Neurocomputing 286 (2018) 58–66

Contents lists available at ScienceDirect

Neurocomputing

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

Object detection via deeply exploiting depth information

Saihui Hou, Zilei Wang*, Feng Wu

CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System, University of Science and Technology of China, Hefei 230027, China

ARTICLE INFO

Article history:
Received 2 June 2016
Revised 24 January 2018
Accepted 27 January 2018
Available online 2 February 2018

Communicated by Steven Hoi

Keywords:
Property derivation
Property fusion
RGB-D perception
Object detection

ABSTRACT

This paper addresses the issue on how to more effectively coordinate the depth with RGB aiming at boosting the performance of RGB-D object detection. Particularly, we investigate two primary ideas under the CNN model: property derivation and property fusion. Firstly, we propose that the depth can be utilized not only as a type of extra information besides RGB but also to derive more visual properties for comprehensively describing the objects of interest. Then a two-stage learning framework consisting of property derivation and fusion is constructed. Here the properties can be derived either from the provided color/depth or their pairs (e.g. the geometry contour). Secondly, we explore the fusion methods of different properties in feature learning, which is boiled down to, under the CNN model, from which layer the properties should be fused together. The analysis shows that different semantic properties should be learned separately and combined before passing into the final classifier. Actually, such a detection way is in accordance with the mechanism of the primary visual cortex (V_1) in brain. We experimentally evaluate the proposed method on the challenging datasets *NYUD2* and *SUN RGB-D*, and both achieve remarkable performances that outperform the baselines.

1. Introduction

Thanks to the availability of affordable RGB-D sensors, e.g. the Microsoft Kinect, the RGB-D images have been widely provided in the read-world visual analysis systems. Compared with the primitive RGB, the RGB-D can bring remarkable performance improvement for various visual tasks due to the access to the depth information complementary to RGB [1–3]. Actually, the depth has some profitable attributes for visual analysis, e.g. being invariant to illumination or color variations, and providing geometrical cues for image structures [4]. For object detection, which is one of typical complex visual tasks, the acquisition of RGB-D images is applicable and beneficial. However, how to effectively utilize the provided depth information of RGB-D images is still an open question.

Recent years have witnessed the great success of Convolutional Neural Network (CNN) in computer vision, which has boosted the performance of various visual tasks to a new level [5–7]. The CNN is generally considered as an end-to-end extractor to automatically learn discriminative features from millions of input images [8]. In this paper, we also adopt CNN to extract rich features from the RGB-D images, i.e. we are under the CNN model to investigate the exploitation of the depth information.

For the RGB-D object detection with CNN, the key is how to elegantly coordinate the RGB with depth information in feature learning. In the previous literatures, some intuitive methods have been proposed [9,10]. Roughly, we can divide them into two broad categories according to the strategy the depth is treated. The first one is to straightforwardly add the depth map to CNN as the fourth channel along with the RGB [9]. That is, the depth is processed in the same way as the RGB, and they are together convolved for granted. However, it makes no semantic sense to directly merge the depth and color maps, since they contain disparate information. The second is to process the color and depth separately, and they are combined before being fed into the final classifier, where the extracted features are joint. Specifically, two independent CNN networks are learned: one for RGB and one for depth [10]. As for the depth network, the input can be the original depth data or encoded data from the depth, e.g. height above ground, and angle with gravity [10]. It has been empirically shown that the second way usually outperforms the first one. In this paper, we further investigate how to deeply exploit the depth information with the aims of boosting the detection performance.

Before introducing the proposed method, we review the primary mechanism of human visual systems. First, multiple visual properties are always used together to describe one object when people try to recognize it, e.g. geometry contour, color, and contrast [11]. And it is usually thought that exploiting more properties is much helpful. Second, the primary visual cortex (V_1), which con-

* Corresponding author.
E-mail addresses: saihui@mail.ustc.edu.cn (S. Hou), zlwang@ustc.edu.cn (Z. Wang), fengwu@ustc.edu.cn (F. Wu).

https://doi.org/10.1016/j.neucom.2018.01.055
0925-2312/

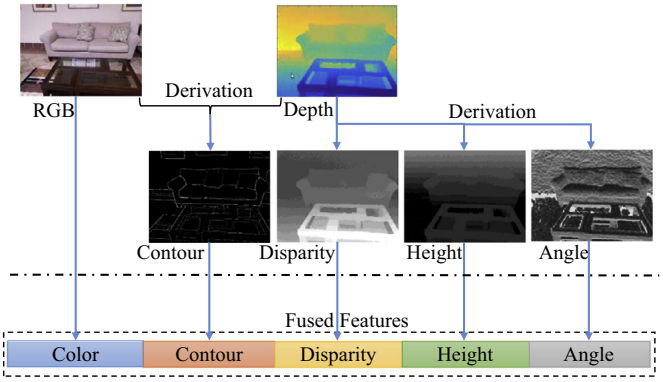

Fig. 1. Illustration of learning rich features for RGB-D object detection. Various property maps are derived to describe the objects from different perspectives. The features for these maps are learned independently and then fused for the final classification. Specifically, the derived maps include geometry contour from the color/depth pairs, and horizontal disparity, height above ground, angle with gravity from the depth data. These maps, as well as the RGB image, are sent into different CNNs for feature learning. And the features are joint before being fed into the classifier.

sists of six functionally distinct layers and is highly specialized in pattern recognition [12], abstracts different visual properties independently in the low layers and integrates in the relatively high layers.

Inspired by the working mechanism of V_1 area, we propose a novel method to deeply exploit the depth information for object detection. Fig. 1 illustrates the main idea of our method. Firstly, various visual property maps are derived through analyzing the provided color and depth pairs. It is believed that more properties can contribute to the accurate description of the objects and thus help boost the detection performance. Specifically, the derived properties include the contour, height, and angle maps. Secondly, we systematically investigate the methods to fuse different visual properties under the CNN model, i.e. how to represent a property, and from which layer the properties need to be fused together. The result of our analysis shows that the multiple properties should have complete and independent semantics in accordance with the human cognition, e.g. RGB channels should be treated as a whole to represent the color property rather than separate them from each other, and it is better to fuse the different properties after they are explicitly transformed into the high-level features.

