A Late Fusion CNN for Digital Matting

Yunke Zhang[1], Lixue Gong[1], Lubin Fan[2], Peiran Ren[2], Qixing Huang[3], Hujun Bao[1] and Weiwei Xu*[1]

[1]Zhejiang University [2]Alibaba Group [3]University of Texas at Austin

{yunkezhang, gonglx}@zju.edu.cn, {lubin.flb, peiran.rpr}@alibaba-inc.com, huangqx@cs.uteaxs.edu,{bao, xww}@cad.zju.edu.cn

Abstract

This paper studies the structure of a deep convolutional neural network to predict the foreground alpha matte by taking a single RGB image as input. Our network is fully convolutional with two decoder branches for the foreground and background classification respectively. Then a fusion branch is used to integrate the two classification results which gives rise to alpha values as the soft segmentation result. This design provides more degrees of freedom than a single decoder branch for the network to obtain better alpha values during training. The network can implicitly produce trimaps without user interaction, which is easy to use for novices without expertise in digital matting. Experimental results demonstrate that our network can achieve high-quality alpha mattes for various types of objects and outperform the state-of-the-art CNN-based image matting methods on the human image matting task.

1. Introduction

Digital matting is to accurately extract the foreground object in an image for object-level image composition. It has the advantage of estimating the alpha (opacity) values of the pixels to create an alpha matte so that the foreground object can be correctly abstracted and then composed with a new background image to render a new scene. Formally speaking, we assume that the observed image $\mathbf{I}$ is generated from three underlying images: the foreground image $\mathbf{F}$, the background image $\mathbf{B}$, and the alpha matte α, through the following model:

$$\mathbf{I}_p = \alpha_p \mathbf{F}_p + (1 - \alpha_p)\mathbf{B}_p \quad (1)$$

where p represents a pixel across all images, and the value of $\alpha_p \in [0, 1]$.

A common approach to digital matting [10, 15, 36] proceeds in three steps, namely, (1) learn the foreground and background color models, (2) compute the probabilities of each pixel belonging to the learned models, and (3) obtain the alpha values. To this end, a critical task in digital matting is to determine the pixel alpha values α, which represents a soft segmentation of the image. [20] leverages the spectral clustering to compute α. However, such methods usually rely on user-inputs such as trimaps and scribbles: The trimap separates an image into foreground region, background region and a transition region to cover fuzzy or transparent foreground object boundaries [10], while the scribbles specify sparse pixels on the foreground and background [35]. Early works exploit the local color as the main feature, which may lead to blurred or chunky artifacts as shown in [39]. Recent works (e.g., [9, 31, 38, 39]) leverage fully convolutional neural network (CNN) to learn multi-scale features, which lead to high-quality semantic image segmentation results. In addition, deep image matting (DIM) [39] has shown that high-quality alpha mattes can be directly predicted through a deep CNN trained on a large-scale image matting dataset. A recent contribution combines the multi-scale features learned by deep CNN and

Image Our alpha matte Composition

Figure 1. Two example matting results of our late fusion CNN that does not need trimaps as input. Left: two images collected from internet outside of our training dataset. Middle: the alpha mattes predicted by our network. Right: the composition results.

*Corresponding author. The authors from Zhejiang University are affiliated with the State Key Lab of CAD&CG.

the spectral matting method to obtain alpha matte [2]. It is fully automatic but has the disadvantage of slow performance due to solving the large-scale spectral problem.

While existing deep learning based digital matting approaches rely on a trimap as input, we propose a fully convolutional network (FCN) for automatic image matting by taking a single RGB image as input. We achieve this goal by designing two decoder branches in the network for the foreground and background classification, and then use a fusion branch to integrate the two classification results, which gives rise to the soft segmentation result. This design provides more degrees of freedom than a single decoder branch for the network to obtain better alpha values. It is based on the observation that the classification branches can well predict the hard segmentation result, but have difficulties in predicting precise probabilities as the alpha values at the pixel level. The two-decoder branch structure allows us to design a fusion branch to correct the residuals left in the classification branches. Moreover, our training loss encourages the two decoder branches to agree with each other at the hard segmentation part and leave the soft segmentation part to be corrected by the fusion branch. Therefore, our approach can implicitly produce trimap without any user interaction, which is easy to use for novice users without expertise in digital matting.

Our two-branch network structure at the decoder stage of FCN follows the late fusion structure which is widely used in deep learning [8, 26] and can be categorized as a type of ensemble learning to improve the accuracy of the predicated alpha values. However, instead of simply maximizing or averaging the output of the two classification branches, we learn the fusion weights. We thus denote our network by late fusion CNN hereafter.

We have evaluated our network on the image matting dataset in [39] to shown that it can produce high-quality matting results for different types of objects. In addition, we also construct a human image matting dataset to test the network on this specific type of images. Fig. 1 illustrates the matting results of our network on two internet images, and the experimental results show that our network that does not need trimaps as input can still achieve comparable results to the state-of-the-art CNN-based methods and outperform on the human image matting task.

2. Related Work

In this section, we briefly review three main approaches, such as sampling-based, affinity-based, and deep learning based approaches, to digital image matting.

