Disentangled Image Matting

Shaofan Cai[1*], Xiaoshuai Zhang[1,2*], Haoqiang Fan[1], Haibin Huang[1],
Jiangyu Liu[1], Jiaming Liu[1], Jiaying Liu[2], Jue Wang[1], and Jian Sun[1]

[1]Megvii Technology

[2]Institute of Computer Science and Technology, Peking University

{caishaofan, fhq, huanghaibin, liujiangyu, liujiaming, wangjue, sunjian}@megvii.com

{jet, liujiaying}@pku.edu.cn

Abstract

Most previous image matting methods require a roughly-specified trimap as input, and estimate fractional alpha values for all pixels that are in the unknown region of the trimap. In this paper, we argue that directly estimating the alpha matte from a coarse trimap is a major limitation of previous methods, as this practice tries to address two difficult and inherently different problems at the same time: identifying true blending pixels inside the trimap region, and estimate accurate alpha values for them. We propose AdaMatting, a new end-to-end matting framework that disentangles this problem into two sub-tasks: trimap adaptation and alpha estimation. Trimap adaptation is a pixel-wise classification problem that infers the global structure of the input image by identifying definite foreground, background, and semi-transparent image regions. Alpha estimation is a regression problem that calculates the opacity value of each blended pixel. Our method separately handles these two sub-tasks within a single deep convolutional neural network (CNN). Extensive experiments show that AdaMatting has additional structure awareness and trimap fault-tolerance. Our method achieves the state-of-the-art performance on Adobe Composition-1k dataset both qualitatively and quantitatively. It is also the current best-performing method on the alphamatting.com *online evaluation for all commonly-used metrics.*

1. Introduction

Image matting refers to the problem of accurately estimating the foreground object opacity in images and video sequences. It serves as a prerequisite for a broad set of applications, including film production and digital image editing. Formally, the input image I is modeled as a linear combination of the foreground and background colors
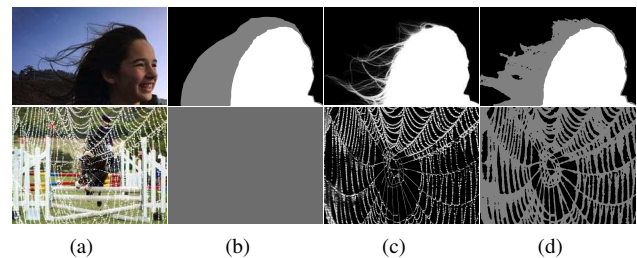

Figure 1. (a) Input image, (b) Input trimap, (c) Our matting results, (d) Corresponding trimap adaptation results. Row 1: the input trimap (b) from low-quality user labelling contains errors. Row 2: the input trimap (b) is a failure case of the Graphcut algorithm where all pixels are marked unknown. In both cases, the proposed method can produce reliable alpha mattes.

as follows [10]:

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i, \ \alpha_i \in [0, 1], \qquad (1)$$

where F_i, B_i and α_i denote the foreground, background color and alpha matte estimation at pixel i respectively. Given an input image I, image matting aims to solve F, B, and α simultaneously. The problem is highly ill-posed, as according to Eq. 1, for an RGB image, 7 values are to be solved but only 3 values are known for each pixel. For most existing matting algorithms, the essential input that constrains the solution space is the trimap, a rough segmentation indicating the opaque and unknown regions. The trimap is generated either interactively by user scribbles, or automatically from binary image segmentation results. In either case, the input trimap is usually coarse, i.e., its unknown region (the gray region in Fig. 1b) contains both real semi-transparent pixels as well as a large amount of opaque ones. This is because providing an accurate trimap is tedious for manual labeling, and is difficult to achieve using existing image segmentation methods that run on low-resolution images.

Unfortunately, previous image matting methods often ignore the inaccuracy of the input trimap, and try to directly

estimate a good alpha matte from it. We argue that there is a classification problem that is not adequately addressed in this process. If we take a closer look at the trimap, pixels in the unknown region fall into three different sets: opaque foreground, opaque background, and the semi-transparent region. We call the first two types *opaque pixels*, and the last type *blended pixels*. The desired behavior of an image matting method is to produce exact 0s or 1s for the opaque pixels, while accurately estimating the fractional opacity (ranged between 0 and 1) for the blended pixels. From this perspective, two related but inherently different tasks are implied in image matting. The first is to classify pixels in the unknown region to iden blended pixel, and we name this task as *trimap adaptation*. The second is to accurately calculate the opacity values for blended pixels, which we call *alpha estimation*.

We observe that these two tasks demand quite different abilities from the algorithm. Trimap adaptation relies more on a good semantic understanding of the object shape and structure, so that it could effectively identify foreground and background regions in the unknown region based on image features. For alpha estimation, careful low-level exploitation of the photometric cues is more critical. Furthermore, trimap adaptation can be modeled as a classification task, and alpha estimation can be viewed as a typical regression task. Most of the existing image matting methods consider image matting as a single regression task, which ignores the classification nature resided in image matting. This observation brings us the question of how to reconcile the two very different aspects of the matting problem in one integrated solution.

