EffiSeg: A High-Performance Lightweight Segmentation Training Framework via Transformer-Aware Optimization and Targeted Distillation

Dain Kim KiPyo Kim Kyungjun Oh

Sungkyunkwan University, Seoul, South Korea

Abstract

Semantic segmentation is a fundamental task in computer vision, widely used in applications such as autonomous driving, medical imaging, and mobile robotics. However, deploying high-performance segmentation models in resource-constrained environments remains a significant challenge due to their heavy computational and memory requirements. To address this, our work proposes novel training strategies to improve **efficient semantic segmentation** by exploring three distinct approaches:. This work enhance lightweight segmentation model performance via: (1) **model architecture optimization** to reduce training time and computational load through structural modification; (2) **knowledge distillation** to transfer representational power from a large teacher network to a compact student model using a combination of feature-based, logit-based, and patch-based model aware knowledge distillation techniques; and (3) **loss function refinement** to address class imbalance. The proposed training strategies achieve competitive segmentation accuracy while maintaining a lightweight architecture, thereby opening avenues for deployment in real-time and edge-device scenarios. One of our methods achieves up to **+5.69%** mIoU improvement over the SegFormer-B0 baseline on the Cityscapes dataset, without compromising inference speed. The code is available at `https://github.com/lucinnnal/lucinaaal-effiseg`

1 Introduction

Semantic segmentation, the task of assigning a class label to each pixel in an image, is a cornerstone of many real-world applications such as autonomous driving, medical imaging, and mobile robotics. With the advent of Transformer-based architectures, semantic segmentation has achieved remarkable progress in both accuracy and generalization. In particular, SegFormer, introduced by Xie et al.[1], has been widely adopted as a strong baseline model, combining hierarchical vision transformers with lightweight decoders to achieve high performance without heavy computational demands.

SegFormer introduces an efficient architecture for semantic segmentation, particularly beneficial for resource-constrained environments. However, as highlighted in the original study, the smallest variant, SegFormer-B0, exhibits comparatively lower performance than its larger counterparts, such as SegFormer-B3, B4, and B5, across standard benchmarks like Cityscapes[2] and COCO-Stuff[3]. This performance gap suggests that while SegFormer-B0 offers computational efficiency, its accuracy may not suffice for certain real-world applications, indicating a trade-off between model size and segmentation performance.

To overcome these limitations while preserving the efficient design of SegFormer, this work proposes a novel training strategies that improves segmentation accuracy without increasing model complexity. Our method is motivated by three key observations and innovations:

First, it has been observed that the inputs and outputs of the LayerNorm modules preceding attention blocks in the SegFormer encoder exhibit a distribution that resembles the shape of the `tanh` activation function introduced by Zhu et al.[4]. Based on this observation, this work explores the effect of inserting a tanh activation after LayerNorm, as proposed in the cited study. In addition, guided by visualization results obtained in this work, the same modification is also applied to the overlap patch merging blocks in the SegFormer encoder.

Second, recent advances in knowledge distillation have extended beyond the traditional logit-based approaches, introducing more sophisticated strategies to enable student models to acquire richer and more structured knowledge from their teacher counterparts. In the context of dense prediction tasks such as semantic segmentation and object detection, it has become increasingly important to not only mimic the output distributions but also to align internal feature representations. As such, recent research has explored distillation in the feature space, encouraging the student to replicate the spatial and semantic patterns encoded in the teacher's intermediate features [5, 6, 7, 8]. Focusing specifically on the *SegFormer* architecture, which extracts multi-stage feature maps through a hierarchical transformer encoder, we explore a tailored distillation strategy that aligns with the model's unique structural characteristics. Rather than employing a generic distillation framework, we design a distillation approach that leverages SegFormer's multi-scale nature by applying feature-level supervision at multiple stages of the encoder. **Cosine Similarity Based Feature Distillation** is utilized to ensure the student model effectively learns the spatial and semantic patterns of each intermediate feature map. Additionally, this work utilized **Patch Embedding Distillation** method to transfer structural and contextual knowledge from the teacher's embedding space, exploiting the transformer-based nature of SegFormer. Complementing these strategies, logit-based supervision using **Kullback-Leibler (KL) Divergence Based Distillation** is also applied to guide the student toward replicating the teacher's output distribution. By attempting to combine these specialized techniques, our goal is to enable the efficient SegFormer model to inherit the rich multi-level representations of the teacher, thereby achieving high segmentation accuracy without increasing computational overhead. Our work highlights the importance of architecture-aware distillation design and demonstrates that strategically guided knowledge transfer can significantly boost the performance of compact models in real-world applications.

Third, to address severe class imbalance—especially for rare categories such as 'bicycle' in Cityscapes, which accounts for only 0.0115% of total pixels—refined the refined loss function was applied in training process.