Sampling-based approaches [10, 13, 14, 16, 29] use the color of sampled pixels to infer the alpha values of the pixels in the transition region in an image. The key tasks of these approaches are (1) to collect the sampled pixels [13, 16, 29, 30], and (2) to build a foreground and background color model from the sampled pixels [10, 16, 34, 35]. These approaches take the advantage of natural image statistics for solving the ill-posed matting problem and work well when the trimap is carefully defined so that there are strong correlations between the pixel color distribution of the transition region and that of the foreground/background regions. Affinity-based approaches [1, 2, 3, 7, 15, 19, 20, 33] propagate the alpha values of the known foreground and background pixels to the unknown regions and have proven to be more robust than sampling-based approaches when dealing with complex images [10, 14, 29]. The quality of generated alpha mattes using these approaches is highly related to the defined affinity score [15, 19, 33]. Global optimization strategies, such as spectral techniques [20], are continuous relaxations of binary optimization techniques, which is not guaranteed to obtain optimal solutions. For a comprehensive survey of traditional approaches, we refer the readers to [37] for more details.

Deep learning based matting approaches directly learn a mapping from an input image to its alpha matte from large-scale labeled results. Cho et al. [9] proposed an end-to-end CNN by combing the closed-form matting formulation described in [19] and the methodology of KNN mating [7]. Xu et al. [39] integrated an encoder-decoder network and a subsequent detail refinement network for digital matting, which takes an image and the corresponding trimap as inputs. Lutz et al. [25] presented a generative adversarial network for image matting. They improved the decoder structure of [39] by adding the atrous spatial pyramid pooling module [5] to resample the features at several scales. Wang et al. [38] proposed a deep propagation based image matting framework by learning an alpha matte propagation principle using a deep neural network. However, these techniques require a trimap as input to initialize the propagation process. Several recent techniques study image matting for a specific type of objects. Chen et al. [6] proposed an automatic approach for human matting. It takes an RGB image as input and first predicts the foreground and background regions as well as the transition region using the three-class segmentation network. The segmentation result is then used as a trimap for the alpha matte generation. In contrast, our approach generates the final alpha matte by blending the foreground and background probability maps using a fusion network, which avoids the difficult trimap generation problem. The CNN-based portrait matting in [31] uses an average mask as a trimap by assuming the upper body appears at similar positions in portrait images. However, this assumption does not apply in our setting. Zhu et al. [41] followed the similar pipeline while designing a smaller network and a fast filter similar to the guided filter for matting to deploy the model on mobile phones. Chen et al. [4] formulates transparent object matting as reflective flow estimation and leverages a multi-scale encoder-decoder network for prediction.

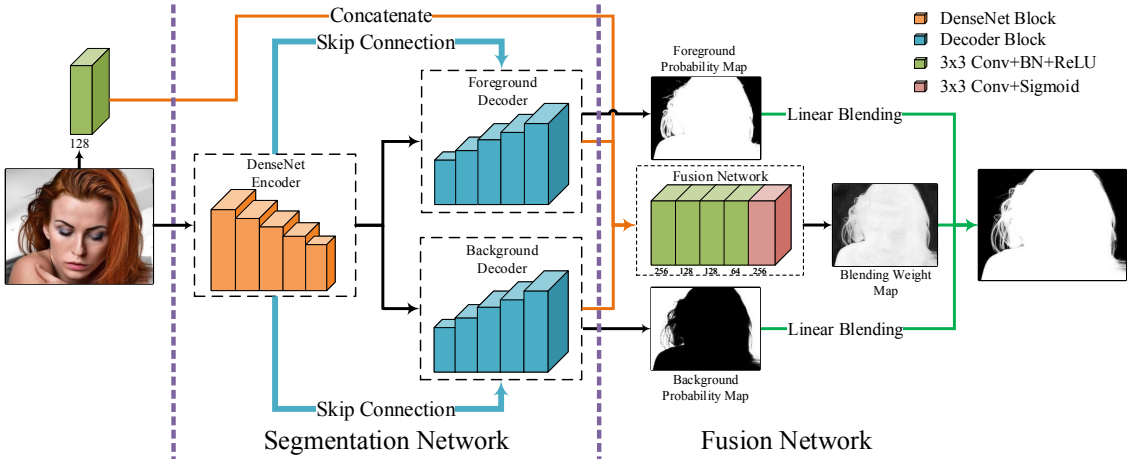


Figure 2. A high-level visualization of our network architecture. The segmentation network consists of one encoder and two decoders. The fusion network is a fully convolutional network without downsampling. The final alpha matte is a linear blending using the outputs of two networks. The number below the block in the fusion network denotes the number of output channels of different convolution layers.

3. Approach

In this section, we introduce the technical details of our approach. We begin with the approach overview in Sec. 3.1. We then elaborate the structure and training loss of segmentation and fusion networks in Sec. 3.2 and 3.3. Finally, we give the training details of our network in Sec. 3.4.

3.1. Approach Overview

We introduce a novel end-to-end neural network that takes an image containing a foreground object as input and outputs an alpha matte of the foreground object.[1] As illustrated in Fig. 2, the key idea of our approach is to use neural network modules to predict three maps, namely, the foreground probability map, the background probability map, and the blending weight map. The output alpha matte is given by using the blending weight map to interpolate the foreground/background probability maps. The network is trained over three consecutive steps: segmentation network pre-training step, fusion network pre-training step and finally end-to-end joint training step whose training loss is imposed on the output alpha matte.

