

Matching-CNN Meets KNN: Quasi-Parametric Human Parsing

Si Liu^{1,2} Xiaodan Liang^{2,4} Luoqi Liu² Xiaohui Shen³ Jianchao Yang³
Changsheng Xu⁵ Liang Lin⁴ Xiaochun Cao¹ Shuicheng Yan²

¹ SKLOIS, IIE, Chinese Academy of Sciences ²National University of Singapore ³Adobe Research

⁴Sun Yat-sen University ⁵ IA, Chinese Academy of Sciences

liusi@iie.ac.cn, xdliang328@gmail.com

Abstract

Both parametric and non-parametric approaches have demonstrated encouraging performances in the human parsing task, namely segmenting a human image into several semantic regions (e.g., hat, bag, left arm, face). In this work, we aim to develop a new solution with the advantages of both methodologies, namely supervision from annotated data and the flexibility to use newly annotated (possibly uncommon) images, and present a quasi-parametric human parsing model. Under the classic K Nearest Neighbor (KNN)-based nonparametric framework, the parametric Matching Convolutional Neural Network (M-CNN) is proposed to predict the matching confidence and displacements of the best matched region in the testing image for a particular semantic region in one KNN image. Given a testing image, we first retrieve its KNN images from the annotated/manually-parsed human image corpus. Then each semantic region in each KNN image is matched with confidence to the testing image using M-CNN, and the matched regions from all KNN images are further fused, followed by a superpixel smoothing procedure to obtain the ultimate human parsing result. The M-CNN differs from the classic CNN [12] in that the tailored cross image matching filters are introduced to characterize the matching between the testing image and the semantic region of a KNN image. The cross image matching filters are defined at different convolutional layers, each aiming to capture a particular range of displacements. Comprehensive evaluations over a large dataset with 7,700 annotated human images well demonstrate the significant performance gain from the quasi-parametric model over the state-of-the-arts [29, 30], for the human parsing task.

1. Introduction

Human parsing, namely partitioning the human body into several semantic regions (e.g., hat, left/right leg, glasses

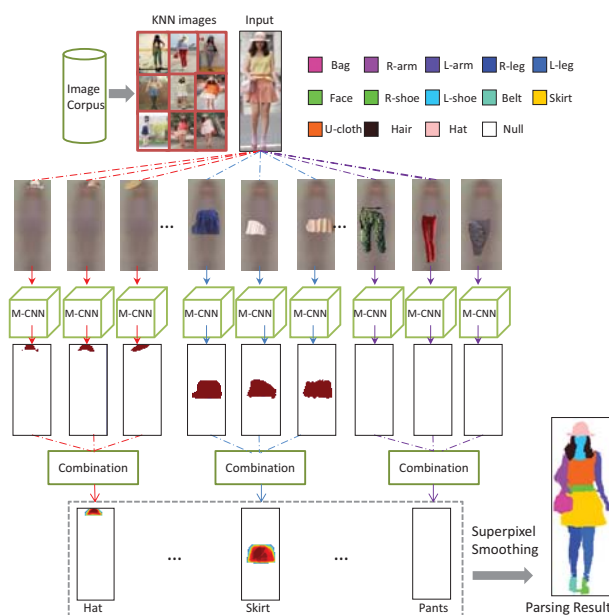


Figure 1. Illustration of our quasi-parametric human parsing framework. Given a testing image, its K Nearest Neighbour (KNN) images are retrieved from the manually-annotated image corpus. Then the input image is paired with each semantic region (e.g., hat, skirt and pants) of its KNN images and each pair is fed into the M-CNN individually. The M-CNN predicts the matching confidence and displacements between the input image pair. Then the corresponding label maps are transferred from the KNN region to the testing image. All transferred label maps from different KNN images are combined to produce a probability map for each label. The probability map is further refined by superpixel smoothing to generate the final parsing result. The color legend is shown in the top. For better viewing of all figures in this paper, please refer to the original zoomed-in color pdf file.

and upper-body clothes), has drawn much attention in recent years [29, 30, 4, 3] and serves as the basis for many high-level applications, such as clothing classification [1] and retrieval [20].

Several parametric and non-parametric human parsing methods are proposed and show very promising perfor-

