Cascade Image Matting with Deformable Graph Refinement

Zijian Yu[1,2*], Xuhui Li[1*], Huijuan Huang[2], Wen Zheng[2], Li Chen[1†]
[1]School of Software, BNRist, Tsinghua University [2]Y-tech, Kuaishou Technology
{zj-yu19,lixh20}@mails.tsinghua.edu.cn, {huanghuijuan,zhengwen}@kuaishou.com, chenlee@tsinghua.edu.cn

Abstract

Image matting refers to the estimation of the opacity of foreground objects. It requires correct contours and fine details of foreground objects for the matting results. To better accomplish human image matting tasks, we propose the Cascade Image Matting Network with Deformable Graph Refinement(CasDGR), which can automatically predict precise alpha mattes from single human images without any additional inputs. We adopt a network cascade architecture to perform matting from low-to-high resolution, which corresponds to coarse-to-fine optimization. We also introduce the Deformable Graph Refinement (DGR) module based on graph neural networks (GNNs) to overcome the limitations of convolutional neural networks (CNNs). The DGR module can effectively capture long-range relations and obtain more global and local information to help produce finer alpha mattes. We also reduce the computation complexity of the DGR module by dynamically predicting the neighbors and apply DGR module to higher–resolution features. Experimental results demonstrate the ability of our CasDGR to achieve state-of-the-art performance on synthetic datasets and produce good results on real human images.

1. Introduction

Image matting refers to the problem of extracting high–quality alpha mattes (the opacity of foreground object at each pixel) from a set of given images. As a practical image processing technology, matting has a variety of applications for image and video editing. Generally, the composition of image $\mathbf{I}$ is expressed as a linear equation as follows:

$$\mathbf{I}_i = \alpha_i \mathbf{F}_i + (1 - \alpha_i)\mathbf{B}_i, \alpha_i \in [0, 1], \quad (1)$$

where $\mathbf{I}_i$ is the RGB color at pixel *i*, α_i is the matte value at pixel *i* and $\mathbf{F}_i$ and $\mathbf{B}_i$ are the RGB colors of the foreground and background at pixel *i*. Matting is a highly ill-posed problem, i.e., there are seven unknown values and only three known values at each pixel, which increases the difficulty of solving matting problems. Although existing works have provided effective ways to perform matting, they still present limitations.

The first limitation is that most existing works [47, 1, 19, 12, 52] have predicted alpha mattes by using a one-pass encoder–decoder network, which may result in inaccurate contours and artifacts when foreground and background have similar local features. This is mainly due to that those methods predict alpha mattes from single–scale features and cannot make full use of the global and local information contained in the image.

The second limitation is that existing CNN-based matting methods cannot well handle certain slender objects (e.g., human hair). In addition to the basic use of CNN in the network architecture, some matting works have tried to refine the details of alpha mattes after the backbone network with CNN-based module. Xu *et al*. [47] used a lightweight fully convolutional neural network (CNN) to generate sharp boundaries for alpha mattes. Cai *et al*. [1] proposed a propagation unit that could refine alpha mattes with accurate details and less artifacts. However, these CNN-based refinement methods are restricted by the fixed shape of convolutional kernels and a limited receptive field, leading to performance degradation when manipulating slender objects.

To overcome the first limitation, we simulate the matting logic of human. While meeting the matting tasks, people generally first determine the overall contour of the foreground object and then iteratively improve the details in boundary areas under the guidance of the contour. Therefore, we design a network cascade architecture for image matting to generate more accurate contours and details of foreground objects. Our method predicts coarse alpha mattes from low–resolution images as contours and then progressively supplements the details from high–resolution images under the guidance of the contours. Through this low-to-high, coarse-to-fine pipeline, our network can supplement local information with global information and estimate extremely finer alpha mattes with correct contours and precise details.

To overcome the limitations of CNN and produce bet-

*Joint first authors.
†The corresponding author is Li Chen.

ter performance on slender objects, we apply a graph neural network(GNN) to extract features with higher quality. Compared with CNN, GNN has shown its ability to better capture long-range dependencies from data. Some existing works [33, 21, 51] used GNN to improve the performance of detection and segmentation. However, these GNN-based methods are limited by high computation complexity and time consumption, as GNN requires large number of nodes and thus can only be applied to low–resolution feature maps or superpoints set obtained by clustering.

Inspired by deformable convolutional networks [10] that can dynamically adjust kernel shapes according to objects, we propose the Deformable Graph Refinement (DGR) module to reduce the computation cost of graph construction and propagation. The DGR module uses the convolutional network to predict the coordinates of neighbors and performs information aggregation and transmission among pixels.

We combine two solutions above and propose a method called the Cascade Image Matting Network with Deformable Graph Refinement (CasDGR). First, the network cascade architecture is designed to enhance the simulation of the coarse-to-fine matting logic. Second, the DGR module is adopted to improve the obtaining of more appropriate features and the handling of slender objects.

The main contributions of this study are as follows:

- We propose an end-to-end automatic image matting approach to produce high-quality alpha mattes from single RGB images.
- We design a network cascade architecture to estimate alpha mattes in a coarse-to-fine manner.
- We present a Deformable Graph Refinement module based on GNN that can preserve more details of the matting results and be applied on higher–resolution features.

