MatteFormer: Transformer-Based Image Matting via Prior-Tokens

GyuTae Park[1,2], SungJoon Son[1,2], JaeYoung Yoo[2], SeHo Kim[2], Nojun Kwak[1]
[1] Seoul National University, South Korea
[2] NAVER WEBTOON AI, South Korea
{gyutae.park, sjson718, yoojy31, seho.kim}@webtoonscorp.com, nojunk@snu.ac.kr

Abstract

In this paper, we propose a transformer-based image matting model called MatteFormer, which takes full advantage of trimap information in the transformer block. Our method first introduces a prior-token which is a global representation of each trimap region (e.g. foreground, background and unknown). These prior-tokens are used as global priors and participate in the self-attention mechanism of each block. Each stage of the encoder is composed of PAST (Prior-Attentive Swin Transformer) block, which is based on the Swin Transformer block, but differs in a couple of aspects: 1) It has PA-WSA (Prior-Attentive Window Self-Attention) layer, performing self-attention not only with spatial-tokens but also with prior-tokens. 2) It has prior-memory which saves prior-tokens accumulatively from the previous blocks and transfers them to the next block. We evaluate our MatteFormer on the commonly used image matting datasets: Composition-1k and Distinctions-646. Experiment results show that our proposed method achieves state-of-the-art performance with a large margin. Our codes are available at https://github.com/webtoon/matteformer.

1. Introduction

Image matting is one of the most fundamental tasks in computer vision which is mainly used to separate a foreground object precisely for the purpose of image editing and compositing. Especially, the foreground not only includes complex objects like human hair and animal fur but also includes transparent objects like glass, bulb and water. Natural image can be represented as a linear combination of foreground $F \in \mathbb{R}^{H\times W\times C}$ and background $B \in \mathbb{R}^{H\times W\times C}$ with alpha matte $\alpha \in \mathbb{R}^{H\times W}$ as follows:

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

Nojun kwak was supported by the NRF (2021R1A2C3006659) and IITP grant (NO.2021-0-01343) funded by the Korea government (MSIT).

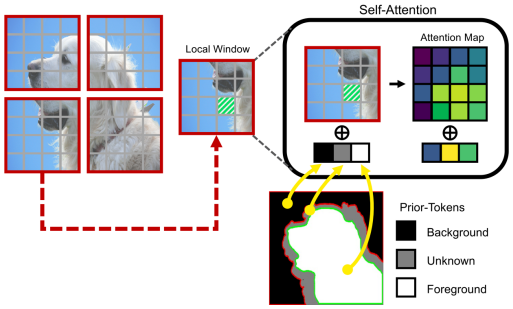


Figure 1. Prior-tokens are generated via the trimap at the bottom and concatenated to the local spatial-tokens to participate in the self-attention mechanism.

where H, W and C denotes the height, the width and the number of channels (3 for a color image) respectively, and $i \in [HW]$ denotes the pixel index.

In image matting, estimating opacity value α given only observed image I, is a highly ill-posed problem if any extra information is not available. Thus, in many works, various types of additional user-inputs (e.g. trimap, scribble, binary mask, background image, etc.) are used, among which, especially trimap is the most common. Trimaps are cost-intensive for users to draw but provide high-quality information about global context such as region information about foreground, background and unknown pixels. Consequently, it is natural to design a model to take a full advantage of this user-input and many works utilizing trimaps have been developed for image matting, most of which are based on convolutional neural networks (CNNs).

While CNNs have been very successful in computer vision tasks, for natural language processing (NLP) tasks, *transformers* earned great success and recently there have been many attempts to introduce transformers in downstream vision tasks as an alternative to CNNs. The seminal work of [11] proposed Vision Transformer (ViT) and showed impressive performances compare to CNN-based models, demonstrating potential on vision tasks. However, implementing the global self-attention in ViT needs high computational cost; it is quadratic to the number of patches.

To overcome this limitation, several general-purpose transformer backbones [10, 30, 52] have been proposed by reducing computational complexity with local self-attention methods. For example, [30] proposed a hierarchical transformer structure by introducing the self-attention within local windows to enable linear cost to the input size and the patch merging layer to reduce the number of spatial-tokens in deeper layers. Besides, they proposed the shifting window scheme to exchange information through neighboring windows. Unfortunately, since the shifting window technique slowly enlarges the receptive field, it is still hard to achieve a sufficiently large receptive field, especially in lower layers.

In this paper, we propose a transformer-based image matting model, named MatteFormer. We first define a prior-token, which represents global context feature of each trimap region; the foreground, background and unknown region as shown in Fig. 1. These prior-tokens are used as global priors and participate in the self-attention mechanism of each block. The encoder stage is composed of the PAST (Prior-Attentive Swin Transformer) blocks, which are based on the Swin Transformer block [30]. However, our PAST block is different from the Swin Transformer block in two aspects. First, it has the PA-WSA (Prior-Attentive Window Self-Attention) layer, where self-attention is computed not only with spatial-tokens but also with prior-tokens as shown in Fig. 1. Second, we introduce the prior-memory, memorizing all prior-tokens generated at each block. Through this, prior-tokens from the previous block can be utilized in the PA-WSA layer of the next block. We evaluate our MatteFormer on Composition-1k and Distinctions-646, which are the commonly used datasets in image matting. The results show that our method achieves state-of-the-art performance. We also conduct some extensive studies about the effectiveness of prior-tokens, visualization of self-attention maps, usage of ASPP, and the computational cost.