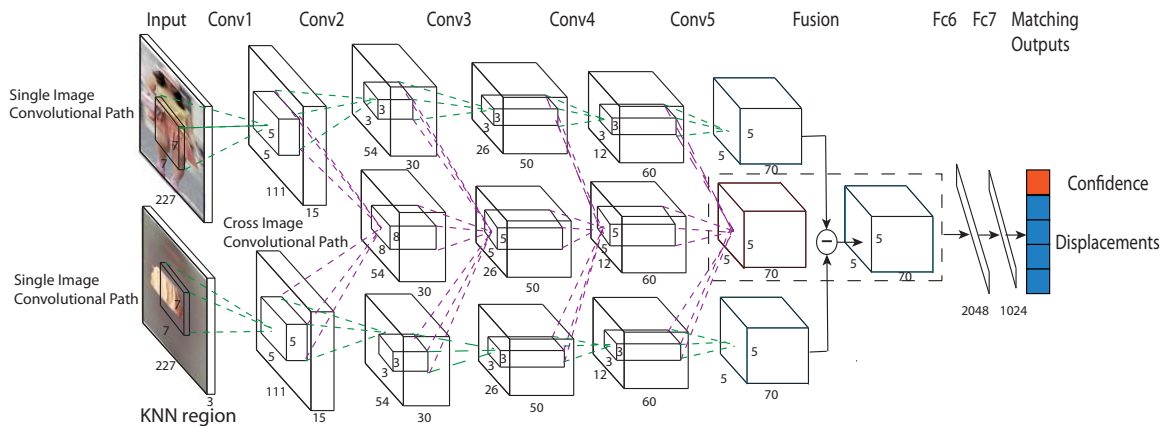


Figure 2. The architecture of the proposed Matching Convolutional Neural Network (M-CNN) with parameters shown. The input image is cropped from a human image by [6], which is paired with the semantic region (e.g., skirt). The image pair is fed into the M-CNN to estimate the matching outputs including 1-dim matching confidence (red square) and 4-dim displacements (blue squares). The M-CNN is composed of two kinds of paths. In the single image convolutional paths (top and bottom rows), the input and KNN region are independently convolved with single image filters (green dashed lines) for hierarchical feature representation. In the cross image convolutional patch (middle row), the cross image matching filters (purple dashed lines) are embedded in Conv2, Conv3, Conv4 and Conv5 layers to achieve multi-ranged matching. Each matching kernel convolves with all features maps of previous convolutional layers. The feature maps of the three paths are fused to fit the network output.

mances on the human parsing task. On one hand, parametric methods [30, 3, 19, 4] learn knowledge, such as the appearance of regions of different semantic labels and the structural relationship among different labels, from annotated images. Those methods usually rely on the manually designed structural models [4], which may not fit specific data well and thus achieve only suboptimal performance. Moreover, for another set of new training data and semantic labels, new models have to be designed/retrained, which makes those parametric models impractical because new clothing styles may come out quite often. On the other hand, the non-parametric methods can flexibly use the newly annotated images on the fly and address the issues of parametric model, which are more appealing for practical applications [17]. This kind of methods usually firstly build the pixel-level [17], superpixel-level [29] or hypothesis-level [27, 13, 14] matching between a testing image and the annotated images in a corpus, then transfer the labels from the manually annotated images to the testing image based on the matching outputs, and finally fuse the transferred labels by heuristic aggregation schemes (typically majority voting). However, the quality of matching is usually limited by the lack of explicit semantic meaning of the bottom-up superpixels or hypotheses.

The above-mentioned parametric and non-parametric human parsing methods rely on the hand designed pipelines composed of multiple sequential components, e.g., hand-crafted feature extraction, bottom-up over-segmentation, human pose estimation, manually designed complex model structure. Therefore, the possibly bad performance of each component may become the bottleneck of the performance

of the whole pipeline. For example, the human pose estimation, an important component in the above pipeline, itself is quite a challenging task. Such a sequential processing strategy usually makes the whole pipeline mostly suboptimal. Instead of a combination of multiple sequential steps, several Convolutional Neural Network (CNN) based methods are proposed for an image parsing [9, 28, 22, 5]. However, these deep models cannot be easily updated to incorporate new semantic labels.

To address these issues, we propose a quasi-parametric human parsing framework, which inherits the merits of both the parametric and non-parametric models. The proposed framework is able to take full advantage of the supervision information from annotation training data, and meanwhile is easy to extend for new added labels. The core part of the proposed framework is a specially designed Matching Convolutional Neural Network (M-CNN) to match any semantic region of a KNN image (also denoted as KNN region in this paper) to the testing image.

As shown in Fig. 1, we first apply the human detection [6] to a testing image and obtain the human centric image. Then the K Nearest Neighbours (KNN) images of the test image is retrieved from the annotated/manually-parsed image corpus. Each KNN image derives several semantic regions, which are generated by masking out the background region with the mean image of the image corpus. Three KNN regions for hat, skirt and pants are shown in Fig. 1. Then, the pair of the test image and each of the KNN regions is fed into the proposed M-CNN to estimate their matching confidence and displacements. The matching confidence measures how the KNN region matches the

input image while the displacements describe the coordinate translations between the KNN region and the matched region in the testing image. The matching confidences are then averaged over all KNN regions and thresholded to predict whether a specific label is present. For the labels predicted to be present, such as hat and skirt in Fig. 1, the corresponding label maps can be transferred from the KNN regions to the testing image based on the estimated displacements. For the labels predicted as invisible, such as pants in Fig. 1, no transferred label map is generated. Then, all matched regions of a specific label are combined to produce a probability map. Finally, the probability maps for all labels are refined by a superpixel smoothing step to get the final parsing result.

Reliable matching between an input image and a KNN region is challenging, because the matching needs to handle the large spatial variance of semantic regions. For example, the bags can be placed on the left, right or in front of the human body. The proposed M-CNN is able to achieve accurate multi-ranged matching. As shown in Fig. 2, M-CNN contains three paths, i.e., two single image convolutional paths and a cross image convolutional path. The *single image convolutional path* receives the input image or a particular KNN region, and produces its discriminative hierarchical feature representations layer by layer. The *cross image convolutional path* embeds cross image filters into every convolutional layer to characterize the multi-ranged matching. The cross image filters are applied to all feature maps in previous convolutional layers, including the single image feature maps and cross image feature maps. Because the scale of receptive fields of the feature maps increase when tracing up the M-CNN, the cross image matching filters capture the displacements from the near-range to the far-range. Therefore the feature maps from the cross image convolutional path can well represent the displacements. Because the feature maps generated by the two single image convolutional paths are excellent feature representations, their absolute difference maps are calculated as another measurement of the displacements. The difference maps are combined with the cross image feature maps and then link to the subsequent fully connected layers. Finally, the matching confidence and displacements are regressed. Since the M-CNN targets at matching an input image and any KNN region of any semantic label, it can work even if new semantic labels are included. Instead of training a M-CNN for each label, we train a unified M-CNN for all KNN regions of all labels.

