Deep Automatic Natural Image Matting

Jizhizi Li, Jing Zhang and Dacheng Tao
1The University of Sydney, Australia
2JD Explore Academy, JD.com, China
jili8515@uni.sydney.edu.au, jing.zhang1@sydney.edu.au, dacheng.tao@gmail.com

Abstract
Automatic image matting (AIM) refers to estimating the soft foreground from an arbitrary natural image without any auxiliary input like trimap, which is useful for image editing. Prior methods try to learn semantic features to aid the matting process while being limited to images with salient opaque foregrounds such as humans and animals. In this paper, we investigate the difficulties when extending them to natural images with salient transparent/meticulous foregrounds or non-salient foregrounds. To address the problem, a novel end-to-end matting network is proposed, which can predict a generalized trimap for any image of the above types as a unified semantic representation. Simultaneously, the learned semantic features guide the matting network to focus on the transition areas via an attention mechanism. We also construct a test set AIM-500 that contains 500 diverse natural images covering all types along with manually labeled alpha mattes, making it feasible to benchmark the generalization ability of AIM models. Results of the experiments demonstrate that our network trained on available composite matting datasets outperforms existing methods both objectively and subjectively. The source code and dataset are available at https://github.com/JizhiziLi/AIM.

1 Introduction
Natural image matting refers to estimating a soft foreground from a natural image, which is a fundamental process for many applications, e.g., film post-production and image editing [Chen et al., 2013; Zhang and Tao, 2020]. Since image matting is a highly ill-posed problem, previous methods usually adopt auxiliary user input, e.g. trimap [Sun et al., 2004; Cui et al., 2019], scribble [Levin et al., 2007], or background image [Sengupta et al., 2020] as constraints. While traditional methods estimate the alpha value by sampling neighboring pixels [Wang and Cohen, 2007] or defining affinity metrics for alpha propagation [Levin et al., 2008], deep learning-based approaches solve it by learning discriminative representations from a large amount of labeled data and predicting alpha matte directly [Lu et al., 2019; Li and Lu, 2020].

However, extra manual effort is required while generating such auxiliary inputs, which makes these methods impractical in automatic industrial applications. To address the limitations, automatic image matting (AIM) has attracted increasing attention recently [Zhang et al., 2019], which refers to automatically extracting the soft foreground from an arbitrary natural image. Prior AIM methods [Qiao et al., 2020; Li et al., 2020] solve this problem by learning semantic features from the image to aid the matting process but is limited to images with salient opaque foregrounds, e.g., human [Shen et al., 2016; Chen et al., 2018], animal [Li et al., 2020]. It is difficult to extend them to images with salient transparent/meticulous foregrounds or non-salient foregrounds due to their limited semantic representation ability.

To address this issue, we dig into the problem of AIM by investigating the difficulties of extending it to all types of natural images. First, we propose to divide natural images into three types according to the characteristics of foreground alpha matte. As shown in Figure 1, the three types are: 1) SO

Figure 1: Matting results of GFM and our method on three types of natural images, i.e., SO, STM, NS, from the top to the bottom.
2 Rethinking the Difficulties of AIM

Matting without auxiliary inputs. Prevalent image matting methods [Sun et al., 2021; Cho et al., 2016] solve the problem by leveraging auxiliary user inputs, e.g., trimap [Tang et al., 2019], scribble [Levin et al., 2007], or background image [Sengupta et al., 2020], which provide strong constraints on the solution space. Specifically, given a trimap, the matting models only need to focus on the transition area and distinguish the details by leveraging the available foreground and background alpha matte information. However, there is usually little chance to obtain auxiliary information in real-world automatic application scenarios. Thereby, AIM is more challenging since the matting models need to understand the holistic semantic partition of foreground and background of an image, which may belong to different types as been described previously. Nevertheless, AIM is more appealing to automatic applications and worth more research efforts.