Formally speaking, we try to predict the alpha values with the following fusion formula:

$$\alpha_p = \beta_p \bar{\mathbf{F}}_p + (1 - \beta_p)(1 - \bar{\mathbf{B}}_p), \tag{2}$$

where $\bar{\mathbf{F}}_p$ and $\bar{\mathbf{B}}_p$ represent the predicted foreground and background probability at pixel p. β_p is the blending weight predicted by the fusion network. In our implementation, the fusion network takes the input image and the features before the logistic regression of foreground and background classification branches as input (see Fig. 2).

[1]Please see the supplementary material for the network details at https://github.com/yunkezhang/FusionMatting.

From the optimization perspective, the derivative of α_p with respect to β_p vanishes when

$$\bar{\mathbf{B}}_p + \bar{\mathbf{F}}_p = 1. \tag{3}$$

The advantages of Eq. 2 are two-fold. First, the fusion network will focus on learning the transition region from the foreground to the background if the predictions of the foreground/background probability maps are accurate (meaning Eq. 3 is satisfied), which is the bottleneck for solving the matting problem. Second, we can carefully design the loss function to encourage that the $\bar{\mathbf{F}}_p + \bar{\mathbf{B}}_p \neq 1$ within the transition region (see Sec. 3.2), which can provide useful gradients to train the fusion network.

3.2. Segmentation Network

We proceed to describe the architecture of the segmentation network and its training loss. In particular, the training loss favors 0 or 1 probability of solid foreground and background regions. It also tries to predict the upper bound and lower bound of the true alpha values in the transition region.

Network structure. The segmentation network consists of one encoder and two decoders. The encoder extracts semantic features from the input image. The two decoders share the same encoded bottleneck and predict the foreground and background probability maps, respectively. Specifically, we use DenseNet-201 [18] without the fully-connected layer head as our encoder. Each branch consists of five decoder blocks which correspond to the five encoder blocks, and the decoder block follows the design of feature pyramid network structure in [22]. To enhance the pixel-level segmentation result, we employ the skip connections in [28] to concatenate the multi-scale features from the encoder block (right before the average down-sampling) with the features upsampled through deconvolution layer.

Training loss. The training loss combines the L1 loss, the L2 loss, and the cross-entropy loss. In particular, we control the behavior of the training process of our network by setting different weights for different pixels according to the alpha matte.

We first measure the difference between the predicted probability values and the ground truth alpha values:

$$L_d(\bar{\mathbf{F}}_p) = \begin{cases} |\bar{\mathbf{F}}_p - \alpha_p|, & 0 < \alpha_p < 1. \\ (\bar{\mathbf{F}}_p - \alpha_p)^2, & \alpha_p = 0, 1. \end{cases} \quad (4)$$

The difference is chosen to be L1 inside transition regions so as to recover the details of the alpha matte there, while the L2 loss is used in the rest of the regions to penalize the possible segmentation error. We find this setting can well balance between the soft segmentation and the hard segmentation.

We also introduce the L1 loss on the gradients of the predicted alpha matte since it is beneficial to remove the over-blurred alpha matte after classification:

$$L_g(\bar{\mathbf{F}}_p) = |\nabla_x(\bar{\mathbf{F}}_p) - \nabla_x(\alpha_p)| + |\nabla_y(\bar{\mathbf{F}}_p) - \nabla_y(\alpha_p)|. \quad (5)$$

The cross-entropy (CE) loss for the foreground classification branch at a pixel p is given by:

$$CE(\bar{\mathbf{F}}_p) = w_p \cdot (-\hat{\alpha}_p \log(\bar{\mathbf{F}}_p) - (1 - \hat{\alpha}_p) \log(1 - \bar{\mathbf{F}}_p)), \quad (6)$$

The weight w_p is set to 1 when $\alpha_p = 1$ or 0 and set to 0.5 when α_p is in $(0, 1)$. We let $\hat{\alpha}_p$ be 1 inside both the foreground and the transition regions (0 inside background region) so that the cross-entropy loss encourages the segmentation network to output probability value towards 1 for an upper bound. However, it does not provide useful gradients within the transition region. We thus adopt a small weight in the transition region and combine it with the L1 and L2 loss below to obtain a preliminary alpha matte.

The final loss function of the foreground classification branch with respect to an image is:

$$L_F = \sum_p CE(\bar{\mathbf{F}}_p) + L_d(\bar{\mathbf{F}}_p) + L_g(\bar{\mathbf{F}}_p). \quad (7)$$

For the background classification branch, its loss L_B can be simply computed by setting $\alpha_p = 1 - \alpha_p$ in Eq. 1, 4 and 5. We also impose the L_F and L_B loss at each decoder block of two branches to further regulate the behavior of the network, similar to the side loss used in [24].

