Referring Image Matting

Jizhizi Li, Jing Zhang, and Dacheng Tao
The University of Sydney, Sydney, Australia
jili8515@uni.sydney.edu.au, jing.zhang1@sydney.edu.au, dacheng.tao@gmail.com *

Abstract

Different from conventional image matting, which either requires user-defined scribbles/trimap to extract a specific foreground object or directly extracts all the foreground objects in the image indiscriminately, we introduce a new task named Referring Image Matting (RIM) in this paper, which aims to extract the meticulous alpha matte of the specific object that best matches the given natural language description, thus enabling a more natural and simpler instruction for image matting. First, we establish a large-scale challenging dataset RefMatte by designing a comprehensive image composition and expression generation engine to automatically produce high-quality images along with diverse text attributes based on public datasets. RefMatte consists of 230 object categories, 47,500 images, 118,749 expression-region entities, and 474,996 expressions. Additionally, we construct a real-world test set with 100 high-resolution natural images and manually annotate complex phrases to evaluate the out-of-domain generalization abilities of RIM methods. Furthermore, we present a novel baseline method CLIPMat for RIM, including a context-embedded prompt, a text-driven semantic pop-up, and a multi-level details extractor. Extensive experiments on RefMatte in both keyword and expression settings validate the superiority of CLIPMat over representative methods. We hope this work could provide novel insights into image matting and encourage more follow-up studies. The dataset, code and models are available at https://github.com/JizhiziLi/RIM.

1. Introduction

Image matting refers to extracting the soft alpha matte of the foreground in natural images, which is beneficial for various downstream applications such as video conferences, advertisement production, and E-commerce promotion [58]. Typical matting methods can be divided into two groups: 1) the methods based on auxiliary inputs, e.g., scribble [17] and trimap [1,17], and 2) automatic matting methods that can extract the foreground without any human intervention [19,44]. However, the former are not applicable for fully automatic scenarios, while the latter are limited to specific categories, e.g., human [2,32,57], animal [19], or the salient objects [40,60]. It is still unexplored to carry out controllable image matting on arbitrary objects based on language instructions, e.g., extracting the alpha matte of the specific object that best matches the given language description.

Recently, language-driven tasks such as referring expression segmentation (RES) [35], referring image segmentation (RIS) [12,25,54], visual question answering (VQA) [8], and referring expression comprehension (REC) [31] have been widely studied. Great progress in these areas has been made based on many datasets like ReferIt [14], Google RefExp [34], RefCOCO [56], VGPhraseCut [50], and Cops-Ref [3]. However, due to the limited resolution of available datasets, visual grounding methods are restricted to the coarse segmentation level. Besides, most of the methods [13,30] neglect pixel-level text-visual alignment and cannot preserve sufficient details, making them difficult to be used in scenarios that require meticulous alpha mattes.

To fill this gap, we propose a new task named Referring Image Matting (RIM), which refers to extracting the meticulous high-quality alpha matte of the specific foreground object that can best match the given natural language description from the image. Different from the conventional matting methods, RIM is designed for controllable image matting that can perform a more natural and simpler instruction to extract arbitrary objects. It is of practical significance in industrial application domains and opens up a new research direction. To facilitate the study of RIM, we establish the first dataset RefMatte, which consists of 230 object categories, 47,500 images, and 118,749 expression-region entities together with the corresponding high-quality alpha mattes and 474,996 expressions. Specifically, to build up RefMatte, we revisit a lot of prevalent public matting datasets like AM-2k [19], P3M-10k [18], AIM-500 [20], SIM [45] and manually label the category of each foreground object (a.k.a. entity) carefully. We also adopt multiple off-the-shelf deep learning models [27,51] to generate various attributes for each entity, e.g., gender, age, and clothes type of human.

*Dr Jing Zhang and Ms Jizhizi Li were supported by Australian Research Council Projects in part by FL170100117 and IH180100002.
Then, we design a comprehensive composition and expression generation engine to produce the synthetic images with reasonable absolute and relative positions considering other entities. Finally, we present several expression logic forms to generate varying language descriptions with the use of rich visual attributes. In addition, we propose a real-world test set RefMatte-RW100 with 100 images containing diverse objects and human-annotated expressions, which is used to evaluate the generalization ability of RIM methods. Some examples are shown in Figure 1 and Figure 2.

