspiring us to design an architecture by fusing the features of the object-centric CNN and scene-centric CNN to improve the object detection performance.

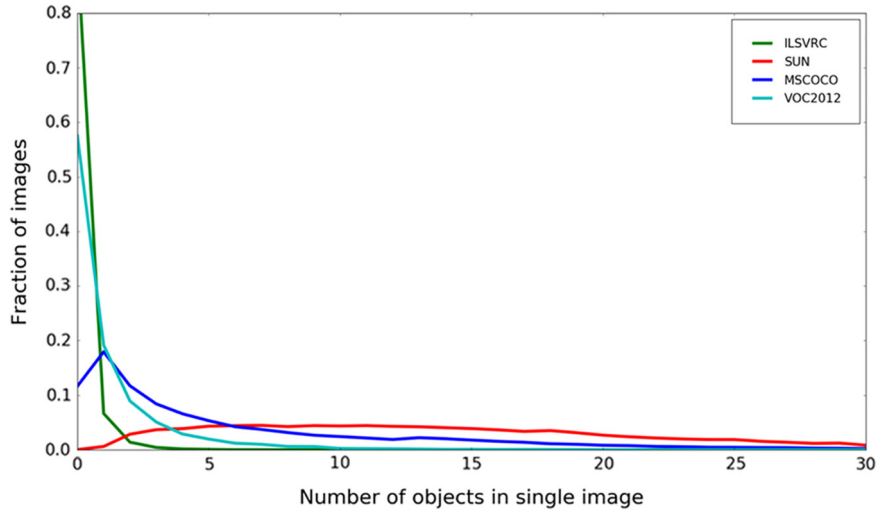


Fig. 4. Distribution of number of object per image.

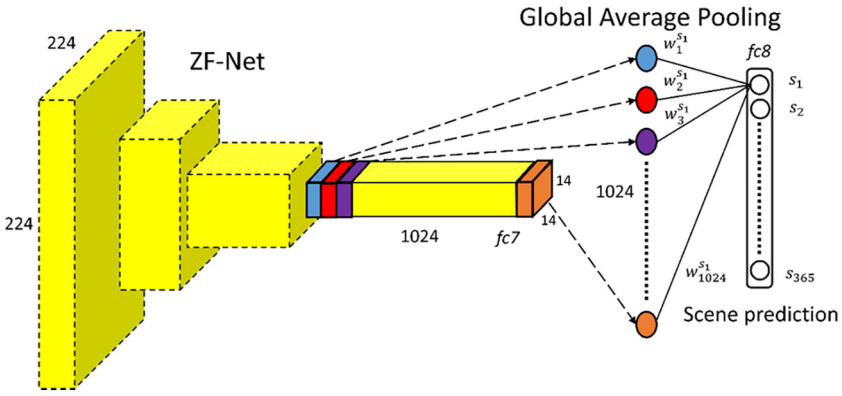


Fig. 5. CNN architecture for scene recognition.

3. Object-centric and scene-centric CNNs

Because combining the uses of datasets with different domain knowledge is our main strategy in this research, we will first introduce two different large image datasets, namely, ImageNet and Places. These dataset are widely used for pre-training a deep convolutional neural network. Then, we will go into the details of the bias between object-centric and scene-centric datasets. Moreover, we will show what our network has actually learned from the data via the visualization of the activation map.

3.1. Large weakly-supervised datasets

In 2009, ImageNet [3] dataset was introduced with over 15 million image-level labeled object images. They collected images from the Internet by querying the search engines and verified these images by humans. This dataset contains over 20 thousand categories, and the object category of the image was determined by the voting score on the Amazon Mechanical Turk (AMT) online platform. Nowadays, the subset of ImageNet with around 1000 images in 1000 object categories, called ILSVRC [22], is widely used for pre-training CNNs from scratch. Some sample images of ImageNet are shown in Fig. 1.

Besides the datasets for object recognition tasks, some datasets such as MIT Indoor67 [23] and SUN [9,10] are published with scene-centric labeled images which can be used for training a scene classifier. As the same purpose of ImageNet, a large scene-centric image dataset with more than 7 million images, called Places, was introduced by Zhou et al. [8]. Places dataset has two subsets, where one is Places205 with 205 scene categories, and the other is Place365. Some sample images of Places dataset are shown in Fig. 2.

3.2. Bias between objects and scenes data

Some recent research [11,24] showed that CNN could perform recognition tasks very well on object dataset (e.g., ImageNet) and scene dataset (e.g., Places) separately. To combine these two datasets, a classifier with single CNN, called Hybrid-CNN [8], was trained on union of these two dataset for classifying 1365 categories (e.g., 1000 object classes and 365 scene classes). However, the result showed that the integration could not improve the overall performance as expected.

Harranz et al. [21] showed that, instead of combining these two datasets and training it directly, the combination of ImageNet-CNNs and Places-CNN features could perform better on scene recognition tasks. They also showed that there was a bias between scene-centric and object-centric data. For example, the objects in ImageNet dataset were the subject of images; however, the images in Places dataset had more backgrounds. The objects in object-centric dataset were large and usually occupied the whole image, whereas the objects in scene-centric dataset were much smaller. Furthermore, they also showed that the object classifier trained with Places-CNN features had better performance on small object recognition than the one trained with ImageNet-CNN features. This was due to the fact that CNNs trained on scene-centric dataset could learn the relations between objects and the background which could help the predictor to discriminate the small objects from background. However, ImageNet-CNN outperformed Places-CNN when the object became larger, given the fact that the sizes of objects in ImageNet were similar to the targets size in this case. Based

