A CNN Model for Semantic Person Part Segmentation With Capacity Optimization

Yalong Jiang and Zheru Chi, *Member, IEEE*

***Abstract*— In this paper, a deep learning model with an optimal capacity is proposed to improve the performance of person part segmentation. Previous efforts in optimizing the capacity of a convolutional neural network (CNN) model suffer from a lack of large datasets as well as the over-dependence on a single-modality CNN, which is not effective in learning. We make several efforts in addressing these problems. First, other datasets are utilized to train a CNN module for pre-processing image data and a segmentation performance improvement is achieved without a time-consuming annotation process. Second, we propose a novel way of integrating two complementary modules to enrich the feature representations for more reliable inferences. Third, the factors to determine the capacity of a CNN model are studied and two novel methods are proposed to adjust (optimize) the capacity of a CNN to match it to the complexity of a task. The over-fitting and under-fitting problems are eased by using our methods. Experimental results show that our model outperforms the state-of-the-art deep learning models with a better generalization ability and a lower computational complexity.**

***Index Terms*— Person part segmentation, convolutional neural network, complementary modules, simplification of CNNs, capacity optimization.**

I. Introduction

SEMANTIC image segmentation has long been a challenging computer vision task due to a lack of ground-truth and knowledge guidance. Convolutional neural networks (CNNs) have been applied to image segmentation because of their capability of representing a wide variety of complex functions with a high Vapnik-Chervonenkis dimension [1]. CNNs have shown a better performance in semantic segmentation [2], image classification [3] and object localization [4]. The major goals of our study include designing a sound model for person part segmentation with comprehensive feature representations and optimizing the capacity of the model according to the complexity of a task to avoid over-fitting/under-fitting and reduce computational complexity.

Manuscript received April 7, 2018; revised August 15, 2018 and October 21, 2018; accepted November 22, 2018. Date of publication December 14, 2018; date of current version February 13, 2019. The work of Y. Jiang was supported by The Hong Kong Polytechnic University for the Ph.D. degree. This work was supported in part by the Natural Science Foundation of China under Grant 61473243 and in part by the The Hong Kong Polytechnic University under Project 4-BCCJ. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tao Mei. *(Corresponding author: Yalong Jiang.)*

Y. Jiang is with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: yalong.jiang@connect.polyu.hk).

Z. Chi is with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, and also with the Shenzhen Research Institute, The Hong Kong Polytechnic University, Hong Kong (e-mail: enzheru@polyu.edu.hk).

Digital Object Identifier 10.1109/TIP.2018.2886785

Existing research works on semantic segmentation include R-CNN [5] and selective search [6] which predict bounding boxes or super-pixels, and perform classifications on the predictions. These methods decouple segmentation from classification and do not consider global information which regularizes spatial relationships between regions. As a result, the predictions are blocky. Improved methods include FCN [2] which utilizes fully convolutional neural networks to produce segmentation masks on images. Most improved models are of encoder-decoder structures. However, these models suffer from a heavy computational burden and a lack of global clues as an efficient guidance. Moreover, the generalization of the models is restricted by limited training data. For instance, Deeplab-V2 proposed in [7] and Deeplab-V3 proposed in [8] suffer from the over-fitting problem due to insufficient training data.

To overcome these problems, a model based on comprehensive feature representations is proposed in this paper. In our model, segmentation is coupled with classification. The overall model (OM) is composed of one module focusing on local information and another module focusing on global information, as is shown in Fig. 1. In our model, the feature representation is enriched and robustness is improved. Module 1 is a CNN for localizing anatomical landmarks and it makes inferences from a global point of view. We name it as Localization Module (LM). Module 2 is implemented by a CNN for segmentation and it makes inferences from a local point of view. We name it as Segmentation Module (SM). The two CNNs are trained on two datasets and their combination brings improvements. Dilated convolutions proposed in [9] and the Conditional Random Fields (CRF) proposed in [10] are also integrated into the proposed model. A preliminary version of this model was published previously [11].

To improve generalization ability while reducing computational complexity, two novel schemes are also proposed. The first scheme partitions a CNN model into independent functional units. The partition is based on the similarity among convolutional kernels and the Expectation-Maximization Algorithm (EM) proposed in [12]. The necessity of functional units is evaluated and only the units that are most useful for the task are kept. The second scheme adjusts the capacity of a CNN model to match it to the complexity of a task through adjusting the linear dependency among convolutional kernels. Moreover, a novel structure of convolutional layers is proposed to simplify the conventional convolutional layers while maintaining a similar performance. By adjusting the model capacity, the over-fitting problem is also eased.

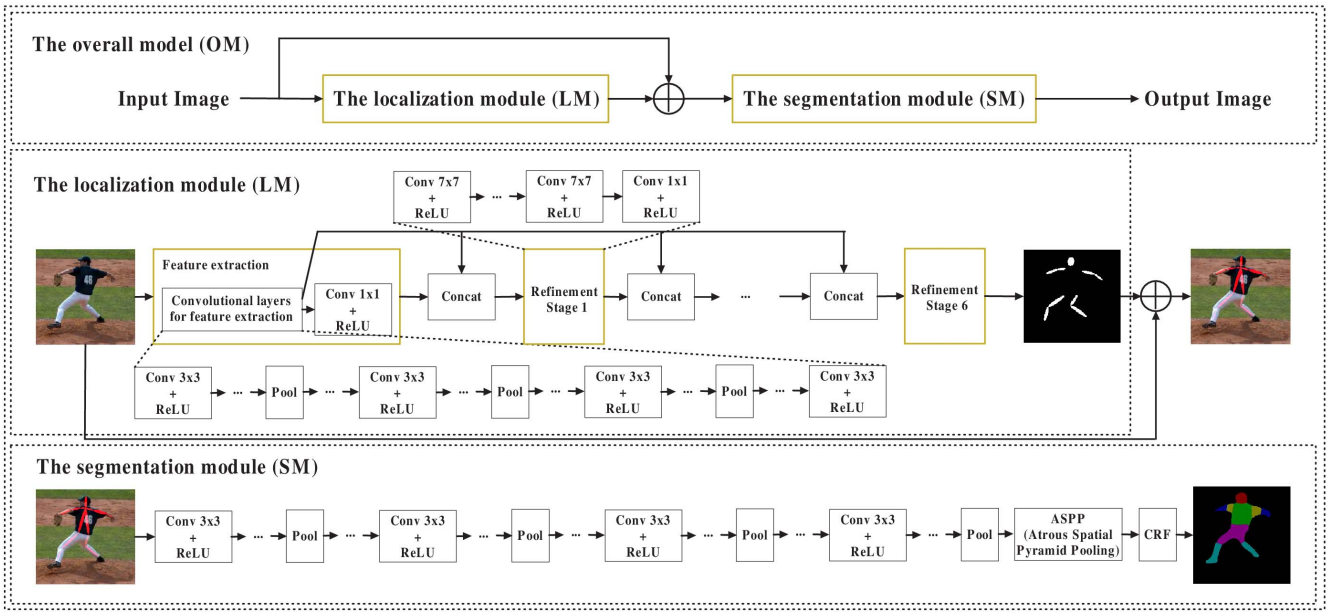

Fig. 1. Our proposed image segmentation model consisting of SM for segmentation and LM for localizing anatomical landmarks. The latter contains one feature extraction stage and six refinement stages. The feature extraction stage contains 15 convolutional layers. The refinement stages are the same, each refinement stage has 7 convolutional layers. The anatomical landmarks from LM produces are input to SM which contains 37 3×3 convolutional layers. The ASPP (Atrous Spatial Pyramid Pooling) will be introduced in Fig. 3. The modules are fine-tuned on three datasets. Firstly, the feature extraction stages in LM and SM are pre-trained on the ImageNet dataset [13]. Secondly, both modules are fine-tuned on the COCO dataset proposed in [14]. Finally, SM is fine-tuned on PASCAL VOC 2010 Person Part Dataset while LM is fine-tuned on the MPII human Pose dataset proposed in [15].

Existing approaches to tackle insufficient training data include weakly supervised methods, such as BoxSup [16] and Segmenting Weakly Supervised Images [17]. However, these methods still require annotations and cannot significantly improve performance in complex cases. Another method proposed is inexact supervision [18]. Inexact supervision is easier to obtain and is helpful to the task. In our case, the supervision in the form of skeletons can be regarded as inexact supervision and utilized to train LM whose predictions are used to regularize SM. The knowledge gained from the task of localizing skeletons is used to enhance the performance in the task of person part segmentation. The feature representation of SM and that of LM are complementary, the combined feature representation significantly improves performance. Different from weakly supervised learning where the overall model is trained with inexact supervision, SM in our proposed method is still trained on the ground truth segmentation labels. This novel model achieves the benefits of using other datasets without expensive annotations. It takes much less time to label skeletons than to label images pixel-by-pixel. Moreover, the pose annotations in [14] and [15] are already publicly available so we do not need to label skeletons manually.

In summary, the contributions of this paper are in three aspects: (1) The development of a model for segmentation based on both global information and local information. The knowledge gained from the task of skeleton localization (LM) helps to enhance the performance in person part segmentation (SM). We have explored carefully the complementary nature of the two types of information with deep learning in our previous work [11]; (2) A method is proposed to adjust the capacity of SM by partitioning convolutional layers into functional units and removing unnecessary functional units. By such doing, computational efficiency is improved; (3) A novel structure of convolutional layers is proposed to simplify LM. The linear dependence among convolutional kernels is analyzed by PCA to determine the capacity of a CNN model which is matched to the complexity of a task.