Since previous visual grounding methods are designed for the segmentation-level tasks, directly applying them [13, 30, 43] to the RIM task cannot produce promising alpha mattes with fine details. Here, we present CLIPMat, a novel baseline method specifically designed for RIM. CLIPMat utilizes the large-scale pre-trained CLIP [41] model as the text and visual backbones, and the typical matting branches [18, 19] as the decoders. An intuitive context-embedded prompt is adopted to provide matting-related learnable features for the text encoder. To extract high-level visual semantic information for the semantic branch, we pop up the visual semantic feature through the guidance of the text output feature. Additionally, as RIM requires much more visual details compared to the segmentation task, we devise a module to extract multi-level details by exploiting shallow-layer features and the original input image, aiming to preserve the foreground details in the matting branch. Figure 1 and Figure 2 show some promising results of the proposed CLIPMat given different types of language inputs, i.e., keywords and expressions.

Furthermore, to provide a fair and comprehensive evaluation of CLIPMat and relevant state-of-the-art methods, we conduct extensive experiments on RefMatte under two different settings, i.e., the keyword-based setting and expression-based setting, depending on language descriptions’ forms. Both the subjective and objective results have validated the superiority of CLIPMat over representative methods. The main contribution of this study is three-fold. 1) We define a new task named RIM, aiming to identify and extract the alpha matte of the specific foreground object that best matches the given natural language description. 2) We establish the first large-scale dataset RefMatte, consisting of 47,500 images and 118,749 expression-region entities with high-quality alpha mattes and diverse expressions. 3) We present a novel baseline method CLIPMat specifically designed for RIM, which achieves promising results in two different settings of RefMatte, also on real-world images.

2. Related Work

Image matting Image matting is a fundamental computer vision task and essential for various potential downstream applications [4, 6, 29]. Previous matting methods are divided into two groups depending on whether or not they use auxiliary user inputs. In the first group, the methods use a three-class trimap [22, 53], sparse scribbles [17], a background image [24], a coarse map [57], or user click [49] as the auxiliary input to guide alpha estimating. Among them, scribble and click-based methods are more controllable since they usually indicate one specific foreground. However, the flexibility of these methods is still limited since the predictions are usually performed with low-level color propagation
and are very sensitive to the scribbles’ density [18, 23]. In the second group, the methods [2, 15, 18–20, 40, 60] automatically extract the foreground objects without any manual efforts. Recently, there is also some work making efforts to control the matting process by determining which objects can be extracted. For example, Xu et al. [52] propose to extract the foreground human and all related objects automatically for human-object interaction. Sun et al. propose to extract each human instance separately rather than extracting all of them indiscriminately [46]. However, it is still unexplored for controllable image matting, especially by using natural language description as guidance to extract specific foreground object that best matches the input text, even though it is efficient and flexible for the matting model to interact with a human.

In this paper, we fill this gap by proposing the RIM task, the RefMatte dataset, and the baseline method CLIPMat.

**Matting datasets** Many matting datasets have been proposed to advance the progress in the image matting area. Typical matting datasets contain high-resolution images belonging to some specific object categories that have lots of details like hair, accessories, fur, and net, as well as transparent objects. For example, the matting datasets proposed by Xu et al. [53], Qiao et al. [40], Sun et al. [45], and Li et al. [20], contain many different categories of objects, including human, animals, cars, plastic bags, and plants. Besides, some other matting datasets focus on a specific category of object, e.g., humans in P3M-10K [18] and animals in AM-2K [19]. In addition to the foreground objects, background images are also helpful for generating abundant composite images. For example, Li et al. [19] propose a large-scale background dataset containing 20k high-resolution and diverse images, which are helpful to reduce the domain gap between composites and natural ones. All the above datasets have open licenses and can serve as valuable resources to construct customized matting datasets, e.g., the proposed RefMatte.

Besides, it is noteworthy that due to the laborious and costly labeling process of matting datasets, existing public matting datasets [40, 45, 60] usually provide only the extracted foregrounds through chroma keying [53] without the original backgrounds. To compose a reasonable amount of trainable data, a typical solution in previous matting methods [15, 26, 57] is to generate synthetic images like in other tasks [7, 33] by pasting the foregrounds with numerous background images. As for the domain gap between the real-world images and the composite ones, some works [15, 19] have already reduced it to an acceptable range through some augmentation strategies. Although some work also present real-world matting datasets, they all contain only one foreground from a specific type, e.g., person [18], animal [19], or objects [40], making them unsuitable to serve as the benchmark for RIM. In our work, we follow the composition route in generating RefMatte and ensure its large scale, diversity, difficulty, and high quality by synthesizing a large number of images, where there are multiple foreground objects with similar semantics and fine details on diverse backgrounds. Furthermore, we present a real-world test set with flowery human annotated expression labels to validate models’ out-of-domain generalization abilities.

