Pattern Recognition 64 (2017) 437–445

Contents lists available at ScienceDirect

Pattern Recognition

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

Hierarchically Supervised Deconvolutional Network for Semantic Video Segmentation

Yuhang Wang[a,b], Jing Liu[a,*], Yong Li[a,b], Jun Fu[a,b], Min Xu[c], Hanqing Lu[a]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[b] University of Chinese Academy of Sciences, Beijing, China
[c] University of Technology, Sydney, NSW, Australia

ARTICLE INFO

Keywords:
Semantic video segmentation
Deconvolutional neural network
Coarse-to-fine training
Spatio-temporal consistence

ABSTRACT

Semantic video segmentation is a challenging task of fine-grained semantic understanding of video data. In this paper, we present a jointly trained deep learning framework to make the best use of spatial and temporal information for semantic video segmentation. Along the spatial dimension, a hierarchically supervised deconvolutional neural network (HDCNN) is proposed to conduct pixel-wise semantic interpretation for single video frames. HDCNN is constructed with convolutional layers in VGG-net and their mirrored deconvolutional structure, where all fully connected layers are removed. And hierarchical classification layers are added to multi-scale deconvolutional features to introduce more contextual information for pixel-wise semantic interpretation. Besides, a coarse-to-fine training strategy is adopted to enhance the performance of foreground object segmentation in videos. Along the temporal dimension, we introduce Transition Layers upon the structure of HDCNN to make the pixel-wise label prediction consist with adjacent pixels across space and time domains. The learning process of the Transition Layers can be implemented as a set of extra convolutional calculations connected with HDCNN. These two parts are jointly trained as a unified deep network in our approach. Thorough evaluations are performed on two challenging video datasets, i.e., CamVid and GATECH. Our approach achieves state-of-the-art performance on both of the two datasets.

1. Introduction

Semantic video segmentation is a fundamental problem in video interpretation which assigns label to each pixel in video sequences. It attracts much attention for the wide applications such as automatic drive, scene understanding and robotics. Compared with general video segmentation tasks that just parse video frames into spatio-temporal volumes, semantic video segmentation demands higher on understanding the content of frames and makes more abundant category judgment.

Much of previous work on semantic video segmentation is based on graphical models, which links adjacent patches in space and time domains. And the label of each supervoxel is assigned by energy minimization and label propagation [1–5]. These methods take the spatial and temporal consistency between video frames into account. But the tremendous computation cost and limited discrimination ability of graph-based algorithms make them hard to be applied to large-scale video data. Some recent work turn to deep learning as a new solution to this problem considering its excellent learning capacity [6,7]. Zhang et al. [6] improve the graph-based method with deep features and Tran et al. [7] empoly a 3D convolutional neural network (3D CNN) [8] to implement pixel-wise labeling in video tasks. Both of these work improve traditional methods with 3D CNN, but as the amount of annotated video data available for model training is small, the performance of these methods is also limited.

As a relevant visual task, semantic image segmentation obtains great progress because of the introduction of deep learning methods and large amount of available training data. The most representative work is FCN [9] which provides an end-to-end network enabling pixel-wise category prediction with a whole image as input. However, the pixel-wise supervision in FCN is attached to the feature maps upsampled via large-span bilinear interpolation, which results in over-smoothed object boundaries in the results of segmentation. Besides, the foreground objects, especially the tiny objects, are possible to be overwhelmed by the large areas of background.

To overcome the problems existing in previous work, we propose a unified framework to effectively use the spatial and temporal information for semantic video segmentation. For a video sequence, each frame is fed into a deep network along with its adjacent frames, and they are conducted in parallel. And then the responses of the frames are

* Corresponding author at: Institute of Automation Chinese Academy of Sciences, National Laboratory of Pattern Recognition, Beijing, China

http://dx.doi.org/10.1016/j.patcog.2016.09.046
Received 16 March 2016; Received in revised form 26 September 2016; Accepted 28 September 2016
Available online 29 September 2016

combined across space and time domains with a set of learned state transition matrices between semantic concepts. The entire process can be integrated into a unified deep network and trained jointly.

In our approach, the video frames are firstly processed as separated images. In this way, it becomes possible for us to utilize models pretrained with large-scale image data like ImageNet [10] and effectively transfer them to video analysis. To solve the limitations of FCN [9], we design a hierarchically supervised deconvolutional neural network (HDCNN) to reconstruct the low resolution features to input resolution through a more refined method and maintain the boundary information as much as possible. HDCNN is constructed on the top of convolutional layers in VGG-net with a mirrored deconvolutional and unpooling architecture, while the fully connected layers are removed. Pooling indices in the convolutional part are employed during unpooling to ensure location consistency, which conducts a more refined upsampling operation than bilinear interpolation. Besides, hierarchical classification layers are presented by mapping multi-scale intermediate deconvolutional features to pixel-wise labeling maps, where unpooling and bilinear interpolation are used cooperatively. As the foreground objects in video frames are always our focus during video interpretation, to prevent them from being overwhelmed by the backgrounds and produce more refined boundaries, a coarse-to-fine training strategy is designed. A primary HDCNN is firstly trained on the entire frames which produces coarse segmentation results. And secondly, object proposals are generated to further refine the model with more object details. In this way, we bring in scale invariance as the large and tiny objects are rescaled to similar resolution with the background scenes, which improves the discrimination ability of HDCNN.

Then we fuse the interframe information for video segmentation. A pixel in a video sequence is always related to its spatially and temporally adjacent pixels. For example, a pixel is more likely to be labeled as “pedestrian” than “car” if its adjacent pixel is labeled as “side walk”. Therefore, the semantic correlationship can be mined within local spatio-temporal volumes to further improve pixel-wise prediction. Aimed at this, we learn a set of state transition matrices on semantic concepts to combine the network responses of pixels in a spatio-temporal volume. The combination process can be implemented as convolutional operations with a set of extra layers. We call them “Transition Layers” and integrate them into a unified network with HDCNN, which makes a jointly optimized system.

In this paper, we evaluate our approach mainly on street scenes, and two challenging video datasets CamVid [11] and GATECH [12] are employed. State-of-the-art results are achieved on both of them, which verify the effectiveness of our approach.

The remainder of this paper is organized as follows: In Section 2, we overview some related work. Section 3 elaborates the proposed models of HDCNN and Transition Layers. The experimental evaluations and discussions are presented in Section 4. And we conclude this paper in Section 5.