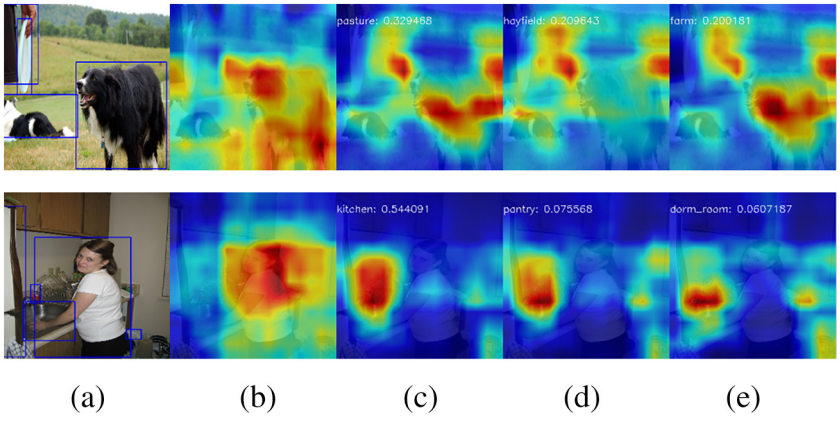


Fig. 6. Example of the regions where different CNNs concentrate on. (a) Input image sampled from MSCOCO dataset. Blue bounding boxes indicate the object annotation provided by MSCOCO dataset. (b) *fc7* activation map of Image-Net CNN. (c)–(e) show CAMs of Places-CNN generated from top-3 scene prediction.

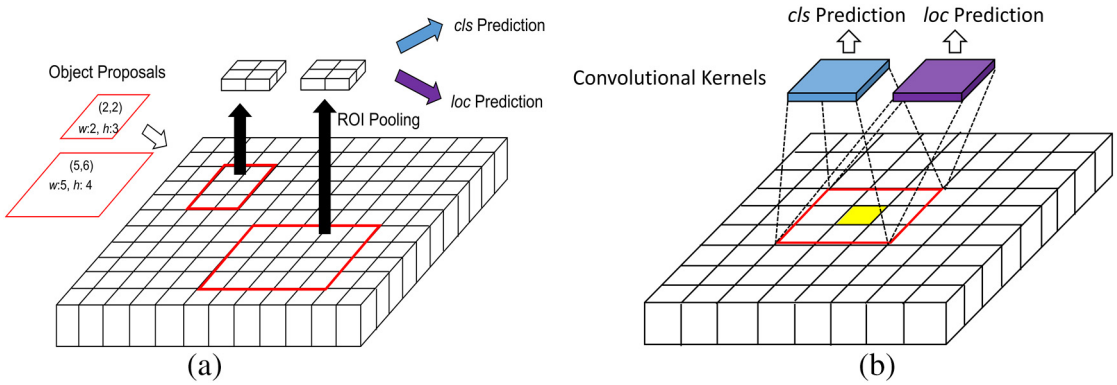


Fig. 7. Comparision of object proposal framework and proposal-free framework.

on these observations, they proposed a multi-scale CNN architecture to combine the CNNs respectively pre-trained on ImageNet dataset and Places dataset separately to perform the scene classification task, which could improve the performance significantly.

3.3. Object detection datasets

Besides image level datasets, there are several object detection purpose datasets which are published with fully annotated bounding boxes and category labels. The most popular two are MSCOCO dataset [5] and PASCAL VOC [4] dataset, which included 20 and 80 categories separately. These dataset are widely used for training and evaluating the performance of object detection models.

To understand the difference between object detection datasets and those dataset we mentioned above, we first analyze the object scales and density distributions of each dataset. Fortunately, SUN397 [10] and ILSVRC (subset of ImageNet) provide bounding box labels for part of the images. Herranz et al. [21] evaluated the bias between SUN397 scene dataset and ILSVRC object dataset. Here we compare the distributions of objects in MSCOCO and PASCAL 2012 dataset against those in SUN397 and ILSVRC, which are shown in Fig. 3.

Fig. 3 shows the distribution of normalized object sizes of the mentioned datasets. We compute the fraction of the object area with respect to the image size. We find that the distributions of object sizes in object detection datasets are much similar to the scene-centric dataset (e.g., SUN397). In the contrast, objects in object-centric dataset (e.g., ILSVRC) usually occupy the whole image. Based on this observation, we can expect that the CNNs pre-trained with object-centric and scene-centric datasets will learn different feature representations from these data. Note that MSCOCO dataset has more small objects than PASCAL VOC has, which shows that MSCOCO dataset is more challenging than PASCAL VOC dataset in general.

Fig. 4 shows the distribution of number of objects contained in a single image. We can find out that most of the images in ILSVRC object-centric dataset only contain single objects. However, images in object detection datasets (e.g., MSCOCO and PASCAL VOC 2012) may contain many small objects, which is more similar to the distribution of SUN scene-centric dataset.

3.4. ImageNet-CNN and place-CNN

As mentioned in the previous section, CNN trained on scene-centric datasets can learn different feature representations from the one trained on object-centric dataset. To explore what CNN actually learns from the data, we visualize the activation of the last few layers in the CNN. This method can demonstrate the region in which CNNs are interested.

To realize that, we first visualize the activation maps of fc7 of ImageNet-CNN directly. We take the well-trained fully convolutional ZF-net as the target, which is converted from original ZF-Net using àtrous algorithm [25]. Input images are resized to 256×256 and fed into the CNNs, and then we will obtain a $16 \times 16 \times 1024$ feature map from each CNN. Note that because we use fully convolutional network here, we can modify input size at will.