The rest of the paper is organized as follows. Section II introduces related work. Section III discusses the details of our proposed model and introduces extra supervision in improving image segmentation performance. Section IV describes our two methods of adjusting CNN capacity. Section V shows some the details of implementation as well as experimental results and discussion. Concluding remarks are drawn in Section VI.

II. Related Work

A. Semantic Segmentation

Early deep learning models for the task are mainly composed of the cascade of bottom-up image segmentation and the classification of regions. One typical method is R-CNN [5] which takes bounding box proposals and masked regions provided by selective search [6] as inputs to incorporate shape information into classification. However, its ability to delineate boundaries is poor. The utilization of hierarchical features for scene labeling [19] has shown some improvements in semantic segmentation. However, segmentation is still decoupled from classification.

To couple segmentation with classification and to improve interpretability, some methods which are trained in an end-to-end fashion and based on encoder-decoder structures have

been proposed. FCN [2] utilizes a fully convolutional neural network to produce segmentation masks. The network is trained end-to-end and performs well on PASCAL VOC 2011 [20]. Although the above-mentioned models have coupled segmentation with classification, the generalization is restricted by limited training data and the lack in global clues. For instance, the structure of the decoder in DeepSaliency [21] is task specific. The models for semantic part segmentation [11], [22] also suffered from the lack in training data. The experiments in [22] can only be conducted on a dataset such as the PASCAL Person Part dataset [23] which has both pose and part segment annotations. Similarly, EdgeNet [24] demanded a set of training data that had both part segment and boundary annotations, making an extension of dataset more difficult. However, our proposed LM can be trained on a larger dataset [14], [15] with only pose annotations. The larger datasets enable LM to outperform the Pose FCN in [22] as shown in Table VIII. Moreover, the structure in our proposed model is different from those in [22] and [25]–[27]. The details of structure differences will be discussed in Section III-A and B.

Additionally, the encoder-decoder structure in existing models, such as the Fully Convolutional Network (FCN) [2], suffers from low efficiency. Training end-to-end is difficult and the training process has to be divided into stages. Besides a heavy computational burden, the implementation also suffers from excessive cost in memory. Although later versions of FCNs improved on the original FCN through introducing a Recurrent Neural Network (RNN) [28], the improvement is still at the cost of efficiency. To address these issues, our proposed model tries to reduce computational burden by adjusting the capacity of a CNN to meet the complexity of a dataset (task).

B. Capacity Optimization

The definition of optimal capacity is introduced in [29]. The optimal capacity corresponds to the boundary between the under-fitting regime and the over-fitting regime. Existing research on capacity adjustment can be divided into three categories. The first category focuses on increasing either the height or width of a CNN [30]. The second category focuses on increasing the order of computations in a CNN [31]. The third category of work focuses on computing the mutual information inside a CNN [32]. One related research area is developmental learning [33].

The implementation of the first category includes increasing either the width or height of a CNN and fine-tuning the entire network. It was shown in [30] that deeper neurons allow for new compositions of existing neurons while wider neurons allow for the discovery of additional task-specific clues. However, the increase in capacity is at the cost of efficiency. One representative work of the second category is proposed in [31]. Second-Order Response Transform (SORT) was developed to better utilize existing features without introducing extra parameters. High-order functions are learned to enrich the hypothesis space and improve the performance of ResNet [34] and WRN [35] on CIFAR10 and CIFAR100 [36]. However, over-fitting is easier to occur. The third category of work aims at describing the learning of CNNs with information theories. According to the Information Bottleneck Principle, the capacity of a CNN is determined by the amount of relevant information. However, only very simple networks have been studied and none of the three categories of work has considered matching the capacity of a CNN to the complexity of a task.

C. Weakly Supervised Segmentation

Existing weakly supervised algorithms for semantic segmentation can be divided into four categories: learning based on bounding boxes, learning based on scribbles, learning based on image tags, and mixing multiple types of annotations.

Two representative methods of the first category are BoxSup [16] and Segmenting Weakly Supervised Images [17]. The basic idea behind BoxSup is iterating between bounding box generation and the training of a CNN. Although the methods perform well on PASCAL VOC and PASCAL CONTEXT, they cannot deal with complex images with multiple objects.

The algorithms based on scribbles include 3D U-Net [37] and ScribbleSup [38]. 3D U-Net performs volumetric segmentation with a semi-automated or fully-automated setup. Scribble annotations include only several lines that lie in the regions of interest. Annotations are still required and the inference lacks a consideration of the spatial relations between super-pixels. In our proposed method, the constraints on spatial relations are involved through incorporating LM.

The algorithms based on image-tags include the constrained CNN proposed in [39]. The labels in the constrained CNN are image-specific without any information on positions. The labels are converted to the constraints on spatial distributions. However, only coarse segmentation masks could be generated.

The algorithms of the fourth category combine image-specific labels with bounding boxes and scribbles [40]. Examples were shown in [40] where all images were divided into super-pixels and clustering was conducted on super-pixels. Although the methods have brought improvements in semantic segmentation, they can only deal with the images without disturbing variances in objects. Our proposed model eases this problem through combining local and global constraints.

III. A New Segmentation Model

The proposed model is shown in Fig. 1. The reason for integrating the two modules (CNNs) lies in the fact that they are of complementary characteristics. SM mainly explores local properties which are robust to the variances in images such as occlusions or complex lightening conditions. LM mainly explores the relationships between different regions and exhibits robustness to the cases where different body parts are covered by similar colors. Although the two modules are integrated, the overall computational burden is lower than that of Deeplab-V2 [7], as will be shown in Section V- B. It will also be shown in V-B that the two modules integrated can each be optimized in terms of capacity using different strategies.

Typical predictions from the two modules are shown in Fig. 2. In Fig. 2 (c), the upper arms and the torso share

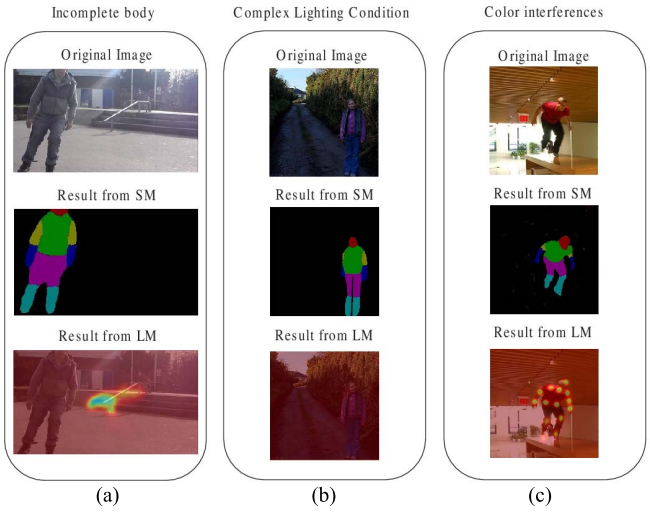

Fig. 2. Three non-ideal cases in person part segmentation. In the first two cases, LM outputs wrong heat-maps indicating the locations of body parts, whereas SM works well. In the third case, SM fails to distinguish the upper arms from the torso, whereas LM works well.

the same color. SM misclassifies the two parts as the same part while LM accurately locates the shoulders and elbows, respectively.

The features extracted by SM are pixel-specific. The module classifies pixels correctly as long as the pixels and their neighbors satisfy certain conditions. It does not take the relations between remote pixels into consideration. Thus the module succeeds in segmenting out certain parts even when remote but correlated parts are occluded. In comparison, LM focuses on the spatial relations between different regions. For example, one region with a high probability to include the head and another region that is likely to include the torso are certain to be neighboring. LM performs well in the situation when different parts are covered by the same color.

In our model, SM is a strong pixel-specific classifier while LM a weak pixel-specific classifier because the latter can only detect the regions that certain pixels are probable in. Suppose that the $i-th$ pixel belongs to the $s-th$ region. Denote the prediction of LM on the sth region as $y_2(s)$ and the output probability of the $i-th$ pixel belonging to the $j-th$ class as $y_{1j}(i, y_2(s))$. The inference of SM depends on both the output of LM and the pixel-specific features. The loss function for training is

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{M} t_j(i) \log y_{1j}(i, y_2(s)) \tag{1}$$

where $t_j(i)$ is a binary variable, indicating whether the $i-th$ pixel belongs to the $j-th$ class ($t_j(i)=1$) or not ($t_j(i)=0$). M is the number of classes and N is the number of pixels on the output map. Eq. (1) actually defines a multi-class cross entropy.

The two modules work in a serial fashion. The output of LM is also part of the input images of SM during training and testing. Segmentation is based on both the original image and the heat-map indicating the locations of skeletons. The reason of placing LM before SM is that the former is trained on a much larger dataset [14] and makes more reliable predictions, as will be shown in Table IV. By applying the techniques discussed in Section IV, the total number of parameters of the model is less than that of Deeplab-V2 [7]. Different from Mask R-CNN [41], our proposed model allocates two streams for the two modalities instead of fusing them in one. The advantage lies in that the features needed for the two modalities differ from each other significantly. The results with Mask R-CNN on the same dataset will be shown in Table V.

