MODNet: Real-Time Trimap-Free Portrait Matting via Objective Decomposition

Zhanghan Ke¹,², Jiayu Sun¹, Kaican Li², Qiong Yan², Rynson W.H. Lau¹
¹Department of Computer Science, City University of Hong Kong  ²SenseTime Research
{zhanghake2-c,jiayusun5-c}@my.cityu.edu.hk, {likaican,yanqiong}@sensetime.com, rynson.lau@cityu.edu.hk

Abstract

Existing portrait matting methods either require auxiliary inputs that are costly to obtain or involve multiple stages that are computationally expensive, making them less suitable for real-time applications. In this work, we present a lightweight matting objective decomposition network (MODNet) for portrait matting in real-time with a single input image. The key idea behind our efficient design is by optimizing a series of sub-objectives simultaneously via explicit constraints. In addition, MODNet includes two novel techniques for improving model efficiency and robustness. First, an Efficient Atrous Spatial Pyramid Pooling (e-ASPP) module is introduced to fuse multi-scale features for semantic estimation. Second, a self-supervised sub-objectives consistency (SOC) strategy is proposed to adapt MODNet to real-world data to address the domain shift problem common to trimap-free methods. MODNet is easy to be trained in an end-to-end manner. It is much faster than contemporaneous methods and runs at 67 frames per second on a 1080Ti GPU. Experiments show that MODNet outperforms prior trimap-free methods by a large margin on both Adobe Matting Dataset and a carefully designed photographic portrait matting (PPM-100) benchmark proposed by us. Further, MODNet achieves remarkable results on daily photos and videos. Our code and models are available at: https://github.com/ZHKKKe/MODNet, and the PPM-100 benchmark is released at: https://github.com/ZHKKKe/PPM.

Introduction

Portrait matting aims to predict a precise alpha matte that can be used to extract the persons in a given image or video. It has a wide variety of applications, such as photo editing and movie re-creation. Most existing matting methods use a pre-defined trimap as a priori to produce an alpha matte (Cai et al. 2019; Hou and Liu 2019; Li and Lu 2020; Lu et al. 2019; Tang et al. 2019; Xu et al. 2017). However, trimaps are costly to annotate. Although a depth sensor (Foix, Alenyà, and Torras 2011) can ease the task, the resulting trimaps may suffer from low precision. Some recent trimap-free methods attempt to eliminate the model dependence on the trimap. For example, background matting (Sengupta et al. 2020) replaces the trimap by a separate background image. Other methods (Chen et al. 2018; Liu et al. 2020; Shen et al. 2016) include multiple stages (i.e., with several independent models that are optimized sequentially) to first generate a pseudo trimap or semantic mask, which is then used to serve as the priori for alpha matte prediction. Nonetheless, using the background image as input has to take and align two photos, while having multiple stages would significantly increase the inference time. These drawbacks make all aforementioned matting methods not suitable for real-time applications, such as camera preview. Besides, limited by insufficient amount of labeled training data, existing trimap-free methods often suffer from the domain shift problem (Sun, Feng, and Saenko 2016) in practice, i.e., the models cannot generalize well to real-world data.

In this work, we present MODNet, a light-weight model for real-time trimap-free portrait matting. As illustrated in Fig. 1, unlike prior methods which require auxiliary inputs or consist of multiple stages, MODNet inputs a single RGB image and applies explicit constraints to solve matting sub-objectives simultaneously in one stage. There are two insights behind our design. First, applying explicit constraints to different parts of the model can address portrait matting effectively under a single input image. In contrast, to obtain comparable results, auxiliary inputs would be necessary for the model trained by only one matting constraint. Second, optimizing sub-objectives simultaneously can further exploit the model capability by sharing intermediate representations. In contrast, training multiple stages sequentially will accumulate the errors from the early stages and magnify them in subsequent stages. We further propose two novel techniques to improve MODNet’s efficiency and robustness, including (1) an Efficient Atrous Spatial Pyramid Pooling (e-ASPP) module for fast multi-scale feature fusion in portrait semantic estimation, and (2) a self-supervised strategy based on sub-objective consistency (SOC) to alleviate the domain shift problem in real-world data.