To generate human readable representations, we convert feature representations to a 2-dimensional heat map. The activations are accumulated across the channel direction to a single value so that we will

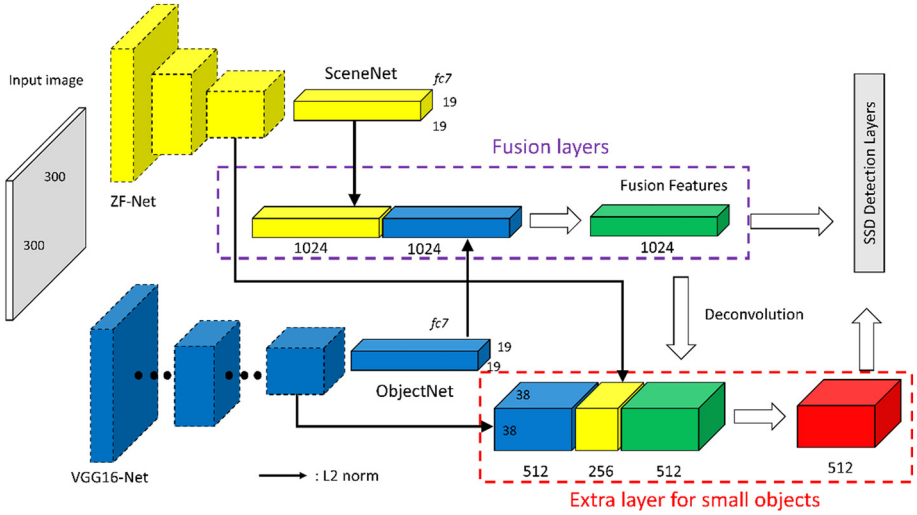


Fig. 8. Overview of OS-Net.

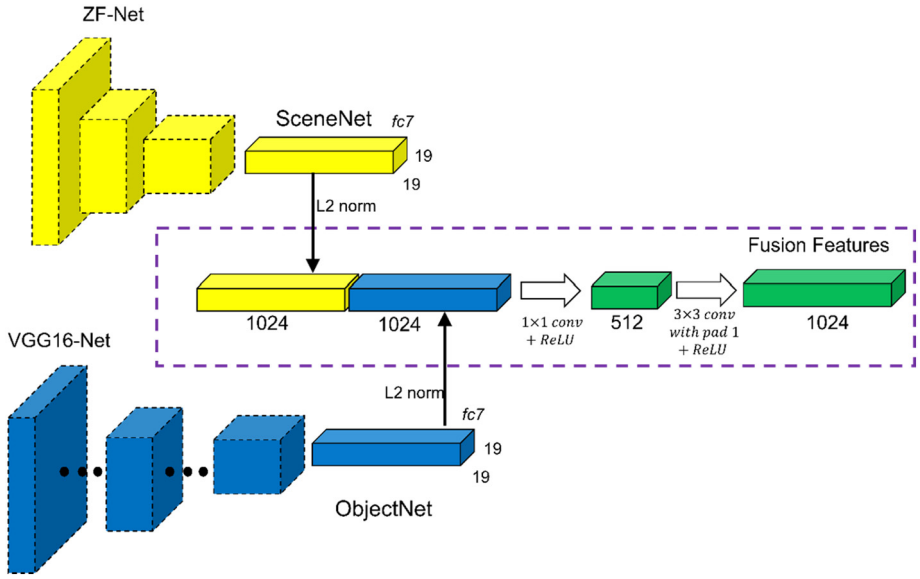


Fig. 9. Fusion layers.

obtain a $16 \times 16 \times 1$ map which summarizes the information of each spatial position. This process can be formulated as the following equation.

$$VisualMap(i,j) = \sum_{k=1}^{K} f_k(i,j) \qquad (\text{III}-1)$$

where $f_k(i,j)$ denotes the feature map of channel k, and i, j indicate the pixel position of the map. K is the total channel number, which is 1024 for *fc7* layer. The pixel-wise values are mapped to *rgb* color space, which are shown as Fig. 6(b).

Then, we train a model on Place365 dataset with the same CNN architecture to show what CNN can learn from different datasets. Instead of directly adding a fully connected layer to the top of *fc7* as the predictor, we concatenate a global average [20] pooling after the *fc7* as a regularizer. The architecture is shown in Fig. 5, where *fc8* has 365 neurons to predict the scene category. 224×224 image patches are randomly cropped from 256×256 images as training samples.

To visualize which discriminative regions are used by the CNN to identify the scenes, we apply Class Activation Mapping (CAM) [20] to the images. Because our main detection target is from MSCOCO dataset, which is the most challenging detection dataset nowadays, some images are sampled from it as the inputs. Note that the CNNs used here are not trained on any image in MSCOCO dataset. The visualization results are shown as Fig. 6.

Although the CNNs used here have not been trained on MSCOCO dataset, we still can observe that the visualization results have the ability to localize the objects in the image. Due to the fact that scenes are constructed by objects, CNN trained on scene-centric datasets can learn the relation between objects and background.

In comparison with the Image-Net CNN which can find the important object components, CNN fine-tuned with Places dataset can highlight the important objects in the images. For example, Fig. 6 shows that ImageNet-CNN concentrates on finding out the main components of the dog, such as head and legs, as the discriminative regions of this object. However, Places-CNN considers the dog as a single component as the discriminative region of the scene. Fig. 6 also shows that ImageNet-CNN can find out the human's head to identify the women. However, Places-CNN highlights the “sink”, which may be helpful to recognize the “kitchen” scene.

Based on this observation, we anticipate that object localization ability, especially for small objects, can be improved by combining a CNN which is trained on a large weakly supervised scene-centric dataset. To verify our anticipation, we propose a two stream CNN object detection framework to combine the feature representations, one focus on large objects and the other focus on small objects.

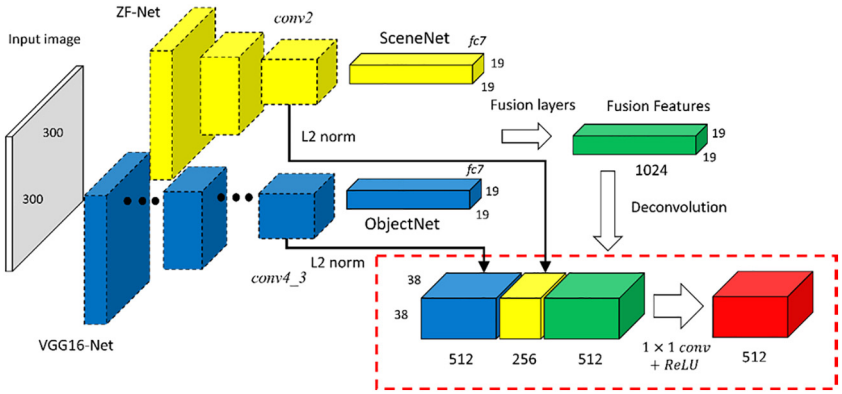


Fig. 10. Extra layers for small object detection.