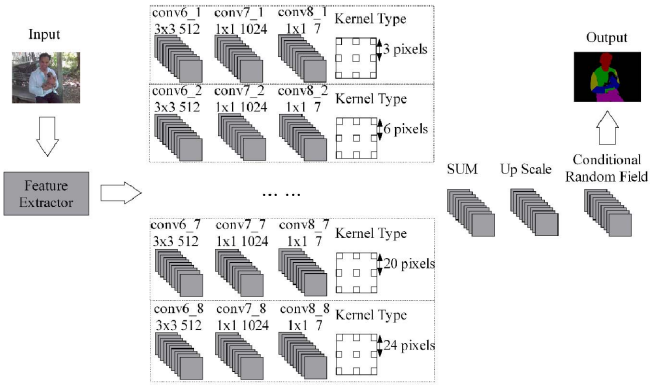

Fig. 3. Fully-convolutional SM. Eight parallel filters with different field-of-views are adopted to extract the features for pixel classification. The Kernel Type refers to the types of the kernels in the layers from conv6_1 to conv6_8. The kernels in the 8 layers are with size 3×3 but differ in the distance between weights in the kernels. The heat-maps generated by the 8 parallel convolutional layers are concatenated in the 6^{th} macro-layer. The feature extractor is composed of the first 5 macro-layers shown in Table II.

A. SM for Segmentation

SM is shown in Fig. 3. Six body parts are defined: head, torso, upper arms, lower arms, upper legs and lower legs. The CNN is trained in an end-to-end fashion [2]. We refer to a group of convolutional layers working together as the feature extractor for a particular scale as a macro-layer. The first 5 macro-layers are inherited from the macro-layers in VGG16 [3] with some modifications. The 6^{th} to 8^{th} macro-layers are task-specific and used to recover detailed local structures [10], [42]. The network is simplified with Algorithm 1. In the 6^{th} macro-layer, a mechanism known as ASPP [7] is adopted to enable 3×3 filters to have different field-of-views. Different from [22] in which segmentation was based on Deeplab-LargeFOV [7] with one field-of-view, SM includes eight field-of-views. Following the 8^{th} macro-layer, the fully connected conditional random field (CRF) proposed in [10] is used. The inference is based on the distances and similarity between pixels [10].

The energy function is defined as

$$\begin{aligned} E(\mathbf{x}) = &\sum_i \varphi_u(x_i;\theta) + \sum_{i\neq j} \mu(x_i, x_j) \\ &\cdot \left[\omega^{(1)} \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right)\right. \\ &\left. + \omega^{(2)} \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right)\right] \end{aligned} \tag{2}$$

Algorithm 1 Partitioning a CNN Layer Into Functional Units

1. Initialize $K, \boldsymbol{\mu}=\{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,...,\boldsymbol{\mu}_K\}, \boldsymbol{\Sigma}=\{\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2,...,\boldsymbol{\Sigma}_K\}$ and $\boldsymbol{\pi}=\{\pi_1,\ldots,\pi_K\}$
2. (Outer loop) Increase K
3. (Inner loop) Perform E step to evaluate the responsibilities $\gamma(z_{nk})=\frac{\pi_k N(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K}\pi_j N(\mathbf{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}$
4. Perform M step to update $\boldsymbol{\mu}=\{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,...,\boldsymbol{\mu}_K\}, \boldsymbol{\Sigma}=\{\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2,...,\boldsymbol{\Sigma}_K\}$ and $\boldsymbol{\pi}=\{\pi_1,\ldots,\pi_K\}$
5. Check the convergence of $\ln p(\mathbf{X}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})$. If the convergence criterion (reaching maximum) is not satisfied, return to Step 3, else proceed.
6. Evaluate the influence of halving the number of clusters within one layer on the performance by comparing the test accuracy before and after the process. If the drop in accuracy is below a threshold (3%), stop, else return to Step 2.

where x_i and x_j are the label assignments for the ith and jth pixels, respectively; $\varphi_u(x_i;\theta)$ is the energy of assigning x_i to the ith pixel with the θ denoting the parameters in the CNN for extracting features; $\mathbf{x}$ is the vector containing all the assigned labels. The second term in Eq. (2) takes into consideration the similarity and smoothness. The first term in the square bracket indicates that two neighboring and similar pixels are likely to belong to the same class; the second term indicates that two nearby pixels are likely to share the same label. $\mu(x_i, x_j)$ equals to one if $x_i \neq x_j$ and zero if $x_i = x_j$; $\omega^{(1)}$ and $\omega^{(2)}$ are the weights for the linear combination; I_i and I_j are the color vectors of pixel i and j, respectively. p_i and p_j are the positions of pixel i and j, respectively. θ_α and θ_β are the parameters controlling the degrees of similarity and smoothness.

B. LM for Localizing Anatomical Landmarks

This module is shown in the middle part of Fig. 1 with a feature extraction stage and six stages for refinement, which is different from [22] which localized skeletons based on Resnet-101. A performance comparison will be given in Table VIII. In contrast to Eq. (2), the prediction is region-specific instead of pixel-specific and is given by (3)

$$\theta_{opt} = \arg\min_{\theta} \left\{ \begin{array}{l} \sum\limits_{(x,y_i)\in D} \|y_i - \varphi_i(x;\theta)\|_2^2 + \sum\limits_{(x,y_i),\,(x,y_j)\in D,\, i\neq j} \\ \left\| [y_i, y_j] - [\varphi_i(x;\theta) - \varphi_j(x;\theta)] \right\|_2^2 \end{array} \right\} \tag{3}$$

where $\varphi_i(x;\theta)$ denotes the prediction of the location of the ith part in image x with θ, the learned parameters of the CNN. y_i denotes the corresponding ground truth. D denotes the training dataset. The first term in Eq. (3) enforces accurate predictions of the absolute locations of single parts and the second term enforces accurate predictions of relative locations. This module focuses on region-specific features. Fig. 1 shows our OM including both LM and SM and the way of integrating LM with SM. The output of LM is used to detect (highlight) critical areas of the input image. Highlighting is carried out by deepening the colors of the critical regions, as is shown by the result of LM in Fig. 1. SM receives the processed images as inputs during both training and test stages. The choice of colors is arbitrary and it should not affect the performance.

The feature extraction stage shown in Fig. 1 is inherited from the VGG network proposed in [3]. The network is pre-trained on the ImageNet classification task [13] and then fine-tuned to the task. It is addressed in [43] that pre-training boosts the performance of CNNs. The convolutional layers in the feature extraction stage are simplified with the novel structure shown in Fig. 5 (b). No loss in test accuracy is caused by such a simplification, as to be discussed in the experiments. The complexity of the feature extraction stage is better matched to the task than the original layers before simplification. The reduction in the number of trainable parameters is around 67%.

In order to better utilize the extracted features, six refinement stages are added to post-process the results from the feature extraction stage by iteratively narrowing the regions of interest. Different from [26] and [27] where each stage takes in the features from one specific level, the input to each stage in our proposed LM includes the belief maps from the previous stage and a common set of features provided by the feature extraction stage. Each stage predicts the difference between the predictions from the previous stage and the common ground truth. The gradients propagate directly from the loss functions to each stage in order to avoid the gradient vanishing problem.

Unlike other skeleton detection methods [44] which use Gaussian peaks to indicate the locations of joints, the heat maps generated by LM are in the form of narrow ellipses showing the locations and orientations of the bones of limbs. The heat map shows the regions in an image that are more probable to cover critical person parts and is used to enrich the input of SM. The COCO dataset [14] for localization includes 80,000 images for training. In comparison, the data available for segmentation includes PASCAL VOC 2010 Person Part [23], [45] with 3,533 images only. As will be shown in our experiments, LM outperforms the SM in localizing regions of interest both on the dataset for localization and that for segmentation. Thus the pre-processing with LM is useful.

IV. Matching the Capacity of a CNN to the Complexity of a Task

Matching a model's capacity to the complexity of a task means increasing the model's capacity if it is not competent for the task and decreasing its capacity if it is more than competent for the task. In this section two methods are proposed to adjust the capacity of a CNN model to fit it into a task.

A. Keeping Only the Channels That Are Informative for Predictions

It is meaningful to adjust the capacity of a CNN through changing the number of uncorrelated feature channels. Moreover, it has been shown from a statistical point of view that the capacity of a CNN [29] can be determined by its hypothesis space. For a CNN model, the increment in feature channels offers a more comprehensive hypothesis space.

A method of matching the capacity of a CNN to a task is studied by us. It is implemented by adjusting the number of uncorrelated feature channels. Our proposed model is trained

on the PASCAL VOC 2010 Person Part dataset [23], [45] and the COCO dataset [14] which include 3,533 images and 80,000 images, respectively. The datasets are far smaller than the one used in ImageNet competition [13] which includes more than 1,000,000 images for training. As a result, our model should not be as complex as the Deeplab-V2 model [7] which is matched to the complexity of the ImageNet dataset.

It is addressed in [46] that the use of a Gaussian mixture model for modeling the distribution of weights in a neural network is appropriate. Each data point corresponds to a kernel and each Gaussian component is a collection of kernels. In a CNN, different kernels within one layer correspond to different clues for the task while different layers correspond to different compositions of clues. The clues and compositions in a CNN can be clustered based on their similarities. As a result, a CNN can be divided into functional units. We propose to implement this based on the EM algorithm [12] and reduce the number of functional units to lower the capacity of the segmentation module (SM). This process can overcome the over-fitting problem of a CNN model.

