AI6126 Project 1: Semantic Segmentation

CelebAMask Face Parsing

Zhou RuoHeng, G2303498K, MSAI, zh0121ng@e.ntu.edu.sg

I. Dataset Introduction

This project will use the data from the CelebAMask-HQ Dataset[1], of which only 1,000 annotated images and unannotated 100 images are selected. For the 1,000 annotated images, 800 were used for training and 200 for validation. The remaining unannotated 100 images are targeted for inference. Here are some exemplar annotated images in our dataset shown in Figure 1.

The masks of CelebAMask-HQ were manually-annotated with the size of 512 x 512 and 19 classes. According to tool-based statistics manually, all classes and corresponding palette are as follows:

1. classes = ('Background', 'Skin', 'Nose', 'Eye glasses', 'Left eye', 'Right eye', 'Left brow', 'Right brow', 'Left ear', 'Right ear', 'Mouth', 'Upper lip', 'Lower lip', 'Hair', 'Hat', 'Ear ring', 'Necklace', 'Neck', 'Cloth')
2. palette = [[0, 0, 0], [204, 0, 0], [76, 153, 0], [204, 204, 0], [51, 51, 255], [204, 0, 204], [0, 255, 255], [255, 204, 204], [102, 51, 0], [255, 0, 0], [102, 204, 0], [255, 255, 0], [0, 0, 153], [0, 0, 204], [255, 51, 153], [0, 204, 204], [0, 51, 0], [255, 153, 51], [0, 204, 0]]

Figure 1: some examples of annotated images

II. Data Preprocessing and Augmentation

The whole pipeline is implemented by *mmsegmentation*[2] framework. Majority of following Augmentation methods are directly supported by *mmsegmentation* except for ***Random Flip with Label Swap***.

We only train on the very small portion, 1000 images and masks, of CelebAMask-HQ Dataset. Therefore, given the limited size of training dataset, data augmentation becomes particularly crucial to introduce greater diversity in input distribution, thereby enhancing the model's robustness and generalization performances.

To evaluate the performance of various Augmentation methods, ***Fast-SCNN*[3] is selected as the baseline model with a limited number of iterations and 8 as the batch size.** The interval of validation checkpoints is 500 iterations. The diverse methods or combinations of methods and their corresponding results are shown in the following Table 1.

Normalization*, *Random Rotate*, *Random Resize* and *Random Crop. These are common data augmentation strategies. Normalization standardizes pixel distributions to improve training stability. Random rotation simulates facial orientation changes, enhancing robustness. Random resizing enables adaptation to varying face scales, improving generalization. Random cropping emphasizes local features and reduces spatial dependence. The performance improvements can be found at No.5 in Table 1 compared to Baseline(No.1).

Table 1

No.	**Data Augmentation Methods**(#iteration)	mFscore*	mIoU*	Overfitting**
1	Baseline***(40k)	42.39	27.63	No
2	Baseline***(80k)	59.29	40.20	No
3	**Random Flip With Label Swap(80k)**	**68.8625**	**52.35**	No
4	Random Flip(80K)	62.43	45.36	No
5	**Normalization + Random Rotation + Resize + Crop(40K)**	**72.52**	**52.38**	No
6	**Photo Metric Distortion(40K)**	**48.44**	**32.16**	No

* An average was computed over the final four validation checkpoints.
** The "Overfitting" here means whether the metrics are still in the trend of improvement. If still improves, No "Overfitting".
*** Baseline means without any Agumentation methods.

1. According to the calculations, the mean and standard deviation of the entire training and validation set were $[130.41, 104.75, 91.3]$ and $[68.62, 61.36, 59.27]$, respectively, in RGB order.
2. Random Rotate was configured with a range of $\pm 25^\circ$ and $\text{prob} = 0.5$ to enhance the model's adaptability to head tilts and selfie angles.
3. Random Resize employed a $\text{scale} = (640, 640)$ $\text{ratio\_range} = (0.8, 1.1)$ to slightly enlarge images, which helps simulate close-up facial shots and improves scale invariance.
4. RandomCrop was set with $\text{cat\_max\_ratio} = 0.75$, preserving the main foreground while directing the model's attention to local regions, thereby enhancing its ability to distinguish fine-grained features such as eyes and lips.

