JMNet: A joint matting network for automatic human matting

Xian Wu¹, Xiao-Nan Fang¹, Tao Chen², and Fang-Lue Zhang³

© The Author(s) 2020.

Abstract We propose a novel end-to-end deep learning framework, the Joint Matting Network (JMNet), to automatically generate alpha mattes for human images. We utilize the intrinsic structures of the human body as seen in images by introducing a pose estimation module, which can provide both global structural guidance and a local attention focus for the matting task. Our network model includes a pose network, a trimap network, a matting network, and a shared encoder to extract features for the above three networks. We also append a trimap refinement module and utilize gradient loss to provide a sharper alpha matte. Extensive experiments have shown that our method outperforms state-of-the-art human matting techniques; the shared encoder leads to better performance and lower memory costs. Our model can process real images downloaded from the Internet for use in composition applications.

Keywords alpha matting; human images; deep learning; pose estimation

1 Introduction

Various graphics applications in the digital media sector, such as mixed reality, film production, and photographic composition, need accurate extraction of human regions from images. Some recent human segmentation methods [1, 2] focus on producing coarse human silhouettes. However, for the aforementioned applications, coarse segmentation without accurate alpha estimation for translucent human regions, such as clothing and hair, is insufficient. Therefore, automatic human alpha matting has attracted much attention in the computer vision community.

Alpha matting takes an image $I$ as input and assumes that it is a linear blend of a foreground image $F$ and a background image $B$:

$$I = \alpha F + (1 - \alpha)B$$ (1)

where $\alpha$ is the per pixel alpha matte with values in the range $[0, 1]$. For a color image, this leads to 7 unknown variables at each pixel but only 3 known values in $I$, making this problem extremely ill-posed. In previous methods, user interaction is used to guide the matting algorithms [3, 4], using, e.g., trimaps or scribbles. However, these interactions need a certain level of professional knowledge and can be time-consuming to get satisfactory results.

In the human matting task, because of the latent intrinsic structure of the human body, a trimap can be intuitively generated by eroding the human segmentation map, as a guide for matte estimation. But this trivial solution rarely produces satisfactory results [5]. One reason is that semantic segmentation aims to coarsely separate humans from the background; it often ignores fine details in the segmentation results. Simple fixed-width boundary erosion of the segmentation map cannot be used universally for different semantic parts. For example, hair needs a wider boundary than the limbs. Chen et al. [6] propose an end-to-end deep learning framework for human matting, which combines a segmentation network with a matting network using a fusion module. Their method can automatically generate alpha mattes, but may miss some body parts or details when the background is complicated.

We propose a novel end-to-end deep learning framework, the Joint Matting Network (JMNet), to improve the accuracy of human matting for
more complicated images. We utilize the semantic structure of the human image regions to assist the neural network’s learning process. We combine the matting network with a pose estimation module, which produces heatmaps of body keypoints representing intrinsic human structures. The heatmaps provide global topology for the trimap generation process and guide the matting network to pay adaptive attention to different body parts. JMNet contains three subnetworks, a pose network, a trimap network, and a matting network. These subnetworks share features extracted from the image by a backbone encoder. The outputs of the earlier subnetworks are also fed into the later subnetworks as guiding inputs. A trimap refinement component is used for better trimap boundary generation. To get a finer and sharper alpha matte, we apply gradient loss and skip connections. Figure 1 shows an alpha matte predicted by JMNet.

Extensive experiments have shown that our method can produce high quality alpha mattes for high-resolution human images. Our experiments used the DIM dataset [7] and a newly collected large-scale human image dataset with carefully annotated matte values. They show that our method outperforms state-of-the-art human matting approaches on several quantitative metrics. We have also tested our model on some Internet images and produced impressive alpha matte results for a composition task, demonstrating the generalizability of our framework.

2 Related work

2.1 Image matting

Most image matting techniques can be classified as being sampling-based, affinity-based, or learning-based. Sampling-based methods [8–11] gather several color samples from the background and the foreground marked by the trimap. These samples help build a foreground and background model to estimate the alpha value in the transition area. Affinity-based methods propagate alpha values of known foreground and background pixels to the unknown regions according to affinity scores based on, e.g., spatial proximity and color similarity, which can lead to high computational and memory costs.