We evaluate the proposed method on the challenging *NYUD2* and newly published *SUN RGB-D*, and the experimental results demonstrate that our method works reasonably well on both datasets and achieves remarkable performances that outperform the baselines. This is an extended version of the work that appeared in [13]. It differs from [13] in that:

- The proposed model is further evaluated on the *SUN RGB-D* with larger training data, i.e., the generalization ability of the method is validated.
- We study the effect of each involved property map and their combinations on the performance of object detection.
- A proper order is put forward to add the extra property maps in turn complementary to the RGB for the sake of boosting the detection performance, if not all of them are needed.
- More theoretical and experimental details are provided in this paper.

The remainder of the paper is organized as follows. In Section 2, we review the related works on RGB and RGB-D object detection. Section 3 provides the details of our approach, and Section 4 experimentally evaluates the proposed method. Finally, we conclude the work in Section 5.

2. Related work

Object detection [14] is to mark out and label the bounding boxes around the objects in an image. Particularly, the adopted features are critical in determining the detection performance [15]. Traditional methods, including the MRF [16] and DPM [17], are all based on the hand-crafted features such as SIFT [18] and HOG [19]. However, these features are difficult to adapt to the specific characteristics in a given task. And more recent works [20–22] have turned to the Convolutional Neural Network (CNN), which can learn discriminative features automatically from millions of RGB images. A typical CNN consists of a number of convolution and pooling layers optionally followed by the fully connected layers [8], and is able to learn multi-level features ranging from edges to entire object [23].

For object detection with CNN, a classical method is to build a sliding-window detector and then take the CNN for classification, which is usually applied on constrained object categories, e.g. faces [24] and pedestrians [15]. And the object detection is formulated as a regression problem in [20,21] then the CNN is involved in predicting the localization and labels of the bounding boxes. The most remarkable work lies in [22]. The system called R-CNN first generates around 2000 category-independent region proposals for an input image and then computes features of each region with the CNN. A category-specific SVM is appended to predict the label and score for each proposal.

When it comes to the RGB-D object detection, Gupta et al. [10] proves that CNN can also be trained to learn depth features from the depth map. In practice, the extra depth exactly makes it easy to recognize human pose [1], align 3D models [25], and detect objects [4,9,10]. Under the CNN model, two typical methods for RGB-D object detection have been proposed about how to utilize the depth information [9,10]. One is to directly add the fourth channel for depth, and then equally convolve all channels in one network [9]. The other is to separately process the depth and color (RGB) using two independent networks [10]. Obviously, these works mainly focus on the extraction of depth features, rather than considering thoroughly how to better coordinate the color and depth pairs for accurately describing the objects.

A more related work is the one by Gupta et al.[10], in which the depth data is encoded to horizontal disparity (D), height above ground (H), angle with gravity (A), and then form the three-channel DHA image into CNN to learn depth features, besides the RGB network. In our work, differently, the D, H, A are relighted and derived as new maps describing the objects from different perspectives, and used to separately learn particular types of features encoding the multiple visual properties. More than that, we propose to use the depth combining with the RGB to derive new maps to provide extra information, e.g. the geometry contour. Indeed, other properties can be also adopted, which may be obtained by specific sensors or more advanced derivation methods, e.g., colorized depth [26], distance from wall [27]. Considering the simplicity, only several directly computable properties are employed here. Furthermore, we systematically investigate the fusion ways of different properties under the CNN model. We believe that our proposed detection framework and the investigation of feature fusion would inspire more advanced works to significantly improve the performance of object detection.

3. Our approach

Intuitively, acquiring more information contents about the objects could yield more accurate recognition. Meanwhile, for human being, the visual cortex of the brain is exactly to abstract various types of visual information from the input scenes in the inception phase [12]. Inspired by such a principle, more informa-

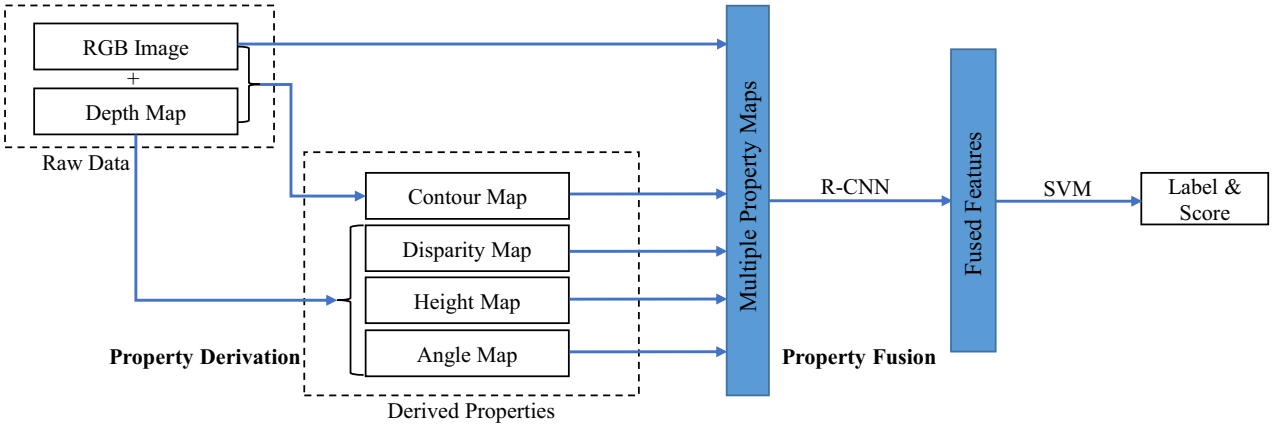

Fig. 2. Overview of our framework for RGB-D object detection. We manage to learn multiple properties of the objects and fuse them better for the detection. Various maps including geometry contour, horizontal disparity, height above ground and angle with gravity are first derived from the raw color and depth pairs. These maps, along with the RGB image, are sent into different CNNs to learn particular types of features. And the features are fused at the highest level, i.e. they are not joint until passing into the classifier. The region proposals for R-CNN [22] are generated using MCG [28] with the depth information. A SVM is appended to predict the label and score for each proposal.

tional sources are always desired in computer vision, e.g. developing more powerful or precise sensors [1]. In practical applications, however, the accessible sources are rather limited because of the constraints on deployment and cost. In this paper, we attempt to mine as much useful information as possible by analyzing thoroughly the available data, for the sake of boosting the performance of RGB-D object detection, i.e. only the color image of single view and the corresponding depth map are originally provided.

