Lightweight Image Matting via Efficient Non-Local Guidance

Zhaoxiang Kang, Zonglin Li, Qinglin Liu, Yuhe Zhu, Hongfei Zhou, and Shengping Zhang

School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China
{zhaoxiang.kang, qinglin.liu}@outlook.com

Abstract. Natural image matting aims to estimate the opacity of foreground objects. Most existing approaches involve prohibitive parameters, daunting computational complexity, and redundant dependency. In this paper, we propose a lightweight matting method termed LiteMatting, which learns the local smoothness of color space and affinities between neighboring pixels to estimate the alpha mattes. Specifically, a modified mobile block is adopted to construct an encoder-decoder framework, which reduces parameters while retaining sufficient spatial and channel information. In addition, a Long-Short Range Pyramid Pooling Module (LSRPPM) is introduced to extend the reception field by capturing long-range dependency between regions distributed discretely. Finally, an Efficient Non-Local Block (ENB) is presented for guiding high-level semantics propagation from low-level detail features to refine the alpha mattes. Extensive experiments demonstrate that our method achieves a favorable trade-off between accuracy and efficiency. Compared with most state-of-the-art approaches, our method attains an immense descent in parameters and FLOPs with 30% and 13%, respectively, while achieving an improvement of over 15% in SAD metrics. Code and model are available at https://github.com/kzx2018/LiteMatting.

Keywords: Image Matting · Lightweight · Efficient Non-Local

1 Introduction

Natural image matting aims to estimate the opacity mask and has many applications, such as photo editing, compositing, and film production [1–5]. Mathematically, the observed image I is defined as a convex combination of the foreground image F and the 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 denotes the opacity of the foreground object at pixel i. This is a seriously ill-posed problem since I is known but F, B, and α are unknown. To address this problem, most existing methods take a trimap as additional input. However, there are still potential challenges in current approaches.

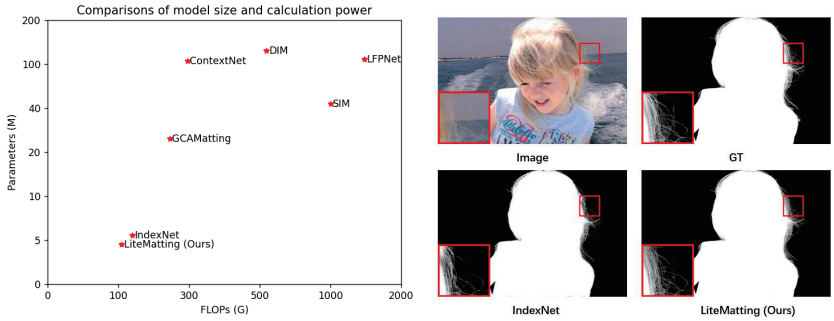

Fig. 1. Comparisons with SOTA methods. Fewer parameters and FLOPs mean that the method has better efficiency. IndexNet is very close to LiteMatting (Ours) in terms of efficiency, but our proposed method performs superiorly by viewing.

On the one hand, the most common way to improve matting performance is by scaling up their network depth and width, which is proved through subsequent work. For instance, Deep Image Matting [6] consists of an encoder-decoder stage and a refining stage. Context-Aware Matting [7] employs two encoders to predict both foreground and alpha mattes. GCA Matting [8] estimates the alpha mattes with a guided contextual attention mechanism. Networks mentioned above need more layers to increase the reception field and more channels to capture more fine-grained patterns, which do not avoid prohibitive parameters and computational complexity. Simultaneously, the vast GPU memory occupation and daunting computational cost hinder their usage in real application scenarios. Therefore, it is crucial to explore a lightweight image matting method when applied to devices with limited storage space and computing power.

On the other hand, constructing lightweight models require trading accuracy for efficiency. For example, IndexNet [9] using a learnable index pooling only captures local features due to the small reception field, which generates inaccurate alpha mattes. It is a challenge that achieves an accuracy gain on matting with better efficiency because of weights simplification and expensive tuning costs.

To address the first problem, we present a lightweight network LiteMatting where the opacity messages are transmitted efficiently across different semantic levels. We modify the original mobile block [10] via the group norm and ReLU/LeakyReLU. The modified mobile block employs depthwise filters with inverted residuals [11] and linear bottlenecks to substantially reduce the memory footprint needed and improve the computational efficiency. Also, our proposed network systematically explores the local smoothness assumptions [12] and extracts affinities between neighboring pixels to model the matte gradient intrinsically. Furthermore, we present a Long-Short Range Pyramid Pooling Module (LSRPPM) that utilizes an adaptive sampling strategy to gather the informative context with multi-scale kernels as a global prior. It improves the capability of modeling the long-range dependencies to extend the reception fields [13] and contributes to eliminating redundant perception information.

For the second problem, we propose an Efficient Non-Local Block (ENB) to refine the alpha mattes. To be specific, ENB models the spatial relevance of different pixels by guiding high-level semantics propagation from low-level details where pixels share similar texture features, which can strengthen the discrimination of feature representation and help refine blurring artifacts. To decrease the memory consumption of common non-local block [14], it introduces a pyramid sampling strategy to reduce the computational overhead of matrix multiplication. It also performs superiorly against the non-local block because the resulted sampling points are more informative from corresponding feature aggregation. ENB dramatically improves efficiency and achieves a considerable accuracy gain. As shown in Fig. 1, our proposed method is more lightweight and efficient than other trimap-based matting methods. Overall, our contributions can be summarized in the following aspects:

- We present a lightweight image matting architecture based on the modified mobile block and leverage LSRPPM to extend the reception field, which expands to more widespread application scenarios.
- We propose ENB that reduces the memory consumption of non-local blocks, which achieves a remarkable accuracy gain by guiding high-level semantics propagation from low-level detail features.
- We conduct extensive experiments on the Adobe Composition-1K dataset and AlphaMatting testing sets, which demonstrates that our method achieves a satisfactory trade-off between accuracy and efficiency.