**Vision-language tasks and methods** Vision-Language tasks, such as RIS [12], RES [55], REC [31], text-driven manipulation [37, 59], and text-to-image generation [38, 39, 42], have been widely studied, which are helpful for many applications like interactive image editing. Among them, RIS aims to segment the target object given language expression, which is most related but totally different from our work. The relevant methods can be divided into single-stage [21, 25, 30, 43, 48] and two-stage ones [11, 13, 28, 55]. The former directly trains a segmentation network on top of the pre-trained models like CLIP [41], and the latter perform sequential region proposal and segmentation. However, due to the task setting (i.e., for segmentation rather than matting) and the lack of high-quality annotations (e.g., alpha mattes) [14, 34, 50, 56], most of them have neglected the pixel-level text-semantic alignment and cannot produce fine-grained mask. Thus, we propose the new task RIM with the dataset RefMatte to facilitate the research of natural language guided image matting. Moreover, the proposed method CLIPMat with specifically designed modules could produce high-quality alpha matte and thus serve as the baseline for RIM.

### 3. The RefMatte Dataset

In this section, we present the overview pipeline of constructing RefMatte (Sec. 3.1 and Sec. 3.2), the task settings, and a real-world test set (Sec. 3.3). Figure 3 shows some examples from RefMatte.

#### 3.1. Preparation of Matting Entities

To prepare high-quality matting entities for constructing RefMatte, we revisit available matting datasets to select the required foregrounds. We then manually label each entity’s category and annotate the attributes by leveraging off-the-shelf deep learning models [27, 51]. We present key details as follows, while more in the supplementary materials.

**Pre-processing and filtering** Due to the nature of the image matting task, all the candidate entities should be in high resolution, with clear and fine details in the alpha matte. Moreover, the data should be publicly available with open licenses and without privacy concerns. With regard to these requirements, we adopt all the foreground images from AM-2K [19], P3M-10k [18], and AIM-500 [20]. For other available datasets like SIM [45], DIM [53], and HATT [40], we filter out those foreground images with identifiable faces in human instances and those in low-resolution or having low-quality alpha mattes. The final number of foreground entities is 13,187 in total, and we use images from BG-20k [19] as the background images for composition.
Figure 3. Some examples from our RefMatte dataset. The first row shows the composite images with different foreground instances while the second row shows the natural language descriptions corresponding to the specific foreground instances indicated by the green dots.

Annotate the category names of entities Previous matting datasets do not provide the specific (category) name for each entity since those matting methods extract all the objects indiscriminately. However, we need the entity name in the RIM task to describe the foreground. Following [36], we label the entry-level category name for each entity, which stands for the most commonly used name by people. Here, we adopt a semi-automatic strategy. Specifically, we use the pre-trained Mask RCNN detector [9] with a ResNet-50-FPN [10] backbone from [51] to automatically detect and label the category names for each foreground instance and then manually check and correct them. In total, we have 230 categories in RefMatte. Furthermore, we adopt WordNet [35] to generate synonyms for each category name to enhance the diversity. We manually check the synonyms and replace some of them with more reasonable ones.

Annotate the attributes of entities To ensure all the entities have rich visual properties to support forming abundant expressions, we annotate them with several attributes, e.g., color for all entities, gender, age, and clothes type for the human entities. A semi-automatic strategy is adopted in retrieving such attributes. For attribute color, we cluster all the pixel values of the foreground image, find the most frequent value, and match it with the specific color in webcolors. For gender and age, we adopt the pre-trained models provided by Levi et al. in [16] and follow common sense to define the age group based on the predicted ages. For clothes type, we adopt the off-the-shelf model provided by Liu et al. in [27]. Furthermore, motivated by the categorization of matting foregrounds in [20], we add the attributes of whether or not salient or transparent for all the entities as they also matter in image matting. In summary, we have at least three attributes for each entity and six attributes for human entities.