Matting on natural images. Since it is difficult to label accurate alpha mattes for natural images at scale, there are no publicly available large-scale natural image matting datasets. Usually, foreground images are obtained by leveraging chroma keying from images captured with a green background screen [Xu et al., 2017]. Nevertheless, the amount of available foregrounds is only about 1,000. To build more training images, they are synthesized with different background images from public datasets like COCO [Lin et al., 2014]. However, synthetic images contain composite artifacts and semantic ambiguity [Li et al., 2020]. Matting models trained on them may have a tendency to find cheap features from these composite artifacts and thus overfit on the synthetic training images, resulting in a poor generalization performance on real-world natural images. To address this domain gap issue between synthetic training images and natural test images, some efforts have been made in [Li et al., 2020; Hou and Liu, 2019]. As for evaluation, previous methods [Zhang et al., 2019; Qiao et al., 2020] also adopt synthetic dataset [Xu et al., 2017] for evaluating AIM models, which is a bias evaluation setting. For example, a composite image may contain multiple objects including the original foreground object in the candidate background image, but the ground truth is only the single foreground object from the foreground image. Besides, the synthetic test images may also contain composite artifacts, making it less possible to reveal the overfitting issue.

Matting on all types of images. AIM aims to extract the soft foreground from an arbitrary natural image, which may have foreground objects with either opaque interior, transparent/meticulous interior, or non-salient foregrounds like texture, fog, or water drops. However, existing AIM methods are limited to a specific type of images with opaque foregrounds, e.g., human [Shen et al., 2016; Chen et al., 2018; Zhang et al., 2019; Li et al., 2021; Ke et al., 2020] and animals [Li et al., 2020]. DAPM [Shen et al., 2016] first generates a coarse foreground shape mask and uses it as an auxiliary input for following matting process. LateFusion [Zhang et al., 2019] predicts foreground and background separately and uses them in a subsequent fusion process.

HAttMat-
ting [Qiao et al., 2020] predicts foreground profile and guides the matting process to refine the precise boundary. GFM [Li et al., 2020] predicts foreground, background, and transition area simultaneously and combines them with the matting result at the transition area as the final matte. Although they are effective for images with salient and opaque foregrounds, extending them to all types of images is not straightforward. In the context of AIM, the term “semantic” is more related to foreground and background rather than the semantic category of foreground objects. Since there are very different characteristics for the three types of images, it is hard to learn useful semantic features to recognize foreground and background, especially for images with transparent objects or non-salient foregrounds, without explicit supervisory signals.

3 Proposed Methods

3.1 Improved Backbone for Matting

In this work, we choose ResNet-34 [He et al., 2016] as the backbone network due to its lightweight and strong representation ability. However, ResNet-34 is originally designed to solve high-level classification problem, while both high-level semantic features and low-level detail features should be learned to solve AIM. To this end, we improve the vanilla ResNet-34 with a simple customized modification to make it better suitable for the AIM task.

In the first convolutional layer conv1 in ResNet-34, they use a stride of 2 to reduce the spatial dimension of feature maps. Followed by a max-pooling layer, the output dimension is 1/4 of the original image after the first block. While these two layers can significantly reduce computations and increase the receptive field, it may also lose many details, which are not ideal for image matting. To customize the backbone for AIM, we modify the stride of conv1 from 2 to 1 to keep the spatial dimension of the feature map as the original image size to retain local detail features. To retain the receptive field, we add two max-pooling layers with a stride of 2. Besides, we change the stride of all the first convolutional layers in stage1-stage4 of ResNet-34 from 2 to 1 and add a max-pooling layer with a stride of 2 accordingly. It is noteworthy that the indices in the max-pooling layers are also kept and will be used in the corresponding max-unpooling layers in the local matting decoder to preserve local details, as shown in Figure 3. We retrain the customized ResNet-34 on ImageNet and use it as our backbone network. Experiment results demonstrate it outperforms the vanilla one in terms of both objective metrics and subjective visual quality.

3.2 Unified Semantic Representation

As discussed in Section 2, the characteristics of different types of images are very different. In order to provide explicit semantic supervisory signals for the semantic decoder to learn useful semantic features and partition the image into foreground, background, and transition areas, we propose a unified semantic representation.

For an image belonging to SO type, there are always foreground, background, and transition areas. Thereby, we adopt the traditional trimap as the semantic representation, which can be generated by erosion and dilation from the ground truth alpha matte. For an image belonging to STM type, there are no explicit foreground areas. In other words, the foreground should be marked as a transition area and a soft foreground alpha matte should be estimated. Thereby, we use a duomap as its semantic representation, which is a 2-class map denoting the background and transition areas accordingly. For an image belonging to NS type, it is hard to mark all the explicit foreground and background areas, since the foreground is always entangled with the background. Thereby, we use a unimap as its semantic representation, which is a 1-class map denoting the whole image as the transition area.