Photo Metric Distortion, *aka Color Jitter*. Photo Metric Distortion perturbs brightness, contrast, saturation, and hue to simulate real-world variations in lighting and color, thereby enhancing the model's robustness to changes in visual conditions. For tasks with limited data, such as face parsing, it helps mitigate overfitting and improves the model's generalization ability. The default settings of PhotoMetricDistortion in MMSegmentation are relatively aggressive, often causing unnatural skin tones and detail distortion. In small, color-unified face parsing datasets, this may lead to overfitting or hinder learning of fine structures. Therefore, all perturbation ranges in PhotoMetricDistortion were reduced to half of their default values ($\text{brightness\_delta} = 16, \text{contrast\_range} = (0.75, 1.25), \text{saturation\_range} = (0.75, 1.25), \text{hue\_delta} = 9$). The performance improvements can be found at No.6 in Table 1 compared to Baseline(No.1).

Random Flip with Label Swap. In face parsing, where facial structures are highly symmetric, *RandomFlipWithLabelSwap* outperforms standard *RandomFlip*. While standard flipping may cause semantic errors (e.g., a flipped "left eye" still labeled as "left eye"), RandomFlipWithLabelSwap swaps labels of symmetric classes during flipping, preserving semantic consistency and improving the learning of symmetric features. The advantages of Label Swap can be found in comparison between Nos.3 and Nos.4 in Table 1, which both improve significantly more than Baseline(No.2). *RandomFlipWithLabelSwap* is implemented as a subclass of RandomFlip, with additional functionality to swap label values for certain classes during flipping by Label Swap map {L eye(4):R eye(5), L brow(6):R brow(7), L ear(8):R ear(9)}.

Based on the results of the metrics of these methods, we decide to adopt the final combination of augmentation methods in the following order:

1. Random Flip the images and masks with swaps for some special labels.
2. Normalize images with mean and std of training + validation datasets.
3. Random Rotate images within range $(0^\circ, 25^\circ)$ and $\text{Prob} = 0.5$
4. Random Resize images within scale ration range $(1, 1.2)$. Then Crop them back into size $(512, 512)$ with $\text{cat\_max\_ratio} = 0.75$.
5. PhotoMetricDistortion, brightness = 16, hue = 9, contrast and saturation range = (0.75, 1.25), Prob = 0.5, which is half of *mmsegmentation* default setting value.

III. Loss Functions and Models

III.1 Loss Functions

For loss function, we attempt to test from CrossEntropy Loss(CE), Weighted CE + Lovász Loss, weighted CE+Dice Loss and weighted CE + Lovász + Dice Loss. **As for the choice of the baseline model, *Fast-SCNN* was retained, with the addition of the only one of augmentation techniques, Normalization, described in the previous section.** The results are shown in Table 2

Table 2

No.	Loss Function****(#iteration)(ratio)	Decode:Auxiliaries	mFscore*	mIoU*	Overfitting**
1	Baseline***(40k)(None)	None	72.93	52.58	No
2	Weighted CE+Lovász(40k)(1:2)	1:0.8	77.26	65.27	Yes
3	Weighted CE+Lovász(40k)(1:2)	1:0.4	78	**65.94**	Yes
4	Weighted CE+Dice(40k)(1:2)	1:0.4	77.7	62.9	No
3	**Weighted CE+Lovász+Dice**(40k)(0.5:0.3:1)	1:0.4	**79.43**	63.92	**No**

* An average was computed over the final four validation checkpoints.
** The "Overfitting" here means whether the metrics are still in the trend of improvement. If still improve, No "overfitting".
*** Baseline here means the model only uses CE loss.
**** For simplicity and consistency, the same combination of loss functions was applied to both the decode head and all auxiliary heads.

Cross Entropy Loss(CE) measures pixel-wise classification error and is well-suited for tasks with clear global structure, though it is less sensitive to small regions. ***Dice Loss***[4] emphasizes the overlap between predicted and ground truth regions, making it effective for scenarios with small foreground areas. ***Lovász Loss***[5] provides a differentiable approximation of IoU, enhancing overall segmentation quality, particularly for boundaries and small objects.

Compared to respectively apply above mentioned loss functions, a weighted combination of CE, Dice and Lovász losses leverages the strengths of each, enabling more robust optimization in segmentation tasks.

1. *CE + Dice* is suitable for tasks with small foregrounds; the former provides stable classification supervision, while the latter enhances overlap awareness, though Dice may be unstable in early training.
2. *CE + Lovász* improves overall segmentation and boundary quality, making it effective for IoU-based evaluation, though Lovász converges slowly.
3. *CE + Dice + Lovász* optimizes classification, region, and boundary performance simultaneously, offering the most comprehensive solution, but requires careful weighting to ensure stable convergence.