Some existing work [47, 31, 26] require trimaps as additional inputs. However, the construction of high–quality trimaps is complicated. Automatic matting methods [52, 38, 34, 30] whose inputs do not contain trimaps are more challenging, but more convenient and feasible for some applications, such as matting for human only. Our CasDGR is also automatic and can achieve good matting performance with single RGB images. Similar to [38], we test our method on Adobe human image dataset [47]. The experimental results demonstrate that our method can achieve state-of-the-art performance and produce excellent visual results. What's more, our automatic matting approach outperforms existing trimap-based methods both quantitatively and qualitatively. We also test the CasDGR on natural human images. Our method shows good performance on real-world human images as well.

2. Related Work

2.1. Image Matting

Current image matting methods can be divided into traditional methods and learning-based methods.

Traditional methods. Sampling-based methods [9, 13, 39, 17, 45] mainly use statistical methods to sample and model the color of known foreground and background regions and determine the best color pair of each unknown pixel and calculate the alpha mattes. Propagation-based methods [6, 23, 42, 22, 24] propagate the alpha values of known regions to unknown regions according to the affinities among adjacent pixels. Nevertheless, traditional methods utilize color information and location information instead of semantic information and context information, which may lead to loss of essential detail.

Learning-based methods. Learning-based matting methods compensate the disadvantages of traditional methods and generally offer better performance. Trimap-based learning methods require annotated trimaps as additional inputs. Cho *et al*. [8] utilized the results of [6] and [23] and normalized RGB color to predict alpha mattes by using a deep CNN. Xu *et al*. [47] first proposed an encoder–decoder structure network to estimate alpha mattes. The refinement stage in their work could produce extrmely sharp boundaries. Hou *et al*. [19] used two encoders to extract local and global context information and perform matting. Cai *et al*. [1] adopted a multi-task learning method to complete two subtasks, and a propagation unit was used to process the results of the two subtasks and consequently obtain the final alpha mattes. Forte and Pitié [12] proposed to predict foregrounds, backgrounds, alpha mattes by using a single encoder–decoder. Hao *et al*. [31] optimized the upsampling operator and applied it to image matting. Tang *et al*. [43] utilized sampling networks and a matting network to perform color sampling and matting.

Automatic methods do not need additional trimaps, hence avoid the constraints of trimaps. Shen *et al*. [40] estimated trimaps by using a CNN and performed matting with method of [23]. Sengupta *et al*. [38] used disturbed backgrounds and segmentation results as additional inputs to simultaneously predict alpha mattes and foregrounds. Zhang *et al*. [52] first obtained the probability maps of a foreground and a background and then fused them to obtain the final alpha mattes. Liu *et al*. [30] used coarse annotated data coupled with fine annotated data to improve matting performance. Qiao *et al*. [34] used channel and spatial attention mechanisms to extract multi-level features from a set of single images.

Most deep learning methods aim to enhance the matting based on the single encoder–decoder architecture and do not provide effective refinement stage.We apply the network cascade architecture to our CasDGR to perform the

matting process in a coarse-to-fine manner. A novel DGR module is proposed for feature refinement. The experimental results prove that both proposed techniques can achieve a considerably improvement in the matting results.

2.2. Network Cascade

Network cascade is an effective architecture for many computer vision tasks, such as detection [28, 3], segmentation [25], and pose estimation [7]. The central idea of using the network cascade is to solve challenging tasks in a coarse-to-fine manner. Cai *et al*. [3] presented Cascade R-CNN to achieve progressive refinement of detection results. Chen *et al*. [7] predicted multiple heatmaps of human keypoints in high-to-low resolution and fused them using RefineNet. Li *et al*. [25] handled easy regions in shallow layers and hard regions in deep layers to improve the accuracy and speed of semantic segmentation. To the best of our knowledge, the CasDGR is an early attempt to adopt a network cascade architecture into image matting tasks.

2.3. Graph Neural Network

Many graph neural networks (GNN) [37, 27, 8, 20, 16, 44] have been proposed to solve the general problems of graphs. Compared with CNN, GNN can better capture long-range dependencies from data, which benefits many computer vision tasks, such as detection [41, 32, 48], segmentation [33, 21, 51, 48] , and pose estimation [50, 2]. Luo *et al*. [32] designed Cascade-GNN for RGBD salient object detection to exploit useful information from RGB and depth images. Cai *et al*. [2] used the graph convolution network to exploit the spatial and temporal relationship of 3D human body and hand pose. In [33, 21, 41], the authors proposed GNN-based methods for segmentation and detection on 3D point clouds. However, the methods manifest restrictions on data size resulting from the high computation cost and low running speed of GNN. DGMN [51] and RepGNN [48] can reduce the computation cost by dynamically sampling the nodes, consequently improving the performance of segmentation and detection. Our work can predict the neighbors of each node and adopted into higher resolution feature maps, thus helping to obtain more details to solve the image matting problems.

3. Approach

In this section, we first introduce the overall network architecture and details of our CasDGR. Then, the loss functions and implementation details are presented.

3.1. Cascade Network Design

As shown in Figure 1, the central idea of our approach is to use a network cascade architecture to predict alpha mattes from low to high resolution. The CasDGR consists of five stages in total. Similarly to most previous works, each stage is composed of an encoder–decoder U-structure network. Inspired by U^2-Net [35], we use Residual U-block (RSU) as the backbone network in each stage owing to its ability to extracting multi-scale features and its low computation cost. The input of each stage contains an image with different resolutions scaled from the original image. No other additional inputs are required by our network. In stage m ($m \in [2, 4]$), we first use a 3×3 convolutional layer to generate the 64-channel feature map $\mathbf{F}_{in}^{m}$ from the input image. Then the $\mathbf{F}_{in}^{m}$ is concatenated with the refined feature map $\mathbf{F}_{re}^{m-1}$ from the previous stage. The RSU block takes the concatenation of two feature maps as the input and then outputs feature map $\mathbf{F}_{out}^{m}$ with the same resolution as $\mathbf{F}_{in}^{m}$. $\mathbf{F}_{out}^{m}$ is fed into the Deformable Graph Refinement (DGR) module to generate the 64-channel refined feature map $\mathbf{F}_{re}^{m}$. Finally, a 3×3 convolutional layer is used to predict the 1-channel alpha mattes from $\mathbf{F}_{re}^{m}$. Moreover, $\mathbf{F}_{re}^{m}$ is upsampled two times for concatenation with $\mathbf{F}_{in}^{m+1}$ in the next stage.

