Natural Image Matting Using Deep Convolutional Neural Networks

Donghyeon Cho[1(✉)], Yu-Wing Tai[2], and Inso Kweon[1]

[1] KAIST, Daejeon, South Korea
cdh12242@gmail.com, iskweon@kaist.ac.kr
[2] SenseTime Group Limited, Hong Kong, China
yuwing@gmail.com
https://sites.google.com/site/cnnmatting/

Abstract. We propose a deep Convolutional Neural Networks (CNN) method for natural image matting. Our method takes results of the closed form matting, results of the KNN matting and normalized RGB color images as inputs, and directly learns an end-to-end mapping between the inputs, and reconstructed alpha mattes. We analyze pros and cons of the closed form matting, and the KNN matting in terms of local and nonlocal principle, and show that they are complementary to each other. A major benefit of our method is that it can "recognize" different local image structures, and then combine results of local (closed form matting), and nonlocal (KNN matting) matting effectively to achieve higher quality alpha mattes than both of its inputs. Extensive experiments demonstrate that our proposed deep CNN matting produces visually and quantitatively high-quality alpha mattes. In addition, our method has achieved the highest ranking in the public alpha matting evaluation dataset in terms of the sum of absolute differences, mean squared errors, and gradient errors.

Keywords: Alpha matting · Deep CNN · Local and nonlocal matting

1 Introduction and Related Work

Image matting aims to extract an alpha matte of foreground given a trimap of an image. This problem can be expressed as a linear combination of foreground and background colors as follows [1]:

$$I = \alpha F + (1 - \alpha)B, \tag{1}$$

where I, F, B, and α denote the observed image (usually in RGB), foreground, background and mixing coefficients (alpha matte) respectively. Given an input I, finding F, B, and α simultaneously is a highly ill-posed problem.

Previous works in image matting have shown that, if we make proper assumptions, *e.g.* the color line model, about F and B, we can solve α in a closed form [2]. Local affinity based methods [2,3] analyze statistical correlation among local pixels to propagate alpha values from known regions to unknown pixels. When their

assumptions about local color distribution were violated, unsatisfactory results can be obtained. Nonlocal affinity based approaches [4–9] and color sampling based methods [10–14] rely on the nonlocal principle. They try to relax the local color distribution assumption by searching nonlocal neighbors and color samples which provide a better description of the image matting equation (Eq. (1)). Moreover, some works utilize multiple frames such as video [9,15] and camera arrays [16,18] to get local and nonlocal information across the images for matting.

Nonlocal methods, however, do not always outperform local methods. This is because these nonlocal methods were also built on top of some assumptions, *e.g.* nonlocal matting Laplacian [6], structure and texture similarity [13], comprehensive sampling sets [14], to search for proper nonlocal neighbors. In practice, alpha mattes from local methods are spatially smoother while alpha mattes from nonlocal methods can better capture long hair structures. There are also a few works [19,20] which implicitly deal with a combination of local and nonlocal principles.

We observe that there is a synergistic effect between local and nonlocal methods. The question is how these two kinds of methods can be combined effectively without losing the advantages of both methods. The answer, however, is not straight forward. An important criterion is that the solution should be able to adapt well to different image structures without depending too much on parameter tuning. Deep learning has recently drawn a lot of attentions in object recognition [21]. It has demonstrated its strength in feature extraction, classification [22,23], object detection [24,25] and saliency detection [26,27], as well as image reconstruction tasks such as image denoising [28], dirt removal [30], super-resolution [31], and image deblurring [33]. Because of its benefits in performance, and its versatility in various tasks, we are interested in applying deep learning to the natural image matting problem to bridge the gap between local and nonlocal methods. In addition, although deep learning has a lot of parameters in its training phase, it is almost parameter-free in its testing phase. Because the testing phase requires only a single forward pass of the deep architecture, it is also very efficient in computation especially with the supports of nowadays GPU implementation [34,35].

We have designed a deep CNN whose inputs are the alpha matte from the closed form matting, the alpha matte from the KNN matting, and the normalized RGB colors of the corresponding input image. We choose the closed form matting and the KNN matting as the representative of local and nonlocal methods because both methods are simple, mathematically solid and with publicly available source codes from their original authors. Also, both methods have a few parameters, and their performance is quite stable across wide range of examples. Our deep CNN is directly learnt from more than a hundred thousand of sampled image patches whose ground truth alpha mattes were collected from various sources [36]. We adopt data augmentation to increase variations and the number of training patches. In addition, we apply clustering using the ground truth alpha mattes to balance the number of training patches of different image structures. This is necessary in order to avoid overfitting of training data to particular type of image structures, *e.g.* long hairs.

After our deep CNN model is trained, we can directly apply our trained model to alpha matte reconstruction at the original resolution of input images. This is possible because our model utilizes only convolutional layers, and the convolutional layer do not have the fixed size limitation as opposed to the fully connected layer [37]. Our proposed deep CNN method can effectively combine the benefits of local and nonlocal information to reconstruct higher quality alpha mattes than both of its inputs. Note that this reconstruction is free of parameter, and the initial alpha mattes were obtained from the default parameters of the closed form matting and KNN matting. Because of the nonlinear units across the multiple layers in our deep CNN architecture, our results cannot be reproduced by a simple linear combination of our inputs. We have also found that different image structures activate different neurons in our deep CNN, which results in the best possible alpha mattes reconstruction from our inputs. Finally, we further extend our work by combining the inputs from closed form matting, KNN matting, and comprehensive matting, and a significant performance boost has achieved.