Table 1
Evaluation metrics provided by MSCOCO evaluation server.

Average precision (AP)	
$AP^{0.5–0.95}$	Average AP at IoU = 0.5, 0.55, 0.6…0.0.95
$AP^{0.5}$	AP at IoU = 0.5
$AP^{0.75}$	AP at IoU = 0.75
AP across scales	
AP^{S}	AP for small objects: area < 32^2
AP^{M}	AP for medium objects: 32^2 < area < 96^2
AP^{L}	AP for large objects: area > 96^2
Average recall (AR) across scales	
AR^{S}	AR for small objects: area < 32^2
AR^{M}	AR for medium objects: 32^2 < area < 96^2
AR^{L}	AR for large objects: area > 96^2

4. Object-scene net (OS-Net)

In this section, we will first describe the difference between Faster-RCNN and SSD briefly. The reason why SSD may struggle with small objects will be pointed out. Then, we introduce our proposed OS-Net framework to improve the overall performance of Single-Shot Detector (SSD) [7], especially for small objects.

4.1. Difference between faster RCNN and SSD

Faster-RCNN uses ROI pooling for encoding the feature of each region proposal into fixed size feature vector, which allows the classifier to utilize full features in each ROI (shown as Fig. 7(a)). Without using object proposal and ROI pooling, proposal-free one-stage methods (e.g., SSD [7] and YOLO [6]) only use the feature representations in each feature map grid to predict the objects. Also, region proposal based methods use a relatively large input image (about ~1000 × 600) as the input of the CNN. However, one-stage methods usually use a small input image size (300 × 300 for SSD and 416 × 416 for YOLO) to reduce the computation complexity. These differences are shown in Fig. 7. These strategies lead to the fact that the feature representations of each grid in the feature maps become more important.

4.2. OS-Net overview

Herranz et al. [21] showed that using a single CNN model as a generic feature extractor could not solve every visual problem. As our analysist in the previous section, the feature representations learned from scene-centric dataset can provide different information from the one pre-trained on object-centric dataset. As a consequence, we suggest that finding a proper combination of these two CNNs is a critical way to improve the overall performance.

The original SSD [7] uses VGG-16 [26] net pre-trained with ImageNet as the base network, which only contains the information learned from object-centric dataset. To enrich the feature representations of the model, we design a two stream network to combine the information learned from different datasets. The overview of our proposed method is shown as Fig. 8. We first train a ZF-net [27] with Places-365 scene dataset, called SceneNet, and integrate it with the original SSD model carefully. The images are resized into 300 × 300 as the input. After network forwarding, each CNN will output a feature map with the same size. To combine the feature maps produced by these two CNNs, we apply a fusion layer after the network forwarding of feature extraction part. To improve the accuracy on small objects, an extra layer is also proposed to increase the feature representations. Finally, these feature maps will be fed into SSD detection module to perform object detection.

Table 2
Our PC specification.

CPU	Intel Core i7-2600 3.4 GHz
Memory	32 GB
Operation system	Ubuntu 16.04
Graphic processor units	NVIDIA GTX Titan X (Maxwell)

Table 3
Comparison of speed with other state-of-the-art proposal free methods, where the bold text indicates that the method achieves real-time speed, while the asterisk to Customized network indicates that it is not a standard YOLO architecture.

Method	Network	FPS	Input resolution
YOLO [6]	Customized network*	**45.0**	448 × 448
SSD300* [7]	VGG16	**46.0**	300 × 300
SSD321 [31]	Residual-101	11.2	321 × 321
DSSD321 [31]	Residual-101	9.5	321 × 321
YOLOv2 544 × 544 [32]	Darknet-19	**40.0**	544 × 544
Ours	VGG16 + ZF	**33.9**	300 × 300

4.3. OS-Net implementation

As we mentioned in the previous section, the single input image will be passed to two different CNNs in our proposed architecture. One is the original VGG16-Net, called ObjectNet, and the other is ZF-Net pre-trained with Places dataset, which is called SceneNet. Both fc7 layer of these CNNs will output a feature map with 1024 channels, which encode high level feature representations of the image. To fuse these two feature maps and learn more semantic information from them, we apply a fusion layer to combine them into a single feature map, which is as shown in Fig. 9.

These two feature maps will be concatenated along the channel axis so that each grid of the concatenated feature map will contain the feature representations from both networks with corresponding spatial information. Because these two CNNs are constructed with different architecture and trained separately, their activations have different scale, which will lead to unstable training. To solve this problem, we first apply L2 normalization [28] on both of these feature maps to normalize the feature value into the same scale before concatenating them.

Instead of using the concatenated feature map for performing the detection directly, we add several convolutional layers on top of it to learn the fusion ability. A convolutional layer with 1 × 1 kernel size is added first to merge the two feature maps into single cube. Then, we add a ReLU layer to increase the non-linearity. The 1 × 1 convolutional filters not only compress the input feature representations into low dimension, but also allow the network to learn how to combine the input features. Then, a convolutional layer with 3 × 3 kernel size is added to learn more semantic feature representations. Note that we can reduce the number of parameters of convolutional kernels by the 1 × 1 convolutional layer as compared with the direct application of a convolutional layer with large kernel size (e.g., the 3 × 3 convolution here). With this parameter reduction, we can prevent overfitting caused by too many parameters of convolutional kernels.

Due to the max-pooling layers are able to down-sample the feature maps, a deep convolution network will encode input image into a

Table 4
Evaluation results on MSCOCO dataset, where the bold text indicates the best performance work in each of the evaluation metrics.