The details of the RSU blocks are also shown in Figure 1. The encoder part continuously performs convolution and downsampling on the feature map, whereas the decoder part performs upsampling and convolution to restore the feature map to the original resolution. Skip connections are applied to corresponding layers between the encoder and decoder. Atrous convolution is used in the deep layers to further enlarge the receptive field. Different from original RSU blocks, we use group normalization (GN) [46] instead of batch normalization (BN) after each convolutional layer in our network, because our CasDGR is trained with small a batch size (2 on each GPU). Furthermore, the performance of BN may degrade when the batch size is small.

The CasDGR handles image matting tasks in a coarse-to-fine manner and predicts multiple alpha mattes from low to high resolution. In the earlier stages, the network extracts more global information in concordance with the much larger receptive field from the downsampled input image. This approach can help to improve the detection of the foreground object area. The predicted alpha mattes from these earlier stages can be regarded as coarse segmentation masks of the foreground object in visual perception. In the later stages, the predicted alpha mattes are further improved by using both higher–resolution input images and feature maps from previous stage; the former supplements the detail that may have been lost in earlier stages, whereas the latter provides rich semantic information. The DGR module further improves the quality of the generated feature maps by means of a graph-based model, which will be discussed in the succeeding sections. Thus, our CasDGR can progressively refine the details from stage 1 to 5 while maintaining the correct contour of the foreground correct and consequently produce high–quality alpha mattes.

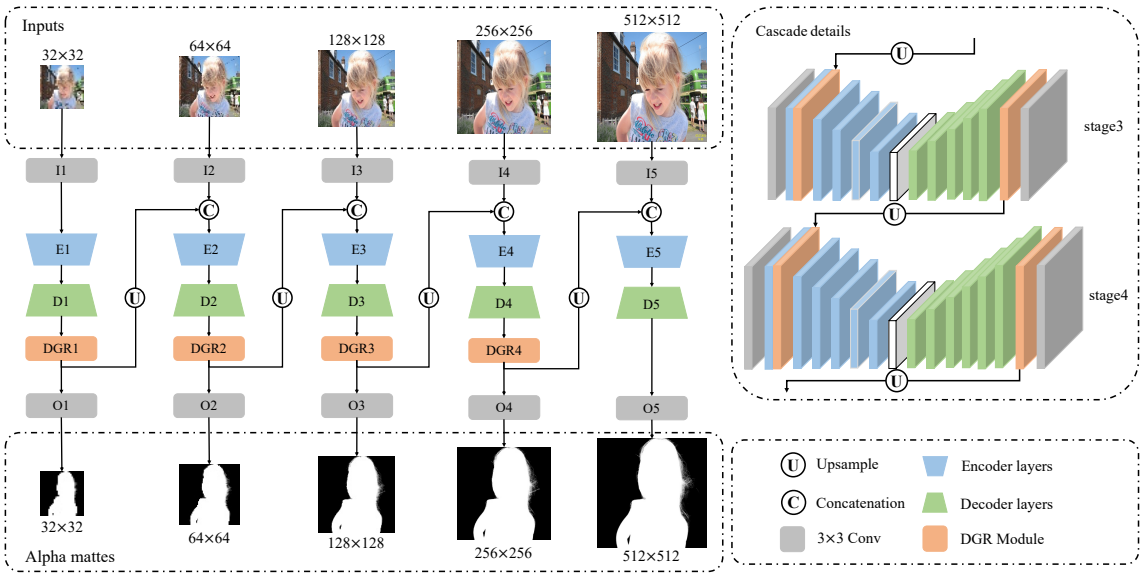

Figure 1. Overview of the proposed CasDGR. The main architecture is a cascade network contains 5 stages, where each stage is an encoder–decoder network followed by a DGR module. Given an input image, we downsample it to multi-scale inputs for each stage and estimate multi–resolution alpha mattes from low to high. We only use the predicted alpha matte of the last stage for further evaluation.

3.2. Deformable Graph Refinement

We propose the Deformable Graph Refinement (DGR) module for feature map refinement. The details of the DGR module are shown in Figure 2. We regard the feature map with a shape of $H \times W \times C$ as a composition of $H \times W$ nodes and construct a graph on them in which each node entails a C-dimension feature. The DGR module is inspired by deformable convolutional networks [10], which dynamically adjust convolution kernels. We assume that each pixel in the feature map $\mathbf{F}_{out} \in \mathbb{R}^{H \times W \times C}$ outputed from the decoder initially has K adjacent neighbors initially and use a convolutional layer to apply a 2D offset to each neighbor. Then, we calculate the neighbor coordinates and use the bilinear interpolation method to obtain the neighbor feature values from $\mathbf{F}_{out}$. We design a model for the neighbors' information aggregation and the feature map refinement. For a node i in $\mathbf{F}_{out}$, we refine its feature as follows:

$$s_{ij} = (\mathbf{W}_1 \mathbf{F}^i_{out})^T (\mathbf{W}_2 \mathbf{F}^j_{out}), j \in \mathcal{N}(i), \tag{2}$$

$$\beta_{ij} = \frac{\exp(s_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(s_{ik})}, \tag{3}$$

$$\mathbf{F}^i_{re} = \sigma\Big(\sum_{j \in \mathcal{N}(i)} \beta_{ij} \mathbf{W}_2 \mathbf{F}^j_{out}\Big), \tag{4}$$