In summary, this paper offers the following contributions:

1. We introduce a deep CNN model for natural image matting. To our knowledge, this is the first attempt to apply deep learning to the natural image matting problem.
2. Our deep CNN model can effectively combine alpha mattes of local and non-local methods to reconstruct higher quality alpha mattes than both of its inputs. This is because our deep CNN model can "recognize" local image structures through the activations of different neuron units, and apply appropriate reconstruction scheme to adapt different image structures. This whole process is efficient and parameter-free once our deep CNN model is trained.
3. Our deep CNN method demonstrates outstanding performance in the public alpha matting evaluation benchmark dataset [36]. Our method has achieved the highest ranking in terms of the sum of absolute differences, mean squared errors, and gradient errors.

2   Review of Closed Form and KNN Mattings

Our method takes results from the closed form matting [2] and the KNN matting [8] as parts of our inputs. In this section, we briefly review these two methods, and discuss their strength and weakness in the natural image matting problem.

2.1   Closed Form Matting

The closed form matting assumes that the local color distribution follows a color line model where colors within a local window can be expressed as a linear combination of two colors. Based on this assumption, Levin *et al.* derived the matting Laplacian and proved that the alpha matte of foreground can be solved

in a closed form without explicit estimation of foreground and background colors. Since then, the matting Laplacian has been extensively used as a regularization term to enhance smoothness of estimated alpha mattes [5,12,13], and other applications [38,39].

Strength: In the closed form matting, there are only a few parameters: local window size, ϵ, and λ. In practical usages, a user only needs to adjust λ which is the regularization weight to define the strength of smoothness defined by the matting Laplacian. The value of λ can also be fixed for wide range of applications since the performance of matting Laplacian is quite robust to the values of its parameters. Its performance is guaranteed when the color line model assumption is satisfied.

Weakness: Although the color line model is quite general, there are a lot of cases where the color line model assumption is violated. It happens when background contains textures or multiple colors in a local region, or when local color distributions of foreground and background are overlapped. In addition, in order to satisfy the color line model, the local window size needs to be unavoidably small (*e.g.* 3×3). A large local window also makes the sparse matting Laplacian matrix computationally intractable. Consequently, the matting Laplacian contains only local information. Since alpha mattes are estimated through the propagation by the matting Laplacian, if the initial definite foreground or definite background samples provided by a user are far away from the matting regions, the estimated alpha mattes will still be over smoothed even through the color line model is satisfied. Also, alpha mattes in isolated regions of a trimap can never be correctly estimated since alpha values are propagated to local neighborhood only. These weaknesses are illustrated in Figs. 1 and 2.

2.2 KNN Matting

The KNN matting was derived based on the nonlocal principle in matting originally proposed by the nonlocal matting [6]. Its goal is to resolve the limitations of matting Laplacian by allowing alpha values to be propagated across nonlocal
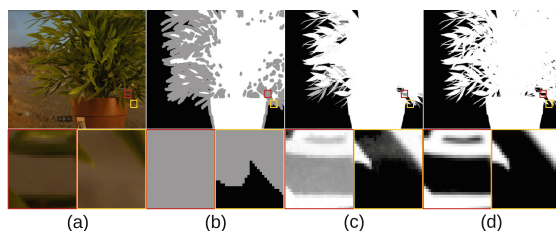

(a)   (b)   (c)   (d)

Fig. 1. (a) Input image. (b) Trimap. (c) Alpha matte from the closed form matting. (d) Ground truth alpha matte. Because the definite background samples are far away from object boundaries, the alpha matte in (c) is over smoothed.
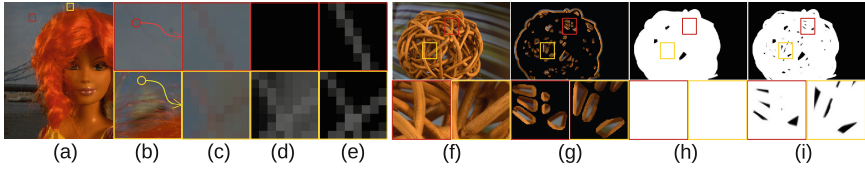
Fig. 2. Limitation of local principle for fine structures (a-e) and isolated regions (f-i). (a) Input image. (b) Cropped regions. (c) Double-zoom of the cropped regions. (d) Alpha mattes from the closed form matting. (e) Ground truth alpha mattes. (Red box) The alpha matte of the fine structures has disappeared because the definite foreground samples are too far away. The estimated background $(1-\alpha)$ is over smoothed. (Yellow box) The estimated foreground α of fine structures is over smoothed. (f) Input image. (g) Isolated regions within a trimap. (h) Alpha matte from the closed form matting. (i) Ground truth alpha matte. In this example, background pixels within the isolated trimap regions are considered as foreground because matting Laplacian cannot propagate alpha values across nonlocal neighbors. (Color figure online)

neighbors. Similar to the closed form matting, the nonlocal matting also makes an assumption about the sampled nonlocal neighbors. It assumes that the alpha value of a pixel can be described by a weighted sum of the alpha values of the nonlocal pixels that have similar appearance. In the nonlocal matting, the similar appearance is defined by colors, distance, and texture similarities. The computation of nonlocal matting, however, is very high due to the comparisons of nonlocal neighbors. The KNN matting improved the nonlocal matting by considering only the first K-nearest neighbors in a high dimensional feature space. It reduces the computation by considering only colors (in the HSV color space) and location similarity in their feature space. It also introduced a better preconditioning to further speed up computations. Interestingly, the alpha mattes by the KNN matting outperform the alpha mattes by the nonlocal matting because nonlocal neighbors at farther distance can be considered owing to the reduction of computations.

Strength: Similar to the closed form matting, the KNN matting also has a few parameters, and their parameters can be fixed for wide range of examples. Because the KNN matting utilizes nonlocal information, it can handle isolated regions, and better propagate alpha values across fine structures which are usually over smoothed by the closed form matting at a long distance.