MODNet has several advantages over previous trimap-free methods. First, MODNet is much faster, running at 67 frames per second (fps) on a GTX 1080Ti GPU with an input size of $512 \times 512$ (including data loading). Second, MODNet achieves state-of-the-art results on both open source Adobe Matting benchmark and our newly proposed PPM-100 benchmark. Third, MODNet can be easily optimized end-to-end as it is a single model instead of a complex
pipeline. Finally, MODNet has better generalization ability, due to our SOC strategy.

Since open-source portrait matting datasets (Shen et al. 2016; Xu et al. 2017) are typically small and have limited precision, prior works train and validate their models on private datasets of diverse quality and difficulty levels. As a result, it is difficult to conduct a fair evaluation. In this work, we evaluate existing trimap-free methods under the same environment, i.e., all models are trained on the same dataset and validated on the portrait images from Adobe Matting Dataset (Xu et al. 2017) and our newly proposed benchmark. Our benchmark is labelled in high quality, and is more diverse than those used in previous works.

In summary, we present a light-weight network architecture, named MODNet, for real-time trimap-free portrait matting. MODNet includes two novel techniques, an e-ASPP module for efficient semantic feature fusion and a self-supervised SOC strategy to generalize MODNet to new data domain. In addition, we have also designed a new validation benchmark for portrait matting. We will release our code, pre-trained model, and benchmark to the public.

Related Works

Image Matting

The purpose of the image matting task is to extract the desired foreground $F$ from a given image $I$. This task predicts an alpha matte with a precise foreground probability value $\alpha$ for each pixel $i$ as:

$$I^i = \alpha^i F^i + (1 - \alpha^i) B^i,$$

where $B$ is the background of $I$. This problem is ill-posed since all variables on the right hand side are unknown. Most existing matting methods take a pre-defined trimap as an auxiliary input, which is a mask containing three regions: absolute foreground ($\alpha = 1$), absolute background ($\alpha = 0$), and unknown area ($\alpha = 0.5$). They aim to estimate the foreground probability inside the unknown area only, based on the prior from the other two regions.

Traditional matting algorithms heavily rely on low-level features, e.g., color cues, to determine the alpha matte through sampling (Chuang et al. 2001; Feng, Liang, and Zhang 2016; Gustaf and Oliveira 2010; He et al. 2011; Johnson et al. 2016; Karacan, Erdem, and Erdem 2015; Ruzon and Tomasi 2000; Yang et al. 2018) or propagation (Aksyoy, Aydin, and Pollefeys 2017; Aksyoy et al. 2018; Bai and Sapiro 2007; Chen, Li, and Tang 2013; Grady et al. 2005; Levin, Lischinski, and Weiss 2007; Levin, Rav-Acha, and Lischinski 2008; Sun et al. 2004), which often fail in complex scenes. With the tremendous progress of deep learning, many methods based on CNNs have been proposed with notable successes. Cho et al. (Cho, Tai, and Kweon 2016) and Shen et al. (Shen et al. 2016) combined the classic algorithms with CNNs for alpha matte refinement. Xu et al. (Xu et al. 2017) proposed an auto-encoder architecture to predict an alpha matte from a RGB image and a trimap. Some works (Li and Lu 2020; Lu et al. 2019) used attention mechanisms and showed the advantages of using the refined trimaps.

Since creating trimaps requires users’ additional efforts and the quality of the resulting mattes depends on users’ experiences, some recent methods (including our MODNet) attempt to avoid them, as described below.

Trimap-Free Portrait Matting

Image matting is extremely difficult if trimaps are not available, as a semantic estimation step will then be needed to locate the foreground, before an alpha matte can be predicted. To constrain the problem, most trimap-free methods focus on just one type of foreground objects, e.g., portraits. Nevertheless, feeding RGB images to a single network often produce unsatisfactory alpha mattes. Sengupta et al. (Sengupta et al. 2020) proposed to capture a less expensive background image as a pseudo green screen to alleviate this problem. Other works designed their pipelines with multiple stages. For example, Shen et al. (Chen et al. 2018) assembled a trimap generation network before the matting network. Zhang et al. (Zhang et al. 2019) applied a fusion network to combine the predicted foreground and background. Liu et al. (Liu et al. 2020) concatenated three networks to utilize coarse labeled data in matting.

