Analysis on Image Matting Algorithms

Wuyue Lu, Kehui Li

Abstract

This paper compares two deep learning algorithms of automatic image matting without a manually created trimap. Most previous matting methods require user interaction to provide the trimap to recognize complex regions like hair. *Semantic Human Matting (SHM)* and *Glance and Focus Matting network (GFM)* are two algorithms that learn semantic information from a RGB image and produce fine alpha mattes without human interaction. We are going to reproduce the experimental results on a portrait image dataset (P3M-10k), and perform sensitivity analysis on how the training image is preprocessed would affect the performance of the SHM model. In addition, the two models would be tested on a new natural animal dataset (AM-2K).

1. Introduction

Image matting is a common technique for extracting foreground information, which is widely used in photo editing tools. Designing a matting method is not a minor task. Consider the compositing equation for an image I with foreground F, background B and alpha matte α:

$$I = \alpha F + (1 - \alpha)B, \quad \alpha \in [0, 1]$$

where for color images, there are 7 unknown variables (F, B and α) and three known variables (I). Thus, the matting problem is under constrained [1]. Therefore, many algorithms require users to manually create a trimap (a partition of foreground, background, and unknown regions), or some propose taking background as an extra constraint [2]. However, such constraints can be very time consuming and makes it difficult to be applied to large scale data.

Many automatic, trimap-free algorithms are present, and we are going to reproduce two of the methods – *Semantic Human Matting (SHM)* [1] and *Glance and Focus Matting network (GFM)* [3] – on human matting dataset (P3M-10k) [4], compare their behaviors and see how each performs a new natural animal image dataset (Animal Matting - 2k) [5].

Both SHM and GFM consist of a network to learn semantic information and a fusion module to generate the final alpha matte. The difference is GFM has an additional network to learn the high-level semantics besides the easy semantic parts (trimap). We will mainly analyze the performance and computation time of two methods around this difference.

2. Related Works

Image matting is a popular discussion topic these days. Except for the two CNN-based algorithms with a single RGB image input that we are discussing, there are also other approaches. Traditionally, except for the RGB image, the matting algorithm also requires a trimap as an input. These algorithms can be roughly categorized as sampling-based approaches and propagation-based approaches [2]. Both approaches use information from known regions to solve matts in the unknown region. However, the limitation is obvious when trimap cannot be carefully defined. Without trimap, this approach will not work well as it cannot find a clear correlation between the transition region, foreground and background when missing manual help.

The deep learning-based method is the trend. Recently, Chen et al. [1] proposed an end-to-end model that uses a segmentation network to produce a trimap then use it to aid the prediction of alpha matte. Zhang et al. [6] also proposed a similar segmentation-fusion structure. Both methods only take a single RGB image as input. In addition to one RGB image as input, there are some methods that use distinct and diverse input. For example, Sengupta et al. [2] in 2020 proposed a method that required an additional background image as input.

Overall, there are plenty of image matting deep learning-based algorithms sprouting up in those several years. However, additional analysis of those algorithms is still limited. Our goal is to perform analysis on two convolutional based algorithms.

3. Method / Algorithm

3.1 Semantic Human Matting

SHM aims to capture semantic information and matting details. It first uses T-net to extract semantic information and roughly classify the foreground. It outputs three classifications: the foreground, the background, and the unknown region. T-net is composed of encoder-decoder structure but overcomes some limitations of other similar encode-decode structure algorithms, for example, the U-Net. With the output from T-net and the original RGB image, M-net captures more detailed information and generates a new alpha matte. The M-Net is also a deep convolutional encoder-decoder network that helps to extract semantic information. With the foreground, the background, and unknown region channels produced by T-net and alpha matte by M-net, the fusion module produces the final alpha matte.

3.2 Glance and Focus Matting

The GFM model consists of a shared encoder and two separate decoders to learn both tasks in a collaborative manner for end-to-end image matting. The shared encoder uses a pretrained ResNet-34 as the backbone. The glace decoder aims to recognize definite background and foreground and leave the others as unknown regions. In this paper, we would use the GFM-TT model, where the glance decoder uses the trimap as the supervisory signal. Finally, the predictions from glance and focus decoders are merged with a collaborative matting module (CM).