In short, our contributions can be summarized as follows:

- We propose MatteFormer, the first transformer-based architecture for the image matting problem.
- We introduce prior-tokens which imply global information of each trimap region (foreground, background and unknown) and use them as global priors in our proposed network.
- We design the PAST (Prior-Attentive Swin Transformer) block, a variant of the Swin Transformer block, which includes the PA-WSA (Prior-Attentive Window Self-Attention) layer and the prior-memory.
- We evaluate MatteFormer on Composition-1k and Distinctions-646, showing that our method achieves state-of-the-art performance with a large margin.

2. Related Works

2.1. Natural Image Matting

Traditional methods. As natural image matting is considered as an ill-posed problem, most of matting algorithms make use of user-input as an additional prior. The *trimap* is frequently used which provides information about foreground, background and unknown regions. Traditional methods predict the alpha matte by primarily utilizing color features. They are mainly divided into sampling-based and propagation-based methods, according to the way of using the color features.

Sampling-based methods [8, 13, 15, 37, 45] make use of color similarities between unknown regions and known (foreground and background) regions to estimate the alpha matte with foreground and background color statistics. On the other hand, propagation-based methods [6, 16, 19–21, 38], also known as affinity-based methods, propagate the alpha values from a known region (foreground and background) to the unknown region to estimate alpha matte via the affinities of neighboring pixels.

Learning-based methods. Since traditional approaches highly depend on color features, these may produce artifacts due to lack of semantic information. Like deep-learning has achieved great success on many other vision tasks, image matting performance also has been significantly improved through CNNs. Meanwhile, because drawing a trimap requires expensive labor cost, approaches without trimap also have been studied.

For the trimap-based methods, [51] proposed a two-stage architecture and released the Composition-1K dataset. [33] utilized a GAN (generative adversarial network) framework to improve performance. [40] mixed the sampling-based method with deep learning method. [18] designed two encoders for local feature and global context, estimating both foreground and alpha matte. [31] proposed an index-guided encoder-decoder structure with the concept of learning to index. [22] developed a guided contextual attention module propagating high-level opacity globally based on low-level affinity. [53] proposed a patch-based method for high-resolution inputs with a module addressing cross-patch dependency and consistency issues between patches. [27] designed the model with the textural compensate path to enhance fine-grained details. [39] used the class of matting patterns and proposed the semantic trimap.

For the trimap-free methods, [34, 57] predicted alpha matte with a single RGB only. [57] proposed structure of two decoders for classifying foreground and background, and fused them in the extra network. [34] designed hierarchical attention structure and proposed Distinctions-646 dataset. [25, 36] proposed the method of using an additional background image instead of the trimap. [54] used binary mask as additional input and proposed a method to

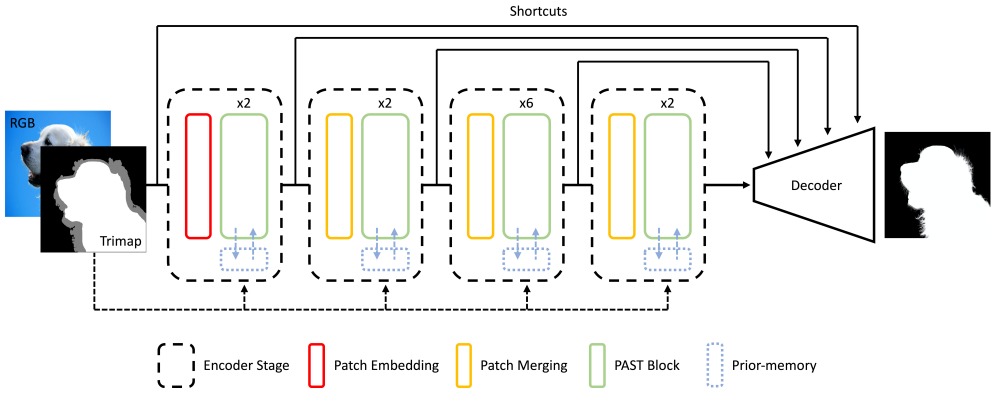


Figure 2. Overall architecture of our proposed MatteFormer, which has a simple encoder-decoder structure with shortcut connections. Each encoder stage includes the proposed PAST (Prior-Attentive Swin Transformer) block. The trimap contributes to generating prior-tokens in each PAST block. The prior-tokens are stored at the prior-memory for usage at later blocks.

progressively refine the uncertain regions through the decoding process.

Since trimap plays a role as a strong hint, the trimap-based methods generally perform better than the trimap-free methods. However, many algorithms have used the trimap by simply concatenating it with input RGB channels, which do not fully utilize the potential of trimap. In this respect, we follow the trimap-based method and propose method of fully making use of the trimap in networks via prior-tokens.

2.2. Transformer on Vision Tasks

Solely based on self-attention mechanism, *transformer* [43] has shown great success on NLP (natural language processing) tasks and has been a base structure for most language models. Besides, [11] applied the transformer to the image classification task demonstrating that transformer-based architectures can achieve competitive performance as an alternative to CNN.

