Highly Efficient Natural Image Matting [1]

Yijie Zhong
dun.haski@gmail.com
Bo Li[1]
libraboli@tencent.com
Lv Tang
luckybird1994@gmail.com
Hao Tang[2]
hao.tang@vision.ee.ethz.ch
Shouhong Ding[1]
ericshding@tencent.com

[1] Youtu Lab,
Tencent,
Shanghai, China

[2] Computer Vision Lab,
ETH Zurich,
Switzerland

Abstract

Over the last few years, deep learning based approaches have achieved outstanding improvements in natural image matting. However, there are still two drawbacks that impede the widespread application of image matting: the reliance on user-provided trimaps and the heavy model sizes. In this paper, we propose a trimap-free natural image matting method with a lightweight model. With a lightweight basic convolution block, we build a two-stages framework: Segmentation Network (SN) is designed to capture sufficient semantics and classify the pixels into unknown, foreground and background regions; Matting Refine Network (MRN) aims at capturing detailed texture information and regressing accurate alpha values. With the proposed cross-level fusion Module (CFM), SN can efficiently utilize multi-scale features with less computational cost. Efficient non-local attention module (ENA) in MRN can efficiently model the relevance between different pixels and help regress high-quality alpha values. Utilizing these techniques, we construct an extremely light-weighted model, which achieves comparable performance with 1% parameters (344k) of large models on popular natural image matting benchmarks.

1 Introduction

The natural image matting is an important task in computer vision, which serves as a prerequisite for a broad set of applications, including image or video editing, compositing and film post-production [26, 35, 38, 39, 40, 41, 42, 43, 44]. Following [32], the natural image I is defined as a convex combination of foreground image F and background image B at each pixel i as:

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

where α_i is the alpha value at pixel i that denotes the opacity of the foreground object. Mathematically, image matting requires expressing pixel colors in the transition regions

[1] **Bo Li is equal contribution to the first author. Corresponding authors are Bo Li and Lv Tang.**

from foreground to background as a convex combination of their underlying foreground and background colors. Since neither the foreground and background colors nor the opacities are known, estimating the opacity values is a highly ill-posed problem. To alleviate the difficulty of this problem, typically a trimap is provided in addition to the original image [1]. The trimap can be provided by user or generated from the network. The trimap is a rough segmentation of the input image into foreground($\alpha_i = 1$), background($\alpha_i = 0$), and regions with unknown opacity($\alpha_i \in (0,1)$) which like other segmentation tasks [18, 19, 20, 22, 23, 24, 45, 47]. From then on, many traditional methods [1, 8, 11, 17, 36, 37, 48] are proposed for matting task. These methods leverage color features to predict alpha mattes, which will produce obvious artifacts when foreground and background share similar colors.

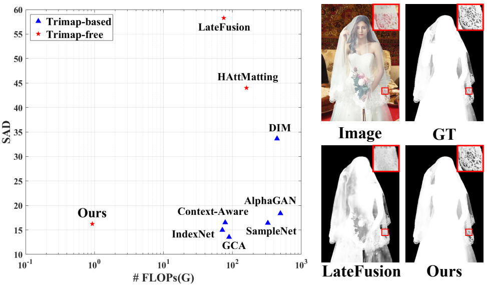


Figure 1: Comparison with SOTA methods. Best viewed by zooming in. Smaller SAD and FLOPs mean better performance and less computational cost.

Deep learning has recently demonstrated success in many computer vision applications. Recent researches use deep neural networks to improve the performance of natural image matting. Compared to traditional methods, deep-learning matting methods [2, 6, 14, 25, 29, 44, 50] have achieved remarkable performance and further advance the matting field. However, the widespread application of image matting still faces two major challenges: first, most methods rely on user-provided trimaps as assistant input, which is not easy to use for novice users without expertise in digital matting; second, their heavy model sizes make them resource-hungry and hardly applicable on low-power devices with limited storage/computational capability.

To address first problem, [33, 53] design trimap-free natural image matting models. [53] investigated semantic segmentation variant for foreground and background weight map fusion to obtain alpha values. [33] explores advanced semantics and appearance cues synthetically to achieve high-quality alpha values. However, to capture sufficient semantics, these two methods utilizate large backbones (DenseNet-201 and ResNeXt101), which make it difficult to apply these methods to some devices with limited storage/computational capability. To address second problem, [51] designs a light-weighted natural image matting model with the help of knowledge distillation. However, this method also needs extra trimaps, which is not user-friendliness. Hence, designing a trimap-free natural image matting method with a light-weighted model is barely explored.