To this end, we propose a novel two-stage feature learning framework for object detection on the RGB-D data, as shown in Fig. 2. Specifically, we first derive more property maps from the input color and depth pairs. This procedure functionally emulates the abstracting mechanism of primary visual cortex. These properties describe the objects from multiple views, and combine with the raw data to form a relatively complete set of feature maps. Then we adopt the well-performed Convolutional Neural Network (CNN) model to generate image representation from these feature maps. Specially, the method to fuse different properties under the CNN model would be investigated systematically in this work. Finally, we feed the joint representation into the SVM for classification. In the proposed framework, the property derivation and property fusion are especially important to determine the performance of object detection.

3.1. Property derivation

It is unlikely for any of existing methods, including the CNN, to learn the various properties of the objects accurately from the same input scene as the human brain does, especially when only limited data is available. So it is essential to derive more property maps complementary to the original color and depth. For human visual system, the geometric properties usually play an important role in recognizing the objects, e.g. shape and outline. Fortunately, the availability of depth information makes it possible to compute the geometric properties with greater accuracy, compared to only using the plain color data. In this paper we mainly focus on the properties that can be derived from the raw color and depth data directly.

Specifically, the adopted properties include Ultrametric Contour Map (UCM), Horizontal Disparity (D), Height Above Ground (H), and Angle With Gravity (A). In particular, UCM representing the geometry contour is calculated using both the color and depth data, and the other three properties are only from the depth map. The horizontal disparity is an encoded version of the original depth map since it is more suitable for feature leaning with CNN [10]. Along with the color image, the final property maps comprise of RGB, UCM, D, H, and A. Now we elaborate on the computation of *geometry contour, height above ground* and *angle with gravity*.

3.1.1. Geometry contour

In philosophy, *geometry contour* tells directly the boundaries between the objects of interest and the background [29]. In object detection with CNN, however, the input data is only the images of rectangular area, which are produced by either sliding windows or region proposals [22]. That is, both the objects and context in bounding boxes are processed blindly. So explicitly providing the contour map would help CNN to mark out the objects more accurately. In this work, we particularly adopt the Ultrametric Contour Map (UCM) produced by the gPb-ucm algorithm [30] combining the RGB and depth data.

3.1.2. Height

Height above ground indicates the position of the objects *w.r.t.* different categories, e.g. a television is usually put on a table and a pillow is on the bed or sofa [31]. The height map exactly represents such relations between the objects and thus is beneficial to distinguish the objects of interest. In this work, we produce the height map in an approximate way. Specifically, the 3D point cloud is first gained and aligned by processing the depth map. Then the height value of each pixel is roughly calculated by subtracting the lowest point (with the minimum height) within an image, which can be regarded as a surrogate for the supporting ground plane.

3.1.3. Angle

Angle with gravity gives a lot of cues about the image structures and what the real world looks like, e.g. the surface of a table or bed is usually horizontal while the wall or door is vertical. Such information provides important clues to the shape of the objects, and thus would be helpful for the object detection. In this work, we adopt the method in [31] to calculate the angle map.

Specifically, we first estimate the direction of gravity (DoG), denoted by g, using the depth data. In practice, g is updated in an iterative manner and initialized with the Y-axis. For the current DoG (g_{i-1}), the aligned set $\mathcal{N}_{||}$ and orthogonal set $\mathcal{N}_{\perp}$ are produced with an angle threshold d, i.e.

$$\mathcal{N}_{||} = \{n : \theta_{n,g_{i-1}} < d \mid \theta_{n,g_{i-1}} > 180^\circ - d\}, \tag{1}$$

$$\mathcal{N}_{\perp} = \{n : 90^\circ - d < \theta_{n,g_{i-1}} < 90^\circ + d\}, \tag{2}$$

where $n \in \mathcal{N}$ represents the local surface normal, and $\theta_{n,g_{i-1}}$ denotes the angle between n and g_{i-1}. Then a new g_i is estimated by solving the following optimization problem,

$$\min_{g_i:||g_i||=1} \sum_{n\in\mathcal{N}_{||}} \sin^2\theta_{n,g_i} + \sum_{n\in\mathcal{N}_{\perp}} \cos^2\theta_{n,g_i}. \quad (3)$$

It means the estimated gravity is expected to be aligned to the normals in $\mathcal{N}_{||}$ and orthogonal to the ones in $\mathcal{N}_{\perp}$.

Once the DoG is obtained, we assign each pixel with the value of angle made by its local surface normal with g_i. Consequently, the angle map is formed with the same size as the original image.

3.2. Property fusion

So far we have obtained multiple visual property maps to represent the rich information of the objects, which include the color (RGB), geometry contour (UCM), horizontal disparity (D), height above ground (H), and angle with gravity (A). For object detection, an unified image representation is always needed for the final classification that integrates the useful information from all properties. For each property represented by one map (except the color with three channels of RGB), the sophisticated learning model can be directly applied to produce the corresponding feature, e.g. CNN. But how to coordinate multiple different properties in feature learning is not fully explored yet. For example, an intuitive method is to straightforwardly input all property maps together into CNN with multiple channels [9], but the resulting performance may be unsatisfying. Thus the key of generating the joint discriminative image representation is to determine an integration method, which can better fuse the multiple properties.

In this section, we attempt to deeply investigate the ways to fuse the different properties under the CNN model. Particularly, part of property maps obtained in the previous stage would be directly adopted when the numerical analysis is needed.

3.2.1. Fusion analysis

CNN can naturally learn the hierarchical features by different layers [23], in which the input map is finally transformed into a feature vector through a series of operations represented by convolution and pooling [8]. When multiple property maps of the same scene are fed into CNN, we can view these maps as the lowest level features. Then the property fusion is essentially to determine that, from which layer different property features should be arithmetically calculated together.