Recently, with the rapid development of the deep learning techniques, learning-based image matting methods have been introduced, achieving impressive results. Cho et al. [12] use a deep convolutional network to refine alpha matte values produced by closed-form matting [3] and KNN matting [4]. Xu et al. [7] present an end-to-end deep learning network trained by alpha and composition loss functions. They also provide a large natural image dataset with precise matte annotations, the first large-scale image matting dataset available for training neural networks. AlphaGAN [13] applies generative adversarial networks (GANs) to the image matting task for more realistic and sharper results. Tang et al. [14] propose a hybrid sampling-based and learning-based matting method, which estimates the background and foreground color values for unknown regions as guidance in predicting alpha matte values.

All the above matting techniques rely on user interaction, generally in the form of providing trimaps labeling background, foreground, and uncertain regions. Drawing a trimap with per-pixel annotation is not easy for novice users. Shen et al. [5] propose an automatic portrait matting method, utilizing a fully-convolutional network [15] to predict the trimap for the input image. The network also learns to predict the parameters of the Laplacian matrix. The alpha matte is eventually computed by the formula provided by Levin et al. [3]. Chen et al. [6] combine PSPNet [16] with the matting network proposed by Xu et al. [7] and simultaneously train them with a fusion module. Zhang et al. [17] present a novel matting method taking a single RGB image as input.

Fig. 1 Results produced by our method: pose heatmaps, trimap, and alpha matte. The heatmaps represent keypoints of the human body, including nose, neck, right shoulder (RShoulder), right ear (REar), left ear (LEar), right eye (REye), left eye (LEye), etc.
The framework contains two branches for foreground and background classification respectively and applies a fusion branch to predict alpha matte values from the two classification results. Nevertheless, all of these automatic image matting techniques ignore the intrinsic structures of the human body, possibly leading to erroneous decisions when facing complex scenes.

2.2 Semantic segmentation and pose estimation

Many successful semantic segmentation techniques are based on deep convolutional neural networks. FCN [15] was the first fully CNN-based method for semantic segmentation. U-Net [18] uses skip connections to build a bridge between the encoder and decoder, allowing the network to be aware of low-level details. DeepLab [19] applies atrous convolution in the segmentation network for dense feature extraction and field-of-view enlargement. PSPNet [16] presents a pyramid pooling module to take advantage of multi-scale features containing both global structure and local texture.

Human segmentation methods are usually built upon general segmentation networks like the aforementioned ones. PFCN+ [2] calculates an average mask from the training set and aligns it to each portrait image by detecting facial landmarks. The mask is then fed into the segmentation network as prior information along with the input image. Chen et al. [1] use a boundary attention module to refine boundary areas in portrait segmentation; boundary attention is not used in general segmentation frameworks but is important for portrait images.

DCNN-based pose estimation methods either use regression to determine the positions of keypoints [20, 21], or predict heatmaps generated by the keypoints [22–24]. Some papers use pose heatmaps to guide human parsing [25, 26] and human image completion [27]. Our method uses heatmaps as a pose representation, because heatmaps can readily encode body structures and provide convenient inputs to the other neural networks.

3 Approach

3.1 Overview

Our deep learning model, the Joint Matting Network (JMNet), addresses automatic human image matting. Its input is an RGB image, its output is an alpha matte mask giving the transparency value for each foreground pixel. Unlike previous image matting techniques, our approach does not need user interaction, but generates a trimap automatically during the prediction process. Given an input image, our framework produces pose heatmaps, a trimap, and the alpha matte in turn. Earlier outputs are fed into the next subnetwork as guidance. This end-to-end deep learning framework allows our method to predict a human image alpha matte quickly and accurately.

3.2 Architecture

JMNet contains an encoder and three subnetworks: a pose network, a trimap network, and a matting network. The encoder first extracts multi-scale features from the input image. These features are then fed into the pose network to produce the heatmaps. The heatmaps and the features are fed into the trimap network to generate the trimap. Finally, the trimap, the heatmaps, and the features are fed into the matting network to predict the alpha matte. Figure 2 provides an overview of JMNet.