Note that the combination of the cross-entropy and the L1 loss inside the transition regions tries to give larger probabilities than the ground truth values since the cross-entropy loss will drag the probabilities to 1. Thus, the true alpha values can be bracketed in the interval formed by the two probabilities predicted by the two branches, since the $1 - \bar{\mathbf{B}}_p$ in Eq. 2 should be less than the α_p in our setting. This design enables us to regress for the precise alpha values after applying the fusion network. Moreover, enforcing the foreground and background segmentation branches to be trained with different losses helps to learn different features of the input image. These characteristics benefit the result of ensemble learning. As illustrated in Fig. 3 and Fig. 4, this design of the segmentation loss does lead to the generation of meaningful implicit trimaps. Moreover, the alpha values between 0 and 1 are mostly bracketed by the two predicted probabilities.

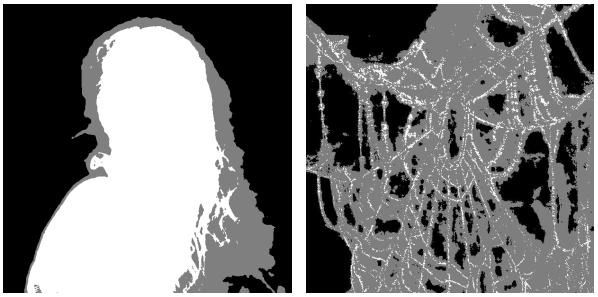


Figure 3. Implicit trimaps predicted by our network for two images in Fig. 1. The implicit transition regions are indicated by gray pixels where the predicted foreground/background probabilities are less than 1.

3.3. Fusion Network

The goal of the fusion network is to output β_p at pixels to fuse the foreground and background classification results.

Network structure. It is a fully convolutional network with five convolution layers and one sigmoid layer to compute the blending weights β_p (see Fig. 2). The input of the network consists of (1) the feature maps from the last block of the foreground and background decoders; (2) the feature from the convolution with the input RGB image. We set the size of convolution kernel to 3×3 according to the experiments and found that the fusion network with this kernel size can better produce the details of the alpha matte.

Training loss. Assuming that the foreground and background decoders already provide reasonable segmentation results for the solid pixels, we design the training loss to lean towards pixels in the transition region. The loss function of the fusion network can be directly derived according to Eq. 2:

$$L_u = \sum_p \mathbf{w}_p \cdot |\beta_p \bar{\mathbf{F}}_p + (1 - \beta_p)(1 - \bar{\mathbf{B}}_p) - \alpha_p|. \quad (8)$$

Specifically, the weights of pixels w_p are set to 1 whenever $0 < \alpha_p < 1$, and 0.1 otherwise.

3.4. Training Details

We use DenseNet-201 network pre-trained with ImageNet-1K [11] as our encoder backbone. We first perform the segmentation network pre-training for 15 epochs. In the fusion network pre-training step, we freeze

Figure 4. The bracketed alpha values after segmentation. Left: an input image. Middle: the ground truth alpha matte. Right: the groundtruth alpha values of the red highlighted pixels are bracketed by two probabilities $\bar{\mathbf{F}}_p$ and $1 - \bar{\mathbf{B}}_p$ which are outputted by our two decoder branches.

the segmentation stage and train the fusion stage alone for 4 epochs. Finally, we perform the end-to-end joint training for 7 epochs, which back-propagates the gradient of the fusion result to both the segmentation and fusion network to further reduce the training loss. All batch normalization layers are frozen in the joint training step to save the memory footprint. Cyclical learning rate strategy [32] is used to accelerate the convergence speed during the whole training procedure. The base learning rate is 5.0×10^{-4} for all steps. The maximum learning rate in pre-training steps is 1.5×10^{-3}. A smaller maximum learning rate 1.0×10^{-3} is set during the joint training steps.

We also use a special loss while performing the end-to-end joint training for fine-tuning the whole network. The loss is based on the loss of the fusion network while adding the loss of the segmentation network to avoid overfitting. The overall joint training loss is described as following:

$$L_J = L_u + w_1(L_F + L_B) + w_2 L_s. \tag{9}$$

We set $w_1 = 0.5$ and $w_2 = 0.01$ in our implementation. The third term L_s is directly adopted from [20] to penalize the amount of soft segmentation pixels, i.e.:

$$L_s = \sum_p \alpha_p^\gamma + (1 - \alpha_p)^\gamma, \ \gamma \in [0, 1]. \tag{10}$$

where γ is set to 0.9 in our experiments.

4. Experimental Results

In this section, we evaluate our late fusion CNN on two testing datasets. (1) Human image matting testing dataset, which is to measure the performance of our method on a specific task. To this end, we collect 40 human images in which 29 are from the internet whose alpha mattes are carefully matted by designers and 11 are from composition-1k testing dataset in [39] due to their abundant details.