Method	$AP^{0.5–0.95}$	$AP^{0.5}$	$AP^{0.75}$	AP^{S}	AP^{M}	AP^{L}	AR^{S}	AR^{M}	AR^{L}
SSD300 [7]	23.2	41.2	23.4	5.3	23.2	39.6	9.6	37.6	56.5
YOLOv2 [32]	21.6	44.0	19.2	5	22.3	35.5	9.8	36.5	54.4
SSD300* [7]	25.1	43.1	25.8	6.6	25.9	41.4	11.2	40.4	58.4
SSD321 [31]	28.0	45.4	**29.3**	6.2	28.3	**49.3**	11.5	43.3	**64.9**
DSSD321 [31]	28.0	46.1	29.2	7.4	28.1	47.6	12.7	42.0	62.6
Ours	**28.2**	**47.9**	29.0	**9.6**	**29.2**	43.5	**15.9**	**43.7**	59.8

small size feature map. For example, a 16 × 16 patch will be encoded into a 1 × 1 feature grid in VGG-16 network, which is too coarse for small object detection.

To improve the detection performance on small objects, we add some extra layers to enrich the fine-grained features. First, we concatenate the feature maps of conv4_3 of Object-Net (denoted as conv4_3_o) and conv2 of Scene-Net (denoted as conv2_s). Because the receptive field of feature map will become smaller when the convolution layer going deeper, the feature maps from shallow layers can provide local information. Also, the resolution of these two feature maps are twice of the fc7 feature map (e.g., 38 × 38 vs. 19 × 19), which are more suitable for small object detection. We also add an up-sample layer on top of the fusion layer, which can interpret CNN high-level features to a larger feature map. We use an up-sample layer with 4 × 4 kernel size, 512 group size, 1 padding size and a stride of 2 to up-sample the fusion feature map. We use bilinear interpolating instead of learnable deconvolution layer due to the computation complexity issues. Finally, we will obtain a 38 × 38 × 512 feature map which is shown in Fig. 10.

This up-sampling process can produce a feature map with a resolution the same as that of the feature map of conv4_3 layer of Object-Net and conv2 layer of Scene-Net, which allows us to concatenate them into a single feature cube without losing spatial corresponding relation. We also apply L2 normalization on each feature map to convert the activation into the same scale before concatenating them. A convolutional layer with 1 × 1 kernel size and a ReLU layer are also applied to compress the feature map dimension and learn more semantic feature representations. A default box set with small scale is assigned to the output feature to predict small objects.

4.4. Training strategy

The well-tuned training parameters and strategy are the key points of training a deep network. We extract the convolutional layers of feature extraction part (e.g., until fc7) of the original SSD model, which is based on VGG16-net, as our Object-Net.

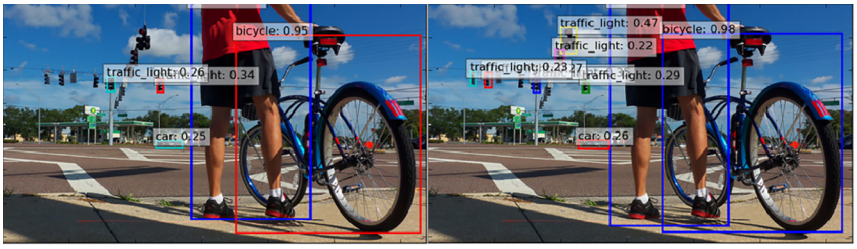
The feature extraction parts of Object-Net and Scene-Net are the base network of our OS-Net. The weights of fusion layers and detection module are initialized by Xavier [29] algorithm. The learning rate of each layer is set by multiplying initial learning rate by a pre-define factor lr_mult. We set the lr_mult of base network to 1 and the one of fusion layers and detection module to 10. This training strategy allows us to train the detection module without losing the feature extracting ability of the base network.

5. Experiments

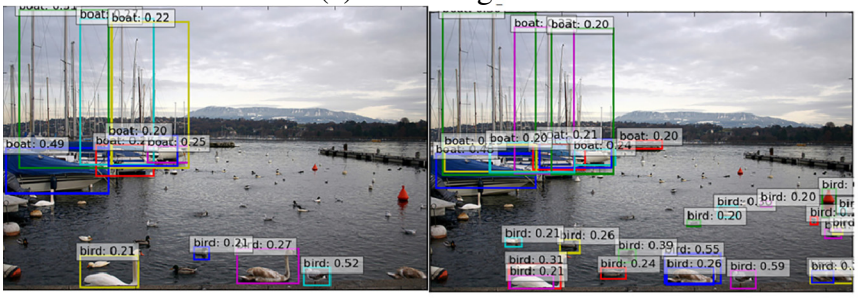
5.1. The datasets

MSCOCO [5] is the most challenging dataset nowadays, which is our main experimental target. MSCOCO dataset has 80 different object classes with 80 k training images and 40 k validation images and 40 k testing images. The size of each image in MSCOCO dataset is about 640 ∗ 480. Comparing to PASCAL VOC dataset [4], images in MSCOCO dataset contain more small instances, which make MSCOCO dataset become more challenging than PASCAL VOC dataset.

We test our model on MSCOCO test-dev2015 set, which contains 20,288 testing images. Because MSCOCO only provides the ground truth for training data, the testing results need to be submitted to their evaluation server. MSCOCO evaluation server provides several



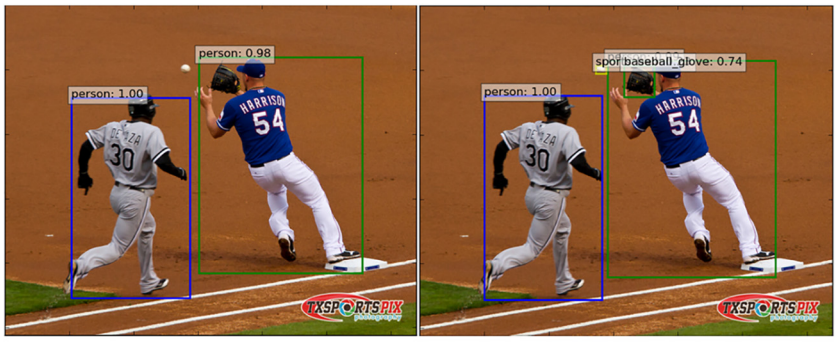
(a) Traffic lights.