2 Related Work

Natural image matting approaches are roughly categorized into sampling-based, propagation-based, and learning-based methods.

Sampling-based methods. Sampling-based methods [15–21] usually sample nearby the foreground and background colors for each unknown pixel, then design metrics to use the similarity of the pixels to estimate the alpha matte. Bayesian Matting [19] uses a well-defined Bayesian framework to predict the alpha value. Robust Matting [20] considers the spatial information and selects samples along the boundaries with confidence. Global Matting [21] samples all pixels in the image to prevent missing information.

Propagation-based methods. Propagation-based methods [22–28] are also known as affinity-based methods, which allow propagation of the alpha values from the known foreground and background regions to unknown regions. Closed-form Matting [26] establishes a linear system to find the optimal solution by smoothness assumption on the foreground and background colors. KNN Matting [27] globally collects K nearest neighbors to increase the speed meanwhile keeping the accuracy of matting. Information-flow Matting [28] combines the local and non-local affinities of colors with spatial smoothness.

Learning-based methods. Learning-based methods [6–9, 29–38] utilize a deep network to directly estimate the alpha matte with the given image and trimap. DIM [6] provides the first large-scale image dataset and presents the first

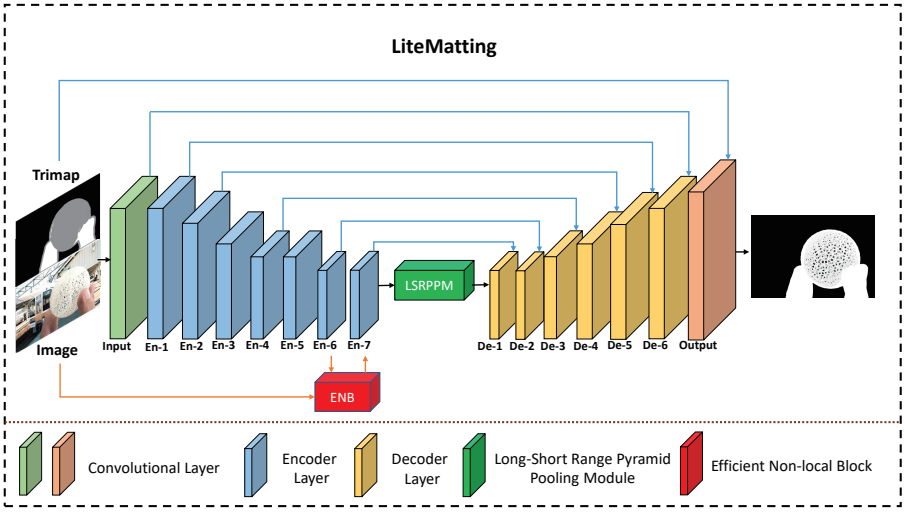

Fig. 2. The overview of the LiteMatting Network. Our proposed network consists of encoder, decoder, LSRPPM, and ENB four parts.

end-to-end matting architecture with a refinement network. AlphaGan [29] introduces GAN into the image matting task. Context-Aware Matting [7] proposes a dual encoder-decoder structure to capture semantic information for foreground and alpha prediction. AdaMatting [30] uses trimap adaptation to refine the alpha matte. IndexNet [9] utilizes the pooling indices for unpooling operation. GCA Matting [8] presents a guided contextual attention mechanism to analyze image inpainting processing in matting. HDMatt [31] designs a cross-patch contextual module to improve accuracy in patch-based inference.

3 Proposed Method

In this section, we will first introduce the backbone of our network architecture and then illustrate the Long-Short Range Pyramid Pooling Module. Afterward, we will present the design of the Efficient Non-Local Block. Finally, we will describe the loss function.

3.1 Network Architecture

We construct an encoder-decoder architecture like U-net [39], which is illustrated in Fig. 2. Our proposed network is formed by stacking the modified mobile block. It utilizes the depthwise separable convolution to reduce parameters and computation, which splits convolution into two separate layers called Depthwise Conv and Pointwise Conv. The former performs light-weighted filtering by a single convolution per input channel, and the latter builds new features through

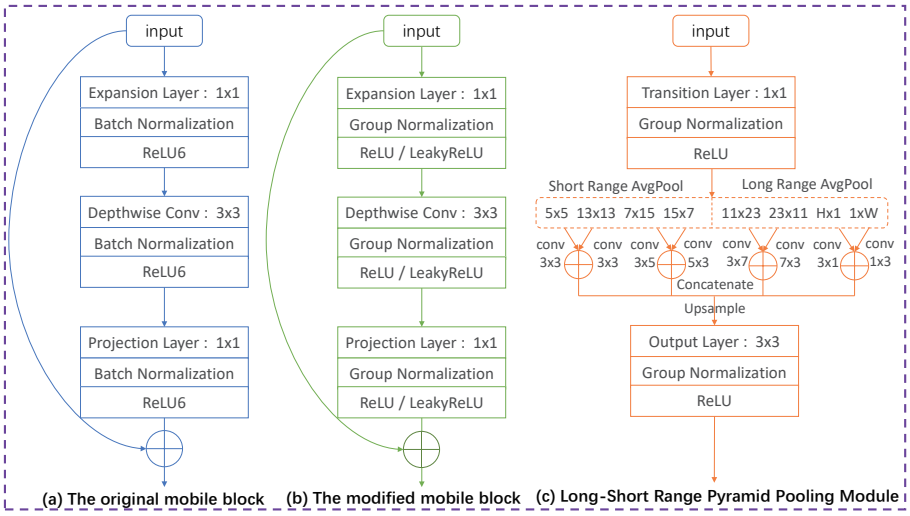

Fig. 3. Difference between the original mobile block and the modified mobile block. The detailed structure of the Long-Short Range Pyramid Pooling Module.

the linear combinations of the input channel. Specifically, the modified mobile block takes a low-dimensional compressed representation as an input, which is expanded to high-dimensional and filtered with a lightweight Depthwise Conv. These features are projected back to low-dimensional with a linear convolution at last. It reduces the parameters while retaining sufficient spatial and channel information. Overall, the network can efficiently explore the local smoothness of color space and learn affinities between neighboring pixels.