Methods	SAD	MSE	Gradient	Connectivity
Shared Matting [19]	16.54	0.022	30.85	15.75
Comprehensive [30]	13.31	0.014	18.92	11.80
Learning Based [27]	15.80	0.020	25.04	13.77
Global Matting [7]	27.47	0.029	33.76	24.98
Closed-form [40]	15.92	0.021	25.71	13.87
KNN Matting [14]	18.27	0.023	25.11	16.88
DCNN [9]	14.92	0.017	21.56	13.02
SHM [6]	13.34	0.017	24.41	12.71
DIM [39]	10.39	0.014	19.20	9.64
Ours-FG/BG-Only-25	20.93	0.033	44.01	20.34
Ours-Fusion-Only-25	14.23	0.019	24.46	13.26
Ours-raw-25	10.08	0.010	15.57	9.24
Ours-refined-25	9.75	0.010	15.60	8.96
Ours-raw-full	10.87	0.002	16.91	9.80
Ours-refined-full	10.49	0.002	16.97	9.52

Table 1. The quantitative results of our human image matting testing dataset. Ours-FG/BG-only: pre-trained segmentation network stage. Ours-Fusion-only: pre-trained fusion network stage. Ours-raw: end-to-end jointly trained network. Ours-refined: refined by guided filter [17]. "-25": computed in the transition regions generated by 25 pixels dilation. "-full": computed over the whole image.

We compose each testing image with 25 random background images from PASCAL VOC [12] to form a testing dataset with 1000 images. The training dataset for this task is independent on the testing images, which consists of 228 human images with high-quality alpha mattes combined with another 211 human foreground objects from the DIM dataset [39]. Similarly, we compose these foregrounds with randomly picked unique background images from MS-COCO [23] to form the final dataset, totally 28610 images for training. (2) Composition-1k testing dataset in [39], which is to evaluate how our network performs on natural images. This testing dataset contains 1000 images, composed of 50 unique foregrounds and 20 background images. For this evaluation, we train our network on the DIM dataset [39] independent on the testing images. It consists of 431 unique foreground objects with alpha mattes. Each object is composed of 100 background images randomly picked from MS-COCO.

For the data augmentation during the training process, we crop the image and trimap pairs centered on random pixels in the transition regions indicated by trimaps. The crop sizes are selected to be 512×512 and 800×800. We also resize all the training images to the size of 512×512 to warm-up the network. Random flipping and rotation are applied to all the cropped and resized training data. Due to the memory limit, we require the longer side of the image to be less than 800 in the training. This size constraint is also imposed during inference. The training time of our network on a GPU server (configuration: E5-2682 CPU, 32G RAM, and 8 Tesla P100 graphics cards) is 2.5 days for human image matting dataset and 4 days for the DIM dataset. For

Image Trimap SSS [2] DIM [39] Ours GT

Figure 5. The visual comparisons on human image matting testing dataset. The segments in SSS [2] are hand-picked.

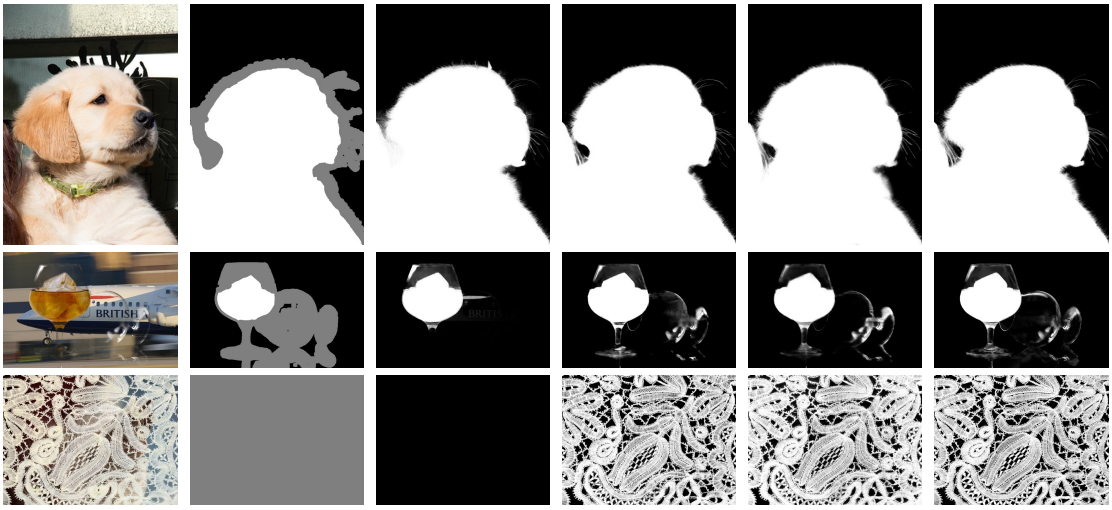


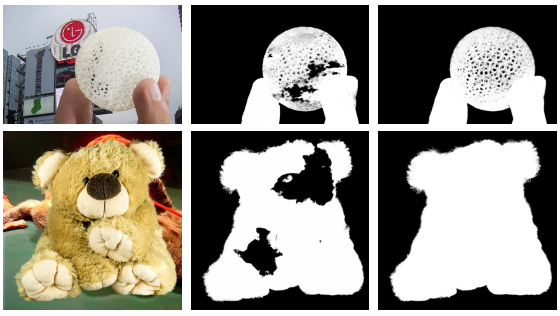
Image Trimap Closed-form [19] DIM [39] Ours GT

Figure 6. The visual comparisons on the composition-1k testing dataset.