With the potential of ViT [11] and its follow-ups [7, 14, 41, 55], researchers have studied applying transformers to typical vision tasks. For example, numerous transformer-based models have been proposed on image classification [10, 23, 28, 30, 35, 42, 48, 49, 56], object detection [1, 3, 30, 47], semantic segmentation [30, 50, 58], image completion [44] and low-level vision tasks [4, 24, 32].

Meanwhile, many researches [10, 30, 46, 49, 52, 56] have been conducted on designing general-purpose transformer backbone for downstream vision tasks. Especially, [10, 30, 52] focus on variation of self-attention for reducing computational cost and on hierarchical structures for extracting multi-scale features. For example, Swin Transformer [30], which is used as our baseline, performs self-attention in a local window with the shifted window scheme allowing for cross-window connections. However, it has a limitation of still-insufficient receptive field, especially in lower layers. Our approach addresses this issue via prior-tokens and achieves state-of-the-art performance on image matting problem.

3. Methodology

3.1. Network Structure

As shown in Fig. 2, our proposed network called MatteFormer has a typical encoder-decoder structure with shortcut connections. Each encoder stage is composed of a PAST (Prior-Attentive Swin Transformer) block which is a variant version of the Swin Transformer block, and the patch merging layer which reduces the number of tokens. Our proposed PAST block compensates for the insufficient receptive field, originating from the self-attention within local windows, via prior-tokens. In the decoder, we use a quite simple structure as used in [54] with a few convolution and upsample layers. Intermediate features in the encoding layers are delivered to the corresponding decoder layer in a direct way by shortcut connections.

3.2. Prior-token

As a trimap includes definite region information, it is natural to design the network to make full use of this powerful hint, for achieving better results. We first generate prior-tokens of three query regions; foreground, background and unknown areas, by averaging all tokens belonging to the corresponding query region. For example, in Fig. 1, the foreground prior-token is the mean feature of all spatial-tokens lying on the foreground area (white trimap region in the figure). Specifically, the prior-token $\mathbf{p}^q$ can be formulated as

$$\mathbf{p}^q = \frac{1}{N_q} \sum_{i=1}^{N} r_i^q \cdot \mathbf{z}_i, \quad q \in \{\text{fg}, \text{bg}, \text{uk}\}, \tag{2}$$

where q is the query which takes a value among foreground (fg), background (bg) or unknown (uk), i is the to-

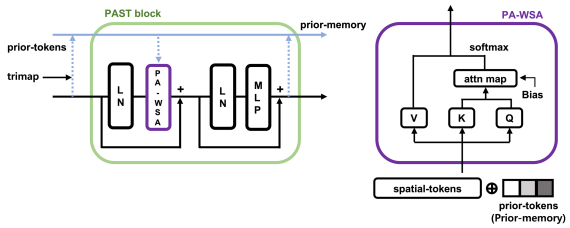


Figure 3. Proposed PAST (Prior-Attentive Swin Transformer) block (left). It is based on the Swin Transformer block but differs in two aspects. One is that not only spatial-tokens in the local window but also prior-tokens are used in the PA-WSA (Prior-Attentive Window Self-Attention) layer (right). The other is that it has a prior-memory.

ken index and $\mathbf{z}_i$ denotes the feature of a spatial-token. r_i^q is a binary value depending on whether the i-th token is on the corresponding query region ($r_i^q = 1$) or not ($r_i^q = 0$).

For example, r_i^{fg} is 1 if the i-th spatial-token is in the foreground region. $N_q = \sum_i^N r_i^q$ is the number of tokens belonging to the region q, indicating how many spatial-tokens are in the corresponding query region. N denotes the total number of spatial-tokens. For the sake of easy computation, if the area corresponding to a spatial-token has more than one query region, i.e, the area is in the boundary, the spatial-token is allocated to the dominant query region. Finally, a prior-token $\mathbf{p}^q$ is assumed to have informative representation as a global prior for the corresponding query region.

3.3. Prior-Attentive Swin Transformer Block

In this work, we use Swin Transformer [30] as our base model, which showed great promise on mainstream vision tasks such as image classification, object detection and semantic segmentation with a hierarchical architecture design and the shifted window scheme. It has a great efficiency with two aspects. One is that local self-attention on non-overlapping windows enables for model to cover large image size with efficient computation. The other is that shifted window partitioning allows for cross-window connection with enhancing long-range dependency. However, it has a limitation that shifted windows enlarge the receptive field slowly, which leads to an insufficient attention region, especially in lower layers.

To address the issue, we propose a Prior-Attentive Swin Transformer (PAST) block as shown in Fig. 3. Our PAST block is based on the Swin Transformer block but different from it in two aspects; First, in the self-attention layer, a token can attend not only to spatial-tokens in the local window but also to global features represented by prior-tokens. Specifically, three prior-tokens are used in our method; foreground, background and unknown prior-tokens. Second, it has a prior-memory, which stores prior-tokens generated from the previous block. Accumulated prior-tokens are used in the next block as informative priors when calculating the self-attention.