The encoder. Firstly, the input layer is a conventional convolution layer that increases the number of input channels from 3 to 8 with a given trimap and a transformable map. The transformable map uses Gaussian blurs of the definite foreground and background masks at a prior scale to encode the given trimap [40]. Secondly, in contrast to the original mobile block [10], the modified mobile block replaces ReLU6 and the batch norm [41] with ReLU and the group norm, respectively, which is shown in Fig. 3 (a & b). It benefits accelerating regression convergence and increasing accuracy. Moreover, ENB is embedded in the encoder to guide the information flow for refining the alpha mattes. Finally, the encoder has seven levels named En-1 to En-7, which help extract context features and propagate the semantic information to the decoder.

The decoder. The decoder consists of the modified mobile block followed by up-sampling layers, which differs from the encoder because it employs the LeakyReLU instead of ReLU to avoid the dead ReLU issue. Specifically, the decoder first receives global priors from LSRPPM. Then it leverages six levels named De-1 to De-6 to upsample rich context features to the original size while fusing semantic information from each encoder. Finally, the output layer stacks three convolutional layers to estimate the alpha mattes.

(a) The Short-Range Block (b) The Long-Range Block

Fig. 4. Schematic illustration of the Short-Range Block and the Long-Range Block.

3.2 Long-Short Range Pyramid Pooling Module

The bottleneck layer [42, 43] between the encoder and the decoder simulates the receptive field of human vision to enhance the feature extraction capability of the network. We present the Long-Short Range Pyramid Pooling Module based on [43] to enlarge the receptive field of the network.

LSRPPM first takes the high-level feature from En-7 as input to feed into eight parallel pathways, each of which contains a different scale pooling layer followed by a convolution with a narrow kernel shape to obtain multiple representations. Afterward, it concatenates and upsamples them to the same size as the input. Finally, it generates a composite feature that combines multiple scales as output. Fig. 3 (c) depicts the detail of above steps.

LSRPPM is divided into the short-range block and the long-range block. The former extracts global information by pooling operations at four short-range scales (5×5, 13×13, 7×15, 15×7). However, there are limitations in capturing wide context scenes since the observed target may have a long-range structure (e.g., the cobweb in Fig. 4). Using short rectangle pooling windows cannot deal with this issue well, so we design the latter to capture long-range dependency by a longer pooling kernel. Inspired by [44], the long-range block expands long-range scales (11×23, 23×11, H×1, 1×W) layers, where H and W are the spatial height and width, respectively. It improves the capability of capturing dependencies between regions distributed discretely and avoids contaminating information from irrelevant regions. Mathematically, given the two-dimensional tensor $\mathbf{x} \in \mathcal{R}^{H \times W}$, the output $\mathbf{y}^h \in \mathcal{R}^H$ from horizontal pooling (H×1) can be written as

$$y_i^h = \frac{1}{W} \sum_{0 \le j < W} x_{i,j} \tag{2}$$

Similarly, the output $\mathbf{y}^v \in \mathcal{R}^W$ from vertical pooling (1×W) can be written as

$$y_j^v = \frac{1}{H} \sum_{0 \le i < H} x_{i,j} \tag{3}$$

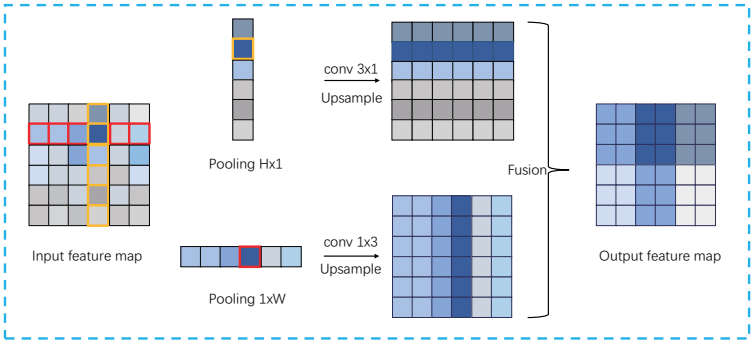

Fig. 5. The illustration of the Long-Range Block ($H \times 1, 1 \times W$). It builds long-range dependencies between regions distributed discretely to extend the reception field.

LSRPPM merges $\mathbf{y}^h$ and $\mathbf{y}^v$ together to obtain more useful global priors, which is shown in Fig. 5. Afterward, it repeats the same operation for other scales as in horizontal and vertical pooling layers. Finally, it uses multi-scale feature aggregation to fuse contextual information. Overall, LSRPPM builds long-range dependencies to extend the reception field and eliminates redundant information, which is essential for improving performance.

3.3 Efficient Non-Local Block

Most deep learning methods predict the alpha matte by learning the propagation of pixels in known regions to unknown regions according to their similarity of opacity. However, pixels in the unknown regions cannot be correlated with pixels in known regions because of the locality of the convolutional neural network. Non-Local [14] skillfully leverages the global dependencies to capture the relationship between pixels, which is beneficial to the matting task.