The algorithm is carried out once for a layer. Suppose that the input to a certain layer has M channels and the output has N channels. The trained kernels are divided into N sets, each set contains M kernels corresponding to the connections between all M input channels and one output channel. The kernels within each set are concatenated to produce $\mathbf{x}_i$, $i = 1, \ldots, N$. $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. A Gaussian mixture model is constructed to cluster the N concatenated vectors into K groups. A functional unit denotes an assembling of similar vectors. $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K\}$, $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \ldots, \boldsymbol{\Sigma}_K\}$ and $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ denote the parameters of the K clusters where $\mu_i, i = 1, \ldots, K$ denote the mean vectors, $\Sigma_i, i = 1, \ldots, K$ denote the covariance vectors and $\pi_i, i = 1, \ldots, K$ denote the mixing coefficients. This method is described in Algorithm 1 and is evaluated on SM shown in Fig. 3.

In the beginning, $K = 1$. Halving the cluster will cause a great loss in accuracy. The number of clusters K increases until different clues can be expressed by different clusters. For a proper K, each cluster should corresponds to one functional unit. There are two loops in the algorithm. In the outer loop, K increases and the number of elements in each cluster decreases. For a large cluster, different functional units are integrated in the cluster and the operation of halving a cluster in Step 6 leads to a complete loss of some functional units. For the clusters with appropriate sizes, halving one cluster only results in a partial loss of one functional unit. The system accuracy will not drop significantly since useful portions of the functional units are still kept. The threshold for evaluating the drop in accuracy is chosen to be 3% without re-training.

Moreover, whether a functional unit is useful or not for the task is determined by the influence on the overall performance brought by removing the unit. By keeping only useful functional units, similar performance can be achieved and therefore, the capacity of the model can match well to the complexity of the dataset. Experimental results will show that a smaller network with a suitable capacity outperforms a larger network of overcapacity.

Algorithm 2 Adjusting the Linear Dependency Between Kernels

1. Organize the N vectors and construct a $K^2 \times N$ matrix $\mathbf{X}$.
2. Subtract each entry by the mean of that row.
3. Compute the covariance matrix $\mathbf{C} = 1/N(\mathbf{X}\mathbf{X}^{\mathrm{T}})$.
4. Conduct singular value decomposition (SVD) on $\mathbf{C}$ and record the singular values as well as singular vectors.
5. Determine the necessary number of principal components H based on the needed representation power. The variance ratio is the sum of H largest singular values divided by the sum of all singular values.
6. Construct a $H \times K^2$ matrix $\mathbf{P}$. Its rows are the singular vectors corresponding to the H largest singular values.
7. Approximate the original N vectors with the columns of $\hat{\mathbf{X}}$ where $\hat{\mathbf{X}} = \mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{X}$.

B. Adjusting the Dependency Between Kernels Based on the Complexity of a Task

Besides the number of convolutional kernels, the dependency among kernels also has an influence on capacity. For a fixed number of convolutional kernels, the more independent these kernels are, the more discriminative features can be extracted. We propose to evaluate the linear dependency among convolutional kernels by converting a set of kernels into a set of linearly uncorrelated kernels using principal component analysis (PCA) [47]. The kernels of a CNN are regarded as random variables in a high-dimensional space. An orthogonal transformation is used to find the components that account for as much of the variability in the variables as possible.

Suppose that the input to a certain layer has M channels and the output has N channels. For each of the M input channels, a set of N kernels extracts N types of features. All kernels are formed as vectors. The size of each vector is $1 \times K^2$ for a $K \times K$ kernel. The N kernels are organized into a $K^2 \times N$ matrix $\mathbf{X}$. The procedure for adjusting the linear dependency is described in Algorithm 2.

The H row vectors in matrix $\mathbf{P}$ in Algorithm 2 are linearly independent. Each of the N column vectors in $\mathbf{X}$ can be approximated by a linear composition of the H vectors. The larger H is, the greater the representation power becomes, and the better the variances in the N vectors in $\mathbf{X}$ are represented by the H components. A small H contributes to high linear dependency among these kernels. In the extreme case, if $H = 1$, all the N kernels in $\mathbf{X}$ are linearly correlated. Fig. 4 (a) and (b) show the approximation of the N kernels with the linear combination of three ($H = 3$) and five components ($H = 5$), respectively.

For each layer, the process described in Algorithm 2 is carried out for M times with a fixed H value. The optimal capacity corresponds to the model with the minimum number of principal components achieving the same segmentation accuracy as the original network. It has been shown in our experiments that for all sets of kernels, three components can explain over 85% of the variances in the original kernels. Its loss in accuracy is less than 1%. The reduction in the number of parameters is around 67%.

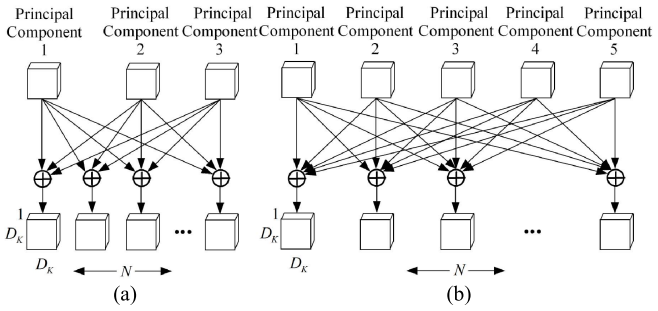

Fig. 4. The method of controlling the linear dependency within layers. (a) The approximation of the original N kernels with the linear combination of three principal components. (b) The approximation of the original N kernels with the linear combination of five principal components.

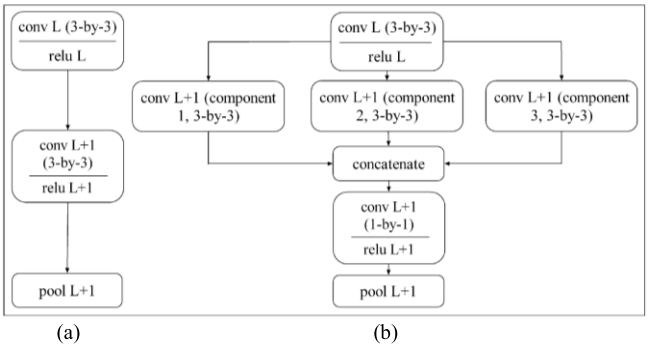

Fig. 5. The structure of a conventional convolutional layer (a) and our proposed novel structure of convolutional layers (b).

Algorithm 2 partitions each of conventional convolutional layers into two sequential layers. The first layer includes $M \times H$ kernels with each of H kernels extracting the features from the same input channel. M is the number of input channels. The second layer includes 1×1 convolutions for approximating the original $M \times N$ kernels with a linear combinations of $M \times H$ principal components. Fig. 5 illustrates the structures of the original and our proposed convolutional layers with H set to 3.

In Fig. 5, "conv L (3-by-3)" and "conv L + 1 (3-by-3)" are conventional convolutional layers with 3-by-3 convolutional kernels. Each layer has $M \times N$ kernels. "conv L + 1 (component i, 3-by-3)" $i = 1, 2, 3$ are special layers. Each layer has only M convolutional kernels each of which corresponds to one input feature channel. "conv L + 1 (1-by-1)" is a conventional 1-by-1 layer with $3M \times N$ convolutional kernels. The original network in Fig. 5 (a) is trained firstly and Algorithm 2 is applied to the trained kernels. The resulting kernels are used to initialize the network in Fig. 5 (b). Then the network in Fig. 5 (b) is re-trained. The goal is to find the network which can perform as well as the original network with the minimum number of principal components. The number of principal components can be selected from one to eight. This is due to the fact that in the PCA analysis, each input variable is a 3-by-3 convolutional kernel with 9 dimensions, and that the network with nine components has the same capacity as the network before simplification. The strategy is starting from the median number of eight, that is, four. If the simplified network with four principal components cannot perform as well as the original network, then try five, six, seven, and eight components in turn; otherwise try one, two and three components in turn. The process is conducted recursively. For each number of principal components, the network is re-trained and tested. If none of the simplified networks can perform as well as the original network, then we have to use the original network for inference. The maximum number of re-training is four.

For any CNN, the procedure for simplification is unique. Removing unnecessary functional units is firstly conducted and the dependence among kernels is then adjusted. As SM is trained on a small dataset [23] and has great redundancy, it can be firstly simplified with Algorithm 1. However, LM is trained on a larger dataset [14], the removal of any functional unit will cause a drop in accuracy. As a result, it can only be simplified through adjusting the correlation between channels. In comparison with Deeplab-V2 and Deeplab-V3, the number of parameters in our simplified network is significantly reduced, as will be shown in Section V-D.

V. Evaluations

A. Datasets and Implementation Details

The two modules in our proposed model were trained on different datasets. LM was trained on the COCO key-points dataset proposed in [14] and the MPII Human Pose dataset proposed in [15]. The former includes 80,000 training images and the latter 28,000 training images. The ground truth labels are the locations of key-points. SM was trained on images from the PASCAL VOC 2010 Person Part Dataset for body part segmentation [23] and [45]. Both LM and SM are pre-trained on the ImageNet [13] dataset. The RGB images are processed by LM before training. The Person Part Dataset includes annotations on 3,533 images where 1,716 images are used for training. The ground truth labels are segmentation masks. To evaluate on larger datasets, the SM is also trained on the LIP (Look Into Person) Dataset [48] with 30,462 training images and 10,000 test images. The capacity of the two modules is optimized using the two methods introduced in Section IV.