The main problem of all these methods is that they cannot be used
in interactive applications as: (1) the background images may change across frames, and (2) computationally expensive due to having multiple stages in the pipeline. Compared to the above methods, MODNet is light-weight in terms of pipeline complexity. It takes only one RGB image as input and uses a single model for real-time matting with better performances, by optimizing a series of sub-objectives simultaneously with explicit constraints.

**Other Techniques**

We briefly discuss two techniques that are relevant to the design and optimization of our method.

**Atrous Spatial Pyramid Pooling (ASPP).** ASPP (Chen et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks. Although ASPP has a huge number of parameters and a high computational overhead, some matting models (Qiao et al. 2020; Li et al. 2017) has been widely explored and proved to boost the performance notably in segmentation-based tasks.

**Efficient ASPP (e-ASPP).** Semantic masks predicted by MobileNetV2 may have holes in some challenging foregrounds or backgrounds. Many prior works showed that ASPP was a feasible solution for improving such erroneous semantics. However, ASPP has a very high computational overhead. To balance between performance and efficiency,
we design an efficient ASPP (e-ASPP) module to process \( S(I) \), as illustrated in Fig. 3.

The standard ASPP utilizes atrous convolutions for multi-scale feature extraction and applies a standard convolution for multi-scale feature fusion. We modify it to e-ASPP via three steps. First, we split each atrous convolution into a depth-wise atrous convolution and a point-wise convolution. The depth-wise atrous convolution extracts multi-scale features independently on each channel, while the point-wise convolution is appended for inter-channel fusion at each scale. Second, we switch the order of inter-channel fusion and the multi-scale feature fusion. In this way, (1) only one inter-channel fusion is required, and (2) the multi-scale feature fusion is converted to a cheaper depth-wise operation. Third, we compress the number of input channels by 4 \( \times \) for e-ASPP and recover it before the output.

Compared to the original ASPP, our proposed e-ASPP has only 1% of the parameters and 5% of the computational overhead. In MODNet, our experiments show that e-ASPP can achieve performance comparable to ASPP.

**Detail Prediction**

We process a transition region around the foreground portrait with a high-resolution branch \( D \), which takes \( I, S(I) \), and the low-level features from \( S \) as inputs. The purpose of reusing the low-level features is to reduce the computational overhead of \( D \). In addition, we further simplify \( D \) in the following three aspects: (1) \( D \) consists of fewer convolutional layers than \( S \); (2) a small channel number is chosen for the convolutional layers in \( D \); (3) we do not maintain the original input resolution throughout \( D \). In practice, \( D \) consists of 12 convolutional layers, and its maximum channel number is 64. The resolution of the feature maps is reduced to 1/4 of \( I \) in the first layer and restored in the last two layers. The impact of the downsampling operation on \( D \) is negligible since it contains a skip link.

We denote the outputs of \( D \) as \( D(I, S(I)) \), which implies the dependency between sub-objectives — high-level portrait semantics \( S(I) \) is a priori for detail prediction. We calculate the boundary detail matte \( \alpha_g \) from \( D(I, S(I)) \) and learn it through the L1 loss as:

\[
L_d = m_d \|d_p - \alpha_g\|_1,
\]

where \( m_d \) is a binary mask to let \( L_d \) focus on the portrait boundaries, \( m_d \) is generated through dilation and erosion on \( \alpha_g \). Its values are 1 if the pixels are inside the transition region, and 0 otherwise.

**Semantic-Detail Fusion**

The fusion branch \( F \) in MODNet is a straightforward CNN module, combining semantics and details. We first upsample \( S(I) \) to match its size with \( D(I, S(I)) \). We then concatenate \( S(I) \) and \( D(I, S(I)) \) to predict the final alpha matte \( \alpha_p \), constrained by:

\[
L_\alpha = \|\alpha_p - \alpha_g\|_1 + L_c,
\]

where \( L_c \) is the compositional loss from (Xu et al. 2017). It measures the absolute difference between input image \( I \) and the composited image obtained from \( \alpha_p \), the ground truth foreground, and the ground truth background.

MODNet is trained end-to-end through the sum of \( L_s, L_d, \) and \( L_\alpha \), as:

\[
L = \lambda_s L_s + \lambda_d L_d + \lambda_\alpha L_\alpha,
\]

where \( \lambda_s, \lambda_d, \) and \( \lambda_\alpha \) are hyper-parameters balancing the three losses. The training process is robust to these hyper-parameters. We set \( \lambda_s = \lambda_\alpha = 1 \) and \( \lambda_d = 10 \).