In fact, a common non-local operation is very time and memory-consuming, which is shown in Fig. 6 (a). Firstly, it takes feature $X \in \mathcal{R}^{N \times H \times W \times C}$ as an input, where N, C, W, H indicate the batch, channel, width and height, respectively. Secondly, using convolutions W_θ, W_φ, W_g transforms X for obtaining the output of three embeddings θ, φ and g as

$$\theta = W_\theta(X), \varphi = W_\varphi(X), g = W_g(X) \tag{4}$$

Thirdly, the similarity matrix M is generated by the matrix multiplication and normalization, and then multiplied by g to obtain the attention layer A as

$$M = Softmax(\theta^{\mathrm{T}} \times \varphi) \tag{5}$$

$$A = M \times g^{\mathrm{T}} \tag{6}$$

Finally, it uses a weight parameter W_y to adjust the importance of the attention layer and merges the original input X, the final output Y is given by

$$Y = W_y\left(A^{\mathrm{T}}\right) + X \tag{7}$$

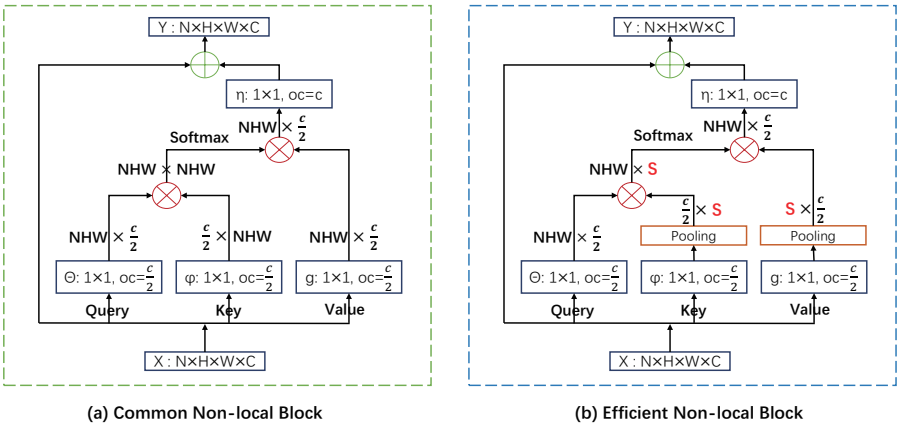

Fig. 6. Difference between Common Non-Local Block and Efficient Non-Local Block.

We clearly find that the matrix multiplication of Eq.(5) and Eq.(6) dominate the heavy computation. The straightforward pipeline is

$$\underbrace{\mathcal{R}^{NHW\times\frac{C}{2}}\times\mathcal{R}^{\frac{C}{2}\times NHW}}_{Eq.(5)}\rightarrow\underbrace{\mathcal{R}^{NHW\times NHW}\times\mathcal{R}^{NHW\times\frac{C}{2}}}_{Eq.(6)}\rightarrow\mathcal{R}^{NHW\times\frac{C}{2}} \tag{8}$$

Inspired by [45], we choose the pyramid pooling operation (scales=1, 3, 6, 8) to reduce the inefficiency of the non-local block. As shown in Fig. 6 (b), it samples more important features after φ and g to filter out irrelevant information by changing NHW to number S ($S \ll NHW$), the pipeline after pooling is

$$\underbrace{\mathcal{R}^{NHW\times\frac{C}{2}}\times\mathcal{R}^{\frac{C}{2}\times S}}_{Eq.(5)}\rightarrow\underbrace{\mathcal{R}^{NHW\times S}\times\mathcal{R}^{S\times\frac{C}{2}}}_{Eq.(6)}\rightarrow\mathcal{R}^{NHW\times\frac{C}{2}} \tag{9}$$

According to the above method, ENB reduces the computational overhead of matrix multiplication to improve the efficiency of the non-local block. Pixels sharing similar texture information have similar opacity features [46]. However, blurs occur in the representation of each pixel as the network layers go deep. Therefore, ENB utilizes the low-level feature to guide high-level semantics propagation to increase the accuracy of the alpha mattes.

In our approach, ENB first receives a high-level feature X_h from the encoder En-6. Then it multiplies X_h with the values U involved in the pixels of the unknown area and known foreground object for adjusting weights to get the query feature, named as X_q. The key/value feature is extracted from the input image by the low-level feature guidance block, named as X_k and X_v.

The X_k and X_v are filtered to the sufficient feature statistics about global semantic cues by the pyramid pooling meanwhile decreasing the computational cost of non-local. ENB utilizes the similarity of the X_k and X_q by the related matrix and takes the normalizing function applied to them. Afterward, it multiplies X_v with the result from the previous step and merges the origin value X_h

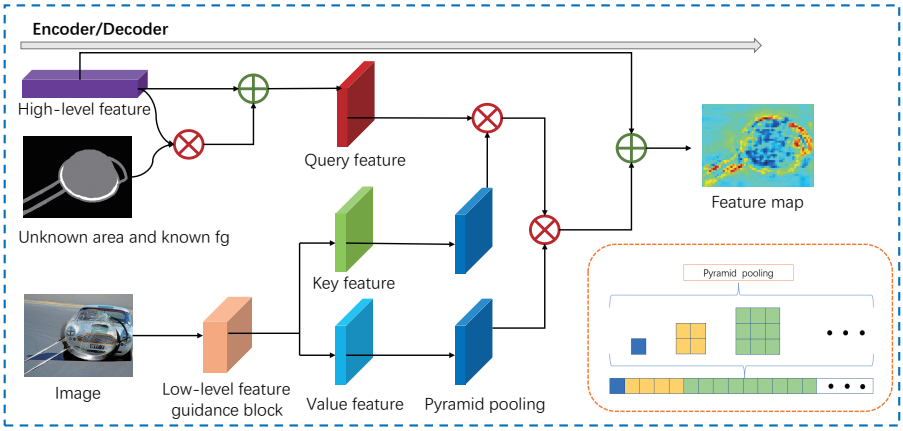

Fig. 7. The architecture of the Efficient Non-Local Block. ENB guides high-level semantics propagation from low-level features and introduces pyramid pooling to reduce the computational overhead of the common non-local block.

to obtain the output feature maps. Finally, it feeds the output into the encoder En-7 to strengthen the discrimination of feature representation. The architecture of our proposed ENB is depicted in Fig. 7. Overall, ENB successfully achieves a considerable improvement of alpha mattes with high efficiency.