Comprehensive evaluations over a large dataset with 7,700 annotated human images well demonstrate the effectiveness of our quasi-parametric framework. The major contributions are summarized as follows:

- We build a novel deep quasi-parametric human parsing framework. It can learn from annotated data and also flexibly use newly annotated (possibly uncommon) im-

ages.

- We propose a Matching Convolutional Neural Network (M-CNN) to match a semantic region of a KNN image to a testing image. The novel cross image filters are embedded into different convolutional layers, each aiming to capture a particular range of displacements.
- We integrate all the step-by-step components (over-segmentation, pose estimation, feature extraction, label modeling, etc.) in traditional pipelines into one unified deep CNN framework.

2. Related Work

In this section, we sequentially review the parametric human parsing methods, non-parametric methods and deep learning based methods.

For parametric human parsing, Yamaguchi et al. [30] proposed to boost the human parsing with pixel-level classification by relying on human pose estimation. To capture more complex contextual information, Dong et al. [4] designed an And-or Graph structure to model the correlations of a group of parselets, and their extension work [3] unified the human parsing and pose estimation in one framework. The image co-segmentation and region co-labeling for human parsing were also used to capture the correlations between different human images [31]. In addition, Liu et al. [19] utilized user-generated category tags to build a human parser. For general image parsing, Tighe et al. [26] proposed a segmentation by detection approach. Firstly, the bounding boxes of the objects are estimated by exemplar SVM [26], based on which the segmentation masks are transferred from the image corpus to the input image. In general, the power of existing parametric methods is largely limited by the suboptimal performance of many hand designed intermediate components, such as pose estimation, and also cannot be easily extended to parse new labels.

In non-parametric human parsing, pixels [17], superpixels [25, 7, 14] and object proposals [27, 13, 14, 23] were used to facilitate non-parametric image parsing. Specifically, the model of Yamaguchi et al. [29] transferred parsing masks from retrieved examples to the query image. Their label transferring is based on superpixels, which are generated by over-segmentation based on the low level appearance cues and therefore lack semantic meaning. Liu et al. [17] used SIFT Flows to build the pixel-pixel correspondence and the dense deformation field between images. However, the optimization problem for finding the SIFT flow is rather complex and expensive to solve. Recently, Long et al. [21] proved the better performance of convolutional activation features over traditional features, such as SIFT, for tasks requiring correspondence. Overall, the non-parametric methods are limited by the inaccurate

matching, which results in the noises/outliers during the label transferring.

Our quasi-parametric model integrates the advantages of both parametric models and non-parametric models by the proposed M-CNN. There exist some works on semantic segmentation with CNN architectures. Girshick et al. [6] and its extension work [9] proposed to classify the candidate regions by CNN for semantic segmentation. Wang et al. [28] presented a joint task learning framework, in which the object localization task and the object segmentation task are tackled collaboratively via CNN. Farabet et al. [5] trained a multi-scale CNN from raw pixels to extract deep features for assigning the label to each pixel. The recurrent CNN [22] was proposed to speed up scene parsing and achieved the state-of-the-art performance. Our M-CNN inherits the merit of existing CNN parsing models in our single image convolutional path. It differs from all existing CNN based parsing models in that we handle a pair of images instead of a single image and we incorporate cross image filters to specifically characterize multi-ranged matching. Last but not least, M-CNN can effortlessly handle new semantic labels.

3. Quasi-parametric Human Parsing

For each human image, we first retrieve its KNN images from the annotated image corpus (Sec. 3.1). Then, M-CNN predicts the matching confidence and displacements between the input image and a semantic region from one KNN image, based on which a label map is generated (Sec. 3.2). Finally, all label maps are fed into a post processing procedure to produce the parsing result (Sec. 3.3).

3.1. K-Nearest-Neighbor Retrieval

For each of the input images, we use the human detection algorithm [6] to detect the human body. The resulting human centric image I is then rescaled to $227 \times 227 \times 3$. We then extract a global 4,096-dimensional feature from the penultimate fully-connected layer in the pre-trained CNN model trained on ILSVRC 2012 classification dataset based on the Krizhevsky architecture [12]. Its KNN images $G = \{g_1, g_2, \dots, g_K\}$ are retrieved from the image corpus based on the deep features.

3.2. Matching Convolutional Neural Network

Input, Output and Loss Function: Given a label $l \in \{1, \dots, L\}$, where L is the total number of labels, the input image I and each KNN regions g_{kl} from KNN image g_k form a pair and are fed into the M-CNN. Note that g_k is from the image corpus and thus its label map is known. g_{kl} is generated by keeping the regions of the label l in g_k and masking out other regions by the mean image calculated from the image corpus. If g_k does not contain label l , g_{kl} is exactly the mean image.