Without the user-provided trimap, there are two basic tasks for automatic natural image matting: identify the foreground, background and unknown regions in the image; predict the precise alpha values of unknown regions. The first one is a classic classification task while the second task is typical regression task. As discussed in [53], it is hard to handle the two tasks simultaneously with a single structure, we thus propose a two-stage end-to-end deep learning framework for trimap-free light-weighted matting task. The first stage is designed to capture sufficient semantics to identify the foreground, background, and unknown regions. Then under the guidance of the first stage, the second stage can give the precise alpha values of unknown regions. Although such two-stage network design has appeared before (e.g. [55], [16], [4]), it is an intutitive idea. The most important is that [55] designs redundant structures besides low-resolution input (128 × 128), and the strong portrait prior limit their application prospects. Then the core problem lies in how to accomplish these two tasks with light-weighted network structures.

We approach this core problem from two perspectives: a light-weighted backbone and efficient feature aggregation/enhancement strategies of each stage. To be specific, we first build a light-weighted basic block with some popular efficient convolutional operations. Then we stack these basic blocks to construct light-weighted U-Net like backbones for two stages. The first stage is segmentation network (SN), and the second stage is matting refine network (MRN). In the first stage, we design a cross-level fusion module (CFM) to directly fuse the high-level semantic feature with low-level feature, which can improve the discriminability of feature representation. In the second stage, we design a novel efficient non-local attention module (ENA) to model the relevance between different pixels in encoder/decoder feature and image feature extracted from octave block (OCBlock), which can help regress high-quality alpha values. Moreover, in matting task, modeling the relevance of different pixels in long-range is not as important as modeling the relevance of pixels in short-range. So the proposed ENA first calculates long-range relevance with sampling strategy, then directly calculates short-range relevance. This design meets the principle of a light-weighted model. As can be seen in Fig.1, compared with LateFusion [53], our proposed method can predict a better result with accurate semantics and alpha value.

Our major contributions can be summarized as:

- We make one of the earliest effort on designing a light-weighted trimap-free natural image matting method, which expands matting to much wider application scenarios.
- We construct two U-Net like backbone with light-weighted basic blocks for the proposed two-stage framework and propose two corresponding efficient feature aggregation/enhancement strategies CFM and ENA for two stages to better identify the unknown regions and give precise alpha values.
- We perform extensive experiments to demonstrate the proposed method refreshes the SOTA performance on Composition-1k and Distinctions-646 testing sets with 1% parameters (344k) of SOTA large models.

2 Method

In this section, we first describe the overall architecture of the proposed trimap-free light-weighted natural image matting network, then introduce our proposed light-weighted backbone, SN and MRN. The architecture of the proposed approach is illustrated in Fig.2.

2.1 Network Overview

SN has a simple U-Net like Encoder-Decoder architecture, and the backbone is formed by stacking our proposed OCBlock. Similar to ResNet-50 [12], the proposed backbone has four levels, named as En1-1, En1-2, En1-3, En1-4. The number of OCBlock in each level is 3, 4, 6, 4. Because En1-1 is too close to the input and its resolution is too high, to save computational cost, we only use the last three levels features for the following process. Cross-level fusion module (CFM) is added between En1-4 and En1-2 to improve the discriminability of feature representations. SN has 3 decoders, and each decoder is stacked by 1 OCBlock.

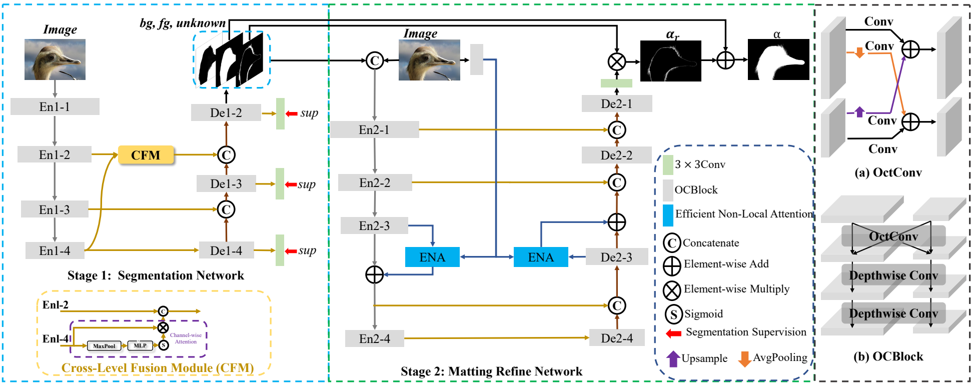


Figure 2: The framework of the our trimap-free light-weighted natural image matting model.

Decoder fuses the processed features from the encoders and the upsampled features from the previous decoder level in a bottom-up manner. The output of each decoder is defined as De1-4, De1-3, and De1-2. Finally, the network generates the 3-channel classification logits T by a 3×3 convolution.