Self-attention with prior-tokens. The detailed structure of our PAST block is as follows: First, prior-tokens generated in Sec. 3.2 are concatenated with spatial-tokens in a local window. With these global prior-tokens and local spatial-tokens, self-attention is performed in the Prior-Attentive Window Self-Attention (PA-WSA) layer with matrix multiplications of queries, keys and values in a multi-head manner. Specifically, the self-attention mechanism is as follows

$$Attention(Q, K, V) = SoftMax(QK^T \times s + B)V \quad (3)$$

where $K, V \in \mathbb{R}^{(M^2+N_p)\times d}$, $Q \in \mathbb{R}^{M^2\times d}$ are key, $value$ and $query$ matrices respectively. d is the feature dimension of key, $value$ and $query$. M is the window size, thus M^2 is the number of tokens in a local window. N_p is the number of prior-tokens and s is a scaling factor. B is a relative position bias slightly modified from [30]. Between spatial-tokens in a local window, relative positions lies on range $[-M+1, M-1]$ both along x- and y-axis. In our case, as we use extra prior-tokens, we set an auxiliary matrix $\hat{B} \in \mathbb{R}^{(2M-1)^2+N_p}$. Values of $B \in \mathbb{R}^{M^2\times(M^2+N_p)}$ are taken from $\hat{B}$ and the relative position bias B adjusts the values of attention map according to the relative position. In this way, a query spatial-token can attend to both local spatial-tokens and global prior-tokens.

Prior-memory. Prior-tokens generated from different PAST blocks have different representations for the corresponding query region. That is, the prior-tokens from the previous blocks can deliver informative context to the current block. To implement this, we first define a prior-memory. Like local spatial-tokens, prior-tokens sequentially pass through the PA-WSA, normalization and MLP layer. After that, the prior-tokens are appended to the prior-memory. In the next block, accumulated tokens are used as priors in the PA-WSA layer, i.e, the later block has larger N_p than the former blocks. More specifically, the b-th block in each of the four stages in Fig. 2 has $3 \times b$ prior-tokens in the corresponding prior-memory.

3.4. Training Scheme

The total loss function is defined as a weighted sum of three loss functions; L_{l1} loss, composition loss [51] and Laplacian loss [18], like in [54].

$$L_{total} = L_{l1} + L_{comp} + L_{lap} \quad (4)$$

where L_{l1} is the absolute difference between the ground truth alpha and the predicted alpha. L_{comp} indicates the absolute difference between the ground truth image and the composited image, which is calculated from Eq. (1) with the

ground truth foreground, background and predicted alpha mattes. L_{lap} measures the differences of Laplacian pyramid representations of the alpha maps and captures the local and global difference.

In the decoding process, we use PRM (Progressive Refinement Module) [54] to generate a precise output map as a coarse-to-fine manner. First, the decoder outputs three alpha mattes from different intermediate layers whose output size is 1/8, 1/4 and 1/1 of an input resolution respectively, and then resized to input resolution. Next, through the PRM, outputs are selectively fused and uncertain regions are progressively refined. Specifically, for a current output index l, a refined alpha map $\alpha_l \in \mathbb{R}^{H\times W}$ is calculated with raw matting output $\alpha_l{}' \in \mathbb{R}^{H\times W}$ and self-guidance mask $g_l \in \mathbb{R}^{H\times W}$ as follows:

$$\alpha_l = \alpha_l{}' \odot g_l + \alpha_{l-1} \odot (1 - g_l), \tag{5}$$

$$g_l(x,y) = \begin{cases} 1, & \text{if } 0 < \alpha_{l-1}(x,y) < 1 \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where $\odot$ denotes element-wise multiplication. The self-guidance mask g_l is obtained from the previous alpha map α_{l-1}. It is defined to have 0 if predicted pixel is definite region (foreground or background), and 1 if transparent region. The non-confident pixels in the previous output α_{l-1} are replaced with pixels of the current output $\alpha_l{}'$ according to g_l. Meanwhile, the confident pixels of α_{l-1} are not updated. In this way, confident regions are preserved and the current output can focus only on refining the non-confident region.

Our data augmentation setting is set similar to that of [22] and [54]. We first perform an affine transformation with a random degree, scale and flip. Then we randomly crop both the image and the trimap to a fixed size. After that, random color jittering is applied. Finally, the augmented foreground is composited with a background image.

4. Experiments

In this section, We evaluate our MatteFormer on two public datasets, Composition-1k [51] and Distinctions-646 [34], which are commonly used in the image matting task. We first describe the experimental environments; datasets, evaluation metrics and implementation details. Next, we compare the results of our MatteFormer to those of other state-of-the-art works. Finally, we conduct some ablation studies on our proposed method.

4.1. Datasets and Evaluation

Composition-1k provides unique 50 foreground images with the corresponding ground truth alpha mattes as the test set. Background test images are pre-defined from PASCAL VOC2012 [12]. By compositing the foreground and background images, the number of test samples is 1,000 in total. The train set consists of 431 foreground object images with ground truth alpha mattes. Contrast to the test set, train background images are sampled from MS COCO [26].

Distinctions-646 is comprised of 646 distinct foreground images and it has more versatility and robustness than Composition-1k. Foreground samples are divided into 596 train and 50 test samples. Like Composition-1k, test background samples are pre-defined with PASCAL VOC2012. Foreground images are synthesized with background images following the same composition rule in [51]. Unfortunately, as distinctions-646 does not release official trimaps unlike Composition-1k, a fair comparison with the previous works is hard.

We evaluate our MatteFormer using four major quantitative metrics in image matting: sum of absolute differences (SAD), mean square error (MSE), slope (Grad), and connectivity (Conn). We use official evaluation code provided by [51]. Note that a model with lower metric values can predict more precise alpha mattes.