Furthermore, existing matting methods, especially optimization-based ones, rely too much on low-level features such as color distributions and local textures, and lack the ability of incorporating high-level semantics. As shown in recent works [38, 2], inducing better understanding towards object shape and structure could help image matting. Although claiming to utilize high-level features, these methods typically rely solely on pretrained features and do not use explicit semantic objective as guidance. This is witnessed by the incomplete object structure extracted by the existing methods from areas where the background color is similar to the foreground object.

Motivated by the new observation that image matting should be disentangled into trimap adaptation and alpha estimation, we propose a simple yet powerful image matting framework named AdaMatting (**Ada**ptation and **Matting**), which resolves the limitations discussed above at the same time. AdaMatting performs trimap adaptation and alpha estimation within two distinct decoder branches in a multi-tasking manner. By explicitly supervising the model to distinguish blended pixels from opaque ones, and then using the refined trimap to afterwards constrain the alpha estima-

tion output, the two branches separately handles the two different aspects of the task. Furthermore, the understanding towards object shape and structure information of the model is greatly enhanced by sharing features from the two tasks. See Fig. 2 for our detailed pipeline.

Our major contributions can be summarized as follows:

- We provide a new perspective that image matting should be disentangled into two tasks, namely *trimap adaptation* and *alpha estimation*, and demonstrate that the disentanglement of the two tasks is essential for improving performance of CNN-based image matting models.

- Following the new perspective, we propose a novel pipeline where trimap adaptation and alpha estimation are jointly optimized in a multi-tasking manner. Extensive experiments show that the proposed pipeline can better use semantic information to provide additional structural awareness and trimap fault-tolerance to the trained CNN model.

- The proposed method refreshes the state-of-the-art results on the most commonly used dataset Adobe Composition-1k [38], and ranks 1st on alphamatting.com [26].

2. Related Work

Natural Image Matting Natural image matting is essentially the per-pixel opacity estimation of the foreground region. The typical input to natural matting algorithms is in the form of *scribbles* [33] or *trimaps* [10], which help reduce the solution space of this ill-posed problem.

Existing traditional methods can be categorized into color sampling based and alpha propagation based methods. Color sampling based methods [10, 12, 16, 11] collect a set of known foreground and background samples to find candidate colors for a given pixel's foreground and background. According to the local smoothness assumption on the image statistics, these sampling colors are supposed to be "close" to the true foreground and background colors. Once the foreground color and background color are determined, we can calculate the corresponding alpha value base on Eq. 1. Following this assumption, various sampling-based methods are proposed, including Bayesian matting [10], shared sampling matting [12], global sampling matting [16], and sparse coding matting [11].

Compared to sampling-based methods, propagation-based approaches [31, 13, 7, 1, 21] avoid matte discontinuities which sampling-based approaches may suffer from. These methods utilize the affinities of neighboring pixels to propagate alpha values from the known regions into unknown ones. A popular approach among these is the closed-form matting [21], which finds globally optimal alpha matte by solving a sparse linear system of equations.

Other propagation-based approaches include Poisson matting [31], random walk matting [13], KNN matting [7] and information-flow matting [1].

Recently, deep learning has shown impressive performance on various computer vision tasks including image matting. Cho et al. [9] proposed an end-to-end architecture named DCNN that utilizes the results of closed-form matting [21] and KNN matting [7] to predict better alpha mattes. Shen et al. [29] proposed a fully automatic matting system for portrait photos based on end-to-end CNNs. Lutz et al. utilized [23] the power of adversarial learning to extract alpha mattes which led to visually pleasing compositions. Wang et al. [35] showed a semantic-level pairwise similarity for propagation based matting can be learned via deep learning mechanism.

Trimap Generation To the best of our knowledge, there are no existing work for trimap adaptation (*i.e.* generating the accurate optimal trimaps). The most related topic is automatic trimap generation. Automatic trimap generation has been an important part for traditional matting methods. Wang et al. [34] used depth information acquired by a time-of-flight range scanner to obtain trimap. Some other algorithms [8, 18] rely on the binary segmentation to obtain the coarse trimaps. [30] used the RGB image feature maps together with morphological dilation to automatically generate trimap, and refined trimap using region growing mechanism. [3] first introduced the Gestalt laws to the matting problem, making more robust trimap generation possible. More recently, [6, 29] utilized neural networks to generate trimaps, greatly improving the matting performances.

Multi-task Learning Multi-task learning is a sub-field of machine learning, in which multiple learning tasks are solved within a single model simultaneously. Compared with training separate models for each task, multi-task learning improves learning efficiency and prediction accuracy for each task by utilizing their inter-relation. In computer vision, there are various exemplars of using multi-task learning, *e.g.* joint object detection and semantic segmentation [15], simultaneous depth estimation and scene parsing [37], and universal network for handle low, middle and high-level vision tasks [20]. Recently, Kendall et al. [19] proposed a general way of combining multiple loss functions to simultaneously learn multiple objectives using homoscedastic task uncertainty. By dynamically adjusting the weights for each objective, their model could obtain superior performance compared to separately trained models.

3. Method

According to the disentangled view of image matting aforementioned, two related but subtly different tasks are implied in image matting, namely the trimap adaptation, a classification task, and the alpha estimation, a regression task. We propose a novel pipeline for image matting, in which the two sub-tasks are solved simultaneously in a multi-task learning manner, and then the final mattes are propagated based on the results of the sub-tasks.