There are two extreme cases for property fusion. One is to directly increase the number of channels in the input layer of CNN and accept all property maps equivalently. The other is to separately learn different property features with independent networks and concatenate them before passing into the final classifier. In this work, both cases are denoted by *input* and *final* respectively for convenience. Of course, we can also fuse different properties from certain middle layer, which actually means that different property features are learned independently before that layer and then drawn from the synthesis of properties in the subsequent layers.

We first analyze the underlying reasons for the difference of fusing the properties in multiple layers, and the classical AlexNet [8] is adopted here. Fig. 3 provides an exemplar of network architecture for fusion, where the properties are fused from the *fc6* layer, and the three property maps of D, H and A are used without loss of generality. For a specific fusion network, e.g. fusing from *fc6* here, the learned features in the previous layer are straightly stacked into multiple feature maps, and then these maps are processed without distinction in the subsequent layers. It can be observed that the inter-layer connections between different property features are added after the first fused layer (*fc6*), which formally increases the network parameters. However, it is still not clear that

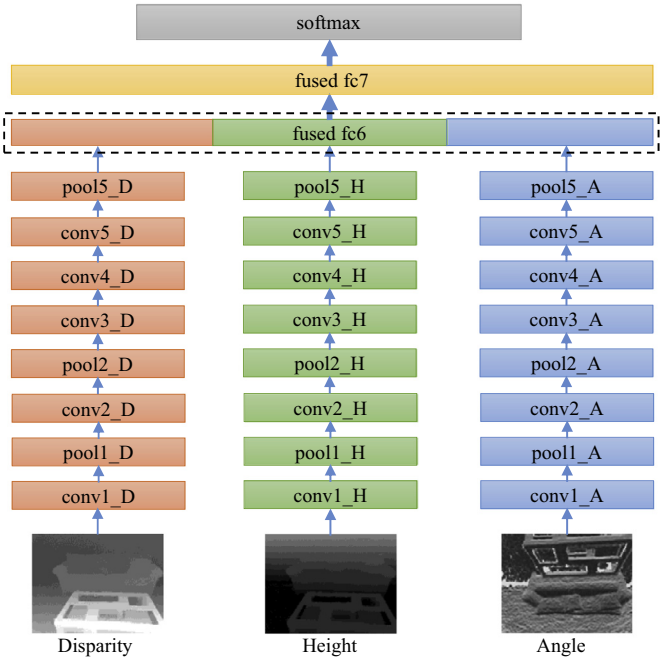

Fig. 3. The architecture of network when fusing from *fc6*. The D, H, A networks are independent before the first fused layer (*fc6*) and initialized with the individual trained model. The *fused fc6* and *fused fc7* are fully connected. And then the whole network is globally finetuned. The parameter is increased compared to the individual three networks, by the connections between the layers after the fused layer.

Table 1
The ablation study to exploit which components in CNN cause the performance gap of different fusion strategies. For fusion schemes of *input* and *final*, the architecture of CNN for feature learning are respectively *full, conv+pool, conv, conv+relu*. The results are all mean AP^b in percentage on *NYUD2 val* set. See Section 3.2.1.

Input			
Arch	DHA *(input)*	D+H+A *(final)*	*Gap*
full	24.75	28.85	4.10
conv+pool	12.42	14.52	2.10
conv	1.58	1.95	0.37
conv+relu	13.91	20.56	6.65

how these connections impact on the final representation and also on the detection performance.

The typical CNN model has three major functional components, i.e. *convolution, rectified linear units(ReLU)*, and *max-pooling*. In particular, the *convolution* belongs to the linear operators and thus theoretically has no great impact on fusing from different layers. In contrast, both the *ReLU* and *max-pooling* are of high nonlinearity. So it is naturally supposed that different fusion schemes are through the *ReLU* and *max-pooling* to impact the learning results. We conduct an evaluation experiment in order to show the effect of three CNN components more intuitively. Here the dataset of *NYUD2* [32] is used, and the performance is evaluated on its *val* set (See Sections 4.1 and 4.2 for the details of dataset and model training). It should be noted that we measure the impact of functional components by comparing the resulting performance of the two extreme fusion schemes (i.e. *input* and *final*). In addition, the PCA (Principal Component Analysis[1]) is conducted on the joint image representation in *final* scheme to keep the same dimension of classification features. The results are shown in Table 1.

[1] https://en.wikipedia.org/wiki/Principal_component_analysis.

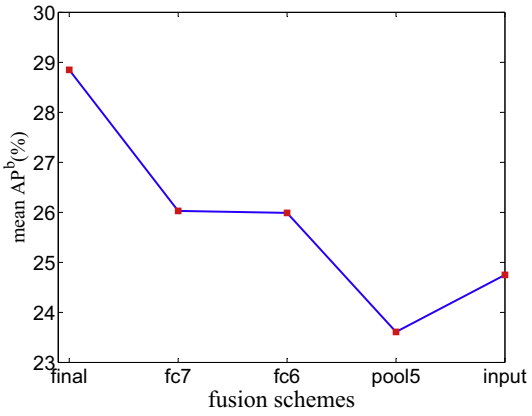

Fig. 4. The performance on *NYUD2 val* set when fusing features from different layers. We intuitively fuse the D, H, A properties with the schemes of *input, final, fc7, fc6, pool5* in turn. See Section 3.2.2.

According to Table 1, for the *full* network (AlexNet), the detection performance of fusing with *final* scheme is obviously higher than that of fusing with *input*. Then the *ReLU* (also normalization and dropout) is removed (*conv+pool*), the gap goes down but still exists. In the conditions that only the *convolution* is adopted[2] (*conv*), the mean performance of the detection for both fusion schemes are approximately equal. But the performance gap is widened when the *ReLU* is imposed (*conv+relu*). In summary, it is the nonlinear functional components (i.e. *ReLU* and *max-pooling* for CNN), that make the fusion schemes achieve a wide range of detection accuracy. Thus it is necessary to systematically explore the fusion schemes and choose the one that performs best.

