Deep Human Parsing with Active Template Regression

Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, *Member, IEEE*, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan, *Senior Member, IEEE*

Abstract—In this work, the human parsing task, namely decomposing a human image into semantic fashion/body regions, is formulated as an active template regression (ATR) problem, where the normalized mask of each fashion/body item is expressed as the linear combination of the learned mask templates, and then morphed to a more precise mask with the active shape parameters, including position, scale and visibility of each semantic region. The mask template coefficients and the active shape parameters together can generate the human parsing results, and are thus called the structure outputs for human parsing. The deep Convolutional Neural Network (CNN) is utilized to build the end-to-end relation between the input human image and the structure outputs for human parsing. More specifically, the structure outputs are predicted by two separate networks. The first CNN network is with max-pooling, and designed to predict the template coefficients for each label mask, while the second CNN network is without max-pooling to preserve sensitivity to label mask position and accurately predict the active shape parameters. For a new image, the structure outputs of the two networks are fused to generate the probability of each label for each pixel, and super-pixel smoothing is finally used to refine the human parsing result. Comprehensive evaluations on a large dataset well demonstrate the significant superiority of the ATR framework over other state-of-the-arts for human parsing. In particular, the F1-score reaches 64.38 percent by our ATR framework, significantly higher than 44.76 percent based on the state-of-the-art algorithm [28].

Index Terms—Active template regression, CNN, human parsing, active template network, active shape network

1 Introduction

With the fast growth of on-line fashion sales, fashion related applications, such as clothing recognition and retrieval [21], [28], automatic product suggestions [20], have shown huge potential in e-commerce. Among them, human parsing, namely decomposing a human image into semantic fashion/body regions, serves as the basis of many high-level applications, and has drawn much research attention in recent years [8], [28].

However, there are still some problems with existing algorithms. First, some previous works often take the reliable human pose estimation [7] as the prerequisite [29], [28], [19]. However, the possibly bad result from pose estimation shall degrade the performance of human parsing. Second, some parsing methods, such as parselets [8] and co-parsing [30], which take advantage of the bottom-up hypotheses generation methods [2], are implemented based on a critical assumption that the objects or semantic regions have a large probability to be tightly covered by at least one of the generated hypotheses. This assumption does not always hold. When the semantic regions appear with larger appearance diversity, it is very difficult to obtain a single hypothesis to cover the whole region, as the object hypotheses by the over-segmentation tend to capture the appearance consistency other than the semantic meanings. Third, all existing methods do not sufficiently capture the complex contextual information among the key elements of human parsing, including semantic labels, label masks and their spatial layouts. We argue that human parsing can greatly benefit from the structural information among these elements. As shown in Fig. 1, the presence of the skirt (i.e., its visibility) will hinder the probability of the dress/pants, and meanwhile encourage the visibilities and constrain the locations of left/right legs in (a). For example, the mask of a specific label can also provide the informative guidance for predicting the masks and locations of other labels, especially for the neighboring regions. The mask of the upper-clothes is a single segment due to the presence of the skirt in (c), while the upper-clothes mask is composed of two separate regions due to the dress in (b). Without capturing such structure information, the methods based on low level pixel or region hypotheses are not fully capable of accurately predicting the masks of different labels.

Different from these previous works, we propose a novel end-to-end framework for human parsing and formulate it as an active template regression (ATR) problem. Instead of

- *X. Liang is with the School of Information Science and Technology, Sun Yat-sen University, and also with the Department of Electrical and Computer Engineering, National University of Singapore. E-mail: xdliang328@gmail.com.*
- *X. Shen and J. Yang are with the Adobe Research, San Jose, California. E-mail: {xshen, jiayang}@adobe.com.*
- *L. Lin is with the School of Advanced Computing, Sun Yat-Sen Unviersity, and also with the SYSU-CMU Shunde International Joint Research Institute, Shunde, China. E-mail: linliang@ieee.org.*
- *L. Liu, J. Dong, and S. Yan are with the Department of Electrical and Computer Engineering, National University of Singapore. E-mail: {llq667, timeflydj}@gmail.com, eleyans@nus.edu.sg.*
- *S. Liu is with the Institute of Information Engineering, Chinese Academy of Sciences, and also with the Department of Electrical and Computer Engineering, National University of Singapore. E-mail: fifthzombiesi@gmail.com.*

Manuscript received 1 Aug. 2014; revised 25 Dec. 2014; accepted 22 Feb. 2015. Date of publication 2 Mar. 2015; date of current version 6 Nov. 2015. Recommended for acceptance by D. Ramanan.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2408360

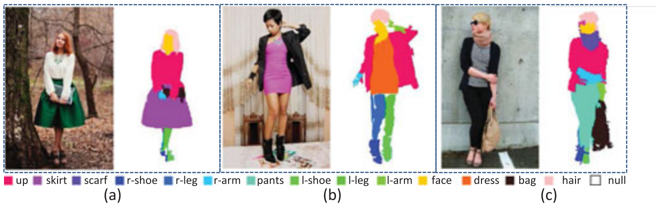


Fig. 1. Exemplar parsing results by our Active Template Regression (ATR) model. For better viewing of all figures in this paper, please see original zoomed-in color pdf file.

assigning a label to each pixel or hypothesis, we directly predict and locate the mask of each label. The parsing result for the test image is represented by the set of semantic regions (as in Fig. 2), which are morphed by the normalized masks with the corresponding active shape parameters, including the position, scale and visibility. In terms of the label mask generation, we first collect all the binary masks of the training images and then learn a batch of mask bases to construct the template dictionary for each label. Intuitively, the template dictionaries can be used to span the subspaces of label masks, which encode the shape priors of each label mask. Any mask with the specific shapes can be generated by adjusting the corresponding template coefficients, inspired by the classic active appearance model (AAM) [5] and active shape model (ASM) [6]. In this way, our representation is able to capture the natural variability within a set of mask templates for each label. The normalized mask of each label is thus expressed as the linear combination of the mask template dictionary and parameterized by the template coefficients. In terms of active shape parameters, we predict the positions, scales of each semantic region and the visibility flag which indicates whether the specific label appears in the image or not. In this paper, we denote the template coefficients and active shape parameters for each label as two types of structure outputs. Our active template regression framework targets on effectively regressing these structure outputs.

Inspired by its outstanding performance on traditional classification and detection tasks [27], [16], [31], we utilize the deep convolutional neural network (CNN) to build the end-to-end relation between the input human image and the structure outputs for human parsing, including the mask template coefficients and the active shape parameters. To predict the template coefficients, we aim to find the best linear combination of the learned mask templates. Larger coefficients indicate higher similarities between the label masks and the corresponding templates. The active shape parameters can be predicted similarly as the CNN-based detection task [27]. We thus use two separate networks, namely active template network and active shape network, to predict the structure outputs. First, the template coefficients of all labels are together regressed by using the designed active template network which is capable of capturing the contextual correlations among all label masks. Second, the active shape network is designed to predict the position, scale and visibility of each label. To make our active shape network sensitive to position variance, we eliminate the max-pooling layer in the traditional CNN infrastructure [16], which is often designed to be invariant to scale and translation changes. For a new photo, the structure outputs of the two networks are fused to generate the probability of each label for each pixel. The super-pixel smoothing is finally used to refine the parsing result.

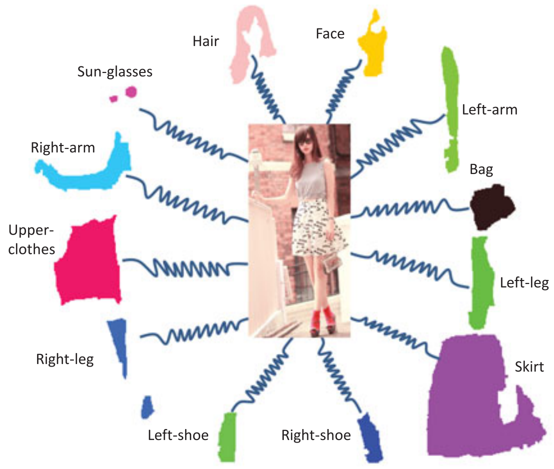


Fig. 2. Predicted label masks by our model. We directly predict each label mask and then morph them into the absolute image coordinates. Different colors indicate different labels.

To effectively train our networks, we conduct the experiments on a large dataset combining three public parsing dataset and our collected human parsing dataset. Comprehensive evaluations and comparisons well demonstrate the significant superiority of the ATR framework over other state-of-the-arts for human parsing. Furthermore, we also visualize our learned label masks, which demonstrate that our model can generate label masks with strong semantic meanings. Our contributions can be summarized as

- Our ATR framework provide an end-to-end approach for human parsing, which directly predicts the label masks and morphs them into the parsing result with active shape parameters. There is no need to explicitly design feature representations, the model topology or contextual interaction among labels.
- Our active template network can efficiently predict the most appropriate template coefficients for each label mask, represented by the linear combinations of the template dictionary.
- Our active shape network is designed to eliminate max-pooling for accurate position prediction and shows superiority in accurately regressing the active shape parameters over the generic network for classification [16].