Through these contributions, this work presents a novel training strategies that enhances the capacity while maintaining the efficiency of lightweight segmentation models without additional computational burden. Our method achieves competitive performance on standard benchmarks and is nearly suited for deployment in real-time or edge-based environments.

2 Related Work

2.1 Segformer

SegFormer, Xie et al.[1] is a simple, efficient, and powerful semantic segmentation framework that combines a hierarchical Transformer encoder with a lightweight MLP decoder. It has two key characteristics. First, the encoder produces multiscale features without the need for positional encoding, thus avoiding performance degradation when the testing resolution differs from training. Second, instead of relying on complex decoders, SegFormer employs a simple MLP decoder that aggregates features from multiple encoder stages, effectively combining both local and global attention mechanisms. This lightweight and streamlined design has proven to be a crucial factor for enabling efficient and high-performing segmentation using Transformers. This work focuses on leveraging the feature maps and patch embeddings extracted from the multi-stage encoder to improve segmentation performance.

2.2 Dynamic Tanh

Normalization layers such as LayerNorm have long been considered essential components in transformer architectures, stabilizing training, and enabling deeper networks. However, recent work challenges this assumption by introducing Dynamic Tanh (DyT), Zhu et al.[4], a simple element-wise function defined as $\mathrm{DyT}(x) = \tanh(\alpha x)$, which can replace normalization layers entirely. Motivated by the empirical observation that normalization layers often exhibit tanh-like activation behavior, DyT enables normalization-free Transformers to achieve comparable or superior performance across a wide range of tasks, including image recognition, text generation, and both supervised and self-supervised learning, without requiring extensive hyperparameter tuning. This finding questions the indispensibility of normalization and offers new insights into its functional role in deep models. In our work, we adopt

DyT in SegFormer, showing that it not only accelerates training but also improves final performance, further validating the potential of normalization-free architectures in dense prediction tasks.

2.3 Knowledge Distillation

Recent advancements in knowledge distillation (KD) have moved beyond traditional response-based methods, extending toward more nuanced strategies for transferring rich internal representations[5, 6, 7]. In dense prediction tasks such as semantic segmentation and object detection, these innovations are particularly vital due to the spatially structured and high-dimensional nature of the outputs.

Liu et al.[9] introduces a transformer-to-transformer distillation framework explicitly tailored for semantic segmentation. Unlike traditional CNN-to-CNN KD approaches, TransKD takes full advantage of the hierarchical structure of transformer models. It performs comprehensive knowledge transfer from both feature maps and transformer-specific patch embeddings. Two core modules, Cross Selective Fusion (CSF) and Patch Embedding Alignment (PEA), facilitate this process by aligning cross-stage features and harmonizing patch embedding spaces. Moreover, TransKD incorporates the Global-Local Context Mixer (GL-Mixer) and the Embedding Assistant (EA) to bridge the architectural capacity gap between the teacher and student transformers. This multi-perspective approach enables the student model to absorb both spatial and sequential representations effectively, significantly improving segmentation performance.

Park et al.[10] presents an elegant solution to the limitations of MSE-based distillation, especially in object detection. CSKD introduces a hybrid loss function that combines cosine similarity and MSE, allowing the student model to focus more on the directional alignment of feature vectors rather than absolute magnitude. This results in better feature map imitation, particularly in spatially complex tasks. Additionally, CSKD proposes an assistant prediction branch to further enhance response-based distillation. Although primarily developed for object detection, the principles of cosine-guided feature alignment are broadly applicable and inspire our use of cosine similarity loss in feature space for semantic segmentation.

These two works collectively illustrate the importance of holistic, architecture-aware, and learning-efficient KD strategies. Building upon these insights, our proposed framework synergistically integrates patch embedding alignment, cosine-similarity-driven feature supervision to fully exploit the structural properties of SegFormer for efficient and accurate semantic segmentation.

2.4 Loss Function

Class imbalance is a well-known challenge in image segmentation tasks. To address this issue, various loss functions have been proposed.

Focal Loss, introduced by Lin et al. [11], was designed to focus the learning process on hard-to-classify examples by down-weighting the loss assigned to well-classified instances. This approach has shown significant improvements in dense object detection and has since been widely adopted in various segmentation domains.

On the other hand, Dice Loss has gained popularity in segmentation due to its ability to directly optimize for overlap-based evaluation metrics such as the Dice Similarity Coefficient. Milletari et al. [12] proposed the Dice Loss in the context of volumetric segmentation, demonstrating improved performance in handling imbalanced data distributions.

To leverage the strengths of both loss types, several studies have explored their combinations. Roy et al. [**?**] introduced Combo Loss, which integrates Dice and Cross-Entropy losses to address both input and output imbalance. Similarly, Tversky Loss, proposed by Salehi et al. [13], generalized the Dice formulation by introducing tunable parameters that allow balancing between false positives and false negatives.

Building on these ideas, Abraham and Khan [14] proposed the Focal Tversky Loss, which incorporates a focal mechanism into the Tversky formulation to further emphasize hard examples. This loss function has been shown to improve performance, especially in cases with significant class imbalance.