3.2.2. Fusion scheme

In this subsection, we explore the property fusion schemes under the CNN model, i.e. from which layer the different property features need to be processed jointly. To address such an issue, we particularly adopt the strategy of experimental evaluation here, due to the intractableness of theoretical analysis. Specifically, we conduct multiple experiments using different fusion schemes, each of which represents starting the fusion from certain layer. We use the same experimental settings as in the previous subsection. The working mechanism of visual cortex in the brain inspires that the properties should be fused in the high level. So here we only investigate several relatively high layers, i.e. fusing from *fc7, fc6* and *pool5* are particularly chosen along with the *input* and *final* schemes. Fig. 4 displays the detection performance of different fusion schemes.

It can be seen from Fig. 4 that the *final* scheme results in the highest detection accuracy. Fusing from *fc6* and *fc7* reach better performance than *input*. But when fusing from *pool5* a little accuracy decrease occurs, which may be caused by introducing too much network parameters in the middle layers, while only limited number of training samples are provided. For the current architecture of CNN, therefore, it is recommended to fuse different visual properties in the final step, i.e. the learning models for encoding the property maps into the features should be trained separately. In the following experiments, the fusion scheme of *final* is directly adopted.

[2] The *fully connected layer* can be seen *convolution* layer in some way. And for networks without *max-pooling* layers, including the *conv* and *conv+relu*, the crop size for input images is 71*71.

4. Experimental result

4.1. Dataset

In this section, we evaluate the proposed method and compare the results with other related works on the challenging datasets *NYUD2* [32] and *SUN RGB-D* [33]. The box detection average precision (denoted by AP^b [6]) adopted in the PASCAL VOC Challenges is taken as the metric to measure the detection performance.

NYUD2 acts as one of classical datasets for RGB-D object detection and has been widely used in the previous works [9,10,25]. It consists of 1449 images of indoor scenes from Kinect v1 with vast variations of clutter and noise. And it is split here following the standard way: 795 images for *trainval* set and 654 images for *test* set. The 795 images are further divided into 381 for *train* and 414 for *val* as in [10].

SUN RGB-D is newly released to alleviate the lack of a large-scale benchmark with RGB-D data and annotations. It is much larger than *NYUD2* by one order of magnitude, with 5285 *train* and 5050 *test* images of indoor scenes. The images are collected by multiple sensors including RealSense, Xtion, Kinect v1 and v2, and hence presented with different resolutions. The objects come in various shapes, sizes, poses and frequent partial occlusions, making it complex and challenging to perform the recognition. There are already some previous works [34,35] which adopt the *SUN RGB-D* as benchmarks. However, only RGB is taken as input modality in those works.

4.2. Experimental setup

In our experiments, we consider the typical CNN architecture described in [8] and its Caffe implementation [36] for object detection, as in the previous works [6,10,22]. And the *liblinear* [37] for SVM is adopted. Our model training mainly consists of two aspects: finetune the CNNs for property feature learning, and train the SVM for proposal classification.

4.2.1. Finetune CNN

There are five CNNs in our final system, which are absolutely independent with each other and can be trained in parallel. The derived D, H, A and UCM maps, all have the same size with the raw RGB image. They are linearly scaled to 0–255 range and replicated three times to match the architecture of CNN.

For method evaluation on the *val* set of *NYUD2*, we first finetune the networks on its *train* set, which is started with the Caffemodel pretrained on ILSVRC12 dataset. The learning rate is initialized to 10^{-3} and decreased by a factor of 10 every 20k iterations. And the finetuning lasts for 30k iterations. Region proposals that overlap with the ground truth by more than 50% are taken as positives, and labeled with the maximal overlapping instance's class, while the rest proposals are all treated as background. When it comes to the *test* set of *NYUD2*, the CNNs are finetuned again on the *trainval* set. The learning rate is decreased every 30k iterations and there are 50k iterations in all. As for the *SUN RGB-D*, the step size for decreasing the learning rate is changed to 40k and the network training ends at 60k iterations.

Note that there is no data augmentation all through the experiments. After the finetuning is done, we cache and concatenate the features from *fully connected layer 6 (fc6)* of each network for SVM training.

4.2.2. Train SVM

The training is started with the hyper-parameters $C = 0.001, B = 10, w_1 = 2.0$. The positive set of each category is fixed to the ground truth boxes for the target class in each image, and the negative set is the boxes which overlap less than 30% with the ground

Table 2
Method evaluation on *NYUD2 val* set for RGB-D object detection. "+" between different maps means they are sent into different CNNs for feature learning and not joint until passing into the SVM. All the results are AP^b in percentage. See Section 4.3.

Input for CNN	I RGB+ DHA	II RGB+ D+H+A	III RGB+ D+H+A	IV R+G+B +DHA	V RGB+D+ H+A+UCM
PCA	no	no	yes	no	no
bathtub	19.90	36.76	36.70	22.03	33.93
bed	64.67	66.31	66.84	61.64	63.95
bookshelf	13.40	12.07	10.46	12.43	15.02
box	2.14	2.35	2.81	3.75	2.56
chair	39.15	44.99	43.71	36.98	44.83
counter	34.32	40.89	40.51	38.82	40.13
desk	10.94	11.50	10.37	6.31	11.50
door	19.77	20.87	19.78	22.71	18.71
dresser	23.91	23.97	23.98	19.47	26.69
garbage-bin	37.19	42.31	41.89	32.65	42.52
lamp	35.51	39.95	39.58	32.05	39.42
monitor	41.84	42.15	41.14	41.62	44.23
night-stand	33.69	38.58	38.64	38.26	43.56
pillow	32.18	35.93	33.72	32.89	38.18
sink	34.86	42.45	42.14	38.36	43.47
sofa	39.88	45.72	45.57	40.04	47.70
table	17.77	24.20	21.15	17.94	23.21
television	44.46	37.16	35.58	41.26	34.75
toilet	51.62	65.97	59.94	51.40	69.45
mean	31.43	35.48	34.45	31.09	**35.99**

truth instance from that class. The SVM is trained on *fc6* features of the *NYUD2 train* set for method evaluation on its *val* set, while trained on features of the *trainval* for the *test* set performance. For the *SUN RGB-D*, the SVM training data is made up with the feature of the *train* set for performance comparison on the *test* set. At test time, non-maximum suppression with the threshold 0.3 is first carried out on the proposals for each image and then perform evaluation.