B. Matching the Capacity of Modules to a Task

This section is based on the discussion in Section IV. Matching the capacity of a CNN to a task is necessary to avoid under-fitting and over-fitting, and to improve efficiency. The measure adopted for evaluating segmentation performance is mean Intersection Over Union (mIOU) addressed in [49]. It is a common metric for evaluating semantic segmentation models. mIOU is computed by dividing the number of true positive samples by the summation of true positive, false negative and false positive samples:

$$mIOU = \frac{1}{N} \sum_{i=1}^{N} \frac{n_{ii}}{t_i + \sum_{j \neq i} n_{ji}} \tag{4}$$

where n_{ji} is the number of pixels of class j which are predicted to class i, and $t_j = \sum_i n_{ji}$ ($t_i = \sum_j n_{ij}$) is the total

TABLE I

THE CHANGES IN TEST ACCURACY (%) WHEN DROPPING FUNCTIONAL UNITS IN LAYERS CONV6_1, CONV6_2 AND CONV6_3

	conv6_1		conv6_2		conv6_3	
	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)
Complete	0.00	0.00	0.00	0.00	0.00	0.00
Unit 1	-0.33	-0.87	-0.14	-0.10	-0.14	-3.03
Unit 2	-0.02	-0.08	-0.09	-0.97	-0.09	-0.64
Unit 3	-0.05	-0.01	-1.04	-3.19	+0.00	+0.00
Unit 4	-0.58	-1.71	+0.00	-0.32	+0.00	+1.02
Unit 5	-1.11	-4.20	-0.01	+0.18	-0.29	-1.04
Unit 6	+0.00	+0.00	-0.36	-1.16	-0.03	-0.83
Unit 7	-0.01	-0.52	-0.01	-0.33	-0.44	-1.31
Unit 8	-0.14	+0.00	+0.00	+0.00	-0.41	-1.66

TABLE II

THE STRUCTURE OF THE ORIGINAL CNN FOR SEGMENTATION AND THE STRUCTURE OF THE SIMPLIFIED CNN WITH REDUCED FEATURE CHANNELS. $N = 8$ IN BOTH THE ORIGINAL CNN AND THE SIMPLIFIED CNN. THE ASPP [7] IS COMPOSED OF CONV6_$i(i = 1, \ldots, N)N = 8$. THE DILATIONS ARE 3, 6, 9, 12, 16, 18, 20, 24. "BN" DENOTES BATCH NORMALIZATION

Macro-layer	Number of Filters in each 3x3 convolutional layer original CNN/ simplified CNN	Description
Macro-layer1	64/32	BN and ReLU
Macro-layer2	64/32	BN and ReLU
Macro-layer3	128/64	BN and ReLU
Macro-layer4	256/128	BN and ReLU
Macro-layer5	512/256	BN and ReLU
conv6_i $(i=1,...,N)$ in parallel	1024/512	BN and ReLU
conv7_i in parallel	1024/1024	BN and ReLU
conv8_i in parallel	7/7	BN and ReLU

number of pixels belonging to class $j(i)$. The measure mIOU takes into account both false positives and false negatives. Accuracy is defined as $\sum_i n_{ii} / \sum_i t_i$ which divides the total number of correctly classified pixels by the number of pixels in the image.

The change in test accuracy on the reduction of functional units is shown in Table I. This process is conducted layer-by-layer. Each pre-trained layer is divided into several functional units. The feature channels within the same functional unit share similar semantic meanings.

As is discussed in Part A of Section IV, whether a functional unit is useful for prediction is evaluated by comparing the segmentation accuracy before and after removing the unit. The removal of a unit involves dropping the inter-connected feature channels in adjacent layers. If the removal of a functional unit causes no loss to the test accuracy and training accuracy, the functional unit will be removed. Table I shows the influences on performance of removing each functional unit in layers conv6_1, conv6_2 and conv6_3 from the network in Fig. 3.

As is shown in Table I, there is one functional unit in each layer whose absence brings no harm to test accuracy as well as training accuracy. The removal of the 5th unit in conv6_2 brings an improvement in test accuracy and a decrease in training accuracy. It can be inferred that the unit has caused over-fitting. Removing functional units and re-training is carried out iteratively until all the remaining functional units are necessary for the task.

Fig. 6 shows the changes in test accuracy over the reduction in functional units. The "100" on the horizontal axis corresponds to the SM performance before simplification. Table II shows the numbers of feature channels before and after simplification. It can be inferred from Fig. 6 that as the number of the overall feature channels reduces, test accuracy increases first and then decreases with a peak at 46% which corresponds to a network with the architecture shown in Table II. The simplified network outperforms the original one because the removed functional units are not as important in testing as in training. Removing clusters can speed up convergence in

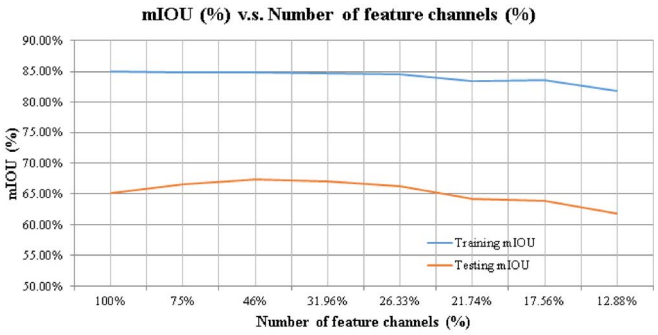

Fig. 6. The changes in training and test accuracy versus the reduction in the number of feature channels. The horizontal axis represents the portion of the number of feature channels remianed in SM while the vertical axis represents the training mIOU (%) and test mIOU (%).

re-training. Test accuracy is measured after removing the redundant functional units and re-training. For speeding up, the number of feature channels has to be multiples of four. Thus we reduce the number of feature channels by 50%.

Different from traditional solutions to the over-fitting problem, the improvement on test accuracy is not at the cost of training accuracy in our approach. In this way, our model is matched better to the task. The over-fitting region is to the left of the peak while the under-fitting region is to the right of the peak, as shown in Fig. 6. Note that the curve is task-specific. Although the location of the peak varies from task to task, the procedure for simplifying networks is unique and involves two steps. In the first step, the method introduced in IV-A is applied and in the second step, the method introduced in IV-B is applied. The first method is suitable for simplifying a CNN which has trained on a small dataset and/or with great redundancy. The second method is suitable for simplifying a CNN which has no clear over-fitting problem or which has already been simplified by the first method. SM is simplified by the first method, while LM is simplified by the second method because it is trained on a much larger dataset in comparison to SM. A comparison on computation will be shown in Part D. Moreover, the reduction in time during

TABLE III

A COMPARISON OF PERFORMANCE AND THE NUMBER OF PARAMETERS USED FOR DIFFERENT MODELS. (ACCURACY IS MEASURED IN PCKH TOTAL (%) FOR LM AND IN MIOU (%) FOR DEEPLAB-V2, SM AND OM)

Module/Model	PCKh total (%)	mIOU (%)	Number of parameters
Deeplab-V2 [7]	-	64.94%	13.161e+7
The original SM	-	65.07%	6.559e+7
The Simplified SM	-	66.31%	4.708e+7
SM in the simplified OM	-	67.43%	1.589e+7
Simplified OM	-	75.62%	4.276e+7
The original LM	72.05%	-	3.472e+7
LM in the simplified OM	71.59%	-	2.687e+7

inference over-weights the extra time spent on simplifying the model. This is due to the fact that training is conducted for once while inference is conducted for countless times.

In our experiments, the maximum number of iterations of training varies from 20,000 to 70,000. Training is terminated when the average training accuracy does not change in two consecutive epochs. Batch size is set to 6. As the number of feature maps reduces, regression needs more training iterations. The training accuracy of models with different numbers of feature channels is shown in Fig. 6.

It is also shown in Fig. 6 that the training accuracy keeps almost unchanged as the number of parameters reduces from 100% to 46%. Therefore, during this process, the capacity of the model is matched to the complexity of the task. As a model becomes smaller, the features extracted by the model are more limited. Under-fitting may occur and the model performs badly on the training set and test set.

For SM and LM, the changes in accuracy over the change in complexity are shown in Table III. Except for LM and simflified LM which were trained and tested the MPII human pose database, all other experiemnets were conducted on the Pascal VOC person part dataset. The metric for evaluating SM is mIOU (%) which is introduced in Part B. Deeplab-V2 [7] is also evalauted with mIOU (%). The performance of LM is evaluated using the PCKh metric proposed in [15] with error tolerance 50% of the head segment length. All critical parts are equally weighted in calculating the PCKh metric. From Table III it can be inferred that the methods of adjusting the dependency between kernels can reduce the complexity of LM without deteriorating accuracy. The simplified SM outperforms the original SM because of the removal of redundant functional units. As is shown in the fourth row in Table III, the SM is optimized to be with 4.708e+7 parameters. If segmentation is conducted only with this version of SM, the accuracy that can be achieved is 66.31%. However, by firstly integrating SM with LM and then optimizing both modules, the overall model (OM) can achieve an accuracy of 75.62% with 4.276e+7 parameters used only.