In this section, we first formulate the task of trimap adaptation, and then describe the pipeline and training schemes of our proposed AdaMatting (**Ada**ptation and **Matting**).

3.1. Trimap Adaptation

We begin by formally defining the task of trimap adaptation. Let α_{gt} be the ground truth alpha mattes. The corresponding *optimal trimap* T_{opt} of an image can be naturally defined as:

$$T_{opt}(x,y) = \begin{cases} background & \text{if } \alpha_{gt}(x,y) = 0, \\ unknown & \text{if } 0 < \alpha_{gt}(x,y) < 1, \\ foreground & \text{if } \alpha_{gt}(x,y) = 1, \end{cases} \quad (2)$$

where (x,y) stands for each pixel location on the image. Given an input image conditioned with a trimap (which could be coarse), the trimap adaptation aims to predict the optimal trimap T_{opt}. Intuitively, in trimap adaptation, we separate the semitransparent regions from the opaque foreground and background. This is reminiscent of the semantic segmentation task which also divides the image into discrete parts. From the T_{opt} defined above, the image matting task naturally factors into two steps: (1) deciding if the alpha should be exactly zero, one or neither, (2) computing the exact alpha if the region is considered semitransparent. Note that we do not require the predicted label to be strictly compatible with the input trimap: if the user input contains minor error, we would like our model to correct it.

There are several reasons why separating trimap adaptation and alpha estimation is helpful. First, the two tasks require different training strategies and mode-of-operation of the model. Also, the classification task and the regression task usually desire different loss objectives. Therefore, on one hand, separating the two tasks relieves the burden of the regressor to generate exact zero or one values for the opaque pixels to a great extent. On the other hand, when the exact fractional value of α is hidden, the semantic and structural information of the object is expected to be more eminent for the classifier to exploit. Second, our final results are propagated using the predicted $\tilde{T}$ instead of the coarse input trimap, making our model more robust and fault-tolerant to the coarse input trimap.

Fig. 1 (d) shows examples of trimap adaptation performed by our model. As can be observed, the unknown region in the first input trimap is wide and erroneous, not covering all of the hair due to low-quality labeling. After performing trimap adaptation, the output trimap is not only narrowed but also corrected, resulting in more credible alpha mattes. For the second input image ("cobweb" of the Adobe Composition-1k testing set), the automatic trimap generation algorithm (Graphcut [14] based) fails to provide a meaningful trimap. However, the proposed AdaMatting
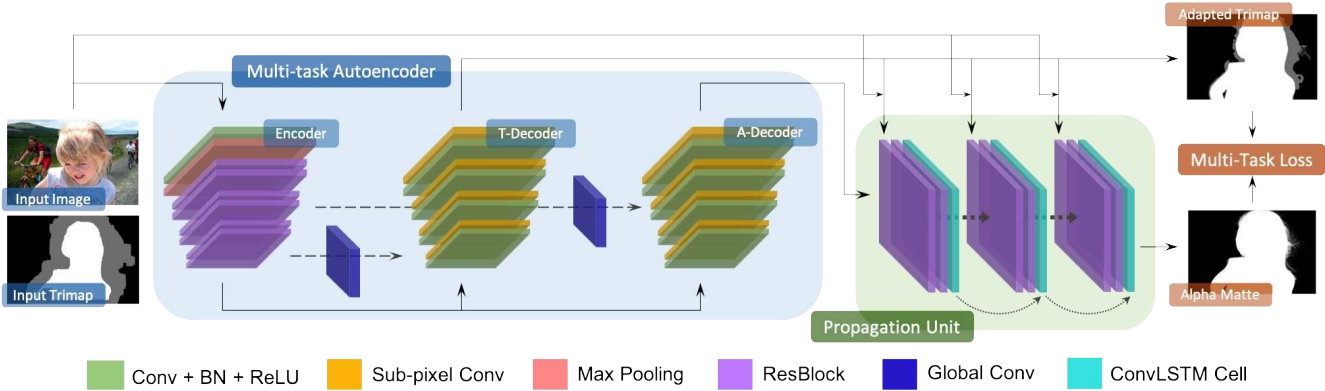
Figure 2. Pipeline of the proposed AdaMatting. T-Decoder and A-Decoder stand for trimap decoder and alpha decoder, respectively. Symmetric shortcuts are linked to different levels of layer for the two decoders.

can surprisingly adapt a rather precise trimap under this extreme condition, thus perfectly solves this hard case of image matting.

3.2. Network Overview

As mentioned above, the trimap adaptation requires more semantic understanding of the object shape and structure, and the image matting relies more on careful low-level exploitation of the photometric cues. Solving these two tasks simultaneously, while sharing intermediate representations, can reasonably enhance the performance of the entire model. Hence we designed a fully end-to-end CNN model named AdaMatting. Fig. 2 depicts the pipeline of AdaMatting, which consists of one encoder producing shared representations, followed by two dependent decoders, solving trimap adaptation and alpha estimation respectively. The result of trimap adaptation and the intermediate alpha matte are then sent into the propagation unit, forming the final alpha matte.

