Treating Pseudo-labels Generation as Image Matting for Weakly Supervised Semantic Segmentation

Changwei Wang[1,3,*], Rongtao Xu[1,3,*], Shibiao Xu[2,†], Weiliang Meng[1,3,†], Xiaopeng Zhang[1,3]
[1]The State Key Laboratory of Multimodal Artificial Intelligence Systems,
Institute of Automation, Chinese Academy of Sciences, China
[2]School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences, China
shibiaoxu@bupt.edu.cn, weiliang.meng@ia.ac.cn

Appendix

1. More Ablation Study

1.1. Mat-Label *w/* Different Image Matting Methods

Table 1. Mat-Label *w/* different matting algorithms on PASCAL VOC 2012 train set

PASCAL VOC 2012 Train set	
Configuration	mIoU (%)
D2CAM (Baseline)	58.0
+ Closed Form Matting [6]	61.4
+ LBDM [11]	61.8
+ RW [4]	61.3
+ MatteFormer [10]	61.9
+ KNN Matting [2]	**62.3**

Table 1 shows a comparison of our Mat-Label using different image matting algorithms [6, 11, 4, 2, 10] for initializing the labels on PASCAL VOC 2012 train set. We implement these methods based on PyMatting[1] and MatteFormer[2]. All variants use exactly the same inputs. Closed Form Matting [6], LBDM [11], RW [4] are hand-crafted methods while KNN Matting [2] and MatteFormer [10] are learning-based methods. It can be seen that KNN Matting [2] achieves the highest performance improvement (+4.3%). It is worth noting that the performance of our Mat-Label pipeline on WSSS task can be further improved with the advancement of image matting technology.

Table 2. Mat-Label *w/* different CAMs on PASCAL VOC 2012 train set with mIoU (%).

PASCAL VOC 2012 Train set			
Pure CAMs	mIoU	Mat-Label *w/* CAMs	mIoU
CAM [12]	48.0	CAM [12]	$50.2_{+2.2}$
ReCAM [3]	54.8	ReCAM [3]	$56.3_{+1.5}$
D2CAM (ours)	58.0	D2CAM (ours)	$62.3_{+4.3}$

1.2. Mat-Label as a Refinement of CAMs

Table 2 reports the results of our Mat-Label pipeline as a refinement of existing CAMs. CAM [12] is the classical class activation map generation method, while ReCAM [3] is the recent advanced class activation map generation method. It can be seen that our Mat-Label pipeline can be used as a refinement to bring consistency improvements to existing CAMs. Our specially designed D2CAM for Mat-Label can enjoy a higher performance boost due to its ability to produce good foreground-background division. We follow the suggestion to set up Table 3 to explore the impact of each step on the final mask performance. The results show that applying only KNN matting (Non-deep learning) brings less improvement than IRN (deep learning based), but applying image matting is crucial for the final high performance (+1.5%) because it provides better initialization for IRN.

2. Hyperparameters Exploration

2.1. Exploration of the Loss Functions' Weights

For methods that use multiple loss function terms for joint optimization, setting weights for the loss functions

* Equal Contribution and † Corresponding Authors.
[1]https://github.com/pymatting/pymatting
[2]https://github.com/webtoon/matteformer

Table 3. Impact of different refine technologies on overall pipeline performance.

+ CRF [5]	+ IRN [1]	+ Mat-Label (image matting)	mIoU (%)
✔			63.9
✔	✔		71.4
✔		✔	65.8
✔	✔	✔	**72.9**

is unavoidable. As shown in Table 4, (γ_1,γ_2,γ_3) are the weights of $\mathcal{L}_{d2}$,$\mathcal{L}_{ma}$,$\mathcal{L}_{oe}$, respectively. We determine the loss function weights by sequentially imposing. Specifically, the weight of classification loss $\mathcal{L}_{cls}$ is fixed to 1. The optimal weights (γ_1) of $\mathcal{L}_{d2}$ is searched first, then the optimal weight (γ_2) of $\mathcal{L}_{ma}$ is searched, and finally the weight (γ_3) of $\mathcal{L}_{oe}$ is searched. Notice that we fix the optimal parameters of the previous phase in each search phase. In finding the optimal weights, we use a grid search [7] (*i.e.*, adjusting the weights by a fixed distance) to adjust the weights. Finally, we choose the optimal **(0.50,2.00,0.50)** as the default setting for the loss functions weights.

