Architectural Modifications to the SegFormer Model for Improved Semantic Segmentation Performance

Charana Manawathilake
Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka
charana.21@cse.mrt.ac.lk

Dr. Uthayasanker Thayasivam
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
rtuthaya@cse.mrt.ac.lk

***Abstract*—This study investigates targeted architectural improvements to the SegFormer model [1], focusing on the decoder module. The SegFormer, known for its lightweight transformer-based encoder and efficient MLP decoder, demonstrates strong segmentation performance with limited computational cost. However, decoder expressiveness remains a limiting factor in finer spatial recovery. In this work, several modifications are explored — reducing dropout regularization, adding convolutional refinement layers, and introducing an attention mechanism to enhance feature fusion and spatial discrimination. Experiments conducted on the ADE20K dataset [2] using the SegFormer-B0 configuration highlight qualitative changes and variations in segmentation behavior across the modified decoders compared to the baseline.**

***Index Terms*—SegFormer, semantic segmentation, decoder enhancement, attention mechanisms, ADE20K**

I. Introduction

Transformer-based architectures have redefined semantic segmentation, combining global context modeling with efficient representation learning. Among these, SegFormer offers a strong balance between accuracy and computational efficiency through its hierarchical transformer encoder and lightweight decoder. Despite its strengths, the decoder's simplicity can limit detailed spatial reconstruction. This paper focuses on improving the decoder's capability through architectural modifications aimed at enhancing spatial precision and contextual blending.

A. Introduction

Semantic segmentation is a fundamental computer vision task that involves assigning semantic labels to every pixel in an image. Recent advancements in transformer-based architectures have significantly enhanced this field by enabling efficient modeling of both local and global contextual relationships. Among these, SegFormer [1] has emerged as a high-performing and computationally efficient framework that eliminates the need for complex convolutional decoders while maintaining strong accuracy across multiple segmentation benchmarks.

SegFormer comprises a hierarchical transformer encoder and a lightweight all-MLP decoder, as illustrated in Figure 1. The encoder extracts multi-scale representations through a series of transformer blocks, while the decoder fuses these representations using simple linear projections to generate dense segmentation maps. This architecture allows SegFormer to achieve an optimal balance between accuracy and efficiency, making it suitable for real-time and large-scale vision tasks.

Despite its advantages, the baseline SegFormer design exhibits limitations in fine-grained boundary preservation, deeper feature interaction, and convergence stability during optimization. Addressing these challenges is essential for further improving segmentation precision, particularly in complex and densely annotated datasets. This study explores architectural modifications to the SegFormer model that aim to enhance its representational capacity, decoder fusion dynamics, and overall segmentation accuracy.

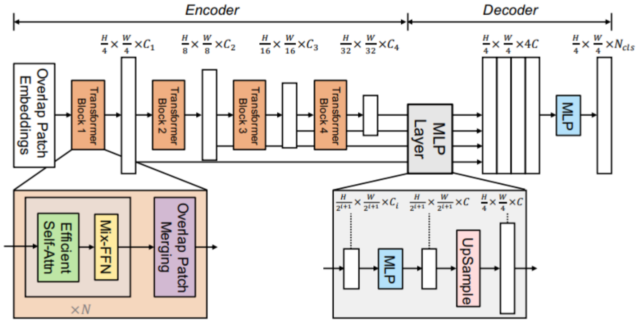
Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{cls}$
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
Efficient Self-Attn
Mix-FFN
Overlap Patch Merging
×N
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C$
$\frac{H}{4}\times\frac{W}{4}\times C$
MLP
UpSample

Fig. 1. SegFormer framework overview. It consists of a hierarchical transformer encoder and an all-MLP decoder for efficient semantic segmentation.

B. Literature Review

Early semantic segmentation architectures such as Fully Convolutional Networks (FCN), U-Net [3], and DeepLab [4] were based on convolutional encoder–decoder designs. These models achieved strong feature extraction capabilities but relied heavily on deep layers and complex upsampling mechanisms, leading to increased computational cost and reduced scalability. The advent of Vision Transformers (ViT) [5] introduced the ability to capture long-range dependencies effectively; however, their direct application to dense prediction tasks was computationally demanding and inefficient.