The proposed AdaMatting takes an image concatenated with the corresponding trimap as input. First, a front-end fully convolutional encoder (adopted from the ResNet-50 [17]) produces deep features as shared representations. Then two separate decoders are employed for each task, aiming to learn mappings from the shared representations to the desired output. Each decoder consists of several 3×3 convolutional layers and up-sampling modules. The trimap decoder outputs 3-channel classification logits, guided by the cross-entropy loss. The alpha decoder outputs a 1-channel intermediate alpha estimation, which is forwarded into the propagation unit for further refinement.

Detailed network architecture is depicted in Fig. 2. Here we explain the major modules of our model:

Multi-task Autoencoder The primary module of our pipeline is the multi-task autoencoder, designed based on the widely used U-Net architecture, as it has achieved a great success for numerous computer vision tasks [27, 39, 40]. According to the observation that the trimap adaptation relies more on high-level features and the the alpha
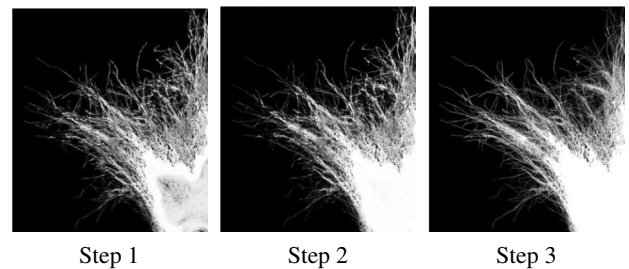

Figure 3. Visualization of the propagated intermediate alpha mattes. Artifacts are gradually removed and the hair progressively becomes clearer and more distinguishable.

estimation relies more on low-level ones, the symmetric shortcuts are linked between different levels of layer for the two decoder. More specifically, the trimap decoder employs deep and middle layer symmetric shortcuts, and the alpha decoder employs middle and shallow layer symmetric shortcuts. Also, recent works [25, 41] show that the size of effective receptive field plays an important role on segmentation tasks. To further enlarge the receptive field while keeping acceptable computational costs, global convolutions [25] are employed on the shortcuts. This modification further enlarges the receptive field, contributing to more reliable and locally consistent results.

Propagation Unit Inspired by the widely used propagation-based approaches [31, 13, 7, 1, 21], we designed a propagation unit empowered by recent advance of the convolutional long short term memory (LSTM) networks [36]. As shown in our pipeline in Fig. 2, the unit is composed of two ResBlocks [17] and a convolutional LSTM cell. In each recurrent iteration, the input image, the adapted trimap, and the previous alpha matte propagation result are taken as input. The ResBlocks extract features from the inputs, while the convolutional LSTM cell keeps memory between the propagation steps.

Similar to the traditional propagation based methods, the propagation unit progressively refines predicted alpha mattes, yielding final results with more accurate edge details

and significantly less undesired artifacts. Fig. 3 illustrates an examples of how alpha matte is refined within the designed propagation unit. As can be seen, the hair progressively becomes more clearly distinguishable. Also, the blurring artifacts are eliminated within the propagation process.

3.3. Multi-task Loss

Multi-task learning aims to solve multiple tasks within one model, while achieving superior efficiency and performance compared to separately trained models. It can be considered as an approach to induce knowledge transfer by sharing the domain information between complementary tasks [32, 19]. From an implementation aspect, by utilizing shared representations and designed objectives, multiple tasks are capable of learning from each other in an effective and efficient way.

Specifically in the AdaMatting, the two tasks are trimap adaptation and alpha estimation. Trimap adaptation, as mentioned above, can be modelled as a segmentation task, splitting the input images into solid foreground, solid background, and semitransparent regions. The process of solving such kind of segmentation problem could lead to rich semantic features, which help solve the alpha matte regression in return.

Instead of a linearly combined loss, we adopt the task uncertainty loss [19]. Our loss can be formulated as:

$$\mathcal{L}(\{\tilde{T}, \tilde{\alpha}\}, \{T_{opt}, \alpha_{gt}\}) = \frac{1}{2\sigma_1^2}\mathcal{L}_T(\tilde{T}, T_{opt})$$
$$+ \frac{1}{\sigma_2}\mathcal{L}_\alpha(\{\tilde{T}, \tilde{\alpha}\}, \alpha_{gt}) + \log 2\sigma_1\sigma_2,$$
$$(3)$$

where $\tilde{T}$ and $\tilde{\alpha}$ stand for the output of trimap adaptation and alpha estimation, σ_1 and σ_2 stand for dynamically adjusted task weights, $\mathcal{L}_T$ and $\mathcal{L}_\alpha$ stand for trimap adaptation loss and alpha estimation loss, respectively. More specifically, $\mathcal{L}_T$ is the cross-entropy loss, and $\mathcal{L}_\alpha$ is the L_1 loss, calculated on the unknown regions of $\tilde{T}$ (denote as $\tilde{T}_u$) only:

$$\mathcal{L}_\alpha(\{\tilde{T}, \tilde{\alpha}\}, \alpha_{gt}) = \frac{1}{|\tilde{T}_u|}\sum_{s\in\tilde{T}_u}|\tilde{\alpha}(s) - \alpha_{gt}(s)|, \quad (4)$$

where $|\tilde{T}_u|$ is the number of pixels in $\tilde{T}_u$. The loss actually disentangles the image matting into two parts, as described in Section 3.1, assuring each decoder to learn structural semantics and photometric information respectively.