, where $\mathcal{N}(i)$ is the neighbors set of node i. $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{C' \times C}$ are two weight matrices that can be optimized. Eq. 2 calculates the similarity s_{ij} between node i and its neighbor j. Then, Eq. 3 is calculated with a softmax function to normalize s_{ij}. The β_{ij} after normalization are the weights of neighbor j for node i. Finally, in Eq. 4, we aggregate the features of all neighbors with different weights. σ is the ReLU activation function. This feature refinement stage can be performed iteratively. By using the DGR module, our network can capture long-range relations between the distant pixels. DGR can also reduce the computation complexity and time consumption of graph construction by dynamically predicting the neighbors. We apply the DGR module to stages 1 to 4 in the cascade network and the highest resolution of the feature map refined by DGR can reach 256×256, which is higher than those in the previous works [33, 48]. We use the refined feature maps for feature connection and alpha prediction.

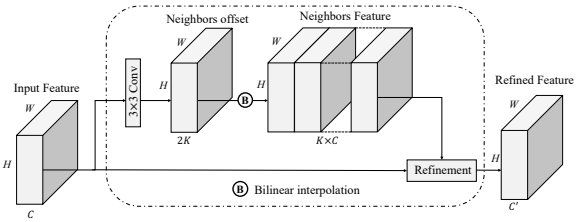

Figure 2. Illustration of the Deformable Graph Refinement (DGR) module. The input is the feature map output by the RSU block. DGR first predicts K neighbor coordinates for each node and calculates feature values of neighbors. Then DGR updates feature values of each node by a refinement stage.

3.3. Loss Functions

In the training process, we use supervision at each stage of the CasDGR. Our loss function is defined as follows:

$$\mathcal{L} = \sum_{m=1}^{M-1} \lambda_a^m \mathcal{L}_a^m + \mathcal{L}_a^M + \lambda_c \mathcal{L}_c^M + \lambda_g \mathcal{L}_g^M, \quad (5)$$

, where $\mathcal{L}_a^m (M = 5$ represents five stages) is the alpha prediction loss between the output alpha of stage m and the labels with the same resolution, $\mathcal{L}_c$ is the compositional loss, and $\mathcal{L}_g$ is the gradient loss. We use all three losses for the last stage and only use alpha prediction loss for the previous stages. λ_a^m, λ_c, and λ_g are the weights of each loss item. We use the normalized L1 loss to calculate all three losses:

$$\mathcal{L}_a^m = \frac{1}{|\Omega|} \sum_{i \in \Omega} ||\hat{\alpha}_i^m - \alpha_i^m||_1, \quad (6)$$

, where α_i^m is the predicted alpha values of stage m at pixel i, $\hat{\alpha}_i^m$ is the ground truth alpha values resized to the same resolution as α^m at pixel i, and $|\Omega|$ is the number of pixels in α_i^m and $\hat{\alpha}_i^m$.

$$\mathcal{L}_c = \frac{1}{|\Omega|} \sum_{i \in \Omega} ||\mathbf{I}_i - \alpha_i \mathbf{F}_i - (1 - \alpha_i)\mathbf{B}_i||_1, \quad (7)$$

, where $\mathbf{I}$ is the input image combined by foreground $\mathbf{F}$, background $\mathbf{B}$, and ground truth alpha matte, similiar to those in Eq. 1. α is the predicted result of the last stage.

$$\mathcal{L}_g = \frac{1}{|\Omega|} \sum_{i \in \Omega} ||\nabla \hat{\alpha}_i - \nabla \alpha_i||_1, \quad (8)$$

, where $\nabla \hat{\alpha}$ and $\nabla \alpha$ represent the normalized gradient of the predicted alpha and the ground truth alpha.

The training process aims to minimize the $\mathcal{L}$ of Eq. 5. $\mathcal{L}_a, \mathcal{L}_c$ can improve the pixel-wise accuracy of the predicted alpha mattes, and $\mathcal{L}_g$ is beneficial to the production of highly precise boundaries. We choose the predicted results of the last stage as the final of the output alpha mattes.

3.4. Implementation Details

We implement CasDGR by using PyTorch. In the training process, all images are randomly cropped to a resolution between 512×512 and 800×800 and then resized to 512×512. For the data augmentation, we adopt horizontally random flipping together with brightness, contrast, and saturation augmentation on each training pair to avoid overfitting. We downsample the 512×512 images to lower resolutions and feed them into the different stages of our method. The training set is shuffled at each epoch. For the group normalization layer in our network, the input feature map is separated into several 32-channel groups. We train our network from scratch until the loss converges. All convolutional layers in RSU blocks are initialized using the Xavier method [14]. Parameters of the 3×3 convolutional layers in the DGR module are initialized to zero. The adam optimizer is used for loss optimization, with the initial learning rate set to 1e-4 and the other hyper parameters set to default. We clip the predicted alpha values of each stage to 0 to 1 for loss calculation and set $\lambda_a^m = \lambda_c = \lambda_g = 1$ in Eq. 5 in the experiments.