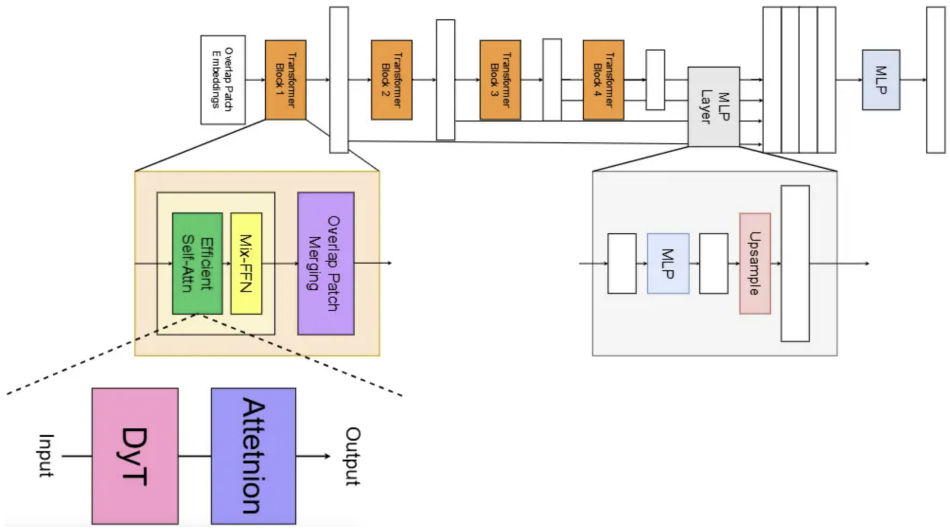

Figure 1: Dynamic Tanh

Inspired by these approaches, our work adopts a hybrid Dice-Focal loss to improve segmentation performance under class-imbalanced conditions across diverse datasets.

3 Method

3.1 Dynamic Tanh and Dynamic Linear Normalization

Dynamic Tanh (DyT) as a drop-in replacement for normalization layers, as shown in Figure 1, and this work apply it to semantic segmentation tasks. Given an input tensor x, DyT is defined as:

$$\mathrm{DyT}(x) = \gamma \cdot \tanh(\alpha x) + \beta, \tag{1}$$

where $\alpha \in R$ is a scalar parameter shared across all channels, and $\gamma, \beta \in R^C$ are learnable affine parameters applied per channel. Although DyT was originally intended to replace the LayerNorm preceding the attention block in Transformer architectures, the non-linear activation patterns resembling the tanh function were most prominent immediately after the OverlapPatch embedding module. In contrast, the other LayerNorm layers exhibited nearly linear activation behaviors. Based on this observation, we apply DyT exclusively to the normalization layer following the OverlapPatch embedding module. For all remaining normalization layers, this work adapt a linear variant called **Dynamic Linear (DyL)**, defined as:

$$\mathrm{DyL}(x) = \gamma \cdot \alpha x + \beta, \tag{2}$$

where $\alpha \in R^C$ is a *channel-wise* learnable parameter. this work choose to make α channel-wise in DyL due to the unbounded nature of linear functions; this allows each channel to be scaled independently, ensuring better numerical stability and adaptive behavior. In contrast, DyT's use of tanh constrains outputs within $[-1, 1]$, allowing stable learning even with a scalar α. This hybrid approach enables dynamic and context-sensitive normalization: non-linear scaling where needed (after OverlapPatch), and efficient linear scaling elsewhere, adapting to the activation characteristics observed across different stages of the model.

3.2 Knowledge Distillation

To enhance the performance of the lightweight segmentation model, a multi-faceted knowledge distillation (KD) strategy was employed. As Segformer employs a hierarchical multi-stage transformer architecture, where each stage extracts feature maps of different resolutions via patch embedding. This structural trait enables a multi-level knowledge distillation framework, facilitating both low-level and high-level representation transfer. The following four loss function terms were utilized, each playing a distinct role in transferring specific aspects of knowledge from teacher to student.