2. Related work

2.1. Semantic video segmentation

Previous methods for semantic video segmentation are usually graph-based [1–5]. Badrinarayanan et al. [1,2] solve the problem with semi-supervised methods assuming that only a few key frames are labeled and propagating the labels to other frames with motion flows. Liu et al. [3] jointly model supervoxel labeling with object instances and their geometric relations. Floros et al. [4] bring in 3D scene geometry to improve segmentation quality. While Jain et al. [5] proposed a multi-grained graph learning strategy for semantic video segmentation. In these work, motion or tracking information is also used to help guarantee the spatial and temporal consistency in a long-range video sequence. Tighe et al. [13] extend their image parsing approach to video sequences. However, the performance of these methods are always limited by the computation cost and discrimination ability of their models.

As deep learning brings huge progress in visual tasks, some researchers attempt to solve the problem with deep models [6,7]. Zhang et al. [6] improve their performance by learning discriminative hierarchical features with a 3D CNN. Tran et al. [7] also employ a 3D CNN to encode video sequences and make dense predictions for them in a similar framework with FCN [9]. The 3D CNN [8] processes video data with an extra temporal channel in its convolution kernels and combine the information in different frames by convolving pixels at corresponding positions together. However, as the available video data for 3D CNN training is relatively limited and lack of diversity, the pre-trained 3D CNN models are always disadvantaged in discrimination and generalization performance compared with those 2D CNN models trained with large-scale image datasets. This is also the reason why we choose to process video frames as single images in the first part of our framework.

2.2. Semantic image segmentation

The introduction of deep learning contributes to the recent breakthrough in semantic image segmentation, represented by FCN [9]. FCN extends a convnet to adapt arbitrary-sized inputs with fully convolutional structure, and realises pixel-wise prediction by upsampling the output feature maps to the input resolution with bilinear interpolation, which results in rough edges and object vanishing. Much following work tries to alleviate these problems, and the most representative practice is combining FCN with conditional random field (CRF) or markov random field (MRF) [14–17]. Moreover, deconvolutional neural network inspires us with a finer method for upsampling than direct bilinear interpolation. It is firstly proposed by Zeiler et al. [18] and later used for visualization [19]. It reverses the process of convolutional neural network and maps the feature activities back to each pixel. This method is firstly applied in object localization and segmentation by Simonyan et al. [20] with a non-learning based framework. Noh et al. [21] further proposed Deconvnet for semantic image segmentation and extend their method to semi-supervised scenarios [22]. Badrinarayanan et al. [23] also proposed SegNet with the similar structure. However, the existing methods with deconvolutional neural network always suffer from burdensome model size or training effort. And we bring in different features with them in our detailed network structure.

3. Our approach

3.1. Overview

The overall framework of our approach is illustrated in Fig. 1. It is implemented as a unified deep network, which can be further divided into two components. The first one is HDCNN which is a deconvolutional neural network consisting of convolutional and deconvolutional parts. It encodes the input frame with convolutional layers to extract the semantic information, and then decodes the feature maps to the input resolution with deconvolutional layers to reconstruct detailed boundaries. Hierarchical supervision is also used to assist training. The second component is Transition Layers which are implemented as a set of extra convolutional layers, it combines the responses of a video frame with those of its adjacent frames by considering the semantic correlation of spatially and temporally neighbouring pixels. With this network structure, each video frame is fed into the network along with its previous and latter frames, the frames are conducted in parallel in HDCNN. Their corresponding output feature maps are fed into Transition Layers and combined via the state transition matrices to make pixel-wise prediction for the current frame..

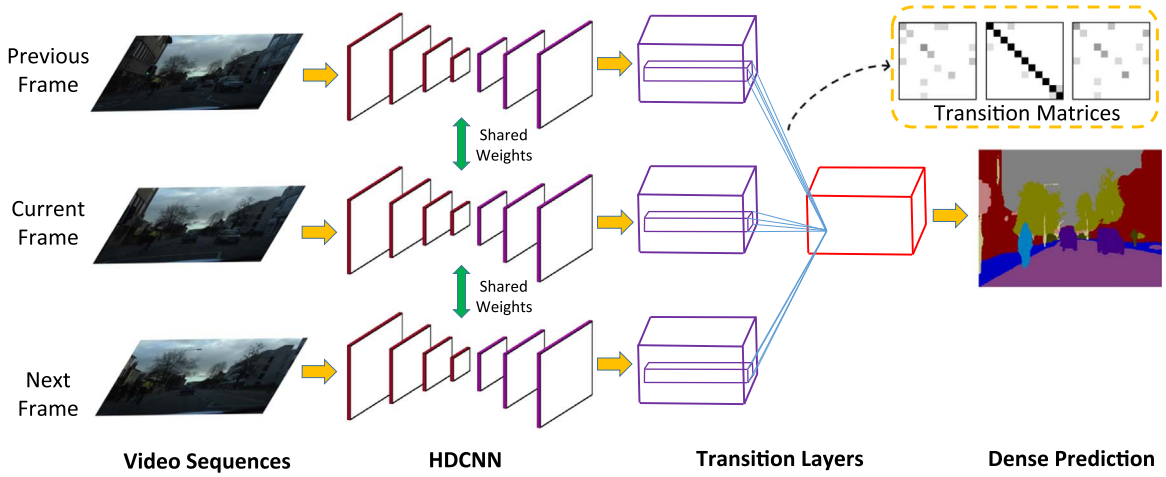

Fig. 1. Overall framework of our approach. The input video sequences are firstly fed into HDCNN in parallel. And then the output feature maps are combined across space and time domains to make dense prediction for the current frame with a set of transition matrices, which are implemented as Transition Layers. The two parts are linked together and jointly trained as a unified deep network.

3.2. HDCNN

Unlike FCN [9] which obtains pixel-wise prediction through large-span bilinear interpolation, HDCNN provides a more refined reconstruction via progressively upsampling the feature maps to larger resolution. Illustration of our network structure is shown in Fig. 2..

3.2.1. Architecture

HDCNN is composed of two parts. The first part is a convolutional network that takes a 2D image as input and encodes it into feature maps. And the second part is a deconvolutional network that takes the feature maps as input and propagates the responses back to each pixel. The two networks are spliced and optimized together. For the convolutional network, we directly inherit the network structure and parameters from VGG 16-layer net [24]. The deconvolutional network is built following [19], which is a reverse process of the convolutional network. The main components of the deconvolutional network are deconvolutional layers and unpooling layers.