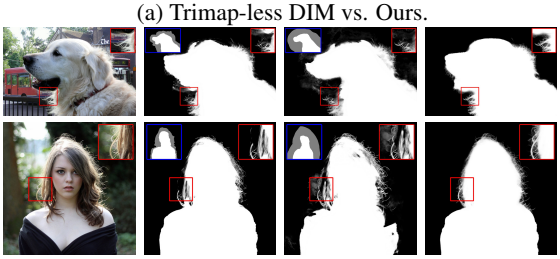
testing, the average running time on a 800×800 image is 0.39 seconds.

Evaluation metrics. There are four metrics used in the evaluations: SAD (sum of absolution difference), MSE (mean square error), gradient and connectivity defined in [39] . The lower values of the metrics, the better the predicted alpha matte is. The details of gradient and connectivity metrics can be found in [27], and they are used to reflect the visual quality of the alpha matte when observed by a human. For the computation of all the metrics, after summing the metrics at each pixel p for a testing image, we then compute their average over all the images in the testing dataset.

Evaluation on human image matting testing dataset. To compare our network with the state-of-the-art image matting methods, we also train the DIM network on this dataset by feeding both RGB images and trimaps generated by random dilation at pixels whose alpha values are neither 0 nor 1, while the transition regions of the trimaps used for the metric computation are generated by 25 pixels dilation. Since narrowing down the image type to human image lowers the difficulty of the segmentation, our network can closely match the DIM network w.r.t. different metrics as reported in Tab. 1. If only computing the metrics in the trimap transition regions as [39], our method outperforms the DIM network in all four metrics (see "ours-raw-



(a) Trimap-less DIM vs. Ours.



(b) Our network and DIM using different trimaps.

Figure 7. Comparisons to DIM.Top-left in (b): manually specified trimaps. 'small' and 'large' indicate the size of the trimap.

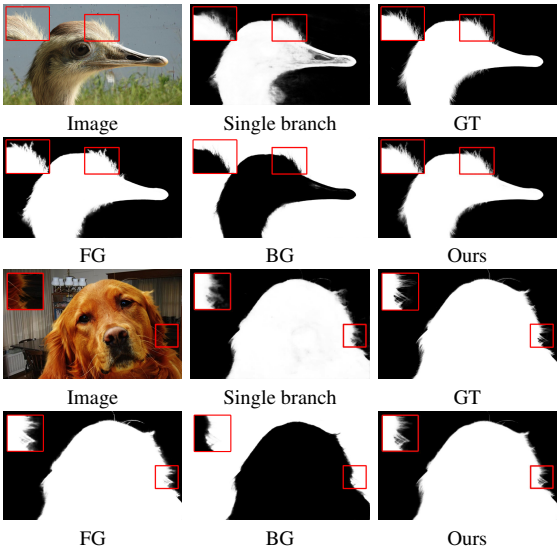


Figure 8. Self-comparisons. Single branch: our foreground branch plus the DIM refinement network trained with L1 loss. 'FG' and 'BG': our foreground and background probability maps.

25" and "ours-refined-25" in Tab. 1). After computing the four metrics on the whole image, the metrics of our algorithm get increased slightly indicating that the segmentation error is well controlled in this case (see "ours-raw-full" and "ours-refine-full" in Tab. 1). The "ours-FG/BG-Only-25" and "ours-Fusion-Only-25" also verify that the matting results get improved as we gradually add each sub-network into late fusion CNN. Fig. 5 illustrates three selected matting results in the testing images. Note that our network works well for various poses and scales of the human in the foreground. For instance, the woman viewed from the back (second row in Fig. 5) is difficult for the deep automatic portrait matting [31].

Methods	SAD	MSE	Gradient	Connectivity
Shared Matting [19]	115.20	0.074	139.88	121.35
Comprehensive [30]	109.80	0.066	116.27	107.86
Learning Based [27]	100.51	0.058	94.68	104.74
Global Matting [7]	121.46	0.078	125.11	133.23
Closed-form [40]	121.18	0.076	130.63	120.16
KNN Matting [14]	133.99	0.098	140.29	134.03
DCNN [9]	122.40	0.079	129.57	121.80
DIM-Trimap-less-25	70.31	0.110	70.06	70.05
DIM [39]	33.64	0.017	30.23	31.92
Ours-FG/BG-Only-25	103.21	0.077	91.85	109.27
Ours-Fusion-Only-25	66.05	0.034	69.80	69.80
Ours-raw-25	49.05	0.022	36.58	50.70
Ours-refined-25	49.02	0.020	34.33	50.60
Ours-raw-full	58.34	0.011	41.63	59.74
Ours-refined-full	58.29	0.011	36.58	59.63

Table 2. The quantitative results on the Composition-1k testing dataset. The metrics measured on our results are same with Tab. 1. 'DIM-Trimap-less-25' denotes the results of the DIM method without trimap as input during training.

Evaluation on composition-1k testing dataset. Fig. 6 shows three qualitative results and visual comparisons on this dataset. It can be observed that our results are comparable to the results of the DIM [39] even in challenging lace image case. The corresponding metrics are reported in Tab. 2. Due to the image size constraint imposed in the training of late fusion CNN, we also compute the metrics of the DIM network on the resized testing images for a consistent comparison. We first compute the four metrics inside the transition regions in the trimaps provided in the testing dataset, a similar strategy adopted in [39]. It is easy to observe that our method surpasses all the non-CNN image matting methods on this dataset by a large margin since our network can exploit multi-scale features to better understand the semantics in the image. Comparing to CNN-based method, our network is better than DCNN but still inferior to DIM. It is as expected since DIM requests a much stronger input compared to our setting. Specifically, the trimap fed into the DIM network can avoid the possible segmentation errors in our case. After computing the metrics over the complete dataset, our results still rank No.2.