2 Related Work

Many research efforts have been devoted into human parsing. Despite the important role of human parsing in many fashion-related and human-centric applications [4], [21], it has not been fully solved. Previous methods are generally based on two types of pipelines: the hand-designed pipeline and the deep learning pipeline.

2.1 Hand-Designed Pipeline

The traditional pipeline often requires many hand-designed processing steps to perform human parsing, each of which needs to be carefully designed and tuned [25], [1], [29], [11], [30], [19]. These steps use the low-level over-segmentation and pose estimation as the building blocks of human parsing. The classic composite And-Or graph template [3], [18] is utilized to model and parse clothing configurations. Yamaguchi et al. [29] performed human pose estimation and attribute labeling sequentially and then improved clothes parsing with a retrieval-based approach [28]. Dong et al. [8] proposed to use a group of parselets under the structure learning framework. However, such approaches based on hand-crafted relations often fail to fully capture the complex correlations between human appearance and structure. Although great progress has been achieved in human parsing, the involved representative model usually requires a lot of prior knowledge about the specific tasks, and these previous methods heavily rely on over-segmentation and pose estimation.

2.2 Deep Learning Pipeline

Recently, rather than using hand-crafted features and model representations, capturing contextual relations and extracting features with deep learning structures, especially deep convolutional neural network, have shown great potential in various vision tasks, such as image classification [16], [31], object detection [12], pose estimation [27]. To our best knowledge, convolutional neural network has not been applied to human parsing. However, there exist some works on scene parsing and object segmentation with CNN architectures. Farabet et al. [9] trained a multi-scale convolutional network from raw pixels to extract dense features for assigning the label to each pixel. However, multiple complex post-processing methods were required for accurate prediction. The recurrent convolutional neural network [24] was proposed to speed up scene parsing and achieved the state-of-the-art performance. Girshick et al. [12] also proposed to classify the candidate regions by CNN for semantic segmentation. All of these approaches use the CNNs as local or semi-local classifiers either over super-pixels or region hypotheses. However, our approach builds an end-to-end relation between the input image and the structure outputs, which is a more efficient application of CNN.

The above-mentioned hand-crafted and deep models share a similar pipeline: each image is decomposed into small units (pixels, super-pixels or region hypotheses) and local features (hand-crafted features or rich features learned by deep networks) are extracted; then the additional classifiers (shallow models like SVM, or deep models) are trained. In contrast, our approach builds an end-to-end relation between the input image and the structure outputs, which is simple and more efficient. Taking an image as the input, our deep model directly predicts the label masks and the corresponding shape parameters of each semantic region. All the components (e.g., hypothesis generation, feature-extraction and then classification) used in the traditional pipelines are integrated into one unified framework, which distinguishes us from all previous parsing approaches. The closest approaches to ours are [27], [26] which use CNN-based regression for predicting landmark locations and bounding boxes of the objects, respectively. Their approaches are intuitively similar with our active shape network except that our model eliminates the max-pooling layer for position effectiveness. Moreover, the other important component in our model (i.e., the active template network) is designed to predict the mask template coefficients to actively generate the arbitrary masks of the semantic labels.

3 Active Template Regression

We formulate the task of human parsing as an active template regression problem. Our framework targets on predicting two kinds of structure outputs: active template coefficients and shape parameters. First, for K different semantic labels (e.g. hair, hat, dress, etc.), we encode the normalized mask of each label as the linear combination of the mask template dictionary $D_k, k = 1, \ldots, K$. Each label mask is parameterized by the corresponding template coefficients, $\boldsymbol{\alpha}_k$, which are treated as the first type of structure outputs. Second, the position of each label mask is parameterized by its top-left coordinates $(b_k^x, b_k^y) \in \mathbb{R}^2$ as well as the width b_k^w and height b_k^h. The visibility flag v_k for each label indicates whether the label (e.g. hat, belt) appears in this image. The second type of structure outputs, the active shape parameters, can thus be represented as $\boldsymbol{s}_k = (b_k^x, b_k^y, b_k^w, b_k^h, v_k)$. Finally, the parsing result of the input image x is generated by morphing the masks of all K different labels with the corresponding active shape parameters. In this paper, we train these two types of structure outputs with two separate neural networks: active template network and active shape network, which predict the template coefficients $\{\boldsymbol{\alpha}_k\}_1^K$ and the active shape parameters $\{\boldsymbol{s}_k\}_1^K$, respectively. The reason for training two separate networks is that the learning of template coefficients and shape parameters can be treated as two different tasks: the first one is essentially selecting the most appropriate templates for reconstructing label masks with the template dictionaries, similar to the classification problem, and the second one aims at regressing the precise locations, similar to the detection problem.

As shown in Fig. 3, given an input image, we first detect the human body by using the state-of-the-art detector, i.e., the region convolution neural network method [12]. Considering that the detected bounding box of the human body may not contain all of the body parts, we thus enlarge the detected bounding box with the factor 1.2. The pixels outside the enlarged bounding box are regarded as the background. The normalized mask of each label is reconstructed by using the predicted template coefficients $\{\boldsymbol{\alpha}_k\}_1^K$ and the template dictionaries $\{D_k\}_1^K$. We then morph these masks into the absolute image coordinates indicated by the shape parameters $\{\boldsymbol{s}_k\}_1^K$. The confidence maps of each label and the background can be obtained according to the morphed masks. Finally, we use the super-pixel smoothing to generate and refine the final parsing result y.

3.1 Active Template Network

The masks of different individual semantic regions for the same label often show various shapes but also common patterns which can distinguish one label from the others. We can thus represent each label mask by the linear

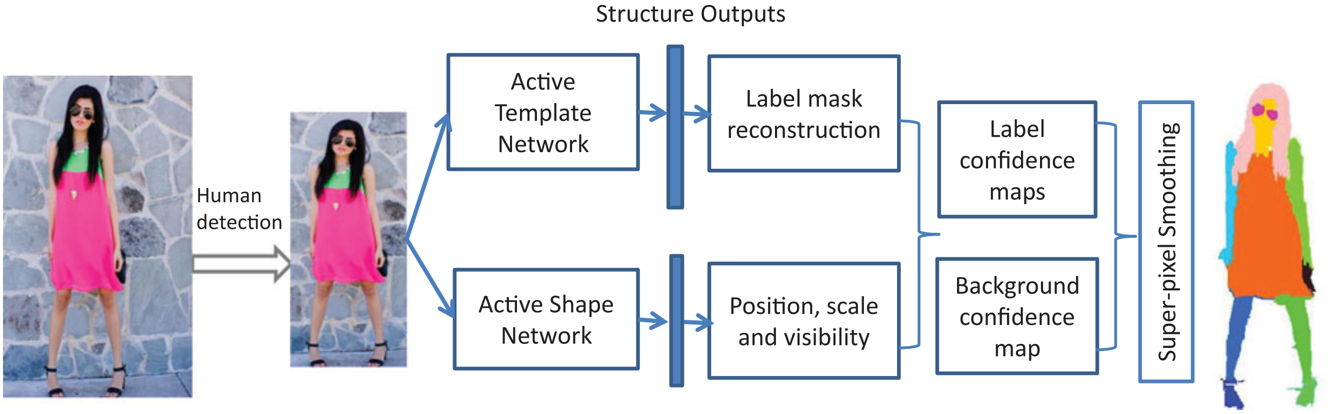


Fig. 3. Framework of our active template regression model. Given a test image, we first locate the bounding box for human body and then feed it into two separate networks. The template coefficients from the active template network are used to reconstruct the normalized mask. The masks of all labels are then fused together to generate the label confidence maps and the background confidence map by morphing with the active shape parameters (i.e., position, scale and visibility). The super-pixel smoothing is finally used to refine the parsing result.

combination of the corresponding template dictionary for each label and the label masks can be parameterized by the corresponding template coefficients to best fit the image. Intuitively, the template dictionaries span the subspace of the label masks and incorporate the shape priors for all labels. By selecting the appropriate template coefficients, we can obtain diverse semantic regions for each label. And the output size of the network can also be significantly reduced by using the template coefficients, rather than using all pixels of the whole mask.