MRN has a basic Encoder-Decoder architecture, and the input of MRN is predicted logits T and an image. The backbone is also stacked by our proposed OCBlock, and all encoder levels contain 2 OCBlocks, respectively. To accurately regress alpha values in unknown regions, we stick two efficient non-local attention (ENA) module to the encoder and decoder symmetrically in our stem. Moreover, image features extracted from OCBlocks contain more detailed texture information, which can help regress alpha values. Thus, we utilize the image features extracted from OCBlocks to guide the information flow on Encoder/Decoder features. MRN has 4 decoders, and each decoder is stacked by 1 OCBlock. Decoder fuses the features in each encoder and the upsampled features from the previous level in a bottom-up manner. Finally, together with the unknown and foreground maps predicted by SN, MRN can regress a 1-channel result α.

2.2 Light-weighted backbone

Recently, some researchers have designed various light-weighted backbones [7, 15, 21, 46, 52] with special convolution operations. A classic example is MobileNet which constructs a light-weighted backbone by using the depthwise convolution. However, these light-weighted models are designed for classification, it is not appropriate to directly use them for matting task. Unlike classification task, matting needs multi-scale information to capture sufficient semantics for pixel-level prediction. Some researchers try to perform extra multi-scale feature extraction after backbone, like atrous spatial pyramid pooling (ASPP), global convolutional network (GCN) [31], which need more computational cost. Inspire by MobileNet, we want to use depthwise convolution to build our lightweight backbone. But only using depthwise convolution cannot capture sufficient multi-scale information. To address this problem, by leveraging the OctConv [5], we design our lightweight OCBlock, and stack OCBlocks to construct a light-weighted backbone. To be specific, we replace a depthwise convolution with a vanilla OctConv. The vanilla OctConv [5] is presented to conduct the convolution operation on features across high/low scales, as shown in Fig.2(a). In OctConv, features from different scales are fused to generate output. The proposed OCBlock is shown in Fig.2(b), which combines the advantages of depthwise convolutions and OctConv to achieve a better weight/performance balance.

2.3 Segmentation Network

SN plays the role of capturing sufficient semantics and classifying the pixels into unknown, foreground, and background regions with light-weighted backbone. Thus, learning discriminant feature representations with less computational cost is essential. To achieve this goal, we first utilizate our proposed light-weighted backbone to obtain multi-scale feature representations. Moreover, we utilize CFM module to leverage the advantages of features at high-level to learn more discriminant feature representations.

Cross-Level Fusion Module. High-level feature from En1-4 contains much semantic information, which can make the network better distinguish the appearance difference between unknown, foreground, and background regions. However, our proposed SN is a U-like architecture, when information is progressively returned from the high-level to the low-level, the high-level semantic information is gradually diluted. Thus, we directly fuse low-level feature En1-2 with high-level feature En1-4, to propagate the high-level semantic information to the low-level. As different channels of feature in high-level response to different semantics, it is unwise to treat all channels without distinction. To alleviate the interference of the irrelevant semantic information, we use channel-wise attention on En1-4 to assign larger weights to channels which are more important for matting task. The operations in channel-wise attention are with low time complexity. Therefore, it demands negligible computational cost.

2.4 Matting Refine Network

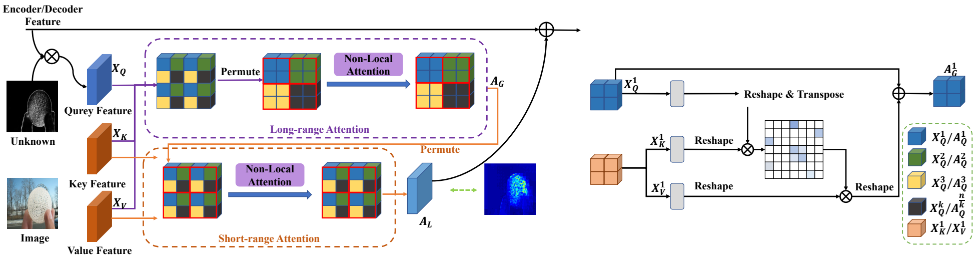
MRN aims at regressing the alpha values of pixels in unknown regions. The MRN takes the concatenation of 3-channel images and the 3-channel segmentation results from SN as 6-channel input. For easy remembering, we named unknown regions as U. We design an efficient non-local attention module which can help better regress the alpha values with less computational cost.

Efficient Non-Local Attention. Modeling the relevance of different pixels can help regress the accurate alpha value. Because it can help the network effectively group pixels with similar texture, then pay more attention to these pixels. Inspired by [49], we use non-local attention to achieve this goal. Common non-local attention can be defined as:

$$A = Sim(Q,K)V, \tag{2}$$

where Q,K and V mean *query*, *key* and *value* feature, and Sim is the similarity between query and key feature.