**SOC for Real-World Data**

The training data for portrait matting requires excellent labeling in the hair area, which is difficult to do for natural images with complex backgrounds. Currently, most annotated data comes from photography websites. Although these images have monochromatic or blurred backgrounds, the labeling process still needs to be completed by experienced annotators with considerable amount of time. As such, the labeled datasets for portrait matting are usually small. Xu et al. (Xu et al. 2017) suggested using background replacement as a data augmentation to enlarge the training set, and it has become a common setting in image matting. However, the training samples obtained in such a way exhibit different properties from those of the daily life images. Therefore, existing trimap-free models always tend to overfit the training set and perform poorly on real-world data.

To address this domain shift problem, we propose to utilize sub-objectives consistency (SOC) to adapt MODNet to unseen data distributions. The three sub-objectives in MODNet should have consistent outputs in semantics or details. We take semantic consistency as an example, MODNet outputs portrait semantics \( S(I) \) and alpha matte \( F(S(I), D(S(I))) \) for input image \( I \). As \( S(I) \) is the prior of \( F(S(I), D(S(I))) \), they should have consistent portrait semantics. In the labeled source domain, there is good consistency among the predictions of sub-objectives. However, inconsistent predictions occur in the unlabeled target domain, which may cause poor results. Motivated by this
Our Benchmark

Observation, our self-supervised SOC strategy imposes the consistency constraints among the predictions of the sub-objectives (Fig. 1(b)) to improve the performance of MODNet in the new domain, without ground truth labels. Formally, we denote MODNet as $M$. As described in Sec. , $M$ has three outputs for an unlabeled image $\hat{I}$:

$$\tilde{s}_p, \tilde{d}_p, \tilde{\alpha}_p = M(\hat{I}).$$

(6)

We enforce the semantics in $\tilde{\alpha}_p$ to be consistent with $\tilde{s}_p$ and the details in $\tilde{\alpha}_p$ to be consistent with $\tilde{d}_p$ by:

$$L_{cons} = \frac{1}{2} \left\| G(\tilde{\alpha}_p) - \tilde{s}_p \right\|_2 + \tilde{m}_d \left\| \tilde{\alpha}_p - \tilde{d}_p \right\|_1,$$

(7)

where $\tilde{m}_d$ indicates the transition region in $\tilde{\alpha}_p$, and $G$ has the same meaning as the one in Eq. 2. However, adding the L2 loss to blurred $G(\tilde{\alpha}_p)$ will smooth the boundaries in the optimized $\tilde{\alpha}_p$. As a result, the consistency between $\tilde{\alpha}_p$ and $\tilde{d}_p$ will remove the details predicted by the high-resolution branch. To prevent this problem, we duplicate $M$ to $M'$ and fix the weights of $M'$ before performing SOC. Since the fine boundaries are preserved in $\tilde{d}_p'$ output by $M'$, we append an extra regularization term to maintain the details in $M$ as:

$$L_{dd} = \tilde{m}_d \left\| \tilde{d}_p' - \tilde{d}_p \right\|_1.$$

(8)

The sum of $L_{cons}$ and $L_{dd}$ is optimized during SOC.

Experiments

In this section, we first introduce our PPM-100 benchmark for portrait matting. We then compare MODNet with existing matting methods on both Adobe Matting Dataset (AMD) (Xu et al. 2017) and our PPM-100. We further conduct ablation experiments to evaluate various components of MODNet. Finally, we demonstrate the effectiveness of SOC in adapting MODNet to real-world data.

Photographic Portrait Matting Benchmark

Existing works constructed their validation benchmarks from a small amount of labeled data through image synthesis. Their benchmarks are relatively easy due to unnatural fusion or mismatched semantics between the foreground and the background (Fig. 4(a)). Hence, trimap-free models may have comparable performances to the trimap-based models on these benchmarks, but unsatisfactory performances on natural images, i.e., images without background replacement. This indicates that the performances of trimap-free methods have not been accurately assessed.

In contrast, we propose a Photographic Portrait Matting benchmark (PPM-100), which contains 100 finely annotated portrait images with various backgrounds. To guarantee sample diversity, we consider several factors in order to balance the sample types in PPM-100, including: (1) whether the whole portrait body is included; (2) whether the person is holding additional objects. We regard small objects held by a foreground person as a part of the foreground, which is more in line with practical applications. As shown in Fig. 4(b)(c)(d), the samples in PPM-100 have more natural backgrounds and richer postures. Hence, PPM-100 can be considered as a more comprehensive benchmark.