Deconvolutional Layer: The deconvolutional layer takes the opposite position to the convolutional layer. The convolutional layer aggregates feature vectors in a local region and map them into a single response, while the deconvolutional layer attempts to disassemble the response back to each individual position. We use the learned convolutional filters to initialize their corresponding deconvolutional layers in our work, but flip each filter vertically and horizontally, as Zeiler et al. did in [19].

Unpooling Layer: The unpooling layer carries out the inverse process of the pooling layer. In our work, we use max pooling in the convolutional network, which means that only the max value in a feature map region is reserved. Through the unpooling layer, we send the response value back to its original position and supplement 0 for the abandoned positions.

All the convolutional and deconvolutional layers are placed symmetrically, and the same applied to the pooling and unpooling layers. Through out the deconvolutional part of the network, the resolution of convolutional feature maps are restored hierarchically, with precise position correspondence.

3.2.2. Network details

Besides, our network is designed with several unusual features, which make it small-sized and easy to converge.

Parameter Inheritance: We use the parameters of the learned convolutional layers to initialize the deconvolutional layers in our network, rather than Gaussian random numbers. It supplies our

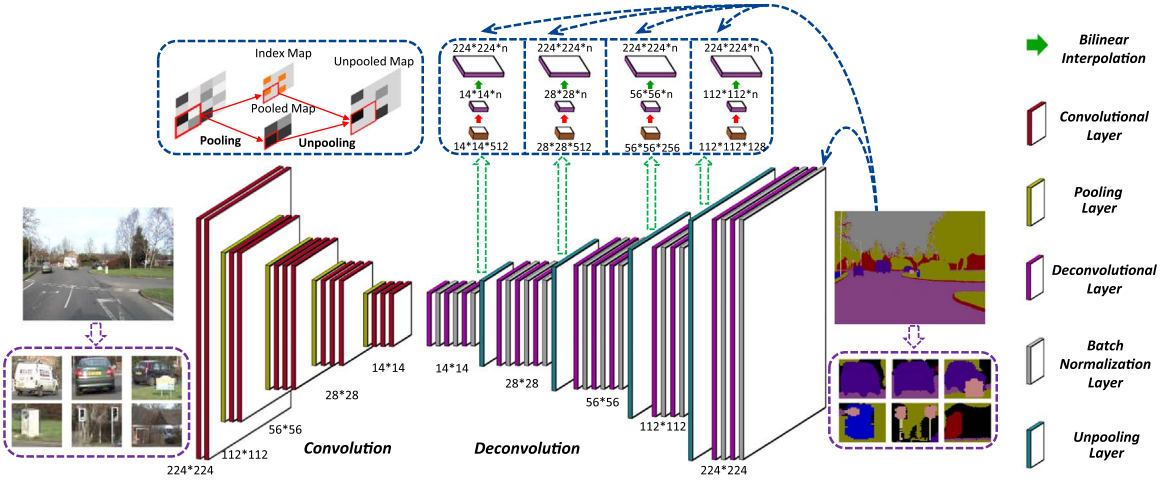

Fig. 2. Network structure of the proposed HDCNN, which is composed of convolutional and deconvolutional parts. Blocks with different colors indicate different kinds of layers. The parameters in the deconvolutional layers are inherited from the learned convolutional layers as initialization. Furthermore, we remove the fully connected layers and add batch normalization layers only after the deconvolutional layers. Moreover, each deconvolutional layer ahead of unpooling layer is connected with a classification layer and then upsampled to the input resolution of the network, which makes hierarchical prediction and supervision by cooperatively using unpooling and bilinear interpolation (n stands for the number of categories). A coarse-to-fine training strategy is applied with entire images and object proposals to train the network respectively.

network with applicable initial values during deconvolutional computation. And this treatment makes our network easy to converge.

Discarding Fully Connected Layers: We discard the fully connected layers and reverse the net from the last convolutional layer, as the convolutional layer always shows much better ability for reserving spatial information compared with fully connected layer. And with this design, we decrease the model size by nearly 10 times, which makes our network more lightweight and easier to train because of the fewer parameters.

Asymmetrical Batch Normalization: We employ batch normalization [25] in our network to reduce the internal-covariate-shift during training, but we only add it in the deconvolutional part of HDCNN. In our framework, the convolutional network works as a feature extractor which encodes semantic information, and we don't want to break the correlations between the well-learned net layers. While the deconvolutional network is trained as a decoder, so we add a batch normalization layer after each deconvolutional layer for better optimization.

Hierarchical Predictions: Although with strict positional correspondence, the unpooling layers tend to break some correlations within image regions for the reason of zero supplementary. To alleviate this, we connect a classification layer to each deconvolutional layer that is ahead of unpooling layer, and supervise it with pixel-wise groundtruth. In this way, we attempt to guarantee the discrimination and region correlations of our model before every unpooling. And the pixel-wise prediction is realized by feeding the deconvolutional feature maps into a fully convolutional classification layer and upsampling them with bilinear interpolation. Meanwhile, it can be found that, with the hierarchical predictions, we are actually cooperatively using unpooling and bilinear interpolation for resolution recovery. That is, the lower layer is attached greater scale change with bilinear interpolation, while the higher layer considers less scale change but more unpooling operations, and there exists a balance for these two kinds of upsampling methods.

3.2.3. Coarse-to-fine training

During the encoding and decoding process of neural networks, the foreground objects are easy to be confused with each other, and the tiny objects are easy to be overwhelmed by the backgrounds because of the relatively low resolution. However, the objects are usually what we concern with in video interpretation. To make the model more focused on the objects, we train HDCNN with a two-stage strategy to implement a coarse-to-fine learning process.

In the first stage, we train HDCNN with the entire frames and their corresponding groundtruth as input and supervision. A primary model is obtained in this stage which makes coarse segmentations.

In the second stage, we sample object proposals to refine the primary model obtained. To better understand the content of video frames, the proposal sampling is mainly conducted on classes such as “car”, “pedestrian/human”, etc, which are considered as “foreground objects”. Edge-box [26] is used to generate candidate bounding boxes for each video frame. And then the object proposals are selected from the candidates according to their IoU (intersection-over-union) scores with the groundtruth of the foreground objects. Some objects with relatively poor segmentation results in the first stage are given more emphasis during selection. Afterwards, the selected object proposals are rescaled and used to refine the primary HDCNN with more object details. Through this processing, we also bring in scale invariance for objects and scenes during rescaling which improves the discrimination ability of our model.