According to traditional methods, pixels with almost identical texture should have similar opacity. Hence, pixels that share similar texture information should have similar opacity features. However, as the network layers deepen, the feature representations of each pixel will be blurred, which will affect the regression of alpha value. Therefore, we use detailed image features to guide the information flow on Encoder/Decoder features, which improves the regression accuracy of alpha values. In our proposed work, the *query* feature is the feature from Encoder/Decoder, named as X_Q. The $key/value$ feature is the feature extracted from the input image, named as X_K and X_V.



(a) Efficient Non-Local Attention Module (b) Non-Local Attention in Long-range

Figure 3: The illustration of the proposed ENA. The query feature is obtained from the multiplication of encoder/decoder features and unknown region. Image is passed through OCBlocks to obtain the key and value feature. Our proposed ENA is a cross-level non-local attention, but for the sake of simplicity, we have only shown the long-range and short-range attention operations on query feature X_Q. In this figure, we set $k=4$ and $n=16$.

Because MRN only cares about regressing the alpha values in unknown region, we multiply X_Q with U to make *query* feature more focused on the unknown region. It can be formulated as:

$$X_Q = X_Q * U. \tag{3}$$

Directly using conventional non-local attention in our work is time-consuming, which is not in line with the design principle of light-weighted models. Moreover, motivated by [3], modeling the relevance of different pixels in long-range is not as important as modeling the relevance of pixels in short-range, and sometimes will lead to some noise because of irrelevant matches. Thus, our proposed ENA is the combination of long-range attention with sampling strategy and short-range attention. The main point of the long-range attention is to apply the non-local attention to the subsets of positions that have long spatial interval distances, which can not only help reduce the probability of irrelevant matches between long-range pixels, but also save computational cost. The details of long-range attention can be seen in Fig.3(a). The input feature $X \in \{X_Q, X_K, X_V\}$ are first divided into k partitions, then sampling these pixels in different partitions at intervals $\sqrt{k}$ into new subsets. In our work, we use permute operation on behalf of sampling. So each subset contains $\frac{n}{k}$ pixels, where n represents the size of input features. Thus the new subsets are defined as: $X_L = [X^1, X^2, \cdots, X^k]$, where $X^k \in \{X_Q^k, X_K^k, X_V^k\}$, and the shape of X^k is $\mathbb{R}^{\frac{n}{k} \times C}$. C means the channel numbers. Then we apply the non-local attention on X^k independently:

$$A_G^k = Sim(X_Q^k, X_K^k) X_V^k, \tag{4}$$

which is shown in Fig.3(b). Finally, we merge all the A_G^k to get the output A_G. For short-range attention, we permute the A_G to group the originally nearby positions together, which is the *query* of short-range attention. So the input of short-range attention is $Z \in \{A_G, X_K, X_V\}$. Similarly, We divide the into $\frac{n}{k}$ partitions and each partition contains k neighboring positions, defined as: $Z_S = [Z^1, Z^2, \cdots, Z^{\frac{n}{k}}]$, where $Z^{\frac{n}{k}} \in \{A_G^{\frac{n}{k}}, X_K^{\frac{n}{k}}, X_V^{\frac{n}{k}}\}$, and the shape of $Z^{\frac{n}{k}}$ is $\mathbb{R}^{k \times C}$. Then we apply the non-local attention on each $Z^{\frac{n}{k}}$ independently. Accordingly, we can get $A_L^{\frac{n}{k}}$ and merge all of them to get the output A_L. Finally, we directly fuse the A_L with Encoder/Decoder features to get the refined features. In our proposed model, we set $k=16$.

We analyze the computation/memory cost of the common non-local attention mechanism and our ENA in supplementary materials Section 3.

MRN Output. Together with unknown maps, MRN will output the refined alpha values α_r in unknown regions. The final alpha matte result α can be written as:

$$\alpha = F + \alpha_r. \tag{5}$$

2.5 Loss Function

To supervise our proposed network, the loss function contains two parts: L_{SN} and L_{MRN}. Let α^{gt} be the groundtruth of final prediction α.

The prediction of SN is 3-channel classification logits T. Following [2], we generate its groundtruth T^{gt} according to α^{gt}, which is defined as:

$$T^{gt}(x,y) = \begin{cases} 0, & \alpha^{gt}(x,y) = 0 \\ 1, & \alpha^{gt}(x,y) = 1 \\ 2, & 0 < \alpha^{gt}(x,y) < 1 \end{cases} \tag{6}$$

where (x,y) stands for each pixel location on the image. The value $\{0,1,2\}$ means the pixel is in background, foreground or unknown region. We use Softmax cross-entropy loss to supervise SN, which is defined as:

$$L_{SN} = \sum_{l}^{l=3} CE(T_l, T_{gt}). \tag{7}$$

As show in Fig.2, we use multi-level supervision as an auxiliary loss to facilitate sufficient training. T_l means the prediction result in $l-$level.