3.4 Loss Function

A combination of loss functions is used in the network training, including the reconstruction $\mathcal{L}_1$ loss named $\mathcal{L}_u$, the composition loss $\mathcal{L}_c$ and the Laplacian pyramid loss $\mathcal{L}_{\text{lap}}$. Given the original image C, the estimated alpha matte α, and the ground truth α^{gt}, F^{gt}, B^{gt}. The loss $\mathcal{L}_u$ is defined as

$$\mathcal{L}_u = \frac{1}{|\mathcal{T}_u|} \sum_{i \in \mathcal{T}_u} \left\| \alpha_i - \alpha_i^{gt} \right\|_1 \tag{10}$$

where $\mathcal{T}_u$ is the set of unknown pixels in the trimap. The loss $\mathcal{L}_c$ is defined as

$$\mathcal{L}_c = \sum_i \left\| C_i - \alpha_i F_i^{gt} - (1 - \alpha_i) B_i^{gt} \right\|_1 \tag{11}$$

The Laplacian pyramid loss is calculated by the Laplacian pyramid L^s_{pyr} with multiple scales s [47], which is defined as

$$\mathcal{L}_{\text{lap}} = \sum_{s=1}^{5} 2^{s-1} \left\| L^s_{\text{pyr}}(\alpha) - L^s_{\text{pyr}}(\alpha^{gt}) \right\|_1 \tag{12}$$

Finally, our total loss function is computed as

$$\mathcal{L}_\alpha = \lambda_1 \mathcal{L}_u + \lambda_2 \mathcal{L}_c + \lambda_3 \mathcal{L}_{\text{lap}} \tag{13}$$

where λ_1, λ_2, λ_3 are proportion factor to balance the loss function weights.

4 Experiments

4.1 Datasets

Adobe Composition-1k. The Composition-1k testing set contains 1,000 composed images with a unique trimap. These images are synthesized by 50 foreground images and 1,000 background images from the PASCAL VOC dataset. **AlphaMatting.** The AlphaMatting dataset is a matting dataset that consists of real-world images for the online benchmark. There are eight testing images, which of each has three different trimaps (i.e. 'small', 'large', and 'user').

4.2 Implementation Details

We train our network on the Adobe Composition-1K training dataset with end-to-end mode. We use several common data augmentation ways [48], including affine transformation, flip transformation, contrast transformation, saturation transformation, and random foreground composition. Images are cropped into patch of dimensions 1,024×1,024. In addition, their trimaps are generated by the alpha matte ground truth with random erosion and dilation of 3 to 35 pixels. We use the RAdam optimizer with β_1=0.5 and β_2=0.999. The learning rate is initialized as 5×10^{-5} and proportion factor of loss function λ_1, λ_2, λ_3 are set to 1. We train our network for 150 epochs with a batch size of 4. And our model is trained from scratch without any pretrained models.

4.3 Comparison with the SOTA methods

There are four metrics used in the evaluation: the sum of absolute difference (SAD), the mean square error (MSE), the gradient error (Grad), and the connectivity error (Conn). Furthermore, we count the number of parameters and computational cost at 1,024×1,024 resolution as shown in Table 1.

Table 1. The quantitative results on the Adobe Composition-1k testing set. The best results are highlighted in bold.

Methods	SAD	MSE	Grad	Conn	Params	FLOPs
KNN Matting	175.4	103.0	124.1	176.4	-	-
Closed-Form	168.1	91.0	126.9	167.9	-	-
DIM	50.4	14.0	31.0	50.8	130.6M	511.0G
IndexNet	45.8	13.0	25.6	43.7	6.0M	116.6G
ContextNet	35.8	8.2	17.3	33.2	107.5M	292.5G
GCAMatting	35.3	9.1	16.9	32.5	25.3M	257.3G
LiteMatting (Ours)	30.1	6.1	13.1	26.2	**4.3M**	**101.5G**
SIM	27.7	5.6	10.7	24.4	44.5M	1001.9G
LFPNet	**23.6**	**4.1**	**8.4**	**18.5**	112.2M	1539.4G

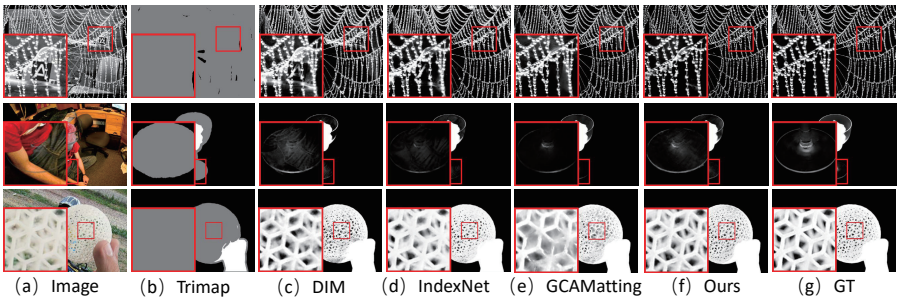

Fig. 8. The qualitative comparison results on the Adobe Composition-1k testing set.