To further verify whether or not the refinement network used in [39] can correct the residual left in single classification branch results, we train the DIM network without input trimap channel as a comparison and report the result of this setup in the row of "DIM-Trimap-less-25" in Tab. 2. The qualitative comparison is illustrated in Fig. 7.a. Fig.7.b shows two additional qualitative comparisons where the DIM network is fed with the manually specified trimaps. It demonstrates that the quality of the matting results of the DIM network downgrades as the size of the transition region

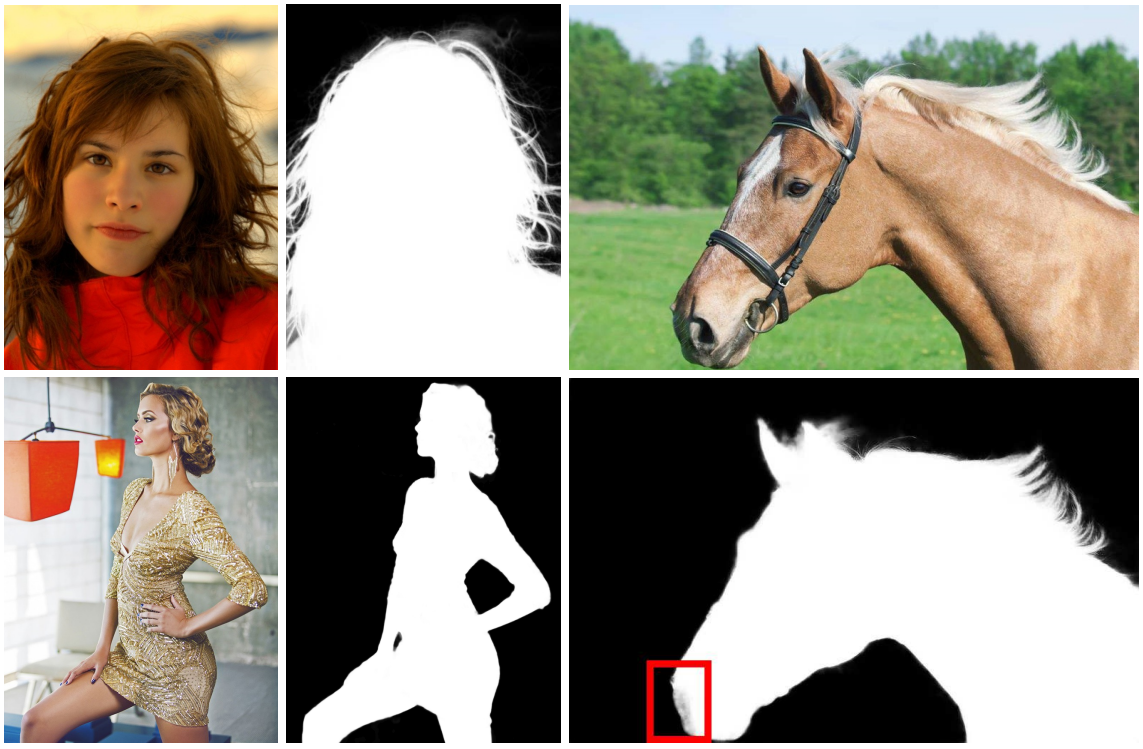


Figure 9. Internet image matting results.

increases. Therefore, it is important to have image matting algorithms robust to the trimap quality.

Self-comparisons. The two-branch design provides three degrees of freedoms, which allows the optimizer to balance among them for better results. The single branch network in Fig. 8 is created by discarding the background and fusion branch. Similar to the DIM method, we also add a fully convolutional network as refinement and use L1 loss only during training. Its results contain segmentation errors, which are removed by the two-branch network, as illustrated in Fig. 8. In contrast, the foreground and background probability maps of our method are more 'solid' in the non-transition regions since our segmentation loss there favors a hard segmentation. The final results of our late-fusion CNN demonstrate that our fusion network is able to fuse the foreground and background probability maps for detailed alpha mattes (see the second and fourth row of Fig. 8).

Evaluation on internet images. Fig. 1 and Fig. 9 shows the matting results for collected internet images to test the generalization ability of our method.[2] All the human image matting results are obtained through our network trained with human image matting dataset, and the other results are from the late fusion CNN trained with the DIM dataset. The results prove that our network has the ability to capture transition regions of different types of various objects. However, the difficulty in capturing the semantic feature of the foreground can possibly lead to segmentation errors in our results, for example, the error around the mouth of the horse as shown in the bottom row of Fig. 9.

[2]Please see the supplementary material for more matting results.

5. Conclusions and Future Work

In this paper, we proposed a late fusion fully convolutional neural network for image matting. It utilizes two decoder branches for foreground/background classification and fuses the classification results to obtain the final alpha values through a fusion network. The network does not need trimap as input, which greatly improves the efficiency of the image matting.