Weakness: A major limitation of nonlocal methods is that it is difficult to define a universal feature space which can properly evaluate the nonlocal neighbors to adapt different structures of an image. Considering the KNN matting, it utilizes the HSV space instead of the RGB space in its feature vectors because the HSV feature has better quantitative performance in the alpha matting evaluation dataset [36]. However, it is controversial to conclude that the HSV feature always outperforms the RGB feature. This is illustrated in Fig. 3(a-d). Similarly, it is controversial to conclude that features that utilize more texture information can always outperform features that utilize only color information, especially for
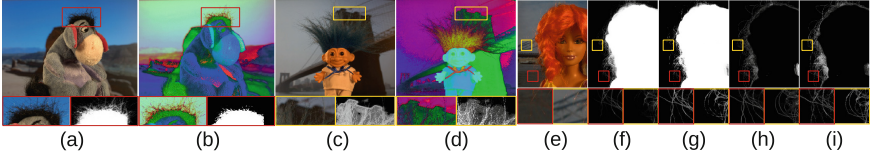
Fig. 3. Limitation of nonlocal principle in terms of feature space (a-d), and comparison of the closed form matting and KNN matting (e-i). (a, c) RGB images. (b, d) HSV images. The zoom-ins show the corresponding alpha mattes from the KNN matting with different color space features. The RGB feature produces better result than that of HSV feature in (a, b), but worse result in (c, d). (e) Input image. (f, g) Alpha mattes from the closed form matting and the KNN matting, respectively. (h, i) The corresponding error maps (enhanced for better visualization) of (f, g). (Color figure online)

the natural image matting problem. Because of the limitation of feature spaces, nonlocal methods can perform worse than local methods when improper nonlocal neighbors are considered.

To conclude, we compare the performance of the closed form matting and the KNN matting in Fig. 3(e-i). The closed form matting performs better in preserving local smoothness which has smaller errors in sharp, and short hair regions. In contrast, the KNN matting performs better in protecting long hair regions as shown in the zoom-in regions.

3   Deep CNN Matting

In this section, we first describe our deep CNN architecture. After that, we provide a deeper analysis to the activation of neurons in our deep CNN model.

3.1   Architecture

The architecture of our deep CNN model is illustrated in Fig. 4. Our network directly maps the input patches (27×27×5) to the output alpha matte (15×15×1) as follows:

$$\alpha = \mathcal{F}(\bar{I}, \alpha_c, \alpha_k), \tag{2}$$

where $\mathcal{F}(\cdot)$ denotes a forward pass of our network, $\bar{I} = \frac{I}{\sqrt{r^2+g^2+b^2}}$ is the input image whose intensity is normalized by the magnitude of RGB vector, α_c is the alpha matte from the closed form matting, and α_k is the alpha matte from the KNN matting. The main reason that the normalized RGB is adopted is to reduce magnitude variations of input signals since the magnitude variations are better captured in the initial alpha mattes (α_c and α_k). Similarly, we do not include the trimap in our input signals because the trimap information has already implicitly encoded in α_c and α_k. Also, strong edges in a trimap can
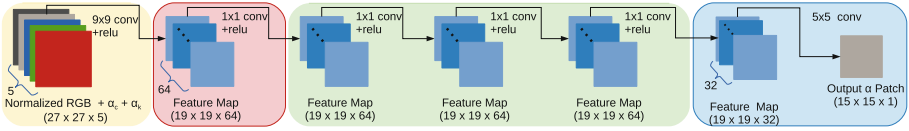
Fig. 4. The deep CNN architecture of our method. It consists of 6 convolutional layers. Except for the last layer, each convolutional layer is followed by a ReLU layer for the nonlinear mapping operation. The size of the convolutional kernels, and the number of channels in each layer are illustrated in the figure. In training, the input size is equal to 27×27×5, and the output size is equal to 15×15×1. The *Euclidean loss* cost function is used to evaluate the errors during the training. In testing, the spatial dimension of inputs and outputs are equal to the resolution of input images (with padding for input). Our deep CNN method directly outputs the resulting alpha mattes after a forward pass.

give inaccurate high activation responses which can hinder the accuracy of our reconstructed alpha mattes. The initial alpha mattes, α_c and α_k, are obtained using the default parameters provided in the original source codes of [2,8].

Our deep CNN model can be roughly divided into three stages according to the size of convolution kernels. In the first stage ($\mathcal{F}_1$), the first convolutional layer is convolved with the 5-channel inputs using 64 9×9 kernels which results in 64 response maps[1]. Mathematically, the response map after the first convolutional layer and the ReLU layer is defined as:

$$\mathcal{F}_{1,n}(\bar{I}, \alpha_c, \alpha_k) = \max(0, W_{1,n} \otimes [\bar{I}, \alpha_c, \alpha_k] + b_{1,n}), \tag{3}$$

where $W_{1,n}$ denotes the weight of the n-th filter in the first layer, and $b_{1,n}$ is the bias term. We set the bias term equal to zero, and the filter weights, $W_{1,n}$, are directly learnt from training examples. The first stage serves as structure analysis which activates response of different neurons according to the weights of the filters. After this stage, the response maps capture different local image structures in different output channels.

In the second stage ($\mathcal{F}_2 \sim \mathcal{F}_5$), it stacks multiple 1×1 convolutional layers to remap the response maps to enhance or suppress the neuron responses nonlinearly according to cross channel correlation. This process is similar to the nonlinear coefficient remapping in sparse coding for image superresolution [40] as discussed in [31]. In [31], only one layer of 1×1 convolutional layer is used for the remapping. We found that stacking multiple 1×1 convolutional layers can significantly enhance the performance of our method.