Given an image pair $\{I, g_{kl}\}$, M-CNN learns a regressor to estimate the matching confidence and displacements between them. The 1-dim matching confidence c_{kl} indicates how well g_{kl} can be matched to I . In the training phase, it is a binary index indicating whether the KNN region is matched to the input image. g_{kl} is considered to be matched with I if and only if I contains the label l . In the testing phase, higher value of c_{kl} indicates that better matching is found. We denote the coordinates of the upper left and lower right corner of the KNN region in g_{kl} as $U_g = [g_{kl}^{x_1}, g_{kl}^{y_1}]$ and $W_g = [g_{kl}^{x_2}, g_{kl}^{y_2}]$. Similarly, the coordinates of the matched region in I are denoted as $U_I = [I^{x_1}, I^{y_1}]$ and $W_I = [I^{x_2}, I^{y_2}]$. The coordinates are normalized by the height and width of the image into the range $[0, 1]$. The 4-dim displacements t_{kl} represents the differences between U_g and U_I , W_g and W_I . Instead of using a classification loss [12], we train the M-CNN by minimizing ℓ_2 distance between the ground truth $[c_{kl}, t_{kl}]$ and the prediction $[\tilde{c}_{kl}, \tilde{t}_{kl}]$. The corresponding ℓ_2 loss for each image I is defined as

$$J = \frac{1}{L \times K} \sum_{k=1}^K \sum_{l=1}^L \|c_{kl} - \tilde{c}_{kl}\|^2 + \frac{1}{L \times K} \sum_{k=1}^K \sum_{l \in \varphi(g_k) \cap \varphi(I) \cap \{1, \dots, L\}} \|t_{kl} - \tilde{t}_{kl}\|^2 \quad (1)$$

where $\varphi(\cdot)$ is the label set contained in the specific image. The first term in Eq. (1) is the loss for matching confidence while the second term corresponds to the loss for displacements. We penalize the displacements loss when both the KNN image g_k and the input image I contain the label l . Then the losses of all training pairs are summed and the parameters are learned by back propagation.



Figure 3. Illustration of the multi-range matching. For a specific label, e.g., bag, the matchings between the input image and KNN regions may be near/middle/far-ranged. To facilitate the display, we only keep the bags regions unchanged and gray other pixels.

Architecture: Since the KNN images are retrieved based on the global appearance similarity, the KNN region of each label may locate quite differently in images. For example, in Fig. 3, the bags can be placed on the left side or right side or in front of the human body. M-CNN is designed to estimate the multi-ranged matching by embedding the cross image matching filters in different convolutional layers. As shown in Fig. 2, M-CNN contains two kinds of paths, i.e., two single image convolutional paths and one

cross image convolutional path the outputs of these three paths are further fused to estimate the matching confidence and displacements. The single image path aims for hierarchical feature representation while the cross image convolutional path estimates the displacements between the input pair.

Single Image Convolutional Path: We have two instantiations of the single image path in the top and bottom row of Fig. 2, each of which separately processes I or g_{kl} . They share the same architecture and extract the hierarchical feature representations of I or g_{kl} . The outputs are their respective feature maps in “conv5”. In this path, the single image filters of the next convolutional layers are connected to those feature maps in the previous layer, shown as the green dashed line in Fig. 2. The ReLU non-linearity is applied to the output of every convolutional layer. The sizes of feature maps are gradually reduced by using the stride of 2 for all the convolutional layers. The most important difference between M-CNN and the infrastructure in [12] is that M-CNN removes the pooling layer. Although pooling is useful for enhancing translation invariance for object recognition, it loses precise spatial information that is necessary for accurately predicting the locations of the labels [10]. The details about the network parameters, such as image/feature map sizes, kernel size/numbers are shown in Fig. 2. The powerful representation capability of the single image convolutional path lays the foundation for the accurate estimation of the matching confidence and displacements.

Cross Image Convolutional Path: The cross image convolutional path lies in the middle row of Fig. 2. It outputs the cross image feature maps in “conv5” layer. The m -th cross image feature map in the j -th layer $x_{j,m}^C$ is generated by convolving the corresponding matching filter (including three components $f_{j,p,m}^I$, $f_{j,q,m}^R$ and $f_{j,t,m}^C$) with both single image ($x_{j-1,p}^I$, $x_{j-1,q}^R$) and cross image feature maps ($x_{j-1,t}^C$) in the $j-1$ -th layer. The $f_{j,p,m}^I$ component links the p -th (out of all P) input image feature map $x_{j-1,p}^I$ in the $j-1$ -th layer to $x_{j,m}^C$. Analogously, the $f_{j,q,m}^R$ component links the q -th (out of all Q) KNN region feature map $x_{j-1,q}^R$ to $x_{j,m}^C$. Moreover, the $f_{j,t,m}^C$ component links the t -th (out of all T) cross image feature map $x_{j-1,t}^C$ to $x_{j,m}^C$. The operation of the matching filters from one layer to the next is shown as the purple dashed line in Fig. 2. Mathematically, the cross feature map $x_{j,m}^C$ is calculated by:

$$x_{j,m}^C = \max(0, b_{j,m} + \sum_{p=1}^P f_{j,p,m}^I * x_{j-1,p}^I + \sum_{q=1}^Q f_{j,q,m}^R * x_{j-1,q}^R + \sum_{t=1}^T f_{j,t,m}^C * x_{j-1,t}^C) \quad (2)$$

where $*$ denotes convolution and $b_{j,m}$ is the bias for the m -th output map. $\max(0, \cdot)$ is the non-linear activation func-

tion, and is operated element-wisely. From Eq. (2), we can see that the cross image feature map in the next layer is calculated by considering both single and cross image feature maps in the previous layer, and thus the displacements between the input image I and KNN region g_{kl} can be effectively estimated. Note that along with the M-CNN, the receptive fields of different layers of the single image and cross image convolutional path gradually increase [32]. In this way, multi-ranged matchings can be achieved.