The active template network is designed to predict the template coefficients. We first generate the mask template dictionaries $\{D_k\}_1^K$ for each label using dictionary learning. More precisely, given the set of training samples $\{x_i, y_i\}_1^n$, we first collect a set of ground-truth binary masks for all K labels. The mask set is denoted by $B_k = \{\boldsymbol{b}_{1,k}, \boldsymbol{b}_{2,k}, \ldots, \boldsymbol{b}_{n,k}\}$ for the kth label, where $\boldsymbol{b}_{i,k}$ represents the binary mask of the kth label from the sample (x_i, y_i). Specifically, for each label mask, values of the pixels assigned with the specific label are set as 1 and otherwise 0. The binary mask is obtained by the minimum bounding rectangle of the label mask. To learn the template dictionary for each label, we rescale all these cropped binary masks into a fixed width r^w and height r^h. We denote the dictionary for each label as $D_k \in \mathbb{R}^{Z \times M}$ where $Z = r^w \times r^h$, and M as the number of learned templates. The template coefficients of each training sample are denoted as $\boldsymbol{\alpha}_i = \{\boldsymbol{\alpha}_{i,k}\}_1^K$. To jointly predict the template dictionary D_k for each label and the template coefficients $\boldsymbol{\alpha}_{i,k} \in \mathbb{R}^M$, we optimize the following cost function for kth label,

$$\min_{\boldsymbol{D_k}, \boldsymbol{\alpha}_{i,k}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} ||\boldsymbol{b}_{i,k} - D_k \boldsymbol{\alpha}_{i,k}||_2^2 + \lambda ||\boldsymbol{\alpha}_{i,k}||_2^2, \tag{1}$$

where λ is the regularized parameter. It is well-known that ℓ_1 penalty yields a sparse solution for $\boldsymbol{\alpha}_{i,k}$. However, our active template network with sparse solution may be difficult to converge because of the dominance of the zero values. We thus use the ℓ_2-norm to regularize the template coefficients. Our experiments demonstrate the superiority with the ℓ_2-norm than the ℓ_1-norm. Moreover, we constrain D_k and $\boldsymbol{\alpha}_{i,k}$ to be non-negative, which can help our network generate more reasonable mask templates with semantic meanings than the traditional principal component analysis (PCA) [15] methods with both negative and non-negative values [17]. Specifically, the NMF learns part-based decompositions for covering diverse visual patterns of each label and the additive combinations of active templates are beneficial for our reconstruction and network optimization. This non-negative matrix factorization (NMF) problem can thus be effectively solved by the on-line dictionary learning based on stochastic approximations [22].

We normalize the coefficient values $\boldsymbol{\alpha}_{i,k}$ into the Gaussian distribution with the mean $\boldsymbol{\mu}_k$ and standard deviation $\boldsymbol{\sigma}_k$ for each label. Let $\boldsymbol{\mu}_k = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\alpha}_{i,k}$ and $\boldsymbol{\sigma}_k = \sqrt{\frac{1}{n}\sum_{i=1}^n ||\boldsymbol{\alpha}_{i,k} - \boldsymbol{\mu}_k||^2}$. The normalized temporal coefficients $\hat{\boldsymbol{\alpha}}_{i,k}$ can be defined

$$\hat{\boldsymbol{\alpha}}_{i,k} = \frac{\boldsymbol{\alpha}_{i,k} - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k}. \tag{2}$$

We train our active template network to predict the normalized coefficients $\hat{\boldsymbol{\alpha}}_i$ based on the convolutional neural network. The convolutional network consists of several layers and each layer is a linear transformation followed by a non-linear one. The first layer takes an $227 \times 227 \times 3$ input image as the input. The last layer outputs the target values of the regression, in our case $M \times K$ dimensions for all labels. Our network is based on the architecture used by Zeiler and Fergus [31] for image classification since it has shown better performance on the ImageNet benchmark than the one used by Krizhevsky et al. [16]. Each layer consists of: (1) convolution of the previous layer output (or, in the case of the first layer, the image) with a set of filters; (2) passing the responses through a rectified linear function; (3) (optionally) max pooling over local neighborhood; (4) (optionally) the local contrast function that normalizes the responses across feature maps. The top few layers of the network are fully-connected and the final layer is an ℓ_2-norm regressor. We refer the reader to Zeiler and Fergus [31] and Krizhevsky et al. [16] for more details. Fig. 4 shows the model used in our active template network. The

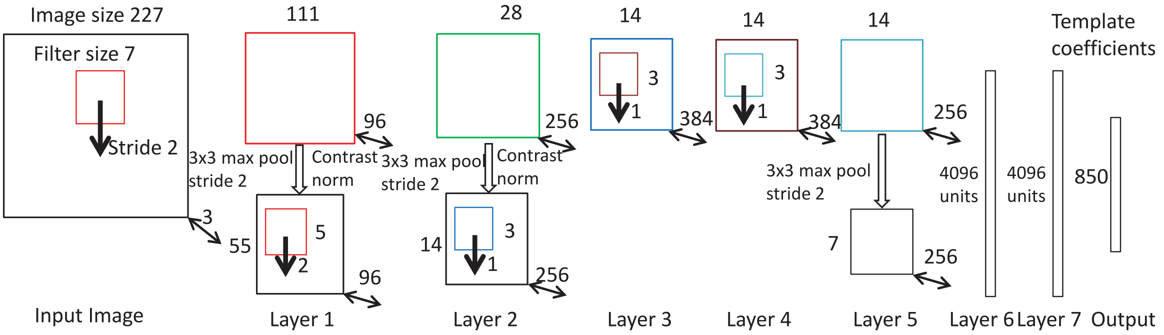


Fig. 4. Our active template network. A $227 \times 227 \times 3$ image is taken as the input. We convolve it with 96 different first layer filters (red), each of which with the size 7×7, using a stride of 2. The obtained feature maps are then: (1) passed through a rectified linear function (not shown), (2) max pooled (within 3×3 filter, using stride 2) and (3) contrast normalized. Similar operations are repeated in the second, third, fourth, fifth layers. The last two layers are fully-connected, taking features from the top convolutional layer. The output layer with $850 = 17 \times 50$ units is a regression function with ℓ_2-norm for $K = 17$ labels and each with $M = 50$ coefficients.

difference from [31] is the loss function we use. Instead of a classification loss, we predict the normalized coefficients by minimizing ℓ_2 distance between the prediction and the ground truth coefficients. Suppose the predicted coefficients are denoted as $\bar{\boldsymbol{\alpha}}_{i,k}$ and the ground truth coefficients as $\hat{\boldsymbol{\alpha}}_{i,k}$. The ℓ_2 loss is defined as

$$J = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}||\hat{\boldsymbol{\alpha}}_{i,k} - \bar{\boldsymbol{\alpha}}_{i,k}||^2. \tag{3}$$

The network parameters (filters in the convolutional layers, weight matrices in the fully-connected layers and biases) are trained by Back-propagation. For the simplicity, we eliminate the subscript i for each image in the following. Given an input image x, our active template network can predict template coefficients $\{\bar{\boldsymbol{\alpha}}_k\}_1^k$ for all labels and then we obtain the absolute coefficients $\{\tilde{\boldsymbol{\alpha}}_k\}_1^k$ by using the inverse function of Eq. (2). The normalized mask $\boldsymbol{b}_k$ for each label can be reconstructed by the linear combination of the specific template dictionary with $\tilde{\boldsymbol{\alpha}}_k$, as $\boldsymbol{b}_k = D_k\tilde{\boldsymbol{\alpha}}_k$.

3.2 Active Shape Network

After obtaining the normalized mask of each label, we need to morph them into the more precise masks at accurate positions in the image. In this paper, we denote the positions, scales and visibilities of the label masks as the active shape parameters $\{s_k\}_1^K$, predicted by our active shape network. The structure outputs $\boldsymbol{s}_k = (b_k^x, b_k^y, b_k^w, b_k^h, v_k)$ include the top-left coordinates $(b_k^x, b_k^y) \in \mathbb{R}^2$, the width b_k^w, the height b_k^h and the visibility flag v_k which is set as 1 if the kth label appears in the image.

Fig. 5 shows the architecture of our active shape network. The first convolutional layer filters a $227 \times 227 \times 3$ input image with 48 kernels of size $7 \times 7 \times 3$ with a stride of 2 pixels. The second convolutional layer takes the rectified output of the first convolutional layer as the input and filters it with 128 kernels of size $5 \times 5 \times 48$ with a stride of 2 pixels. The third, fourth and fifth convolutional layers are connected to one another, and the third and fourth layers are also with a stride of 2 pixels. The last two fully-connected layers have $2{,}048$ and $1{,}024$ units, respectively. The output layer predicts $\{s_k\}_1^K$ for all labels, resulting in 85 units. Furthermore, since the positions and scales are in absolute coordinates, it will be beneficial to normalize them with respect to the mean and standard deviations of positions and scales, similar as in Eq. (2). We keep the original values of visibility flags which are either 1 or 0. We minimize ℓ_2 distance between the prediction and the ground truth parameters. Suppose the predicted parameters are denoted as $\bar{s}_k$ and the ground truth parameters as $\hat{s}_k$. The corresponding ℓ_2 loss is defined as

$$J = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}||\hat{\boldsymbol{s}}_k - \bar{\boldsymbol{s}}_k||^2. \tag{4}$$

The previous infrastructures for the classification tasks [16], [31] include the max-pooling layer to make the