Table 4. Exploration of loss function weights on PASCAL VOC 2012 train set with mIoU (%).

(γ_1,γ_2,γ_3)	mIoU (%)	(γ_1,γ_2,γ_3)	mIoU (%)
(0.25, 0.00, 0.00)	50.1	(0.75, 0.00, 0.00)	52.8
(0.50, 0.00, 0.00)	**53.7**	(1.00, 0.00, 0.00)	49.6
(0.50, 0.25, 0.00)	53.9	(0.50, 1.50, 0.00)	55.8
(0.50, 0.50, 0.00)	54.8	(0.50, 1.75, 0.00)	56.7
(0.50, 0.75, 0.00)	54.3	**(0.50, 2.00, 0.00)**	**56.8**
(0.50, 1.00, 0.00)	55.6	(0.25, 2.25, 0.00)	56.2
(0.50, 1.25, 0.00)	55.9	(0.25, 2.50, 0.00)	55.3
(0.50, 2.00, 0.25)	57.4	(0.50, 2.00, 0.75)	57.5
(0.50, 2.00, 0.50)	**58.0**	(0.50, 2.00, 1.00)	57.2

2.2. The Impact of Margin Threshold m in Eq. (7)

The foreground ($\mathcal{M}_{fg}$) contains too much area also makes $\mathcal{L}_{d2}$ drop low, so it is necessary to impose $\mathcal{L}_{ma}$ as a constraint. To prevent the optimization from dropping into a local optimum at the beginning of the optimization (only $\mathcal{L}_{ma}$ is optimized), $\mathcal{L}_{ma}$ sets margin m, and resulting in these two losses can be steadily decreased together with mutual constraints at the later optimization stage (after $\mathcal{L}_{d2}$ also start optimization). Figure 1 shows the effect of different margin thresholds m of $\mathcal{L}_{ma}$ on the performance of D2CAM. Specifically, we conduct experiments on the PASCAL VOC 2012 train set using D2CAM with exactly the same configuration except for m. The yellow five-pointed star represents the highest mIoU. We can see that D2CAM

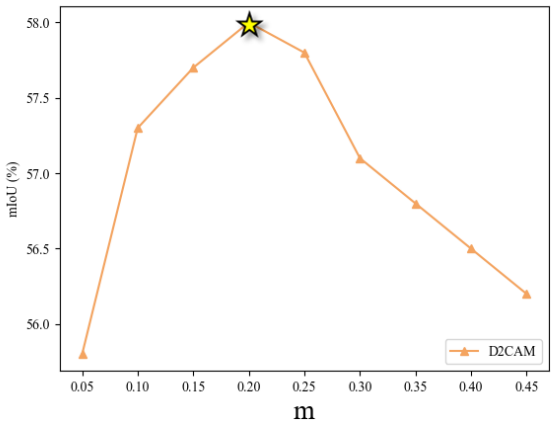

Figure 1. The impact of margin threshold m in margin-wise area loss.

achieves the best performance when m is set to 0.2. One possible explanation is that the average value of the foreground pixels as a percentage of the whole image pixels may be around 20%.

2.3. The Impact of Weight λ in Eq. (7)

Table 5. The Impact of weight λ in margin-wise area loss.

PASCAL VOC 2012 train set	
λ	mIoU (%)
0.000	57.4
0.025	57.7
0.050	**58.0**
0.075	57.6
0.100	56.9

Table 5 shows the effect of different λ of $\mathcal{L}_{ma}$ on the performance of D2CAM. The default optimal configuration is used for all configurations except λ. The experimental results show that the performance is higher when $\lambda = 0.05$ (in $\mathcal{L}_{ma}$) than when $\lambda = 0$. So the result shows that setting λ to provide a continuous area constraint for optimization is necessary. Since m (in $\mathcal{L}_{ma}$ and be explored in Figure 1) is a hard threshold, the area occupied by the object is less than m in some samples and therefore can benefit from continued optimization.

