CelebAMask-HQ Face Parsing with MicroSegFormer

Anonymous CVPR submission

Paper ID *****

Abstract

Face parsing is a fundamental task in computer vision that aims to segment facial regions into semantic categories. This report presents a lightweight transformer-based approach for face parsing on the CelebAMask-HQ dataset. We propose MicroSegFormer, an efficient architecture inspired by SegFormer that achieves strong performance with only 1.72M parameters (94.6% of the 1.82M limit). Our model employs hierarchical transformer encoders with efficient self-attention mechanisms and a lightweight MLP decoder for multi-scale feature fusion. Through careful optimization of data augmentation, loss functions, and training strategies, we demonstrate competitive face parsing results while maintaining strict parameter constraints. The implementation includes comprehensive experiments on model architecture, regularization techniques, and augmentation strategies.

1. Introduction

Face parsing, the task of pixel-level semantic segmentation of facial images into different regions (e.g., eyes, nose, mouth, hair, skin), is crucial for numerous computer vision applications including face recognition, makeup transfer, and face editing. Despite significant advances in semantic segmentation using deep learning, face parsing remains challenging due to the need for fine-grained boundary detection and the large variation in facial appearance, pose, and occlusions.

Recent transformer-based architectures [1] have shown promising results for semantic segmentation tasks. However, these models often contain millions of parameters, making them computationally expensive and unsuitable for resource-constrained scenarios. This work addresses the challenge of developing an efficient face parsing model that achieves competitive performance while adhering to strict parameter constraints.

1.1. Problem Statement

The task is to perform pixel-wise semantic segmentation on the CelebAMask-HQ dataset, which contains facial images annotated with 19 different semantic classes including background, skin, nose, eyes, eyebrows, ears, mouth, lips, hair, hat, earrings, necklace, neck, and clothing. The primary challenges include:

- **Parameter Efficiency**: The model must contain fewer than 1,821,085 trainable parameters
- **Fine-grained Segmentation**: Accurate boundary delineation between adjacent facial regions
- **Class Imbalance**: Significant variation in the spatial extent of different facial components
- **Limited Training Data**: Only 1,000 training images with 100 validation samples

1.2. Our Approach

We propose MicroSegFormer, a lightweight transformer-based architecture that incorporates:

- Efficient hierarchical transformer encoder with spatial reduction attention
- Lightweight MLP decoder for multi-scale feature aggregation
- Combined loss function (Cross-Entropy + Dice Loss) for handling class imbalance
- Comprehensive data augmentation strategy including geometric and photometric transforms
- Advanced optimization techniques with cosine annealing and gradient clipping

Our model achieves 1,721,939 parameters (94.6% utilization), demonstrating effective use of the parameter budget while maintaining strong segmentation performance.

2. Method

2.1. Overview

MicroSegFormer is a hierarchical transformer-based architecture designed for efficient face parsing under strict parameter constraints (¡ 1.82M parameters). The model consists of two main components: (1) a four-stage hierarchical encoder that extracts multi-scale features through efficient self-attention, and (2) a lightweight MLP decoder that fuses these features for pixel-wise classification. Our final model contains 1,721,939 parameters, utilizing 94.6% of the allowed budget.

2.2. Hierarchical Transformer Encoder

Architecture Configuration: The encoder employs four stages with progressively increasing channel dimensions $C = [32, 64, 128, 192]$ and depths $D = [1, 2, 2, 2]$, processing input images at multiple resolutions. This hierarchical design captures both fine-grained local details and high-level semantic information.

2.2.1. Overlapping Patch Embedding

Unlike standard vision transformers that use non-overlapping patches, we employ overlapping patch embeddings to preserve local continuity—critical for accurate facial boundary segmentation.

For each stage i, the patch embedding is implemented as:

$$\text{PatchEmbed}_i(x) = \text{LayerNorm}(\text{Flatten}(\text{Conv2D}(x))) \tag{1}$$

Stage 1 uses a 7×7 convolution with stride 4 to downsample the input image ($3 \times 512 \times 512$) to $32 \times 128 \times 128$, reducing spatial resolution by 4× while preserving overlapping receptive fields (padding=3).

Stages 2-4 use 3×3 convolutions with stride 2, progressively downsampling features:

- Stage 2: $32 \times 128 \times 128 \rightarrow 64 \times 64 \times 64$
- Stage 3: $64 \times 64 \times 64 \rightarrow 128 \times 32 \times 32$
- Stage 4: $128 \times 32 \times 32 \rightarrow 192 \times 16 \times 16$

The overlapping design (patch_size $>$ stride) ensures that boundary information is not lost during downsampling, which is crucial for segmentation tasks.

2.2.2. Efficient Self-Attention Mechanism

Standard self-attention has $\mathcal{O}(N^2)$ complexity where N is the sequence length. For high-resolution images, this becomes prohibitively expensive. We adopt **spatial reduction (SR) attention** to reduce computational cost while maintaining representation power.