The different ratios of loss weights between loss functions form different combinations. Besides, there also exists loss weight ratios between decode head and auxiliary heads. For example, *Fast-SCNN*, our baseline model, normally has two auxiliary head and every auxiliary head can be set up by their own combinations of loss functions. After a series of experiments on different combinations, several efficient and well-performed combinations are listed in Table 2.

As results shown in Table 2, weighted combinations of loss function all achieve remarkable improvements compared to Baseline. Additionally, it can be concluded that the *CE + Lovász* yielded promising results but tended to suffer from overfitting or performance degradation. Compared to *CE + Dice* and *CE + Lovász*, ***CE + Lovász + Dice*** **achieved better results and there remained untapped potential for performance improvements respectively.**

III.2 Models

Based on aforementioned experiments, in this section we are going to apply the complete augmentation strategy. Besides, ***CE + Lovász + Dice*** **was selected as final combination of loss functions because of its better performance.**

After investigation and selection, three model architectures were ultimately chosen: *Fast-SCNN*, *MobileNetV3 LRASPP*[6] and *BiSeNetV2*[7].

1. *Fast-SCNN* is a lightweight real-time segmentation network that fuses high- and low-resolution features, suitable for edge devices and basic face parsing tasks.
2. *MobileNetV3 LRASPP* combines efficient MobileNetV3 with a lightweight ASPP module, balancing speed and accuracy for facial structure segmentation in resource-limited settings.
3. *BiSeNetV2* uses parallel Detail and Semantic branches with a BGA module for fusion, enabling efficient and structure-aware segmentation, especially effective for boundaries and local features in face parsing.

Given the requirement of limited number of trainable parameters, 1,821,085, we experiment further on *Fast-SCNN* and *MoblienetV3 LRASPP*, of which parameters are both under the restriction. As for *Bisenetv2*, Efforts were made to reduce the number of parameters while maintaining performance, enabling the model to meet constraints. The modifications are shown in the following Table 3. **For simplicity and convenience, the modified *BisenetV2* will be denoted as *BisenentV2 XS*.**

Table 3

Component	Attribute	Original Value	Current Value
backbone	detail_channels	(64, 64, 128)	(32, 64, 96)
	semantic_channels	(16, 32, 64, 128)	(16, 32, 64, 96)
	bga_channels	128	96
decode head	in_channels	128	96
	channels	1024	512
auxiliary heads	head[1].channels	64	32
	head[2].channels	256	64
	head[3].in_channels	128	96
	head[3].channels	1024	128

Finally, in this section the complete training pipeline consisting will be adopted to train these three models respectively. The batch size remains 8 and maximum iteration of training is set 24k. The best validation metrics results for respective models are shown in following Table 4.

Table 4

No.	**Model**)(best iteration*)	mFscore**	mIoU**	#params
1	*Fast-SCNN*(10,500)	83.17	68.78	≈ 1.4M
2	*MobileNetV3 LRASPP*(13,000)	84.22	69.73	≈ 1.14M
3	***Bisenetv2 XS*(22,500)**	**85.6**	**71.61**	≈ 1.792M

* the iteration with the highest mFscore.
** The performance metrics corresponding to the iteration with the highest mFscore.

In conclusion, as shown in the No.3 entry in Table 4, BiSeNetV2-XS achieved the best overall performance.

IV. Results and Summary

In summary, the best configuration of this face parsing task is as follows:

1. Data Augmentations consist of Normalization, Random Rotate, Random Resize, Random Crop, Photo Metric Distortion and Random Flip with Label Swap.
2. Weighted CE + Dice + Lovász is selected as the loss function.
3. $\text{batch\_size} = 8, \quad \text{max\_iteration} = 24,000, \quad \text{validation\_interval} = 500$
4. *Bisenetv2 XS* is selected as final model consisting of Bisenetv2 as backbone, 1 FCN as decode head and 4 auxiliary heads which are all FCN.

The following Figure 2 displays the variations of training loss from warming up phase to approximate convergence. In addition, the metric results of validation dataset in the lower part of Figure 2 improve quickly in the first 5,000 iterations and finally approach the stable convergence after 20,000 iterations. Figure 3 list 2 pairs of inference results from trained *BiseNetv2 XS* of which checkpoint was saved at 22,500-th iteration.