We compare our model with traditional matting methods: KNN Matting [27], Closed-Form [26], and learning-based methods: DIM [6], IndexNet [9], ContextNet [7], GCAMatting [8], SIM [32] and LFPNet [48] on Composition-1k testing set. In addition, we also compare with AdaMatting [30], A2U Matting [49] and SampleNet [50] on the AlphaMatting benchmark. To be specific, we utilize the patch-based crop-and-stitch method for inference [31], where the images are cropped into patches and then fed into the network.

As shown in Table 1 and Table 2, our LiteMatting attains an immense descent in parameters and FLOPs with 30% and 13%, respectively, while achieving an improvement of over 15% in SAD metrics of most methods (e.g., GCAMatting). Although our method performs inferior to SIM and LFPNet in terms of SAD, we are more lightweight (at 3.8%-9.7% in Params and 6.6%-10.1% in FLOPs) than them. It shows that our method has effectiveness in practical application scenarios, especially in resource-limited environments. As shown in Fig. 8 and Fig. 9, we qualitatively compare with SOTA approaches. Our method generates more meticulous alpha mattes in background interference and shows robustness in performance where it estimates the opacity of foreground objects.

Table 2. Our average ranking results on the AlphaMatting testing set. S, L, and U denote the small trimap, large trimap, and user trimap respectively. The best results are highlighted in bold.

Methods	SAD				MSE				Grad			
	Overall	S	L	U	Overall	S	L	U	Overall	S	L	U
Ours	**13.8**	**12.5**	**11.0**	**17.8**	**14.6**	13.1	**11.6**	**19.0**	**12.2**	**10.4**	**8.8**	**17.5**
AdaMatting	15.2	13.6	14.1	**17.8**	16.0	13.1	14.9	19.9	16.0	11.5	13.8	22.6
A2U Matting	15.4	14.0	12.8	19.4	18.2	15.8	15.3	23.6	14.9	13.9	11.9	19.0
SampleNet	15.8	12.8	16.0	18.8	16.7	**12.6**	17.4	20.0	18.2	13.8	16.3	24.6
GCAMatting	17.3	18.0	15.3	18.5	18.3	18.1	17.3	19.4	17.0	17.1	15.9	18.1
DIM	19.2	20.1	18.8	18.8	22.2	20.4	21.4	24.8	27.0	24.0	23.9	33.0
IndexNet	22.5	24.4	21.5	21.5	26.5	29.0	25.1	25.3	22.1	20.6	21.1	24.6

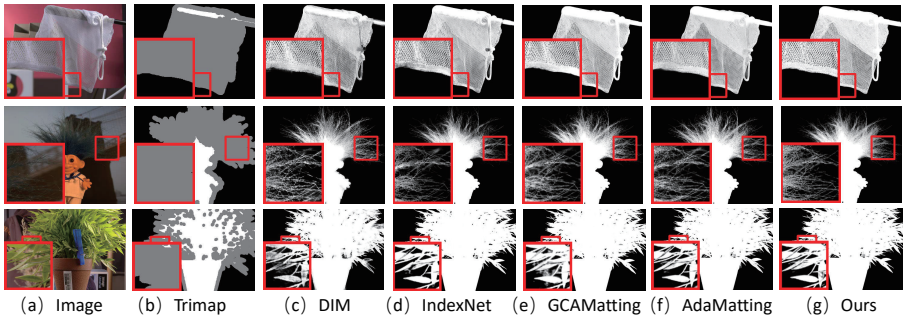

Fig. 9. The qualitative comparison results on the AlphaMatting testing set.

Finally, we pay attention to the real-world high-resolution images. In Fig. 10, we test DIM [6], IndexNet [9], GCAMatting [8] and our proposed LiteMatting. These images are too large to be fed into a single GPU, so we implement inference on the CPU to avoid insufficient memory. It has to spend a long inference time for each image where the usage scenarios for high-resolution images are limited. The results demonstrate that our method extracts finer details and outperforms other state-of-the-art matting methods with a far faster inference speed. In conclusion, we achieve a promising trade-off between accuracy and efficiency through the above experiments.

4.4 Ablation Study

To validate the efficacy of our proposed backbone based on the modified mobile block, we first compare it to other light-weighted backbones, including ShuffleNet [51], EfficientNet [52] and ConvNetXt [53]. These backbones are known for being lightweight and efficient. As shown in Table 3, we notice that our backbone achieves better performance with fewer parameters and FLOPs than others while training for the same epochs.

Afterward, we confirm that the group norm is more suitable for matting than the batch norm. The matting task demands pixel-level relationships, but

Table 3. Ablation study of existing lightweight backbones, normalization method, and loss function on the Adobe Composition-1k testing set.

Backbone	Normalization	Loss Function	SAD	MSE	Params	FLOPs
ShuffleNet	GroupNorm	Alpha Loss	37.3	8.3	10.6M	45.2G
EfficientNet			38.7	9.1	4.7M	55.0G
ConvNetXt			39.9	9.5	5.2M	52.0G
Ours	GroupNorm	Alpha Loss	36.8	7.9	3.6M	52.8G
	BatchNorm	Alpha Loss	37.9	8.6	3.6M	51.7G
	GroupNorm	F, B, A Loss	40.6	10.3	3.6M	52.8G