3.2. Image Composition and Expression Generation

Based on these collected entities, we propose an image composition engine and an expression generation engine to construct RefMatte. In order to present reasonably looking composite images with semantically clear, grammatically correct, as well as abundant and fancy expressions, how to arrange the candidate entities and build up the language descriptions is the key to constructing RefMatte, which is also challenging. To this end, we define six types of position relationships for arranging entities in a composite image and leverage diverse logic forms to produce appropriate expressions. We present the details as follows.

Image composition engine We adopt two or three entities for each composite to keep the entities at high resolution while arranging them with a reasonable position relationship. We define six kinds of position relationships: left, right, top, bottom, in front of, and behind. For each relationship, we generate the foregrounds by [17] and composite them with the backgrounds from BG-20k [19] via alpha blending. Specifically, for the relationships left, right, top, and bottom, we ensure there are no occlusions in the instances to preserve their details. For the relationships in front of and behind, we simulate occlusions between the foreground instances by adjusting their relative positions. We prepare a bag of candidate words to denote each relationship and present in the supplementary materials. Some examples are in Figure 3.

Expression generation engine To provide abundant expressions for the entities in the composite images, we define...
three types of expressions for each entity regarding different logic forms, where \(<att_i>\) is the attribute, \(<obj_j>\) is the category name, and \(<rel_i>\) is the relationship between the reference entity and the related one \(<obj_j>\):

1. **Basic expression** This is the expression that describes the target entity with as many attributes as one can, e.g., the/a \(<att_i>\) \(<att_i>\)...\(<obj_j>\) or the/a \(<obj_j>\) which/that is \(<att_i>\) \(<att_i>\), and \(<att_j>\). For example, as shown in Figure 3(a), the basic expression for the entity flower is ‘the lightpink and salient flower’;

2. **Absolute position expression** This is the expression that describes the target entity with many attributes and its absolute position in the image, e.g., the/a \(<att_i>\) \(<obj_j>\)...\(<obj_j>\) \(<rel_i>\) the photo/image/picture or the/a \(<obj_j>\) \(<rel_i>\) which/that is \(<att_i>\) \(<att_i>\) \(<rel_i>\) the photo/image/picture. For example, as shown in Figure 3(a), the absolute position expression for the flower is ‘the plant which is lightpink and salient at the rightmost edge of the picture’;

3. **Relative position expression** This is the expression that describes the target entity with many attributes and its relative position with another entity, e.g., the/a \(<att_j>\) \(<att_j>\)...\(<obj_j>\) \(<rel_i>\) the/a \(<att_i>\) \(<att_i>\)...\(<obj_j>\) or the/a \(<obj_j>\) which/that is \(<att_i>\) \(<att_i>\) \(<rel_i>\) the/a \(<obj_j>\) \(<obj_j>\) which/that is \(<att_i>\) \(<att_i>\). For example, as shown in Figure 3(a), the relative position expression for the flower is ‘the flower which is lightpink at the right side of the cat which is dimgray and non-transparent’;

### 3.3. Dataset Split and Task Settings

In total, We have 13,187 matting entities. We split out 11,799 for constructing the training set and 1,388 for the test set. For the training/test split, we reserve the original split in the source matting datasets except for moving all the long-tailed categories to the training set. However, the categories are not balanced since most of the entities belong to the human or animal categories. The proportion of humans, animals, and objects is 9186:1800:813 in the training set and 977:200:211 in the test set. To balance the categories, we duplicate some entities to modify the proportion to 5:1:1, leading to 10550:2110:2110 in the training set and 1055:211:211 in the test set. We then pick 5 humans, 1 animal, and 1 object as one group and feed them into the composition engine to generate an image in RefMatte. For each group in the train split, we composite 20 images with various backgrounds. For the one in the test split, we composite 10 images. The ratio of relationships left/right/top/bottom in front of/behind is set to 7:2:1. The number of entities in each image is set to 2 or 3 but fixed to 2 for relationships front of/behind to preserve each entities’ high resolution. Finally, we have 42,200 training and 2,110 test images. To further enhance the diversity of the composite images, we randomly choose entities and relationships from all candidates to form another 2,800 training images and 390 test images. Finally, we have 45,000 training images and 2,500 test images.