To use a unified semantic representation for all the three types of images, we derive the trimap, duomap, and unimap from the traditional trimap as follows:

$$ U_i = T_i, \quad type = SO $$
$$ U_i = 1.5T_i - T_i^2, \quad type = STM $$
$$ U_i = 0.5, \quad type = NS $$(1)

where $T$ represents the traditional trimap obtained by erosion and dilation from the ground truth alpha matte. For every pixel $i$, $T_i \in \{0, 0.5, 1\}$, where the background area is 0, the foreground area is 1, and the transition area is 0.5, respectively. $U_i$ is the unified semantic representation at pixel $i$. Note that for images belonging to type STM and NS, the traditional trims always contain trivial foreground and background pixels as shown in Figure 2(b), which are very difficult for the semantic decoder to predict. Instead, our unified representations are a pure duomap or unimap as shown in Figure 2(c), which represents the holistic foreground objects or denotes there are no salient foreground objects.

In order to predict the unified semantic representation, we redesign the semantic decoder in GFM [Li et al., 2020] as shown in Figure 3. Specifically, we also use five blocks in the decoder. For each decoder block, there are three sequential $3 \times 3$ convolutional layers and an upsampling layer. To further increase the capability of the decoder for learning discriminative semantic features, we adopt the Squeeze-and-Excitation (SE) attention module [Hu et al., 2018] after each
decoder block to re-calibrate the features, thereby selecting the most informative features for predicting unified representations and filtering out the less useful ones. We also adopt pyramid pooling module (PPM) [Zhao et al., 2017] to enlarge the receptive field. The upsampled PPM features are concatenated with the output of SE module and used as the input for the next decoder block. We use the cross-entropy loss to supervise the training of the semantic decoder.

3.3 Guided Matting Process

As shown in Figure 3, there are six blocks following a U-Net structure [Ronneberger et al., 2015] in matting decoder. Each block contains three sequential $3 \times 3$ convolutional layers. The encoder feature is concatenated with first decoder block output and fed into the following block. The output then being concatenated with the corresponding encoder output and passes a max unpooling layer with reversed indices to recover fine structural details, serves as input of next block.

Motivated by previous study proving that attention mechanism can provide support on learning discriminating representations [Ma et al., 2020], we devise a spatial attention module to guide the matting process by leveraging the learned semantic features from the semantic decoder to focus on extracting details only within transition area. Specifically, the output feature from the last decoder block in the semantic decoder is used to generate spatial attention, since it is more related to semantics. Then, it goes through a max-pooling layer and average pooling layer along the channel axis, respectively. The pooled features are concatenated and go through a convolutional layer and a sigmoid layer to generate a spatial attention map. We use it to guide the matting decoder to attend to the transition area via an element-wise production operation and an element-wise sum operation.

Given the predicted unified representation $U$ and guided matting result $M$ from the semantic decoder and matting decoder, we can derive the final alpha matte $\alpha$ as follows:

$$\alpha = (1 - 2 \times |U - 0.5|) \times M + 2 \times |U - 0.5| \times U. \quad (2)$$

We adopt the commonly used alpha loss [Xu et al., 2017] and Laplacian loss [Hou and Liu, 2019] on the predicted alpha matte $M$ and the final alpha matte $\alpha$ to supervise the matting decoder. Besides, we also use the composition loss [Xu et al., 2017] on the final alpha matte $\alpha$.

4 Experiment

4.1 Benchmark: Automatic Image Matting-500

As discussed in Section 2, previous work [Zhang et al., 2019; Qiao et al., 2020] evaluate their models either on synthetic test set such as Comp-1k [Xu et al., 2017] or in-house test set of natural images limited to specific types, e.g., human and animal images [Li et al., 2020; Chen et al., 2018]. In this paper, we establish the first natural image matting test set AIM-500, which contains 500 high-resolution real-world natural images from all three types and many categories. We collect the images from free-license websites and manually label the alpha mattes with professional software. The shorter side of each image is at least 1080 pixels. In Table 1, we compare AIM-500 with other matting test sets including DAPM [Shen et al., 2016], Comp-1k [Xu et al., 2017], HAtt [Qiao et al., 2020], and AM-2k [Li et al., 2020] in terms of the volume, whether or not do they provide natural original images, the amount of three types images, and the object classes. We can see that AIM-500 is larger and more diverse than others, making it suitable for benchmarking AIM models. AIM-500 contains 100 portrait images, 200 animal images, 34 images
Figure 4: Some examples from our AIM-500 benchmark.

with transparent objects, 75 plant images, 45 furniture images, 36 toy images, and 10 fruit images. We present some examples and their alpha mattes in Figure 4. Note that due to the privacy concern, all the portrait images have no identifiable information and are ready to release.