During testing, the input images are resized to 512×512 before feeding them into the network. We evaluate different metrics between the 512×512 predicted alpha mattes and ground truth. We train our CasDGR on 2 RTX 3090 GPUs with a batch size of 4. Only about 1 day are needed for the network to converge on the training set.

4. Experiments

In this section, we compare our approach with existing matting methods on the Adobe human image dataset, which is collected from the Adobe Composite-1k Dataset [47]. We show the quantitative and visual results of all testing methods and perform ablation studies on our CasDGR to demonstrate the importance of essential architectures and components in our method.

4.1. Dataset and Evaluation Metrics

Dataset. Adobe Composite-1k Dataset [47] contains 431 foreground images for training and 50 foreground images for testing with high–quality alpha annotations. Following Sengupta's work [38], we use a subset of 280 images in the experiments (269 images for training and 11 images for testing). We filter the semi-transparent objects in the dataset to closely simulate the data distribution to the human matting scene in the real world. For the training set, each foreground image is combined with 100 background images from the COCO dataset [29]. For the testing set, each foreground image is combined with 20 background images from the PASCAL VOC2012 dataset [11].

Evaluation metrics. We use four common metrics in image matting to evaluate the predicted alpha mattes, namely sum of absolute differences (SAD), mean squared error (MSE), gradient error (Grad), and connectivity error (Conn). Generally, the SAD and MSE metrics are more focused on numerical differences, whereas the Grad and Conn metrics proposed by [36] pay more attention to the visual perception of human observers.

4.2. Ablation Study on the Adobe Testing Dataset

To verify the role of some architectures and components of our method, we completed the ablation studies discussed below by using the Adobe testing dataset.

Ablation on DGR. Table 1 shows the influence of different number of neighbors and iteration times of refinement stage on matting performance. Compared with Cascade net-

Model	SAD	MSE	Grad	Conn
Ours-Baseline	3.78	0.0065	4.67	3.73
Ours-Cascade	2.92	0.0046	2.85	2.77
CasDGR$_{K=1,1-layer}$	2.25	0.0025	2.45	2.10
CasDGR$_{K=1,2-layer}$	2.05	0.0021	2.16	1.88
CasDGR$_{K=5,1-layer}$	1.93	0.0018	1.95	1.74
CasDGR$_{K=5,2-layer}$	**1.76**	**0.0015**	**1.66**	**1.54**
CasDGR$_{K=9,1-layer}$	2.16	0.0023	2.30	1.99
CasDGR$_{K=9,2-layer}$	1.84	0.0017	1.79	1.63

Table 1. Ablation study on DGR module. Ours-Baseline: 1-stage network. Ours-Cascade: 5-stage cascade network without DGR. CasDGR: cascade network with DGR, K means the number of neighbors in DGR, and $layer$ means the number of iterations.

Model	Ours-CasDCN	Ours-CasDGR	
$layer$	1	1	2
SAD	2.13	1.93	1.76
MSE	0.0023	0.0018	0.0015
Grad	2.27	1.95	1.66
Conn	1.93	1.74	1.54
Flops(G)	+8.71	+5.65	+8.36
Params(M)	+0.57	+0.12	+0.16
Inference Time(ms)	51.23	41.33	48.56

Table 2. Ours-CasDCN vs Ours-CasDGR($K = 5$). The value of FLOPs and Params are the increased value in contrast to Ours-Cascade. The results are measured with 512×512 input size on one GeForce RTX 3090 card. The batch size is 1.

work without DGR, CasDGR with different settings can improve all four metrics on on Adobe testing dataset. We find that only considering 1 neighbor can increase four evaluation metrics effectively. As the number of neighbors increases form 1 to 5, test results are improved too. However, further increasing the number of neighbors to 9 will lead a decline of matting performance.

Increasing the iteration times of refinement stage is also beneficial to image matting. For different settings of K, CasDGR with 2 iterations produce better results than 1 iteration. More iterations will increase the time and memory consumption as well. To balance the efficiency and effectiveness of the model, we choose $K = 5$ and set iteration times to 2 as default in other experiments.

Role of Network Cascade Architecture. As shown in Table 1, the cascade network has achieved substantially improvement on all metrics compared with the the baseline network, which only uses the 1-stage network for matting. According to the visual results in Figure 3, Ours-Baseline produces some artifacts in the results, whereas Ours-Cascade can generate more visually accurate alpha mattes. The network cascade architecture has effectively improved the quantitative and visual results for matting.