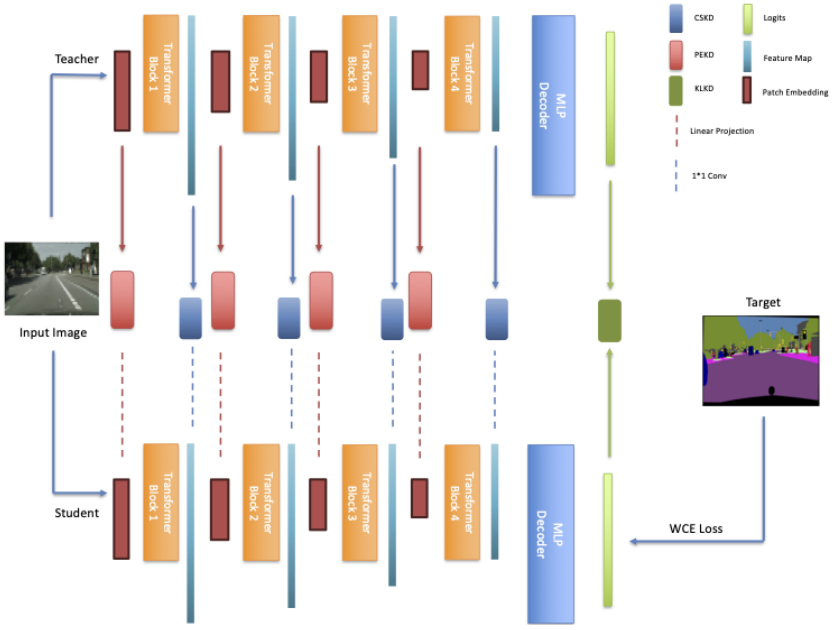

Figure 2: **Overview of KD.** Four types of loss terms are used for knowledge distillation. Class-Weighted Cross-Entropy Loss is applied for standard label supervision. Cosine Similarity Guided Feature Distillation (CSKD) is also employed stage-wise to align the spatial and structural representations, enhancing intermediate feature alignment. Patch Embedding Knowledge Distillation (PEKD) is also used at each stage to transfer low-level spatial representations from the teacher's patch embeddings to the student. Additionally, KL Divergence-Based Knowledge Distillation (KLKD) is used in the logit space to guide the student toward the teacher's output distribution, encouraging better semantic consistency.

3.2.1 Class-Weighted Cross-Entropy Loss

This component acts as the foundation for training by directly supervising the student with ground-truth annotations. A class-weighted cross-entropy loss is employed to account for class imbalance, ensuring that under-represented classes contribute proportionally to the loss. This helps the student model align with explicit human-annotated semantic targets, setting a baseline for semantic correctness and enforcing the primary segmentation objective.

$$\mathcal{L}_{\text{WCE}} = -\sum_{i=1}^{C} w_i \, y_i \log(\hat{y}_i) \tag{3}$$

3.2.2 KL Divergence-Based Knowledge Distillation (KLKD)

In addition to hard labels, the soft predictions (logits) from the teacher provide valuable information such as class ambiguity and inter-class relationships. This component aims to align the student's output distribution with that of the teacher using a pixel-wise KL divergence. By mimicking the teacher's soft logits, the student learns to generalize better, especially in complex scenes with overlapping or ambiguous class boundaries.

$$\mathcal{L}_{\text{KL}} = \sum_{h,w} \sum_{c=1}^{C} p_t^{(c,h,w)} \log \left(\frac{p_t^{(c,h,w)}}{p_s^{(c,h,w)}} \right) \tag{4}$$

3.2.3 Cosine Similarity Guided Feature Distillation (CSKD)

Instead of simply matching feature magnitudes, we employed **Cosine Similarity Guided Feature Distillation (CSKD)** to emphasize the alignment of directional feature vectors between the teacher

and student models. Specifically, cosine similarity was used to measure both channel-wise and spatial-wise directional consistency. To further enhance the effectiveness of supervision, we introduced a magnitude-based guidance map derived from the feature differences. This map serves as a dynamic mask, placing more emphasis on regions where the student features deviate from the teacher. As shown in Equation 7, the final distillation loss $\mathcal{L}_{\text{FD}}$ consists of three terms:

1. a magnitude-based regression loss weighted by the guidance map
2. a spatial-wise cosine similarity loss
3. a channel-wise cosine similarity loss

To compute this, cosine similarity S between teacher feature F^T and student feature $\hat{F}^S$ is given by:

$$S = \frac{\sum F^T \cdot \hat{F}^S}{\sqrt{\sum (F^T)^2} \cdot \sqrt{\sum (\hat{F}^S)^2}} \tag{5}$$

To enable meaningful comparison between the teacher and student feature maps at each stage, the student feature maps were first projected to match the teacher's dimensions using 1×1 convolution. Channel-wise and spatial-wise cosine similarity maps $S^C \in R^{H \times W}$ and $S^S \in R^C$ have been generated respectively. These are used to form a guidance map:

$$G = \left| (1 - S^C(F^T, \hat{F}^S)) \cdot (1 - S^S(F^T, \hat{F}^S)) \right| \tag{6}$$

This guidance map de-emphasizes regions where similarity is already high and focuses the loss on challenging regions. The total feature distillation loss is:

$$\mathcal{L}_{\text{FD}} = \sum_{l=1}^{L} \alpha_l \left[\frac{\lambda^m}{C_l H_l W_l} \sum_{k=1}^{C_l} \sum_{j=1}^{H_l} \sum_{i=1}^{W_l} G_{ijk}^{(l)} \cdot \left(F_{ijk}^{T(l)} - \hat{F}_{ijk}^{S(l)} \right)^2 + \frac{\lambda^S}{H_l W_l} \sum_{j=1}^{H_l} \sum_{i=1}^{W_l} \left(1 - S_{ij}^{C(l)} \right) + \frac{\lambda^S}{C_l} \sum_{i=1}^{C_l} \left(1 - S_i^{S(l)} \right) \right] \tag{7}$$

where λ^m and λ^S are hyperparameters that control the contributions of the guided MSE and cosine similarity losses, respectively. To ensure balanced supervision across stages, each of the four stages contributes equally to the total loss with stage-wise weights α_l uniformly set to 1.