3.2.1 Feature encoder

We use a pre-trained ResNet-50 [28] to extract features since it is a highly successful image classifier and has proved to be a reliable feature extractor in other computer vision tasks [16, 29, 30]. Then we apply the atrous spatial pyramid pooling (ASPP) module from DeepLab-v2 [19]. ASPP can exploit multi-scale image representations from the extracted features, where features with large dilations contribute to the global body structure while features with small dilations are helpful for local parts.

3.2.2 Pose network

The pose subnetwork consists of several convolutional layers. The input to the pose network comprises the features extracted by the encoder; its output comprises $K$-dimensional heatmaps, where $K$ is the number of keypoints in the human pose. The ground-truth heatmaps consist of 2D Gaussians (with a standard deviation of 1 pixel) centered on the joint locations. The pose network is able to explore intrinsic human structures, encoding them into the heatmaps as guidance for downstream subnetworks.

3.2.3 Trimap network

The trimap network aims to produce the trimap automatically to avoid the need for user interaction.
Generation of the trimap can be regarded as a three-class semantic segmentation task, determining background, foreground, and the uncertain region. The input to the trimap network is a combination of the features and the heatmaps. The output is a three-channel map indicating the probabilities that the pixel belongs to the foreground, background, and uncertain region. The original output trimap is $1/8$ the size of the input image and simply upsampling it will lead to an over-smooth result. Therefore, we append a refinement module after the trimap network to sharpen the upsampled trimap, making it more similar to a manual annotation.

### 3.2.4 Matting network

The features, the heatmaps, and the trimap are concatenated and fed into the matting network, which predicts the alpha matte for the human image. The matting task focuses on local details while the segmentation task pays more attention to global semantics. Nevertheless, the final features extracted by the encoder generally represent high-level abstract information and may not capture many details. Inspired by U-Net [18], we use skip connections from the shallow layers of the encoder to the deep layers of the matting network, so that low-level features can be transmitted to the matte estimation process. We also adopt the fusion module proposed in SHM [6]. The final alpha matte combines the alpha values predicted by the matting network and the probability of each class predicted by the trimap network:

$$\alpha = F_s + U_s \alpha_m$$  \hspace{1cm} (2)

where $\alpha$ denotes the final alpha matte, $\alpha_m$ denotes the alpha matte predicted by the matting network, and $F_s$ and $U_s$ denote the probabilities that the pixel belongs to the foreground and the uncertain regions respectively after a softmax function.

Table 1 provides detailed architectural details for the above three subnetworks.

### 3.3 Loss functions

The loss function for optimizing the parameters of each subnetwork is defined by the difference between the predicted result and the ground truth for each subnetwork.

For the pose network, following past pose estimation methods [22–24], we use the mean squared error (MSE) loss function for heatmap generation. It can be formulated as

$$L_p = \| p - \hat{p} \|_2$$  \hspace{1cm} (3)

where $p$ denotes heatmaps produced by the pose network and $\hat{p}$ denotes ground-truth heatmaps.

For the trimap network, cross entropy loss is
Table 1 Detailed configuration of JMNet. Each convolutional layer except the last one of each subnetwork is followed by a BatchNorm layer and a ReLU layer. Concat denotes concatenation and Upsample denotes bilinear upsampling.