Role of the DGR. Table 1 has shown the improvement of

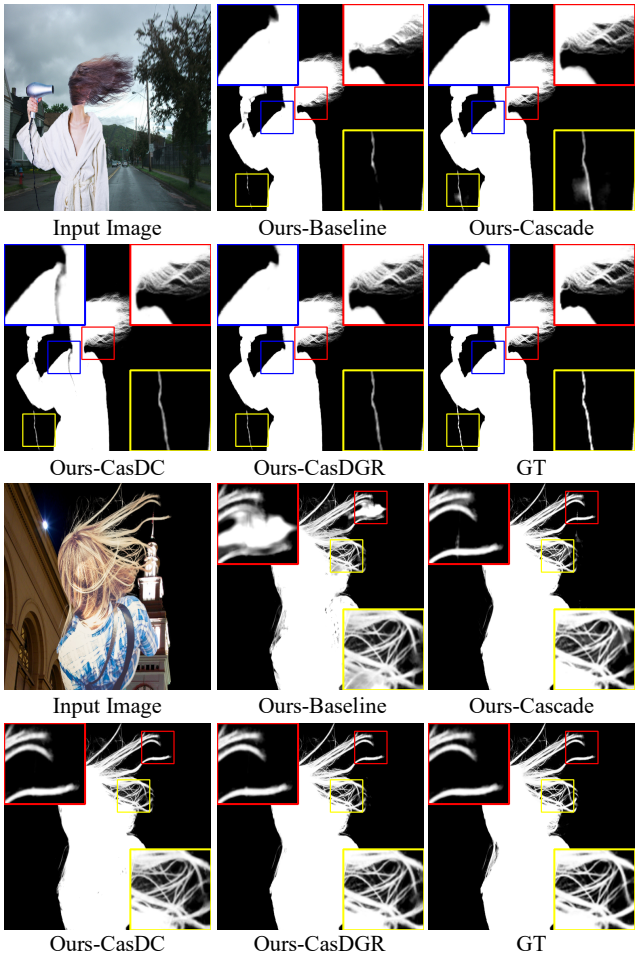

Figure 3. Visual results of ablation studies.

DGR on evaluation metrics. In terms of the visual results in Figure 3, Ours-CasDGR can further refine the results compared with Ours-Cascade, which reduces some artifacts and is completed with more detail for the alpha mattes.

In addition, certain details in Figure 3, can be used to clearly analyze the matting refinement process of our method. In the case of the image of women with hand-held hair dryers, Ours-Baseline does not predict the complete wire of hair dryer in the lower left corner. After the network cascade, Ours-Cascade can predict a relatively complete wire, but some artifacts around it are apparent. Finally, after the refinement processing of Ours-CasDGR, the artifacts are removed, and a complete and fine wire is obtained. This step-by-step refinement process also verifies the design ideas and feasibility of our method.

Comparision with the DCN. We demonstrate the superiority of DGR module by replacing the DGR in our matting network with deformable convolutional networks [10]. As shown in Table 2, Ours-CasDGR outperforms Ours-

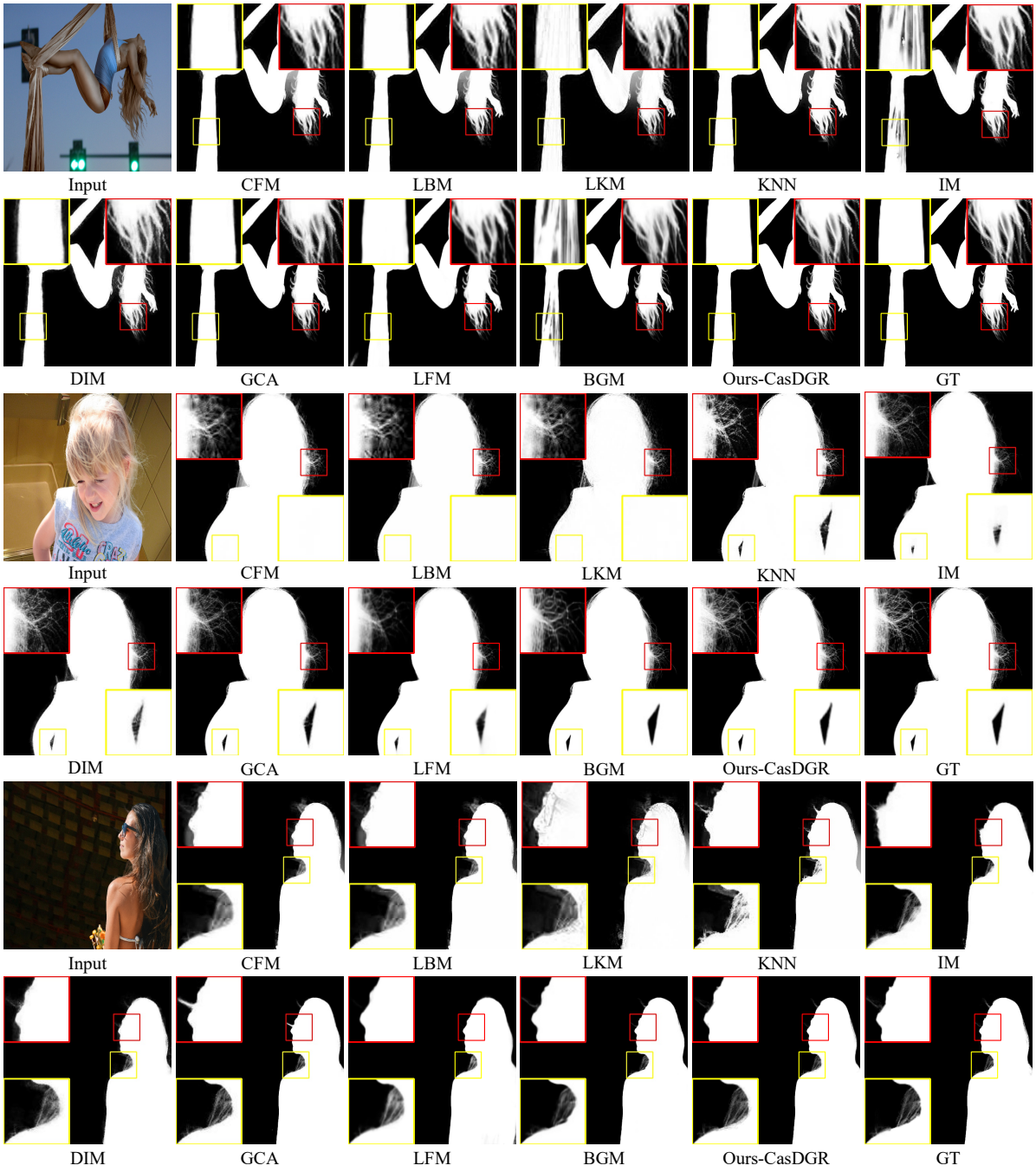

Figure 4. The visual results on Adobe testing dataset.