To supervise MRN, following [53], we adopt the weighted L_1 loss and gradient loss L_g. The weighted L_1 loss is defined as:

$$L_1 = \sum_{p} w_p \cdot |\alpha_p - \alpha_p^{gt}|, \tag{8}$$

where α_p means predicted alpha value at pixel p. The weight w_p is set to 1 when $0 < \alpha_p < 1$ and set to 0.1 when $\alpha_p = 1$ or $\alpha_p = 0$. This operation can make the network pay attention to unknown region. The gradient loss L_g is used to remove the over-blurred alpha matte, which is defined as:

$$L_g = |\nabla_x(\alpha) - \nabla_x(\alpha^{gt})| + |\nabla_y(\alpha) - \nabla_y(\alpha^{gt})|. \tag{9}$$

Thus L_{MRN} can be written as: $L_{MRN} = L_1 + L_g$. Note that all parts of our proposed network are trained jointly, so the overall loss function is given as:

$$L = L_{SN} + L_{MRN}. \tag{10}$$

3 Experiments

3.1 Dataset and Evaluation Metrics

Dataset. We evaluate our proposed network on the public Adobe Composition-1k [50]. The training set consists of 431 foreground objects with the corresponding ground truth alpha

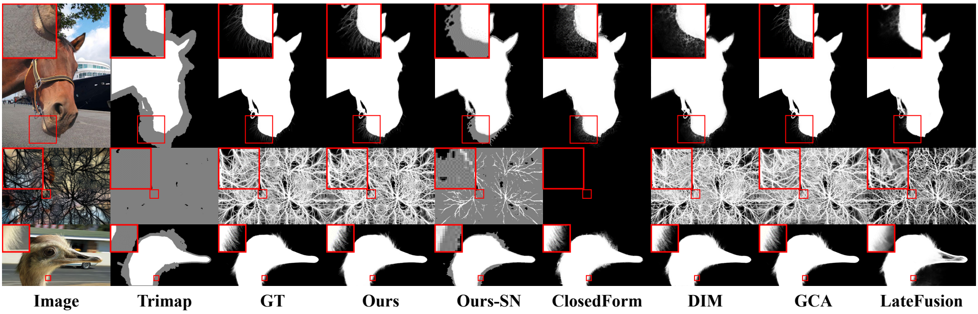


Figure 4: The visual comparisons on the Composition-1k testing dataset. Zoom in for the best view.

mattes. Each foreground image is combined with 100 background images from MS COCO dataset [27] to composite the input images. For the test set, the Composition-1k contains 50 foreground images as well as the corresponding alpha mattes, and 1000 background images from PASCAL VOC2012 dataset [9]. The training and test sets were synthesized through the algorithm provided by [50].

For the data augmentation, following [25], the foreground object and the alpha image will be resized to 640 × 640 images with a probability of 0.25 to make the network see the whole foreground image instead of a cropped snippet. Then, a random affine transformation is used. Subsequently, we randomly crop one 512 × 512 patch from each foreground. All of the patches are centered on an unknown region according to the alpha images. The foreground images are then converted to HSV space, and different jitters are imposed to the hue, saturation, and value. During testing, the resolution of input image is 512 × 512.

Evaluation metrics. There are four metrics used in the evaluations: SAD (sum of absolution difference), MSE (mean square error), gradient, and connectivity defined in [50]. The lower values of the metrics, the better the predicted alpha matte is. The details of gradient and connectivity metrics can be found in [34], they are used to reflect the visual quality of the alpha matte when observed by a human.

Implementation Details. We use Pytorch to implement our model. The whole network is trained in an end-to-end manner. For loss optimization, we use the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is initialized to 4×10^{-4}. Warmup and cosine decay [13, 28] are applied to the learning rate. The network is trained for 70 epochs with a batch size of 12. We set the weight for L_{SN} to 1, for L_g to 0.5, and for L_1 to 5. If not specially mentioned, we use this as the default settings. Our model can be trained from scratch without other datasets.

3.2 Evaluation on Composition-1k Test Set

Here we compare our network with 5 traditional matting methods: Shared Matting [10], Learning Based [54], Global Matting [11], ClosedForm [17], KNN Matting [3], and 7 deep learning based methods: DCNN [6], DIM [50], AlphaGAN [30], IndexNet [29], Late Fusion [53], HAttMatting [33], GCA [25]. LateFusion, HAttMatting and our proposed network are trimap-free matting methods. For other methods, we feed RGB images and trimaps produced by 25 pixels random dilation refer to [50]. We compute the metrics at 800 × 800 for all methods. The testing settings in our work are following two trimap-free methods [33, 53]. The qualitative results are illustrated in Fig.4 and quantitative comparisons are reported in Tab.1. Because we cannot obtain the executable code or results of HAttMatting, we do not show their qualitative results and directly cite quantitative results from their original paper.