| Component | Name      | Input   | Kernel size | Channels |
|-----------|-----------|---------|-------------|----------|
| Pose network | Conv-1    | 3 x 3  | 256         |
|            | Conv-2    | 3 x 3  | 256         |
|            | Conv-3    | 3 x 3  | 256         |
|            | Conv-4    | 1 x 1  | 25          |
|            | Concat-1  |         |             | 281      |
|            | Conv-1    | 3 x 3  | 256         |
|            | Conv-2    | 3 x 3  | 256         |
|            | Conv-3    | 3 x 3  | 256         |
|            | Conv-4    | 1 x 1  | 29          |
| Trimap network | Upsample  |         |             | 29       |
|            | Concat-2  |         |             | 32       |
|            | Refine-1  | 3 x 3  | 32          |
|            | Refine-2  | 3 x 3  | 32          |
|            | Refine-3  | 3 x 3  | 32          |
|            | Refine-4  | 3 x 3  | 3           |
| Matting network | Res-2     |         |             | 284      |
|            | Concat-1  |         |             | 256      |
|            | Skip-1    | 3 x 3  | 128         |
|            | Conv-1    | 3 x 3  | 64          |
|            | Conv-2    |         |             | 128      |
|            | Conv-3    | 3 x 3  | 64          |
|            | Conv-4    |         |             | 92       |
|            | Concat-2  |         |             | 3 x 3    |
|            | Skip-2    |         |             | 3 x 3    |
|            | Conv-1    | 3 x 3  | 64          |
|            | Conv-2    | 3 x 3  | 64          |
|            | Conv-3    | 3 x 3  | 32          |
|            | Conv-4    |         |             | 63       |
|            | Concat-3  |         |             | 3 x 3    |
|            | Conv-1    | 3 x 3  | 64          |
|            | Conv-2    | 3 x 3  | 64          |
|            | Conv-3    | 3 x 3  | 32          |
|            | Conv-4    |         |             | 1        |

adopted, which can be formulated as

\[
L_t = -\frac{1}{C} \sum_{c=1}^{C} \hat{t}_c \log(t_c) + (1 - \hat{t}_c) \log(1 - t_c)
\]  (4)

where \( C \) denotes the number of classes (3 in our task), \( t_c \) denotes the predicted probability that the pixel belongs to class \( c \), and \( \hat{t}_c \) denotes the ground-truth class (1 or 0).

For the matting network, we use alpha loss and the compositional loss proposed by Xu et al. [7]. Alpha loss is defined as the \( L_1 \) distance between the predicted alpha matte \( \alpha \) and the ground-truth alpha matte \( \hat{\alpha} \). Compositional loss is defined as the \( L_1 \) distance between the predicted compositional image values \( I_p \) and the ground-truth image values \( \hat{I} \). However, the matting network produces blurry alpha mattes if we only train it using the above two losses. To address this issue, we introduce the gradient loss defined by the \( L_1 \) distance between the spatial gradients of the predicted and the ground-truth alpha mattes, denoted by \( G \) and \( \hat{G} \) respectively. The final loss function of the matting network is

\[
L_m = \lambda_\alpha \| \alpha - \hat{\alpha} \|_1 + \lambda_I \| I_p - \hat{I} \|_1 + \lambda_G \| G - \hat{G} \|_1
\]  (5)

where \( \lambda_\alpha \), \( \lambda_I \), and \( \lambda_G \) balance the three losses. We set \( \lambda_\alpha = 0.4 \), \( \lambda_I = 0.3 \), and \( \lambda_G = 0.3 \) in our experiments.

The overall loss function is a sum of above three losses:

\[
L = \lambda_p L_p + \lambda_I L_I + \lambda_m L_m
\]  (6)

We set \( \lambda_p = 0.001 \), \( \lambda_I = 0.01 \), \( \lambda_m = 1 \) in our experiments.

3.4 Implementation details

Since JMNet has several joint subnetworks, and it is hard to train them simultaneously, we divide the training process into two stages. In the first stage, we use the ground-truth pose heatmaps and trimaps as inputs to the subnetworks. In the second stage, we use the output of the corresponding subnetwork as the input instead of the ground truth for end-to-end training. We train the entire network for 5 epochs in each of the two stages. We use Adam as the optimizer and set the learning rate to 0.00001. The batch size is 32. It takes about 1 day for the two-stage training process on four NVidia GTX 1080 Ti GPUs.

We apply OpenPose [31] to find 2D pose keypoints for all the images in the human dataset. The heatmaps are then generated using these poses and regarded as the ground truth for training. The ground-truth trimap is generated by dilating the ground-truth alpha matte by a random kernel size in the range [5, 20]. Following Xu et al. [7], data augmentations is performed by random cropping and horizontal flipping. The images are cropped to different sizes: 320 \( \times \) 320, 480 \( \times \) 480, 640 \( \times \) 640, and then resized to a fixed size of 320 \( \times \) 320.
4 Experiments

4.1 Dataset

Xu et al. [7] provided the Deep Image Matting (DIM) dataset; its training set includes 493 foreground objects. We selected all 216 training images which only contain humans from the DIM dataset. To enhance the scalability of our method, we also collected an extra 30,000 human images for training. To obtain matte annotations, we asked a few volunteers to draw trimaps and used the closed-form matting algorithm [3] to estimate the alpha matte values as well as foreground color values. Since the backgrounds of these images are all pure colors, the estimated results are mostly accurate and can be regarded as ground truth.