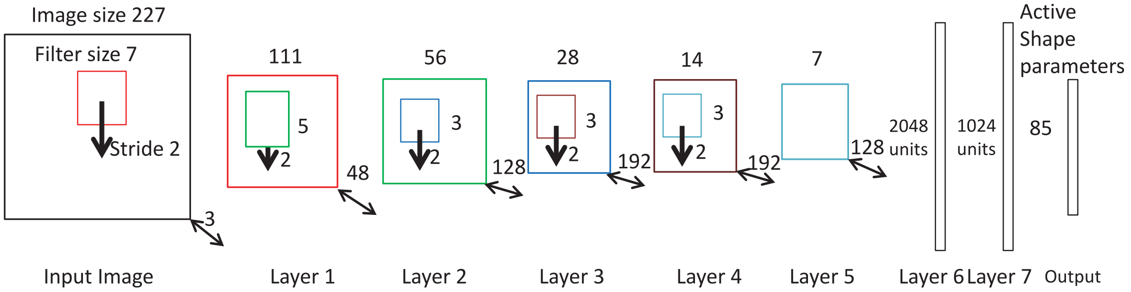


Fig. 5. Architecture of our active shape network. We take a $227 \times 227 \times 3$ image as the input and convolve it with 48 different frist layer filters (red), each of which with the size 7×7, using a stride of 2 in both dimensions. The obtained feature maps are then passed through a rectified linear function (not shown) to get 48 different 111×111 feature maps. Similar operations are repeated in second, third, fourth, fifth layers. The last two layers are fully-connected with 2,048 units and 1,024 units, respectively. The output layer with $85 = 17 \times 5$ units is a regression function with ℓ_2-norm for $K = 17$ semantic labels and each with five dimensions, including positions, scales and visibility flag.

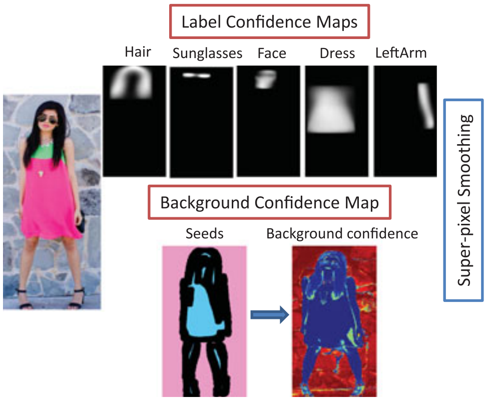


Fig. 6. Our structure output combination. The confidence maps of all foreground labels are predicted by fusing two types of structure outputs. Then we can produce the background confidence map: we first generate the foreground (blue pixels) and background seeds (pink pixels) and then predict the background confidence map (the red colored pixels have the highest probability for background). Finally the superpixel smoothing is used to refine the parsing result.

network invariant to scale/translation changes and reduce the scale of feature maps. However, our network for regressing shape parameters is sensitive to position variance. To remedy this problem, our network eliminates the pooling layer and keeps the same overall depth of the network with [31]. The new architecture retains much more information in the first few layers (e.g. the feature map with size 111×111 versus 55×55 in the first layer and 56×56 versus 14×14 in the second layer, compared with the model in Fig. 4). Additionally, we reduce the scale of feature maps gradually, using a stride of 2 pixels as well in the second, third, and fourth layers. Given that our dataset is much smaller than the ImageNet dataset, we decrease the filter number in each convolution layer and the size of the fully-connected layers to prevent over-fitting.

The contextual interactions between all semantic label masks (e.g. label exclusiveness and spatial layouts) are intrinsically captured by all of the hidden layers. Given a test image x, the active shape network predicts shape parameters $\{\bar{s}_k\}_1^K$ for all label masks and the absolute image coordinates $\{\tilde{s}_k\}_1^k$ are obtained by using the inverse normalization.

Bounding box refinement. In addition, considering the prediction error of shape parameters, we utilize the bounding box refinement to further reduce the mislocalizations. Specifically, we train K linear regression models to predict new positions (e.g., $b_k^x, b_k^y, b_k^w, b_k^h$) for all labels, following the method proposed for object detection [12]. To train the bounding box regressor for each label, all the training images are cropped around the predicted positions and then enlarged by a factor of 1.5 to contain more surrounding context. The input for training is a set of training pairs, i.e., the predicted positions from our network and the ground-truth bounding boxes for each label. Note that only the predicted label mask which has an over 0.5 overlap ratio with the ground-truth box is considered. The features for each training image are extracted from the outputs of the fully-connected layer of the ImageNet model [16]. Finally, we use the same strategy to learn the position transformation. Please refer to [12] for more details.

3.3 Structure Output Combination and Super-Pixel Smoothing

After feeding the image into the above two networks, we can obtain the normalized mask $\boldsymbol{b}_k$ and the shape parameters $\boldsymbol{s}_k$ for each label. The confidence map $\boldsymbol{c}_k$ of each label k is obtained by morphing the mask $\boldsymbol{b}_k$ into the absolute image coordinates with $\boldsymbol{s}_k$. Note that the visibility flag v_k denotes whether the kth label appears in the image or not. Only if the visibility flag satisfies $v_k \geq 0.5$, the associated masks are considered. Note that this threshold is only used to prune the less likely appeared label masks. The final label masks are mainly decided by the predicted template coefficients and active shape parameters.

Our network can only predict the confidence maps of all foreground labels. For the background label, we predict its probability for each pixel by adopting the interactive image segmentation method [13]. We automatically obtain the reliable foreground and background seeds from the confidence maps of all labels. Specifically, we first calculate the foreground confidence map $\boldsymbol{c}_f$ by maximizing the confidences of each label as $\boldsymbol{c}_f = \max_{k=1}^K \boldsymbol{c}_k$. Only the pixels of $\boldsymbol{c}_f$ with the confidence larger than 0.5 is regarded as the foreground. Then the erode operation with the filter size 10 based on the foreground mask is performed to produce the foreground seeds, displayed as the blue pixels of seed images in Fig. 6. The background seeds are obtained by dilating the inverse of the foreground mask within 10 neighborhoods, displayed as the pink pixels of seed images in Fig. 6. Based on the seeds, we can predict the background confidence map by learning the color model as in [13].

Super-pixel smoothing. To combine the confidence maps of all semantic labels and the background, we apply superpixel smoothing and refine the parsing results for more precise pixel-level segmentation. In particular, our approach first computes an over-segmentation of the input image using a fast segmentation algorithm [10]. We denote the background label as $k = 0$ and thus we have $K + 1$ possible labels for each pixel i'. The confidence map set is denoted as $C = \{\boldsymbol{c}_k\}_{k=0}^K$, where $\boldsymbol{c}_0$ is the obtained background confidence map using [13]. The super-pixel which contains the pixel i' is defined as $q_{i'}$ and the predicted label of the pixel i' is denoted as $y_{i'}$. Our final parsing result is thus calculated as

$$y_{i'} = \max_k \sum_{j' \in q_{i'}} \boldsymbol{c}_k(j'), \tag{5}$$

where j' denotes each pixel in the super-pixel $q_{i'}$ and $\boldsymbol{c}_k(j')$ is the probability of the pixel j' in the map $\boldsymbol{c}_k$. Since we only perform the maximization of the average confidences for all labels, our super-pixel smoothing method is very simple and fast.

4 Experiments

4.1 Experimental Settings

Datasets. A large number of training samples are required for most of the deep models [16], [12]. However, existing public available datasets for human parsing are relatively small. The largest existing human parsing dataset, to our best knowledge, contains only $2{,}682$ images, which is insufficient for training a robust deep network model. Thus, we

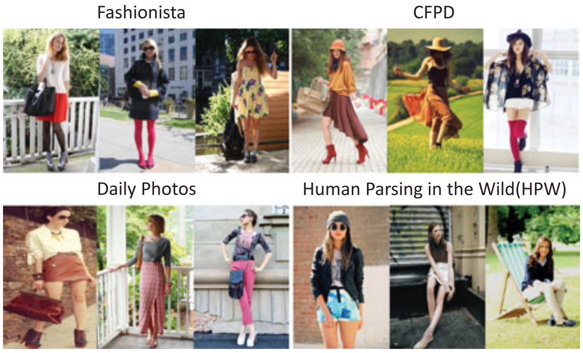


Fig. 7. Exemplar images in the combined dataset.

combine data from three small benchmark datasets: (1) the Fashionista dataset [29] containing 685 images, (2) the colorful fashion parsing data (CFPD) dataset [19] containing 2,682 images, and (3) the Daily Photos dataset [8] containing 2,500 images. All images in these three datasets contain standing people in frontal/near-frontal view with good visibilities of all body parts. Following the label set defined by Dong et al. [8], we merge the labels of Fashionista and CFPD datasets to 18 categories: face, sunglass, hat, scarf, hair, upper-clothes, left-arm, right-arm, belt, pants, left-leg, right-leg, skirt, left-shoe, right-shoe, bag, dress and background. To enlarge the diversity of our dataset, we crawl another 1,833 challenging images to construct the human parsing in the wild (HPW) dataset and annotate pixel-level labels following [8]. As shown in Fig. 7, our newly annotated data are mainly more realistic images containing challenging poses (e.g. sitting) and occlusion, which is a good supplement to the existing three datasets. The final combined dataset from the four datasets contains 7,700 images. We use 6, 000 images for training, 1,000 for testing and 700 as the validation set. The occurrences of each label in our collected dataset are reported in Table 2. For fair comparison with published algorithms, we use the same evaluation criterion as in [28], which contains accuracy, average precision, average recall, and average F-1 scores over pixels.