4.2. Implementation Details

We implement PAST block based on the Swin Transformer block. We introduce prior-tokens to participate in the self-attention mechanism with local spatial-tokens and feed them to other layers in the same way as spatial-tokens. A prior-memory is put on each stage, memorizing prior-tokens from only the PAST blocks in the same stage for simplicity.

Our encoder is first initialized with the Tiny model of Swin Transformer pretrained on ImageNet [9], then trained on the image matting dataset in an end-to-end manner. Since we use a trimap and an RGB image as network input, the number of input channels is 6 which is different from that of the pretrained model. Thus, we bring the weight of the pretrained patch-embedding layer only to the first 3 channels (RGB) of our patch-embedding layer. As the size of our relative position bias table $\hat{B}$, is bigger than that of the pretrained model due to prior-tokens, we bring the pretrained bias table to the front of our bias table weight. The shortcut layers which deliver the encoder features to the decoder layers are composed of 3×3 convolutions with normalization layers. As a decoder, we simply followed [54] which use a simple CNN-based structure with 3×3 convolution layers and upsample layers. Both shortcut and decoder layers are randomly initialized.

When training, we set the network input size to 512x512 and the batch size to 20 in total on 2 GPUs. The learning rate is initialized to $4 \cdot 10^{-4}$. We use Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

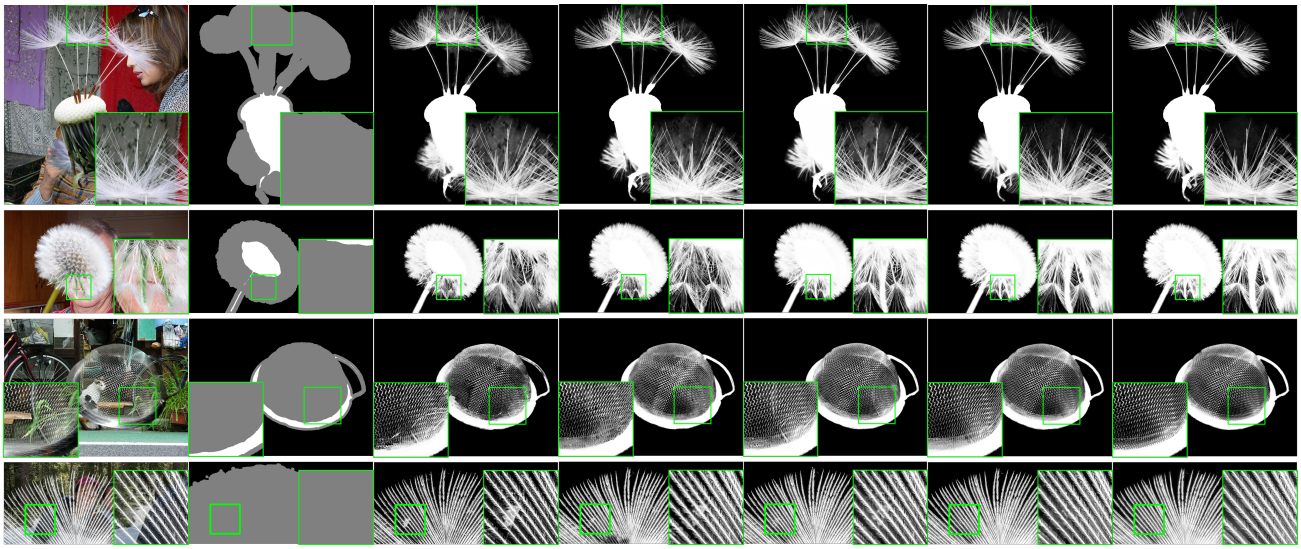


Figure 4. The qualitative comparison results on Composition-1k. Best viewed by zooming in.

Method	SAD	MSE (10^{-3})	Grad	Conn
Learning Based Matting [59]	113.9	48	91.6	122.2
Closed-Form Matting [20]	168.1	91	126.9	167.9
KNN Matting [6]	175.4	103	124.1	176.4
Deep Image Matting [51]	50.4	14	31.0	50.8
AlphaGan [33]	52.4	30	38.0	-
IndexNet [31]	45.8	13	25.9	43.7
HAttMatting [34]	44.0	7.0	29.3	46.4
AdaMatting [2]	41.7	10.0	16.8	-
SampleNet [40]	40.4	9.9	-	-
Fine-Grained Matting [27]	37.6	9.0	18.3	35.4
Context-Aware Matting [18]	35.8	8.2	17.3	33.2
GCA Matting [22]	35.3	9.1	16.9	32.5
HDMatt [53]	33.5	7.3	14.5	29.9
MG Matting [54]	31.5	6.8	13.5	27.3
MG Matting-trimap*	28.9	5.7	11.4	24.9
MG Matting-trimap,res50*	28.4	5.4	11.1	24.3
TIMINet [29]	29.1	6.0	11.5	25.4
SIM [39]	28.0	5.8	10.8	24.8
Ours (MatteFormer)	**23.8**	**4.0**	**8.7**	**18.9**

Table 1. Results on Composition-1k test set. * denotes the baseline of comparison which is our reproduction of MG Matting with trimap input. Original MG Matting is based on ResNet-34 and MG Matting-trimap,res50* uses ResNet-50.