Query Computation (full resolution):

$$Q = \text{Linear}_q(X) \in \mathbb{R}^{N \times C} \tag{2}$$

Key-Value Computation (reduced resolution): For stages with SR ratio $R > 1$, we spatially reduce the feature map before computing K and V:

$$X' = \text{LayerNorm}(\text{Conv2D}(X; k = R, s = R)) \tag{3}$$

$$K, V = \text{Linear}_{kv}(X') \in \mathbb{R}^{N/R^2 \times C} \tag{4}$$

Our SR ratios are $[8, 4, 2, 1]$ for stages 1-4. For example, in Stage 1 with $N = 128 \times 128 = 16384$, the KV sequence length is reduced to $16384/64 = 256$, reducing attention complexity from $\mathcal{O}(16384^2)$ to $\mathcal{O}(16384 \times 256)$—a 64× reduction.

Attention Operation:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{5}$$

where $d_k = C/H$ is the dimension per head. We use single-head attention ($H = 1$) to minimize parameters while maintaining effectiveness.

2.2.3. Feed-Forward Network

Each transformer block contains a two-layer MLP with GELU activation:

$$\text{FFN}(x) = \text{Linear}(\text{GELU}(\text{Linear}(x))) \tag{6}$$

with expansion ratio 2 (hidden dimension = $2C$). This provides non-linear transformation capacity without excessive parameters.

2.2.4. Transformer Block

Each stage contains D_i transformer blocks with residual connections and pre-normalization:

$$x' = x + \text{Attention}(\text{LayerNorm}(x), H, W) \tag{7}$$

$$x'' = x' + \text{FFN}(\text{LayerNorm}(x')) \tag{8}$$

The pre-normalization design stabilizes training and enables deeper networks.

2.3. Lightweight MLP Decoder

The decoder fuses multi-scale features from all four encoder stages through a purely MLP-based approach, avoiding heavy convolutional layers.

2.3.1. Channel Unification

Features from different stages have different channel dimensions $[32, 64, 128, 192]$. We first project all features to a unified dimension (128) using linear layers:

$$\hat{f}_i = \text{Linear}(f_i) \in \mathbb{R}^{H_i \times W_i \times 128}, \quad i = 1, 2, 3, 4 \tag{9}$$

2.3.2. Spatial Alignment

All features are upsampled to the resolution of Stage 1 (128×128) using bilinear interpolation:

$$\tilde{f}_i = \text{Upsample}(\hat{f}_i, \text{size} = (128, 128)) \tag{10}$$

This alignment enables direct concatenation and fusion across scales.

2.3.3. Feature Fusion

The aligned features are concatenated and fused through a 2-layer MLP:

$$f_{\text{fused}} = \text{MLP}([\tilde{f}_1, \tilde{f}_2, \tilde{f}_3, \tilde{f}_4]) \tag{11}$$

where MLP consists of: Linear(512→128) + GELU + Linear(128→128).

2.3.4. Final Prediction

The fused features (128 × 128 × 128) are upsampled 4× to match input resolution (512×512), then passed through a 1×1 convolution for classification:

$$\text{Output} = \text{Conv2D}(\text{Upsample}(f_{\text{fused}}, \text{scale} = 4), k = 1) \tag{12}$$

producing the final prediction map of shape $(19 \times 512 \times 512)$.

2.4. Loss Function

We employ a weighted combination of Cross-Entropy and Dice Loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{CE} + 0.5 \cdot \mathcal{L}_{Dice} \tag{13}$$

Cross-Entropy Loss provides per-pixel, per-class supervision:

$$\mathcal{L}_{CE} = -\frac{1}{HW} \sum_{h,w} \sum_{c=1}^{19} y_{hwc} \log(\hat{y}_{hwc}) \tag{14}$$

Dice Loss addresses class imbalance and improves boundary quality:

$$\mathcal{L}_{Dice} = 1 - \frac{1}{19} \sum_{c=1}^{19} \frac{2 \sum_{h,w} y_{hwc} \hat{y}_{hwc} + \epsilon}{\sum_{h,w} y_{hwc} + \sum_{h,w} \hat{y}_{hwc} + \epsilon} \tag{15}$$

The Dice coefficient ranges [0,1], with 1 indicating perfect overlap. The weight 0.5 balances the two objectives.

2.5. Training Strategy

Optimizer: AdamW with learning rate $\eta = 1.5 \times 10^{-3}$ and weight decay $\lambda = 5 \times 10^{-4}$. Weight decay provides L2 regularization on all parameters.

Learning Rate Schedule: Cosine annealing with linear warmup:

$$\eta_t = \begin{cases} \frac{t}{T_w} \cdot \eta_{\max} & t \leq T_w \\ \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})(1 + \cos(\frac{t - T_w}{T - T_w}\pi)) & t > T_w \end{cases} \tag{16}$$

where $T_w = 5$ epochs for warmup, $T = 150$ total epochs, $\eta_{\min} = 0$.

Gradient Clipping: Maximum gradient norm of 1.0 prevents gradient explosion.

Early Stopping: Training stops if validation F-Score does not improve for 30 consecutive epochs.