As can be seen in Tab.1, our proposed method outperforms all the traditional image matting methods by a large margin. Compared to trimap-based deep-learning methods, our method is slightly inferior to GCA and IndexNet, and is better than DIM and AlphaGAN. However, trimaps are strongly required during their training and inference phase, which restricts their effectiveness in practical applications. Compared to trimap-free methods LateFusion and HAttMatting, our proposed method can already outperform these two methods. Moreover, our method achieves SOTA performance with fewer parameters (0.3% $\sim$ 7.8% of large models) and FLOPs (0.2% $\sim$ 1% of large models) than all of the above methods which will upgrade high practicality.

Table 1: Quantitative and complexity comparison on the Composition-1k testing dataset. 'DIM-Trimap-less' denotes the results of the DIM method without trimap as input during training. The underline indicates the best of the trimap-based methods. The bolded numbers represent the best performace in trimap-free methods.

Model	Complexity		Composition-1k			
	#PARM.	FLOPs	SAD	MSE	Grad	Conn
Shared Matting	-	-	115.20	0.031	139.88	121.35
Learning Based	-	-	100.51	0.027	107.83	104.74
Global Matting	-	-	121.46	0.034	125.11	133.23
Closed-form	-	-	118.98	0.032	130.77	118.17
KNN Matting	-	-	133.99	0.051	140.29	134.03
DCNN	-	-	122.40	0.030	129.57	121.80
DIM-Trimap-less	130.55M	443.56G	103.07	0.108	73.76	106.64
DIM	130.55M	443.56G	33.64	0.008	30.23	31.92
IndexNet	5.72M	71.41G	14.97	0.006	12.02	14.58
AlphaGAN	40.3M	496.36G	20.28	0.010	21.11	20.46
GCA	25.16M	89.41G	13.54	0.006	11.39	12.96
Late Fusion	37.91M	74.85G	58.29	0.011	36.58	59.63
HAttMatting	59.2M	159.7G	44.01	0.007	29.26	46.41
Ours	344.82K	0.93G	**19.59**	**0.006**	**18.70**	**20.34**

In Fig.4, compared to LateFusion, our method is able to find a more complete foreground of the matte objects since our network can exploit multi-scale features to better capture the semantics. Because of our proposed ENA, our method can regress more accurate alpha values in the unknown region.

3.3 Ablation Study

The core idea of our method contains three parts: light-weighted backbone, cross-level fusion module (CFM), and Efficient Non-Local Attention (ENA). To validate the effectiveness of our proposed light-weighted backbone, we replace it with other light-weighted backbones, including MobileNet and ShuffleNet, and results are shown in Tab.2. Compared to these two light-weighted backbones, our proposed backbone can achieve better performance with fewer parameters and FLOPs. Because our backbone can achieve multi-scale feature representations for better capturing sufficient semantics. Moreover, only using CFM or ENA can already heavily improve the performance. Better performance has been achieved through the combination of these two architectures.

Table 2: Ablation study on CFM and ENA. We also replaced our backbone with the existing lightweight backbone.

Backbone	Configurations		Model		Metrics			
	CFM	ENA	# Param.	FLOPs	SAD	MSE	Grad	Conn
Ours	×	×	285.70K	0.63G	56.86	0.019	36.81	48.16
	×	✓	342.70K	0.93G	34.33	0.021	28.24	32.98
	✓	×	287.83K	0.63G	28.90	0.013	24.27	26.35
	✓	✓	344.93K	0.93G	**19.59**	**0.006**	**18.70**	**20.34**
MobileNetV2	✓	✓	37.76M	27.02G	47.21	0.014	30.28	36.46
ShuffleNetV2	✓	✓	30.22M	20.52G	48.72	0.015	33.84	38.21

To further verify the validity and reasonableness of our designed efficient non-local attention module. We trained our model with different types of attention: common non-local

Table 3: Ablation study for different modalities using attention. *S* and *L* means short- and long-range attentions respectively.

Configurations		Model		Metrics			
		# Param.	FLOPs	SAD	MSE	Grad	Conn
Common Non-Local		421.69K	1.33G	25.13	0.010	20.89	22.67
Efficient Non-local(Ours)	S	325.15K	0.82G	25.98	0.011	21.93	23.02
	L($\sqrt{k}$=4)	292.19K	0.72G	27.01	0.013	23.94	24.31
	S+L($\sqrt{k}$=1)	424.11K	1.34G	25.01	0.010	20.31	22.96
	S+L($\sqrt{k}$=2)	371.43K	1.02G	23.27	0.009	19.37	21.61
	S+L($\sqrt{k}$=4)	344.83K	0.93G	**19.59**	**0.006**	**18.70**	**20.34**
	S+L($\sqrt{k}$=8)	340.27K	0.91G	19.90	0.008	19.05	21.35