In the future, we would like to explore how to improve the decoder network structure to further reduce segmentation errors. The recent development of multi-scale feature fusion network, such as Refine-net [21], can be tested in the late fusion CNN. It is also interesting to explore how to apply the two-branch design to video object matting.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. Weiwei Xu is partially supported by NSFC (No. 61732016) and Zhejiang Lab. Qixing Huang would like to thank for the gift from snap research. Weiwei Xu and Hujun Bao are also supported by the Fundamental Research Funds for the Central Universities.

References

[1] Y. Aksoy, T. O. Aydin, and M. Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 228–236, 2017. 2

[2] Y. Aksoy, T.-H. Oh, S. Paris, M. Pollefeys, and W. Matusik. Semantic soft segmentation. *ACM Transactions on Graphics (TOG)*, 37(4):72, 2018. 2, 6

[3] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. *International Journal of Computer Vision (IJCV)*, 82(2):113–132, 2009. 2

[4] G. Chen, K. Han, and K.-Y. K. Wong. Tom-net: Learning transparent object matting from a single image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[5] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR, abs/1706.05587*, 2017. 2

[6] Q. Chen, T. Ge, Y. Xu, Z. Zhang, X. Yang, and K. Gai. Semantic human matting. *arXiv preprint arXiv:1809.01354*, 2018. 2, 5

[7] Q. Chen, D. Li, and C.-K. Tang. Knn matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2175–2188, 2013. 2, 5, 7

[8] J. Cheng, B. Aurlien, and v. d. L. Mark J. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *CoRR*, abs/1704.01664, 2017. 2

[9] D. Cho, Y.-W. Tai, and I. Kweon. Natural image matting using deep convolutional neural networks. In *The European Conference on Computer Vision (ECCV)*, pages 626–643. Springer, 2016. 1, 2, 5, 7

[10] Y.-Y. Chuang, B. Curless, D. Salesin, and R. 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] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009. 4

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, June 2010. 5

[13] X. Feng, X. Liang, and Z. Zhang. *A Cluster Sampling Method for Image Matting via Sparse Coding*. Springer International Publishing, 2016. 2

[14] E. S. Gastal and M. M. Oliveira. Shared sampling for real-time alpha matting. In *Computer Graphics Forum*, volume 29, pages 575–584. Wiley Online Library, 2010. 2, 5, 7

[15] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann. Random walks for interactive alpha-matting. In *Proceedings of VIIP*, volume 2005, pages 423–429, 2005. 1, 2

[16] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun. A global sampling method for alpha matting. 2011. 2

[17] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, (6):1397–1409, 2013. 5

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[19] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Computer Society Conference on*, volume 1, pages 61–68. IEEE, 2006. 2, 5, 6, 7

[20] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1699–1712, 2008. 1, 2, 5

[21] G. Lin, A. Milan, C. Shen, and I. D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. *CoRR*, abs/1611.06612, 2016. 8

[22] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 3

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ICCV)*, pages 740–755. Springer, 2014. 5

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 4

[25] S. Lutz, K. Amplianitis, and A. Smolic. Alphagan: Generative adversarial networks for natural image matting. *arXiv preprint arXiv:1807.10088*, 2018. 2

[26] Y. Ren, L. Zhang, and P. N. Suganthan. Ensemble classification and regression-recent developments, applications and future directions. *IEEE Comp. Int. Mag.*, 11(1):41–53, 2016. 2

[27] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1826–1833. IEEE, 2009. 5, 6, 7

[28] O. Ronneberger, P. Fischer, and T. 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. 3

[29] M. A. Ruzon and C. Tomasi. Alpha estimation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1018. IEEE, 2000. 2

[30] E. Shahrian, D. Rajan, B. Price, and S. Cohen. Improving image matting using comprehensive sampling sets. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 636–643, 2013. 2, 5, 7

[31] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia. Deep automatic portrait matting. In *European Conference on Computer Vision (ECCV)*, pages 92–107. Springer, 2016. 1, 2, 7

[32] L. N. Smith. Cyclical learning rates for training neural networks. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 464–472. IEEE, 2017. 5

[33] J. Sun, J. Jia, C. K. Tang, and H. Y. Shum. Poisson matting. *ACM Transactions on Graphics*, 23(3):315–321, 2004. 2

[34] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: an interactive tool for realtime high quality matting. In *ACM SIGGRAPH*, page 9, 2007. 2

[35] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Tenth IEEE International Conference on Computer Vision*, pages 936–943, 2005. 1, 2

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

[37] J. Wang, M. F. Cohen, et al. Image and video matting: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(2):97–175, 2008. 2

[38] Y. Wang, Y. Niu, P. Duan, J. Lin, and Y. Zheng. Deep propagation based image matting. In *IJCAI*, pages 999–1006, 2018. 1, 2

[39] N. Xu, B. L. Price, S. Cohen, and T. S. Huang. Deep image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 4, 2017. 1, 2, 5, 6, 7

[40] Y. Zheng and C. Kambhamettu. Learning based digital matting. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 889–896. IEEE, 2009. 5, 7

[41] B. Zhu, Y. Chen, J. Wang, S. Liu, B. Zhang, and M. Tang. Fast deep matting for portrait animation on mobile phone. pages 297–305, 2017. 2