Mixed Precision Training: FP16 computation with FP32 master weights reduces memory usage and accelerates training on modern GPUs.

2.6. Data Augmentation

Given limited training data (1,000 images), aggressive augmentation is crucial:

Geometric Transformations:

- Random horizontal flip ($p = 0.5$)
- Random rotation (± 15)
- Random scaling ($[0.9, 1.1]\times$)

Photometric Transformations:

- Color jitter: brightness ($\pm 20\%$), contrast ($\pm 20\%$), saturation ($\pm 10\%$)

Normalization: ImageNet statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$) are applied to leverage pre-training knowledge in convolutional layers.

Critical Implementation Detail: Augmentation is applied *only* to the training set. The validation set uses only center crop and normalization to ensure consistent, reproducible metrics.

3. Experimental Analysis

3.1. Dataset and Implementation

Dataset: CelebAMask-HQ mini with 1,000 training and 100 validation images (512×512), annotated with 19 semantic classes.

Training Details:

- Batch size: 32
- Training epochs: 150 with early stopping
- Hardware: NVIDIA A100 GPU
- Framework: PyTorch 2.0

3.2. Model Optimization Experiments

Architecture Search: We experimented with different encoder configurations:

Configuration	Parameters	F-Score	Speed
Shallow (d=[1,1,1,1])	1.12M	0.72	45 FPS
Medium (d=[1,2,2,2])	**1.72M**	**0.81**	38 FPS
Deep (d=[2,3,3,3])	2.45M	-	-

Table 1. Ablation on encoder depth (d=depths). Medium configuration selected for optimal parameter usage (94.6%).

Loss Function Ablation:

Loss Configuration	F-Score	Boundary IoU
CE only	0.76	0.64
Dice only	0.73	0.71
CE + 0.5×Dice	**0.81**	**0.73**
CE + 1.0×Dice	0.79	0.72

Table 2. Loss function comparison. Combined loss with 0.5 weight achieves best overall performance.

3.3. Regularization Analysis

Data Augmentation Impact:

Augmentation Strategy	F-Score
None	0.68
Geometric only	0.75
Photometric only	0.72
Geometric + Photometric	**0.81**

Table 3. Data augmentation ablation shows complementary benefits.

Learning Rate Schedule:

We compared different scheduling strategies:

- **Constant LR**: Slower convergence, F-Score 0.73
- **Step decay**: Better than constant, F-Score 0.77
- **Cosine annealing**: Best performance, F-Score 0.81

The cosine schedule with warmup provides smooth convergence and avoids local minima.

3.4. Class-wise Performance

Analysis of per-class F-Scores reveals:

- **High accuracy** (¿0.90): Background, skin, hair - large regions with clear boundaries
- **Medium accuracy** (0.75-0.85): Eyes, nose, mouth - moderate-sized regions
- **Challenging** (¡0.70): Earrings, necklace - small, sparse objects with occlusions

The Dice loss component significantly improved performance on smaller classes by reducing the impact of class imbalance.

3.5. Computational Efficiency

Our MicroSegFormer achieves:

- **Parameters**: 1,721,939 (94.6% of 1.82M limit)
- **Inference speed**: 38 FPS on A100 (512×512 images)
- **Training time**: 2 hours for 150 epochs
- **Memory usage**: 4.2 GB GPU memory with batch size 32

The efficient self-attention with spatial reduction enables this performance while maintaining accuracy competitive with larger models.

4. Conclusion

This work presents MicroSegFormer, a lightweight transformer-based architecture for efficient face parsing under strict parameter constraints. Through careful architectural design and comprehensive optimization, we achieve competitive segmentation performance with only 1.72 million parameters (94.6% of the allowed budget).

Key Contributions:

- Efficient hierarchical transformer encoder with spatial reduction attention, reducing computational complexity while preserving accuracy
- Lightweight MLP decoder for effective multi-scale feature fusion
- Combined loss function (Cross-Entropy + Dice) addressing class imbalance and boundary accuracy
- Comprehensive data augmentation strategy maximizing generalization from limited training data

Experimental Insights: Our ablation studies demonstrate that:

1. The combined CE+Dice loss significantly outperforms either loss alone, particularly for small facial regions
2. Data augmentation (both geometric and photometric) is crucial for generalization with limited training data
3. Cosine annealing with warmup provides more stable and effective convergence than step-based schedules
4. The medium-depth encoder configuration (depths [1,2,2,2]) achieves optimal parameter efficiency

Limitations and Future Work: While MicroSegFormer demonstrates strong performance within parameter constraints, several directions could further improve results:

- Test-time augmentation (TTA) for improved prediction robustness
- Advanced post-processing techniques like Conditional Random Fields (CRF) for boundary refinement
- Multi-scale training and inference strategies
- Exploration of knowledge distillation from larger pretrained models

Our implementation demonstrates that transformer-based architectures can be effectively scaled down for resource-constrained scenarios while maintaining competitive performance, making face parsing accessible for edge devices and real-time applications.

Code Availability: The complete implementation including training scripts, model architecture, and inference pipeline is available in the supplementary materials.

References

[1] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021.