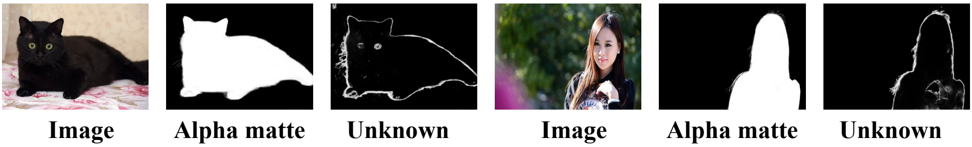


Image **Alpha matte** **Unknown** **Image** **Alpha matte** **Unknown**

Figure 5: The results on real world images and the unknown region predicted by our method.

attention and efficient non-local attention model. As shown in Tab.3, compared to common-local attention, our proposed ENA is not only more efficient but also more usable for matting tasks. Using only short-range attention (S) or long-range attention (L) can improve model performance because they can model the relevance between different pixels, which can help regress alpha values. As can be seen in Tab.3, different sampling intervals $\sqrt{k}$ lead to different results. When $\sqrt{k} = 1, 2$, ENA cannot fully exploit its convenience and effectiveness, because the sampling is too intensive. And the long-range attention will lead to some artifacts because of irrelevant matches. Experimentally, we set $\sqrt{k}$ to 4 in all experiments. For more comprehensive analyses of our proposed method, please refer to the supplementary materials. Fig.5 shows the matting results for internet images for further testing of our method.

4 Conclusion

In this paper, we design a two-stage trimap-free light-weighted natural image matting model, which makes one of the earliest effort on this area. SN is designed to capture sufficient semantics and classify the pixels into unknown, foreground and background regions. MRN aims at capturing detailed texture information and regressing accurate alpha values. With proposed CFM, SN can efficiently utilize multi-scale features with less computational cost. ENA in MRN can efficiently model the relevance between different pixels and help regress high-quality alpha values. Experiment results on Composition-1k testing set demonstrate that our method outperforms state-of-the-art (SOTA) approaches in natural image matting with 1% parameters (344k) of large models. We believe our proposed backbone, CFM and ENA can contribute to other light-weighted computer vision tasks in the future.

References

[1] Yagiz Aksoy, Tunç Ozan Aydin, and Marc Pollefeys. Designing effective inter-pixel information flow for natural image matting. In CVPR, 2017.

[2] Shaofan Cai, Xiaoshuai Zhang, and Haoqiang Fan. Disentangled image matting. In ICCV, 2019.

[3] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. KNN matting. In CVPR, 2012.

[4] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In ACM MM, 2018.

[5] Yunpeng Chen, Haoqi Fan, and Bing Xu. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In ICCV, 2019.

[6] Donghyeon Cho, Yu-Wing Tai, and In-So Kweon. Natural image matting using deep convolutional neural networks. In ECCV, 2016.

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In CVPR, 2017.

[8] Yung-Yu Chuang, Brian Curless, David Salesin, and Richard Szeliski. A bayesian approach to digital matting. In CVPR, 2001.

[9] Mark Everingham, Luc Van Gool, and Christopher K. I. Williams. The pascal visual object classes (VOC) challenge. IJCV., 2010.

[10] Eduardo Simoes Lopes Gastal and Manuel M. Oliveira. Shared sampling for real-time alpha matting. CGF, 29(2), 2010.

[11] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In CVPR, 2011.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016.

[13] Tong He, Zhi Zhang, Hang Zhang, and Zhongyue Zhang. Bag of tricks for image classification with convolutional neural networks. In CVPR, 2019.

[14] Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. In ICCV, 2019.

[15] Andrew G. Howard, Menglong Zhu, and Bo Chen. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017.

[16] Guanqing Hu and James J. Clark. Instance segmentation based semantic matting for compositing applications. In CRV, 2019.

[17] Anat Levin, Dani Lischinski, and Yair Weiss. A closed form solution to natural image matting. In CVPR, 2006.

[18] Bo Li, Zhengxing Sun, Jiagao Hu, and Junfeng Xu. Co-saliency detection via sparse reconstruction and co-salient object discovery. In PCM, 2017.

[19] Bo Li, Zhengxing Sun, and Yuqi Guo. Supervae: Superpixelwise variational autoencoder for salient object detection. In AAAI, 2019.

[20] Bo Li, Zhengxing Sun, Qian Li, Yunjie Wu, and Anqi Hu. Group-wise deep object co-segmentation with co-attention recurrent neural network. In ICCV, 2019.

[21] Bo Li, Zhengxing Sun, Lv Tang, and Anqi Hu. Two-b-real net: Two-branch network for real-time salient object detection. In ICASSP, 2019.