Finally, we fuse the feature maps from two single image paths and one cross image path. More specifically, the absolute differences of the feature maps of the input image and the KNN region (from the single image convolutional path) are first calculated and then are stacked with the output of the cross image convolutional path. Our fusion is applied on the feature maps. It is different from “Siamese” [2] architecture which calculates the absolute differences of the fully-connected representations. Experiments show the our fusion strategy outperforms “Siamese” by keeping more spatial information.

3.3. Post Processing

Given the matching confidences and displacements estimated by M-CNN, the parsing result of the input image can be calculated as follows. Firstly, the confidence of I containing the l -th label is calculated by averaging matching confidence c_{kl} for all KNN regions satisfying $l \in \varphi(g_k)$. If the confidence is greater than a threshold ξ_1 , the label l is predicted as visible in the input image, otherwise predicted as invisible. Secondly, we estimate the locations of the visible labels. More specifically, the coordinates of the matching region in the input image I is calculated based on the matching displacements t_{kl} and the ground-truth coordinates of g_{kl} . Then, we morph the associated ground-truth label mask of g_{kl} into matched region in I . In this way, we get a probability map M_l of I for each label $l \in [1, L]$. We pixel-wisely max all M_l for all the labels and get the foreground probability. The pixels with the probability larger than a threshold ξ_2 are regarded as the rough foreground, while the remaining are the rough background. The rough foreground and background are further eroded by a filter size 10 to produce the final foreground and background seeds. Based on the seeds, we can obtain the background probability by the algorithm [8]. The obtained background probability is combined with the foreground probability map M_l , $l \in [1, L]$, based on which, we can get an initial human parsing results with the pixel-wise Maximum a Posterior Probability (MAP) assignment. Finally, to respect boundaries of actual semantic labels, we further over-segment I using the entropy rate based segmentation algorithm [18] and assign the label of the superpixel by the majority of its covered pixels’ initial parsing results.

4. Experiments

4.1. Experimental Settings

Datasets: We use the dataset in [15] pixel-wisely labeled by the 18 categories defined by Daily Photos dataset [4]. The dataset contains 7,700 images (6,000 for training, 700 for validation and 1,000 for testing). We adopt four evaluation metrics, i.e., accuracy, average precision, average recall, and average F-1 scores over pixels [30].

Training Image Pairs Generation: To reduce overfitting in the model training and partially address the detection error, we enlarge the cropped human centric images region by 1 and 1.2 times. We also horizontally mirror the images. In short, each image has 4 variations and training data can be greatly augmented.

For each of 6,000 training images, 50 KNN images are retrieved from the image corpus. Each training image and each KNN region form a training pair. After unevenly sampling of the training pairs to balance different labels, we finally have 5 million pairs, which even outnumbers the that of ILSVRC2012 [12]. We shuffle the training pair in order to increase the diversity of each epoch.

Implementation Details: We implement the M-CNN under the Caffe framework [11] and train it 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. The learning rate is adjusted manually by dividing 10 when the validation error rate stops decreasing with the current learning rate. The learning rate is initialized at 0.0005. We train M-CNN for roughly 50 epochs, which takes 11 to 12 days on one NVIDIA GTX TITAN. In the training phase, we first calculate the element-wise mean and variance of the matching confidence and displacements of the whole image corpus, and element-wisely normalize the training output by the mean and variance. When testing, we project the matching confidence and displacements estimated by M-CNN to their absolute values by the mean and variance. In the post processing step, the thresholds ξ_1 and ξ_2 are set to be 0.8 and 0.5. The number of KNN regions for each input image is set as 9.

4.2. Results and Analysis

Comparison with The State-of-the-arts: We compare our M-CNN based quasi-parametric human parsing framework with two state-of-the-arts: Yamaguchi et al. [30] and PaperDoll [29]. We use their publicly available codes and train their models with the same 6,000 training images as our method for fair comparison. We do not compare with Dong et al. [4] because their code is not publicly available and their method is reported to be slower than ours.

The average results for all labels are in Table 1. The methods of Yamaguchi et al. [30] and the PaperDoll [29] trained on the same 6,000 training images and tested on

the same 1,000 images as M-CNN, and their average F_1 scores achieve 41.80% and 44.76%. Our “M-CNN” significantly outperforms these two baselines by over 21.01% for Yamaguchi et al. [30] and 18.05% for PaperDoll [29]. “M-CNN” also gives a huge boost in foreground accuracy: the two baselines achieve 55.59% for Yamaguchi et al. [30] and 62.18% for PaperDoll [29] while “M-CNN” obtains 73.98%. “M-CNN” also obtains much higher precision (64.56% vs 37.54% for [30] and 52.75% for [29]) as well as higher recall (65.17% vs 51.05% for [30] and 49.43% for [29]). This verifies the effectiveness of our M-CNN based quasi-parametric framework.