In the last stage ($\mathcal{F}_6$), the alpha mattes are directly reconstructed from the response maps after the second stage:

$$\alpha = \mathcal{F}_6(\mathcal{F}_5(\bar{I}, \alpha_c, \alpha_k)) = W_6 \otimes \mathcal{F}_5(\bar{I}, \alpha_c, \alpha_k). \tag{4}$$

---

[1] In CAFFE [34], each response map is a weighted sum of the response map of each input channel after passing through the 9×9 convolution. Thus, the number of output channels is equal to the number of filters defined in each layer, instead of the multiplication of the number of filters multiplied with the number of input channels.

We use kernels with size 5×5 for the reconstruction in order to consider spatial smoothness of the reconstructed alpha mattes.

During the training phase, the reconstructed alpha mattes are compared with the ground truth alpha mattes using the Euclidean loss cost function. The errors are back propagated to each layer to update the weights of kernels in each layer. In the testing phase, only a single forward pass is needed to reconstruct the resulting alpha mattes. We use zero padded input images, and directly apply the forward pass at the original image resolution to reconstruct a full resolution alpha matte directly from its inputs. There is no parameter tuning once the deep CNN model is learnt.

3.2   Analyses

Internal Response. We analyze the functionality of each stage by plotting the response maps ($\mathcal{F}_1 \sim \mathcal{F}_6$) at each layer. Figure 5 shows the response maps of two local patches with different local structures. In the top example, the alpha matte from the KNN matting is more accurate, while in the bottom example, the alpha matte from the closed form matting is more accurate. As visualized in the response maps after the first stage, their filter responses are significantly different from each other. One can interpret that each of the learnt kernels in the first stage are local classifiers which detect particular type of image structures within a local window. It can also be interpreted that the learnt kernels compare the alpha mattes from the closed form matting, and the KNN matting which result in very different response maps with respect to the original image structures, and evaluate which alpha mattes are more accurate.

Compared to the response maps in the previous layers, the reconstructed alpha mattes depends only on a subset of response maps which has been activated. Because the response maps are content adaptive, the reconstruction of alpha mattes can choose the best possible weighted combinations (learnt from training examples) of response maps to reconstruct the alpha mattes which are also adaptive to local image structures. An interesting observation in this example is that the response maps in the nonlinear remapping stage are getting closer
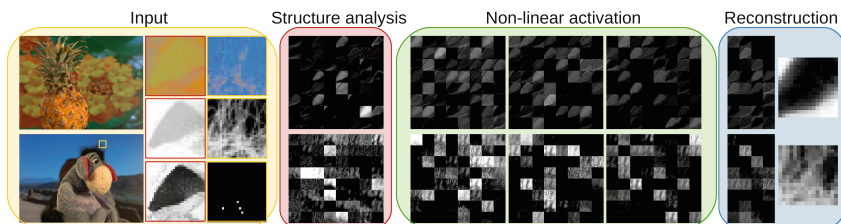

Fig. 5. Examples of response maps in each layer. Note that different image structures activate different neuron responses after the structure analysis stage. The non-linear activations remap the responses maps nonlinearly so that the reconstruction stage can directly reconstruct resulting alpha mattes from the response maps.

to reconstructed alpha mattes after each 1×1 convolutional and ReLU layers. This indicates that stacking multiple 1×1 convolutional and ReLU layers indeed helps to enhance the performance of our method.

Effect of the Number of Nonlinear Activation Layers. To further analyze the effects of nonlinear activation layers, we compare the performance of our CNN architecture by changing the number of nonlinear activation layers. We compare the performance with one, four, and eight layers of nonlinear activations. To provide a fair comparison, all architectures were trained from scratch with the same set of training examples and parameters, *e.g.* , the same initialization, learning rate, and number of iterations. Also, the number of channels of each nonlinear activation layer is fixed to 64. Table. 1 reports the average errors on a validation set and the processing time of a forward pass. As expected, more nonlinear activations can improve the results but the improvement with eight-layer architecture is marginal in compare with the results from four-layer architecture. Therefore, we choose the architecture with four-layer nonlinear activations to reduce running time and to avoid overfitting.

Table 1. Effect of the number of nonlinear activation layers.

[Avg.]	Sum of absolute difference	Mean squared error	Gradient error	Time (sec.)
Single layer	11.10	0.596	0.842	4.617
Four layers	10.30	0.563	0.804	5.126
Eight layers	10.17	0.546	0.792	6.528

Effects of Initial Alpha Mattes. We experiment the effectiveness of our network with different initial alpha mattes. In particular, without changing the network architectures, we re-train the network from scratch with different inputs: **RGB+Trimap**, **RGB+closed form matting**, **RGB+KNN matting**, **RGB+closed form matting+comprehensive sampling matting**, and **RGB+closed form matting+KNN matting+comprehensive sampling matting**. The **RGB+Trimap** is the standard input setting of the image matting problem.