3.2.4 Patch Embedding Knowledge Distillation (PEKD)

In transformer-based architectures, patch embeddings serve as the input tokens to subsequent transformer layers. Thus, aligning these embeddings ensures consistent encoding of spatial structure between student and teacher. This component minimizes differences in token representations via L2 loss, after aligning the student's embeddings to the same dimensional space as the teacher's via linear transformation. Because later stages encode more abstract semantics, progressively larger weights $\beta = [0.1,\ 0.1,\ 0.5,\ 1.0]$ were applied to the loss terms across the four stages.

$$\mathcal{L}_{\text{PE}} = \sum_{i=1}^{L} \beta_i \left\| P_s^{(i)} - P_t^{(i)} \right\|_2^2 \tag{8}$$

3.2.5 Total Loss

The final training objective aggregates all loss components as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{WCE}} + \lambda^{\text{KL}} \mathcal{L}_{\text{KL}} + \lambda^{\text{FD}} \mathcal{L}_{\text{FD}} + \lambda^{\text{PE}} \mathcal{L}_{\text{PE}} \tag{9}$$

This composite loss framework captures diverse semantic cues, hierarchical representations, and probabilistic knowledge, enabling the student to closely approximate the teacher's performance while retaining architectural efficiency.

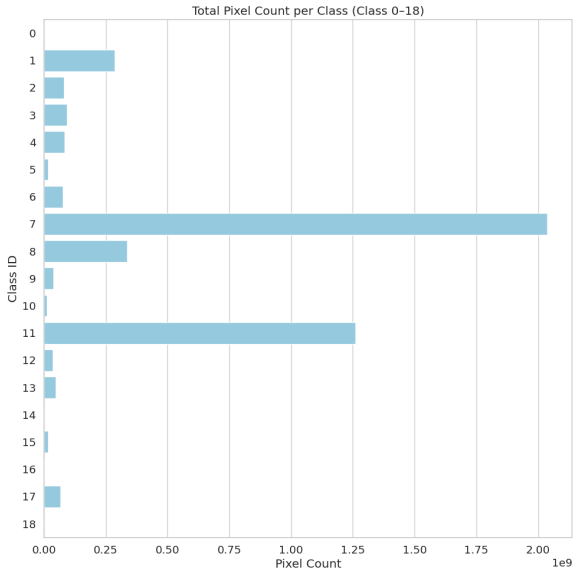

Figure 3: Class pixel distribution.

Class ID	Class Name	Ratio (%)	Weight
0	road	0.02	2.5959737
1	sidewalk	7.09	6.741505
2	building	2.02	3.5353868
3	wall	2.33	9.866315
4	fence	2.08	9.690922
5	pole	0.44	9.369371
6	traffic light	1.87	10.289124
7	traffic sign	50.48	9.953209
8	vegetation	8.33	4.3098087
9	terrain	0.97	9.490392
10	sky	0.28	7.674411
11	person	31.26	9.396925
12	rider	0.90	10.347794
13	car	1.20	6.3928986
14	truck	0.01	10.226673
15	bus	0.44	10.241072
16	train	0.08	10.28059
17	motorcycle	1.68	10.396977
18	bicycle	0.01	10.05567

Table 1: Cityscapes 0–18 Class Labels with Pixel Ratios and Training Weights

3.3 Loss Function Refinement

The Cityscapes dataset presents a significant challenge in terms of class imbalance (see Figure 3). While it includes 19 commonly used semantic classes for urban scene parsing, the pixel distribution among these classes is highly skewed. Dominant classes such as *road*, *building*, and *sky* occupy large regions in most images, accounting for a substantial portion of the total pixel count. In contrast, classes like *traffic light*, *pole*, *rider*, and *bicycle* appear sparsely and occupy relatively few pixels. This imbalance leads to a bias during training, where models tend to prioritize the majority classes and often neglect the minority ones. As a result, the segmentation performance on small or rare objects is significantly lower. Therefore, handling class imbalance effectively is essential for training robust and generalizable models on the Cityscapes dataset.
To address class imbalance in cityscapes, class distribution aware weighted Dice-Focal loss was adopted which is designed to mitigate the impact of dominant classes and enhance learning for rare ones. By incorporating this in training process, we aim to guide the network toward a more balanced representation and improve segmentation accuracy across all classes, especially focusing more on those with low pixel presence.