### 4.2 Implementation Details

We trained our model and other representative matting models on the combination of matting datasets Comp-1k [Xu et al., 2017], HAtt [Qiao et al., 2020] and AM-2k [Li et al., 2020]. To reduce the domain gap issue, we adopted the high-resolution background dataset BG-20k and the composition route RSSN proposed in [Li et al., 2020] to generate the training data. We composited each foreground image from Composition-1k and HAtt with five background images and each foreground image from AM-2k with two background images. The total number of training images is 8,710. To achieve better performance, we also adopted a type-wise data augmentation and transfer learning strategy during training.

**Type-wise data augmentation.** After inspecting real-world natural images, we observed that NS images usually have a bokeh effect on the background. To simulate this effect, for all NS images, we added the blur effect on the background as RSSN to make the foreground prominent.

**Transfer learning.** Although data augmentation could increase the number of training data, the number of original foreground images and their classes are indeed small. To mitigate the issue, we leveraged the salient object detection dataset DUTS [Wang et al., 2017] for training since it contains more real-world images and classes. However, since images in DUTS have small resolutions (about 300 × 400) and little fine details, we only used it for pre-training and adopted a transfer learning strategy to finetune the pre-trained model further on the above synthetic matting dataset.

We trained our model on a single NVIDIA Tesla V100 GPU with batch size as 16, Adam as optimizer. When pre-training on DUTS, we resized all images to 320 × 320, set learning rate as 1 × 10^{-4}, trained for 100 epochs. During fine-tuning on the synthetic matting dataset, we randomly crop a patch with a size in {640 × 640, 960 × 960, 1280 × 1280} from each image and resized it to 320 × 320, set the learning rate as 1 × 10^{-6}, and trained for 50 epochs. It took about 1.5 days to train the model. We adopted the hybrid test strategy [Li et al., 2020] with the scale factors 1/3 and 1/4, respectively.

### 4.3 Objective and Subjective Results

We compare our model with several state-of-the-art matting methods including SHM [Chen et al., 2018], LF [Qiao et al., 2020], HAtt [Qiao et al., 2020], GFM [Li et al., 2020], DIM [Xu et al., 2017], and an salient object detection method U2UNET [Qin et al., 2020] on AIM-500. For LF, U2UNET, and GFM, we used the code provided by the author. For SHM, HAtt, and DIM, we re-implemented the code since they are not available. Since DIM required a trimap for training, we used the ground truth trimap as the auxiliary input and denoted it as DIM*. We adopted the same transfer learning strategy while training all the models. We chose the commonly used sum of absolute differences (SAD), mean squared error (MSE), Mean Absolute Difference (MAD), Connectivity(Conn.), Gradient(Grad.) [Rhemann et al., 2009], as the main metrics, and also calculated the SAD within transition areas, SAD by type and category for comprehensive evaluation. The results of objective metrics are summarized in Table 2. Some visual results are presented in Figure 5.

As shown in Table 2, our model achieves the best in all metrics among all AIM methods and outperforms DIM* in most of the metrics. Although DIM* performs better in some SO category, e.g., animal, human, furniture, it indeed uses ground truth trimaps as auxiliary inputs while our model does not have such a requirement. Nevertheless, our model still outperforms DIM* in the transition areas as well as on the STM and NS types, implying that the semantic decoder predicts accurate semantic representation and the matting decoder has a better ability for extracting alpha details. Besides, DIM is very sensitive to the size of trimap and may produce a bad result when there is a large transition area, e.g. boundary of the bear in Figure 5. U2UNET can estimate the rough foregrounds for SO images, but it fails to handle STM and NS images, e.g. the crystal stone and the net, implying that a single decoder is overwhelmed to learn both global semantic features and local detail features to deal with all types of images since they have different characteristics.