Figure 6 shows the qualitative comparisons. Without the initial alpha mattes, the results from **RGB+Trimap** (Fig. 6(g)) are worse than the alpha mattes from conventional methods. The worse results may be deal to the usage of small network (with only one layer for structure analysis) or may be deal to the usage of small patches (27×27) for training. However, using larger network or larger patches (or entire images) for training would require significantly more training examples, and longer time to process. Also, the results from the state-of-the-art deep learning algorithm [41] for image segmentation are still imperfect which is not suitable for the image matting application. These shortcomings motivate us
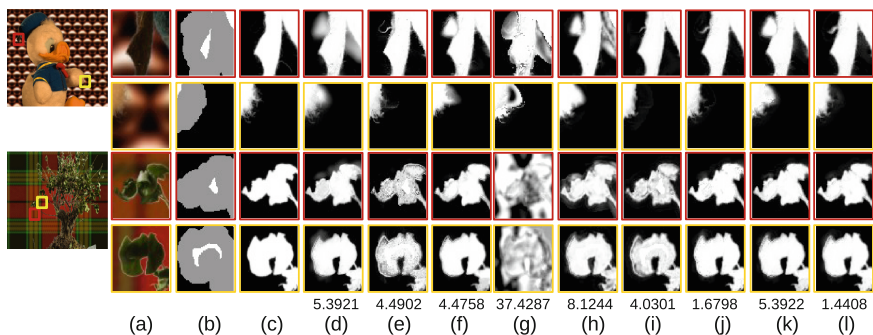
Fig. 6. Effect of initial alpha mattes. (a) Input Images. (b) Trimaps. (c) Ground truth alpha mattes. (d, e, f) Alpha mattes from the closed form matting [2], KNN matting [8], comprehensive sampling matting [14], respectively. (g-l) Results from our CNN model with different inputs. (g) RGB+Trimap. (h) RGB+Closed form. (i) RGB+KNN. (j) RGB+Closed form+KNN (Our standard setting). (k) RGB+Closed form+Comprehensive. (l) RGB+Closed form+KNN+Comprehensive. Numbers in the bottom are average RMSE of this examples.

to utilize alpha mattes from conventional methods, *e.g.* closed form and KNN mattings, as an approximate solution for refinement.

Figure 6(h, i) show the results where inputs are from **RGB+closed form matting** and **RGB+KNN matting**. With the alpha mattes from closed form matting, or KNN matting, the matting results are significantly improved. However, both results depend too much on the quality of input alpha mattes. Also, the limitations of local and nonlocal methods remain in their results respectively. Our results (**RGB+closed form matting+KNN matting**) which combine the alpha mattes from closed form matting and KNN matting are presented in Fig. 6(j). The results from (**RGB+closed form matting+KNN matting**) are significantly better than both of its inputs, which favourably combines the benefits of local and nonlocal methods.

To further analyze the effects of inputs, we have also trained a network which inputs are from **RGB+closed form matting+comprehensive sampling matting**. The comprehensive matting is chosen because its algorithm combines both local and nonlocal information, and its performance is better than both closed form matting and KNN matting. However, the results (Fig. 6(k)) from **RGB+closed form matting+comprehensive sampling matting** are worse than the results (Fig. 6(j)) from **RGB+closed form matting+KNN matting**. This may be because the comprehensive sampling matting also consider local information, which introduces bias to the inputs. Consequently, this combination cannot fully utilize the nonlocal information, and their results are worse than our results from **RGB+closed form matting+KNN matting**.

Finally, on top of initial alpha mattes from the closed form matting and KNN matting, we add the alpha mattes from the comprehensive matting. As shown in Fig. 6(l), results from three initial alpha mattes are slightly better than the

combination of the closed form matting and KNN matting. We submitted the results from **RGB+closed form matting+KNN matting+comprehensive sampling matting** together with the results from **RGB+closed form matting+KNN matting** to the evaluation site of alpha matting algorithms. Both results achieve the highest rank, and the results from three initial trimaps are better.

4   Experiment

In this section, we first describe our processes to prepare the training data. Then, we evaluate the performance of our deep CNN matting on the public alpha matting evaluation dataset [36], as well as some real world examples. Limitations and a failure case are also discussed. The trained model and testing codes are released in our website.[2]

4.1   Training

We collect training dataset from [36]. There are 27 examples which are composed of a RGB image, trimaps, and a ground truth alpha matte. For each example, we apply the closed form matting [2] and the KNN [8] matting to obtain their alpha mattes as part of our inputs: α_c and α_k. Since our training phase processes on each 27×27 image patches, we can generate a lot of training patches from the 27 examples. We have also increased the number of training data through data augmentation. In particular, using their ground truth alpha mattes, we composite the foreground onto different background to increase variations and the number of training examples. We have also exploited different rotation, reflection and resizing to increase the number of training patches. Using data augmentation, we can generate more than a hundred thousand of training patches.

While increasing the number of training patches can enhance performance of our trained deep CNN model, we noticed that data balancing is also very important when preparing the training data. We want to avoid overfitting of the training data to a particular type of alpha mattes. To resolve this issue, we cluster the training patches according to the number of pixels with non-zero alpha values, and with non-zero alpha gradients. If a patch has many pixels with non-zero alphas, but has a few pixels with non-zero alpha gradients, the patch can be considered as a sharp boundary patch. In contrast, if a patch has a few pixels with non-zero alphas, but has many pixels with non-zero alpha gradients, the patch can be considered as a long hair patch. Based on this analysis, we cluster the training patches into 20 groups. When preparing the training data for the deep learning, we balance the number of sampled patches from each group in order to avoid the overfitting problem of training data.

After preparing the training data, we train our deep CNN model using the back propagation. It takes around 2∼3 days for 10^6 number of iterations on

---

a machine with GTX 760 GPU and intel i7 3.4 GHz CPU. We use the method "xavier" (caffe parameter) to initialize the training weights. The xavier algorithm automatically determines the scale of initialization based on the number of input and output neurons. The learning rate, momentum and batch size are set to $10^{-5}, 0.9$ and 128 respectively. In the training phase, we did not pad the image patches. Therefore, the resolution of output (15×15) is smaller than the input (27×27). In testing phase, we zero padded the boundary of input images with 6-pixel width ($\frac{27-15}{2} = 6$), and directly apply the forward pass at the original image resolution with zero padded boundary to reconstruct a full resolution alpha matte. Thus, the resolution of our alpha matte is the same as the resolution of input image. A forward pass takes around $4 \sim 6$ s to process an image with a resolution of 800×640 pixels.