Dice Loss Dice Loss is particularly effective in tasks with class imbalance, such as semantic segmentation where certain classes occupy significantly fewer pixels than others. It is derived from the Dice Similarity Coefficient (DSC), a measure of overlap between the predicted segmentation and the ground truth. Given predicted soft probabilities p_i and one-hot encoded ground truth labels g_i over N pixels, the Dice Loss is defined as:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{N} p_i g_i + \epsilon}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} g_i + \epsilon} \tag{10}$$

where ϵ is a small constant added for numerical stability. For multi-class segmentation, the Dice loss is typically computed per class and averaged over all C classes:

$$\mathcal{L}_{\text{Dice}} = \frac{1}{C}\sum_{c=1}^{C}\left(1 - \frac{2\sum_{i=1}^{N} p_i^{(c)} g_i^{(c)} + \epsilon}{\sum_{i=1}^{N} p_i^{(c)} + \sum_{i=1}^{N} g_i^{(c)} + \epsilon}\right) \tag{11}$$

This loss encourages the model to focus on overlapping regions, making it particularly useful for improving segmentation of underrepresented or thin classes.

Focal Loss Focal Loss was introduced to address the problem of class imbalance by down-weighting the contribution of easy examples and focusing learning on hard negatives. It modifies the standard cross-entropy loss by adding a modulating factor $(1-p_t)^\gamma$ to the loss, where p_t is the model's estimated probability for the correct class. The loss for a single example is formulated as:

$$\mathcal{L}_{\text{Focal}} = -\alpha(1-p_t)^\gamma \log(p_t) \tag{12}$$

Here, $\alpha \in [0,1]$ is a weighting factor for balancing classes, and $\gamma \geq 0$ is the focusing parameter that controls how much to down-weight easy examples. As shown in Table 1, each class weight was assigned to α_c based on the observed class imbalance. For multi-class problems, Focal Loss can be extended as follows:

$$\mathcal{L}_{\text{Focal}} = -\sum_{c=1}^{C} \alpha_c(1-p_c)^\gamma y_c \log(p_c) \tag{13}$$

where y_c is the one-hot encoded ground truth and p_c is the predicted probability for class c. In semantic segmentation, Focal Loss is particularly effective in scenarios where dominant classes (e.g., road, building) overshadow minority classes (e.g., poles, signs), as it emphasizes learning from hard-to-classify pixels.

Dice-Focal Loss To address the limitations of using either Dice Loss or Focal Loss alone in the presence of severe class imbalance, this work adopts a composite loss function that linearly combines both:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{dice}} \cdot \mathcal{L}_{\text{Dice}} + \lambda_{\text{focal}} \cdot \mathcal{L}_{\text{Focal}} \tag{14}$$

where $\lambda_{\text{dice}} \in [0,1]$ and $\lambda_{\text{focal}} = 1 - \lambda_{\text{dice}}$ are user-defined weighting coefficients that determine the relative contribution of each component. This formulation enables flexible trade-offs between optimizing for region-level accuracy and hard-pixel discrimination.
Dice Loss effectively promotes spatial alignment between predicted masks and ground-truth labels, making it suitable for tasks that require accurate boundary preservation and shape reconstruction. However, it can still be biased toward dominant classes due to their larger spatial extent. Conversely, Focal Loss emphasizes misclassified or rare examples by adaptively down-weighting well-classified pixels, thus mitigating the class frequency bias.
By combining these two complementary objectives, the Dice-Focal Loss leverages the strengths of both region-based and pixel-wise supervision. This synergistic approach encourages the network to learn precise segmentation boundaries while simultaneously focusing on hard-to-classify or underrepresented classes.

4 Experiment

4.1 Dataset

We conducted our experiments using the Cityscapes dataset, a widely adopted benchmark for semantic segmentation tasks. This dataset is publicly available and has become a standard choice for evaluating segmentation algorithms, making it a suitable candidate for our study.

4.2 Model Selection

For experiments, Segformer-B0 model was employed for training. Segformer belongs to the transformer-based family of architectures and is known for its balance between efficiency and accuracy. The B0 variant, in particular, offers a lightweight design with fast inference speed, aligning well with our goal of efficient semantic segmentation.
In knowledge distillation experiments, Segformer-B0 was selected as the student model and Segformer-B2 as the teacher model. While larger variants such as B3, B4, and B5 exist, B2 was used as a teacher since a large disparity in model capacities can negatively affect the effectiveness of knowledge transfer. Both Segformer-B0 and B2 were trained from scratch in all methods without using pre-trained weights.