Method	SAD	MSE (10^{-3})	Grad	Conn
Learning Based Matting [59]	105.0	21	94.2	110.4
Closed-Form Matting [20]	105.7	23	91.8	114.6
KNN Matting [6]	116.7	25	103.2	121.5
Deep Image Matting [51]	47.6	9	43.3	55.9
HAttMatting [34]	49.0	9	41.6	49.9
MG Matting-trimap*	23.9	7.4	14.0	22.4
Ours (MatteFormer)	**21.9**	**6.6**	**11.2**	**20.5**

Table 2. Results on Distinctions-646 test set. * denotes the baseline of comparison which is our reproduction of MG Matting with trimap input.

4.3. Results on Image Matting Datasets

Composition-1k. We first compare our MatteFormer with state-of-the-art models on Composition-1k dataset. Tab. 1 tabulates the quantitative results of recent methods and shows that our method outperforms others achieving a new state-of-the-art performance. We set [54] as our strong baseline for comparison because we follow many of its experimental details. However, in original paper, it use a binary mask as an extra input instead of trimap. For fair comparison, we retrain the same model and more larger model (with ResNet-50 [17] backbone) with a trimap and set them as baselines for comparison (marked as * in Tab. 1). Some reasons we do not take [39] and [29] as our baseline are that [39] used classes of matting patterns as additional semantic information and [29] showed slightly less performance than our base model, a reproduction of MG Matting with the trimap input. Fig. 4 shows visual comparison results on Composition-1k among different methods demonstrating the effectiveness of our method.

Distinctions-646. In the case of Distinctions-646, it is hard to fairly compare with previously reported results because it does not offer official trimaps for the test set. We first produce the trimap from the ground truth alpha matte by binarizing the foreground with a threshold and randomly dilating it. We train both the baseline (MG Matting marked as *) described above and our MatteFormer on Distinctions-

Method	SAD	MSE (10^{-3})	Grad	Conn
Baseline (no prior-token)	26.43	5.20	9.57	21.89
Baseline + GAP prior-token	25.30	4.72	9.63	20.61
Baseline + uk prior-token	24.70	4.46	9.10	19.73
+ uk/fg/bg prior-token	24.19	4.05	8.72	19.19
+ prior-memory (MatteFormer)	23.80	4.03	8.68	18.90

Table 3. Ablation study on the usage of prior-tokens and prior-memory. Baseline uses a Swin Transformer Tiny model as its encoder without prior-token. Results are on the Composition-1k.

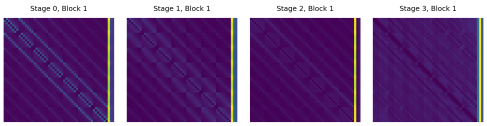
646, and then evaluate them with the same testing environment. The results are shown in the last two rows of Tab. 2. The first five methods in the table are from [34] as references, which are trained on Distinctions-646 train set. The results show that MatteFormer has superior performance compared to the baseline model.

4.4. Ablation Study

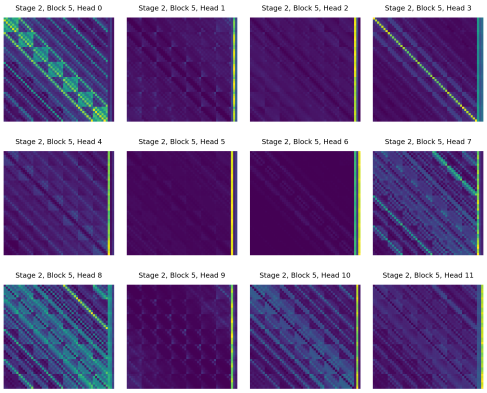
Prior-tokens and Prior-memory. In MatteFormer, the building block of each encoder stage is the PAST block which is a variant version of Swin Transformer block, as described in Sec. 3.3. To show how the PAST block contributes to improving performance, we conduct an ablation study. We start from a baseline model in which we set the encoder as a pure Swin Transformer Tiny model without prior-token. The decoder and shortcuts of the baseline are the same as MatteFormer.

In the PA-WSA (Prior-Attentive Window Self-Attention) layer, a token in the local window can attend not only to the in-window spatial-tokens but also to the global prior-tokens. As a global prior, we first use an averaged token of all spatial-tokens (Global Average Pooling (GAP)-token). In this setting, we do not use any trimap information in self-attention layer. Next, we use unknown prior-token as a global prior. Further, we use all 3 prior-tokens; foreground, background and unknown prior-token. Finally, our proposed MatteFormer model uses all prior-tokens and introduces prior-memory which make it possible to access all prior-tokens generated from the previous blocks.

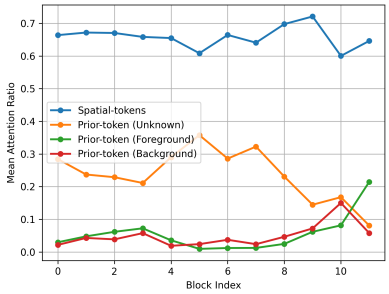
The results are shown in Tab. 3. The baseline which uses no prior-token performs worst in all metrics as expected. When using unknown prior-token, the metric values were lower than when using GAP-token as a global prior. This implies that the prior-token generated via the trimap delivers more useful information than the simple GAP-token. In the case of using 3 prior-tokens (foreground, background, unknown), it shows better performance than using 1 prior-token (unknown prior-token) only. This suggests that a token in a local window can refer all three prior-tokens properly in the self-attention layer. Introducing prior-memory



(a) Mean attention maps are averaged on multi-heads. We plot only the second block (block index 1) of each stage to show in a simple.