**Task settings** To benchmark RIM methods given different forms of language descriptions, we set up two settings upon RefMatte. We present their details as follows:

1. **keyword-based setting** The text description in this setting is the keyword, which is the entry-level category name of the entity, e.g., flower, human, and alpaca in Figure 3. Please note that we filter out images with ambiguous semantic entities for this setting;

2. **Expression-based setting** The text description in this setting is the generated expression chosen from the basic expressions, absolute position expressions, and relative position expressions, as seen in Figure 3.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RefMatte</td>
<td>train</td>
<td>30,391</td>
<td>77,849</td>
<td>77,849</td>
<td>230</td>
<td>1.06</td>
</tr>
<tr>
<td>Keyword</td>
<td>test</td>
<td>1,602</td>
<td>4,085</td>
<td>4,085</td>
<td>66</td>
<td>1.04</td>
</tr>
<tr>
<td>Expression</td>
<td>train</td>
<td>45,000</td>
<td>112,506</td>
<td>449,624</td>
<td>230</td>
<td>16.86</td>
</tr>
<tr>
<td>RefMatte-RW100</td>
<td>test</td>
<td>22452</td>
<td>42,200</td>
<td>2,110</td>
<td>12.01</td>
<td>1,600</td>
</tr>
</tbody>
</table>

### 4.1. Overview

Motivated by the success of large-scale pre-trained vision-language models like CLIP [41] on downstream tasks, we also adopt the text encoder and image encoder from CLIP.

1 More details of RefMatte, including the distribution of matting entities, linguistic details, and statistics are in the supplementary materials.

---

**Real-world test set** Since RefMatte is built upon composite images, a domain gap may exist when applying the models to real-world images. To further investigate the out-of-domain generalization ability of RIM models, we establish a real-world test set **RefMatte-RW100**, which consists of 100 high-resolution natural images with 2 to 3 entities in each image. The expressions are annotated by specialists following the same rules in Sec. 3.2, but in freestyles. The high-quality alpha mattes are generated by specialists via image editing software, e.g., Adobe Photoshop and GIMP. We show some examples in Figure 2. Furthermore, we show some statistics of RefMatte and RefMatte-RW100 in Table 1, including the number of images, alpha mattes, text descriptions, categories, and the average length of texts.
as our backbone. We choose ViT-B/16 and ViT-L/14 [5] as the image encoder backbone (to demonstrate that the scalability of model size also matters in image matting for the first time). As for the decoder, different from RIS methods [30, 43] that predict a coarse segmentation mask through a single decoder, RIM is a task that requires both global semantic and local details information [19]. Thus, we utilize the dual-decoder framework from state-of-the-art matting methods [18, 20] to predict a trimap and the alpha matte in the transition area, respectively. We name them the matting semantic decoder and matting details decoder in CLIPMat. 

The input of our method is an image with a text description, which can be either a keyword (e.g., people) or an expression (the handsome man that is smiling and playing with his dog), as shown in Figure 4. The output is the meticulous alpha matte of the target object.

### 4.2. CP: Context-embedded Prompt

Although some previous works have already adopted prompt engineering [41, 61] to enhance the understanding ability of the text input, how to adapt them in RIM is unexplored. In our work, we design two kinds of contexts to be embedded in the original prompt, named pre-embedding context and post-embedding context, as shown in Figure 4. Both of them have been proven effective in the experiments. We present the details as follows.

**Pre-embedding context** For the keyword setting, to reduce the gap between a single word and the CLIP model pretrained on long sentences, we create a bag of matting-related customized prefix context templates, including “the foreground of {keyword}”, “the mask of {keyword}”, “to extract the {keyword}” and so on. We add the pre-embedding context to the keyword directly before tokenization, ensuring that the text encoder can understand the image matting task by adapting the encoded knowledge during pre-training.

**Post-embedding context** To improve the ability of the text encoder to understand the text, we follow the work [61] to add some learnable context appended to the tokenized text in both keyword and expression settings. Since the length of text space and context is different in the two settings, we use 14 and 69 for text length in keyword and expression settings, respectively, while the length of learnable context is fixed to 8 for both settings.