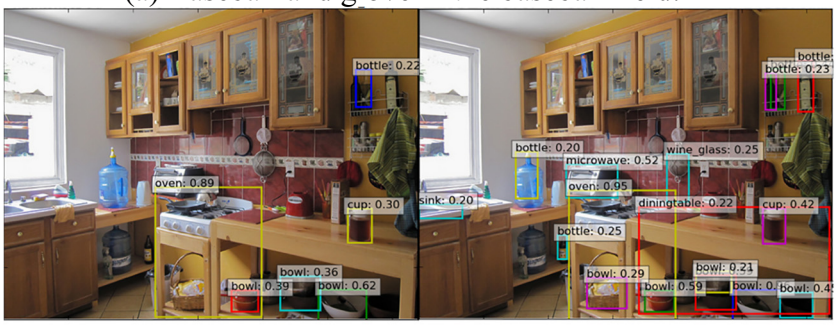
(b) Birds.

(c) Baseball, glove and bat.

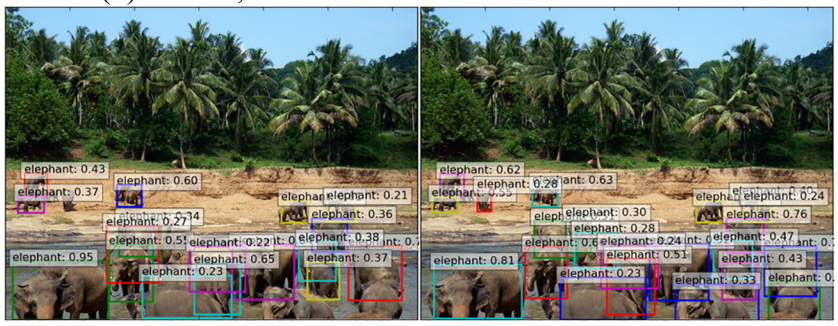
Fig. 11. Detection results of MSCOCO test-dev 2015, part I. The left side is the results of SSD300* and the right side is the results of our proposed method. These results show the performance on small object detection.



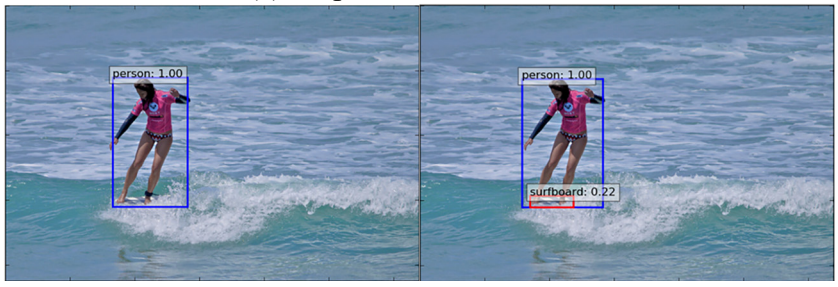
(a) Baseball and glove in the baseball field.



(b) Bottles, microwave and sink in the kitchen.



(c) Elephants in the wild.



(d) Surfboard on the wave.

Fig. 12. Detection results of MSCOCO test-dev 2015, part II. The left side is the results of SSD300* and the right side is the results of our proposed method. These figures show the relationship between objects and scene.

Table 5
mAP of each category on PASVOC 2012 testing set, where the bold text indicates the best performance work in each category.

Method	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV
YOLO	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD321	87.9	82.9	73.7	61.5	45.3	81.4	75.6	92.6	57.4	78.3	65.0	90.8	86.8	85.8	81.5	50.3	78.1	75.3	85.2	72.5
DSSD321	87.3	83.3	75.4	64.6	46.8	82.7	76.5	92.9	59.5	78.3	64.3	91.5	86.6	86.6	82.1	53.3	79.6	75.7	85.2	73.9
YOLOv2	88.8	87.0	77.8	64.9	51.8	85.2	79.3	93.1	**64.4**	81.4	**70.2**	91.3	88.1	87.2	81.0	**57.7**	78.1	71.0	88.5	76.8
SSD300*	91.0	86	78.1	65.0	55.4	84.9	84	**93.4**	62.1	83.6	67.3	91.3	88.9	88.6	85.6	54.7	83.8	**77.3**	88.3	76.5
Ours	**91.1**	**87.3**	**78.8**	**66.9**	**57.0**	**86.1**	**84.3**	92.8	63.3	**85.5**	68.0	**91.4**	**90.2**	**89.2**	**86**	54.4	**84.2**	75.7	**88.9**	**77.2**

Table 6
PASCAL VOC 2012 detection results, where the bold text indicates the best performance work.

Method	Data	mAP
YOLO [6]	07++12	57.9
SSD321 [31]	07++12	75.4
DSSD321 [31]	07++12	76.3
YOLOv2 544 [32]	07++12 + coco	78.2
SSD300* [7]	07++12 + coco	79.3
Ours	07++12 + coco	**79.9**

metrics for validating the performance of each method, which are shown in Table 1.

PASCAL VOC [4] is a public object benchmark introduced which is widely used for evaluating an object detector. There are two subsets in PASCAL VOC, one of which is for 2007 challenge and the other one is for 2012. Both of these subsets have 20 labeled object categories. We evaluate our method on PASCAL VOC 2012 testing which has 10,991 testing data.

5.2. Experiment platform

Our network is trained and tested on a personal computer with single NVIDIA GTX Titan X GPU (Maxwell). The specifications of our experiment platform are listed in Table 2.

We implement our method based on Caffe [30]. Caffe is a deep learning framework developed by Berkeley AI Research (BAIR) and community contributors. Caffe integrates NVIDIA Cuda and cudnn toolkit, which allow us to utilize the GPU efficiently. We train and test our model using Cuda 8.0 and cudnn v6.