Data augmentation. To reduce over-fitting on image data, we manually enlarge the training data to increase the diversity using the translations and horizontal reflections. Specifically, we first detect the bounding box of the human body [12] and then incrementally cover more context outside the box with the stride of 20 pixels in eight directions (i.e., top/down, left/right, topleft/topright, downleft/downright). In addition, we enlarge the scale of the detected bounding box with three factors, i.e., 1.2, 1.5, 1.8. The horizontal reflections are used for all the cropped images. Then we resize all these images into $227 \times 227 \times 3$ using the nearest-neighbor interpolation. This increases size of our training set by a factor of $24 = (8 + 3) \times 2$. Although the resulting training examples are highly inter-dependent, the data augmentation can significantly increase the diversity of features, especially for predicting the active shape parameters.

Implementation details. Our two networks aim to predict the masks and shape parameters of $K = 17$ labels. To learn the template dictionary for each label, we normalize the binary mask into a regularized size r^w and r^h as 100 and the template number M as 50. The penalty λ for the NMF is set as 0.001. When the training image does not have certain labels, we set their corresponding template coefficients and shape parameters as zeros. We implement the two networks under the caffe framework [14] and train them using stochastic gradient descent with a batch size of 128 examples, momentum of 0.9, and weight decay of 0.0005. We use an equal learning rate for all layers and adjust it manually. The strategy is to divide the learning rate by 10 when the validation error rate stops decreasing with the current learning rate. The learning rate is initialized at 0.0005 for the two networks. We train the networks for roughly 120 epochs, which takes two to three days on one NVIDIA GTX TITAN 6 GB GPU. Our algorithm can rapidly process one 227×227 image within about 0.5 second, as measured on a NVIDIA GTX TITAN 6 GB GPU. This compares favorably to other approaches, as some of the current state-of-the-art approaches have higher complexity: [29] runs in about 10 to 15 seconds, while [8] runs in 1 to 2 minutes.

4.2 Results and Comparisons

We compare our ATR framework with the two state-of-the-art works [28], [29]. We use their public available codes and carefully tune the parameters according to [29], [28] and train their models with the same training images as our method for the fair comparison. Note that Dong et al. [8] is not compared in this work because our experiments show the PaperDoll [28] can achieve the accuracy of 87.8 percent on the 229 test images of the Fashionista data set, which is better than the accuracy of 86 percent reported in [8] with the same label set. We implement two versions of our method. (1) “ATR (noSPR)”: the parsing results are obtained by maximizing the all confidence maps where no superpixel refinement (SPR) is used. (2) “ATR”: we refine the parsing results with the super-pixel smoothing. The results are listed in Table 1.

The method of Yamaguchi et al. [29] and the PaperDoll [28] with 456 training images as on the public Fashionista dataset achieve 35.78 and 37.54 percent of average F1-score on evaluating our 1,000 test images, respectively. When training the model with more data (e.g., 6,000 images), the performances of the two baselines can be increased by 6.02 [29] and 7.22 percent [28]. However, our “ATR” can significantly outperform these two baselines by over 22.58 percent for Yamaguchi et al. [29] and 19.62 percent for PaperDoll [28]. Our method also gives a huge boost in foreground accuracy: the two baselines achieve 55.59 percent for Yamaguchi et al. [29] and 62.18 percent for PaperDoll [28] while “ATR” obtains 71.04 percent. “ATR” also obtains much higher precision (71.69 percent versus 37.54 percent for [29] and 52.75 percent for [28]) as well as higher recall (60.25 percent versus 51.05 percent for [29] and 49.43 percent for [28]). The pixel-level accuracy is also increased by at least 2.15 percent. This verifies the effectiveness of our algorithm though it does not require explicit definition of any contextual relations and incorporation of complicated prior knowledge. For “ATR (noSPR)”, it also achieves superior performance than the baselines. The superiority of “ATR (noSPR)” over the baselines demonstrates that our network has the capability of directly predicting reasonable label masks without any low-level segmentation methods which are commonly used by all previous methods. The

TABLE 1
Comparison of Parsing Performances with Several Architectural Variants of Our Model and Two State-of-the-Arts

Method	**Accuracy**	**F.g. accuracy**	**Avg. precision**	**Avg. recall**	**Avg. F-1 score**
Yamaguchi et al. [29] (456)	82.54	46.70	31.67	43.74	35.78
PaperDoll [28] (456)	86.74	50.34	43.38	41.21	37.54
Yamaguchi et al. [29] (6,000)	84.38	55.59	37.54	51.05	41.80
PaperDoll [28] (6,000)	88.96	62.18	52.75	49.43	44.76
Yamaguchi et al. [29] (6,000 test 229)	87.87	58.85	51.04	48.05	42.87
PaperDoll [28] (6,000 test 229)	89.98	65.66	54.87	51.16	46.80
ATR (unified)	84.95	45.65	51.90	33.07	38.62
ATR (PCA)	86.43	52.83	63.50	43.39	48.87
ATR (NMFℓ_1)	88.49	61.44	62.00	49.64	53.77
ATR (zeilernet)	88.59	60.77	62.66	48.55	53.62
ATR (lessfc)	90.16	67.74	68.17	56.59	60.50
ATR (lessfcfilters)	90.21	67.17	69.16	56.04	60.77
ATR (nopool)	91.01	70.40	69.61	58.82	62.78
ATR (noSPR)	89.33	64.79	63.75	56.19	59.60
ATR	**91.11**	**71.04**	**71.69**	**60.25**	**64.38**
ATR (test 229)	92.33	76.54	73.93	66.49	69.30
Upperbound	98.67	93.61	95.45	92.79	94.04

improvements from "ATR (noSPR)" to "ATR" show that the super-pixel smoothing enables the parsing result to preserve more accurate boundary information. For the fair comparison, we also report the parsing results on the 229 test images of the Fashionista dataset [29]. Our method "ATR (test 229)" can also significantly outperform these two baselines by over 26.43 percent for "Yamaguchi et al. [29] (6,000 test 229)" and 22.5 percent for PaperDoll [28] (6,000 test 229)" of average F1-score on evaluating 229 test images. This speaks well that our collected dataset contains much more realistic images with the challenging poses and occlusions than the small Fashionista dataset [29].

We also present the F1-scores for each label in Table 2. Generally, both versions of our method show much higher performance than the baselines. In terms of predicting small labels such as hat, belt, bag and scarf, our method achieves a large gain, e.g. 57.07 percent versus 11.43 percent [29] and 2.95 percent [28] for scarf, 53.66 percent versus 24.53 percent [29] and 30.52 percent [28] for bag. It demonstrates that our two networks can capture the internal relations between the labels and robustly predict the label masks with various clothing styles and poses. The qualitative comparison of parsing results is visualized in Fig. 9. Our methods predict much more reasonable and meaningful label masks than the PaperDoll method [28] despite their large appearance and position variations. We can successfully predict small labels (e.g. sun-glasses, hat) while the PaperDoll [28] often fails and confuses them with the neighboring regions. For example, for the left image of the third row in Fig. 9, we can detect sunglasses and hat while the PaperDoll totally misses