4.3. Method evaluation

In Table 2, we report the performance in AP^b on the *val* set of *NYUD2* for method evaluation. It was implied in Section 3.2.2 that learning the features of different properties separately can achieve better performance. Here we further validate that with the complete property set. We start from the model in [10]. The depth map is encoded with the three channels of DHA images for depth feature learning, and then combine with the color features from RGB network to get the result mean AP^b of 31.43% in Column I. It's the best performance on the *NYUD2 val* set as we know before this paper (a little lower than [10] because we work with no data augmentation).

The first experiment is to separate the D, H, A maps into three networks (called DHA Separation for short) to train and cache features respectively, and then combine with the color features to get the mean AP^b of 35.48% in Column II. This procedure gives us a 4.05% improvement (31.43–35.48%, 12.89% relative), which is much surprising. However, we notice that the dimension of joint features after the DHA Separation increase by three times than that from the DHA network. In Column III, we take dimension reduction using PCA on the concatenated features from D, H, and A networks, to keep the dimension the same, and then combine with the color features too. We get the mean AP^b of 34.45%, which is still obviously higher than that before the DHA Separation (31.43% in Column I). It proves again that learning different property features independently is indeed more powerful.

Unlike the D, H, A maps which represent quite different types of properties, the R, G and B channels all encode color information and are always treated as a whole to represent the color property. The DHA Separation has brought a significant improvement for the object detection. What if we separate the RGB channels (denoted as RGB Separation) like we did on the DHA? The features from R, G, B networks are joint with that from DHA network to feed into the SVM. And the result is shown in Column IV. We can see that the RGB Separation can't boost the performance and the result mean AP^b is even a little lower (31.43–31.09%). It implies that each property should be represented completely and then learned in an independent way.

Our final system lies in Colomn V. The UCM is added as another property map for CNN, besides the RGB, D, H, and A, to provide extra information. The performance is further boosted than that in Column II, and the UCM proves its effectivity to help the object detection. And we get the best mean AP^b 35.99% on the *val* set, exceeding the strongest baseline in Column I by 4.56% (14.51% relative). It is worth noting that, in our method, the dimension of the region features to SVM is increased, and thus overfitting is likely to occur, which is the possible reason for the performance drop on certain specific category, e.g., the category of door shown in Table 2.

4.4. Property investigation

In this part we investigate thoroughly the effect of each property map and their combinations on the object detection. In some cases, we have to trade off between the performance and speed, in which not all of the extra property maps are necessary. After all, more property maps mean high-dimension concatenated features and much training complexity. So we propose a proper order to add the extra property maps (D, H, A, UCM) in turn complementary to the RGB, for the sake of boosting the detection performance.

The result of some combinations are reported on the *val* set of *NYUD2* in Table 3. And the first row shows the suggested order to add the property maps involved here to boost the detection performance, if not all of them are needed: RGB (19.79%), RGB+H (29.22%), RGB+H+A (33.62%), RGB+D+H+A (35.48%), RGB+D+H+A+UCM (35.99%). The mean AP^b are improved step by step, indicating that each of the property maps are beneficial for the detection.

Table 3
Property investigation on *NYUD2 val* set for RGB-D object detection. In each column, not all the extra property maps (D, H, A, UCM) are added as complements to the RGB. All the results are mean AP^b in percentage. See Section 4.4.

Input maps	Mean acc	Input maps	Mean acc	Input maps	Mean acc	Input maps	Mean acc
RGB	**19.79%**	**RGB+H**	**29.22%**	**RGB+H+A**	**33.62%**	**RGB+D+H+A**	**35.48%**
–	–	RGB+D	28.92%	RGB+A+D	33.06%	RGB+H+A+UCM	34.72%
–	–	RGB+A	28.45%	RGB+D+H	32.68%	RGB+A+D+UCM	34.24%
–	–	RGB+UCM	24.60%	RGB+A+UCM	31.75%	RGB+D+H+UCM	33.72%
–	–	–	–	RGB+D+UCM	30.91%	–	–
–	–	–	–	RGB+H+UCM	30.81%	–	–

Table 4
Test set performance on *NYUD2* and *SUN RGB-D* for object detection. We compare the results of our final system with the existing remarkable methods: RGBD DPM, RGB+D CNN, RGB+DHA CNN. All the results are AP^b in percentage. See Sections 4.5.1 and 4.5.2.

Dataset	A_1 ***NYUD2***	A_2	A_3	A_4	B_1 ***NYUD2***	B_2	C_1 ***SUN RGB-D***	C_2
Method	RGBD DPM	RGB+ D CNN	RGB+ DHA CNN	Ours	RGB+ DHA CNN	Ours	RGB+ DHA CNN	Ours
Finetuning set	–	*train*	*train*	*train*	*trainval*	*trainval*	*train*	*train*
bathtub	19.3	39.97	40.99	47.68	49.10	45.50	39.49	44.88
bed	56.0	64.57	68.64	73.53	72.80	74.97	69.47	71.97
bookshelf	17.5	36.38	35.17	38.77	37.81	42.86	28.64	30.79
box	0.6	1.35	2.05	2.49	4.23	5.23	6.33	6.35
chair	23.5	42.04	42.84	49.30	47.85	52.49	37.53	39.17
counter	24.0	41.37	44.46	47.54	49.17	47.49	7.32	9.35
desk	6.2	8.23	13.28	11.76	18.41	15.94	14.96	17.20
door	9.5	18.99	21.91	25.90	23.54	27.25	8.52	8.84
dresser	16.4	20.82	29.43	28.74	33.15	29.98	29.47	32.52
garbage-bin	26.7	29.90	30.28	41.56	42.27	45.53	42.98	42.66
lamp	26.7	34.71	36.39	36.12	38.49	40.53	20.06	21.29
monitor	34.9	42.69	45.54	56.75	54.80	57.81	14.04	14.41
night-stand	32.6	27.45	31.95	47.27	36.16	45.39	37.20	40.71
pillow	20.7	32.68	39.82	42.04	40.58	47.97	14.58	16.45
sink	22.8	37.26	35.47	45.61	41.76	46.10	33.91	35.94
sofa	34.2	43.43	49.44	53.40	51.85	54.63	45.17	49.31
table	17.2	23.45	24.64	28.83	22.47	27.08	39.28	42.37
television	19.5	32.63	40.25	35.58	27.55	37.30	27.33	31.68
toilet	45.1	49.14	54.17	50.47	47.84	51.16	69.49	70.02
mean	23.9	33.00	36.14	**40.18**	38.94	**41.85**	30.83	**32.94**