Following Xu et al. [7], we constructed our training set by compositing the foreground images with the background images by alpha blending. We randomly sampled background images excluding humans from the MS-COCO dataset [32]. Each foreground image is composited with $N$ background images. We set $N = 100$ for the DIM dataset and $N = 1$ for our own dataset. Therefore, we had 51,600 human images in total for training; some examples are shown in Fig. 3. Meanwhile, we followed the same approach to construct the test set. We selected all 10 human images from the DIM test set where $N = 20$ and collected 300 human images with matte annotations where $N = 1$. Consequently, the test set for evaluations consisted of 500 images.

4.2 Comparison

Our approach is compared with Semantic Human Matting (SHM) [6], the state-of-the-art human matting method. We implemented and trained the SHM model by using our new dataset. We also compared our method with Closed Form (CF) matting [3] and Deep Image Matting (DIM) [7], where the trimap is produced by PSPNet50 [16] pre-trained for trimap generation. Figure 4 shows several predicted alpha matte results using the above methods. Because JMNet exploits the intrinsic structure of humans by pose estimation, our method can recover the complete human body when other methods fail. Our method also performs better for fine details owing to the gradient loss and skip connections.

We also conducted quantitative comparison experiments. Four metrics were used to measure the quality of the predicted alpha matte: SAD (sum of absolute differences), MSE (mean square error), gradient error, and connectivity error defined by Rhemann et al. [33]. All metrics were calculated over the entire image instead of only the unknown region and averaged by the total number of pixels. The results in Table 2 show that our method outperforms other state-of-the-art matting methods, for all metrics.

Due to the fully-convolutional architecture of JMNet, arbitrary resolutions of images can be adapted to our method for testing. We evaluated the inference time as well as the GPU memory cost of our method for different resolutions of input images.

Fig. 3 Pairs of images and alpha mattes in our new compositional dataset. The two leftmost foreground images are from the DIM dataset; the right three pairs were collected by ourselves.
Fig. 4 Comparison of JPPNet with state-of-the-art matting methods, including Closed Form (CF) matting, Deep Image Matting (DIM), and Semantic Human Matting (SHM), using our test set.

Table 2 Comparison of our method with Closed Form (CF) matting, Deep Image Matting (DIM), and Semantic Human Matting (SHM), using four metrics: SAD, MSE, gradient error, and connectivity error. Lower errors are better.

| Method  | SAD    | MSE    | Gradient | Connectivity |
|---------|--------|--------|----------|--------------|
| CF [3]  | 0.0418 | 0.0308 | 0.0715   | 0.0400       |
| DIM [7] | 0.0387 | 0.0289 | 0.0792   | 0.0372       |
| SHM [6] | 0.0353 | 0.0232 | 0.0826   | 0.0344       |
| JMNet   | 0.0175 | 0.0102 | 0.0275   | 0.0165       |

and compared them to those of SHM [6]. Experiments were conducted by using a 3.40 GHz, 4 core, Intel Core i7-4770 CPU, and an nVidia GTX 1080 Ti GPU. Since JMNet only utilizes the encoder to extract and share features with subsequent subnetworks, its computational cost and memory cost are both less than for SHM: see Table 3.

4.3 Ablation study

In order to investigate the effectiveness of the various components in our framework, we conducted several ablation studies. We excluded the pose network, skip connections, and the gradient loss respectively, and trained the resulting models in the same way as for the full model. We used SAD and gradient error as metrics to quantitatively analyze the effect of each component. Table 4 demonstrates that these components indeed improve the matting results.