SHM uses a two-stage network to predict trimap and the final alpha, which may accumulate the semantic error and mislead the subsequent matting process. Consequently, it obtains large SAD errors in the whole image as well as in the transition area, i.e., 170.44 and 69.41. Some failure results can be found in the bear and crystal stone in Figure 5. LF adopts a classification network to distinguish foreground and background, but it is difficult to adapt to STM and NS images as they do not have explicit foreground and background, thereby resulting in large average SAD errors. HATT tries to learn foreground profile to support boundary detail matting. However, there are no explicit supervisory signal for the semantic branch, which makes it difficult to learn explicit semantic representations. As a strong baseline model, GFM outperforms other AIM methods, but it is still worse than us, i.e., 52.66 to 43.92, especially for STM and NS images, as seen from the crystal stone and the net. The results demonstrate that the customized designs in the network matter a lot for dealing

---

1More results can be found in the supplementary material.
Table 2: Quantitative results on AIM-500. DIM$^*$ denotes the DIM method using ground truth trimap as an extra input. Tran.: Transition Area, Transp.: Transparent, Furni.: Furniture.

| Method | SAD | MSE | MAD | Conn. | Grad. | SO | STM | NS | Avg. |
|--------|-----|-----|-----|-------|-------|----|-----|----|------|
| UNI TL MP SE SA | 81.08 | 0.0363 | 0.0480 | 88.70 | 61.02 | 43.37 | 69.69 | 211.98 | 134.09 |
| DIM$^*$ | 76.90 | 0.0296 | 0.0456 | ✓ | ✓ | ✓ | 75.90 | 210.34 | 134.09 |
| U2NET | 52.66 | 0.0213 | 0.0313 | ✓ | ✓ | ✓ | 43.92 | 210.34 | 134.09 |
| SHM | 48.52 | 0.0195 | 0.0287 | ✓ | ✓ | ✓ | 43.92 | 210.34 | 134.09 |
| LF | 48.95 | 0.0207 | 0.0293 | ✓ | ✓ | ✓ | 44.50 | 0.0158 | 0.0262 |
| GFM | 44.50 | 0.0161 | 0.0262 | ✓ | ✓ | ✓ | 44.50 | 0.0158 | 0.0262 |

Table 3: Ablation study results. UNI: unified semantic representations; TL: transfer learning; MP: backbone with max pooling; SE: SE attention; SA: spatial attention.

| Method | SAD | MSE | MAD |
|--------|-----|-----|-----|
| UNI TL MP SE SA | 81.08 | 0.0363 | 0.0480 |
| ✓ ✓ ✓ ✓ ✓ | 76.90 | 0.0296 | 0.0456 |
| ✓ ✓ ✓ ✓ ✓ | 52.66 | 0.0213 | 0.0313 |
| ✓ ✓ ✓ ✓ ✓ | 51.23 | 0.0205 | 0.0307 |
| ✓ ✓ ✓ ✓ ✓ | 48.52 | 0.0195 | 0.0287 |
| ✓ ✓ ✓ ✓ ✓ | 48.95 | 0.0207 | 0.0293 |
| ✓ ✓ ✓ ✓ ✓ | 44.50 | 0.0158 | 0.0262 |
| ✓ ✓ ✓ ✓ ✓ | 43.92 | 0.0161 | 0.0262 |

5 Conclusion

In this paper, we investigate the difficulties in automatic image matting, including matting without auxiliary inputs, matting on natural images, and matting on all types of images. We propose a unified semantic representation for all types by introducing the new concepts of duomap and unimap, proved to be useful. We devise an automatic matting network with several customized new designs to improve its capability for AIM. Moreover, we establish the first natural image test set AIM-500 to benchmark AIM models, which can serve as a test bed to facilitate future research.
References

[Cai et al., 2019] Shaofan Cai, Xiaoshuai Zhang, Haoqiang Fan, Haibin Huang, Jianguo Liu, Jiaming Liu, Jiaying Liu, Yue Wang, and Jian Sun. Disentangled image matting. In ICCV, 2019.

[Chen et al., 2013] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2175–2188, 2013.

[Chen et al., 2018] Quan Chen, Tiezhen Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In *Proceedings of the ACM International Conference on Multimedia*, pages 618–626, 2018.