4.2   Evaluation

Quantitative Comparisons. Table 2 shows the quantitative comparisons on the testing dataset in [36]. The ground truths of the testing dataset are unavailable to public. The quantitative results are obtained by submitting our resulting alpha mattes to the evaluation website as "anonymous_submission" and "anonymous_submission (modified version)", and the scores are directly obtained from the evaluation website. As shown in Table. 2, our results (DCNN (Closed from + KNN)) and extensions (DCNN (Closed from + KNN + Comprehensive)) have dominated the first two rank in terms of SAD, MSE, and Gradient errors. Note that all results are obtained using the same network without parameter tuning. The initial alpha mattes are obtained by using the default parameters of closed form matting, KNN matting, and comprehensive sampling matting. Also, we did not separate the training examples nor separately train the network for small/large/user trimaps. Thus, our trained network is general and is applicable to different set of inputs.

Qualitative Comparisons. Figure 7 shows the qualitative performance of our deep CNN matting. We compared our results with results from the state-of-the-art matting algorithms: closed form matting [2], KNN matting [8], weighted color and texture matting [13], and comprehensive matting [14]. Our results are more stable and visually pleasing for various object structures: solid boundary (*elephant*), semi transparency (*net*), overlapped color distribution (*pineapple*), and long hair (*troll*). More qualitative comparisons with other methods and the whole set of our results can be found at www.alphamatting.com.

Additional Results. To further evaluate the performance, we generate synthetic data using the training dataset in [36] by replacing the original backgrounds with new backgrounds. These new backgrounds are very colorful and highly textured. The color line model assumption in the closed form matting is violated, and the local color distribution of foreground and background can be overlapped in these new examples. We use the *small* trimap provided by [36] to generate the matting results. These new examples are not included in our training dataset. Figure 8 shows the qualitative comparisons on these new synthetic

Table 2. Quantitative Comparisons in terms of sum of absolute differences, mean squared errors, and gradient errors. Only the top 10 results are displayed. The whole set of comparisons can be found in www.alphamatting.com (While we submitted multiple results (DCNN (Closed from + KNN + Comprehensive) and DCNN (Closed from + KNN)) for the evaluation, only the first rank results, DCNN (Closed from + KNN + Comprehensive), are public. Because the second rank results are not included in the public entries, the relative ranking scores in the public entries are slightly different from the scores reported in our table). Note that our results dominate the first two rank in the evaluations. The red, green, and blue highlight the top-3 results.