4.4 Image composition result

In order to investigate the effectiveness of the various components in our framework, we conducted several ablation studies. We excluded the pose network, skip connections, and the gradient loss respectively, and trained the resulting models in the same way as for the full model. We used SAD and gradient error as metrics to quantitatively analyze the effect of each component. Table 4 demonstrates that these components indeed improve the matting results.
Table 4  Effectiveness of different components, as shown by omitting them

| Method                  | SAD  | Gradient error |
|------------------------|------|----------------|
| w/o pose               | 0.0183 | 0.0279       |
| w/o gradient loss      | 0.0175 | 0.0334       |
| w/o skip connections   | 0.0236 | 0.0560       |
| Full method            | 0.0175 | 0.0275       |

Fig. 5  Composition results produced by our method on the test set. Using the multi-stage joint network exploring the intrinsic structure of the human body, our method is able to recover the global silhouette of the human image as well as fine details in local regions like the hair or the tassel.

matte determined by our method. However, these input images come from the test set of the composited dataset. To demonstrate the generalizability of our method, we tested our model on several real images. Results in Fig. 6 show that our method is able to produce reasonable alpha matte results when compositing real images downloaded from the Internet.

5 Conclusions

We have proposed the Joint Matting Network (JMNet) for human image matting. JMNet is an end-to-end deep learning framework able to predict an alpha matte for human images without any user interaction. Even for some images with sophisticated backgrounds, we are able to generate accurate alpha matte results. With the assistance of global semantic structural information extracted by the pose estimation subnetwork, our trimap and matting subnetworks are capable of generating better matting masks than previous methods. The shared encoder also reduces the computational cost of extracting features. We further apply gradient loss and skip connections to produce finer and sharper alpha mattes. The composited results for Internet images demonstrate the generalizability of our approach.

In future, more semantic information from natural images could be incorporated to extend our method to a more generic solution. More accurate structural representations like the human parsing map, which are able to indicate per-pixel semantic labels such as hair or face, could be considered to improve the matting process by providing more adaptive attention. Our method could also be extended to human video by taking temporal coherence into consideration.

Acknowledgements

The authors would like to thank all the reviewers. We gratefully acknowledge the support of Jian-Cheng Liu who helped prepare and preprocess the dataset. This work was supported by National Natural Science Foundation of China (Grant Nos. 61561146393 and 61521002). Fang-Lue Zhang was supported by a Victoria Early-Career Research Excellence Award.

References

[1] Chen, X.; Qi, D.; Shen, J. Boundary-aware network for fast and high-accuracy portrait segmentation. arXiv preprint arXiv:1901.03814, 2019.
[2] Shen, X. Y.; Hertzmann, A.; Jia, J. Y.; Paris, S.; Price, B.; Shechtman, E.; Sachs, I. Automatic portrait segmentation for image stylization. *Computer Graphics Forum* Vol. 35, No. 2, 93–102, 2016.

[3] Levin, A.; Lischinski, D.; Weiss, Y. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 30, No. 2, 228–242, 2008.

[4] Chen, Q. F.; Li, D.; Tang, C. K. KNN matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 35, No. 9, 2175–2188, 2013.

[5] Shen, X. Y.; Tao, X.; Gao, H. Y.; Zhou, C.; Jia, J. Y. Deep automatic portrait matting. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, Vol. 9905. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 92–107, 2016.

[6] Chen, Q.; Ge, T. Z.; Xu, Y. Y.; Zhang, Z. Q.; Yang, X. X.; Gai, K. Semantic human matting. In: Proceedings of the 26th ACM International Conference on Multimedia, 618–626, 2018.

[7] Xu, N.; Price, B.; Cohen, S.; Huang, T. Deep image matting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2970–2979, 2017.

[8] Chuang, Y.-Y.; Curless, B.; Salesin, D. H.; Szeliski, R. A Bayesian approach to digital matting. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 264–271, 2001.

[9] Wang, J.; Cohen, M. F. Optimized color sampling for robust matting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–8, 2007.

[10] Gastal, E. S. L.; Oliveira, M. M. Shared sampling for real-time alpha matting. *Computer Graphics Forum* Vol. 29, No. 2, 575–584, 2010.

[11] He, K.; Rhemann, C.; Rother, C.; Tang, X.; Sun J. A global sampling method for alpha matting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2049–2056, 2011.