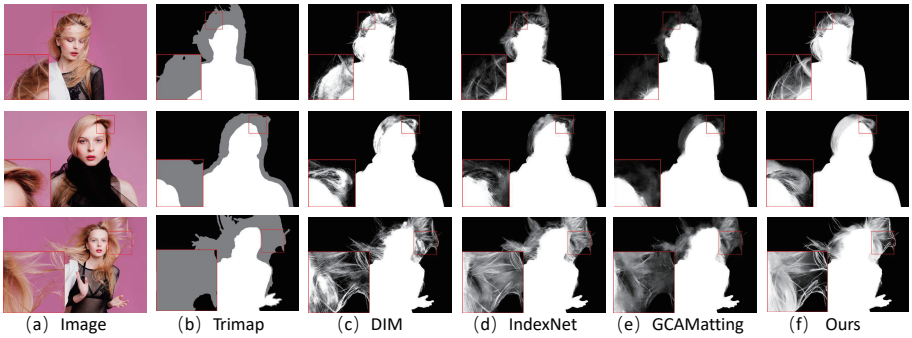

Fig. 10. The qualitative comparison results on the real-world high resolution images. Image sizes from top to bottom: 5863×3909, 5636×3757, 6240×4160.

the high-resolution training samples lead to a mini-batch size of 1-16 in GPU. The group norm contributes to increasing the accuracy of alpha mattes since it is an expert in dealing with the mini-batch size questions. We also attempt to use F, B, A loss [54] which means training with the loss of foreground and background in addition to the alpha, but the result gets worse than training with the Alpha Loss. We think it may be easy to learn redundant information for a lightweight model because of the irrelevant feature interference. Therefore, we choose the modified mobile block and the group norm to build up our backbone, training our network with the alpha loss.

Moreover, we further verify the reasonableness of our efficient non-local block. We compare our model with the common non-local block and then explore different sampling methods (max pooling, average pooling, and pyramid pooling) on ENB. These non-local blocks help gain a significant performance improvement through our ablation experiments. It is effective for image matting to build a long-range contextual dependency by modeling the relevance between pixels. As shown in Table 4, compared to the common non-local block, ENB based on the pyramid pooling not only regresses more accurate alpha values but also performs more efficiently by reducing the FLOPs. The reason is that the sampling points are more informative by receiving the provided features from the pooling kernel. Then we conduct several experiments to study the effect of sampling strate-

Table 4. Ablation study of Efficient Non-Local Block in terms of different sampling methods. 's' represents the scale of a pooling layer.

Configuration	Sampling Method	SAD	MSE	FLOPs
Common Non-Local Block	-	31.3	6.6	144.2G
Efficient Non-Local Block (Ours)	max pooling (s=15)	32.0	6.9	101.4G
	average pooling (s=15)	31.6	6.8	101.4G
	pyramid pooling (s=1, 3, 6, 8)	30.1	6.1	101.5G

Table 5. Ablation study of LSRPPM and ENB on the Adobe Composition-1k testing set. The best results are highlighted in bold.

Backbone	LSRPPM	ENB	SAD	MSE	Grad	Conn	Params	FLOPs
Ours			36.8	7.9	19.4	34.0	**3.6M**	**52.8G**
	✓		34.2	7.2	16.1	31.0	3.8M	53.0G
	✓	✓	**30.1**	**6.1**	**13.1**	**26.2**	4.3M	101.5G

gies by altering the scales of pooling layers. ENB based on the pyramid pooling (scales=1, 3, 6, 8) performs better than others based on the max pooling and the average pooling. The second ablation experiments show that ENB improves accuracy and decreases the computational cost of the model.

In the final ablation study, we reveal the effectiveness of each component in LiteMatting on the Adobe Composition-1k testing set. As shown in Table 5, we make a remarkable gain in the accuracy with the combination of LSRPPM and ENB. They are not only efficient but also help refine the alpha mattes. LSRPPM captures the long-range context features and utilizes multi-scale information to enlarge the reception field. Furthermore, ENB guides rich high-level semantic information from low-level details by modeling the relevance between different pixels for regressing the high-precision alpha mattes. And we think ENB that improves performance is positive for light-weighted matting despite the increase in computation. These two architectures are verified to be beneficial to the prediction of the alpha mattes. In conclusion, the ablation study suggests that our method is effective for the image matting task.

5 Conclusion

In this paper, we propose a lightweight method termed LiteMatting for image matting. Our method leverages the modified mobile block to extract sufficient spatial and channel representations with fewer parameters. LSRPPM captures the long-range dependencies to extend the reception field. ENB guides high-level semantics propagation from low-level detail features. With the proposed LSRPPM, ENB can efficiently estimate more accurate alpha mattes with less computational cost. Extensive experiments on the Adobe Composition-1k and AlphaMatting testing set demonstrate that our LiteMatting is more lightweight and performs superiorly against most SOTA approaches, which attains an immense descent in parameters and FLOPs with 30% and 13%, respectively, while achieving an improvement of over 15% in SAD metrics. Overall, we successfully achieve a trade-off between accuracy and efficiency.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (Nos. 61872112 and 6207071409) and the Taishan Scholars Program of Shandong Province (No. tsqn201812106).