4.3 Experimental Environment

All experiments were conducted using an NVIDIA GeForce RTX 3090 GPU. Image resolution of 512×1024 pixels was used for both training and evaluation. The training was configured with a batch size of 2 per device for both training and evaluation. Models were trained for 200 epochs using the AdamW optimizer with a learning rate of 6×10^{-4}, $\beta = (0.9, 0.999)$, and $\epsilon = 1 \times 10^{-8}$. A polynomial learning rate scheduler was applied with a warmup ratio of 0.05. Based on empirical tuning, the loss weights were set as follows: $\lambda^{\text{KL}} = 0.0$, $\lambda^{\text{FD}} = 1.0$, and $\lambda^{\text{PE}} = 1.0$ for knowledge distillation and $\lambda_{\text{dice}} = 0.5$, and $\lambda_{\text{focal}} = 0.5$ for loss refinement.

4.4 Results

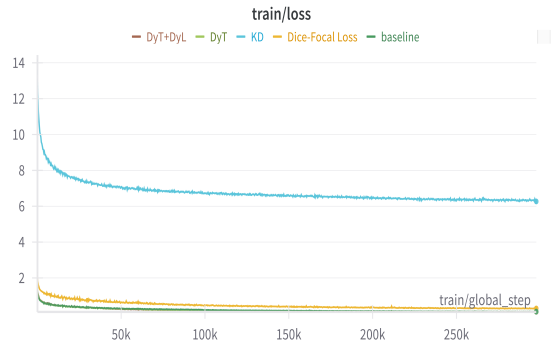

Figure 4: Training Loss Curve.

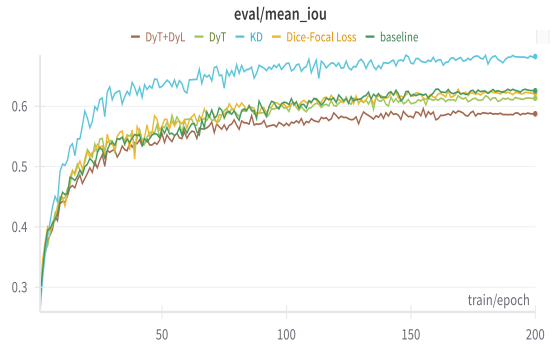

Figure 5: Evaluation mIoU per Epoch.

Network	#Params (M)	FLOPs (G)	mIoU (%)	FPS
SegFormer-B2	27.36	114.09	76.49	35.58
SegFormer-B0 (Baseline)	3.72	13.74	62.56	119.48
+ Tanh	3.72	13.74	60.10	119.61
+ Tanh + Linear	3.72	13.73	59.646	**126.22 (+6.74)**
+ KD	3.72	13.74	**68.25 (+5.69)**	119.48
+ Dice-Focal Loss	3.72	13.74	62.15	119.48

Table 2: Cityscapes eval mIoU on SegFormer variants with different training strategies.

As shown in Table 2, our knowledge distillation (KD) method achieves a mean Intersection-over-Union (mIoU) of **68.25%**, indicating an absolute improvement of **+5.69%** over the SegFormer-B0 baseline (**62.56%**) without increasing the model size. The number of parameters remains the same at **3.72M**, and the computational cost in terms of FLOPs also remains virtually unchanged at **13.74G**. This demonstrates the strength of the proposed training strategy, particularly when compared to the teacher model SegFormer-B2, which, while achieving a higher mIoU of **76.49%**, requires **7.4**× more parameters and operates at **3.4**× slower inference speed (**35.58 FPS** vs. **119.48 FPS**).

Among the tested training strategies, applying DYT specifically to the overlap patch merging components while using DTL for the remaining layers. This strategic combination resulted in a modest decrease in mIoU (from 62.56% to 59.646%) but achieved a notable FPS improvement of 6.74, reaching 126.22 FPS compared to the baseline's 119.48 FPS. The experimental results demonstrate that selective application of dynamic activation functions can effectively balance accuracy and inference speed.

The Dice-Focal Loss strategy, designed to mitigate class imbalance, achieved an mIoU of **62.15%**, slightly below the baseline, suggesting that more sophisticated task-specific loss function tuning is crucial for semantic segmentation tasks.

From the training loss curves (Figure 4), it is observed that the KD-based method exhibits a higher overall loss during training. This is attributed to the inclusion of multiple loss components, such as feature-level distillation and patch embedding-level distillation, which add complexity to the learning

process. However, this increased loss does not hinder final performance; rather, it contributes to better feature alignment and ultimately results in superior mIoU performance during evaluation (Figure 5).