3.3. Transition layers

3.3.1. Transition learning

As the category judgement of a pixel can be influenced by its spatially and temporally adjacent pixels, we learn a set of state transition matrices to describe the semantic correlations between different concepts across space and time domains. Given a pixel $P(x, y, t)$, where x, y and t indicate its spatial and temporal coordinates respectively. Its adjacent pixels can be represented as $\widetilde{P}(\tilde{x}, \tilde{y}, \tilde{t})$, where $\tilde{x} \in [x-m, x+m]$, $\tilde{y} \in [y-n, y+n]$, $\tilde{t} \in [t-l, t+l]$. For the given pixel, its state can be represented as $S^{\boldsymbol{P}}$, which is a C dimensional vector with each element $S_j^{\boldsymbol{P}}$ of it indicates the response on the jth semantic category. Assumpting the influence factor of $S_i^{\widetilde{P}}$ to $S_j^{\boldsymbol{P}}$ is $w_{ij}^{\widetilde{P}}$, then we can get the total influence from $S^{\widetilde{P}}$ to $S_j^{\boldsymbol{P}}$ as:

$$I_j^{\widetilde{P}} = \sum_{i=1}^{C} S_i^{\widetilde{P}} \times w_{ij}^{\widetilde{P}} = S^{\widetilde{P}} \times (v_j^{\widetilde{P}})^T \tag{1}$$

where $v_j^{\widetilde{P}} = [w_{1j}^{\widetilde{P}}, \ldots, w_{Cj}^{\widetilde{P}}]$. Thus the state transition matrix from $\widetilde{P}$ to P can be obtained as:

$$W^{\widetilde{P}} = [(v_1^{\widetilde{P}})^T, \ldots, (v_C^{\widetilde{P}})^T] \tag{2}$$

with the entire influence as:

$$I^{\widetilde{P}} = S^{\widetilde{P}} \times W^{\widetilde{P}} \tag{3}$$

Taking all the adjacent pixels of P into consideration, as well as itself, we can get the final state of P as:

$$\hat{S}^P = \sum_{\tilde{t}=t-l}^{t+l} \sum_{\tilde{y}=y-n}^{y+n} \sum_{\tilde{x}=x-m}^{x+m} I^{\widetilde{P}} = \sum_{\tilde{t}=t-l}^{t+l} \sum_{\tilde{y}=y-n}^{y+n} \sum_{\tilde{x}=x-m}^{x+m} S^{\widetilde{P}} \times W^{\widetilde{P}} \tag{4}$$

In this way, we achieve the response of P on each semantic category considering the transition relation of all its spatially and temporally adjacent pixels, and then we use $\hat{S}^P$ to predict the label of P via softmax.

3.3.2. Convolutional implementation

As for implementation, we accomplish this process through a set of extra convolutional layers, which are jointly trained with HDCNN. As mentioned in Section 3.2.2, we employ hierarchical classification layers in HDCNN. While before the joint training with Transition Layers, a layer selection is conducted on the validation set and network branches above the best classification layer are discarded, which are the redundant classification layers as well as their corresponding deconvolutional layers.

With frames of time $\tilde{t} \in [t-l, t+l]$ as input, a set of output feature maps can be obtained, which are of the same resolution as input frames with C channels. And we can get $S^{\boldsymbol{P}}$ from the feature maps with corresponding spatial and temporal location. The calculating process of $\sum_{\tilde{y}=y-n}^{y+n} \sum_{\tilde{x}=x-m}^{x+m} S^{\widetilde{P}} \times W^{\widetilde{P}}$ with a fixed $\tilde{t}$ can be implemented in a convolutional layer with the feature maps of frame $\tilde{t}$ as input. The convolutional kernels of such a layer is of size $(2m+1) \times (2n+1) \times C$ with kernel number equal to C. The output is a subcomponent of $\hat{S}^P$, we represent it as $\hat{S}^{P,\tilde{t}}$. Thus the computation process can be represented as:

$$\hat{S}_j^{P,\tilde{t}} = F^{P,\tilde{t}} * K_j^{\tilde{t}} \tag{5}$$

where “*” represents the convolutional operation. $F^{P,\tilde{t}}$ stands for the spatial neighbourhood of P in the feature maps of frame $\tilde{t}$. It contains all the $S^{\widetilde{P}}$ with $\widetilde{P}$ ranging within $\tilde{x} \in [x-m, x+m]$, $\tilde{y} \in [y-n, y+n]$ and a fixed $\tilde{t}$. $K_j^{\tilde{t}}$ stands for the jth convolutional kernel, and its parameters at each spatial location are assigned with corresponding $v_j^{\widetilde{P}}$.

With the same operation applied to feature maps of frame $\tilde{t} \in [t-l, t+l]$, $(2l+1)$ extra convolutional layers are connected to HDCNN, and we obtain the final state of pixel P by summing the output of them all:

$$\hat{S}_j^P = \sum_{\tilde{t}=t-l}^{t+l} F^{P,\tilde{t}} * K_j^{\tilde{t}} \tag{6}$$

In this way, we implement the Transition Layers with a set of

Table 1
Results of hierarchical predictions on CamVid val set.

Predictions	224 × 224	112 × 112	56 × 56	28 × 28	14 × 14
Class avg	86.4	**86.5**	85.8	83.5	81.6
Global avg	91.1	**91.6**	91.4	90.1	88.1
Mean IoU	66.4	**67.5**	66.9	63.7	60.4

parallel convolutional layers and the output $\hat{S}^P$ is used for category judgment. We initialize the parameters $w_{ij}^{\tilde{P}}$ with 1 where $\tilde{x} = x$, $\tilde{y} = y$, $\tilde{t} = t$, i=j, while the other $w_{ij}^{\tilde{P}}$ are initialized with 0. With this setting, we actually close up all the state transition at the beginning of training. And then we jointly train the Transition Layers with HDCNN as a unified deep network using standard stochastic gradient descent (SGD) for optimation and pixel-wise softmax as loss function.

4. Experiment

Our experiments are mainly conducted on two public datasets, CamVid [11] and GATECH [12]. Effectiveness of each part of our framework is discussed in details. Performance of the overall framework is compared with state-of-the-art methods and we win the best results on both of the two datasets. We also conduct an extra experiment on PASCAL VOC 2012 dataset [27] for semantic image segmentation, to verify the performance of the proposed HDCNN.

CamVid: The CamVid [11] dataset consists of 5 video sequences with street scenes. The videos are taken in daytime and at dusk with resolution 960 × 720. Totally 701 densely labeled frames are provided belonging to 11 semantic categories, i.e., sky, building, tree, side-walk, car, column-pole, fence, pedestrian, bicylist and sign-symbol. Following [23], we split the dataset into training, validation and test sets, with "video/frame" number of each set being 3/367, 1/101 and 2/233 respectively.