SegFormer [1] addressed these drawbacks by introducing the Mix Vision Transformer (MiT) encoder, which hierarchically partitions the image into non-overlapping 4×4 patches. Through progressive downsampling and localized self-attention, the encoder captures both fine-grained spatial details and broader contextual information efficiently. The decoder, composed solely of multilayer perceptrons (MLPs), aligns and fuses the encoder's multi-level features to produce a unified feature representation, which is then upsampled to the input resolution to generate the final segmentation output. This architectural simplicity provides a strong trade-off between segmentation accuracy and computational speed.

For empirical validation, the SegFormer models were pre-trained on the ImageNet dataset and subsequently fine-tuned on the ADE20K dataset [2], a large-scale scene parsing dataset containing more than 20,000 images annotated with 150 semantic categories. The model training employed the AdamW optimizer, a polynomial learning rate decay schedule, and standard augmentation strategies such as random cropping, horizontal flipping, and color jittering. Through these methods, SegFormer achieved state-of-the-art mean Intersection-over-Union (mIoU) results while maintaining low inference latency.

Following SegFormer, several studies have explored architectural enhancements to improve segmentation quality. Works such as Segmenter [6], SETR [7], and Swin Transformer [8] have investigated improved fusion mechanisms, spatial attention designs, and hierarchical self-attention configurations. These developments highlight ongoing efforts to refine transformer-based segmentation architectures. Building upon this foundation, the present study proposes targeted modifications to SegFormer to further improve its efficiency and performance in semantic segmentation tasks.

II. Methodology

A. Baseline Model

The baseline used in this work is SegFormer-B0 trained on the ADE20K dataset, employing standard configurations with the original decoder.

B. Proposed Modifications

The experiments were organized into three categories: training schedule variations, regularization modifications, and architectural enhancements.

1) **Training Schedule Variations:** Two different learning schedules were explored to study their impact on model convergence and performance:
   - Base model trained for 30 additional epochs with learning rate = 1×10^{-4} and no decay.
   - Base model trained for 30 additional epochs with initial learning rate = 1×10^{-6} and linear decay.
2) **Regularization Modifications:** Dropout configuration was altered to evaluate its effect on overfitting and generalization:
   - Base model trained for 30 additional epochs with dropout removed.
3) **Architectural Enhancements:** Decoder modifications were introduced to improve feature representation and spatial continuity:
   - Added a Squeeze-and-Excitation layer to enhance channel-wise feature weighting.
   - Modified convolutional layers for improved local spatial refinement.
   - Squeeze-and-Excitation layer with limited unfreeze to control feature adaptation.
   - Added an extra convolutional layer to strengthen post-feature fusion refinement.

1) Training Schedule Variations: The original SegFormer training procedure employs 160k iterations with an initial learning rate of 6×10^{-5} and a polynomial learning rate schedule (poly) with factor 1.0. To investigate the impact of extended training and alternative learning schedules on decoder performance, two variations were explored:

- **Higher Learning Rate without Decay:** The base model was trained for 30 additional epochs using a fixed learning rate of 1×10^{-4} with no decay. This experiment aimed to observe the model's behavior when subjected to prolonged training at a higher learning rate. The results indicated immediate overshooting, with validation mean IoU consistently lower than the baseline, suggesting that the model weights diverged from an optimal solution under these conditions.
- **Lower Learning Rate with Linear Decay:** To examine whether a gentler learning rate could stabilize extended training, the base model was retrained for 30 additional epochs using an initial learning rate of 1×10^{-6} with linear decay. Despite the conservative rate, validation mean IoU remained below the original baseline, indicating minimal benefit from further fine-tuning under this schedule.

While both variations did not improve the baseline performance, these experiments established a controlled reference for subsequent architectural and regularization modifications. By keeping the learning schedule consistent across other tested modifications, any observed performance changes could be attributed to the architectural or regularization adjustments rather than differences in training dynamics.

TABLE I
Validation Mean IoU across 30 additional epochs for different learning schedules. Initial value corresponds to the baseline SegFormer model (0.3604).