TABLE 2
F-1 Scores of Foreground Semantic Labels

Method	Hair	Bag	Belt	Dress	Face	Hat	L-arm	L-leg	L-shoe	Pants	R-arm	R-leg	R-shoe	Scarf	Skirt	S-gls	U-cloth
Label occurrences	7,059	3,517	1,952	2,303	7,387	1,918	6,956	5,330	6,146	3,501	6,615	5,571	6,203	440	2,484	2,221	5,933
Yamaguchi et al. [29] (6,000)	59.96	24.53	14.68	40.94	72.10	8.44	45.33	58.52	38.24	55.42	46.65	57.03	38.33	11.43	17.57	12.09	56.07
PaperDoll [28] (6,000)	63.58	30.52	16.94	59.49	61.63	1.72	45.23	52.19	45.79	69.35	46.75	55.60	44.47	2.95	40.20	0.23	71.87
Yamaguchi et al. [29] (6,000 test 229)	62.58	27.31	18.50	54.26	60.26	1.48	42.96	47.93	44.83	66.37	45.17	52.22	44.01	2.44	35.49	0.19	68.98
PaperDoll [28] (6,000 test 229)	64.45	31.22	16.78	65.42	62.32	2.12	48.20	56.16	46.79	73.51	48.62	58.35	45.40	3.93	47.17	0.28	74.36
ATR (unified)	58.87	4.47	11.74	59.25	63.74	34.10	30.64	45.96	18.28	46.60	31.27	50.69	20.08	19.25	35.62	1.79	69.33
ATR (PCA)	59.12	45.53	5.27	53.74	65.42	42.06	51.60	64.04	47.47	60.84	49.76	60.59	43.88	17.63	43.29	5.37	69.73
ATR (NMFℓ_1)	56.04	42.06	5.70	74.90	63.74	65.47	50.84	62.90	47.16	70.26	43.42	61.96	46.62	28.57	73.34	7.58	72.28
ATR (zeilernet)	63.78	31.33	1.13	73.43	69.02	63.82	45.89	60.86	41.83	70.18	42.74	65.68	38.84	46.89	66.02	13.37	74.87
ATR (lessfc)	67.55	39.42	17.79	77.85	72.28	71.26	51.13	63.90	52.09	77.82	51.75	69.12	44.63	60.58	79.13	18.44	78.02
ATR (lessfcfilters)	67.54	36.93	21.80	78.10	72.12	73.26	57.23	66.43	50.73	76.39	55.44	67.30	48.74	47.29	77.83	22.66	77.58
ATR (nopool)	**71.67**	**56.59**	14.31	**82.15**	**76.53**	59.18	**57.41**	**69.36**	47.73	77.94	**60.73**	69.98	48.72	53.16	79.89	33.69	**79.50**
ATR (noSPR)	69.11	49.79	18.00	76.63	74.55	68.61	49.17	59.95	47.21	72.29	52.07	63.04	45.87	45.85	73.87	**35.66**	75.21
ATR	68.18	53.66	**22.88**	82.02	74.71	**77.97**	53.79	69.07	**53.51**	**79.77**	58.57	**71.69**	**50.26**	**57.07**	**80.36**	29.20	79.39
ATR (test 229)	69.35	66.91	30.50	85.38	78.48	77.14	64.37	74.56	57.76	82.96	63.25	76.07	55.87	63.26	83.35	38.14	82.77

Comparison of F1-scores with several architectural variants of our model and two state-of-the-art methods.

TABLE 3
Detailed Experimental Settings by Varying the Model Architectures of Our Networks

		Active Template		Active Shape				Structure Output Combination	
	ATR (unified)	ATR (PCA)	ATR (NMFℓ_1)	ATR (zeilernet)	ATR (lessfc)	ATR (lessfcfilters)	ATR (nopool)	ATR (noSPR)	ATR
Template generation	NMF	PCA	NMF with ℓ_1-norm	NMF	NMF	NMF	NMF	NMF	NMF
AT net	No	ours	ours	ours	ours	ours	ours	ours	ours
AT output num	No	850	850	850	850	850	850	850	850
AS net	NA	ours+BB	ours+BB	Our replication of [31]	Adjust layers 6,7: 2,048,1,024 units (based on [31])	Adjust layers 1-5: 48,128,192,192,128 maps and layers 6,7: 2,048,1,024 units (based on [31])	ours	ours+BB	ours+BB
AS output num	NA	85	85	85	85	85	85	85	85
Unified network	Our replication of [31]	NA	NA	NA	NA	NA	NA	NA	NA
Unified output num	935	NA	NA	NA	NA	NA	NA	NA	NA
Structure Output Combination	SPR	SPR	SPR	SPR	SPR	SPR	SPR	MAP	SPR

them. The parsing results of our methods are cleaner and label masks bears strong semantic meanings while the results of [28] are heavily influenced by the low-level information, such as image clarity and color similarity. It demonstrates that our framework performs better in solving the high-level human parsing problem than the models based on low-level features. Finally, comparing the results of “ATR (noSPR)” and “ATR”, we can find that “ATR” can provide refined parsing results with respect to the region boundary. For example, for the left image in the first row in Fig. 9, “ATR” with super-pixel smoothing can effectively fill the gaps between the shoes and pants.

4.3 Ablation Studies of Our Networks

We further evaluate the effectiveness of our two components of ATR, including the active template network and the active shape network, respectively.

Active template network. To justify the rationality of using the template coefficients rather than the binary label masks, we test the reconstruction errors in dictionary learning, named as “Upperbound”. The label masks are reconstructed using the ground truth template coefficients and the learned dictionaries, and all active shape parameters are fixed. Table 1 shows that our “Upperbound” can achieve 98.67 percent in accuracy and 95.45 percent in average precision. This well demonstrates that the strategy of representing the binary masks with the corresponding coefficients results in very few reconstruction errors.

We also evaluate other mask reconstruction approaches: “ATR (PCA)” and “ATR (NMFℓ_1)”. The results are listed in Tables 1 and 2. Table 3 shows the details of experimental settings. First, we use the principal component analysis method [15] for dictionary learning instead of the NMF, named as ”ATR (PCA)”. The same number of bases (i.e., 50 for each label) as in NMF is selected to construct the template dictionary. ”ATR (PCA)” results in accuracy decrease by 4.68 percent as well as 15.51 percent in average F1-score, compared with “ATR”. PCA can be viewed as the eigenvector-based multivariate analysis that projects the data using only a few principle components, and the reconstruction coefficients and the basis vectors are either negative or positive. However, the NMF can learn the part-based decompositions and only additive combinations of templates are allowed, which is beneficial for our reconstruction. We also visualize our learned templates of each label as Fig. 10 shows. Most of the learned templates are in good shapes and bear strong semantic meanings. In addition, the templates are very diverse that can capture the large variances of label masks. These results verify that the nonnegative basis vectors can generate more expressiveness in the reconstruction. Second, to evaluate the effects of different norms upon the template coefficient prediction in Eq. (1), we use the ℓ_1-norm for “ATR (NMFℓ_1)” to yield more sparse template coefficients. Even though the ℓ_1-norm has shown promising results in image reconstruction [23] and is commonly used in a wide range of computer vision problems, its performance is inferior to the “ATR (SPR)” that uses the ℓ_2-norm to constrain too many sparse values, that is, 88.49 percent versus 91.11 percent in accuracy. The possible reason may be that our network can hardly predict optimal values with the sparse coefficients which contain too many zeros.

Fig. 8 visualizes the predicted label masks for six semantic labels with our active template network. The pixel in each mask with brighter color indicates its larger probability to be the specific label. Our network performs well in predicting the various shapes of the label masks. In particular, the predicted masks of ”hat” and ”hair” are highly consistent with the ground truth masks, and the fine-grained shapes for each label can also be visually distinguished (e.g. long hair versus short hair). For example, the third row in Fig. 8 shows several scarfs of different shapes. Even though the first scarf contains two disconnected regions and the second scarf is an entire region, our network can actively predict their respective shapes.

Active shape network. In Tables 1 and 2, we also explore other model architectures for regressing the active shape parameters by adjusting the layer size gradually. We evaluate four cases of architectures: 1) “ATR (zeilernet)” which

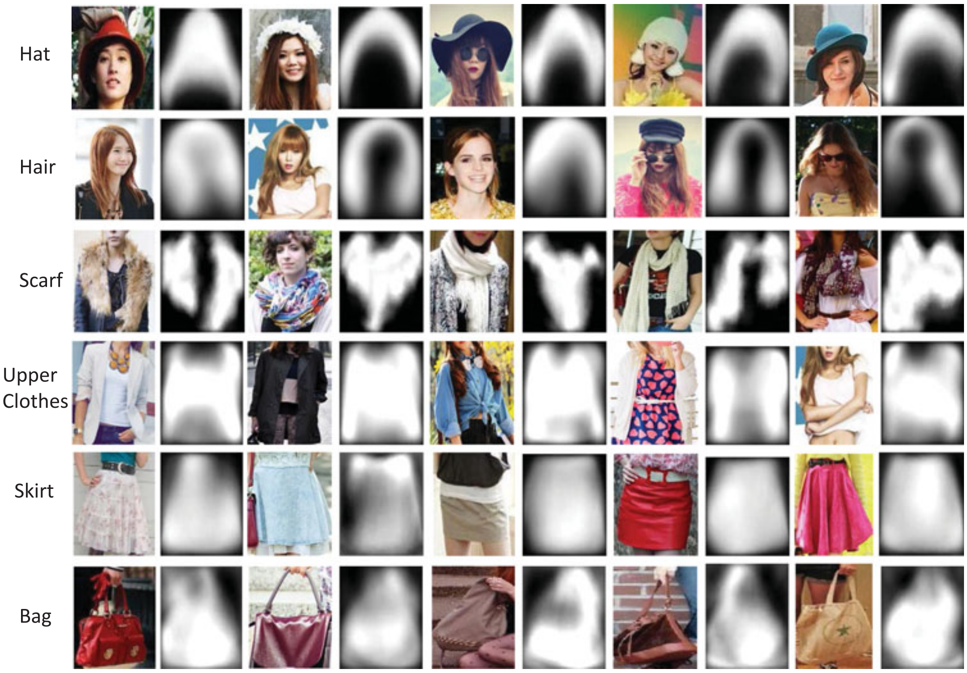


Fig. 8. Visualization of our predicted label masks with the active template network. We take the six semantic labels as the examples, such as hat, hair, scarf, upper-clothes, skirt and bag. The pixel with brighter color indicates that it is more likely to be assigned as the specific label.