GATECH: The GATECH [12] dataset is a big video set of outdoor scenes with frame resolution varying from 320 × 480 to 600 × 800. It contains 101 labeled videos which are publicly partitioned into two sets: 63 videos with 12241 frames for training and 38 videos with 7071 frames for testing. The dataset is labeled with 8 semantic categories taking geometry into consideration, which are sky, ground, solid, porous, cars, humans, vertical mix, and main mix.

PASCAL VOC 2012: We conduct our experiment on the PASCAL VOC 2012 dataset [27] with extended annotations from [28], following the same setting as [9,21,23]. The extended dataset is labeled with 20 object categories and one background category, containing 10582 training images, 1449 validation images and 1456 testing images.

Evaluation Metrics: We take Class avg (Percentage of correctly classified pixels in a class, and then averaged over all classes), Global avg (Percentage of correctly classified pixels over the whole dataset) and Mean IoU (Ratio of correctly classified pixels in a class over the union set of pixels predicted to this class and groundtruth, and then averaged over all classes) as our metrics to evaluate our models [29].

4.1. Implementation details

We implement our network with Caffe [30] on the pre-trained VGG 16-layer net [24]. For the training of HDCNN on CamVid and GATECH, the initial learning rate, momentum and weight decay used for standard stochastic gradient descent (SGD) are set to 1.25e-4, 0.99 and 0.0005 respectively. While for PASCAL VOC 2012, we use a smaller initial learning rate as 2.5e-5. The networks are all trained with batchsize set to 1, and the video frames as well as the object proposals are all rescaled to 224 × 224 for input. For the joint training of Transition Layers and HDCNN, we chose a small time span of 3 frames for both CamVid and GATECH, i.e., the pixel-wise prediction for a video frame is made by combining the feature maps of its previous and next frames. The three frames are input into HDCNN as a triplet that share the network parameters, and the entire network is optimized with the triplet batchsize equal to 1. CRF [14] is also employed as a post-processing of our approach.

4.2. Classification layer selection

HDCNN is connected to hierarchical classification layers by cooperatively using unpooling and bilinear interpolation, as discussed in Section 3.2.2. When we reconstruct the network up to the input resolution, we will get 5 segmentation predictions from different deconvolutional layers with ×16, ×8, ×4, ×2, ×1 upsampling via bilinear interpolation respectively, while the upsampling ratio of unpooling is just the inverse. Thus, given an image of size 224 × 224, we indicate each of the 5 segmentation predictions with the sizes of their corresponding feature maps, which are 14 × 14, 28 × 28, 56 × 56, 112 × 112, 224 × 224, respectively. We evaluate our HDCNN model on the validation set of CamVid dataset and list the performance of each prediction in Table 1.

It can be found that, for all the three metrics, the best results appear in the 112 × 112 layer. While the 224 × 224 layer, which is upsampled to the input resolution all by unpooling, achieves poorer performance. So as to the 14 × 14 layer which recovers resolution only with bilinear interpolation. The results prove the effectiveness of cooperatively using unpooling and bilinear interpolation. As two different upsampling methods, bilinear interpolation reserves local consistency but is easy to over smooth object edges, while unpooling restores more refined contexture features but tends to break up the correlations within local regions. However, the results of 112 × 112 layer benefit from an appropriate proportion for each of them, which are achieved through three unpooling layers followed by ×2 bilinear interpolation.

During the joint training stage, the output feature maps of the 112 × 112 classification layer are upsampled via bilinear interpolation and connected to the Transition Layers, while the redundant layers are all discarded. And in the following experiments, for HDCNN, we report the results of 112 × 112 layer for comparison. As there is no validation set in GATECH, we follow the same layer selection settings as in CamVid.

4.3. Visualization of transition layers

To confirm the effect of Transition Layers, we visualize the learned transition matrices in this section, which are trained on CamVid dataset with 11 semantic concepts. For concise visualization, we just show as examples the transition matrices from pixel $\tilde{P}(x, y, t-1)$ and $\tilde{P}(x, y, t+1)$ to $P(x, y, t)$ as well as the transition matrix of $P(x, y, t)$ itself, which describe how a pixel is influenced by the pixels in its previous and next frames with the same spatial location as well as itself. As shown in Fig. 3, the transition matrices are visualized in grids with the jth column represents the influence factors of $\tilde{P}$ to the jth category of P. And the intensity of each block indicates the value of corresponding influence factor, where the factor with the largest absolute value is colored with black and the smallest one is colored with white. During visualization, the positive and negative influence factors are shown separately and the factors close to 0 are removed and replaced with white blocks because of their slight influences..

It can be noticed in Fig. 3 that the transition matrices are sparse and show obvious correlations between different semantic concepts. The correlations are data-driven and affected by the spatial relationships and visual similarities of objects. For detailed analysis, the positive transition matrix of current frame shows obvious diagonal features and possesses the highest factors, which proves that the final state of a pixel is mainly influenced by the responses of itself, while the adjacent pixels work as assistance. And it is interesting that in this matrix the influence factor of "pedestrian" to "bicyclist" and that of