Results on AMD and PPM-100

We compare MODNet with trimap-free FDMPA (Zhu et al. 2017), LFM (Zhang et al. 2019), SHM (Chen et al. 2018), BSHM (Liu et al. 2020), and HAtt (Qiao et al. 2020). We use DIM (Xu et al. 2017) and IndexMatter (Lu et al. 2019) as the trimap-based baselines. For methods without publicly available codes, we follow their papers to reproduce them.

For a fair comparison, we train all models on the same dataset, which contains nearly 3,000 annotated foregrounds. Background replacement (Xu et al. 2017) is applied to extend our training set. For each labeled foreground, we generate 5 samples by random cropping and 10 samples by compositing with the images from the OpenImage dataset (Kuznetsova et al. 2018) (as the background). We use MobileNetV2 pre-trained on the Supervisely Person Segmentation (SPS) (supervise.ly 2018) dataset as the backbone of all trimap-free models. For the compared methods, we explore the optimal hyper-parameters through grid search. For
Figure 5: Visual Comparison of Trimap-free Methods on PPM-100. MODNet performs better in hollow structures (the 1st row) and hair details (the 2nd row). However, it may still make mistakes in challenging poses or costumes (the 3rd row). DIM (Xu et al. 2017) here does not take trimaps as input, but is pre-trained on the SPS (supervise.ly 2018) dataset.

Table 1: Quantitative Results on PPM-100. A ‘†’ indicates that the model is pre-trained on SPS.

| Method                   | Trimap | MSE ↓  | MAD ↓ |
|--------------------------|--------|--------|-------|
| DIM (Xu et al. 2017)     | ✓      | 0.0016 | 0.0067|
| IndexMatter (Lu et al. 2019) | ✓      | 0.0015 | 0.0064|
| MODNet (Our)             | ✓      | 0.0013 | 0.0054|
| DIM (Xu et al. 2017)     |        | 0.0221 | 0.0327|
| FDMPA† (Zhu et al. 2017) |        | 0.0101 | 0.0160|
| LFM† (Zhang et al. 2019) |        | 0.0094 | 0.0158|
| SHM† (Chen et al. 2018)  |        | 0.0072 | 0.0152|
| HAtt† (Qiao et al. 2020) |        | 0.0067 | 0.0137|
| BSHM† (Liu et al. 2020)  |        | 0.0063 | 0.0114|
| MODNet† (Our)            |        | 0.0044 | 0.0086|

Table 2: Quantitative Results on AMD. We pick the portrait foregrounds from AMD for validation. A ‘†’ indicates that the models pre-trained on SPS.

| Method                   | Trimap | MSE ↓  | MAD ↓ |
|--------------------------|--------|--------|-------|
| DIM (Xu et al. 2017)     | ✓      | 0.0014 | 0.0069|
| IndexMatter (Lu et al. 2019) | ✓      | 0.0013 | 0.0066|
| MODNet (Our)             | ✓      | 0.0011 | 0.0063|
| DIM (Xu et al. 2017)     |        | 0.0075 | 0.0159|
| DIM† (Xu et al. 2017)    |        | 0.0048 | 0.0116|
| FDMPA† (Zhu et al. 2017) |        | 0.0047 | 0.0115|
| LFM† (Zhang et al. 2019) |        | 0.0043 | 0.0101|
| SHM† (Chen et al. 2018)  |        | 0.0031 | 0.0092|
| HAtt† (Qiao et al. 2020) |        | 0.0034 | 0.0094|
| BSHM† (Liu et al. 2020)  |        | 0.0029 | 0.0088|
| MODNet† (Our)            |        | 0.0023 | 0.0077|

MODNet, we train it by SGD for 40 epochs. With a batch size of 16, the initial learning rate is set to 0.01 and is multiplied by 0.1 every 10 epochs. We use Mean Square Error (MSE) and Mean Absolute Difference (MAD) as quantitative metrics.