[Cho et al., 2016] Donghyeon Cho, Yu-Wing Tai, and Inso Kweon. Natural image matting using deep convolutional neural networks. In *ECCV*, 2016.

[He et al., 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

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

[Hu et al., 2018] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[Ke et al., 2020] Zhanghan Ke, Kaican Li, Yurou Zhou, Qihua Wu, Xiangyu Mao, Qiong Yan, and Rynson W.H. Lau. Is a green screen really necessary for real-time portrait matting? *ArXiv*, abs/2011.11961, 2020.

[Levin et al., 2007] 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.

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

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

[Li et al., 2020] Jizhizi Li, Jing Zhang, Stephen J Maybank, and Dacheng Tao. End-to-end animal image matting. *arXiv preprint arXiv:2010.16188*, 2020.

[Li et al., 2021] Jizhizi Li, Sihan Ma, Jing Zhang, and Dacheng Tao. Privacy-preserving portrait matting. *arXiv preprint arXiv:2104.14222*, 2021.

[Lin et al., 2014] 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 *ECCV*, 2014.

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

[Ma et al., 2020] Benteng Ma, Jing Zhang, Yong Xia, and Dacheng Tao. Auto learning attention. *Advances in Neural Information Processing Systems*, 33, 2020.

[Qiao et al., 2020] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. In *CVPR*, 2020.

[Qin et al., 2020] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020.

[Rhemann et al., 2009] Christoph Rhemann, Carsten Rother, Yue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *CVPR*, 2009.

[Ronneberger et al., 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[Sengupta et al., 2020] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *CVPR*, 2020.

[Shen et al., 2016] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. Deep automatic portrait matting. In *ECCV*, 2016.

[Sun et al., 2004] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. *ACM Transactions on Graphics*, 23(3):315–321, 2004.

[Sun et al., 2021] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Semantic image matting. In *CVPR*, 2021.

[Tang et al., 2019] Jingwei Tang, Yagiz Aksoy, Cengiz Oztireli, Markus Gross, and Tunc Ozan Aydin. Learning-based sampling for natural image matting. In *CVPR*, 2019.

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

[Wang et al., 2017] Lipun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017.

[Xu et al., 2017] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *CVPR*, 2017.

[Zhang and Tao, 2020] Jing Zhang and Dacheng Tao. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 2020.

[Zhang et al., 2019] Yunke Zhang, Lixue Gong, Lubin Fan, Peiran Ren, Qixing Huang, Hujun Bao, and Weiwei Xu. A late fusion cnn for digital matting. In *CVPR*, 2019.

[Zhao et al., 2017] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.
In this supplementary material, we present the network structure detail of our model, more details about the proposed natural image matting test set AIM-500, more visual results of our model and state-of-the-art methods on AIM-500, and more analysis of the ablation study.

1 More details about our model

1.1 Network structure

Table 2 shows the detailed structure of some essential parts in our network including the shared encoder, the semantic decoder based on the unified semantic representation, and the guided matting decoder. Our network has 55.3M parameters and 191.52GMac computational complexity. It takes about 0.1633s for the network to process a 800×800 image.

2 More details about AIM-500

2.1 Compare with synthetic validation set

As we mentioned in the paper, due to the unavailability of natural test with labeled alpha mattes, previous AIM methods evaluate their results on the synthetic test set, which is generated by pasting high-resolution foregrounds from matting dataset [Xu et al., 2017; Qiao et al., 2020] on backgrounds from low-resolution COCO dataset [Lin et al., 2014] or PASCAL VOC [Everingham et al., 2010].

However, as can be seen from Figure 2, such procedure results in large bias with real-world natural images and wrong label because of the two reasons, 1) Pasting other salient objects on backgrounds will make the original ground truth unreasonable, for examples, pasting a transparent cup in front of another cup or pasting a squirrel in front of a lot of people. Such cases appear a lot in the synthetic matting test set and result in the biased evaluation of AIM models; and 2) Synthetic images have non-negligible composite artifacts, performing well on such a test set does not necessarily imply a good generalization ability on real-world images.

In contrast to them, the proposed natural image matting test set named AIM-500 contains natural images rather than composite images with manually labeled alpha mattes as shown in Figure 2. We believe AIM-500 can serve as a good test bed to benchmark AIM models.