CasDCN on all metrics. According to the comparison of some model attributes, Ours-CasDGR can achieve better matting results with less Flops, Params, and inference time, which further demonstrates the improvements from using our approach. The visual results in Figure 3 show that Ours-CasDGR is also visually superior to Ours-CasDCN, based on the finer alpha mattes of the former method.

4.3. Comparison on the Adobe Testing Dataset

We compare our approach on the constructed Adobe human image dataset with different kinds of available approaches. **The traditional methods**: Closed-Form Matting (CFM) [23], Learning Based Matting (LBM) [53], KNN Matting (KNNM) [6], Random Walks Matting (RWM) [15],

Method	SAD	MSE	Grad	Conn
CFM [23]	3.48	0.0040	3.87	3.35
LBM [53]	3.68	0.0047	4.17	3.65
KNNM [6]	3.73	0.0044	3.90	3.67
RWM [15]	4.96	0.0093	10.56	4.93
LKM [18]	5.52	0.0053	5.32	4.65
IM [31]	2.29	0.0022	2.51	2.06
DIM [47]	2.58	0.0025	2.93	2.42
GCAM [26]	1.89	0.0017	1.99	1.68
BGM [38] - *Seg,* B'	2.30	0.0025	2.34	2.10
BGM [38] - *Seg,* B	2.28	0.0024	2.29	2.08
LFM [52]	4.35	0.0067	4.01	3.98
Ours-CasDGR	**1.76**	**0.0015**	**1.66**	**1.54**

Table 3. Results on the Adobe testing dataset. *Seg,* B', B: coarse segmentation results, disturbed backgrounds with Gaussian noises, and original backgrounds for Background-Matting [38].

and Large Kernels Matting (LKM) [18]. **The trimap-based learning methods**: Deep Image Matting (DIM) [47], IndexNet Matting (IM) [31], and Guided Contextual Attention Matting (GCAM) [26]. **The automatic learning methods**: Late Fusion Matting (LFM) [52] and Background Matting (BGM) [38].

During the evaluation, we resize input images to 512×512 resolution to inference the alpha mattes and compute four metrics between the predicted alpha mattes and ground truths. For the approaches requiring trimaps, we resize the original trimaps in Adobe dataset to 512×512 resolution as additional inputs. As BGM [38] needs segmentation results and disturbed backgrounds as additional inputs, we generate the segmentation results by applying person segmentation [4] and adding erosion (5 iterations), dilation (10 iterations) and a Gaussian blur (σ = 5). We also generate the disturbed backgrounds by adding Gaussian noises $\eta \sim \mathcal{N}(\mu = 3, \sigma = 3)$ to the original backgrounds. The manner of generating segmentation results and disturbed backgrounds are the same as those in BGM [38].

The quantitative results are shown in Table 3. The implications of our experimental results are as follows:

Our CasDGR can achieve state-of-the-art results on all metrics among all testing approaches on Adobe testing dataset, i.e., the traditional methods, trimap-based, and automatic methods mentioned above. The experimental results demonstrate that our approach can achieve the best human matting performance by using a single input image.

Our CasDGR outperforms other matting methods especially on the Grad and Conn metrics. As Grad and Conn focus more on the visual effects of human observers, the comparison results indicate that the CasDGR can achieve great matting performance in visual perception, which is also proven by the visual results in Figure 4.

Figure 5. Results on real-world images.

As shown in Figure 4, our CasDGR has a high–quality visual effect on human images and can preserve fine contour and detail of the foreground object. Although GCA [26] and BGM [38] can also generate precise alpha mattes, they require fine trimaps or backgrounds when inferencing. Our CasDGR only needs single RGB images, which is much more convenient for matting applications.

4.4. Results on Real Image Dataset

As a real-world application, the performance on real-world data is also significant for matting methods. To verify the matting effect of our CasDGR on real-world images, we test our approach on 1) human matting dataset constructed by Chen *et al*. [5] and 2) Real World Portrait-636 dataset provided by Yu *et al*. [49]. Figure 5 shows that our CasDGR model trained on Adobe human image dataset can also produce high-quality alpha mattes on real-world images without additional inputs.

5. Conclusions

In this study, we propose a Cascade Image Matting Network with Deformable Graph Refinement (CasDGR), that can produce high–quality alpha mattes from single RGB images. We adopt the network cascade architecture to progressively refine the foreground details. The proposed DGR module applies GNN on higher–resolution features to further improve matting performance. The experimental results on the synthetic dataset and real-world images demonstrate the superiority and generalization of our approach.

6. Acknowledgement

We are grateful for the valuable feedback and comments provided by the anonymous reviewers. This research was partially supported by the Tsinghua-Kuaishou Institute of Future Media Data, the National Natural Science Foundation of China (Grant Nos.61972221, 62021002) and the National Key R&D Program of China (2019YFB1405703), TC190A4DA/3.

References

[1] Shaofan Cai, Xiaoshuai Zhang, Haoqiang Fan, Haibin Huang, Jiangyu Liu, Jiaming Liu, Jiaying Liu, Jue Wang, and Jian Sun. Disentangled image matting. *International Conference on Computer Vision (ICCV)*, pages 8818–8827, 2019.

[2] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat-Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. *International Conference on Computer Vision (ICCV)*, pages 2272–2281, 2019.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6154–6162, 2018.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *In The European Conference on Computer Vision (ECCV)*, 2018.