We also present the F1-scores for each label in Table 2. Generally, “M-CNN” shows much higher performance than the baselines. In terms of predicting labels for small semantic regions such as hat, belt, bag and scarf, our method achieves a large gain, e.g. 43.38% vs 11.43% [30] and 2.95% [29] for scarf, 57.87% vs 24.53% [30], 30.52% [29] for bag and 38.45% vs 14.68% [30] and 16.94% [29] for belt. It demonstrates that our quasi-parametric network can effectively capture the internal relations between the labels and robustly predict the label masks with various clothing styles and poses.

Ablation of Our Networks: We also extensively explore different CNN architectures to demonstrate the effectiveness of each component in M-CNN more transparently. The architecture of M-CNN is shown in Fig. 2 and other variants are constructed by gradually adding/eliminating the cross image filters in different layers. M-CNN contains 4 cross image matching filters from layers *conv2* to *conv5*. “M-CNN (cross 5,4,3,2,1)” is obtained by adding $11 \times 11 \times 6$ cross image filters in the “conv1” layer to the “M-CNN”. The added matching filters are applied on the stacked image composed of the RGB channels of input image and KNN region. In addition, we continue to remove the cross image matching filters layer by layer, producing “M-CNN (cross 5,4,3)”, “M-CNN (cross 5,4)”, “M-CNN (cross 5)” and “M-CNN(w/o cross)”. Note that no matching filters are used in the “M-CNN(w/o cross)” architecture, where “w/o” stands for without. For fair comparison, we keep the number of feature maps of each convolutional layer unchanged for different M-CNN variations. Therefore, the number of removed cross image filters is evenly added to the corresponding two single image layers. For example, the number of feature maps for both single and cross image convolutional paths are all 30 in “conv2”. After we remove the cross matching filters to derive “M-CNN (cross 5,4,3)”, “M-CNN (cross 5,4)” and “M-CNN (cross 5)”, the number of feature maps in the single image convolutional path is set as 45. In addition, we compare with a classic CNN based verification architecture named “Siamese” [2] which is a composite structure of two identical sub-networks. The outputs of the two sub-networks are fully-connected linear

Table 1. Comparison of parsing performances with several architectural variants of our model (cross image matching filters embedded into different convolutional layers, with and without superpixel smoothing) and two state-of-the-arts.

Method	Accuracy	F.g. accuracy	Avg. precision	Avg. recall	Avg. F_1 score
Yamaguchi et al. [30]	84.38	55.59	37.54	51.05	41.80
PaperDoll [29]	88.96	62.18	52.75	49.43	44.76
Siamese [2]	85.24	56.42	50.27	48.88	47.08
M-CNN (w/o cross)	88.30	69.84	58.63	59.52	56.99
M-CNN (cross 5)	88.62	69.88	60.89	60.47	58.07
M-CNN (cross 5,4)	89.41	72.44	58.93	63.16	60.03
M-CNN (cross 5,4,3)	88.97	70.84	60.27	62.23	60.36
M-CNN	89.57	73.98	64.56	65.17	62.81
M-CNN (cross 5,4,3,2,1)	89.42	71.86	63.13	63.49	61.53
M-CNN(w/o ss)	87.08	71.73	55.88	65.32	59.39

Table 2. F_1 scores of foreground semantic labels. Comparison of F_1 -scores with several architectural variants of our model and two state-of-the-art methods.

Method	Hat	Hair	S-gls	U-cloth	Skirt	Pants	Dress	Belt	L-shoe	R-shoe	Face	L-leg	R-leg	L-arm	R-arm	Bag	Scarf
Yamaguchi et al. [30]	8.44	59.96	12.09	56.07	17.57	55.42	40.94	14.68	38.24	38.33	72.1	58.52	57.03	45.33	46.65	24.53	11.43
PaperDoll [29]	1.72	63.58	0.23	71.87	40.2	69.35	59.49	16.94	45.79	44.47	61.63	52.19	55.6	45.23	46.75	30.52	2.95
Siamese [2]	74.02	44.70	11.78	65.69	74.69	61.52	70.30	1.85	37.12	34.16	39.36	50.24	51.65	32.54	30.59	24.23	39.37
M-CNN (w/o cross)	77.67	63.11	29.68	72.26	73.66	70.92	78.79	25.51	43.25	44.53	67.99	60.90	64.90	47.35	39.78	41.14	40.06
M-CNN (cross 5)	67.61	65.30	38.51	69.06	73.95	69.33	81.88	28.41	46.78	41.01	70.82	59.49	66.28	52.61	39.73	40.89	46.01
M-CNN (cross 5,4)	76.42	65.91	46.97	71.51	74.29	68.43	82.16	41.31	41.83	43.03	73.27	59.35	62.18	52.63	54.04	46.25	40.58
M-CNN (cross 5,4,3)	79.38	67.64	34.50	70.72	76.57	69.17	83.81	25.69	54.82	44.61	75.22	63.44	67.74	54.94	48.59	40.10	46.40
M-CNN	80.77	65.31	35.55	72.58	77.86	70.71	81.44	38.45	53.87	48.57	72.78	63.25	68.24	57.40	51.12	57.87	43.38
M-CNN (cross 5,4,3,2,1)	77.34	65.56	45.19	70.64	78.64	69.95	82.72	46.52	45.72	44.45	71.27	61.59	63.49	53.23	50.72	55.96	45.48
M-CNN(w/o ss)	72.26	61.32	49.89	68.74	74.80	66.41	78.24	40.13	47.86	43.86	67.66	51.61	59.37	50.34	44.38	50.78	46.04

layer responses, whose absolute differences are calculated as features to fit the final matching confidence and displacements. Finally, we compare with “M-CNN w/o ss” which is the same with M-CNN except that the superpixel smoothing processing step is skipped.