2.4. Exploration of Thresholds ε_{fg} and ε_{bg}

In fact, our Mat-Label pipeline has only two additional hyperparameters, *i.e.* ε_{fg} and ε_{bg}, than the original pure CAM [12] solution. Due to the clear distinction between foreground and background regions in D2CAM, this causes it to be robust to the selection of ε_{fg} and ε_{bg}. Specifically, Figure 3 reports our exploration of thresholds selec-

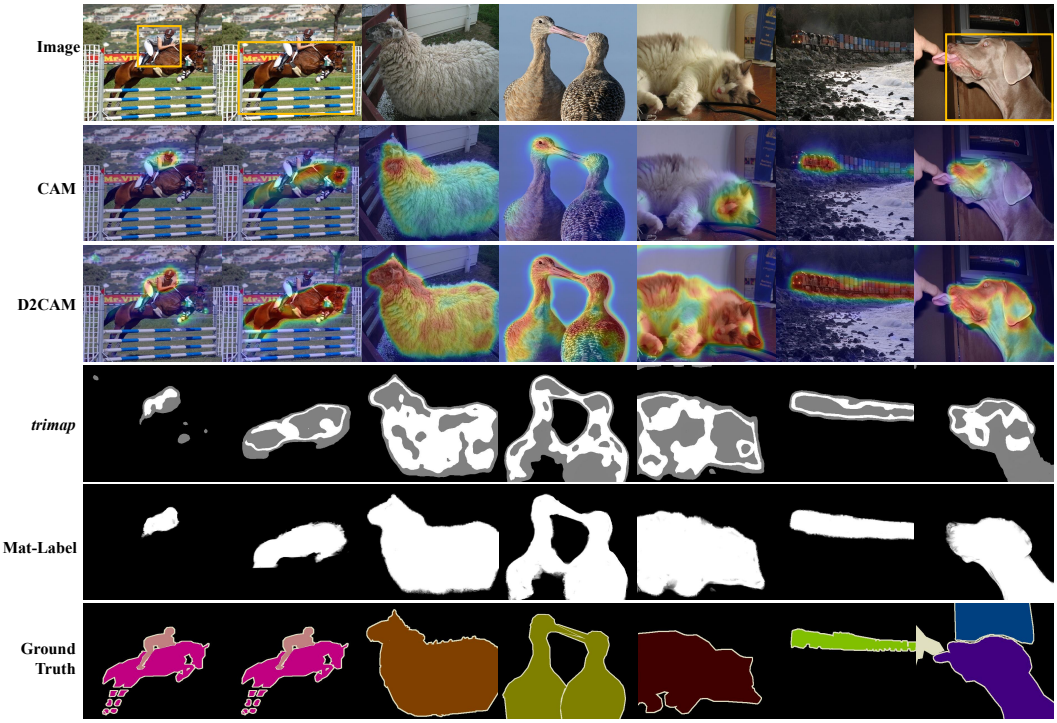

Figure 2. Some failure cases about our Mat-Label on PASCAL VOC 2012 dataset.

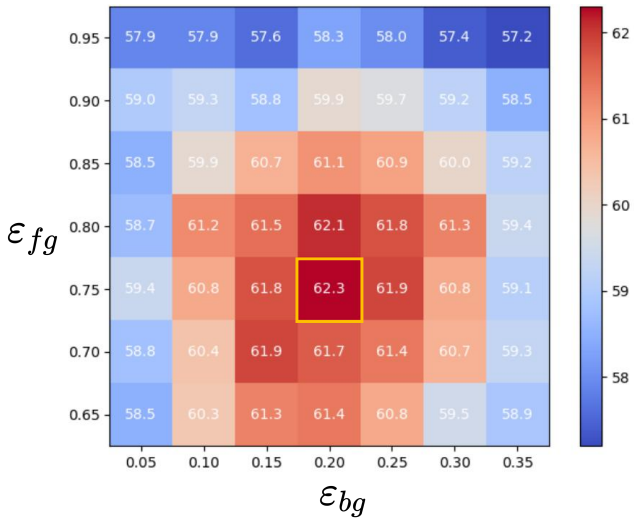

Figure 3. Exploration of generating *trimap* thresholds on PASCAL VOC 2012 train set with mIoU (%). The x-axis is ε_{bg} and the y-axis is ε_{fg}.

tion. The x-axis is ε_{bg} and the y-axis is ε_{fg}. It can be observed that similar average performance can be obtained for $\varepsilon_{bg} \in (0.15, 0.25)$ and $\varepsilon_{fg} \in (0.7, 0.8)$, and we use the optimal one ($\varepsilon_{bg} = 0.2$ and $\varepsilon_{fg} = 0.75$) as the default setting.