[5] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. *ACM International Conference on Multimedia*, pages 618–626, 2018.

[6] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. KNN matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2175–2188, 2013.

[7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7103–7112, 2018.

[8] Donghyeon Cho, Yu-Wing Tai, and In-So Kweon. Natural image matting using deep convolutional neural networks. *In The European Conference on Computer Vision (ECCV)*, 9906:626–643, 2016.

[9] Yung-Yu Chuang, Brian Curless, David Salesin, and Richard Szeliski. A bayesian approach to digital matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 264–271, 2001.

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.

[11] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[12] Marco Forte and François Pitié. F, b, alpha matting. *CoRR*, abs/2003.07711, 2004.

[13] Eduardo Simoes Lopes Gastal and Manuel M. Oliveira. Shared sampling for real-time alpha matting. *Computer Graphics Forum (CGF)*, 29(2):575–584, 2010.

[14] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 9:249–256, 2010.

[15] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive alpha-matting. *VIIP*, 2005:423–429, 2005.

[16] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Conference and Workshop on Neural Information Processing Systems*, pages 1024–1034, 2017.

[17] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2049–2056, 2011.

[18] Kaiming He, Jian Sun, and Xiaoou Tang. Fast matting using large kernel matting laplacian matrices. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:2165–2172, 2010.

[19] Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. *International Conference on Computer Vision (ICCV)*, pages 4129–4138, 2019.

[20] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017.

[21] Loïc Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4558–4567, 2018.

[22] Philip Greggory Lee and Ying Wu. Nonlocal matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2193–2200, 2011.

[23] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008.

[24] Anat Levin, Alex Rav-Acha, and Dani Lischinski. Spectral matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1699–1712, 2008.

[25] Xiaoxiao Li, Ziwei Liu, Ping Luo, Chen Change Loy, and Xiaoou Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6459–6468, 2017.

[26] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. *AAAI*, pages 11450–11457, 2020.

[27] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. *International Conference on Learning Representations*, 2016.

[28] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

[29] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *In The European Conference on Computer Vision (ECCV)*, 8693:740–755, 2014.

[30] Jinlin Liu, Yuan Yao, Wendi Hou, Miaomiao Cui, Xuansong Xie, Changshui Zhang, and Xian-Sheng Hua. Boosting semantic human matting with coarse annotations. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8560–8569, 2020.

[31] Hao Lu, Yutong Dai, Chunhua Shen, and Songcen Xu. Indices matter: Learning to index for deep image matting. *International Conference on Computer Vision (ICCV)*, pages 3265–3274, 2019.

[32] Ao Luo, Xin Li, Fan Yang, Zhicheng Jiao, Hong Cheng, and Siwei Lyu. Cascade graph neural networks for RGB-D salient object detection. *In The European Conference on Computer Vision (ECCV)*, 12357:346–364, 2020.

[33] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3d graph neural networks for RGBD semantic segmentation. *International Conference on Computer Vision (ICCV)*, pages 5209–5218, 2017.

[34] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13673–13682, 2020.

[35] Xuebin Qin, Zichen Vincent Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaïane, and Martin Jägersand. U^2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020.

[36] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1826–1833, 2009.

[37] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2009.

[38] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2288–2297, 2020.

[39] Ehsan Shahrian, Deepu Rajan, Brian L. Price, and Scott Cohen. Improving image matting using comprehensive sampling sets. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 636–643, 2013.

[40] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. Deep automatic portrait matting. *In The European Conference on Computer Vision (ECCV)*, 9905:92–107, 2016.

[41] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1708–1716, 2020.

[42] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. *ACMTransactions on Graphics (TOG)*, 23(3):315–321, 2004.

[43] Jingwei Tang, Yagiz Aksoy, Cengiz Öztireli, Markus H. Gross, and Tunç Ozan Aydin. Learning-based sampling for natural image matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3055–3063, 2019.

[44] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *International Conference on Learning Representations*, 2018.

[45] Jue Wang and Michael F. Cohen. Optimized color sampling for robust matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[46] Yuxin Wu and Kaiming He. Group normalization. *In The European Conference on Computer Vision (ECCV)*, 11217:3–19, 2018.

[47] Ning Xu, Brian L. Price, Scott Cohen, and Thomas S. Huang. Deep image matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 311–320, 2017.

[48] Changqian Yu, Yifan Liu, Changxin Gao, Chunhua Shen, and Nong Sang. Representative graph neural network. *In The European Conference on Computer Vision (ECCV)*, pages 379–396, 2020.

[49] Qihang Yu, Jianming Zhang, He Zhang, Yilin Wang, Zhe Lin, Ning Xu, Yutong Bai, and Alan L. Yuille. Mask guided matting via progressive refinement network. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[50] Hong Zhang, Hao Ouyang, Shu Liu, Xiaojuan Qi, Xiaoyong Shen, Ruigang Yang, and Jiaya Jia. Human pose estimation with spatial contextual information. *CoRR*, abs/1901.01760, 2019.

[51] Li Zhang, Dan Xu, Anurag Arnab, and Philip H. S. Torr. Dynamic graph message passing networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3723–3732, 2020.

[52] Yunke Zhang, Lixue Gong, Lubin Fan, Peiran Ren, Qixing Huang, Hujun Bao, and Weiwei Xu. A late fusion CNN for digital matting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7469–7478, 2019.

[53] Yuanjie Zheng and Chandra Kambhamettu. Learning based digital matting. *International Conference on Computer Vision (ICCV)*, pages 889–896, 2009.