(b) Multi-head attention maps in the PA-WSA layer. We plot only the last block (block index 5) of stage index 2 to show examples in a simple. The self-attention layer has 12 multi-heads.



(c) Mean attention ratios on spatial-tokens and prior-tokens across all blocks.

Figure 5. Ablation study of attention map in PA-WSA layer. Mean attention maps and mean attention ratios show how much attention is on both spatial-tokens and prior-tokens.

also shows a improvement of performance. Prior-tokens from the previous blocks contribute to making better representation in the PA-WSA layer of the current block.

Visualization of attention maps in PA-WSA layer. In this subsection, we demonstrate that the PA-WSA layer indeed has a prior-attentive property. In Fig. 5, we visualize the mean attention maps, multi-head attention maps and the mean attention ratios between local spatial-tokens and global prior-tokens. We study both of them on 50 test images with different foregrounds. We averaged the attention maps over all windows and samples. For simplicity, we conduct the ablation study on the model without prior-memory.

In Fig. 5a, we first visualize the mean attention maps of

Methods	w ASPP		w/o ASPP	
	SAD	MSE (10^{-3})	SAD	MSE (10^{-3})
Baseline (no prior-token)	26.97	5.35	26.43	5.20
+ u.k prior-token	25.60	4.68	24.71	4.46
+ u.k/f.g/b.g prior-token	25.52	4.38	24.19	4.05
+ prior-memory (MatteFormer)	25.15	4.30	23.80	4.03

Table 4. Ablation study on the usage of ASPP (Atrous Spatial Pyramid Pooling) on MatteFormer.

PA-WSA layers. Since local window size is set to 7, y-axis represents 49 query spatial-tokens in a local window. Meanwhile, x-axis represents 49 spatial-tokens and 3 prior-tokens (unknown, foreground and background token in order). We can observe that the prior-token regions (last three columns) in attention maps are activated. It means that one token can attend to the prior-tokens in a self-attention layer to refer to global priors as we intended.

Fig. 5b shows the attention maps of each head. We can see that each head has a different pattern of attention, especially in terms of prior-token usage. For example, head 4 and 5 mainly use the unknown prior-token and head 6 focuses on both unknown and background prior-tokens. Meanwhile, head 0 attends more on the spatial-tokens rather than the prior-tokens. Head 3 uses all three prior-tokens together with the spatial-tokens.

In Fig. 5c, we visualize mean attention ratios on local spatial-tokens and global prior-tokens over all blocks, quantitatively showing that how much attention is on each prior-token. There are four lines in Fig. 5c, one is the summation of attention ratios over local spatial-tokens, and the three others are the ratios of prior-tokens. We observe that prior-tokens are used in the PA-WSA layers across all blocks as we have expected. Tokens tend to attend more on the unknown prior-token than the known prior-tokens (foreground and background), which means that the unknown region is more informative to predict the alpha matte.

ASPP (Atrous Spatial Pyramid Pooling). Many state-of-the-art image matting models [25, 29, 34, 39, 54] and many semantic segmentation methods use ASPP (Atrous Spatial Pyramid Pooling) [5] between encoder and decoder to grow receptive fields for global representation. ASPP employs multiple atrous convolution filters with various atrous rates to capture spatially far-away context features.

However, we discover that ASPP is rather hinder the performance in our model. We evaluate MatteFormer on the Composition-1k both with ASPP and without ASPP in Tab. 4. All models with no ASPP show better performances than those with ASPP. Since our encoder is based on the transformer, MatteFormer already has a larger global receptive field than CNN-based models. As forcibly increasing receptive field with pre-defined atrous convolution, ASPP is redundant in our model. So, we do not use ASPP in our model unlike other recent methods.

Method	Params	FLOPs	SAD	MSE (10^{-3})
MG Matting-trimap*	29.7M	45.7G	28.89	5.73
MG Matting-trimap,res50*	52.7M	58.9G	28.35	5.42
Baseline (no prior-token)	44.8M	55.9G	26.43	5.20
Ours (MatteFormer)	44.8M	57.2G	23.80	4.03

Table 5. Parameters and FLOPs.

Parameters and FLOPs. In Tab. 5, we compare the number of parameters and FLOPs of our MatteFormer to those of baseline models, for demonstrating the effectiveness of our method. Input image size is assumed to be 512 for calculating FLOPs. SAD and MSE are calculated on the Composition-1k test set for comparison. Note that the baseline (no prior-token) uses a pure Swin Transformer Tiny model as its backbone with shortcut layers and uses no prior-token. Also, it does not use the ASPP module unlike MG Matting-trimap* and MG Matting-trimap,res50*.

First, our baseline model has fewer parameters/FLOPs than MG Matting-trimap,res50* model, but shows better performance. It means that transformer-based architecture works effectively on the image matting problem. Next, comparing the baseline and our MatteFormer, we can observe a quite large gap on evaluation scores with a little increase of parameters and FLOPs. Consequently, Tab. 5 suggests that our proposed method, using prior-tokens and prior-memory, effectively works on the our transformer-based architecture.