### 4.3. TSP: Text-driven Semantic Pop-up

To ensure the text feature from the text encoder can provide better guidance on dense-level visual semantic perception, we propose a module named TSP (text-driven semantic pop-up) to process the text and visual features before the matting semantic decoder. Specifically, we abandon the last project layer in both the image encoder and text encoder to keep the original dimension. Thus, the input of TSP is the visual feature $x_v \in \mathbb{R}^{(N+1) \times D_v}$ and text feature $x_t \in \mathbb{R}^{L \times D_t}$, where $N = HW/P^2$ stands for the resulting number of patches after ViT transformer [5]. On the other hand, $L$ stands for the total length of the text and embedding context, in our cases, which is 22 for the keyword-based setting and 79 for the expression-based setting. We first normalize them through layer norm, linear projection, and another layer norm to achieve the same dimension $D$. We then pop up the semantic information from the visual feature under the guidance of the text feature via cross-attention [47]. In addition, we adopt self-attention to further refine the visual feature with a residual connection. Finally, we pass through the fea-
Table 2. Results on the RefMatte test set in two settings and the RefMatte-RW100 test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Refiner</th>
<th>Keyword-based setting</th>
<th>Expression-based setting</th>
<th>RefMatte-RW100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SAD</td>
<td>MSE</td>
<td>MAD</td>
</tr>
<tr>
<td>MDETR [13]</td>
<td>ResNet-101</td>
<td>-</td>
<td>32.27</td>
<td>0.0137</td>
<td>0.0183</td>
</tr>
<tr>
<td>CLIPSeg [30]</td>
<td>ViT-B/16</td>
<td>-</td>
<td>17.75</td>
<td>0.0064</td>
<td>0.0101</td>
</tr>
<tr>
<td>CLIPMat</td>
<td>ViT-B/16</td>
<td>-</td>
<td>9.91</td>
<td>0.0028</td>
<td>0.0057</td>
</tr>
<tr>
<td>CLIPMat</td>
<td>ViT-B/16</td>
<td>yes</td>
<td>9.13</td>
<td>0.0026</td>
<td>0.0052</td>
</tr>
<tr>
<td>CLIPMat</td>
<td>ViT-L/14</td>
<td>-</td>
<td>8.51</td>
<td>0.0022</td>
<td>0.0049</td>
</tr>
<tr>
<td>CLIPMat</td>
<td>ViT-L/14</td>
<td>yes</td>
<td>8.29</td>
<td>0.0022</td>
<td>0.0027</td>
</tr>
</tbody>
</table>

ture to layer norm and a multilayer perception, obtaining the feature of size $\mathbb{R}^{D' \times h \times w}$, where $h = \frac{H}{P}$ and $W = \frac{W}{P}$. The output feature is used as the input to the semantic decoder. Since it has already encoded high-level visual semantic information, we only use two convolution blocks to predict the trimap. Each contains two convolution layers and a bilinear upsampling layer with a stride 4. We adopt the cross-entropy loss in the semantic decoder following [19].

4.4. MDE: Multi-level Details Extractor

Same as TSP, we also abandon the final projection layer from the CLIP image and text encoder. Since the matting detail decoder requires local detail information to generate meticulous alpha matte, we design the MDE to extract useful local details from both the original image and multi-level features from the image encoder. Specifically, we take the output features from all four transformer blocks in the CLIP image encoder, denoted as $x_i$, where $i \in \{1, 2, 3, 4\}$. For each $x_i$, we pass it and the original image $X_m$ to MDE. For $x_1$, we first reshape and then normalize it by a $1 \times 1$ convolution layer. For $x_m$, we first normalize it by a $1 \times 1$ convolution layer and then down-sample it to the same size as $x_1$ via max pooling. They are concatenated to form $x_f$ and fed into a convolution layer, a batch norm layer, and a ReLU activation layer. Finally, the output feature is used as the input to the corresponding decoder layer at each level via a residual connection. Following [19], we use the alpha loss and Laplacian loss in the matting details decoder. The outputs from the two decoders are merged through the collaboration module [19] to get the final output, supervised by the alpha loss and Laplacian loss. More details of the method can be found in the supplementary materials.

5. Experiments

5.1. Experiment Settings

Since there are no prior methods designed for the new RIM task, we choose state-of-art methods from relevant tasks, i.e., CLIPSeg [30] and MDETR [13], which are two representative methods for the RIS and RES tasks, for benchmarking. All the methods, and CLIPMat are trained on the RefMatte training set and evaluated in two settings, i.e., the keyword-based setting and expression-based setting.

Implementation details We resize the image to $512 \times 512$ and adopt data augmentation following [19] to reduce the domain gap of composite images. We use the Adam optimizer. We train CLIPMat on two NVIDIA A100 GPUs with the learning rate fixed to 1e-4. For the ViT-B/16 backbone, the batch size is 12 and is trained for 50 epochs (about 1 day). For the ViT/L-14 backbone, the batch size is 4 and is trained for 50 epochs (about 3 days). For CLIPSeg [30] and MDETR [13], we use the code and the weights pre-trained on VGPhraseCut [50] provided by the authors for training them. However, we have not pre-trained CLIPMat on VGPhraseCut since we find that directly training it on RefMatte could provide better performance.