Table 1 shows the results on PPM-100. MODNet outperforms other trimap-free methods on both MSE and MAD. However, it is unable to outperform trimap-based methods, as PPM-100 contains samples with very challenging poses and costumes. When taking a trimap as input during both training and testing stages, i.e., regarding MODNet as a trimap-based method, it outperforms the compared trimap-based methods. This demonstrates the superiority of the proposed architecture. Fig. 5 shows visual comparison.

Table 2 shows the results on AMD (Xu et al. 2017). We pick the portrait foregrounds from AMD for validation. A ‘†’ indicates that the models pre-trained on SPS.

We further evaluate MODNet on model size and execution efficiency. A small model facilitates deployment on mobile/handheld devices, while high execution efficiency is necessary for real-time applications. We measure the model size by the total number of parameters, and we reflect the execution efficiency by the average inference time over PPM-100 on an NVIDIA GTX 1080Ti GPU (all input images are resized to 512 × 512). Note that fewer parameters do not imply faster inference speed due to large feature maps or time-consuming mechanisms, e.g., attention, that the model may use. Fig. 6 summarizes the results. The inference time of MODNet is 14.9 ms (67 fps), which is twice the fps of the fastest method, FDMPA (31 fps). In addition, our MODNet
Table 3: Ablation of MODNet on PPM-100. SPS indicates the model was pre-trained on SPS.

| $\mathcal{L}_s$ | $\mathcal{L}_d$ | e-ASPP | SPS   | MSE ↓ | MAD ↓ |
|-----------------|-----------------|--------|-------|-------|-------|
|✓                |✓                |        |       | 0.0162| 0.0235|
|✓                |✓                |✓       |       | 0.0097| 0.0158|
|✓                |✓                |✓✓      |       | 0.0083| 0.0142|
|✓                |✓                |✓✓✓     |       | 0.0057| 0.0109|
|✓                |✓                |✓✓✓✓    |       | 0.0044| 0.0086|

The paper has presented a simple, fast, and effective model, MODNet, for portrait matting. By taking only an RGB image as input, our method enables the prediction of a high-quality alpha matte in real time, which is benefited from objective decomposition and concurrent optimization with explicit supervisions. Besides, we have introduced (1) an e-ASPP module to speed up the multi-scale feature fusion process, and (2) a self-supervised sub-objectives consistency (SOC) strategy to allow MODNet to handle the domain shift problem. Extensive experiments show that MODNet outperforms existing trimap-free methods on the AMD benchmark, the proposed PPM-100 benchmark, and a variety of real-world data. Our method does have limitations. The main one is that it may fail to handle videos with strong motion blurs due to the lack of temporal information. One possible future work is to address the video matting problem under motion blurs through additional sub-objectives, such as optical flow estimation.
References

Aksoy, Y.; Aydin, T. O.; and Pollefeys, M. 2017. Designing effective inter-pixel information flow for natural image matting. In CVPR.

Aksoy, Y.; Oh, T.-H.; Paris, S.; Pollefeys, M.; and Matusik, W. 2018. Semantic soft segmentation. TOG.

Bai, X.; and Sapiro, G. 2007. A geodesic framework for fast interactive image and video segmentation and matting. In ICCV.

Cai, S.; Zhang, X.; Fan, H.; Huang, H.; Liu, J.; Liu, J.; Liu, J.; Wang, J.; and Sun, J. 2019. Disentangled Image Matting. In ICCV.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. PAMI, 40(4): 834–848.

Chen, Q.; Ge, T.; Xu, Y.; Zhang, Z.; Yang, X.; and Gai, K. 2018. Semantic human matting. In ACMMM.

Chen, Q.; Li, D.; and Tang, C.-K. 2013. KNN Matting. PAMI.

Cho, D.; Tai, Y.-W.; and Kweon, I. 2016. Natural image matting using deep convolutional neural networks. In ECCV.

Chuang, Y.-Y.; Curless, B.; Salesin, D. H.; and Szeliski, R. 2001. A bayesian approach to digital matting. In CVPR.

Feng, X.; Liang, X.; and Zhang, Z. 2016. A cluster sampling method for image matting via sparse coding. In ECCV.

Foix, S.; Alenyà, G.; and Torras, C. 2011. Lock-in Time-of-Flight (ToF) cameras: A survey. Sensors Journal.

Gastal, E. S. L.; and Oliveira, M. M. 2010. Shared sampling for real-time alpha matting. In Eurographics.