C. Integration of the Two Modules

As is shown in Fig. 1, LM is used to preprocess an image to enrich the inputs to SM. The experiments were conducted to show that the former is able to provide good pre-processed images for the latter. The quality of pre-processing can be evaluated by the accuracy of localization. The evaluation was conducted on the images for segmentation. The prediction results from SM task were converted to the coordinates of critical parts. For example, the joint between head and torso is neck, and the part on the head that is farthest from torso is vertex, and so on. The ground truths on the 3,533 images from PASCAL VOC 2010 were converted to the coordinates in the same way. We compare the performance of SM with that of the LM using the toolkit [50], the mean Average Precision (mAP) of each body part is measured based on the PCK threshold [50]. Table IV shows a comparison of results.

TABLE IV

A COMPARISON OF MAP (%) ON THE PASCAL VOC 2010 PERSON PARTS DATASET PROPOSED IN [23], [45]

Model	mAP (%)
LM (Fig. 1)	45.6%
SM (Fig. 1)	34.1%

TABLE V

A COMPARISON IN MIOU (%) BETWEEN OUR MODEL AND EXISTING MODELS. ASPP DENOTES ATROUS SPATIAL PYRAMID POOLING

Method	mIOU (%)
Attention [51]	56.39%
HAZN [52]	57.54%
LG-LSTM [53]	57.97%
Mask R-CNN based on ResNet-101-C4 [41]	58.06%
Mask R-CNN based on ResNet-101-FPN [41]	58.17%
Mask R-CNN based on ResNeXt-101-FPN [41]	58.41%
Graph LSTM [54]	60.16%
Deep Lab-V2 [7]	64.94%
Deeplab-V3 with the last ResNet block duplicated for 3 copies [8]	70.69%
Deeplab-V3 with ASPP [8]	71.65%
Deeplab-V3 with ASPP, trained on the images pre-processed by LM	74.31%
SM pre-trained on ImageNet [13] and COCO [14]	67.43%
Our overall model (OM) pre-trained on ImageNet [13] and COCO [14]	75.62%

From Table IV it can be seen that LM provides a good pre-processing for segmentation because it outperforms SM in localizing critical parts. Also the last two rows in Table V show that the integration of two modules (LM and SM) outperforms SM.

D. A Comparison With Other Models and Ablation Study

Our proposed model is compared with other existing methods both in accuracy and computational complexity. Table V shows the performance comparison between our model and some existing methods in person part segmentation. The evaluation was performed on the PASCAL VOC 2010 Person Part dataset. The annotations include head, torso, upper arms, lower arms, upper legs and lower legs. Only 1,716 images were used for training and the results on the validation set are shown in Tables V and VI.

Although Deeplab-V3 [8] has not been evaluated on our task, we attempted to obtain a better benchmark with

TABLE VI

A COMPARISON ON TEST ACCURACY BETWEEN OUR MODEL AND DEEPLAB

Method	Accuracy (%)
Deep Lab-V2 [7]	77.69%
Deep Lab-V3 [8]	80.79%
Our SM pre-trained with ImageNet [13] and COCO [14]	79.35%
Our OM pre-trained with ImageNet [13] and COCO [14]	84.63%

Deeplab-V3. The settings of Deeplab-V3 reported in [8] were adopted. For instance, crop size is set to 513 and the "poly" learning rate policy is adopted. The initial learning rate is 0.007 and the *power* parameter is 0.9. Batch size is 16. Two alternative structures of Deeplab-V3 addressed in [8] were attempted and their results are shown in Table V. In the first setting, the last ResNet block is duplicated 3 copies [8]. In the second setting, the last ResNet block is followed by an ASPP module with 5 branches [8]. It is stated in the paper that structure is an important factor determining the performance. The Deeplab-V2 and Deeplab-V3 models were both pre-trained on ImageNet [13] and COCO [14].

Moreover, Mask R-CNN is re-purposed to our task by considering each person part as one type of instance. Three settings including ResNet-101-C4, ResNet-101-FPN and ResNeXt-101-FPN which have been addressed in [41] were tried. The bounding box for each person part is the tightest rectangle containing the mask of the part. The loss function proposed in the paper was adopted. The results are shown in Table V.

It can be seen from Table V that our model significantly outperforms existing methods. The improvement is about 10.68% over Deeplab-V2 and 1.31% over Deeplab-V3. Moreover, the second from last row shows that SM also outperforms Deeplab-V2 with an improvement of about 2.5%, corresponding to the peak in Fig. 6 (a). Tables V and VI show the mIOU and test accuracy of different models, respectively. It can be observed that our model outperforms the existing methods evaluated. Moreover, the introduction of a pre-processing module LM to improve the performance of segmentation can also be applied to Deeplab-V3. After fine-tuning Deeplab-V3 on the images pre-processed by LM for 40,000 iterations (the same as training SM), the mIOU(%) of Deeplab-V3 increases to 74.31%. More importantly, the computational burden of our model is lower in comparison to Deeplab-V2 and V3. The number of floating point operations (FLOPS) in processing one image with our model is only 1.80e+11 compared with 2.25e+11 for Deeplab-V2. Moreover, the number of parameters in our model is 4.276e+7 while the number in Deeplab-V2 is 1.316e+8. The computational complexity of Deeplab-V3 is even greater than that in Deeplab-V2 because of the added branches in ASPP.

Moreover, it can be inferred from Table V that the accuracy of using SM only for segmentation is 67.43% while the accuracy of integrating LM with SM is 75.62%. Therefore, integrating LM into the overall model improves the performance.

In [49], a human body is segmented into 14 parts and an algorithm is proposed to give 14 labels: head, torso, upper right arm, lower right arm, right hand, upper left arm, lower left arm, left hand, upper right leg, lower right leg, right foot, upper left foot, lower left leg, and left foot. Our model was trained based on the ground truth with 14 labels. The method proposed in [49] is pre-trained on the ImageNet dataset [13]. The 3,533 images are shuffled and randomly divided into two subsets: training set (70%) and test set (30%). The random division is conducted for 10 times and the predictions during the 10 times are averaged. The test set should cover as many of the 14 classes as possible. It can be seen from Table VII that the average performance of our model over the 10 random splits is better than that of the Deep method proposed in [49].

TABLE VII

PERFORMANCE COMPARISON OF DIFFERENT MODELS ON THE PASCAL DATASET WITH 14 BODY PARTS

Method	Accuracy (%)	mIOU (%)
FCN [2]	75.60%	53.12%
Deep method [49]	77.00%	54.18%
Deep method (spatial) [49]	84.19%	66.93%
Deep method (spatial + color) in [49] pre-trained on ImageNet [13]	88.20%	71.71%
Deep method (spatial + color) in [49] pre-trained on ImageNet [13] and COCO [14]	88.27%	71.79%
Our model pre-trained with ImageNet and COCO	91.64%	74.65%

TABLE VIII

A COMPARISON IN mAP (%) BETWEEN OUR LM AND THE POSE FCN PROPOSED IN [22]

Part	Head	Shoulder	Elbow	Wrist
Pose FCN [22]	58.0	52.1	43.1	37.2
LM in our proposed method	64.5	60.1	48.8	43.2
Part	Hip	Knee	Ankle	Total (mAP, %)
Pose FCN [22]	22.1	30.8	31.1	39.2
LM in our proposed method	30.4	36.4	35.7	45.6

Different from [22], [24], and [55] which required both pose and part segment annotations to be available on the same set of images, our proposed LM and SM were trained on different sets of image data with different types of ground truth [14], [15], [23]. Pre-training LM on extra data helps to improve the performance of segmentation. The LM in our model significantly outperforms the Pose FCN in [22], as is shown in Table VIII. We use the toolkit [50] to measure the mean Average Precision (mAP) of all body parts based on the PCKh threshold, in the same way as [22]. Pose configurations are firstly matched to ground-truth pose configurations according to the pose box overlap, and then the average precision (AP) for each joint type is computed and reported. Each ground-truth can only be matched to one estimated pose configuration. Unassigned pose configurations are treated as false positives.

TABLE IX
A COMPARISON OF RESULTS ON THE LSP DATASET

Model	PCK total (%)
Deeplab-V2 [7]	74.83%
Deeplab-V3 [8]	81.36%
Our OM pre-trained on ImageNet [13] and COCO [14]	90.04%

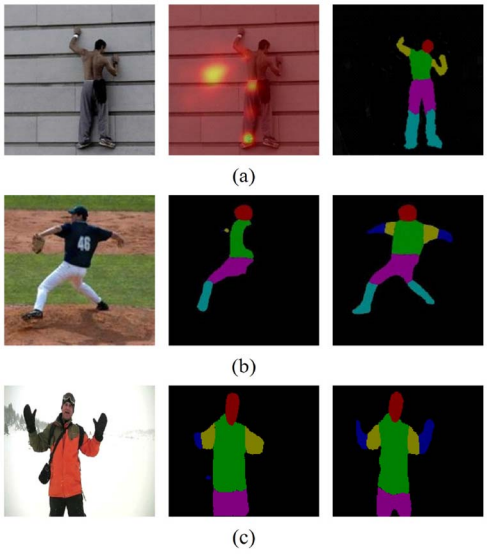

Fig. 7. Performance comparison between SM and OM. (a) A dark input image; (b) an image of pose variations; (c) an image of complex colors on the person's clothes. The left column: the input images; the middle column: the results of SM; the right column: the outputs of our complete model (OM).