The average scores over all labels in Table 1 offer following observations. Incrementally adding cross image matching filters into more convolutional layers produces 4 variations of “M-CNN”, including “M-CNN (w/o cross)”, “M-CNN (cross 5)”, “M-CNN (cross 5,4)”, “M-CNN (cross 5,4,3)” and “M-CNN”. Their F_1 scores increase from 56.99%, 58.07%, 60.36% to 62.81%. The highest F_1 score is 62.81% which is reached by “M-CNN”. The gradually improving performance validates that inserting more cross image matching into multiple convolutional layers can help achieve better matching. Adding a cross image matching kernel in the “conv1” layer drops the F_1 score from 62.81% to 61.53%. The reason for the relatively lower result is that the receptive field corresponding to the first cross image matching kernel is small and only involves certain part of a semantic label. However the targets of this work, i.e., the matching confidence and displacements, are defined on the semantic label level and thus are beyond the receptive field of the cross image matching kernel inserted in the “conv1” layer. When the M-CNN grows deeper, the receptive fields become much larger, with a higher probab-

ity to cover the semantic label, which can facilitate estimating the semantic label-level displacements. In addition, “M-CNN (w/o cross)” performs better than “Siamese” because our ultimate task is to estimate displacements. “M-CNN (w/o cross)” calculates the difference between two “conv5” layer feature maps while “Siamese” calculates the differences of two fully connected features, where spatial structures in the 2-dim images are partially lost. The lower F_1 score of “M-CNN w/o ss” compared with “M-CNN” proves that the adopted superpixel smoothing technique can better preserve the boundary information although it is a simple and fast voting of pixels’ labels. The superior performance of “M-CNN w/o ss” than the state-of-the-arts [30, 29] shows that our M-CNN has the capability of directly predicting more reliable label masks even without the post processing step.

Sensitivity to the Number of K : In Fig. 5, we report the performance of our human parsing method with respect to different numbers of KNN images. We find that M-CNN reaches the highest F_1 score 63.58% when 9 KNN regions are considered. When only 1 KNN region is considered, the performance is still quite competitive (56.92%).

Qualitative Parsing Results Comparison: Fig. 4 shows the comparison between M-CNN and PaperDoll [29]. The results demonstrate that our method can successfully predict the label maps with small regions, which

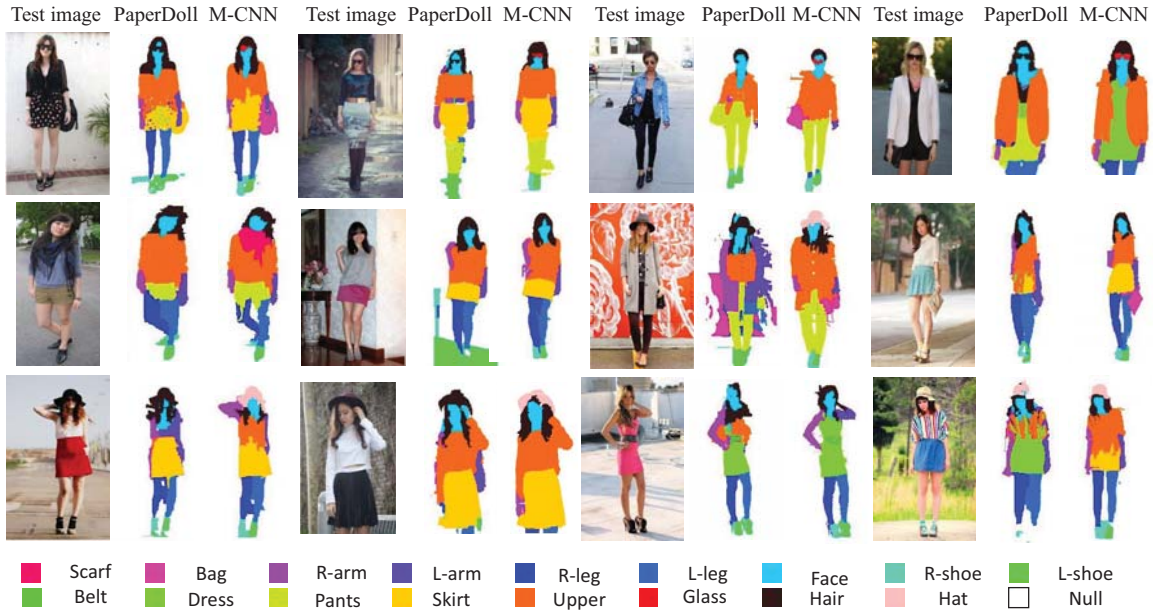


Figure 4. Comparison of our parsing results with the PaperDoll Method. For each image, we show the testing image, parsing results by PaperDoll [29], our “M-CNN” sequentially.