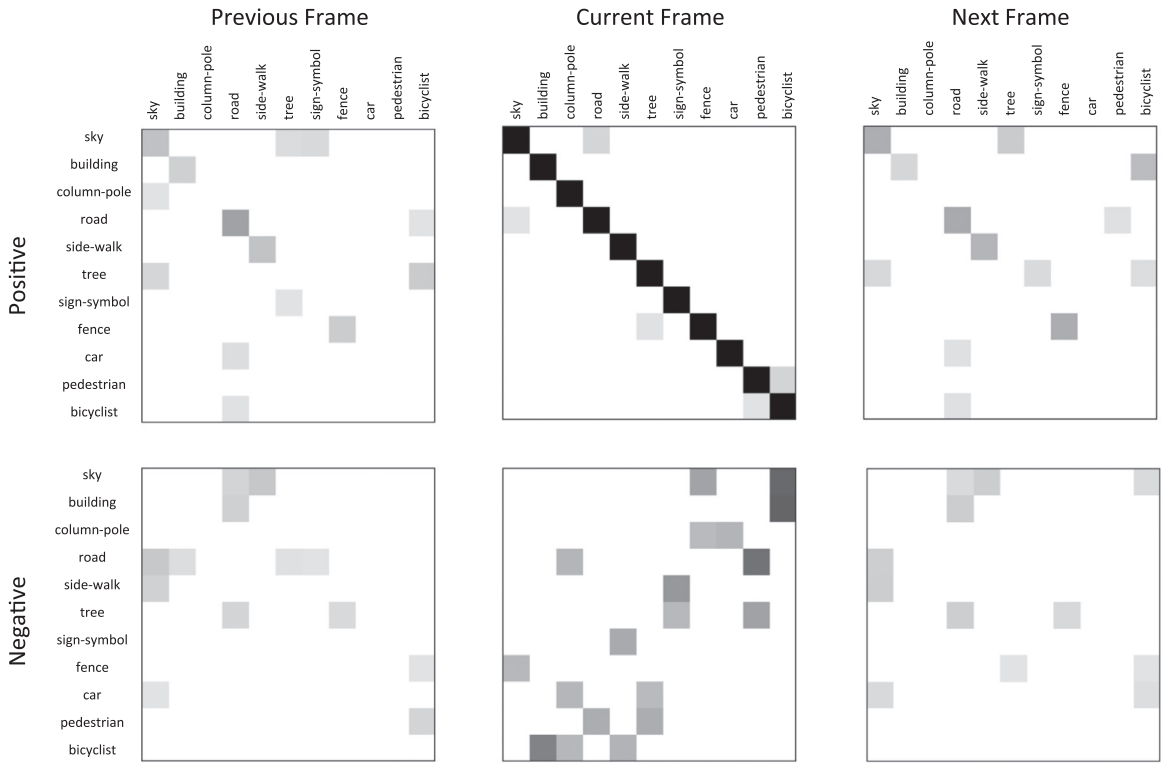

Fig. 3. Visualization of transition matrices learned on CamVid dataset. Only the transition matrices between a pixel in current frame and the corresponding pixels in its previous and next frames with the same spatial location are visualized, as well as the transition matrix of itself.

"bicyclist" to "pedestrian" are relatively high, which indicates the easy confusion of these two concepts in current frame because of the similar semantic meanings and visual features. While correspondingly, the transition matrices of previous and next frames all show high responses on the column of "bicyclist", no matter the positive or the negative ones, which helps to distinguish these two concepts. And there are also some other features in the matrices that show the spatial or cooccurrence relations between different concepts. For example, in the positive transition matrices of both previous and next frames, the influence factors of "car" and "bicyclist" to "road" are high, which indicates that, with the corresponding pixels in the previous or next frames predicted to be "car" or "bicyclist", the pixel in current frame will possess a higher possibility to be predicted as "road".

4.4. Performance on CamVid

Performance of our approach is compared with some of the state-of-the-art methods on the test set of CamVid [11], as shown in Table 2. Prediction accuracy of each class is reported as well as the Class avg, Global avg, and Mean IoU results. Our methods win the best results on all the three metrics.

Among the compared methods, Superparsing [13] and Liu et al. [31] are both graph-based methods which take the sequential relationship of video frames into consideration. While SegNet [23] and Bayesian SegNet [32] are recently proposed methods solving the problem with deep networks. But both SegNet and Bayesian SegNet treat the video frames as independent images with no temporal information used. For our approach, detailed comparison with different settings are provided. Firstly, we provide the performance of the coarsely trained HDCNN where no object proposal is used for training. And then, we add object proposals to accomplish the coarse-to-fine training process of HDCNN. Finally, we provide the results of our entire framework with three adjacent frames combined for the prediction of current frame. Furthermore, we also evaluate the results with

Table 2
Comparison with state-of-the-art methods on CamVid test set.

Methods	Superparsing[13]	Liu et al. [31]	SegNet[23]	Bayesian SegNet[32]	HDCNN-224 (no proposal used)	HDCNN-224	HDCNN-224 +TL	HDCNN-448	HDCNN-448 +TL
Building	87.0	75.4	88.8	80.4	83.9	82.4	85.6	85.4	**89.7**
Tree	67.1	81.3	87.3	85.5	82.0	86.1	83.5	**87.4**	86.9
Sky	**96.9**	95.7	92.4	90.1	95.0	94.9	96.3	96.7	95.0
Car	62.7	70.0	82.1	86.4	84.1	**89.5**	87.0	88.7	87.9
Sign-symbol	30.1	52.1	20.5	67.9	50.5	56.0	57.6	**68.4**	62.0
Road	95.9	95.1	**97.2**	93.8	95.6	94.5	95.3	95.9	95.8
Pedestrian	14.7	61.6	57.1	73.8	77.0	82.6	82.2	**85.7**	84.7
Fence	17.9	34.6	49.3	**64.5**	44.1	47.5	48.5	48.4	48.8
Column-pole	1.7	17.9	27.5	50.8	37.1	44.9	40.7	48.7	**52.6**
Side-walk	70.0	62.0	84.4	91.7	89.6	92.3	**93.1**	91.3	92.8
Bicyclist	19.4	46.0	30.7	54.6	71.6	71.8	**77.0**	69.5	71.0
Class avg	51.2	62.8	65.2	76.3	73.7	76.6	77.0	78.7	**78.8**
Global avg	83.3	81.8	88.5	86.9	87.9	88.4	89.3	90.0	**90.9**
Mean IoU	–	–	55.6	63.1	58.5	59.7	60.7	64.4	**65.6**