Note that the trade-off parameter of the two tasks are dynamically adjusted during training time by the back-propagation algorithm, which avoids the expensive and cumbersome searching process for the optimal weights.

4. Experiments

We conduct extensive experiments and demonstrate the efficacy of our methods on two public datasets: (1) *alphamatting.com* [26] and (2) Adobe Composition-1k [38]

test set. The latter one has a wider range of object types and more complicated background scenes. In this section, we compare our complete AdaMatting with current state-of-the-art methods both quantitatively and qualitatively.

4.1. Experiment Settings

Datasets The first dataset, *alphamatting.com* [26], is a well-known online evaluation benchmark for natural image matting methods. It includes 27 training images and 8 testing images with 3 different kind of trimaps, namely, "small", "large" and "user", representing different input trimap qualities. The second dataset is the *Composition-1k* [38], which provided 431 foreground images as well as their ground truth alpha mattes. 100 background images from COCO [4] are selected for each foreground. We completely followed the composition order used by [38] while using the dataset.

Evaluation Metrics We use four quantitative metrics for matting evaluation. Namely the sum of absolute differences (SAD), mean square error (MSE) and the gradient error (Grad). Empirical studies show that Grad is better suited for perceptual comparisons of matting methods [26].

Implementation Details Inspired by [5], we use the "poly" learning rate policy where current learning rate is defined as the base learning rate multiplied by $(1 - \frac{iter}{max\_iter})^p$. The base learning rate and p is set to 0.0001 and 0.9 respectively. The Adam optimizer is used, with momentum and weight decay set to 0.9 and 0.0001 respectively. The σ_1 and σ_2 are both initialized to 4 in the multi-task loss.

For all experiments, we train for 120 epochs with a batch-size of 16. The training patches sized 800×800 to 320×320 along the unknown regions in the trimap are randomly cropped and then resized to 320×320 patches, as training with larger patches could introduce more semantic information. The training trimaps are generated from the ground truth alpha mattes using the random erode and dilate technique [38]. For data augmentation, we adopt random flip and random resize between 0.75 and 1.5 for all images, and additionally add random rotation between -45 and 45 degrees. Due to the difference between classification and image matting task, the model weights are not initialized by ImageNet pre-training as in [38, 23]. The training data are randomly shuffled in each epoch. The training process takes about 2 days with eight NVIDIA TITAN X GPUs for each experiment. During inference, the full-resolution input images and corresponding trimaps are concatenated as the 4-channel input and fed into the network. The propagation unit recurs for 3 times.

4.2. Results on alphamatting.com

We submitted our result to the alphamatting.com [26]. The AdaMatting achieves state-of-the-art performance, ranking the first for the average performance on all three metrics. The gradient error and the MSE results are shown

Table 1. Average ranking results of our methods and 5 representative state-of-the-art techniques on the alphamatting.com dataset [26]. Best results are shown in bold. S, L, U stand for different type of input trimaps. See alphamatting.com for details.

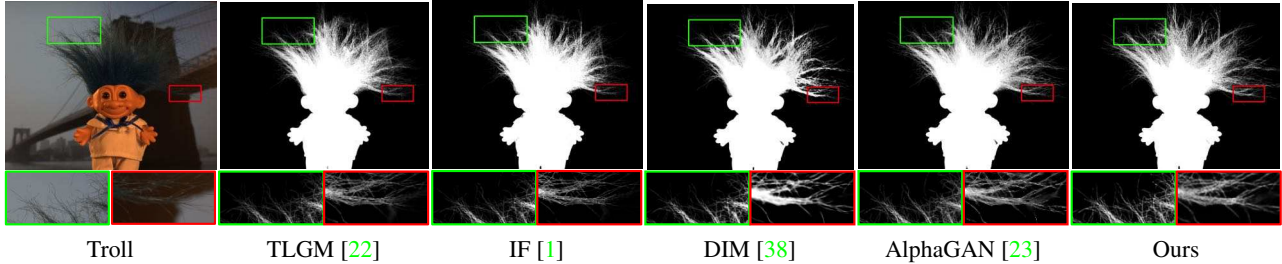| Methods	Gradient Error				MSE				SAD			
	Overall	S	L	U	Overall	S	L	U	Overall	S	L	U
AdaMatting (ours)	**5.2**	**2.8**	**2.8**	**10**	**5.3**	**3.8**	**4.8**	**7.5**	**4.6**	**3.9**	**3.8**	**6.1**
SampleNet Matting	6.2	3.1	3.3	12.1	6.4	4	6.4	8.9	5.3	3.9	4.5	7.4
AlphaGAN [24]	13.2	12	10.8	16.8	14.3	14.8	15.1	13.1	11.2	12	11	10.6
DCNN [9]	14.6	17.9	14.4	11.6	10	11.6	7.9	10.5	10.5	12.5	8.6	10.4
DIM [38]	14.3	10.8	11	21	9.3	8	8	11.9	7.1	8.3	6.1	6.9
IF [1]	16.4	19.5	14.1	15.1	10	12.5	9	8.6	8.8	9.9	8.9	7.5

Figure 4. Qualitative comparisons on two images of alphamatting.com test set. [26] The figure shows the alpha predictions of the test image "troll" with trimap "user".

in Tab. 1.