4. Sensitivity Analysis on Image Preprocessing

Images come in different sizes, that is, their number of pixels varies. However, both SHM and GFM require the inputs to be a fixed size. Therefore, before getting into the training procedures, images need to be resized to a certain size. The size to downscale to is a hyperparameter of the algorithms, which is called patch_size in SHM and resize_size in GFM.

Besides, resizing and other operations like random cropping and rotations are often applied to the input image for data augmentation or avoiding overfitting. Both algorithms performed random cropping, resizing and random flipping and rotation to augment the samples.

4.1 Patch Size

Patch size can highly affect an algorithm's performance. The resize procedure involves downscaling the input with interpolation. In this process, pixels of important details might be lost. Hence, we hope to keep most of the details. However, when patch size is too large, it would be a large burden to the calculation that the runtime would increase largely. We will conduct sensitivity analysis on patch size and investigate the trade-off between performance and running time. The model is trained on 450 training images and tested on 50 testing images.

In SHM, input images with size ($a \times b \times 3$) will reduce to size (patch size $\times$ patch size $\times 3$). Fixing other hyperparameters, we trained the model with patch sizes of 16, 32, 64, 128 and 256.

Within 50 epochs, the training loss of the models with larger patch size decreases much faster as shown in figure 1. The fluctuation in the loss is mainly because of the random processing of data. Regarding the validation performance, the models with patch size of 128 and 256 have the best performance.

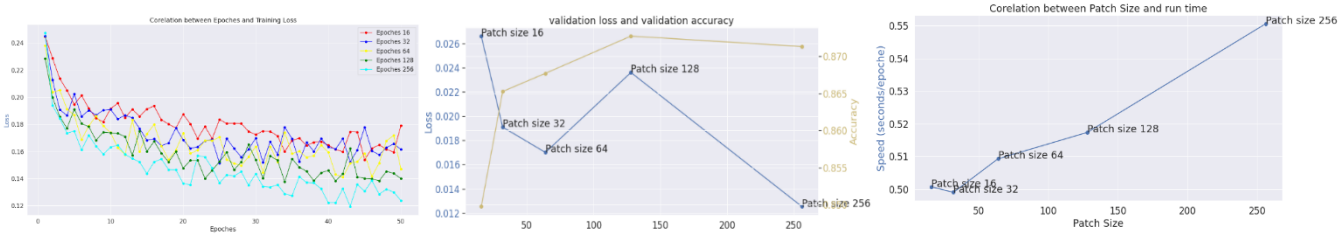

Figure 1. Plots of the training loss, validations loss and validation accuracy trained on different patch size.

In regard to the running time, there is only little difference between all patch sizes where the largest difference in speed is 0.05 minute per epoch. Since a GPU can afford more parallelism, they saturate at a large computation size. Before saturation, the time cost per epoch is about the same. However, the GPU saturated when we tried to run patch size of 512 on Colab. Since the patch size below 256 does not affect training time much, and performs much better on the validation set, we would use the model with patch size of 256 in further discussion.

For BFM, it takes 5 minutes to run over an epoch, which is very slow compared to 0.5 minutes per epoch in SHM, and the performance on small patch size is not good. We experimented with patch size of 32 and 64. Loss of both cases starts from 0.5. With patch size of 64, the loss drops to 0.261 in 50 epochs, while the loss only drops to 0.406 for patch size of 32. Same with SHM, the difference in running time is small. Hence, we choose the model with patch size of 64.

4.2 Random Cropping, Rotation and Flipping

In the training phase, approximately half of images from datasets are randomly chosen to be cropped and flipped. After those operations, all of the images are uniformly resized to (patch size × patch size) by interpolation. The main reason for randomly transforming images is to avoid overfitting. The random part is the main reason for the fluctuation in the loss in figure 1. In this part, we investigate how not performing the random operations would affect the model performance. The model is still trained on 450 training images with patch size of 256.