Evaluation metrics Following the common practice in previous matting methods [18,19,53], we use the sum of absolute differences (SAD), mean squared error (MSE), and mean absolute difference (MAD) as evaluation metrics, which are averaged over all the entities in the test set.

Mattng refiner To further improve the details of alpha matte, we propose a coarse map-based matting method as an optional post-refiner. Specifically, we modify P3M [18] to receive the original image and the predicted alpha matte as input and train it on RefMatte to refine the alpha matte.

5.2. Main Results

5.2.1 Keyword-based Setting

We evaluate MDETR [13], CLIPSeg [30], and CLIPMat on the keyword-based setting of the RefMatte test set, and show the quantitative results in Table 2. As can be seen, CLIPMat outperforms MDETR and CLIPSeg by a large margin using either the ViT-B/16 or ViT-L/14 backbone, validating the superiority of the proposed baseline method. Besides, we also show that using a larger backbone and the refiner could deliver better results. The best CLIPMat model reduces error of MDETR by about 75% and the error of CLIPSeg by about 50%, owing to the special design of the three modules. As seen from the top row in Figure 5, with the input keyword dandelion, CLIPMat is able to extract the very fine details of the target from the background with a
similar color. However, both CLIPSeg and MDETR fail in this case, producing incomplete and blurry alpha mattes.

5.2.2 Expression-based Setting

We also evaluate these models on the RefMatte test set and RefMatte-RW100 under the expression setting. Similar to the keyword-based setting, the results in Table 2 also demonstrate the superiority of CLIPMat over MDETR and CLIPSeg, e.g., the best CLIPMat model reduces the error of MDETR on the RefMatte test set by over 50% and the error of CLIPSeg on RefMatte-RW100 by about 60%. Again, using a larger backbone and the refiner help reduce the error. As seen from the second row in Figure 5, CLIPMat outperforms others in extracting the fine details of the flame, which are very close to the ground truth. The test image in the third row is from RefMatte-RW100. Compared with CLIPSeg, which produces the wrong foreground, CLIPMat is able to find the right foreground by pop-upping the correct visual semantic feature owing to the TSP module. The MDE module helps CLIPMat preserve more details, e.g., the woman’s hair, compared with MDETR. The results show the good generalization ability of CLIPMat on real-world images and confirm the value of the proposed RefMatte dataset.

5.3. Ablation Studies

We conduct ablation studies to validate the effectiveness of our proposed modules. The experiments are carried out in the keyword-based setting of RefMatte. We show the results in Table 3. We can see that each module contributes to performance improvement in terms of all the metrics, e.g., the combination of MDE and TSP reduces the SAD from 22.88 to 14.55. The use of CP further reduces the SAD to 9.91, validating that the customized matting prefix and the learnable queries provide useful context for the text encoder to understand the language instruction for image matting.

<table>
<thead>
<tr>
<th>TSP</th>
<th>MDE</th>
<th>Pre-CP</th>
<th>Post-CP</th>
<th>SAD</th>
<th>MSE</th>
<th>MAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>22.88</td>
<td>0.0097</td>
<td>0.0131</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>18.28</td>
<td>0.0068</td>
<td>0.0105</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>14.55</td>
<td>0.0050</td>
<td>0.0083</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>11.48</td>
<td>0.0036</td>
<td>0.0065</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.91</td>
<td>0.0028</td>
<td>0.0057</td>
</tr>
</tbody>
</table>

Table 3. Ablation studies results. TSP: text-driven semantic pop-up; MDE: multi-level details extractor; Pre-/Post-CP: pre or post context-embedded prompt. We use ViT-B/16 as the backbone.

6. Conclusion

In this paper, we define a novel task named referring image matting (RIM), establish a large-scale dataset RefMatte, and provide a baseline method CLIPMat. RefMatte provides a suitable test bed for the study of RIM, thanks to its large scale, high-quality images, and abundant annotations, as well as two well-defined experiment settings. Together with the RefMatte-RW100, they can be used for both in-domain and out-of-domain generalization evaluation. Besides, the CLIPMat shows the value of special designs for the RIM task and serves as a valuable reference to the model design. We hope this study could provide useful insights to the image matting community and inspire more follow-up research.

2We show more ablation studies, experiment details, failure cases, and more visual results in the supplementary materials.
References