Several visual comparisons are shown in Fig. 4. As can be seen from the figure, our results contain much more details compared to other state-of-the-arts. Specifically, for the "Troll" test image in the first row, we produce sharper details with less artifacts compared to other models.

4.3. Results on Composition-1k

For the Composition-1k test set, we evaluate 6 recent state-of-the-art methods, namely Closed Form [21], KNN [7], DCNN [9], Information Flow [1], AlphaGAN [24], and Deep Image Matting [38]. The quantitative results under the Grad, SAD and MSE are shown in Table 2. Obviously, our model outperforms all other methods on all metrics by a large margin.

As random backgrounds are selected to combine with each foreground object in the dataset, many images do not seem natural or realistic. Moreover, some particularly difficult images are presented in the data set, in which the foreground color is hard to be distinguished from the background. Two examples are shown in Fig. 5. It can be obviously seen that our results contain more vivid details and significantly less artifacts, compared to all other methods.

5. Discussions

From the comparisons to state-of-the-art image matting models, it is obvious that our AdaMatting achieves superior performance both quantitatively and qualitatively. In this section, we conduct more experiments to further analyze the effectiveness of each designed technique, as well as measuring our performance on unseen real-world images.

Table 2. Quantitative comparisons on the Composition-1k test set with other state-of-the-arts. The gradient loss is scaled by 10^3. -PU stands for removing the Propagation Unit.

Methods	Grad	SAD	MSE
CF [21]	126.9	168.1	0.091
KNN [7]	124.1	175.4	0.103
DCNN [9]	115.1	161.4	0.087
IF [1]	38.0	52.4	0.030
AlphaGAN [24]	-	-	0.031
DIM [29]	30.0	50.4	0.014
Ours (-PU)	17.9	44.1	0.011
Ours	**16.8**	**41.7**	**0.010**

5.1. Comparison to the Two-Stage Method

Rather than training a single network in the multi-tasking manner, a more intuitive method is utilizing two cascaded networks, solving trimap adaptation and then image matting sequentially. We call such model Seq-AdaMatting, which does not share intermediate representations between the two tasks.

For fair comparisons, all components including sub-pixel convolutions, global convolutions, propagation unit and multi-task loss are employed for both models. The quantitative results on Adobe Composition-1k is listed in Table. 3. It can be seen from the table that the original AdaMatting which utilizes the shared representations significantly outperforms the sequential version on all three metrics, despite the fact that the Seq-AdaMatting has much more parameters than the AdaMatting (since the Seq-AdaMatting have two different encoders). These results fur-

Figure 5. Qualitative comparisons on the Adobe Composition-1k test set.

ther prove that using shared representations, which contain rich semantic information learned from trimap adaptation, could effectively help to extract better alpha mattes.

Table 3. Quantitative comparisons to the two-stage sequential method. The gradient loss is scaled by 10^3.

Method	Grad	SAD	MSE
AdaMatting - w/o PU	**17.93**	**44.10**	**0.0114**
Seq-AdaMatting	23.97	46.36	0.0129

5.2. Accuracy of Trimap Adaptation

We have provided visualization examples of adapted trimap in paper and supp. For quantitative results, please refer to Tab. 4.

5.3. Effectiveness of the Structural Semantics

To further prove the effectiveness of leading in structural semantics, we designed an experiment to compare our proposed AdaMatting (with multi-task loss and shared representations) with those trained without the help of semantic information. The first is the model trained only with the alpha estimation loss $\mathcal{L}_\alpha$ (*i.e.* $\sigma = 1.0$ in Eq. 5), not involving

Table 4. Perf. of trimap adaptation (Acc, mIoU) and image matting (Grad) on Adobe's testset[38]. "D-n": Using n-dilation of GT alpha as input trimap. "Adobe": Using testset trimap as input.

Method / Trimap Type	Acc (%)	mIoU	Grad
CSS Matting [28] / D-100	84.9	59.0	480.99
CSS Matting [28] / D-10	92.3	77.8	129.8
CSS Matting [28] / Adobe	90.3	77.2	116.27
AdaMatting / D-100	94.7	80.7	17.68
AdaMatting / D-10	96.7	84.2	17.06
AdaMatting / Adobe	96.5	83.6	16.89

high-level semantics learned from trimap adaptation. The second is the two-stage method mentioned in Section 5.1, which does not share representations between the two tasks. Thus the alpha estimation step is not guided by the semantic features from trimap adaptation as well.