3. Limitations and Future Work

3.1. Limitations

The limitation of our Mat-Label is the need to rely on high quality *trimap*. Figure 2 shows some examples of failure cases regarding our Mat-Label. Specifically, (a), (b) and (c) exhibit incomplete foreground regions while (d) and (e) exhibit over-activation of background regions. For example, although our D2CAM in (a) overcomes the challenge of occlusion well (fence), there are missing foreground regions (red bounding box). Similarly, the D2CAM does not get activated in (b) and (c) for the complete foreground region. This trend leads to the acquisition of low quality *trimap*, while common image matting algorithms are unable to accurately estimate α under such conditions. In the *trimaps* of (d) and (e), some background regions are divided into the foreground, which results in the image matting algorithm not being able to obtain an accurate foreground background distinction using the wrong prior information. However, we note that even in the wrong example, our Mat-Label is still more accurate than the original class activation map, which means that Mat-Label does not degrade the quality of the generated pseudo masks (*e.g.* D2CAM *vs* Mat-Label in (e)).

3.2. Future Work

For future work, we can improve the following two aspects to obtain a higher quality *trimap*. **On the one hand**,

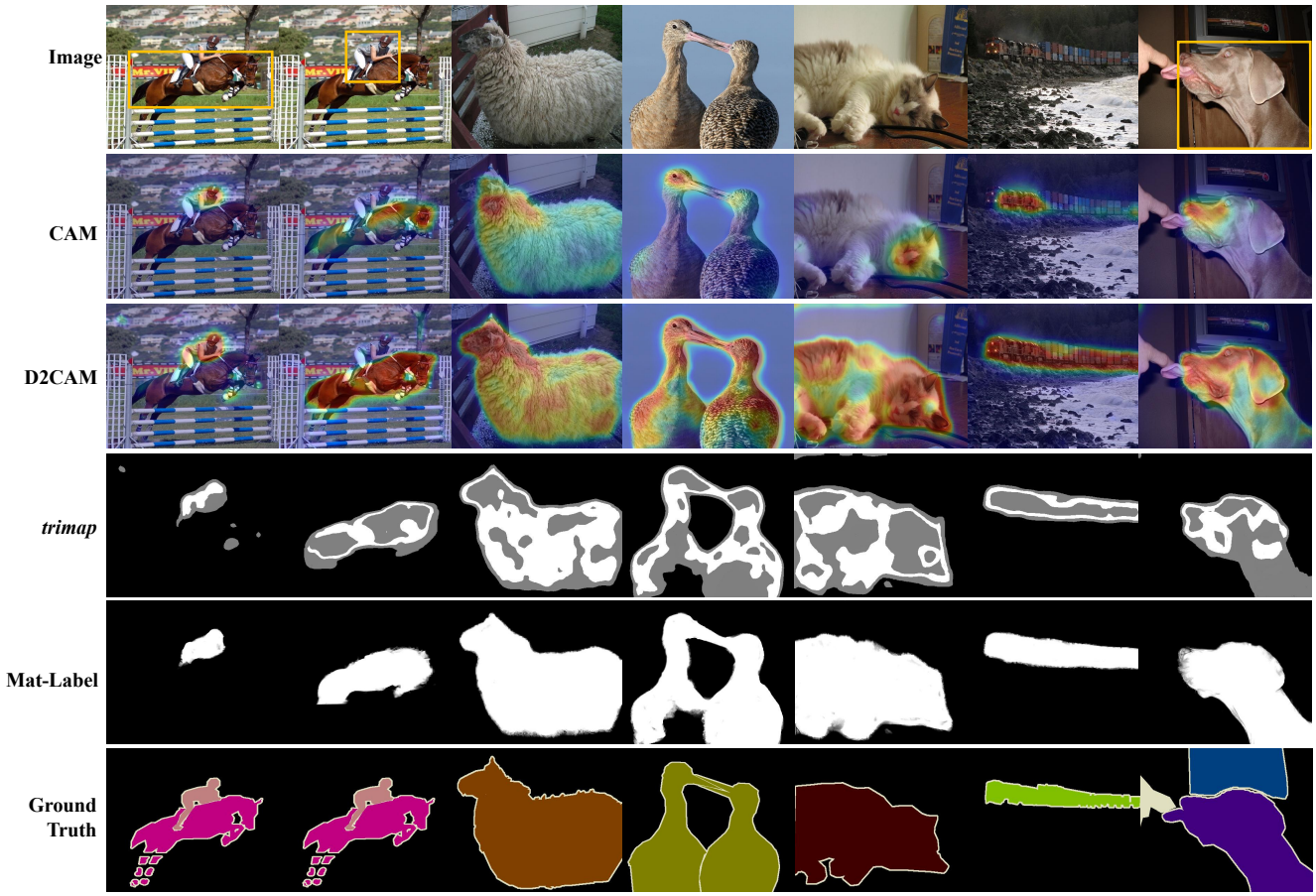

Figure 4. Some examples of Mat-Label generation visualizations.

we can improve the accuracy of foreground-background division of the class activation map to obtain a higher quality *trimap*. For example, the resolution of the class activation map can be increased, and the edge information can be introduced to reduce under-activation and over-activation. **On the other hand**, our approach of using fixed thresholds to generate *trimap* can be optimized because the same thresholds are not perfectly applicable to different images. We can model the unknown region explicitly by uncertainty estimation and explore the threshold-free *trimap* generation approach.

4. More Visualizations

More visualizations of our Mat-Label generation are shown in Fig. 4. It can be seen that our D2CAM has a more complete and accurate foreground-background division than CAM. Our Mat-Label derived pseudo mask is close to the ground truth mask.