[22] Bo Li, Zhengxing Sun, Lv Tang, Yunhan Sun, and Jinlong Shi. Detecting robust co-saliency with recurrent co-attention neural network. In IJCAI, 2019.

[23] Bo Li, Zhengxing Sun, Quan Wang, and Qian Li. Co-saliency detection based on hierarchical consistency. In ACM MM, 2019.

[24] Bo Li, Zhengxing Sun, Junfeng Xu, Shuang Wang, and Peiwen Yu. Saliency based multiple object cosegmentation by ensemble MIML learning. MTAP, 2020.

[25] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In AAAI, 2020.

[26] Shanchuan Lin, Andrey Ryabtsev, and Soumyadip Sengupta. Real-time high-resolution background matting. CoRR, abs/2012.07810, 2020.

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, and Pietro Perona. Microsoft COCO: common objects in context. In ECCV, 2014.

[28] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In ICLR, 2017.

[29] Hao Lu, Yutong Dai, Chunhua Shen, and Songcen Xu. Indices matter: Learning to index for deep image matting. In ICCV, 2019.

[30] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. Alphagan: Generative adversarial networks for natural image matting. In BMVC, 2018.

[31] Chao Peng, Xiangyu Zhang, and Gang Yu. Large kernel matters - improve semantic segmentation by global convolutional network. In CVPR, 2017.

[32] Thomas K. Porter and Tom Duff. Compositing digital images. In SIGGRAPH, 1984.

[33] Yu Qiao, Yuhao Liu, Xin Yang, and Dongsheng Zhou. Attention-guided hierarchical structure aggregation for image matting. In CVPR, 2020.

[34] Christoph Rhemann, Carsten Rother, and Jue Wang. A perceptually motivated online benchmark for image matting. In (CVPR), 2009.

[35] Soumyadip Sengupta, Vivek Jayaram, and Brian Curless. Background matting: The world is your green screen. In CVPR, 2020.

[36] Ehsan Shahrian, Deepu Rajan, and Brian L. Price. Improving image matting using comprehensive sampling sets. In CVPR, 2013.

[37] Jian Sun, Jiaya Jia, and Chi-Keung Tang. Poisson matting. ACM Trans. Graph., 2004.

[38] Hao Tang, Dan Xu, Nicu Sebe, Yanzhi Wang, Jason J Corso, and Yan Yan. Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In CVPR, 2019.

[39] Hao Tang, Song Bai, and Nicu Sebe. Dual attention gans for semantic image synthesis. In Proceedings of the 28th ACM International Conference on Multimedia, pages 1994–2002, 2020.

[40] Hao Tang, Song Bai, Philip HS Torr, and Nicu Sebe. Bipartite graph reasoning gans for person image generation. In BMVC, 2020.

[41] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. In ECCV, 2020.

[42] Hao Tang, Dan Xu, Yan Yan, Philip HS Torr, and Nicu Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In CVPR, 2020.

[43] Hao Tang, Hong Liu, Dan Xu, Philip HS Torr, and Nicu Sebe. Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks. IEEE Transactions on Neural Networks and Learning Systems, 2021.

[44] Jingwei Tang, Yagiz Aksoy, Cengiz Öztireli, and Markus H. Gross. Learning-based sampling for natural image matting. In CVPR, 2019.

[45] Lv Tang and Bo Li. CLASS: cross-level attention and supervision for salient objects detection. In ACCV, 2020.

[46] Lv Tang, Bo Li, Yanliang Wu, Bo Xiao, and Shouhong Ding. Fast: Feature aggregation for detecting salient object in real-time. In ICASSP, 2021.

[47] Lv Tang, Bo Li, Yijie Zhong, Shouhong Ding, and Mofei Song. Disentangled high quality salient object detection. In ICCV, October 2021.

[48] Jue Wang and Michael Cohen. Optimized color sampling for robust matting. In CVPR, 2007.

[49] Xiaolong Wang and Ross B. Girshick. Non-local neural networks. In CVPR, 2018.

[50] Ning Xu, Brian L. Price, Scott Cohen, and Thomas S. Huang. Deep image matting. In CVPR, 2017.

[51] Donggeun Yoon and Jinsun Park. Lightweight deep CNN for natural image matting via similarity-preserving knowledge distillation. SPL, 2020.

[52] Xiangyu Zhang, Xinyu Zhou, and Mengxiao Lin. Shufflenet:an extremely efficient convolutional neural network for mobile devices. In CVPR, 2018.

[53] Yunke Zhang and Lixue Gong. A late fusion CNN for digital matting. In CVPR, 2019.

[54] Yuanjie Zheng and Chandra Kambhamettu. Learning based digital matting. In ICCV, 2009.

[55] Bingke Zhu, Yingying Chen, Jinqiao Wang, and Si Liu. Fast deep matting for portrait animation on mobile phone. In ACM MM, 2017.