| Sum of Absolute Differences	Overall	Small	Large	User	Mean Squared Error	Overall	Small	Large	User	Gradient Error	Overall	Small	Large	User
1. DCNN (Closed from + KNN + Comprehensive)	2.8	4.4	1.1	3	1. DCNN (Closed from + KNN + Comprehensive)	4	5.1	1.8	5	1. DCNN (Closed from + KNN + Comprehensive)	7.2	9.6	5.8	6.1
2. DCNN (Closed from + KNN)	7.1	7.5	5.6	8.1	2. DCNN (Closed from + KNN)	5.5	4.4	4.5	7.5	2. DCNN (Closed from + KNN)	7.4	6.5	7.1	8.5
3. Cluster-based Sampling Matting	7.2	6.5	8.9	6.1	3. LNSP Matting	9.2	6.8	8.4	12.5	3. KL-Divergence Based Sparse Sampling	10.6	9.3	8.3	14.3
4. LNSP Matting	10.4	7.3	9.8	14.1	4. Cluster-based Sampling Matting	9.9	9.1	10.3	10.4	4. Anonymous TIP submission	10.7	8.9	8.5	14.6
5. Anonymous TIP submission	11	11.6	11.1	10.1	5. Trajectory	10.2	7.8	9.4	13.5	5. LNSP Matting	10.9	8.6	10.1	14
6. Trajectory	11.4	8.3	10.8	15.1	6. KL-Divergence Based Sparse Sampling	12.1	11.8	10.5	14	6. Trajectory	11.9	9.6	11.1	15
7. KL-Divergence Based Sparse Sampling	11.7	11.3	10.3	13.6	7. CCM	12.4	15.8	12.3	9.3	7. Comprehensive sampling	12	12.4	11	12.6
8. Comprehensive sampling	13.3	11.1	13	15.6	8. Anonymous TIP submission	13	14.1	12.6	12.3	8. Cluster-based Sampling Matting	13.5	13.9	14.1	12.5
9. Iterative Transductive Matting	13.7	15.5	12.8	12.9	9. Comprehensive sampling	13.6	12.6	13.4	14.8	9. CCM	14.1	16.8	13.6	11.9
10. SVR Matting	14	16.8	13.6	11.6	10. SVR Matting	13.8	17.8	12.3	11.5	10. SVR Matting	14.1	16.8	15	10.6
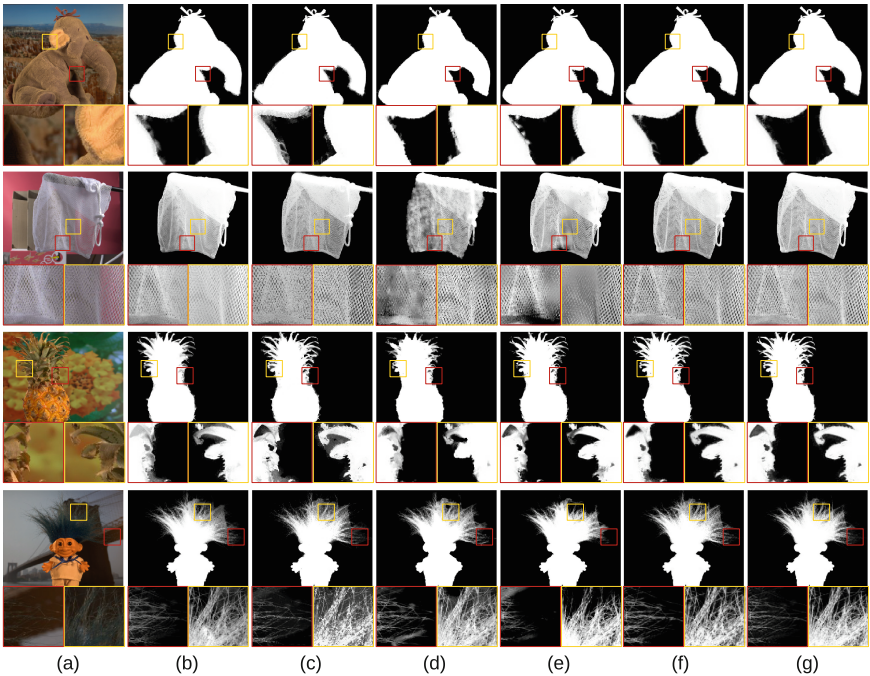
Fig. 7. Qualitative comparisons on the public dataset [36]. (a) Input images. (b, c, d, e) results from the closed form [2], KNN [8], weighted color and texture [13], and comprehensive [14] mattings. (f) Our results **(Closed from + KNN)**. (g) Our results **(Closed from + KNN + Comprehensive)**. (Color figure online)
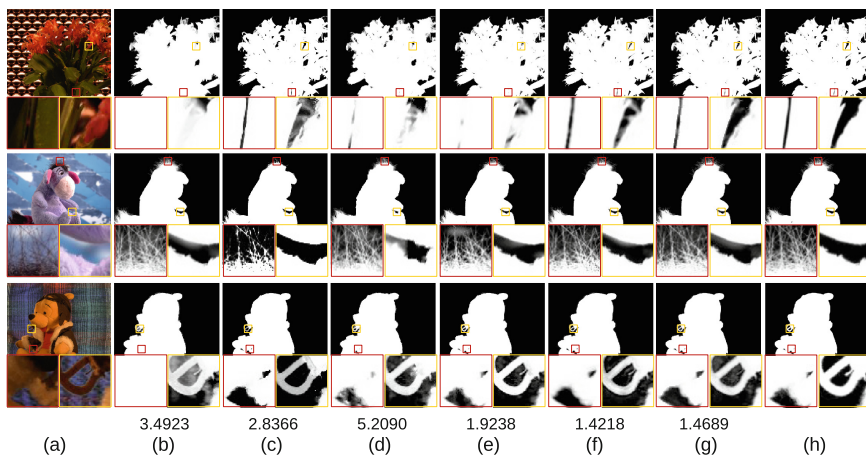
Fig. 8. Qualitative comparisons on synthetic dataset [36]. (a) Input images. (b, c, d, e) results from the closed form [2], KNN [8], weighted color and texture [13], comprehensive [14] matting. (f) Our results **(Closed from + KNN)**. (g) Our results **(Closed from + KNN + Comprehensive)**. (h) Ground truths. Numbers in the bottom are average RMSE of this examples.
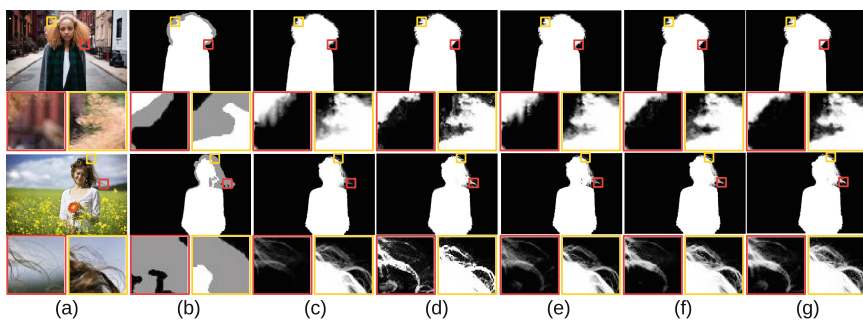

Fig. 9. Qualitative comparisons on a real world image. (a) Input images. (b) Trimaps (c, d, e) Results from the closed form [2], KNN [8], and comprehensive [14] mattings. (f) Our results **(Closed from + KNN)**. (g) Our results **(Closed from + KNN + Comprehensive)**.

datasets. Compared to results from the other methods, our deep CNN matting estimates more accurate alpha mattes.

Figure 9 shows qualitative comparisons on real world images. The top row example contains short curly hairs while the bottom example contains long hairs. In both examples, the closed form matting produces over smoothed alpha mattes while the KNN matting produces visually unpleasing results as shown in Fig 9(c, d), respectively. In contrast, as shown in Fig 9(f), our deep CNN matting can combine results from the closed form matting and KNN matting properly to reconstruct accurate alpha mattes automatically by recognizing local image

structures. In other words, our proposed method can take advantages of both local and nonlocal principles depending on the recognized local image structures.

Failure Case. Our deep CNN matting takes the alpha mattes from the closed form matting, and the alpha mattes from KNN matting as part of the inputs. It is unavoidable that the quality of our results would depend on the quality of inputs. When the alpha mattes from both methods fail simultaneously, our matting results would contain similar artifacts as its inputs. This failure case is illustrated in Fig. 10 (red box). However, because our deep CNN matting can recognize structures, even alpha mattes from the both methods contain artifacts, our method can still produce reasonable alpha mattes better than both of its inputs (Fig. 10 (yellow box)) if their artifacts are different from each other.
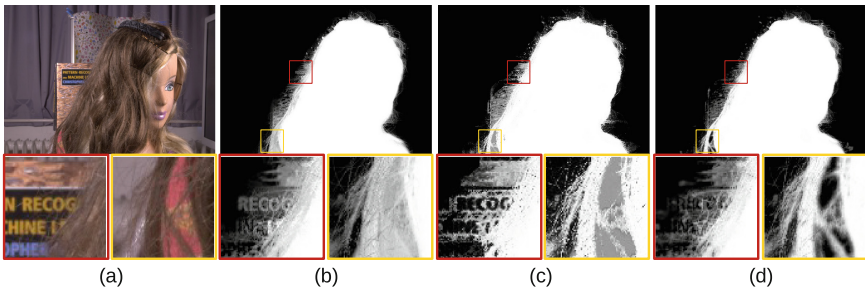