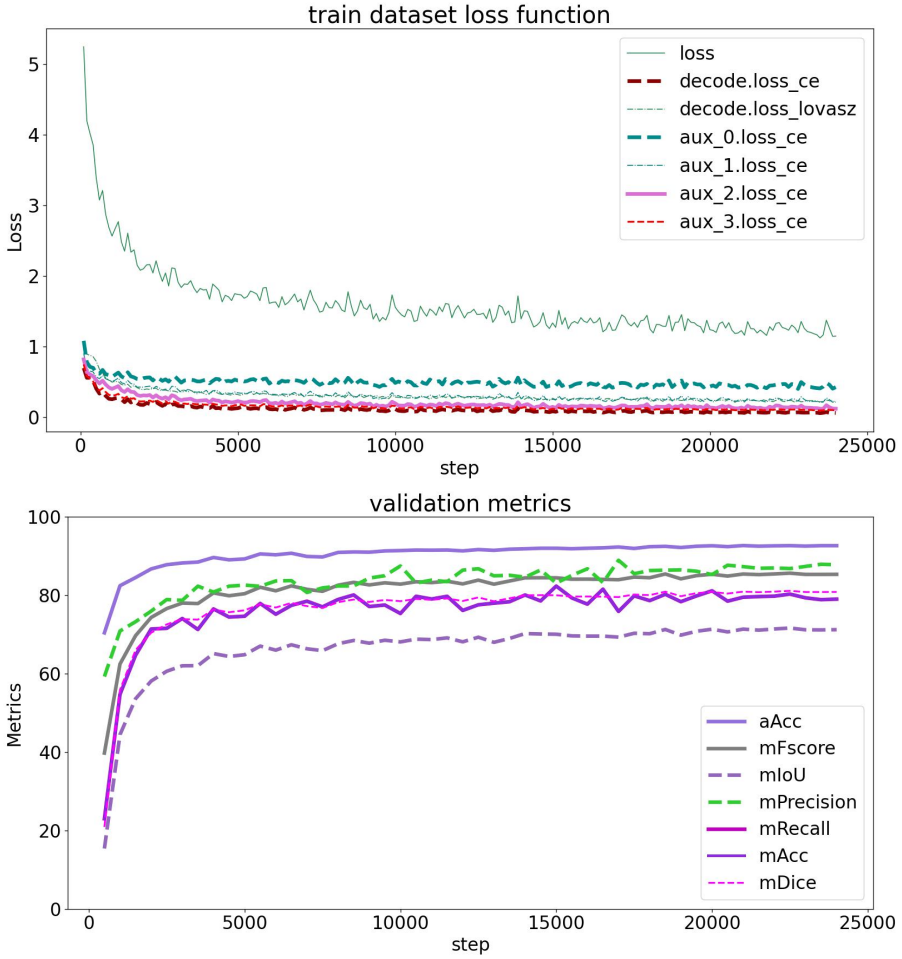

Figure 2

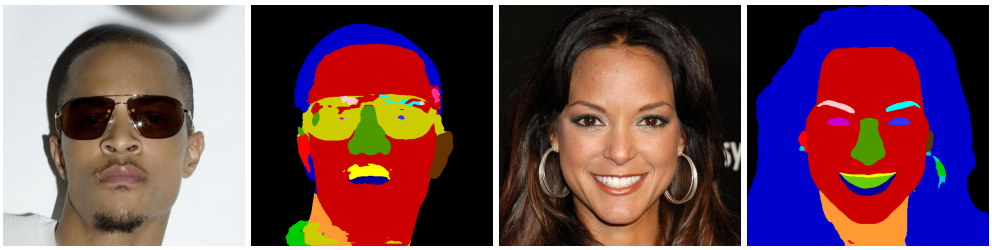

Figure 3: some inference results

Acknowledgements

We would like to thank**students G2405925A(LIHA0032@e.ntu.edu.sg) and G2405880B** for identifying an error in the dataset where left and right labels were mistakenly reversed. They also diligently compiled a list of the affected files, categorized the types of errors, and proposed possible solutions for correction.

References

[1] "Celebamask-hq dataset." https://mmlab.ie.cuhk.edu.hk/projects/CelebA/CelebAMask_HQ.html, 2020. Accessed: 2025-03-26.

[2] OpenMMLab, "MmSegmentation: Openmmlab semantic segmentation toolbox and benchmark." https://github.com/open-mmlab/mmsegmentation, 2020. Accessed: 2025-03-26.

[3] R. P. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," *arXiv preprint arXiv:1902.04502*, 2019.

[4] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, IEEE, 2016.

[5] M. Berman, A. Rannen Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4413–4421, 2018.

[6] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," in *The IEEE International Conference on Computer Vision (ICCV)*, pp. 1314–1324, October 2019.

[7] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *International Journal of Computer Vision*, pp. 1–18, 2021.