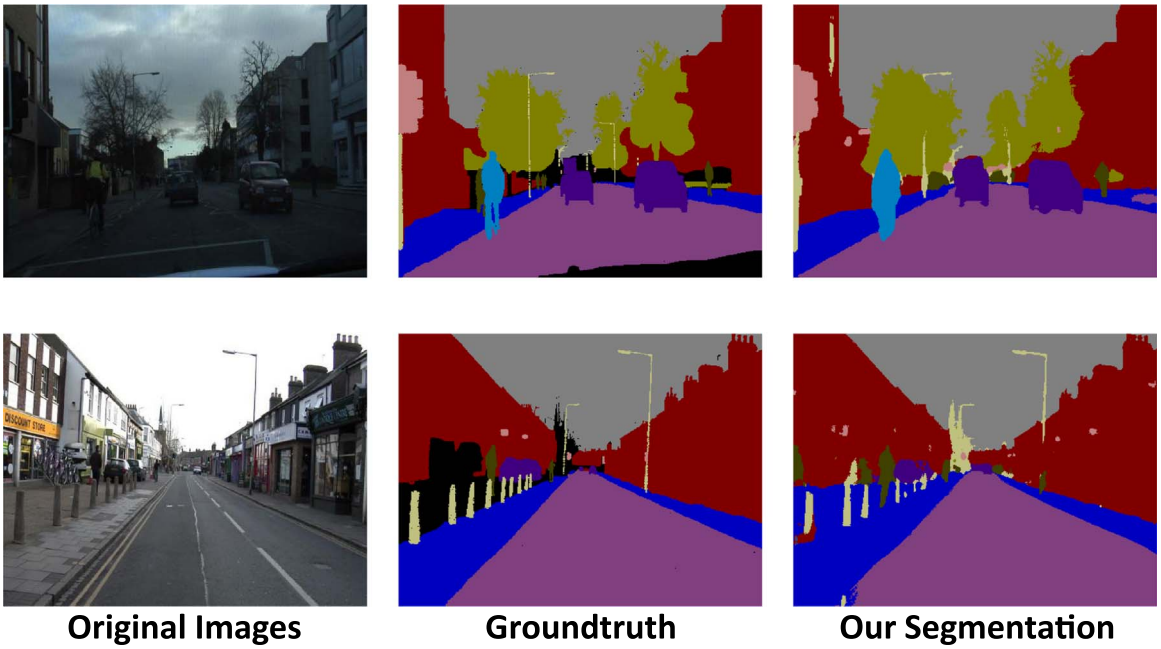

Fig. 4. Our segmentation results on CamVid dataset.

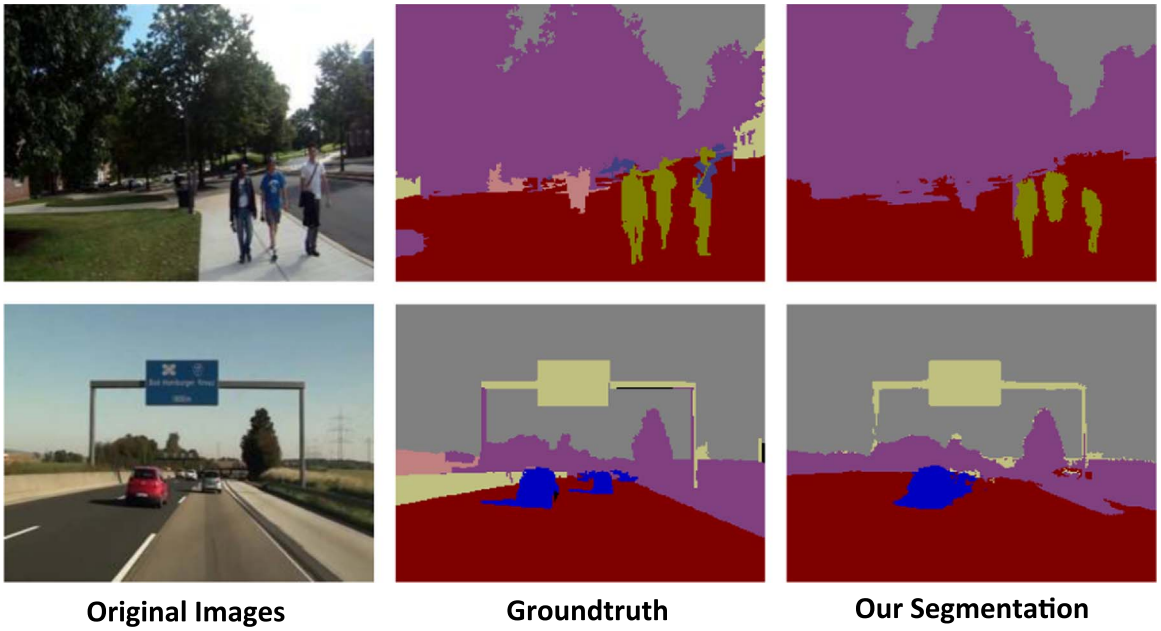

Fig. 5. Our segmentation results on GATECH dataset.

Table 3
Comparison with state-of-the-art methods on GATECH test set.

Methods	Class avg	Global avg	Mean IoU
V2V-scratch[7]	–	66.7	–
V2V-finetune[7]	–	76.0	–
HDCNN	57.6	81.3	47.0
HDCNN+TL	**58.0**	**82.1**	**48.2**

Table 4
Comparison between HDCNN and other deep networks on PASCAL VOC 2012 test set.

Methods	Convergence Time	Model Size	Mean IoU
FCN–8 s[9]	120 h	513 M	62.2
DeconvNet[21]	168 h	961 M	69.6
SegNet[23]	–	115 M	59.1
Bayesian SegNet[32]	–	115 M	60.0
HDCNN	89 h	115 M	**73.1**

larger testing image scale. We rescale the testing images to 448 × 448 and provide the results with single HDCNN and our entire framework. Some of our segmentation results with the entire framework and 448 × 448 input scale are shown in Fig. 4..

It can be found in Table 2 that, with object proposals added, the performance of HDCNN increases by 2.9%, 0.5% and 1.2% on Class avg, Global avg and Mean IoU respectively, for the reason that more multi-scale and detailed information of objects is brought in. When the Transition Layers are added, the results further improve with 0.4%, 0.9% and 1.0% respectively, which confirms the effectiveness of taking the semantic correlations of spatially and temporally adjacent pixels

into consideration. And the results with this setting are already the highest on Class avg and Global avg. With the test resolution increases to 448, obvious improvements are achieved because of the introduction of more object details. And we win the best results on all the three metrics.

4.5. Performance on GATECH

As the video and frame numbers of CamVid [11] dataset are too low, we evaluate our approach on GATECH dataset [12], which is much larger than CamVid, to verify the generalization of our framework. Inference is conducted with image size 224 here, and some of our segmentation results are shown in Fig. 5. We compare our approach mainly with the work of Tran et al. [7] on this dataset, shown in Table 3. The biggest difference between their work and ours is that they employ 3D CNN to solve the problem with a FCN framework, where the video sequences are convolved across space and time domains simultaneously. While we process each video frame through a 2D CNN and combine the temporal information afterward. Two highest results in their work are cited for comparison, which are trained from scratch and finetuned from a pretrained model respectively. Our results outperform theirs obviously with more than 6% on Global avg, which confirms the effectiveness of our approach..