(a)   (b)   (c)   (d)

Fig. 10. Failure case (red box). (a) Input image. (b, c, d) Results from the closed form matting [2], KNN matting [8], and our method. (Color figure online)

5   Conclusion

In this paper, we have introduced the deep CNN matting. Our deep CNN matting takes the advantages of both local and nonlocal methods, and can adaptively reconstruct high quality alpha mattes from its inputs by recognizing local image structures. Our method is effective and parameter-free once the deep CNN model has been trained. Our matting results have achieved the highest rank in the benchmark dataset [36] in terms of sum of absolute differences, mean squared errors and gradient errors. To our knowledge, this is also the first attempt to apply deep learning to the natural image matting problem. We believe that our method is highly innovative and inspires follow-up works. As a future work, we are planning to study how to relax the dependency of our results with respect to the quality of the input alpha mattes.

References

1. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A Bayesian approach to digital matting. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2001)
2. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **30**(2), 0162–8828 (2008)
3. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. ACM Trans. Graph. (ToG) **23**(3), 315–321 (2004)
4. Zheng, Y., Kambhamettu, C.: Learning based digital matting. In: Proceedings of the International Conference on Computer Vision (ICCV) (2009)
5. He, K., Sun, J., Tang, X.: Fast matting using large kernel matting laplacian matrices. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2010)
6. Lee, P., Wu, Y.: Nonlocal matting. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2011)
7. Lin, H.T., Tai, Y.W., Brown, M.S.: Motion regularization for matting motion blurred objects. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2329–2336 (2011)
8. Chen, Q., Li, D., Tang, C.K.: KNN matting. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2012)
9. Choi, I., Lee, M., Tai, Y.-W.: Video matting using multi-frame nonlocal matting laplacian. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 540–553. Springer, Heidelberg (2012)
10. Wang, J., Cohen, M.F.: Optimized color sampling for robust matting. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2007)
11. Gastal, E.S.L., Oliveira, M.M.: Shared sampling for real-time alpha matting. In: EUROGRAPHICS (2010)
12. He, K., Rhemann, C., Rother, C., Tang, X., Sun, J.: A global sampling method for alpha matting. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2011)
13. Shahrian, E., Rajan, D.: Weighted color and texture sample selection for image matting. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2012)
14. Shahrian, E., Rajan, D., Price, B., Cohen, S.: Improving image matting using comprehensive sampling sets. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2013)
15. Li, D., Chen, Q., Tang, C.K.: Motion-aware KNN laplacian for video matting. In: ICCV (2013)
16. Joshi, N., Matusik, W., Avidan, S.: Natural video matting using camera arrays. In: ACM SIGGRAPH (2006)
17. Kim, S., Tai, Y.W., Bok, Y., Kim, H., Kweon, I.: Two-phase approach for multi-view object extraction. In: ICIP (2011)
18. Cho, D., Kim, S., Tai, Y.-W.: Consistent matting for light field images. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part IV. LNCS, vol. 8692, pp. 90–104. Springer, Heidelberg (2014)
19. Rhemann, C., Rother, C., Gelautz, M.: Improving color modeling for alpha matting. In: British Machine Vision Conference (BMVC) (2008)
20. Chen, X., Zou, D., Zhou, S.Z., Zhao, Q., Tan, P.: Image matting with local and nonlocal smooth priors. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2013)

21. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) **115**(3), 211–252 (2015)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
23. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2015)
24. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2014)
25. Girshick, R.: Fast R-CNN. In: Proceedings of the International Conference on Computer Vision (ICCV) (2015)
26. Zhao, R., Ouyang, W., Li, H., Wang, X.: Saliency detection by multi-context deep learning. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2015)
27. Lee, G., Tai, Y.W., Kim, J.: Deep saliency with encoded low level distance map and high level features. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2016)
28. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Proceedings of the Neural Information Processing Systems (NIPS) (2012)
29. Xu, L., Ren, J.S., Yan, Q., Liao, R., Jia, J.: Deep edge-aware filters. In: Proceedings of International Conference on Machine Learning (ICML) (2015)
30. Eigen, D., Krishnan, D., Fergus, R.: Restoring an image taken through a window covered with dirt or rain. In: Proceedings of International Conference on Computer Vision (ICCV) (2013)
31. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part IV. LNCS, vol. 8692, pp. 184–199. Springer, Heidelberg (2014)
32. Ren, J.S., Xu, L., Yan, Q., Sun, W.: Shepard convolutional neural networks. In: Proceedings of Neural Information Processing Systems (NIPS) (2015)
33. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: Proceedings of the Neural Information Processing Systems (NIPS) (2014)
34. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
35. Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., Shelhamer, E.: cuDNN: efficient primitives for deep learning. arXiv:1410.0759v3 (2014)
36. Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., Rott, P.: A perceptually motivated online benchmark for image matting. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2009)
37. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **37**(9), 1904–1916 (2015)
38. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2009)

39. He, K., Sun, J., Tang, X.: Guided image filtering. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 1–14. Springer, Heidelberg (2010)
40. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Process. (TIP) **19**(11), 2861–2873 (2010)
41. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: Proceedings of the International Conference on Computer Vision (ICCV) (2015)