We take a closer look to the second image in Fig. 5, since the ball is highly structured, contains sophisticated patterns on a large scale. Furthermore, the color resemblance between the foreground and background add to the overall difficulty for image matting. Thus solving this image would definitely need the global perception to the object shape and structure. The alpha matte results of the three models are shown in Fig. 7. It can be obviously seen that with the structural semantics learned from trimap adapta-
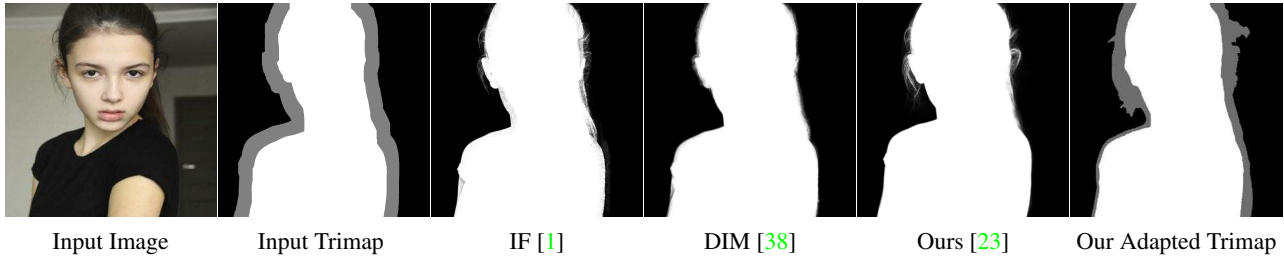
| Input Image | Input Trimap | IF [1] | DIM [38] | Ours [23] | Our Adapted Trimap |

Figure 6. Evaluation results on a real-world image. The input trimap is generated by portrait segmentation followed by boundary eroding.

Input Image   AdaMatting with $\mathcal{L}_\alpha$ Only
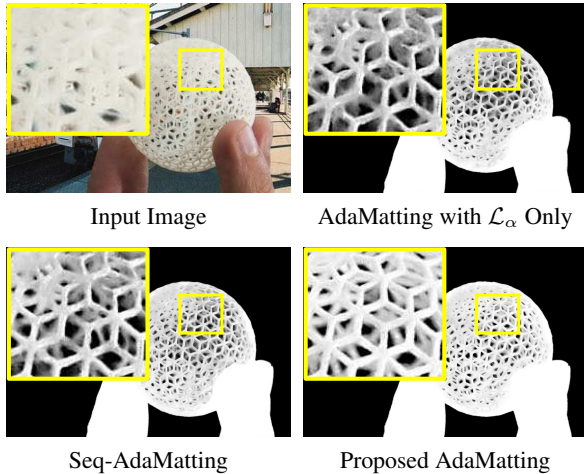
Seq-AdaMatting   Proposed AdaMatting

Figure 7. Comparisons of our model trained with and without structural semantic features from trimap adaptation. Obviously the one with these features (the last alpha mattes) could capture the overall structure, and yield a more accurate result.

tion, the proposed model could precisely capture the overall shape of the foreground object, leading to accurate matting results without loss of details. The other models without high-level features fail to perceive the global structure of the object, resulting in deficiencies in the alpha mattes.

5.4. Analysis on the Multi-Task Loss

To further analysis the impact of the multi-task loss. We carry out experiments on the Adobe Composition-1k testset, comparing two kinds of losses: the deployed dynamically weighted loss (Eq. 3) and naive linearly combined loss:

$$\mathcal{L}_{naive} = (1 - \sigma)\mathcal{L}_T + \sigma\mathcal{L}_\alpha, \qquad (5)$$

where σ stands for a pre-defined fixed weight.

Note that in the extreme case of $\sigma = 1.0$, the method degenerates to the one step regression of alpha used by previous work[38, 9]. We train the AdaMatting under same settings except for the loss function. The resulted model performance with respect to weight σ is shown in Fig. 8.

It can be observed that $\sigma = 1.0$ leads to significantly inferior performance, which verifies the importance of solving trimap adaptation explicitly. Properly adjusting the weight between the classification branch and the regression branch improves performance. However, the dynamically weighted loss leads to markedly better results compared to all other losses.
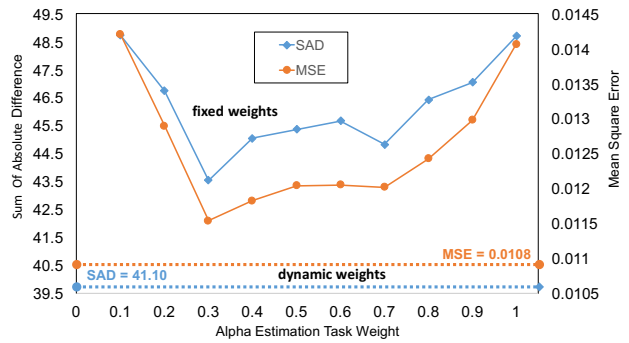

Figure 8. Results on the Adobe Composition-1k test set with different weighted loss functions. The dotted line below represents the result with dynamically weighted loss, and the poly lines above represent the result with linearly combined fixed-weight loss.

5.5. Real-World Image Matting

Because of the additional robustness provided by trimap adaptation, our model could generate accurate alpha mattes even when the input trimaps contain minor errors. The robustness is particularly useful when performing real-world image matting. One of the results is shown in Fig. 6. As observed, our AdaMatting produces much more meticulous details compared to other methods. Furthermore, because of the task of trimap adaptation, our model is capable of correcting the input trimaps, yielding accurate alpha values even at the improperly labelled regions.

6. Conclusions

In this paper, we proposed a disentangled view of image matting, where the task can be divided into two sub-tasks: trimap adaptation and alpha estimation. From this point of view, the AdaMatting is proposed to solve both sub-tasks jointly utilizing the multi-task loss. By explicitly separating the two sub-tasks and optimizing them according to different objectives, the model can greatly benefit from the shared representations, which contains both rich semantic and photometric information. Extensive experiments demonstrate additional structural awareness and trimap fault-tolerance of the AdaMatting. Furthermore, the proposed method shows superior performance on two widely used datasets, both qualitatively and quantitatively, establishing a new state-of-the-art for image matting.