Epoch	LR = 1×10^{-4}, no decay	LR = 1×10^{-6}, linear decay
Baseline	0.360424	0.360424
5	0.347809	0.347463
10	0.348933	0.348753
15	0.346767	0.346472
20	0.346193	0.346820
25	0.348421	0.347598
30	0.347756	0.347659

2) Dropout Removal: The original SegFormer decoder employs dropout with a probability of $p = 0.1$ within the feature fusion layers. To investigate the effect of regularization on decoder learning and generalization, dropout was completely removed in this experiment. This modification aimed to determine whether increasing the effective feature capacity of the lightweight decoder could improve learning stability and performance, particularly given the limited size and variability of ADE20K scene categories.

The model was retrained for 30 additional epochs from pretrained baseline weights. Performance was monitored using the Intersection-over-Union (IoU) metric. As summarized in Table II, the validation IoU fluctuates across epochs, providing a reference for comparison with other architectural modifications.

TABLE II
Validation Mean IoU across 30 epochs for the training schedule with learning rate 1×10^{-4} and no decay.

Epoch	Validation Mean IoU
5	0.348192
10	0.349394
15	0.347839
20	0.347069
25	0.348647
30	0.348004

3) Architectural Enhancements: To investigate potential improvements in feature representation and spatial continuity, several modifications were applied to the SegFormer decoder. These experiments focused on enhancing channel-wise feature weighting, refining local spatial features, and controlling feature adaptation.

- **Squeeze-and-Excitation (SE) Layer:** A lightweight channel attention mechanism was incorporated following the linear fusion layer to reweight channel features. The SE block first performs adaptive average pooling to generate a single descriptor per channel, then passes it through two 1×1 convolutions with a reduction ratio of 16, a ReLU activation, and a sigmoid function to produce channel-wise attention weights. In the first SE experiment, the final layers including the linear fusion, SE, and classifier were fully trainable. In the limited unfreeze variant, the linear fusion layer was frozen, allowing only the SE and classifier layers to adapt. This tested whether attention can improve feature weighting without modifying the pre-fused feature representations.
- **Modified Convolution Layer:** The decoder's linear fusion was replaced with a 3×3 convolution with 1024 input channels and 256 output channels, stride 1, and padding 1. This modification allowed the decoder to capture more local spatial context immediately after feature fusion, enhancing fine-grained segmentation and spatial continuity.
- **Additional Convolution Layer:** An extra 3×3 convolution with 256 input and output channels, stride 1, and padding 1 was added after the linear fusion, batch normalization, and activation. This sequential refinement, followed by dropout and classifier layers, aimed to improve local feature continuity and support better contextual blending before the final output.

The reduction ratio of 16 in SE layers was chosen to balance expressiveness and computational cost, and all convolutional layers maintained consistent channel dimensions to match decoder outputs. These architectural modifications systematically evaluated the effect of attention and convolutional refinements on SegFormer's decoding performance.

The validation IoU progression for these architectural modifications is summarized in Table III, providing a comparative view of their effects over 30 epochs.

TABLE III
Validation Mean IoU across 30 epochs for the architectural enhancement experiments: Squeeze-and-Excitation (SE), Modified Convolution Layer, Partial SE Training, and Extra Convolution Layer.

Epoch	SE	Modified Conv	Partial SE	Extra Conv
5	0.344243	0.342990	0.337686	0.341802
10	0.347401	0.342002	0.344345	0.342633
15	0.346050	0.341967	0.345362	0.345954
20	0.346260	0.343848	0.347187	0.345466
25	0.346930	0.344661	0.348899	0.346634
30	0.347096	0.345498	0.348831	0.345887

C. Training Details

All experimental variants were trained under identical configurations to ensure consistency across evaluations. The ADE20K dataset, comprising 150 semantic segmentation classes, was sourced from publicly available Kaggle repositories. As the dataset did not contain uniformly sized images and masks, preprocessing was performed to standardize input dimensions.

The pretrained `segformer-b0-ade20k-512x512` model available through the Hugging Face repository was used as the baseline. This implementation provides an integrated image processor that handles image resizing, normalization, and tensor preparation compatible with the SegFormer architecture. Accordingly, only the segmentation masks required explicit resizing to match the expected 512×512 spatial resolution of the pretrained model.