follows the model architecture in [31]; 2) "ATR (lessfc)" where the size of two fully-connected layers is changed into 2,048 and 1,024 from the original 4,096, respectively; 3) "ATR (lessfcfilters)" where the number of filter maps is decreased by half, and also the size of fully-connected layers is changed as in "ATR (lessfc)"; 4) "ATR (nopool)" where the max-pooling layer is eliminated and the feature map size is gradually reduced by using the stride in the convolution layers, i.e., our proposed active shape network. The performances of these settings are evaluated without the bounding-box refinement. In "ATR (lessfc)" and "ATR (lessfcfilters)", the model is trained from the scratch with the architecture in [31]. Please refer to Table 3 for more details of experimental settings. The "ATR (zeilernet)" which uses the well-performed model infrastructure in image classification [31] gives inferior performance to our network "ATR (nopool)" (88.59 percent versus 91.01 percent in accuracy and 53.62 percent versus 62.78 percent in average F1-score). The main reason may be that the model for classification is not optimal for predicting our shape parameters which are sensitive to position variances. Besides, our dataset is much smaller than the ImageNet dataset. Using large layer size may result in over-fitting for our model. Thus we decrease the size of fully-connected layers, since they contain the majority of model parameters. The resulting accuracy and average F1-score of "ATR (lessfc)" show significant increase by 1.57 and 6.88 percent, respectively, compared to "ATR (zeilernet)". The "ATR (lessfcfilters)" which decreases the number of filter maps yields slight performance improvements, but largely decreases the training parameter number. This suggests that a small number of filter maps is enough for training our model. Based on the "ATR (lessfcfilters)", our final network "ATR (nopool)" eliminates the max-pooling operation, such that more information is reserved in the first few layers. "ATR (nopool)" gives a large gain in performance compared with "ATR (lessfcfilters)" (91.01 percent versus 90.21 percent in accuracy and 62.78 percent versus 60.77 percent in average F1-score). This verifies the effectiveness of eliminating max-pooling layers for solving the position sensitive problems. Moreover, we test the effectiveness of the bounding box regression for obtaining better shape parameters, by comparing the results of "ATR (nopool)" and "ATR". It shows that the bounding box refinement improves the average F1-score of "ATR (nopool)" by 1.6 percent by using fine-tuned active shape parameters of semantic labels.

Discussion. We evaluate the performance of training one unified network for regressing the template coefficients and active shape parameters. The "ATR (unified)" version follows the network infrastructure in [31] and targets on predicting all the structure outputs together. More details are presented in Table 3. The reported results in Table 2 are much worse than all other versions, especially than "ATR" (84.95 percent versus 91.11 percent in accuracy and 38.62 percent versus 64.38 percent in average F1-score). The reason for the inferiority of the unified network may be that the learning of template coefficients and active shape parameters can be treated as two different tasks and often require different network architectures, as we design. The first task with max-pooling is essentially selecting the most appropriate templates for reconstructing label masks with the template dictionaries and the second one without max-pooling aims at predicting the precise locations. Particularly, our framework with two separated networks has shown significant improvement on performance than previous work [28] (increasing by 19.62 percent of F1-scores). The network for regressing active template coefficients and shape parameters together may further improve the performance by incorporating the complicated contextual

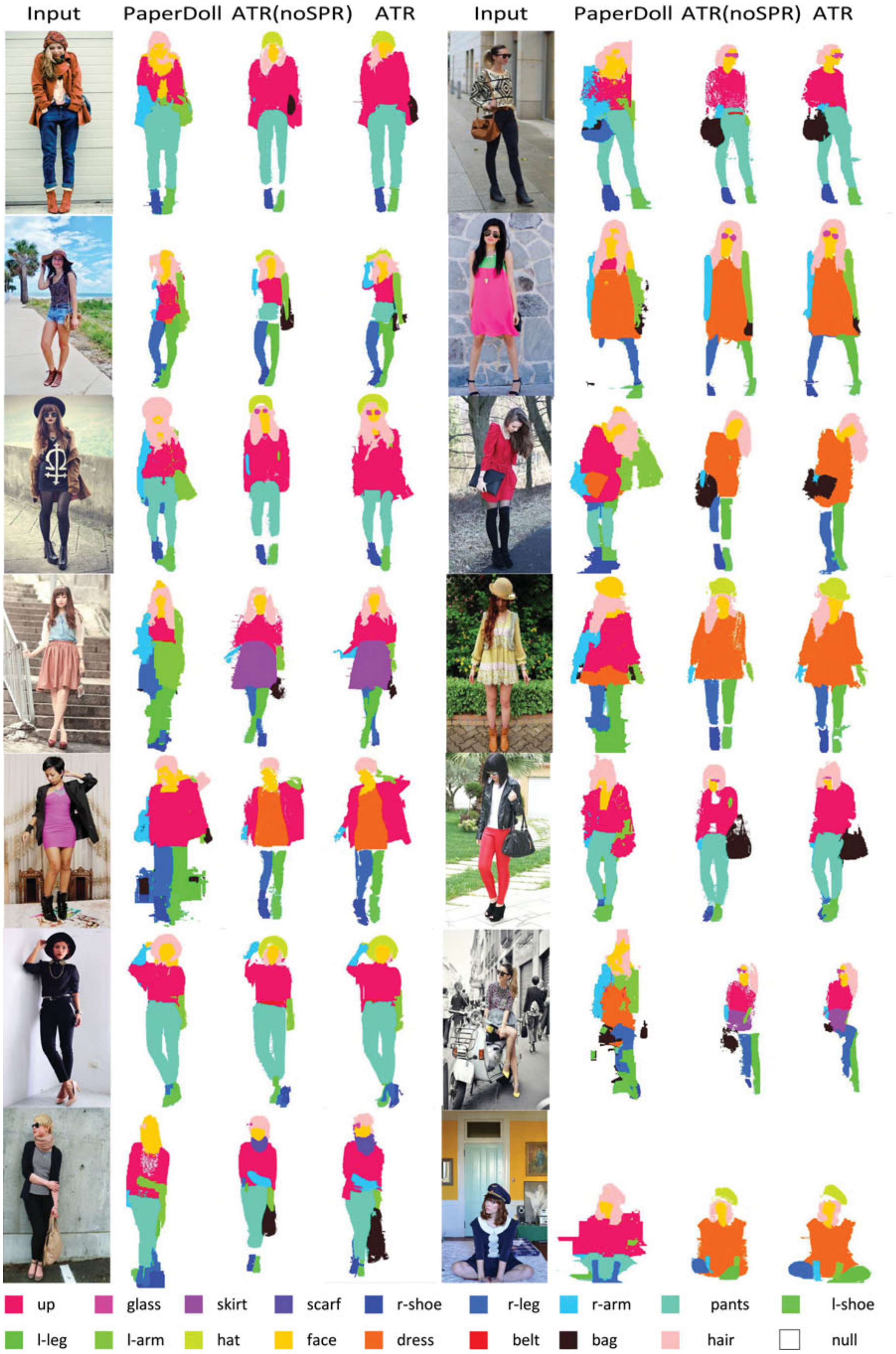


Fig. 9. Comparison of parsing results with the state-of-the-art method and our two versions. For each image, we show the parsing results by PaperDoll [28], our "ATR (noSPR)" with no super-pixel smoothing and our full method "ATR" sequentially.

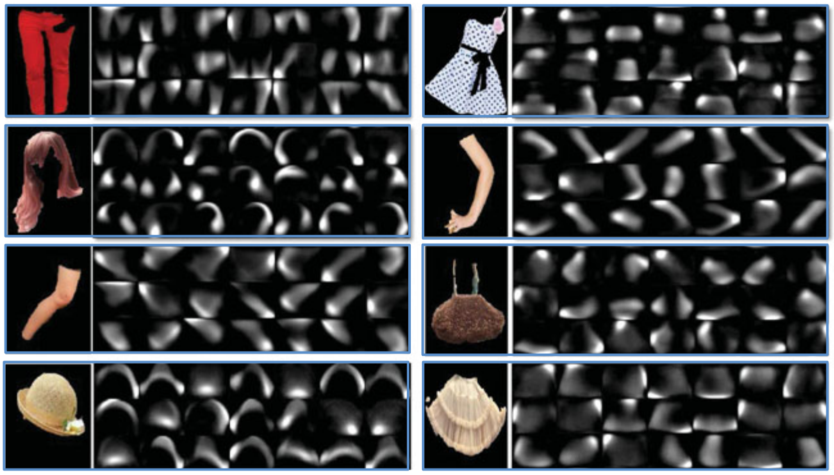


Fig. 10. Visualization of our template dictionaries of eight semantic labels, including pants, dress, hair, left-arm, right-leg, bag, hat and dress that are displayed sequentially. For each label, we display 21 learned templates by the NMF method. Brighter pixels represent the most important parts for distinguishing different label masks.

interactions of label masks and their spatial layouts. But our experiment shows that directly combining two kinds of structure outputs works do not work well for human parsing. In the further works, we will explore how to design a more effective network architecture to combine these two networks.