5. Conclusion

In this work, we propose MatteFormer, a simple yet effective model with a modified Swin Transformer block called PAST (Prior-Attentive Swin Transformer) block for the image matting problem. We introduce prior-tokens which represent the context of global regions separated by the given trimap. In our PAST block, prior-tokens are used as global priors with local spatial-tokens, participating at the self-attention mechanism in the PA-WSA (Prior-Attentive Window Self-Attention) layer. So, one token can attend to both local spatial-tokens and global prior-tokens. We evaluate MatteFormer on the common datasets of the image matting problem. The experimental results show that our method achieves state-of-the-art performance. We hope that our MatteFormer can serve as a strong baseline for future image matting models based on a transformer architecture. One limitation of our work is that it mainly focuses on the encoder structure and the trimap-based approach. In the future work, we would like to design a fully transformer-based model with prior-tokens and also extend our model to trimap-free methods.

References

[1] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020. 3

[2] Shaofan Cai, Xiaoshuai Zhang, Haoqiang Fan, Haibin Huang, Jiangyu Liu, Jiaming Liu, Jiaying Liu, Jue Wang, and Jian Sun. Disentangled image matting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8819–8828, 2019. 6

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3

[4] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021. 3

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 8

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

[7] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Do we really need explicit position encodings for vision transformers? *arXiv e-prints*, pages arXiv–2102, 2021. 3

[8] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. 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, pages II–II. IEEE, 2001. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652*, 2021. 2, 3

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5

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

[14] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021. 3

[15] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In *CVPR 2011*, pages 2049–2056. IEEE, 2011. 2

[16] Kaiming He, Jian Sun, and Xiaoou Tang. Fast matting using large kernel matting laplacian matrices. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2165–2172. IEEE, 2010. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[18] Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4130–4139, 2019. 2, 4, 6

[19] Philip Lee and Ying Wu. Nonlocal matting. In *CVPR 2011*, pages 2193–2200. IEEE, 2011. 2

[20] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2007. 2, 6

[21] Anat Levin, Alex Rav-Acha, and Dani Lischinski. Spectral matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1699–1712, 2008. 2

[22] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11450–11457, 2020. 2, 5, 6

[23] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 3

[24] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 3

[25] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8762–8771, 2021. 2, 8

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[27] Chang Liu, Henghui Ding, and Xudong Jiang. Towards enhancing fine-grained details for image matting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 385–393, 2021. 2, 6

[28] Yun Liu, Guolei Sun, Yu Qiu, Le Zhang, Ajad Chhatkuli, and Luc Van Gool. Transformer in convolutional neural networks. *arXiv preprint arXiv:2106.03180*, 2021. 3

[29] Yuhao Liu, Jiake Xie, Xiao Shi, Yu Qiao, Yujie Huang, Yong Tang, and Xin Yang. Tripartite information mining and integration for image matting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7555–7564, 2021. 6, 8

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 2, 3, 4

[31] Hao Lu, Yutong Dai, Chunhua Shen, and Songcen Xu. Indices matter: Learning to index for deep image matting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3266–3275, 2019. 2, 6

[32] Zhisheng Lu, Hong Liu, Juncheng Li, and Linlin Zhang. Efficient transformer for single image super-resolution. *arXiv preprint arXiv:2108.11084*, 2021. 3

[33] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. Alphagan: Generative adversarial networks for natural image matting. *arXiv preprint arXiv:1807.10088*, 2018. 2, 6

[34] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13676–13685, 2020. 2, 5, 6, 7, 8

[35] Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*, 2021. 3

[36] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2291–2300, 2020. 2

[37] Ehsan Shahrian, Deepu Rajan, Brian Price, and Scott Cohen. Improving image matting using comprehensive sampling sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 636–643, 2013. 2

[38] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. In *ACM SIGGRAPH 2004 Papers*, pages 315–321. 2004. 2

[39] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Semantic image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11120–11129, 2021. 2, 6, 8

[40] Jingwei Tang, Yagiz Aksoy, Cengiz Oztireli, Markus Gross, and Tunc Ozan Aydin. Learning-based sampling for natural image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3055–3063, 2019. 2, 6

[41] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3

[42] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 3

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[44] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. *arXiv preprint arXiv:2103.14031*, 2021. 3

[45] Jue Wang and Michael F Cohen. Optimized color sampling for robust matting. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[46] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 3

[47] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021. 3

[48] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020. 3

[49] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 3

[50] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021. 3

[51] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2970–2979, 2017. 2, 4, 5, 6

[52] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021. 2, 3

[53] Haichao Yu, Ning Xu, Zilong Huang, Yuqian Zhou, and Humphrey Shi. High-resolution deep image matting. *arXiv preprint arXiv:2009.06613*, 2020. 2, 6

[54] Qihang Yu, Jianming Zhang, He Zhang, Yilin Wang, Zhe Lin, Ning Xu, Yutong Bai, and Alan Yuille. Mask guided matting via progressive refinement network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1154–1163, 2021. 2, 3, 4, 5, 6, 8

[55] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 3

[56] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. *arXiv preprint arXiv:2103.15358*, 2021. 3

[57] Yunke Zhang, Lixue Gong, Lubin Fan, Peiran Ren, Qixing Huang, Hujun Bao, and Weiwei Xu. A late fusion cnn for digital matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7469–7478, 2019. 2

[58] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021. 3

[59] Yuanjie Zheng and Chandra Kambhamettu. Learning based digital matting. In *2009 IEEE 12th international conference on computer vision*, pages 889–896. IEEE, 2009. 6