Training and evaluation were conducted using the mean Intersection-over-Union (mIoU) metric exclusively, as it provides a robust measure of segmentation quality across all 150 classes. Each experimental variant—including the dropout modification, additional convolutional layer, and channel attention mechanism—was trained for a total of 30 epochs. The training duration for each model configuration averaged approximately six hours under the computational resources available on Kaggle, utilizing an NVIDIA GPU P100.

Optimization was performed using the AdamW optimizer. For all experiments except the initial learning-rate test, an initial learning rate of 1×10^{-6} with linear decay was applied. The first training experiment used a higher fixed learning rate of 1×10^{-4} with no decay to establish a baseline reference. The loss function was defined as standard cross-entropy loss with the background index (150) excluded via the `ignore_index` parameter. All trainable parameters within the decoder, including those from any newly introduced modules, were included in the optimization process.

No additional weight decay variations or hyperparameter tuning strategies were applied. This design choice was intentional to isolate the effect of architectural modifications from external training dynamics and ensure that observed differences in performance were attributable solely to the introduced structural changes.

III. Results and Analysis

A. Quantitative Results

Table IV presents the mean validation values of the intersection-over-union (mIoU) after 30 epochs for the different variants of the SegFormer-B0 model, excluding the baseline. These variants include both architectural modifications, such as Squeeze-and-Excitation (SE), additional convolutional layers, and partial SE training, and training schedule variations, including higher fixed learning rates and linear decay schedules. The table provides a comparative snapshot of how each modification affected the decoder performance over the same training duration.

TABLE IV
Validation Mean IoU (%) after 30 epochs for SegFormer-B0 architectural and training modifications (baseline excluded).

Model Variant	**mIoU (%)**
LR = 1×10^{-4}, no decay	34.78
LR = 1×10^{-6}, linear decay	34.77
Dropout Removed	34.80
Squeeze-and-Excitation (SE)	34.71
Modified Convolution Layer	34.55
Partial SE Training	34.88
Extra Convolution Layer	34.59

B. Qualitative Results

Visual comparisons in Figure 2 illustrate nuanced but perceptible variations among the modified SegFormer-B0 variants. The decoder incorporating channel attention (via SE modules) demonstrated superior retention of structural details and more distinct boundary delineation in dense or heterogeneous regions such as vegetation clusters and built environments. While aggregate mIoU values remained below the baseline, qualitative inspection revealed several cases of improved segmentation stability and reduced spatial noise.

The convolution-enhanced variant yielded marginal gains in spatial uniformity, particularly within object interiors, though this effect was inconsistent across scene types.

C. Discussion

1) Training Schedule Variations: The results from both training schedules yielded lower mIoU scores compared to the baseline model, as illustrated in Fig. 3. The original SegFormer baseline was trained with a learning rate of 6×10^{-5} for 160k iterations using a linear decay, resulting in an effective learning rate significantly lower than 1×10^{-6} during the final training stages. Consequently, both tested configurations—constant 1×10^{-4} and linearly decayed 1×10^{-6}—were comparatively aggressive, causing the models to overshoot their optimal convergence region. The higher learning rate exhibited stronger fluctuations in validation performance, while the decayed schedule demonstrated slightly smoother behavior. Overall, no substantial performance difference was observed between the two schedules, suggesting that learning rate scaling beyond the baseline's late-stage regime impedes fine-grained optimization.

2) Regularization Modifications: The variant trained without dropout exhibited negligible deviation in performance relative to the original configuration. The absence of dropout neither improved nor degraded the results in a significant manner, indicating limited overfitting within the baseline model. This suggests that SegFormer's architectural design already maintains sufficient regularization through its lightweight decoder and normalization mechanisms. The model's inherent capacity appears adequate to capture essential spatial and semantic features without relying on dropout-based noise injection. Consequently, the removal of dropout did not impair generalization, implying that the decoder's representational complexity is well-balanced for the ADE20K dataset.