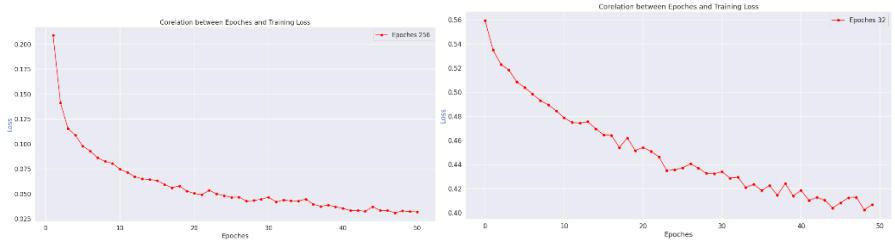

Figure 2. Plot of the training loss of SHM and GFM without random operations.

As expected, the training loss decreases much more smoothly and faster than before. The loss reaches 0.03 within 50 epochs while it only drops to 0.012 with random operations.

Since the model is only trained for 50 epochs, overfitting to the data is small. As there is no problem of overfitting, there is no need to create noise in training data. For small patch size, adding noise makes it more difficult to capture semantic information. Thus, performing random crop and rotations to avoid overfitting does not work well.

Input images / Performance	Training Loss	Validation accuracy
P3M with random transformations	0.147	0.871
P3M without random transformations	0.012	0.962

Table 1. Model performance when input images are processed differently.

5. Extended Work

For the extension part, we are going to investigate how the models generalize to other datasets. The results above shows that the model performs well on portrait images. We still question its performance on other types of images. Using the model trained with patch size 256, we tested 200 images from the AM dataset, which are images of animals.

Model and dataset / performance	Accuracy
SHM with random test on P3M	0.871
SHM with random test on AM	0.749
SHM without random test on P3M	0.962
SHM without random test on AM	0.811

Table 2. SHM performance on portrait and animal test sets.

Using the SHM model with random transformation, the accuracy of predicting 200 images from AM datasets is 0.749 which is much lower than 0.871 on P3M. A similar result appears with the model without random transformation. The result shows that although both P3M and AM has a single object as foreground, there is still domain gap between them.

6. Comparison

Li et al. [3] compared the inference time between several matting algorithms including SHM and GFM-TT(r). It was stated that the inference time of GFM-TT(r) tested on an image that is resized to 800 × 800 is much faster than that of SHM. However, when we trained on a training set of size 450 and resized to 32 × 32 with other hyperparameters same as trained in SHM, the running time per epoch was much slower than SHM and the final performance of the model was bad. Figure. 4 below shows that the training loss decreased slightly even for the 50th epochs. The validation accuracy is 0.4 which is quite low compared to SHM. It might perform better when it is trained on a larger dataset and more epochs.

SHM has a good performance even when the model was trained only for 50 epochs on 450 input images. The posture can be correctly predicted, and the position and contours are clear and accurate. The rough segmentation is fine, however, it fails to predict the detailed information around the hair.

(a) (b) (c)

Figure 5. Visual results of SHM. (a) an input image, (b) true alpha matt, (c) predicted alpha matt.

7. Conclusion

In this report, we focused on two deep learning image matting algorithms and performed a sensitivity test on how image preprocessing would affect the performance of the models. The better settings are then chosen to test on an animal image dataset. For possible applications, as image matting algorithms extract semantic information quite well, we expected it to help the classification network.

Reference

[1] Chen, Quan, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. "Semantic human matting." In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 618-626. 2018.

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

[3] Li, Jizhizi, Jing Zhang, Stephen J. Maybank, and Dacheng Tao. "Bridging composite and real: towards end-to-end deep image matting." *International Journal of Computer Vision* (2022): 1-21.

[4] Li, Jizhizi, Sihan Ma, Jing Zhang, and Dacheng Tao. "Privacy-preserving portrait matting." In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3501-3509. 2021.

[5] Li, Jizhizi, Jing Zhang, Stephen J. Maybank, and Dacheng Tao. "Bridging composite and real: towards end-to-end deep image matting." *International Journal of Computer Vision* (2022): 1-21.

[6] Li J, Ma S, Zhang J, et al. Privacy-preserving portrait matting[C]//Proceedings of the 29th ACM International Conference on Multimedia. 2021: 3501-3509.

[7] Zhang, Yunke, Lixue Gong, Lubin Fan, Peiran Ren, Qixing Huang, Hujun Bao, and Weiwei Xu. "A late fusion cnn for digital matting." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7469-7478. 2019.

Contribution

Both team member contributed equally to reproducing two models, experimenting on sensitivity tests and writing the report.