References

1. Sengupta, S., Jayaram, V., Curless, B., Seitz, S.M., Kemelmacher-Shlizerman, I.: Background matting: The world is your green screen. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 2291–2300
2. Lin, S., Ryabtsev, A., Sengupta, S., Curless, B.L., Seitz, S.M., Kemelmacher-Shlizerman, I.: Real-time high-resolution background matting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 8762–8771
3. Lin, S., Yang, L., Saleemi, I., Sengupta, S.: Robust high-resolution video matting with temporal guidance. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. (2022) 238–247
4. Chen, Y., Guan, J., Cham, W.K.: Robust multi-focus image fusion using edge model and multi-matting. IEEE Transactions on Image Processing **27** (2017) 1526–1541
5. Ke, Z., Li, K., Zhou, Y., Wu, Q., Mao, X., Yan, Q., Lau, R.W.: Is a green screen really necessary for real-time portrait matting? arXiv preprint arXiv:2011.11961 (2020)
6. Xu, N., Price, B., Cohen, S., Huang, T.: Deep image matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 2970–2979
7. Hou, Q., Liu, F.: Context-aware image matting for simultaneous foreground and alpha estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 4130–4139
8. Li, Y., Lu, H.: Natural image matting via guided contextual attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 34. (2020) 11450–11457
9. Lu, H., Dai, Y., Shen, C., Xu, S.: Indices matter: Learning to index for deep image matting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 3266–3275
10. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 4510–4520
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
12. Wang, R., Xie, J., Han, J., Qi, D.: Improving deep image matting via local smoothness assumption. arXiv preprint arXiv:2112.13809 (2021)
13. Liu, Y., Yu, J., Han, Y.: Understanding the effective receptive field in semantic image segmentation. Multimedia Tools and Applications **77** (2018) 22159–22171
14. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7794–7803
15. Ruzon, M.A., Tomasi, C.: Alpha estimation in natural images. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662). Volume 1., IEEE (2000) 18–25
16. Gastal, E.S., Oliveira, M.M.: Shared sampling for real-time alpha matting. In: Computer Graphics Forum. Volume 29., Wiley Online Library (2010) 575–584
17. Guan, Y., Chen, W., Liang, X., Ding, Z., Peng, Q.: Easy matting-a stroke based approach for continuous image matting. In: Computer Graphics Forum. Volume 25., Wiley Online Library (2006) 567–576

18. Feng, X., Liang, X., Zhang, Z.: A cluster sampling method for image matting via sparse coding. In: European Conference on Computer Vision, Springer (2016) 204–219
19. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. Volume 2., IEEE (2001) II–II
20. Wang, J., Cohen, M.F.: Optimized color sampling for robust matting. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2007) 1–8
21. He, K., Rhemann, C., Rother, C., Tang, X., Sun, J.: A global sampling method for alpha matting. In: CVPR 2011, IEEE (2011) 2049–2056
22. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. In: ACM SIGGRAPH 2004 Papers. (2004) 315–321
23. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: 2007 IEEE 11th International Conference on Computer Vision, IEEE (2007) 1–8
24. Levin, A., Rav-Acha, A., Lischinski, D.: Spectral matting. IEEE transactions on pattern analysis and machine intelligence **30** (2008) 1699–1712
25. Lee, P., Wu, Y.: Nonlocal matting. In: CVPR 2011, IEEE (2011) 2193–2200
26. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE transactions on pattern analysis and machine intelligence **30** (2007) 228–242
27. Chen, Q., Li, D., Tang, C.K.: Knn matting. IEEE transactions on pattern analysis and machine intelligence **35** (2013) 2175–2188
28. Aksoy, Y., Ozan Aydin, T., Pollefeys, M.: Designing effective inter-pixel information flow for natural image matting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 29–37
29. Lutz, S., Amplianitis, K., Smolic, A.: Alphagan: Generative adversarial networks for natural image matting. arXiv preprint arXiv:1807.10088 (2018)
30. Cai, S., Zhang, X., Fan, H., Huang, H., Liu, J., Liu, J., Liu, J., Wang, J., Sun, J.: Disentangled image matting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 8819–8828
31. Yu, H., Xu, N., Huang, Z., Zhou, Y., Shi, H.: High-resolution deep image matting. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 35. (2021) 3217–3224
32. Sun, Y., Tang, C.K., Tai, Y.W.: Semantic image matting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 11120–11129
33. Liu, Y., Xie, J., Shi, X., Qiao, Y., Huang, Y., Tang, Y., Yang, X.: Tripartite information mining and integration for image matting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 7555–7564
34. Jiang, W., Yu, D., Xie, Z., Li, Y., Yuan, Z., Lu, H.: Trimap-guided feature mining and fusion network for natural image matting. arXiv preprint arXiv:2112.00510 (2021)
35. Cheng, H., Xu, S., Jiang, X., Wang, R.: Deep image matting with flexible guidance input. arXiv preprint arXiv:2110.10898 (2021)
36. Goel, A., Kumar, M., Sudheendra, P., Team, V., et al.: Iamalpha: Instant and adaptive mobile network for alpha matting. (2021)
37. Liu, Y., Xie, J., Qiao, Y., Tang, Y., Yang, X.: Prior-induced information alignment for image matting. IEEE Transactions on Multimedia (2021)

38. Dai, Y., Price, B., Zhang, H., Shen, C.: Boosting robustness of image matting with context assembling and strong data augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 11707–11716
39. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241
40. Le, H., Mai, L., Price, B., Cohen, S., Jin, H., Liu, F.: Interactive boundary prediction for object selection. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 18–33
41. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). (2018) 3–19
42. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). (2018) 801–818
43. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 2881–2890
44. Hou, Q., Zhang, L., Cheng, M.M., Feng, J.: Strip pooling: Rethinking spatial pooling for scene parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 4003–4012
45. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 593–602
46. Zhong, Y., Li, B., Tang, L., Tang, H., Ding, S.: Highly efficient natural image matting. arXiv preprint arXiv:2110.12748 (2021)
47. He, K., Sun, J., Tang, X.: Fast matting using large kernel matting laplacian matrices. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE (2010) 2165–2172
48. Liu, Q., Xie, H., Zhang, S., Zhong, B., Ji, R.: Long-range feature propagating for natural image matting. In: Proceedings of the 29th ACM International Conference on Multimedia. (2021) 526–534
49. Dai, Y., Lu, H., Shen, C.: Learning affinity-aware upsampling for deep image matting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 6841–6850
50. Tang, J., Aksoy, Y., Oztireli, C., Gross, M., Aydin, T.O.: Learning-based sampling for natural image matting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 3055–3063
51. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). (2018) 116–131
52. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning, PMLR (2019) 6105–6114
53. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022)
54. Forte, M., Pitié, F.: f, b, alpha matting. arXiv preprint arXiv:2003.07711 (2020)