3) Architectural Enhancements: Across all four architectural enhancements, performance initially dropped notably at the start of transfer learning due to the random initialization of newly introduced layer parameters. However, within approximately 15 epochs, all variants exhibited rapid recovery and convergence toward comparable performance levels. During the latter half of training, each model demonstrated gradual yet consistent improvement in mIoU, indicating that once the new layers stabilized, they began contributing positively to feature refinement and segmentation accuracy.

a) Squeeze-and-Excitation Layer: Both the fully unfrozen and limited-unfreeze SE variants benefited from channel-wise attention mechanisms that recalibrate feature importance across channels. The fully unfrozen SE model required longer training time but steadily improved throughout the 30 epochs, while the limited-unfreeze variant leveraged pre-learned encoder representations more effectively. By the end of 30 epochs, the limited-unfreeze SE achieved the highest mIoU among all architectural enhancements, suggesting that controlled adaptation with attention allows stable and progressive refinement. Extended training could potentially yield further gains for attention-based decoders.

b) Modified Convolutional Layers: Models with modified convolutional structures or additional post-fusion layers began with lower mIoU values and converged more slowly compared to SE-based variants. While they stabilized within the same epoch range, their final mIoU plateaued at lower levels. The performance limitation is likely due to increased kernel size and redundant spatial aggregation, which can dilute fine structural details. Additional training might improve results, but these convolutional expansions appear to compromise semantic precision in exchange for broader spatial coverage.