5.3. Experimental results

5.3.1. Inference time

We compare our testing time to other state-of-the-art proposal free methods. To fairly compare the running time, the speed of each method is measured on PASCAL VOC with batch one configuration. The comparison of speed is summarized in Table 3. Note that YOLO [6] uses a customized network which has 24 convolutional layers and SSD300* is the latest SSD results. Our method has a frame rate of 33.9fps (including all processing time, e.g., disk I/O time, pre-processing and NMS). Our proposed model is not as fast as the original SSD caused by the added ZF-Net, but our method is faster than SSD321 and DSSD321 [31] which use Residual-101 as their base network.

5.3.2. The experimental results on MSCOCO datasets

In order to fairly compare our model to original SSD, we follow the training setting of SSD300* [7] which splits the validation set into two sets; one contains 5000 images and the other contains 35,000 images. Then, we add a 35,000-image set into the training set for training, called trainval35k set.

The experimental results are shown in Table 4. Note that YOLO [6] does not provide experimental results on MSCOCO dataset.

Table 4 show that our method can achieves a better overall average precision (**0.5:0.95**) than YOLOv2 [32], SSD300* with VGG-Net [7], and SSD321 with Residual-101 network [31]. And, our method also outperforms DSSD321 [31] which adds several extra layers on each SSD detection layer to improve the accuracy on small objects. The experimental results show that our proposed method can actually improve the accuracy on small objects even with shallower networks and slightly smaller input size compared to DSSD321 (AP^{small}: 9.6% vs. 7.4%, AR^{small}: 15.9% vs. 12.7%).

Fig. 13. Sample detection results of PASCAL VOC 2012 testing set.

Our method outperforms YOLOv2 [32], SSD300* [7] on all evaluation metrics with comparable speed. Compared to SSD321 and DSSD321 [31], although our method cannot improve the accuracy on large object (only from 41.4% to 43.5%) as well as using Residual-101 network, the improvement on small objects still leads to better overall results ($AP^{0.5-0.95}$: 28.2% vs. 28.0%). Also, our model is faster than SSD321 and DSSD321 with about 3× speed. We conclude that our model can improve overall accuracy without losing too much speed.

In Figs. 11 and 12, we visualize some detection results on MSCOCO test-dev 2015 with SSD300* [7] and our proposed OS-Net. We can see our proposed method achieve large improvement on SSD300* [7] in two cases. First, our method can figure out small objects from the input images. For example, although the traffic lights in Fig. 11(a), birds in Fig. 11(b), and the baseball in Fig. 11(c) are extremely small, our model still can figure them out.

Another case is that if there are some specific context relationships between object and object or object and scene, the combination of scene-centric CNN can provide helpful contextual information which can lead to a huge improvement. For example, baseballs, bats and baseball glove in Fig. 11(c) and Fig. 12(a) can be detected by our OS-Net due to their relationships between baseball player and baseball field.

5.3.3. The experimental results on PASCAL VOC 2012

For PASCAL VOC 2012 testing set, we also follow the same training setting used by SSD300*. We fine-tune from the model trained on MSCOCO trainval35k first. Also, we use VOC2007 trainval set, VOC2007 test set and VOC2012 trainval set for training (total 21,503 training images), which is denoted as "07++12 + coco". For PASCAL VOC 2012 testing set, each prediction is determined as corrected answer if their IoU with ground truth is greater than 0.5. Because the testing results of PASCAL VOC 2012 also need to be submitted to their evaluation server, here we show the results provided by them.

We list the detection results for each category in Table 5. Our method can perform better than YOLOv2 [32] and SSD300* [7] on most object categories.

Overall experimental results are shown in Table 6. Note that "07++12" denotes the method which is trained on VOC2007 trainval set, VOC2007 test set and VOC2012 trainval set only.

We show some detection results in Fig. 13. These results show that our method can also perform well on PASCAL VOC dataset after being fine-tuned with their training set.

6. Conclusions

In this research, a novel CNN based object detector, called OS-Net, is proposed. The proposed method is based on two-stage detection framework, which allows the detector to achieve real-time speed. By combining object-centric and scene-centric information, OS-Net can improve the accuracy especially on small object detection, which is a common limitation of two-stage methods.

To explore what CNN actually learns from the data, we first analyze the distributions of the objects in object-centric dataset, scene-centric dataset, and object detection dataset. Then, we visualize the activation of object-centric CNN (e.g., ImageNet-CNN) and scene-centric CNN (e.g., Places-CNN). The results show that scene-centric CNN can provide different information from the one provided by object-centric CNN, which is helpful for localizing small objects in the image. Based on this observation, we propose a framework to combine these two CNNs to perform object detection. The proposed fusion layers can integrate the feature maps produced by these two CNNs into single high-level fusion features. We also present an extra layer, which combines low-level high-resolution feature maps with up-sampled fusion feature, to further improve the performance on small objects.

The evaluation is performed on MSCOCO and PASCAL VOC 2012 dataset. The results on MSCOCO show that our model outperforms other state-of-the-art two-stage methods with 28.2% AP^(0.5–0.95). Then, the average precision for small objects also shows that our method can actually achieve better accuracy on small objects detection. In addition, the experiment results also show that the additional scene contextual information can lead to a huge improvement on the objects which have specific relationship with the scene. Besides, OS-Net also achieves state-of-the-art performance on PASCAL VOC 2012 datasets. Moreover, the proposed method is suitable for being adopted to other detection frameworks to improve their performance.

Conflicts of interest

The authors declare no conflict of interest.

Acknowledgement

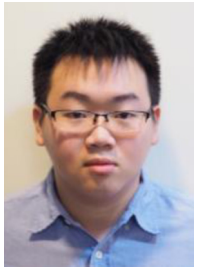
This work was partially sponsored by Ministry of Science and Technology, Taiwan, ROC, under grant 108-2634-F-002-016, 108-2634-F-002-017, 107-2218-E-002-009, 103-2221-E-390-028-MY2 and 105-2221-E-390-024-MY3.