Note that, if only one extra property map is added, the best performance, i.e. RGB+H (29.22%) is a little lower than the baseline RGB+DHA (31.43%), but when two of them are involved, e.g., RGB+H+A (33.62%), the mean AP^b has already been obviously improved.

4.5. Performance comparision

4.5.1. NYUD2

When it comes to the performance on *test* set of *NYUD2*, we compare our final system with the existing remarkable methods: RGBD DPM, RGB+D CNN and RGB+DHA CNN. The results are given in Table 4.

The Column A_1 shows the result of RGBD DPM from [10,38], which is the state-of-art method before the revival of CNN. And the Column A_2 and Column A_3 are the results of RGB+D CNN and RGB+DHA CNN, which agree with [10] (with no data augmentation). Then the Column A_4 gives the mean AP^b of 40.18% achieved by our system when finetuning the CNNs on the *train* set. We improve the performance by 4.04% over the best baseline (36.14–40.18%, 11.18% relative).

Then we finetune the network again on the *trainval* set. Thanks to more training data, the trained CNN gains higher generalization power. We get the performance shown in Column B_1 and B_2. Our system's result (Column B_2) is still much better than the best baseline (Column B_1), and improves the best mean AP^b on the *test* set of *NYUD2* from 38.94% to 41.85%.

We notice that [25] reported competitive performance with us by adding region features besides the bounding boxes. They expanded the system in [10] in a different way from us. Certainly we can also add region features in our system. However, that is not the point of this paper.

4.5.2. SUN RGB-D

To validate the generalization ability of the proposed method, we further apply our model on the newly challenging dataset *SUN RGB-D*, which contains much more RGB-D pairs than *NYUD2*. We compare the results to that of the existing most remarkable method: RGB+DHA CNN. The performance is shown in Table 4.

It is worth noting that, in the experiments, we filter out the images that have no 2D groundtruth bounding boxes. Thus we finally get 5263 images for *train* set and 5028 images for *test* set for the detection task. The region features is not added either. Besides, we remove the *cabinet* category that is occasionally included in the CNN finetuning on the *NYUD2*.

We finally achieve the mean AP^b of 32.94% (Column C_1), which is much better than the best baseline 30.83% of RGB+DHA CNN (Column C_2). It proves that our method can still work reasonably well with large training data.

4.6. Extensive study

As stated above, our experiments are mainly based on AlexNet and R-CNN, which is similar to the settings in [10]. To make the experiments more comprehensive, here we provide some more results with VGGNet and Fast-RCNN [39], referring to the settings

Table 5
Extensive study on NYUD2 with Fast-RCNN. The results are reported on the *test* set of NYUD2 in mean AP^b. See Section 4.6.

	Model	Mean acc
I	Fast-RCNN (RGB)	38.02
II	Fast-RCNN (RGB+D)	43.71
III	Fast-RCNN (RGB+DHA)	44.75
IV	Fast-RCNN (RGB+D+H+A)	**45.45**

in [40].[3] Specifically, we derive the D, H and A maps using the same way (UCM is not involved here for simplicity), and then take them to learn complementary information to RGB features. There is not SVM training anymore and the class scores generated by different CNNs are averaged before the softmax layer for evaluation. The models are finetuned on the trainval set of NYUD2 and the results are reported on the test set. As shown in Table 5, Row I shows the result with only RGB, while Row II shows the result with RGB and Depth. Rows III and IV are the results with the derived property maps. The D, H and A maps are input into one CNN in Row III and three CNNs in Row IV. The results in Rows III and IV outperform those in Rows I and II, indicating that the derived property maps can still benefit the object detection even with more advanced CNN models and detection frameworks. Besides, the comparison between Rows III and IV suggests that the different property features should be learned separately, which is consistent with the observation on the results with AlexNet and R-CNN.

5. Conclusion

In this paper, we addressed the problem of deeply exploiting the deep information for RGB-D object detection and proposed a novel framework. Specifically, we first derived more properties by mining the provided RGB and depth data. Particularly, several properties that could be directly derived from the color/depth or pairs were adopted here, which included the geometry contour, horizontal disparity, height above ground, and angle with gravity. Then we systematically investigated the fusion schemes of different properties under the CNN model. By the means of analysis and evaluation, it was recommended that, with limited training data, the features encoding the different object properties should be learned independently and fused at the highest level, i.e. not joint until passing into the classifier. Finally, we experimentally verified the effectiveness of the proposed method, which indeed achieved remarkable performance on *NYUD2* and *SUN RGB-D*. And it was gained with no data augmentation or region features. Besides, we only considered the properties that could be computed in relatively straightforward methods. Exploring more useful properties is one of our future works, e.g. equipping more powerful sensors or developing more advanced algorithms for property derivation.

Acknowledgments

This work is supported partially by the National Natural Science Foundation of China under Grant 61673362 and 61233003, Youth Innovation Promotion Association CAS (2017496), and the Fundamental Research Funds for the Central Universities (WK3500000002). We are grateful for the generous donation of Tesla GPU K40 from the NVIDIA corporation.

[3] The RGB features are extracted by VGGNet while the other property features are extacted by AlexNet.