Grady, L.; Schiwietz, T.; Aharon, S.; and Westermann, R. 2005. Random walks for interactive alpha-matting. In VIIP.

He, K.; Rhaemann, C.; Rother, C.; Tang, X.; and Sun, J. 2011. A global sampling method for alpha matting. In CVPR.

Hou, Q.; and Liu, F. 2019. Context-aware Image Matting for Simultaneous Foreground and Alpha Estimation. In ICCV.

Johnson, J.; Varnousfaderani, E. S.; Cholakkal, H.; and Rajan, D. 2016. Sparse coding for alpha matting. TIP.

Karacan, L.; Erdem, A.; and Erdem, E. 2015. Image matting with kl-divergence based sparse sampling. In ICCV.

Ke, Z.; Qiu, D.; Li, K.; Yan, Q.; and Lau, R. W. 2020. Guided Collaborative Training for Pixel-wise Semi-Supervised Learning. In ECCV.

Kuznetsova, A.; Rom, H.; Aldrinn, N.; Uijlings, J. R. R.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Duerig, T.; and Ferrari, V. 2018. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. IJCV.

Levin, A.; Lischinski, D.; and Weiss, Y. 2007. A closed-form solution to natural image matting. PAMI.

Levin, A.; Rav-Acha, A.; and Lischinski, D. 2008. Spectral matting. PAMI.

Li, Y.; and Lu, H. 2020. Natural image matting via guided contextual attention. In AAAI.

Liu, J.; Yao, Y.; Hou, W.; Cui, M.; Xie, X.; Zhang, C.; and Hua, X.-S. 2020. Boosting Semantic Human Matting With Coarse Annotations. In CVPR.

Lu, H.; Dai, Y.; Shen, C.; and Xu, S. 2019. Indices Matter: Learning to Index for Deep Image Matting. In ICCV.

Qiao, Y.; Liu, Y.; Yang, X.; Zhou, D.; Xu, M.; Zhang, Q.; and Wei, X. 2020. Attention-Guided Hierarchical Structure Aggregation for Image Matting. In CVPR.

Ruzon, M. A.; and Tomasi, C. 2000. Alpha estimation in natural images. In CVPR.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In CVPR.

Schmarje, L.; Santarossa, M.; Schröder, S.-M.; and Koch, R. 2020. A survey on Semi-, Self- and Unsupervised Learning for Image Classification. ArXiv, abs/2002.08721.

Sengupta, S.; Jayaram, V.; Curless, B.; Seitz, S.; and Kemelmacher-Shlizerman, I. 2020. Background Matting: The World is Your Green Screen. In CVPR.

Shen, X.; Tao, X.; Gao, H.; Zhou, C.; and Jia, J. 2016. Deep automatic portrait matting. In ECCV.

Sun, B.; Feng, J.; and Saenko, K. 2016. Return of Frustratingly Easy Domain Adaptation. In AAAI.

Sun, J.; Jia, J.; Tang, C.-K.; and Shum, H.-Y. 2004. Poisson matting. TOG.

supervise.ly. 2018. Supervisely Person Dataset. supervise.ly.

Tang, J.; Aksoy, Y.; Oztireli, C.; Gross, M.; and Aydin, T. O. 2019. Learning-based Sampling for Natural Image Matting. In CVPR.

Toldo, M.; Michieli, U.; Agresti, G.; and Zanuttigh, P. 2020. Unsupervised Domain Adaptation for Mobile Semantic Segmentation based on Cycle Consistency and Feature Alignment. IMAVIS.

Wilson, G.; and Cook, D. J. 2020. A Survey of Unsupervised Deep Domain Adaptation. TIST.

Xu, N.; Price, B.; Cohen, S.; and Huang, T. 2017. Deep Image Matting. In CVPR.

Yang, X.; Xu, K.; Chen, S.; He, S.; Yin, B. Y.; and Lau, R. 2018. Active matting. Adv. Neural Inform. Process. Syst.

Zhang, Y.; Gong, L.; Fan, L.; Ren, P.; Huang, Q.; Bao, H.; and Xu, W. 2019. A late fusion cnn for digital matting. In CVPR.

Zhu, B.; Chen, Y.; Wang, J.; Liu, S.; Zhang, B.; and Tang, M. 2017. Fast Deep Matting for Portrait Animation on Mobile Phone. In ACMMM.