References

[1] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pp. 580–587.
[2] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, Advances in Neural Information Processing Systems 2015, pp. 91–99.
[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2009, pp. 248–255.
[4] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[5] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context, European Conference on Computer Vision, Springer 2014, pp. 740–755.
[6] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 779–788.
[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, Proceedings of the European Conference on Computer Vision, Springer 2016, pp. 21–37.
[8] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, Advances in Neural Information Processing Systems 2014, pp. 487–495.
[9] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, A. Torralba, Sun database: large-scale scene recognition from abbey to zoo, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE 2010, pp. 3485–3492.
[10] J. Xiao, K.A. Ehinger, J. Hays, A. Torralba, A. Oliva, SUN database: exploring a large collection of scene categories, Int. J. Comput. Vis. 119 (1) (2016) 3–22.
[11] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems 2012, pp. 1097–1105.
[12] A. Berg, J. Deng, F. Li, ILSVRC-2010, URL http://www.image-net.org/challenges/LSVRC/2010/ 2010.
[13] J.R. Uijlings, K.E. van de Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.
[14] C. Cortes, V. Vapnik, Support vector machine, Mach. Learn. 20 (3) (1995) 273–297.
[15] R. Girshick, Fast r-cnn, Proceedings of the IEEE International Conference on Computer Vision 2015, pp. 1440–1448.
[16] E.L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting linear structure within convolutional networks for efficient evaluation, Advances in Neural Information Processing Systems 2014, pp. 1269–1277.
[17] J. Xue, J. Li, Y. Gong, Restructuring of deep neural network acoustic models with singular value decomposition, Interspeech 2013, pp. 2365–2369.
[18] S. Bell, C.L. Zitnick, K. Bala, R. Girshick, Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks, IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 2874–2883.
[19] T. Kong, A. Yao, Y. Chen, F. Sun, HyperNet: towards accurate region proposal generation and joint object detection, IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 845–853.
[20] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 2921–2929.
[21] L. Herranz, S. Jiang, X. Li, Scene recognition with CNNs: objects, scales and dataset bias, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 571–579.
[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.

[23] A. Quattoni, A. Torralba, Recognizing indoor scenes, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2009, pp. 413–420.
[24] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: a deep convolutional activation feature for generic visual recognition, International Conference on Machine Learning 2014, pp. 647–655.
[25] M. Holschneider, R. Kronland-Martinet, J. Morlet, P. Tchamitchian, A real-time algorithm for signal analysis with the help of the wavelet transform, Wavelets, Springer 1990, pp. 286–297.
[26] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-scale Image Recognition, arXiv preprint arXiv:1409.1556 2014.
[27] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, Proceedings of the European Conference on Computer Vision, Springer 2014, pp. 818–833.
[28] W. Liu, A. Rabinovich, A.C. Berg, Parsenet: Looking Wider to See Better, arXiv preprint arXiv:1506.04579 2015.
[29] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics 2010, pp. 249–256.
[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, Proceedings of the 22nd ACM International Conference on Multimedia, ACM 2014, pp. 675–678.
[31] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, DSSD: Deconvolutional Single Shot Detector, arXiv preprint arXiv:1701.06659 2017.
[32] J. Redmon, A. Farhadi, YOLO9000: Better, Faster, Stronger, arXiv preprint arXiv:1612.08242 2016.
[33] X. Zeng et al., Crafting GBD-net for object detection, IEEE Trans. Pattern Anal. Mach. Intell., vol. PP, no. 99, 1.

Zong-Ying Shen was born on January 22, 1993 in New Taipei, Taiwan. He received the B.S. degree in department of electrical engineering from National Cheng Kung University, Tainan, Taiwan in 2015 and the M.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2017. His research includes the area of computer vision and object detection.



Hsiang-Yu Han was born on July 25, 1994 in Tauyuan, Taiwan. He received the B.S. degree in Department of Computer Science & Information Engineering from National Central University, Taiwan in 2016. His research includes the area of computer vision and object detection.

Li-Chen Fu (M'84-SM'94-F'04) received the B.S. degree from National Taiwan University in 1981, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1985 and 1987, respectively. Since 1987, he has been on the faculty of and currently is a professor in both the Department of Electrical Engineering and Department of Computer Science & Information Engineering of National Taiwan University. He is now a senior member of both the Robotics and Automation Society and Automatic Control Society of IEEE, and he became an IEEE Fellow (F) in 2004. His areas of research interest include robotics, FMS scheduling, shop floor control, home automation, visual detection and tracking, E-commerce, and control theory & applications.

Pei-Yung Hsiao (M'90) received the B.S. degree in chemical engineering from Tung Hai University, in 1980 and the M.S. and Ph.D. degrees in electrical engineering from the National Taiwan University, in 1987 and 1990, respectively. In 1990, he was an Associate Professor in the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan. In 1998, he was the CEO of Aetex Biometric Corporation. He is currently a Professor in the Department of Electrical Engineering, National Univ. of Kaohsiung. His research interests and industrial experiences include VLSI/CAD image processing, fingerprint recognition, visual detection, embedded systems, FPGA rapid prototyping, and DIP/SOC.

Yo-Chung Lau received the B.S. and M.S. degree from National Cheng Kung University in 2005 and 2007, respectively. After graduating from school, he served for Novatek Microelectronics Corp. and focused on the development of SOC products in 2008 to 2011. From 2011 to now, he is a researcher in Telecommunication Laboratories, Chunghwa Telecom Co., Ltd. His areas of research interest include AR and VR human-computer interaction technology.

Sheng-Jen Chang received the B.S. degree from National Cheng Chi University in 2000 and M.S. degree from National Chi Nan University in 2002, respectively. After graduating from school, he is a researcher in Telecommunication Laboratories, Chunghwa Telecom Co., Ltd. His areas of research interest include AI, AR and VR human-computer interaction technology.