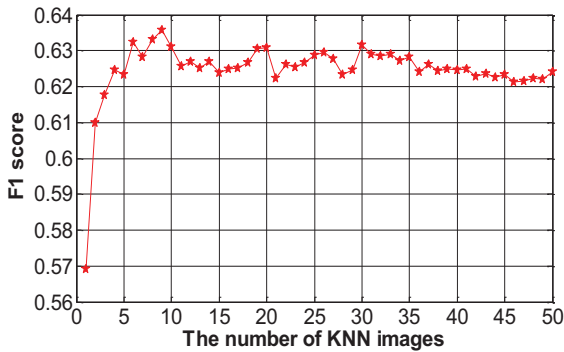


Figure 5. F_1 scores for different numbers of KNN images.

can be attributed to the reliable label transferring from the KNN regions. For example, the bag, scarf and hat in three images in the first column are successfully located by M-CNN but totally missed by PaperDoll. Another example is that M-CNN successfully finds the small sunglasses in the first row, which are missed by PaperDoll. In addition, we can observe that the results from M-CNN are robust to pose variations. As shown in the bottom row, M-CNN can accurately estimate the locations of left and right arms, while PaperDoll cannot. The superior performance is because M-CNN is a model while PaperDoll relies on a separate pose estimation preprocessing. Another observation is that our segmented regions are more complete while the PaperDoll regions are fragmented, such as the lower right result. It is because PaperDoll transfers labels based on oversegments which lack explicit semantic meaning.

5. Conclusion and Future Work

In this work, we tackle the human parsing by a new quasi-parametric model. Our unified quasi-parametric framework inherits the merits of both parametric and non-parametric parsing methodologies. To characterize the multi-ranged matching, we propose a Matching Convolutional Neural Network, which contains two single convolutional paths for better feature representation and a cross image convolutional path where cross image matching filters are embedded into the convolutional layers. Extensive experimental results clearly demonstrate significant performance gain from the quasi-parametric model over the state-of-the-arts. In the future, we will extend the framework to other exemplar-based tasks, such as face parsing. Moreover, we plan to use other more power network structure, e.g. GoogLeNet [24].

Acknowledgement

This work is supported by National Natural Science Foundation of China (No.61422213, 61332012, 61328205), and 100 Talents Programme of The Chinese Academy of Sciences. It is also supported by gift funds from Adobe, Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA06010701, the Hi-Tech Research and Development Program of China (no.2013AA013801), Guangdong Natural Science Foundation (no.S2013050014548), and Program of Guangzhou Zhujiang Star of Science and Technology (no.2013J2200067).

References

- [1] H. Chen, A. Gallagher, and B. Girod. Describing clothing by semantic attributes. In *ECCV*, 2012.
- [2] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.
- [3] J. Dong, Q. Chen, X. Shen, J. Yang, and S. Yan. Towards unified human parsing and pose estimation. In *CVPR*, 2014.
- [4] J. Dong, Q. Chen, W. Xia, Z. Huang, and S. Yan. A deformable mixture parsing model with parselets. In *ICCV*, 2013.
- [5] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2013.
- [6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013.
- [7] S. Gould, J. Zhao, X. He, and Y. Zhang. Superpixel graph label transfer with learned distance metric. In *ECCV*, 2014.
- [8] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, 2010.
- [9] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.
- [10] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler. Learning human pose estimation features with convolutional networks. *arXiv preprint arXiv:1312.7302*, 2013.
- [11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [13] D. Kuettel and V. Ferrari. Figure-ground segmentation by transferring window masks. In *CVPR*, 2012.
- [14] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. In *ECCV*, 2012.
- [15] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, L. Lin, and S. Yan. Deep human parsing with active template regression. 2015.
- [16] M. Lin, Q. Cheng, and S. Yan. Network in network. In *ICLR*, 2014.
- [17] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *TPAMI*, 2011.
- [18] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *CVPR*, 2011.
- [19] S. Liu, J. Feng, C. Domokos, H. Xu, J. Huang, Z. Hu, and S. Yan. Fashion parsing with weak color-category labels. *TMM*, 2014.
- [20] 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 *CVPR*, 2012.
- [21] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence. In *NIPS*, 2014.
- [22] P. H. O. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014.
- [23] J. A. R. Serrano and D. Larlus. Predicting an object location using a global image representation. In *ICCV*, 2013.
- [24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.
- [25] J. Tighe and S. Lazebnik. Superparsing: scalable non-parametric image parsing with superpixels. In *ECCV*, 2010.
- [26] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013.
- [27] F. Tung and J. J. Little. Collageparsing: Nonparametric scene parsing by adaptive overlapping windows. In *ECCV*, 2014.
- [28] X. Wang, L. Zhang, L. Lin, Z. Liang, and W. Zuo. Deep joint task learning for generic object extraction. In *NIPS*, 2014.
- [29] K. Yamaguchi, M. H. Kiapour, and T. L. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *ICCV*, 2013.
- [30] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012.
- [31] W. Yang, P. Luo, and L. Lin. Clothing co-parsing by joint image segmentation and labeling. In *CVPR*, 2014.
- [32] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *arXiv preprint arXiv:1311.2901*, 2013.