4.6. HDCNN performance on PASCAL VOC 2012

We also conduct an experiment on PASCAL VOC 2012 dataset [27], to verify the effectiveness of the proposed HDCNN on semantic image segmentation. We compare our results with other deep networks used for semantic segmentation, which are FCN [9], DeconvNet [21], SegNet [23] and Bayesian SegNet [32]. Comparison on the test set is provided in Table 4. Our approach achieves the best Mean IoU result among all the compared methods and shows advantages in convergence time and model size. SegNet and Bayesian SegNet also use compact models like us, but our result obviously outperforms theirs. While compared with FCN and DeconvNet, our approach provides a much smaller model size and faster training speed, as well as better performance. All these comparisons confirm the proposed HDCNN a lightweight and effective model.

5. Conclusion

We propose a jointly trained framework for semantic video segmentation, which consists of HDCNN and Transition Layers. The HDCNN provides more refined resolution reconstruction for pixel-wise semantic interpretation with a deconvolutional structure, where fully connected layers are removed and hierarchical classification layers are added by cooperatively using unpooling and bilinear interpolation. Transition Layers learn the semantic correlations between categories within a spatio-temporal volume, and make prediction for a pixel considering all of its spatially and temporally adjacent pixels. These two parts are connected and trained jointly as a unified deep network. Extensive experiments are conducted on CamVid and GATECH datasets, and we achieve state-of-the-art results on both of them, which indicates the effectiveness of our approach.

Conflict of interest

None declared.

Acknowledgements

This work was supported by 863 Program (2014AA015104) and National Natural Science Foundation of China (61332016, 61272329, and 61472422).

References

[1] V. Badrinarayanan, I. Budvytis, R. Cipolla, Semi-supervised video segmentation using tree structured graphical models, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2751–2764.
[2] V. Badrinarayanan, F. Galasso, R. Cipolla, Label propagation in video sequences, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 3265–3272.
[3] B. Liu, X. He, S. Gould, Multi-class semantic video segmentation with exemplar-based object reasoning, in: Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on, IEEE, 2015, pp. 1014–1021.
[4] G. Floros, B. Leibe, Joint 2d-3d temporally consistent semantic segmentation of street scenes, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2823–2830.
[5] A. Jain, S. Chatterjee, R. Vidal, Coarse-to-fine semantic video segmentation using supervoxel trees, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1865–1872.
[6] H. Zhang, K. Jiang, Y. Zhang, Q. Li, C. Xia, X. Chen, Discriminative feature learning for video semantic segmentation, in: Virtual Reality and Visualization (ICVRV), 2014 International Conference on, IEEE, 2014, pp. 321–326.
[7] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Deep end2end voxel2voxel prediction, arXiv:1511.06681
[8] S. Ji, W. Xu, M. Yang, K. Yu, 3d convolutional neural networks for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 221–231.
[9] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: CVPR, 2015.
[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248–255.
[11] G.J. Brostow, J. Fauqueur, R. Cipolla, Semantic object classes in video: A high-definition ground truth database, Pattern Recognit. Lett. 30 (2) (2009) 88–97.
[12] S. Raza, M. Grundmann, I. Essa, Geometric context from videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3081–3088.
[13] J. Tighe, S. Lazebnik, Superparsing, Int. J. Comput. Vis. 101 (2) (2013) 329–349.
[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Semantic image segmentation with deep convolutional nets and fully connected crfs, in: ICLR, 2015.
[15] G. Lin, C. Shen, I. Reid, A. van den Hengel, Efficient piecewise training of deep structured models for semantic segmentation, arXiv:1504.01013
[16] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P. Torr, Conditional random fields as recurrent neural networks, in: ICCV, 2015.
[17] Z. Liu, X. Li, P. Luo, C.C. Loy, X. Tang, Semantic image segmentation via deep parsing network, in: ICCV, 2015.
[18] M.D. Zeiler, G.W. Taylor, R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, in: ICCV, 2011, pp. 2018–2025.
[19] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: ECCV, 2014, pp. 818–833.
[20] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, in: ICLR Workshop, 2014.
[21] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: ICCV, 2015.
[22] S. Hong, H. Noh, B. Han, Decoupled deep neural network for semi-supervised semantic segmentation, in: NIPS, 2015.
[23] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, arXiv:1511.00561
[24] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: ICLR, 2015.
[25] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: ICML, 2015, pp. 448–456.
[26] C.L. Zitnick, P. Dollár, Edge boxes: Locating object proposals from edges, in: Computer Vision–ECCV 2014, Springer, 2014, pp. 391–405.
[27] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, IJCV 88 (2) (2010) 303–338.
[28] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, J. Malik, Semantic contours from inverse detectors, in: ICCV, 2011, pp. 991–998.
[29] A. Vezhnevets, V. Ferrari, J.M. Buhmann, Weakly supervised semantic segmentation with a multi-image model, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 643–650.
[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, arXiv:1408.5093
[31] B. Liu, X. He, S. Gould, Joint semantic and geometric segmentation of videos with a stage model, in: Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on, IEEE, 2014, pp. 737–744.
[32] A. Kendall, V. Badrinarayanan, R. Cipolla, Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding, arXiv:1511.02680

Yuhang Wang received the B.E. degree from Zhejiang University, Hangzhou, China, in 2012. He is currently pursuing the Ph.D. degree at National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include deep learning, image or video segmentation and understanding.

Jing Liu received the B.E. and M.S. degrees from Shandong University, Shandong, in 2001 and 2004, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2008. She is a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. Her current research interests include deep learning, image content analysis and classification, multimedia understanding and retrieval.

Yong Li received the B.E. degree from Huazhong University of Science and Technology, Wuhan, China, in 2011. He is currently pursuing the Ph.D. degree at National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include image analysis, understanding and retrieval, machine learning.

Jun Fu received the B.E. degree from Huazhong University of Science and Technology, Wuhan, China, in 2015. He is currently pursuing the M.S. degree at National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include deep learning, video analysis and applications.

Min Xu received the B.E. degree from University of Science and Technology of China, Hefei, China, in 2000, the M.S. degree from National University of Singapore, Singapore, in 2004, and the Ph.D. degree from University of Newcastle, NSW, Australia, in 2010. She is a Lecturer with Faculty of Engineering and Information Technology, University of Technology, Sydney. Her research interests include multimedia content analysis, interactive multimedia, pattern recognition, and computer vision.

Hanqing Lu received his B.S. and M.S. from Department of Computer Science and Department of Electric Engineering in Harbin Institute of Technology in 1982 and 1985. He got his Ph.D. from Department of Electronic and Information Science in Huazhong University of Science and Technology. He is a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include image similarity measure, video analysis, multimedia technology and system.