References

[1] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, R. Moore, Real-time human pose recognition in parts from single depth images, Commun. ACM 56 (1) (2013) 116–124, doi:10.1007/978-3-642-28661-2_5.
[2] H. Fu, D. Xu, S. Lin, J. Liu, Object-based rgbd image co-segmentation with mutex constraint, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, doi:10.1109/cvpr.2015.7299072.
[3] Y. Kong, Y. Fu, Bilinear heterogeneous information machine for rgb-d action recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, doi:10.1109/cvpr.2015.7298708.
[4] R. Socher, B. Huval, B. Bath, C.D. Manning, A.Y. Ng, Convolutional-recursive deep learning for 3d object classification, in: Proceedings of Advances in Neural Information Processing Systems, 2012.
[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, doi:10.1109/cvpr.2015.7298594.
[6] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Simultaneous detection and segmentation, in: Proceedings of European Conference on Computer Vision, 2014, doi:10.1007/978-3-319-10584-0_20.
[7] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, doi:10.1109/cvpr.2015.7298965.
[8] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems, 2012.
[9] C. Couprie, C. Farabet, L. Najman, Y. Lecun, Indoor semantic segmentation using depth information, in: International Conference on Learning Representations, 2013.
[10] S. Gupta, R. Girshick, P. Arbeláez, J. Malik, Learning rich features from rgb-d images for object detection and segmentation, in: Proceedings of European Conference on Computer Vision, 2014, doi:10.1007/978-3-319-10584-0_23.
[11] W. Einhauser, P. Konig, Does luminance-contrast contribute to a saliency map for overt visual attention? Eur. J. Neurosci. 17 (2003) 1089–1097, doi:10.1046/j.1460-9568.2003.02508.x.
[12] M. Gazzaniga, R. Ivry, G. Mangun, Cognitive Neuroscience: The Biology of the Mind, (fourth ed.), W. W. Norton, 2013.
[13] H. Saihui, W. Zilei, W. Feng, Deeply exploit depth information for object detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Robust Features for Computer Vision Workshop, 2016.
[14] 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, doi:10.1007/s11263-009-0275-4.
[15] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013, doi:10.1109/cvpr.2013.465.
[16] P. Krahenbuhl, V. Koltun, Efficient inference in fully connected crfs with gaussian edge potentials, in: Proceedings of Advances in Neural Information Processing Systems, 2011.
[17] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645, doi:10.1109/mc.2014.42.
[18] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110, doi:10.1023/b:visi.0000029664.99615.94.
[19] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005, doi:10.1109/cvpr.2005.177.
[20] C. Szegedy, A. Toshev, D. Erhan, Deep neural networks for object detection, in: Proceedings of Advances in Neural Information Processing Systems, 2013.
[21] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Scalable object detection using deep neural networks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014, doi:10.1109/cvpr.2014.276.
[22] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014, doi:10.1109/cvpr.2014.81.
[23] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proceedings of European Conference on Computer Vision, 2014, doi:10.1007/978-3-319-10590-1_53.
[24] H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1996, doi:10.1109/cvpr.1996.517075.
[25] S. Gupta, P. Arbeláez, R. Girshick, J. Malik, Aligning 3d models to rgb-d images of cluttered scenes, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, doi:10.1109/cvpr.2015.7299105.
[26] M. Schwarz, H. Schulz, S. Behnke, Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features, in: Proceedings of International Conference on Robotics and Automation, 2015, doi:10.1109/icra.2015.7139363.
[27] F. Husain, H. Schulz, B. Dellen, C. Torras, S. Behnke, Combining semantic and geometric features for object class segmentation of indoor scenes, IEEE Robot. Autom. Lett. 2 (1) (2017) 49–55, doi:10.1109/lra.2016.2532927.

[28] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, J. Malik, Multiscale combinatorial grouping, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014, doi:10.1109/cvpr.2014.49.
[29] P. Dollár, C.L. Zitnick, Structured forests for fast edge detection, in: Proceedings of IEEE International Conference on Computer Vision, 2013, doi:10.1109/iccv.2013.231.
[30] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 898–916, doi:10.1109/TPAMI.2010.161.
[31] S. Gupta, P. Arbelaez, J. Malik, Perceptual organization and recognition of indoor scenes from rgb-d images, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013, doi:10.1109/cvpr.2013.79.
[32] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoorsegmentation and support inference from rgbd images, in: Proceedings of European Conference on Computer Vision, 2012, doi:10.1007/978-3-642-33715-4_54.
[33] S. Song, S.P. Lichtenberg, J. Xiao, Sun rgb-d: a rgb-d scene understanding benchmark suite, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015.
[34] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE transactions on pattern analysis and machine intelligence 39 (12) (2017) 2481–2495, doi:10.1109/TPAMI.2016.2644615.
[35] A. Kendall, V. Badrinarayanan, R. Cipolla, Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding, in: Proceedings of the British Machine Vision Conference, 2017.
[36] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R.B. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of ACM International Conference on Multimedia, 2014, doi:10.1145/2647868.2654889.
[37] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: a library for large linear classification, J. Mach. Learn. Res. 9 (2008) 1871–1874.
[38] B.-s. Kim, S. Xu, S. Savarese, Accurate localization of 3d objects from rgb-d data using segmentation hypotheses, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013, doi:10.1109/cvpr.2013.409.
[39] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015.
[40] S. Gupta, J. Hoffman, J. Malik, Cross modal distillation for supervision transfer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

Saihui Hou received the B.S. from University of Science and Technology of China (USTC), Hefei, China, in 2014. Then he continues to pursue the Ph.D. degree in the USTC until now. His research interests include deep learning, object detection and fine-grained classification.

Zilei Wang is currently an Associate Professor with the lab of Advanced Sensing and Control as the leader of Vision and Multimedia (VIM) research group, University of Science and Technology of China (USTC), Hefei, China. His research interests include media streaming systems and cloud, network management and control, computer vision and machine learning.

Feng Wu is currently a Professor in the University of Science and Technology of China (USTC) as the president of School of Information Science and Technology. He has published more than 50 papers on top journals and conferences, such as IEEE TCSVT, IEEE TIP, IEEE CVPR and so on. His research interests include video and image coding, large-scale image processing, computer vision and machine learning.