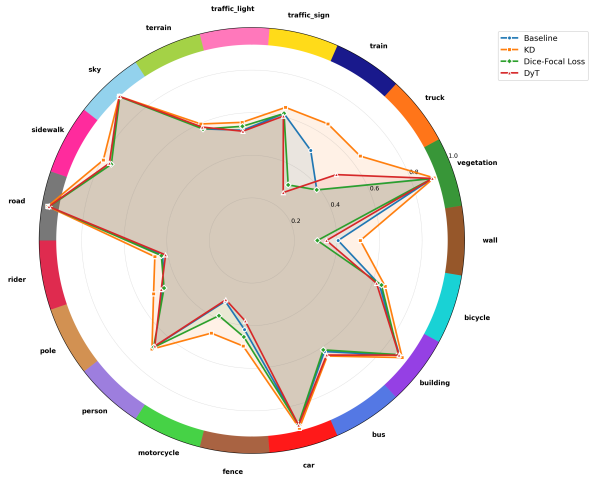

Figure 6: Class-wise IoU Comparison

Figure 6 presents a comprehensive radar chart comparing the class-wise IoU performance across different learning strategies: Baseline, KD, Dice-Focal Loss, and DyT. The visualization reveals distinct performance patterns that provide valuable insights into the effectiveness of each approach.
Figure 6 demonstrates a clear bimodal distribution in class-wise performance. High-performing classes (*e.g.*, road, sky, building) consistently achieve IoU scores exceeding 0.8 across all methods. These classes typically correspond to large-scale objects with distinctive visual characteristics, making them relatively straightforward for segmentation model to identify and delineate. Conversely, challenging classes such as motorcycle, bicycle, and rider exhibit significantly lower IoU scores, reflecting the inherent difficulty in segmenting small objects with complex geometries or those that are easily confused with background elements.
Our KD method demonstrates superior performance across the majority of object classes, with the radar chart showing a consistently expanded coverage area compared to the baseline. Notably, KD achieves substantial improvements for challenging classes observed for train, truck categories. This improvement pattern suggests that our knowledge distillation mechanism effectively transfers semantic understanding from teacher networks, particularly benefiting the recognition of small-scale objects. The balanced improvements across both easy and difficult classes suggest that our KD method enables the teacher network to effectively transfer semantic knowledge to the student network, enhancing feature discrimination across all object categories, including those that are under-represented.

5 Conclusion

This work proposes **EffiSeg**, a high performance semantic segmentation model training strategies for building efficient segmentation models with improved performance. Rather than relying on generic approaches, we designed model-specific strategies that align with the hierarchical transformer encoder structure, including tanh-inspired transformations, stage-wise feature and patch embedding distillation, and tailored loss functions to address class imbalance. One of our methods achieves up to **+5.69%** mIoU improvement over the SegFormer-B0 baseline on the Cityscapes dataset. This result demonstrates that leveraging model-specific training strategies can yield high-performance yet efficient segmentation models, opening up the possibility for real-time and edge deployment.
Moreover, we believe that an effective integration of the three proposed strategies can lead to even greater improvements in performance. Investigating the synergistic combination of these approaches presents a promising direction for future research.

References

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

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[3] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1209–1218.

[4] J. Zhu, X. Chen, K. He, Y. LeCun, and Z. Liu, "Transformers without normalization," *arXiv preprint arXiv:2503.10622*, 2025.

[5] A. Lopes, F. P. dos Santos, D. de Oliveira, M. Schiezaro, and H. Pedrini, "Computer vision model compression techniques for embedded systems: A survey," *Computers & Graphics*, vol. 123, p. 104015, 2024.

[6] C. Xu, N. Bai, W. Gao, T. Li, M. Li, G. Li, and Y. Zhang, "Multiple-stage knowledge distillation," *Applied Sciences*, vol. 12, no. 19, p. 9453, 2022.

[7] A. Amirkhani, A. Khosravian, M. Masih-Tehrani, and H. Kashiani, "Robust semantic segmentation with multi-teacher knowledge distillation," *IEEE Access*, vol. 9, pp. 119 049–119 066, 2021.

[8] P. Chen, S. Liu, H. Zhao, and J. Jia, "Distilling knowledge via knowledge review," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5008–5017.

[9] R. Liu, K. Yang, A. Roitberg, J. Zhang, K. Peng, H. Liu, Y. Wang, and R. Stiefelhagen, "Transkd: Transformer knowledge distillation for efficient semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[10] S. Park, D. Kang, and J. Paik, "Cskd: Cosine similarity-guided knowledge distillation for robust object detectors," 2024.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[12] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *arXiv preprint arXiv:1707.03237*, 2017.

[13] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Tversky loss function for image segmentation using 3d fully convolutional deep networks," *arXiv preprint arXiv:1706.05721*, 2017.

[14] N. Abraham and N. M. Khan, "A novel focal tversky loss function with improved attention u-net for lesion segmentation," *arXiv preprint arXiv:1810.07842*, 2019.

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

[16] Y. Guo, G. Nie, W. Gao, and M. Liao, "2d semantic segmentation: Recent developments and future directions," *Future Internet*, vol. 15, no. 6, p. 205, 2023.

[17] A. M. Mansourian, R. Ahmadi, M. Ghafouri, A. M. Babaei, E. B. Golezani, Z. Y. Ghamchi, V. Ramezanian, A. Taherian, K. Dinashi, A. Miri *et al.*, "A comprehensive survey on knowledge distillation," *arXiv preprint arXiv:2503.12067*, 2025.

[18] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2604–2613.