2.2 More examples of AIM-500

AIM-500 contains 7 categories including portrait, animal, fruit, furniture, toy, plant, and transparent objects. Here we show more examples of each category in Figure 1. As can be seen, images in AIM-500 have diverse foregrounds and backgrounds.

3 More results of Experiment

3.1 More visual results on AIM-500

In Figure 4 and Figure 5, we compare our model with several state-of-the-art methods U2NET [Qin et al., 2020], SHM [Chen et al., 2018], LF [Zhang et al., 2019], GFM [Li et al., 2020], and DIM+Trimap [Xu et al., 2017] on AIM-500. It can be seen that our method achieves the best results on all three types of images. U2Net and GFM fail to extract the fine detail. SHM and LF have many errors on semantic background or foreground. Even with extra ground truth trimap as input, DIM performs worse than us especially when the trimap region is large, e.g., SO images.

3.2 More of ablation study

SE attention block We adopted five Squeeze-and-Excitation attention modules [Hu et al., 2018] in the semantic decoder to
re-calibrate the features and select the most informative features for predicting unified representations. Here we provide objective and subjective results of it. In Table 1, we show the results of IOU (Intersection over Union) and the accuracy of the predicted unified semantic representations by the semantic decoder with or without SE blocks. We also show some visual results in Figure 3. As can be seen, for all three types of images, SE blocks benefit the semantic decoder to predict better unified semantic representations.

| SE Blocks | IOU     | Accuracy |
|-----------|---------|----------|
| ×         | 0.7002  | 0.8756   |
| ✓         | 0.7393  | 0.8900   |

Table 1: IOU and accuracy of the predicted unified semantic representations with or without SE blocks in the semantic decoder.

---

Figure 2: Comparison between our proposed natural images matting test set AIM-500 and previous synthetic matting test set.

Figure 3: Visual results of the predicted unified semantic representations with or without SE blocks in the semantic decoder.
References

[Chen et al., 2018] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In Proceedings of the ACM International Conference on Multimedia, pages 618–626, 2018.

[Everingham et al., 2010] 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, 2010.

[He et al., 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[Hu et al., 2018] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In CVPR, 2018.

[Li et al., 2020] Jizhizi Li, Jing Zhang, Stephen J Maybank, and Dacheng Tao. End-to-end animal image matting. arXiv preprint arXiv:2010.16188, 2020.

[Lin et al., 2014] 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 ECCV, 2014.

[Qiao et al., 2020] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. In CVPR, 2020.

[Qin et al., 2020] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. Pattern Recognition, 106:107404, 2020.

[Xu et al., 2017] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In CVPR, 2017.

[Zhang et al., 2019] Yunke Zhang, Lixue Gong, Lubin Fan, Peiran Ren, Qixing Huang, Hujun Bao, and Weiwei Xu. A late fusion cnn for digital matting. In CVPR, 2019.