References

[1] Yağız Aksoy, Tunç Ozan Aydın, Marc Pollefeys, and ETH Zürich. Designing effective inter-pixel information flow for natural image matting. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3, 4, 6, 7, 8

[2] Yağiz Aksoy, Tae-Hyun Oh, Sylvain Paris, Marc Pollefeys, and Wojciech Matusik. Semantic soft segmentation. *ACM Transactions on Graphics (TOG)*, 37(4):72, 2018. 2

[3] Ahmad Al-Kabbany and Eric Dubois. A novel framework for automatic trimap generation using the gestalt laws of grouping. In *Visual Information Processing and Communication VI*, volume 9410, page 94100G. International Society for Optics and Photonics, 2015. 3

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 5

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 5

[6] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 618–626. ACM, 2018. 3

[7] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2175–2188, 2013. 2, 3, 4, 6, 7

[8] Donghyeon Cho, Sunyeong Kim, Yu-Wing Tai, and In So Kweon. Automatic trimap generation and consistent matting for light-field images. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1504–1517, 2017. 3

[9] Donghyeon Cho, Yu-Wing Tai, and Inso Kweon. Natural image matting using deep convolutional neural networks. In *European Conference on Computer Vision*, pages 626–643. Springer, 2016. 3, 6, 7, 8

[10] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2001. 1, 2

[11] Xiaoxue Feng, Xiaohui Liang, and Zili Zhang. A cluster sampling method for image matting via sparse coding. In *European Conference on Computer Vision*, pages 204–219. Springer, 2016. 2

[12] Eduardo SL Gastal and Manuel M Oliveira. Shared sampling for real-time alpha matting. In *Computer Graphics Forum*, volume 29, pages 575–584. Wiley Online Library, 2010. 2, 7

[13] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive alpha-matting. In *Proceedings of VIIP*, volume 2005, pages 423–429, 2005. 2, 3, 4

[14] Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. Exact maximum a posteriori estimation for binary images.

[15] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014. 3

[16] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2049–2056. IEEE, 2011. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[18] Chang-Lin Hsieh and Ming-Sui Lee. Automatic trimap generation for digital image matting. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pages 1–5. IEEE, 2013. 3

[19] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 3, 2017. 3, 5

[20] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, volume 2, page 8, 2017. 3

[21] 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. 2, 3, 4, 6, 7

[22] Chao Li, Ping Wang, Xiangyu Zhu, and Huali Pi. Three-layer graph framework with the sumd feature for alpha matting. *Computer Vision and Image Understanding*, 162:34–45, 2017. 6

[23] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. Alphagan: Generative adversarial networks for natural image matting. *arXiv preprint arXiv:1807.10088*, 2018. 3, 5, 6, 8

[24] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. AlphaGAN: Generative adversarial networks for natural image matting. 2018. 6

[25] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. *arXiv preprint arXiv:1703.02719*, 2017. 4

[26] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1826–1833. IEEE, 2009. 2, 5, 6

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[28] Ehsan Shahrian, Deepu Rajan, Brian Price, and Scott Cohen. Improving image matting using comprehensive sam-

Journal of the Royal Statistical Society. Series B (Methodological), pages 271–279, 1989. 3

pling sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 636–643, 2013. 7

[29] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. Deep automatic portrait matting. In *European Conference on Computer Vision*, pages 92–107. Springer, 2016. 3, 6

[30] Sweta Singh, Anand Singh Jalal, and Charul Bhatanagar. Automatic trimap and alpha-matte generation for digital image matting. In *Contemporary Computing (IC3), 2013 Sixth International Conference on*, pages 202–208. IEEE, 2013. 3

[31] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 315–321. ACM, 2004. 2, 3, 4

[32] Jonas Uhrig, Marius Cordts, Uwe Franke, and Thomas Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *German Conference on Pattern Recognition*, pages 14–25. Springer, 2016. 5

[33] Jue Wang and Michael F Cohen. An iterative optimization approach for unified image segmentation and matting. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 936–943. IEEE, 2005. 2

[34] Oliver Wang, Jonathan Finger, Qingxiong Yang, James Davis, and Ruigang Yang. Automatic natural video matting with depth. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference On*, pages 469–472. IEEE, 2007. 3

[35] Yu Wang, Yi Niu, Peiyong Duan, Jianwei Lin, and Yuanjie Zheng. Deep propagation based image matting. In *IJCAI*, pages 999–1006, 2018. 3

[36] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 4

[37] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. *arXiv preprint arXiv:1805.04409*, 2018. 3

[38] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 5, 6, 7, 8

[39] Xiaoshuai Zhang, Yiping Lu, Jiaying Liu, and Bin Dong. Dynamically unfolding recurrent restorer: A moving endpoint control method for image restoration. In *International Conference on Learning Representations*, 2019. 4

[40] Xiaoshuai Zhang, Wenhan Yang, Yueyu Hu, and Jiaying Liu. Dmcnn: Dual-domain multi-scale convolutional neural network for compression artifacts removal. In *IEEE International Conference on Image Processing*, 2018. 4

[41] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014. 4