E. Evaluation on Additional Data

Table V shows that our proposed model outperforms both Deeplab-V2 and Deeplab-V3 on the PASCAL VOC 2010 Person Part dataset. However, the dataset is not large enough to cover all types of variations. Therefore, a large data set is needed to show that our proposed model performs better in different cases. 12,000 images from the Extended Leeds Sports Dataset (LSP) adopted in [56] were used to compare our proposed model with Deeplab-V2 and Deeplab-V3. In the dataset, some images suffer from various disturbances such as non-ideal illuminations, variances in poses and physiques, complex colors of clothes, different point-of-views, and occlusions. The ground truths in LSP are locations. We firstly perform segmentation using Deeplab-V2 and our proposed model on the images then use the method addressed in Part C to convert the segmentation results to the coordinates of critical parts. The accuracy of localization is shown in Table IX. The same metric PCK as in Part C was measured. Also, some results for subjective evaluation are shown in Fig. 7.

In Fig. 7 (a), the input image suffers from non-ideal illuminations. In Fig. 7 (b) and Fig. 7 (c), the input images suffer from the variations in pose and complex colors of clothes, respectively. The results of using SM only are shown in the middle columns. The results from our proposed model are shown in the right columns. It can be seen that the integration of the two modules (LM and SM) exhibits robustness to variations.

TABLE X
PERFORMANCE COMPARISON ON THE LIP TEST SET [48]

Method	mIOU (%)
Deeplab-V2 [7]	44.96%
JPPNet (with pose info) [48] pre-trained on ImageNet [13] and COCO [14]	51.36%
Our model pre-trained ImageNet [13] and COCO [14]	51.41%

TABLE XI
PERFORMANCE COMPARISON ON THE FREIBURG DATASET [49]

Method	Accuracy (%)	mIOU (%)
FCN [2]	59.69%	43.17%
Deep method (trained on PASCAL) [49]	78.04%	59.84%
Deep method (trained on PASCAL) in [49] (pre-trained on ImageNet [13] and COCO [14])	78.12%	59.90%
Our model (OM) trained on PASCAL	80.63%	63.57%
Deep method (trained on two-people images, tested on four-people images) [49]	81.78%	64.10%
Deep method (trained on two-people images, tested on four-people images) [49] pre-trained on ImageNet [13] and COCO [14]	81.83%	64.18%
Our model OM (trained on two-people images, tested on four-people images)	86.54%	68.23%

Fig. 8. Some segmentation results on the PASCAL Person Part Dataset proposed in [23] and [45].

Additionally, our proposed model is also used to conduct segmentation on the LIP (Look Into Person) Dataset [48]. The results in Table X show that the proposed model outperforms both Deeplab-V2 and the JPPNet proposed in [48].

Table XI shows a comparison of different models on the Freiburg dataset proposed in [49]. The dataset contains 200 images of 1 to 2 people. The images are taken from different viewpoints and orientations. The annotations include the labels for 14 body parts. Our proposed model was fine-tuned and tested on the dataset. It can be seen that our proposed method outperforms the Deep method proposed in [49].

F. Visualizations on the Output of LM

The visualizations on the output of LM are shown in Fig. 8.

G. Results on Difficult Cases

The segmentation results on three difficult cases are shown in Fig. 9, Fig. 10 and Fig. 11. The images in Fig. 9 contain

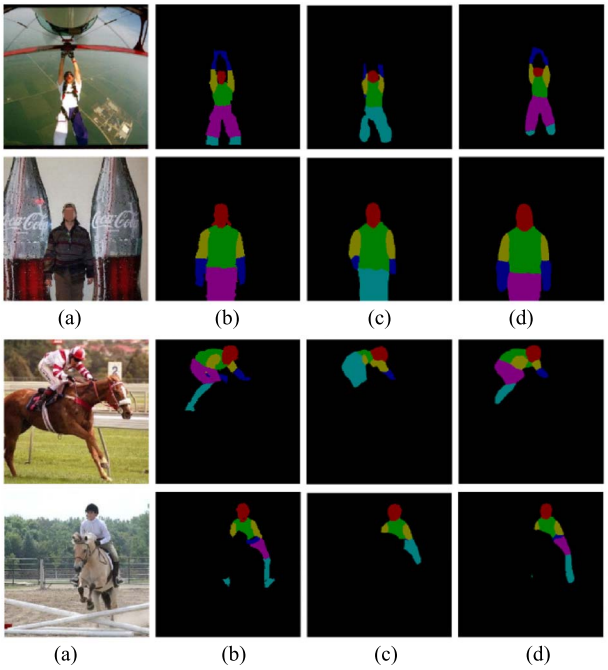

Fig. 9. Some segmentation results on the PASCAL Person Part Dataset proposed in [23] and [45]. The input images are with complex backgrounds. (a) Image. (b) Ground Truth. (c) Deeplab-V3. (d) Ours.

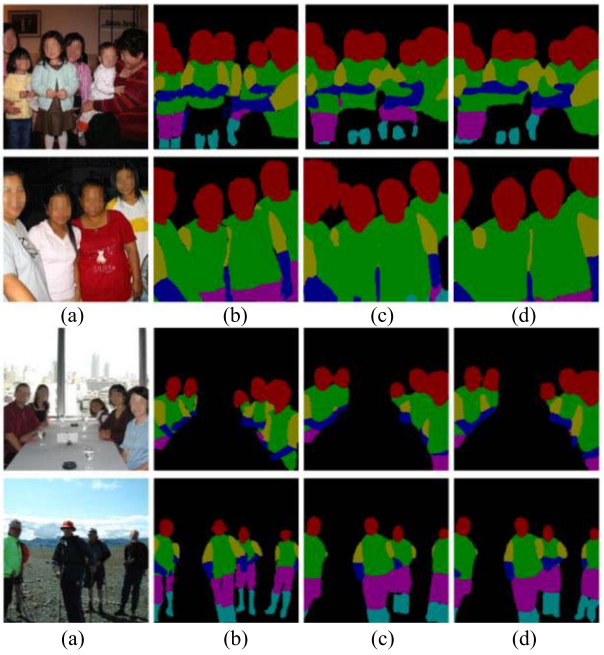

Fig. 10. Some segmentation results on the PASCAL Person Part Dataset proposed in [23] and [45]. The input images are with multiple people. (a) Image. (b) Ground Truth. (c) Deeplab-V3. (d) Ours.

complex backgrounds. The images in Fig. 10 have multiple people while the images in Fig. 11 show great variances in human body poses. Two test sets were evaluated: the images from the PASCAL VOC 2010 Person Part dataset with ground truth (Subset 1) and the images from the LEEDS dataset proposed in [56] without ground truth (Subset 2). The latter is used to augment the test images from Subset 1.

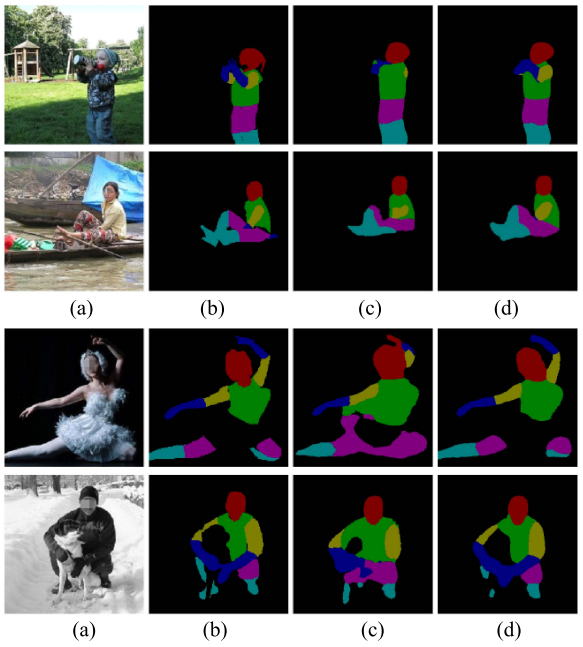

Fig. 11. Some segmentation results on the PASCAL Person Part Dataset proposed in [23] and [45]. The input images are of great variances in people's poses. (a) Image. (b) Ground Truth. (c) Deeplab-V3. (d) Ours.

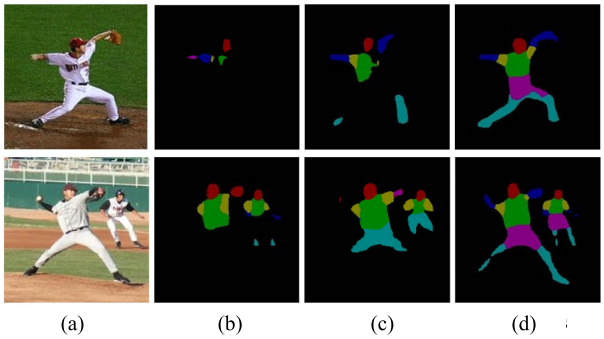

Fig. 12. The segmentation results on two images from LEEDS [56], a dataset that is completely independent from the training data for our model. LEEDS dataset is originally not for body part segmentation, so there is no ground truth. (a) Image. (b) Deeplab-V2. (c) Deeplab-V3. (d) Ours.