[12] Cho, D.; Tai, Y. W.; Kweon, I. Natural image matting using deep convolutional neural networks. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, Vol. 9906. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 626–643, 2016.

[13] Lutz, S.; Amplianitis, K.; Smolic, A. Alphagan: Generative adversarial networks for natural image matting. *arXiv preprint* arXiv:1807.10088, 2018.

[14] Tang, J. W.; Aksoy, Y.; Oztireli, C.; Gross, M.; Aydin, T. O. Learning-based sampling for natural image matting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3055–3063, 2019.

[15] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440, 2015.

[16] Zhao, H. S.; Shi, J. P.; Qi, X. J.; Wang, X. G.; Jia, J. Y. Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2881–2890, 2017.

[17] Zhang, Y. K.; Gong, L. X.; Fan, L. B.; Ren, P. R.; Huang, Q. X.; Bao, H. J.; Xu, W. A late fusion CNN for digital matting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7469–7478, 2019.

[18] Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science*, Vol. 9351. Navab, N.; Hornegger, J.; Wells, W.; Frangi, A. Eds. Springer Cham, 234–241, 2015.

[19] Chen, L. C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A. L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 40, No. 4, 834–848, 2018.

[20] Carreira, J.; Agrawal, P.; Fragkiadaki, K.; Malik, J. Human pose estimation with iterative error feedback. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4733–4742, 2016.

[21] Toshev, A.; Szegedy, C. DeepPose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1653–1660, 2014.

[22] Newell, A.; Yang, K. Y.; Deng, J. Stacked hourglass networks for human pose estimation. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, Vol. 9912. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 483–499, 2016.

[23] Wei, S.-E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4724–4732, 2016.

[24] Wei, Y.; Li, S.; Xu, K.; Zhang, X. Structured feature learning for pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–8, 2017.

[25] Liang, B.; Lin, Z.; Song, J.; Shi, J. Y. Look into person: Joint body parsing & pose estimation network and a new benchmark. *IEEE Transactions on...*
Pattern Analysis and Machine Intelligence Vol. 41, No. 4, 871–885, 2019.

[26] Kikuchi, T.; Endo, Y.; Kanamori, Y.; Hashimoto, T.; Mitani, J. Transferring pose and augmenting background for deep human-image parsing and its applications. Computational Visual Media Vol. 4, No. 1, 43–54, 2018.

[27] Wu, X.; Li, R. L.; Zhang, F. L.; Liu, J. C.; Wang, J.; Shamir, A.; Hu, S.-M. Deep portrait image completion and extrapolation. IEEE Transactions on Image Processing Vol. 29, 2344–2355, 2020.

[28] He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778, 2016.

[29] He, K. M.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 2961–2969, 2017.

[30] Lin, T. Y.; Dollar, P.; Girshick, R.; He, K. M.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2117–2125, 2017.

[31] Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S. E.; Sheikh, Y. OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields. arXiv preprint arXiv:1812.08008, 2018.

[32] Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. Microsoft COCO: Common objects in context. In: Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8693. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars T. Eds. Springer Cham, 740–755, 2014.

[33] Rheemann, C.; Rother, C.; Wang, J.; Gelautz, M.; Kohli, P.; Rott, P. A perceptually motivated online benchmark for image matting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1826–1833, 2009.

Xiao-Nan Fang is currently a Ph.D. student in Tsinghua University. He received his B.S. degree from Tsinghua University in 2018. His research interests include image and video processing, and computer graphics.

Tao Chen received his B.S. degree in fundamental science and Ph.D. degree in computer science from Tsinghua University, China, in 2005 and 2011, respectively. He is currently the Deputy General Manager of the AI Center at Visual China Group and the VP of Machine Learning, 500PX Inc. His research interests include multimedia, computer graphics, and computer vision.

Fang-Lue Zhang is a lecturer at Victoria University of Wellington. He received his doctoral degree from Tsinghua University in 2015 and bachelor degree from Zhejiang University in 2009. His research interests include image and video editing, computer vision, and computer graphics. He is a member of ACM and IEEE.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Other papers from this open access journal are available free of charge from http://www.springer.com/journal/41095. To submit a manuscript, please go to https://www.editorialmanager.com/cvmj.