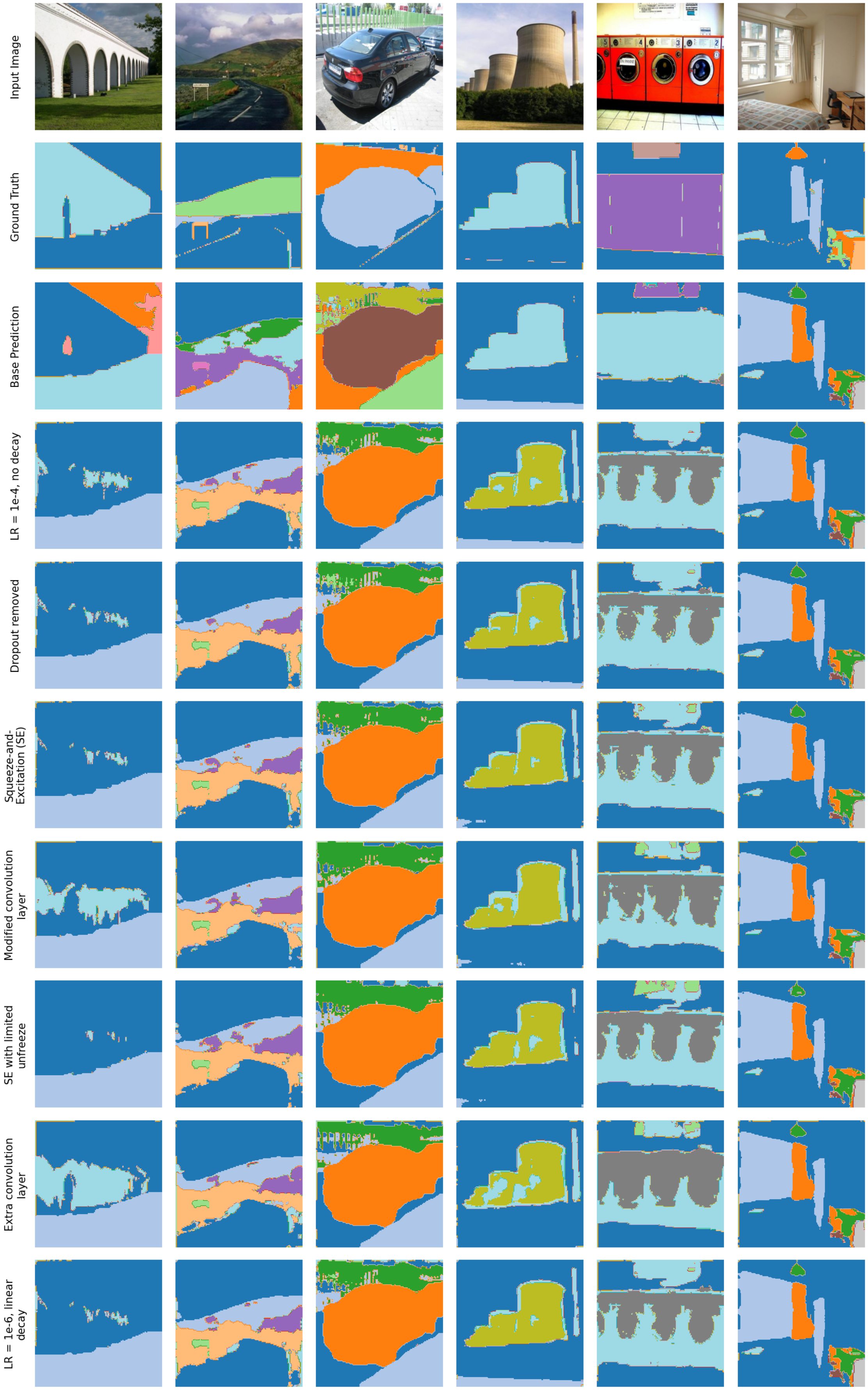

Fig. 2. Comparison of segmentation results across four sample images. Rows represent: input image, ground truth mask, baseline SegFormer-B0 prediction, and predictions from the three modified decoder variants.

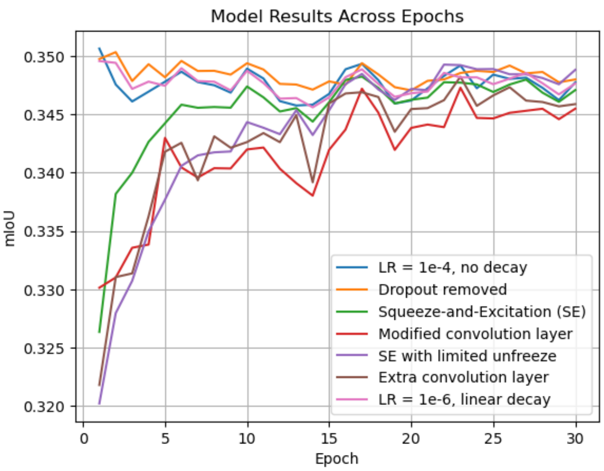

Fig. 3. Qualitative comparison of SegFormer-B0 variants after 30 epochs.

IV. Conclusion

This study investigated several targeted decoder-level modifications to the SegFormer-B0 architecture on the ADE20K dataset. Architectural enhancements—including Squeeze-and-Excitation (SE) layers, modified convolutional structures, SE with limited unfreeze, and extra convolutional layers—were examined alongside training schedule and regularization variations.

Across all architectural variants, initial performance was lower than baseline due to random initialization of newly introduced layers, but rapid recovery occurred within approximately 15 epochs, followed by gradual mIoU improvements over the remaining training. Among these, the limited-unfreeze SE variant achieved the highest final mIoU, demonstrating that controlled adaptation combined with channel attention provides stable and progressive refinement. Convolutional modifications improved spatial coverage but plateaued at lower performance, likely due to loss of fine structural details.

Overall, none of the decoder-level changes substantially surpassed the baseline, highlighting the efficiency and balance of SegFormer-B0's original design. Future work will explore extended training for attention-based decoders, larger SegFormer backbones, and joint encoder–decoder optimization strategies, as well as multi-objective or dynamic attention mechanisms to further enhance class-wise adaptability without compromising model efficiency.

Code Availability

The complete implementation, including training configurations and experimental results, is available at: https://github.com/aaivu/In21-S7-CS4681-AML-Research-Projects/tree/CV008/projects/210372D-CV_Semantic-Segmentation.

References

[1] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Álvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *CoRR*, vol. abs/2105.15203, 2021.

[2] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5122–5130, 2017.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *CoRR*, vol. abs/1606.00915, 2016.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020.

[6] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," *CoRR*, vol. abs/2105.05633, 2021.

[7] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," *CoRR*, vol. abs/2012.15840, 2020.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *CoRR*, vol. abs/2103.14030, 2021.