[Zhao et al., 2017] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In CVPR, 2017.
| Block | Output size | Detail |
|-------|-------------|--------|
| **Improved Backbone** |
| $E_0$ | $N \times 64 \times 320 \times 320$ | conv $(7 \times 7$, 64, stride 1, padding 3) + BN + ReLU |
| $M_0$ | $N \times 64 \times 160 \times 160$ | maxpool $(3 \times 3$, stride 2, padding 1, return indices) |
| $M_1$ | $N \times 64 \times 80 \times 80$ | maxpool $(3 \times 3$, stride 2, padding 1, return indices) |
| $E_1$ | $N \times 64 \times 40 \times 40$ | ResNet-34 [He et al., 2016] layer 1 (stride 1) |
| $M_2$ | $N \times 64 \times 40 \times 40$ | maxpool $(3 \times 3$, stride 2, padding 1, return indices) |
| $E_2$ | $N \times 128 \times 40 \times 40$ | ResNet-34 [He et al., 2016] layer 2 (stride 1) |
| $M_3$ | $N \times 128 \times 20 \times 20$ | maxpool $(3 \times 3$, stride 2, padding 1, return indices) |
| $E_3$ | $N \times 256 \times 20 \times 20$ | ResNet-34 [He et al., 2016] layer 3 (stride 1) |
| $M_4$ | $N \times 256 \times 10 \times 10$ | maxpool $(3 \times 3$, stride 2, padding 1, return indices) |
| $E_4$ | $N \times 512 \times 10 \times 10$ | ResNet-34 [He et al., 2016] layer 4 (stride 1) |
| **Unified Semantic Decoder** |
| $PPM$ | $N \times 512 \times 10 \times 10$ | PSPModule [Zhao et al., 2017] (512, multiscale=1,3,5) |
| $SD_4$ | $N \times 256 \times 20 \times 20$ | conv $(3 \times 3$, 512, stride 1) + BN + ReLU |
| $SE_4$ | $N \times 256 \times 20 \times 20$ | avgpool + linear$(256, 16)$ + ReLU + linear$(16, 256)$ + Sigmoid |
| $SD_3$ | $N \times 128 \times 40 \times 40$ | conv $(3 \times 3$, 256, stride 1) + BN + ReLU |
| $SE_3$ | $N \times 128 \times 40 \times 40$ | avgpool + linear$(128, 8)$ + ReLU + linear$(8, 128)$ + Sigmoid |
| $SD_2$ | $N \times 64 \times 80 \times 80$ | conv $(3 \times 3$, 128, stride 1) + BN + ReLU |
| $SE_2$ | $N \times 64 \times 80 \times 80$ | avgpool + linear$(64, 4)$ + ReLU + linear$(4, 64)$ + Sigmoid |
| $SD_1$ | $N \times 64 \times 160 \times 160$ | conv $(3 \times 3$, 64, stride 1) + BN + ReLU |
| $SE_1$ | $N \times 64 \times 160 \times 160$ | avgpool + linear$(64, 4)$ + ReLU + linear$(4, 64)$ + Sigmoid |
| $SD_0$ | $N \times 64 \times 320 \times 320$ | conv $(3 \times 3$, 64, stride 1) + BN + ReLU |
| $SPA$ | $N \times 1 \times 320 \times 320$ | maxpool + avgpool + conv $(7 \times 7$, 1, padding 3) |
| $SE_0$ | $N \times 64 \times 320 \times 320$ | avgpool + linear$(64, 4)$ + ReLU + linear$(4, 64)$ + Sigmoid |
| $SD_{final}$ | $N \times 3 \times 320 \times 320$ | conv $(3 \times 3$, 3, stride 1, padding 1) |
| **Semantic Guided Matting Decoder** |
| $MD_3$ | $N \times 512 \times 10 \times 10$ | conv $(3 \times 3$, 512, dilation 2, padding 2) + BN + ReLU |
| $MD_4$ | $N \times 256 \times 10 \times 10$ | conv $(3 \times 3$, 512, dilation 2, padding 2) + BN + ReLU |
| $MU_3$ | $N \times 256 \times 20 \times 20$ | max-unpool $(2 \times 2$, stride 2) |
| $MD_3$ | $N \times 128 \times 20 \times 20$ | conv $(3 \times 3$, 256, stride 1) + BN + ReLU |
| $MU_3$ | $N \times 128 \times 40 \times 40$ | max-unpool $(2 \times 2$, stride 2) |
| $MD_2$ | $N \times 64 \times 40 \times 40$ | conv $(3 \times 3$, 128, stride 1) + BN + ReLU |
| $MU_2$ | $N \times 64 \times 80 \times 80$ | max-unpool $(2 \times 2$, stride 2) |
| $MD_1$ | $N \times 64 \times 80 \times 80$ | conv $(3 \times 3$, 64, stride 1) + BN + ReLU |
| $MU_1$ | $N \times 64 \times 160 \times 160$ | max-unpool $(2 \times 2$, stride 2) |
| $MD_0$ | $N \times 64 \times 320 \times 320$ | conv $(3 \times 3$, 64, stride 1) + BN + ReLU |
| $SPA$ | $N \times 64 \times 320 \times 320$ | SPA + SPA $\times MD_0$ |
| $MD_{final}$ | $N \times 1 \times 320 \times 320$ | conv $(3 \times 3$, 3, stride 1, padding 1) |
| **Matting Fusion** |
| $MF$ | $N \times 1 \times 320 \times 320$ | pixel-wise sum and multiplication. |

Table 2: The network structure of our matting model. The input size is $N \times 3 \times 320 \times 320$, where $N$ is the batch size.
Figure 4: Additional visual results on AIM-500.
Figure 5: Additional visual results on AIM-500.