Table 6. Comparisons on the Running Speed. The speed is calculated from a single image (512×512) at the same settings (a single NVIDIA RTX A6000 GPU).

Methds	CAM [12]	ReCAM [3]	D2CAM	Mat-Label
Time (s)	**0.36**	0.71	0.39	1.42
mIoU (%)	48.0	54.8	58.0	**62.3**

5. Operational Efficiency Analysis

Table 6 shows a comparison of the speed of running single image inference. The original CAM extracts the class activation map by applying the FC layer (1×1 Conv layer) to the last feature map ($H \times W \times 2048$), as shown in main paper Figure 1 (a). In contrast, our D2CAM extracts the class activation map by applying the 3×3 Conv layer to the penultimate feature map ($H \times W \times 1024$). In addition, D2CAM does not need to go through the final convolution layer in the inference stage. In summary, our D2CAM is comparable to the original CAM on running speed, as shown in Table 6. Mat-Label needs to perform image mat-

ting operations and therefore requires additional runtime consumption. However, compared with the performance improvement, this extra calculation is worthy. In addition, the image matting algorithm we use runs on the CPU is a major influence on the speed consumption. The operational efficiency can be improved in future work by introducing efficient real-time deep learning based image matting algorithms [8, 9].

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2209–2218, 2019.

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

[3] Zhaozheng Chen, Tan Wang, Xiongwei Wu, Xian-Sheng Hua, Hanwang Zhang, and Qianru Sun. Class re-activation maps for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 969–978, 2022.

[4] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive alpha-matting. In *Proceedings of VIIP*, volume 2005, pages 423–429, 2005.

[5] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.

[6] 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.

[7] Petro Liashchynskyi and Pavlo Liashchynskyi. Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059*, 2019.

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

[9] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 238–247, 2022.

[10] GyuTae Park, SungJoon Son, JaeYoung Yoo, SeHo Kim, and Nojun Kwak. Matteformer: Transformer-based image matting via prior-tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11696–11706, 2022.

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

[12] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.