We use the images in Subset 1 to show that our model significantly outperforms Deeplab-V3 proposed in [8] and the images in Subset 2 to show that our model outperforms both versions of Deeplab. The segmentation results of two typical images from the LEEDS dataset is shown in Fig. 12.

It can be observed that our model outperforms both versions of Deeplab on these difficult cases. There is no ground truth for the images shown in Fig. 12 from LEEDs. However, we can observe that our proposed model works very well in these cases.

H. Capacity Adjustment of Existing Frameworks

We have further applied our methods proposed in Section IV-A to reduce the capacity of Deeplab-V2 [7]. The performance of the models before and after capacity reduction is shown in Table XII. It can be inferred that our proposed

TABLE XII
THE INFLUENCE OF CAPACITY REDUCTION ON THE PERFORMANCE OF DEEPLAB-V2 [7]

Model	mIOU(%)	Number of parameters
Deep Lab-V2 [7]	64.94%	13.161e+7
Deep Lab-V2 [7] with capacity reduction	65.97%	11.257e+7

method of optimizing capacity can also be used to reduce the capacity of Deeplab-V2 [7] while maintaining accuracy.

I. Software Platform

Our model was implemented using the *Caffe* [57] libraries. The source code will be released at https://github.com/AllenYLJiang/Capacity-Optimization.

VI. CONCLUSION

We have proposed a CNN based segmentation model that integrates the feature representations from two CNN modules: skeleton localization module (LM) and segmentation module (SM). The knowledge gained from LM which has been trained on other datasets without person part ground truth is used to enhance the performance of person part segmentation by SM. Furthermore, two methods have been proposed to optimize the capacity of a CNN model based on the complexity of a task. As a result, a novel structure of CNN convolutional layers with an adjustable capacity has been proposed. The complexity of our proposed model is lowered without a loss in segmentation accuracy. Our model achieves a higher computational efficiency and generalizes better than the current state-of-the-art models. Experimental results have shown that our proposed model performs better than the model Deeplab-V3 by 1.31% on the PASCAL Person Part dataset. Our model also slightly outperforms JPPNet on the LIP dataset. In addition, our model can be used to improve the performance of other tasks such as person re-identification and person part segmentation in surveillance videos (low resolution videos), which are the two directions of our future work.

REFERENCES

[1] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory Probab. Appl.*, vol. 16, no. 2, pp. 264–280, 1971.

[2] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[3] K. Simonyan and A. Zisserman. (Mar. 2015). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[4] J. Sun, S. Ren, K. He, and R. Girshick, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE CVPR*, Jun. 2014, pp. 580–587.

[6] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[8] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. (Jun. 2017). "Rethinking atrous convolution for semantic image segmentation." [Online]. Available: https://arxiv.org/abs/1706.05587

[9] F. Yu and V. Koltun. (Nov. 2015). "Multi-scale context aggregation by dilated convolutions." [Online]. Available: https://arxiv.org/abs/1511.07122

[10] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. NIPS*, Dec. 2011, pp. 1–9.

[11] Y. Jiang and Z. Chi, "A fully-convolutional framework for semantic segmentation," in *Proc. IEEE DICTA*, Nov./Dec. 2017, pp. 1–7.

[12] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[13] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and F. F. Li. (2012). *Large Scale Visual Recognition Challenge 2012*. [Online]. Available: http://www.image-net.org/challenges/LSVRC/2012/

[14] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, Aug. 2014, pp. 740–755.

[15] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE CVPR*, Jun. 2014, pp. 3686–3693.

[16] J. Dai, K. He, and J. Sun, "BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proc. IEEE ICCV*, Dec. 2015, pp. 1635–1643.

[17] L. Zhang, Y. Yang, Y. Gao, Y. Yu, C. Wang, and X. Li, "A probabilistic associative model for segmenting weakly supervised images," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4150–4159, Sep. 2014.

[18] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2017.

[19] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2011). *Visual Object Classes Challenge 2011*. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html

[21] X. Li *et al.*, "DeepSaliency: Multi-task deep neural network model for salient object detection," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3919–3930, Aug. 2016.

[22] F. Xia, P. Wang, X. Chen, and A. L. Yuille, "Joint multi-person pose estimation and semantic part segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6769–6778.

[23] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. L. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *Proc. IEEE CVPR*, Jun. 2014, pp. 1971–1978.

[24] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille, "Semantic image segmentation with task-specific edge detection using CNNs and a discriminatively trained domain transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4545–4554.

[25] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang, "*Not all pixels are equal*: Difficulty-aware semantic segmentation via deep layer cascade," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2017, pp. 3193–3202.

[26] G. Lin, A. Milan, C. Shen, and I. D. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2017, pp. 1925–1934.

[27] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2758–2766.

[28] S. Zheng, "Conditional random fields as recurrent neural networks," in *Proc. IEEE ICCV*, Dec. 2015, pp. 1529–1537.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[30] Y.-X. Wang, D. Ramanan, and M. Hebert, "Growing a brain: Fine-tuning by increasing model capacity," in *Proc. IEEE CVPR*, Jul. 2017, pp. 3029–3038.

[31] Y. Wang *et al.*, "SORT: Second-order response transform for visual recognition," in *Proc. IEEE ICCV*, Dec. 2017, pp. 1359–1368.

[32] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. Inf. Theory Workshop (ITW)*, 2015, pp. 1–5.

[33] O. Sigaud and A. Droniou, "Towards deep developmental learning," *IEEE Trans. Cogn. Devel. Syst.*, vol. 8, no. 2, pp. 99–114, Jun. 2016.
[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Jun. 2016, pp. 770–778.
[35] S. Zagoruyko and N. Komodakis. (May 2016). "Wide residual networks." [Online]. Available: https://arxiv.org/abs/1605.07146
[36] A. Krizhevsky and G. E. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., Apr. 2009.
[37] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-net: Learning dense volumetric segmentation from sparse annotation," in *Proc. MICCAI*, Oct. 2016, pp. 424–432.
[38] J. Sun, D. Lin, J. Dai, J. Jia, and K. He, "ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation," in *Proc. IEEE CVPR*, Jun. 2016, pp. 3159–3167.
[39] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, and X. Geng, "Deep label distribution learning with label ambiguity," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2825–2838, Jun. 2017.
[40] J. Xu, A. G. Schwing, and R. Urtasun, "Learning to segment under various forms of weak supervision," in *Proc. IEEE CVPR*, Jun. 2015, pp. 3781–3790.
[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE ICCV*, Dec. 2017, pp. 2980–2988.
[42] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1454–1466, Oct. 2001.
[43] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS*, Dec. 2014, pp. 3320–3328.
[44] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh. (Apr. 2017). "Realtime multi-person 2D pose estimation using part affinity fields." [Online]. Available: https://arxiv.org/abs/1611.08050
[45] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
[46] S. J. Nowlan and G. E. Hinton, "Simplifying neural networks by soft weight-sharing," *Neural Comput.*, vol. 4, no. 4, pp. 473–493, 1992.
[47] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.
[48] X. Liang, K. Gong, X. Shen, and L. Lin, "Look into person: Joint body parsing & pose estimation network and a new benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[49] G. L. Oliveira, A. Valada, C. Bollen, W. Burgard, and T. Brox, "Deep learning for human part discovery in images," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 1634–1641.
[50] M. Rajchl *et al.*, "DeepCut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 674–683, Feb. 2017.
[51] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3640–3649.
[52] F. Xia, P. Wang, L.-C. Chen, and A. L. Yuille, "Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net," in *Proc. ECCV*, Oct. 2016, pp. 648–663.
[53] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with local-global long short-term memory," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3185–3193.
[54] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with graph LSTM," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 125–143.
[55] F. Xia, J. Zhu, P. Wang, and A. L. Yuille. (Nov. 2015). "Pose-guided human parsing with deep learned features." [Online]. Available: https://arxiv.org/abs/1508.03881
[56] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *Proc. IEEE CVPR*, Jun. 2011, pp. 1465–1472.
[57] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM ACMMM*, Nov. 2014, pp. 675–678.

Yalong Jiang received the B.Eng. degree from Harbin Engineering University in 2012 and the M.Eng. degree from the Beijing Institute of Technology in 2015. He is currently pursuing the Ph.D. degree with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His research interests include pattern recognition, computer vision, and machine learning.

Zheru Chi received the B.Eng. and M.Eng. degrees in electrical engineering from Zhejiang University, in 1982 and 1985, respectively, and the Ph.D. degree from the University of Sydney, in 1994. From 1985 to 1989, he was with the Faculty of the Department of Scientific Instruments, Zhejiang University. He was a Senior Research Assistant/Research Fellow with the Laboratory for Imaging Science and Engineering, University of Sydney, from 1993 to 1995. Since 1995, he has been with The Hong Kong Polytechnic University, where he is currently an Associate Professor with the Department of Electronic and Information Engineering. He has published over 210 technical papers. He was an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS from 2008 to 2010. His research interests include machine learning, pattern recognition, and computational intelligence. He has authored/co-authored one book and 11 book chapters. He is a member of the IEEE. He is currently a Technical Editor of the *International Journal of Information Acquisition*. Since 1997, he has served on the organization or program committees in a number of international conferences.