5 Conclusions

In this work, we formulate the human parsing task as an active template regression problem. Two separate convolutional neural networks, namely, active template network and active shape network, are designed to build the end-to-end relation between the input image and the structure outputs. The first CNN network is with max-pooling to predict the mask template coefficients, while the second CNN network is without max-pooling for position sensitiveness to predict the active shape parameters. Extensive experimental results clearly demonstrated the effectiveness of the proposed ATR framework. In the future, we plan to further explore how to adequately utilize the low-level information (e.g. edges and super-pixels). In addition, we will integrate the fine-grained attributes of each semantic label into our framework. Finally, we will build a website to provide a user interface so that any user can upload his/her own photo, and we output the parsing result within one second. Our framework can also be easily extended to improve the generic image parsing (e.g. scene parsing or human pose estimation) by utilizing the area-specific active templates.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 61328205), the Microsoft Research Asia collaboration projects, Guangdong Natural Science Foundation (no. S2013050014548), Program of Guangzhou Zhujiang Star of Science and Technology (no. 2013J2200067), Special Project on Integration of Industry, Education and Research of Guangdong (no. 2012B091000101). This work was done when Xiaodan Liang is an intern in the National University of Singapore. S. Liu is the corresponding author.

References

[1] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, “Semantic segmentation with second-order pooling,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 430–443.

[2] J. Carreira and C. Sminchisescu, “Cpmc: Automatic object segmentation using constrained parametric min-cuts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, Jul. 2012.

[3] H. Chen, Z. Xu, Z. Liu, and S. C. Zhu, “Composite templates for cloth modeling and sketching,” in *Proc. Comput. Vis. Pattern Recog.*, 2006, pp. 943–950.

[4] H. Chen, A. Gallagher, and B. Girod, “Describing clothing by semantic attributes,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 609–623.

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models—their training and application,” *Comput. Vis. Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[7] M. Dantone, J. Gall, C. Leistner, and L. V. Gool, “Human pose estimation using body parts dependent joint regressors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3041–3048.

[8] J. Dong, Q. Chen, W. Xia, Z. Huang, and S. Yan, “A deformable mixture parsing model with parselets,” in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 3408–3415.

[9] 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.

[10] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[11] B. Fulkerson, A. Vedaldi, and S. Soatto, “Class segmentation and object localization with superpixel neighborhoods,” in *Proc. Int. Conf. Comput. Vision*, 2009, pp. 670–677.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. Comput. Vision Pattern Recog.*, 2014.

[13] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, “Geodesic star convexity for interactive image segmentation,” in *Proc. Comput. Vis. Pattern Recog.*, 2010, pp. 3129–3136.

[14] Y. Jia. (2013). Caffe: An open source convolutional architecture for fast feature embedding. [Online]. Available: http://caffe.berkeleyvision.org/

[15] I. Jolliffe, “Principal component analysis,” *Encyclopedia of Statistics in Behavioral Science*. Springer, New York, 2002.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.

[17] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, pp. 788–791, 1999.

[18] L. Lin, X. Wang, W. Yang, and J.-H. Lai, "Discriminatively trained and-or graph models for object shape detection," *CoRR*, 2015, http://arxiv.org/abs/1502.00341

[19] S. Liu, J. Feng, C. Domokos, H. Xu, J. Huang, Z. Hu, and S. Yan, "Fashion parsing with weak color-category labels," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 253–265, Jan. 2014.

[20] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan, "Hi, magic closet, tell me what to wear!" in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 619–628.

[21] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set," in *Proc. Comput. Vis. Pattern Recog.*, 2012, pp. 3330–3337.

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.

[23] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

[24] P. H. O. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 82–90.

[25] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[26] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Adv. Neural Inform. Process. Syst.*, 2013, pp. 2553–2561.

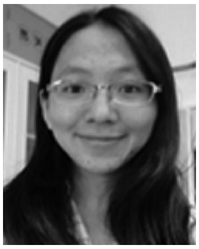
[27] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1653–1660.

[28] K. Yamaguchi, M. H. Kiapour, and T. L. Berg, "Paper doll parsing: Retrieving similar styles to parse clothing items," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 3519–3526.

[29] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, "Parsing clothing in fashion photographs," in *Proc. Comput. Vision Pattern Recog.*, 2012, pp. 3570–3577.

[30] W. Yang, L. Lin, and P. Luo, "Clothing co-parsing by joint image segmentation and labeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3182–3189.

[31] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," European Conference on Computer Vision, pp. 818–833, 2014, Doi: 10.1007/978331910590-1_53.



Xiaodan Liang is working towards the PhD degree from the School of Information Science and Technology, Sun Yat-Sen University, China. She is currently working at the National University of Singapore as a research intern. Her research interests mainly include semantic segmentation, object/action recognition, and medical image analysis.

Si Liu received the PhD degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences in 2012. She is an associate professor in the Institute of Information Engineering, Chinese Academy of Sciences. She used to be a research fellow at the Learning and Vision Group of National University of Singapore. Her research interests includes computer vision and multimedia.

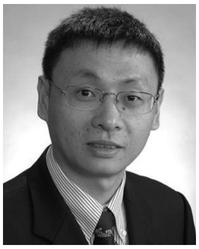
Xiaohui Shen received the MS and BS degrees from the Department of Automation at Tsinghua University, China, and the PhD degree from the Department of EECS at Northwestern University in 2013. He is currently a research scientist at Adobe Research, San Jose, CA. His research interests include image/video processing and computer vision.

Jianchao Yang (S08, M12) received the MS and Ph.D degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Urbana, in 2011. He is currently a research scientist with the Advanced Technology Laboratory, Adobe Systems Inc., San Jose, CA. His research interests include object recognition, deep learning, sparse coding, image/video enhancement, and deblurring. He is a member of the IEEE.

Luoqi Liu is currently working toward the PhD degree with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include computer vision, multimedia, and machine learning.

Jian Dong received the BSc degree from the University of Science and Technology of China in 2010, and the PhD degree from the National University of Singapore in 2014. He is now a research scientist of Amazon, Seattle. His research interests include computer vision and machine learning. He received the winner prizes of classification and segmentation tasks in PASCAL VOC'12, the winner prize of detection task in ImageNet 2014.

Liang Lin received the PhD degrees from the Beijing Institute of Technology, in 2008, and his PhD dissertation was nominated by the China National Excellent PhD Thesis Award in 2010. He is a professor with the School of Advanced Computing, Sun Yat-sen University (SYSU), China. He was a post-doctoral research fellow with the Center for Vision, Cognition, Learning, and Art of UCLA. His research focuses on new models, algorithms and systems for intelligent processing and understanding of visual data. He has published more than 60 papers in top tier academic journals and conferences, and has served as an associate editor for journal *Neurocomputing* and The *Visual Computer*. He was supported by several promotive programs or funds for his works, such as Program for New Century Excellent Talents of Ministry of Education (China) in 2012. He received the Best Paper Runners-Up Award in ACM NPAR 2010, Google Faculty Award in 2012, and Best Student Paper Award in IEEE ICME 2014.



Shuicheng Yan (M'06-SM'09) is currently an associate professor at the Department of Electrical and Computer Engineering at the National University of Singapore, and the founding lead of the Learning and Vision Research Group (http://www.lv-nus.org). His research areas include machine learning, computer vision, and multimedia, and he has authored/co-authored nearly 400 technical papers over a wide range of research topics, with Google Scholar citation $>$12,000 times. He is ISI highly-cited researcher 2014, and IAPR fellow 2014. He has been serving as an associate editor of *IEEE Transactions on Knowledge and Data Engineering*, *Computer Vision and Image Understanding,* and *IEEE Transactions on Circuits and Systems for Video Technology*. He received the Best Paper Awards from ACM MM'13 (Best Paper and Best Student Paper), ACM MM12 (Best Demo), PCM'11, ACM MM10, ICME10, and ICIMCS'09, the runner-up prize of ILSVRC'13, the winner prizes of the classification task in PASCAL VOC 2010-2012, the winner prize of the segmentation task in PASCAL VOC 2012, the honorable mention prize of the detection task in PASCAL VOC'10, 2010 TCSVT Best Associate Editor (BAE) Award, 2010 Young Faculty Research Award, 2011 Singapore Young Scientist Award, and 2012 NUS Young Researcher Award. He is a senior member of the IEEE.